Paths Towards Reliable Explainability In Neural Networks for Image-Processing

Doctoral Dissertation by
Justus Sagemüller

Thesis submitted for
the degree of Philosophiae Doctor (PhD)
in
Computer Science:
Software Engineering, Sensor Networks and Engineering Computing

Department of Computer Science,
Electrical Engineering and Mathematical Sciences

Faculty of Engineering and Science

Western Norway University of Applied Sciences

October 25, 2023

Series of dissertation submitted to
the Faculty of Engineering and Science,
Western Norway University of Applied Sciences.

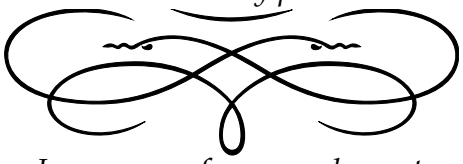
ISBN, print version: 978-82-8461-063-4
ISBN, digital version: 978-82-8461-064-1

Author: Justus Sagemüller
Title: Paths Towards Reliable Explainability
In Neural Networks for Image-Processing

Printed production:
Aksell / Western Norway University of Applied Sciences

Bergen, Norway, 2024

To all who are deservedly proud to be human



In memory of my grandparents

PREFACE

The author of this thesis has been employed as a Ph.D. research fellow at the Departement of Computer Science, Electrical Engineering and Mathematical Science at Western Norway University of Applied Sciences, Norway. The author has been enrolled into the PhD programme in Computer Science: Software Engineering, Sensor Networks and Engineering Computing.

Part of the research presented in this thesis has been accomplished in cooperation with the Division of Mathematical Statistics at the Department of Mathematics in KTH Royal Institute of Technology, Stockholm, Sweden.

This thesis is organised in five parts: Part I gives a general introduction, Parts II, III and IV concern the three broad and interconnected research topics (each with its own introduction), and Part V discusses them together.

Scientific environment

The idea for the saliency topic came from Olivier Verdier (HVL) who followed and counseled most of the research of the entire thesis.

Slobodan Drazic (HVL) suggested use of SIFT, and also provided valuable input on some further details of Part III.

The Cryo-EM project of Part IV is lead by Joakim Andén (KTH), who provided the synthetic data and the (cartesian) reference denoiser models.

Most of the path optimisation runs were carried out on the HVL cluster, administered by Kyrre Skjerdal and Ilker Meric. The denoiser models were trained on C3SE Alvis at Chalmers University, Gothenburg, with funding through the Wallenberg AI, Autonomous Systems and Software Program.

ACKNOWLEDGMENTS

Without Olivier, I would not only not have done this PhD topic, but would not have come to Norway either. Even after he moved away from Bergen, he was very engaged in helping me with the research. We did not always agree, but he ever managed to find ways to achieve constructive compromises. For all of that I owe him a prime-spot thank.

Volker was always reachable when help was needed, as well as ready to open up international collaboration opportunities (though ultimately nothing came of those).

Slobodan was great in his literature suggestions and his eagerness to follow along with the project.

Joakim managed to open up the PhD project again when it was in a rather stalled phase, by interrupting the saliency struggle with completely fresh thoughts.

Violet, Lars-Petter, Rogardt and Pål made my teaching duties not only as bearable as possible (despite the inevitable flood of student problems), but indeed oftentimes seriously interesting and a good learning experience for me as well.

Håvard was about the most friendly and helpful coordinator of a PhD programme imaginable.

Within HVL I want to further give thanks to the PhD-stipendiat community. Special mentions within "my generation": Fatemeh, Michele – I can't believe I'm finishing before you! – for many a tour, adventure and celebration, or just conversation in between.

To some of the "elder" – Faustin, Rui, for good feelings of welcome, Frikk for making the office feel like an actually cool place, and particularly Patrick also for being quite some role model. (Monograph, Nynorsk, injuries in Voss...)

Amongst the "newer", thanks to Ole for the best programmer's nerding-out ever, as well as some really memorable pub evenings; to Tim and to Aurora for being such to-the-point and positive characters, as well as more outdoors company; to Rizwan for good humour and serious thoughts; to Amir, Guy and Mira for bringing life to the office.

And to all the rest for being such a diverse and fun bunch!

A huge thanks to all of Audhilds, most of whom probably have no idea how important they and the music were to me in these Bergen years. Irene, for being an amazing lagsleiar; lately Johanna as well. Christian, for organising the best quarters, grooves and always an interesting conversation. Helge for humour and musical inspiration. Robin for linguistic inspiration.

And with definitely special mentions, Else – nobody else could have made me appreciate the Norwegian nature *that* well – and of course Ingrid, not only for proof-reading my inspired-yet-wrong nynorsk, but moreso for the most wondrous music and so much else.

Finally for some blast from the past and that deep southern country, what was it called... Germany!

Without my parents, I wouldn't be here either. ...In general, and on doctoral course specifically – which may seem odd but I really don't think any other could have supported me in the way that would have worked. Not to forget also my step-parents, who were just as much family and as good family as all the family, all of whom deserve a thank. (And maybe a sorry, for leaving for the North.)

Unsurprisingly important too were some of my teachers and professors, of which I wish to mention particularly Herrn Feld, Herrn Dietz, Rochus Klesse and Joachim Saur. Many others were influential too, in all kinds of different ways.

I don't know if my friends were technically speaking important for the PhD, but they certainly feel important to me. From the start – Dominik, sorry I'll never hold any Nobel acceptance speeches but I can thank you here; David, Martin and Elmar for bringing me through childhood; Ferdinand for early on a sensible engineer's support to a crazy theoretician's ideas; of course Steven for unwavering sound- and joke waves and much good advice; Kerstin, Martin, Corinna, Aglaja, who I only much later realised how supportive they were.

Moving on, we get to the *Füsicists* – Hanno, Henriette, Justus and Matthias, but only with the *Geos* did the future take shape. Christian, Sascha, Mira: you all beat me to the PhD, which maybe says something about what a good springboard for an academic career that community of ours was! It was also just a really fun time.

Here we go. Even without the people I will now inevitably have forgotten to mention[1] – that's a lot of people!

Let's hope this thesis will do their good support due justice.

[1]And some I haven't actually forgotten: Dime Chime, count yourselves amongst "family"!

ABSTRACT

Machine learning systems (often referred to as AI, not always appropriately) are increasingly used in varied applications, including ones with strong impact on human lives. While this is expected to bring economic and scientific progress, it also has several controversial aspects. A major one of these is that black-box models make it hard or impossible to answer questions regarding e.g. the stability of an output or the influence of biases in a training dataset, let alone to rigorously reason about correctness. It is meanwhile known that AI systems can and do often produce convincing yet wrong or misaligned outputs, with a significant potential for detrimental impacts on society.

Better understanding of such systems is therefore needed. The two main approaches to this are: finding explanations for the decisions of existing models, or designing models specifically to be more interpretable. This thesis investigates use of mathematical methods towards both of these goals. We provide a toolkit that improves on existing saliency methods for highlighting parts of images that are important for a classifier on these images. Main contribution is the Ablation Path formalism, which generates perturbations of inputs in a way that is convenient for a human to inspect and assess the faithfulness of the explanation. Additionally we propose a new way of using the SIFT technique as a feature basis for saliency. This overcomes some of the technical challenges with existing methods, and also provides information that can be argued to be more useful than the standard location-heatmaps.

More towards the interpretability front, we study a use case of machine learning denoising in which *symmetries* play a crucial role: Cryo-EM. Symmetries are a known aspect of many neural networks and their applications; for Cryo-EM these can be unusually well exploited and quantified. We propose a variation of convolutional network that is dedicated to the particular symmetries of the application, and investigate how this impacts the performance and other properties.

These contributions push the state of the art for explainability of image classification, and also provide a starting point for multiple further advances on both explainability and interpretability in this application and others.

SAMANDRAG

Bruken av maskinlærings-system (ofte kalla KI, utan at det alltid nødvendigvis er passande) er aukande i ulike anvendingsområde. Dette inneber nokre felt som har sterk innverknad på liva til menneska. Forventninga er at desse systema kjem til å skapa økonomisk og vitskapleg framsteg, men det er også kontroversielle aspekt knytt til bruken av maskinlærings-system. Blant desse er det faktum at svart-boks modellar gjer det vanskeleg eller umogeleg å gje svar på spørsmål om, til dømes, stabiliteten til eit resultat eller kva slags innflytelse data-fordommar har hatt. Og ein kan snautt bevise noko om riktigheita. Imedan er det kjennt at KI-system kan tilverka resultat som er overbevisande men usanne eller misvisande, og at det faktisk skjer i mange tillfelle. Dette har betydeleg evne for negativ påverknad på samfunnet.

Difor trengst det betre forståing av slike system. Dei to hovudtilnærmingsmåtane til dette er: å finna forklaringar for avgjerslene til modellar som allereie finst, eller å utvikla nye modellar med hensikt i å vera interpreterbare. Denne avhandlinga granskar korleis matematiske metodar kan nyttast for å nå begge desse måla. Me har utvikla ei verktøykasse for å forbetra kvaliteten til *saliency*-metodar, med oppgåva å framheva delar til bilete som klassifiserast, nemleg delar som er betydningsfulle for sjølve klassifiseringa. Hovudbidrag er Ablasjons-Pad-formalismen. Den framstiller variasjonar til ei innmating på ein måte som lett kan inspiserast og gje inntrykk av kor trufast forklaringa er. I tillegg føreslår me ein ny måte for å byggja saliency på grunnlag av SIFT-teknikken. Slik slepp ein unna nokre av dei tekniske utfordringane i andre metodar. Den gjev vidare informasjon som er på visse måtar meir nyttig enn dei vanlege fargediagramma.

Til temaet interpreterbare modellar studerer me ein bruk av maskinlæringsbasert støyrydding der symmetriar speler ein avgjerande rolle: kryo-elektronmikroskopi. Symmetriar er eit kjent mønster i nevrale nettverk og anvendingane til desse. I Kryo-EM kan desse nyttast uvanleg godt og målast nøyaktig. Me føreslår ein type av foldings-nevralnettverk skreddarsydd til symmetriane i denne anvendinga, og undersøkjer kva innflytelse dette har på nøyaktigheita til modellen.

Desse forskingsbidraga utvidar det siste skriket av forklarlegheit av bileteklassifisering. Dei gjev også eit utgangspunkt for fleire vidare framdrifter til både forklarlegheit og interpreterbarheit, i bilete-anvendinga og andre.

Contents

Part I

OVERVIEW

CHAPTER α

INTRODUCTION

α.1 The Age of Machine Learning

It is 2023, and AI is everywhere. At the time I am writing this paragraph, it is OpenAI's recently released ChatGPT model [71] which is most talked about – to the point that I hear jokes along the lines that I don't even need to bother writing a thesis anymore, since GPT will be able to do it better already tomorrow. ChatGPT is not the first of its kind, but it has made the term "artificial intelligence" ring more true than anything before it, and made tangible a swath of outcomes that the technology will likely have in the near future.

This includes much excitement about opportunities that are opened up, but also much worry about societally detrimental ramifications.[1] The fact that ChatGPT is literally able to do many homework tasks for students [97] – albeit nothing on the scale of a whole PhD thesis yet – is hardly the most severe of these outcomes.

This thesis is not about ChatGPT or other large language models, or indeed anything else that comes close to general intelligence. In fact, the term "AI" will be largely avoided in the remainder. What the thesis does share with said debate is that it is concerned with machine learning (the heart technological paradigm behind today's AI landscape), and that it attempts to address some of the ethically critical aspects of machine learning and its applications.

To only give a brief overview of the wider AI issues: they can be divided in long-term and short-term ones.

α.1.1 *Short-term*

In the present, machine learning systems are already deployed in numerous applications with various levels of impact upon human lives, and this is expected to increase further. The most obvious are the ones directly interacting with humans and/or taking their jobs, such as autonomously driving cars, but arguably more important are those which invisibly take decisions about things like creditworthyness, criminal risk or simply product recommendations. For them, a major concern is the susceptibility to dataset biases and other undesirable, but hard to detect misalignments [9]. Many of these misalignments are tangible, reflecting biases that humans display too (such as discrimination based on race or gender), but there are likely also related effects for

[1] And thirdly, also some derision claiming it is all simply over-hyped.

which there are no human-known descriptions at all.

In addition, even machine learning systems that are not problematic per se are being used for nefarious purposes such as spreading plausible-looking misinformation. This will not be discussed here, as it is not a problem of machine learning itself but its inappropriate use.

α.1.2 *Long-term*

In the longer term[2], the item of worry are artificial intelligences that truely possess human- or superhuman level intelligence. Such a system could develop its own dynamic, whether in response to human-given tasks or a consequence of its own (possibly misaligned) goals. It could actively trick and instrumentalize humans and make it difficult to put any stop to its actions, if that should for any reason become necessary.

α.1.3 *Courses of action*

In both instances, the problem is not so much AI per se as its intransparency. A system that can be understood by the humans affected by it could be kept in check, at least given sufficient political goodwill. For a system that is not understood it is hard to know even what about it might be necessary to keep in check. This brings with it a dilemma of having to either accept considerable risks of deploying the system with its unknown detrimental potential, or else restricting the use of such systems so much that the benefits are also limited. What course of action to take in this dilemma is a matter of politics and philosophy, and this thesis can not do much to provide an answer.[3] What the thesis does contribute to however (or at least attempts to) is reducing the level to which the dilemma arises in the first place. It does this by investigating how well existing machine learning systems can already be understood, as well as proposing steps that might be taken to improve their understandability.

α.2 Explainable? Interpretable?

In the coarsest terms, learning is an extrapolation problem: one starts with a finite amount of training data, and uses this to infer a function or probability distribution that resides in a much larger space. Specifically, supervised learning (which this thesis deals with) is abstractly of the following form: assume there is a ground-truth function[4] $\mathsf{F}_{\mathrm{GT}}\colon X \to Y$, then given only n data points $(x_i, y_i) \in X \times Y$ such that $\mathsf{F}_{\mathrm{GT}}(x_i) = y_i$, infer a model function $\mathsf{F}\colon X \to Y$ which agrees[5] with F_{GT} as well as possible, in particular also on $x \in X$ that were not in the training data.

[2]The consensus on how long in the future AGI is to be expected is shifting. Until recently, most researchers considered this possibility still distant, unlikely to matter in the 21st century, but the recent progress of large language models has caused many to think it plausible that such systems will emerge within the 2030s.

[3]I, the thesis author, am of the opinion that society should strongly err on the side of caution, and only deploy AI systems where the benefits clearly outweigh the risks and no traditional algorithmic solutions are available.

[4]Unless otherwise specified, the term *function* is always used in the maths sense – i.e. a mapping between two sets, not necessarily continuous, computable or some other sense of benign.

[5]The notion of "agreement" is deliberately left vague at this point.

Clearly, this is in general an ill-conditioned problem: there is only limited information available (though the amount is routinely in the giga- or terabytes nowadays), whereas the space of all candidate functions $\mathbf{F}$ is wont to be infinite-dimensional. This issue will be picked up at multiple points in the thesis.

Only in the most basic special cases, a strategy is available that is both theoretically straightforward and performs well in practice – although the importance of these special cases should not be underestimated. For example, for a signal with low-dimensional domain that has already been sampled at an ample resolution, it is often perfectly appropriate to *inter*polate between the sample points e.g. linearly or cubically; many physical systems have a linear response which can be inferred with a basic least-squares fit; software linters have hard-coded source code patterns that are deemed problematic, etc..

Such models are schoolbook examples of *interpretable models*: we as humans have a good overview of the principles behind them, intuition for when and why they work (and just as importantly, when they fail), and ability to deliberately develop them further to work in settings that require adaptions. They also tend to have good possibilities for *attribution*:[6] in the signal-oversampling example, every sample in the output only depends on its nearest neighbours in the original data; in a linear fit model, perturbations can be tracked by the gradient of the response, which is a global constant; linter antipatterns can have a name and linked documentation with a specification in terms of a formal grammar.

The diametric opposite of interpretable models are opaque *black-box models*. For these, one has initially no information at all available, other than their final output as a function. This is something of a caricature: technically speaking there is always more information available – after all, any model has an implementation. However, even a completely understood implementation will tend to yield an opaque model if its behaviour depends on a sufficiently large number of trained parameters, unless there is a strong mathematical structure behind the way these interact with the input and output.

And even a partially or fully interpretable model may need to be considered as a black box, if the implementation and/or parameters are unavailable to the user. This is particularly relevant for commercial models deployed in a *Software as a Service* manner. Thus, the term *black box* describes more how a model is used, rather than anything intrinsic about the model itself. A model that is not used as a black box ("white box") may or may not be a transparent, interpretable model.

Note that the real, physical world could also be considered as a black box, the only perfectly-opaque one: it is impossible to know what its inner workings are.[7] This illustrates however that it can be possible and useful to obtain information even for a black box: this is precisely the job of *science*. Parts of the world are *explainable* in that sense.[8] This is taken to mean that is is possible to build interpretable auxiliary models which are able to predict at least locally what the outcome of new experiments

[6] See chapter 1 and chapter 3.

[7] Whether the notion of *the world's inner workings* is even in principle meaningful is a philosophical debate of its own right, which will not be carried out here.

[8] All the science examples here are taken from the domain of physics. This does not mean that other scientific disciplines are less relevant (though it might be argued they are indeed less explainable), but is instead mostly an artifact of the author's background.

in the real world would be. By extension, the interpretations and attributions in these auxiliary models also provide to some degree an interpretation of the world. This view on explanations and how it relates to machine learning will be expanded in the next section.

The literature does not always make the distinction between "interpretable" and "explainable" followed here. Othertimes it is vocal about the distinction's importance, like in Rudin [80] who argues that interpretable models should always be preferred whereas explanations about black boxes are generally unreliable.[9] This thesis leans towards that view as well, but also acknowledges that interpretable models are unavailable in many applications. Its contributions work on both interpretability and explainability, with a focus on amending the shortcomings of existing explainability approaches.

α.3 Maths and Physics

α.3.1 *Interpretability*

As said previously, physics is explainable in a peculiar way. Its hallmark is the ability to break phenomena down to the most simple and general principles, from which all more complex behaviour is emergent. These principles are formulated in mathematical language, perhaps best known in equations such as Maxwell's. The model defined by such a set of equations allows deriving predictions about nature from previously measured quantities. It is however worth noting that especially the more fundamental equations, including the (differential) Maxwell equations, do not directly connect prior-measured quantities to predicted experiment-measurable ones, but only provide a description at the microscopic level. The integration procedure from this micro-description to macroscopic predictions is what involves the bulk of mathematical machinery used by the working scientist and engineer. It is also what changes the scope of physical laws from the by-themselves nearly trivial micro-equations to the complex applications frameworks which are based on them.

At first sight it might seem, then, that this complexity also destroys the model's interpretability. The reason this is not the case is that the process is deterministic: to reconstruct a macro-result, it is not necessary to reproduce all the details of the integration procedure, but only the fundamental equations together with the system of mathematical axioms which define their meaning. All of this amounts only to information on the order of a kilobyte (an amount a human can completely and exactly process and memorize)[10], and such information is in principle sufficient to build all the derivations, numerical solvers etc. from scratch and still be confident that they will converge to the same predictions as the previous version – provided all the maths was carried out correctly, which can be objectively checked from the axioms.

Admittedly, the above paints a somewhat over-idealised picture; some caveats need

[9] Rudin furthermore argues that the term "explanation" is also misleading for saliency methods etc., and that these should rather be labelled e.g. "summary statistics".

[10] Information content is a complex topic of its own. The estimate "order of a kilobyte" hardly holds up to much scrutiny, but let's take it to refer to something like the size of a gzipped LaTeX file. Whether this is a good proxy for either Kolmogorov complexity or human-perceived complexity will not be discussed, though Section 3.2 contains related points.

to be discussed. For one thing, the assertion that all the required axioms can be packed in such a small space assumes that a high-level representation is used to formulate them. But that itself requires a foundational framework. Attempts at such frameworks like the Principia Mathematica [108] span hundreds of pages without ever getting close to describing calculus. This could be ignored from the complexity estimation, if one such framework could be accepted as a truly universal convention for all maths and then science to be built on. (After all, in that case there would be no need for additional elaboration when describing a new model.) And although it is in a sense impossible to develop a perfect foundation [33], mathematicians seem at least to have converged on only few systems such as Zermelo-Fraenkel set theory or Homotopy Type Theory [102] as candidate foundations to support all maths used in scientific practice.

Another caveat is that the scientific progress is perhaps less mechanistic than scientists themselves perceive it. Kuhn [51] argues that most scientific work happens within *paradigma*; that is, scientists implicitly work within a certain methodological and mental context. Similar to maths foundations, this shapes what is even possible to express and consider. The paradigm contains information which may be conveyed through education, but is not explicitly associated with a given scientific model. That would mean that for example physical models are indeed interpretable to contemporary physicists, but not to physicists 200 years earlier who were working in a different paradigm.

All of this is, at any rate, in stark contrast to the interpretability level of most deep learning models. For those, data is everything: the best image classifiers are useless before having been trained on many examples. And the resulting parameters are at least megabytes of weights which humans can only interpret to a very limited degree, let alone memorize and reconstruct independently.

α.3.2 *Generalization*

Many physicists would consider the minimalist interpretability – or "elegance", or "beauty" – of physics a central feature of its own right; others would consider this subordinate to more prosaic purposes as a tool for solving concrete problems. The split is perhaps best exemplified by the Bohr-Einstein debates on interpretation of quantum mechanics [64], but it has surfaced in various forms throughout history. The interpretation-focused position goes back at least to Pythagoras and Plato [90].

The practical purposes can largely be summarized to making new predictions. In its strongest form this is the *problem of induction*, which is remarkably similar to the idealised goal of machine learning as per Section α.2: inferring from a limited amount of evidence new truths. The Popperian school rejects the induction problem per se as unsolvable [74], and focuses on falsifiability instead, which is one way of giving preference to simpler, more definitive models.

Historically, both criteria have often gone hand in hand: a simpler, mathematically confined theory (given validity with existing evidence) tended to also generalise better. For example, the geocentric models of the ancients with their complex epicycles were superseded by the Kepler model requiring only ellipses and simple relations, which then turned out to be but a special case of the even simpler (within a suitable mathematical framework!) Newtonian mechanics. The lattermost is ubiquitous even today, whereas highly parameterised models have more typically been – if not outright

disproven (like the geocentric models) – at least unable to produce new predictions that went on the be experimentally confirmed (like string theory, as yet).

That it should be like this is by no means self-evident. The relation is also not quite well-specified, or at least it is not clear that the notion of simplicity therein can be equated with interpretability. For example, general relativity is extremely successful prediction-wise, but while it is simple in the sense of not requiring many parameters, most humans do not find it easy to understand at all. These matters have been discussed at more length by Wigner [111].

There is meanwhile a general and fairly well understood term for the tendency of excessively parameterized models to fail generalising well: overfitting. Crucial for this discussion is that deep learning models often do *not* overfit even with a vast number of parameters. For example, large language models currently improve at almost constant rate as more parameters are added [44]. And although there are other factors involved in the design of models, the sentiment that the sheer size of data, parameters and training (in short also called "the compute") is the predominant cause for good performance is now quite pervasive [98].

CHAPTER β

SUMMARY OF RESEARCH

The contributions in this thesis can be divided in two topics, which roughly parallel the explainability vs interpretability divide discussed in Section α.2. Both lines of research have produced artifacts that stand independent of the other, but there are recurring themes and connecting ideas behind them.

β.1 Saliency

A general approach to explainability of classifiers (particularly, but not necessarily, deep-learning ones) is to disassemble their inputs into a-priori selected *features*, and then attempt to analyse how much each of the features contributes to a given classification result. Such a technique has to contend with two sub-problems: a useful choice of base features, and how to attribute the classification to them.[1]

β.1.1 *Features*

Much of the saliency literature does not venture explicit discussion of what features are or should be. The most common incarnation by far is that of small regions of an image, or indeed single pixels. See Section 3.1 for discussion. In brief, spatially seperated features are both immediately tangible to a human, with their attribution easily visible in form of heat maps (see Figure β .1), and they directly correspond to the default vector-representation of image inputs in deep learning (as well as, indeed, most other digital image processing).

For the most part, this work also complies to the pixel view, however the assumptions

[1]Beware Remark 2.

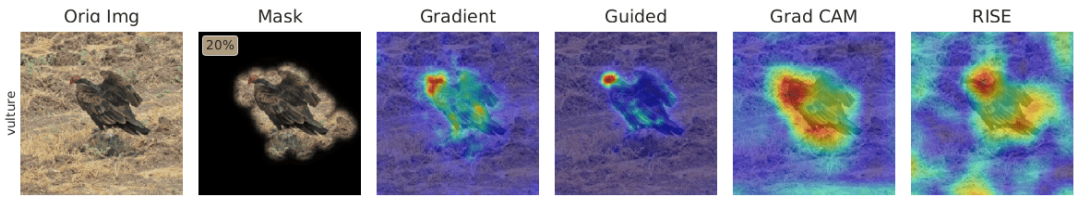


Fig. β.1: Some examples of saliency heatmaps from methods off the literature. Each frame shows a way the cause for the classification of the given image as "vulture" is localised, by the various methods. Taken from [27].

connected with it were regularly questioned during the research and attempts made to keep the developed methods independent of this particular choice. Indeed, it was discovered that some of the encountered difficulties of saliency methods can be seen as resulting from shortcomings of the pixel basis; cf. in particular Section 2.5.1.

Meanwhile, chapter 4 reacts to both these observed shortcomings, as well as insights from older literature, and works to make the features of the *scale-invariant feature transform* [59] viable as an alternative to pixels as saliency features. This decomposition has several properties that are desirable for the purpose, both ones of mathematical and interpretational advantage as well ones that make it easier to obtain attribution results in practice.

It does however also open up new challenges, not all of which could be overcome within the time frame of this work.

β.1.2 *Attribution*

The more technical – and more literature-heavy – concern is how to determine which of the available features are indeed important to the classification process, and which are not. There is a large variety of approaches to this, discussed in Section 1.2; they range from simple and generic like the Integrated Gradient method [96] and from architecture-specific like Grad-CAM [87] to bespoke image decomposition algorithms like Compositional Occlusion Explanation [19].

What these have in common (albeit to different degrees) is that there is no proof that the features they indicate as important do indeed faithfully and universally represent the decision made by the classifier under investigation.[2] *Faithfulness* is perhaps the most important and most challenging concern [80]: when machine learning systems are used for making critical decisions, a wrong explanation could be far more harmful than no explanation at all, because it could lead humans to trust in these decisions when they should not [39].

It is likely that a perfect, universal, proven, faithful saliency method is not even possible in the general black-box case, which after all suffers from a similar problem as the original machine learning conundrum touched upon in Section α.2: having to infer a result of hopefully general validity from only a finite number of evaluations. Nevertheless, it would be desirable to at least have a mathematical notion that can be agreed upon of what importance should entail conceptually, which would have an unambiguous solution at least under simplifying assumptions and/or given unlimited computation resources, and which can be implemented in practice in a way that allows obtaining attribution results in feasible time as well as some estimate of their quality, stability and representativeness.

The present thesis has not reached something the author would consider as *the solution* on this matter. But the Ablation Path method that was developed, as presented in chapter 2, does at least constitute some progress. It provides partial mathematical unification of the existing methods, an implementation that has been demonstrated to work on a typical image classification task with similar ease and stability as the state of the art, whilst providing additional information that is useful in particular for assessing faithfulness.

[2] This is witnessed by the fact that the different method sometimes yield completely different results.

β.2 Symmetry

On the interpretability side, there is an even larger variety of approaches in the literature. Also here, a further split in two sub-fields can be made: interpretation of the inner workings of the model, and interpretation of the training data's influence. An extremely simple example model would be a classifier for data in a metric space, which for a given input looks up its nearest neighbour in the training data set. The result is the class of the neighbour datum, and its interpretation consist of that concrete neighbour and the distance function on the set of images.

This simplistic model is not suitable for e.g. photographic image classification, at least not with distance functions that could be readily computed. In particular, common distance functions like the $\mathcal{L}^2$ metric (cf. Section 3.1) would grade two images as similar if they have a similar distribution of lightness at corresponding locations in the image plane; however, even two images of the exact same scene at slightly different view angle and illumination could be very different in that regard, whereas two images from a stationary camera would rank similar even if they have completely different objects in focus. The model would thus generalise very poorly on a typical image data set.

In that example, there was a *symmetry* present, which the distance function failed to reflect: illumination, translation movement etc. should not have a strong influence on the classification. These will be discussed in chapter 5, as well as parts of chapter 3. Such symmetries are at the core of the rationel for interpretable techniques like SIFT, but also important for elements of architectures that would rather be called black-box, in particular convolutional neural networks (Section 5.3). It is oftentimes not thoroughly investigated how important these built-in symmetry properties are for a system's performance, compared to plain learning from training examples. In chapter 6, *Cryo-EM*, an application where the importance of the rotational symmetry is particularly evident and quantifiable is the topic, both of which makes it a good use case for investigations into increasing intepretability, as well as being an interesting and fruitful research area of its own right.

In chapter 7, a particular variation of the common convolutional neural network architecture is introduced for the task of image denoising, using a novel spiral-based sampling strategy to better exploit rotational symmetries while staying close to the way such networks operate otherwise. It is demonstrated that this slightly improves both denoising performance and equivariance, which indirectly is evidence for the interpretability level.

β.3 tl;dr

This thesis contains:

- A new saliency method (Ablation Path Saliency), with experiments for image classification. See Part II.
- A novel way of using SIFT as the feature palette for saliency methods, also applied to image classification. See Part III.
- Investigations of symmetry in the denoising of cryo-EM images, and a new neural network architecture (Spiral-CNN) to efficiently exploit them. See Part IV.

Each part begins with 1-2 introductory chapters covering fundamentals and relevant literature to the respective topic (chapters 1, 3, 5, and 6), followed by a chapter presenting original work (chapters 2, 4, and 7).

Part II

FROM HEATMAPS TO PATHS

CHAPTER 1

UNDERSTANDING SALIENCY METHODS

1.1 Problem setup

1.1.1 *Concepts of supervised classification*

In the next chapters, a classifier model F is at the center of discourse. In all the practical examples this will be an image classifier, i.e. it maps two-dimensional arrays of colour-valued pixels[1] to (conceptually) discrete labels from a predetermined set. However, most of the methodology does not hinge on that particular setting and can be best approached in a more generic way. The following lays out the notions that are used for both the general problem setup and its concrete instantiations.

Input space

The space $\mathcal{I}$ is the domain of F, i.e. the set of all inputs it could in principle process. In most implementations this is a high-dimensional vector space (particularly the space spanned by individual pixels), but nothing in the present work requires such a strong mathematical structure; for the most part, it suffices to consider $\mathcal{I}$ as some differentiable manifold.

Examples of inputs include:

- Handwritten digits or letters: e.g. MNIST dataset [52], $\mathcal{I} = \mathbb{R}^{64\times64}$
- Low-resolution photos: e.g. CIFAR-10 [50], $\mathcal{I} = \mathbb{R}^{32\times32\times3}$
- Medium-resolution photos, $\mathcal{I} = \mathbb{R}^{h\times w\times3}$: e.g. Pascal VOC [26], ImageNet [82] with $h \approx 500$, $w \approx 300$; Microsoft COCO [56] with $h \approx 600$, $w \approx 400$
- Higher-resolution image, audio, video and even multimodal data are increasingly used as well, but this thesis does not touch on them.

In chapter 3 there is some discussion about the mathematics behind such spaces and why they are used for image-like data.

It is important to note that $\mathcal{I}$ is not the space of all images that F will in fact be capable of classifying, nor the space from which the training inputs have been sampled.

[1] In the typical implementations including PyTorch as used for this work, images are actually stored in a transposed form, i.e. for each colour channel one scalar-valued 2D array. The reason is that this fits

Those are in general much smaller spaces; in the image example, most of the possible constellations of pixels do not correspond to an image that would be physically possible as a photo, and typically $\mathbf{F}$ will have no clue what to do with them. Such inputs, called out-of-distribution or off-manifold, are a case where most current machine learning systems behave in a way that is quite counter-intuitive to humans; see Section 1.4.3.

We denote the space of images that is physically *reasonable* by $\mathcal{I}_{\mathrm{Ph}} \subset \mathcal{I}$; however, this has little merit except symbolically, since it is usually intractable to pin down its exact properties.

Output space

For a pure supervised classification problem, the ground-truth data consists of samples from $\mathcal{I}$ (indeed, from its $\mathcal{I}_{\mathrm{Ph}}$ subspace) together with labels from a discrete set $\mathfrak{L}$.

Another way of looking at this is that each label $\ell \in \mathfrak{L}$ is associated with a set $\mathcal{I}|_{\to \ell}$ of inputs mapping to that class. It is reasonable to assume that $\mathcal{I}|_{\to \ell}$ and $\mathcal{I}|_{\to \tilde{\ell}}$ be disjoint, for $\tilde{\ell} \neq \ell$: this means that a given input is never labelled in two different ways. See Remark 1.4.1 for why this is a useful perspective (and for caveats); for now, suffice it to say that disjoint class-domains allow the ground truth to constitute an at least partial *function*. The idea of supervised learning is then to estimate a completion of this partial function to a larger domain, ideally to all of $\mathcal{I}_{\mathrm{Ph}}$.

In practice, for various reasons another output space is also considered: the free vector space $\mathcal{R} := \mathbb{R}^{|\mathfrak{L}|} \simeq \mathfrak{L}^{\mathbb{R}}$, i.e. the space of tuples of real numbers, one for each possible label. The canonical categorical vector embedding (one-hot encoding)

$$\begin{aligned} \mathbf{1}_{\mathrm{hot}} &: \mathfrak{L} \to \mathcal{R} \\ \mathbf{1}_{\mathrm{hot}}(\ell) &= \Big(\underbrace{0, \ldots, 0}_{\text{Index of } \ell}, 1, 0, \ldots, 0\Big) \end{aligned} \tag{1.1}$$

allows considering any function to $\mathfrak{L}$ instead as a function to $\mathcal{R}$, and is used to bring the data in that format before training.

A more elegant way of expressing the same is to treat $\mathcal{R}$ directly as the space of functions $\mathfrak{L} \to \mathbb{R}$ (this is still a free vector space), in which case the one-hot encoding is simply[2]

$$\ell \mapsto \tilde{\ell} \mapsto \begin{cases} 1 & \text{if } \ell = \tilde{\ell} \\ 0 & \text{else} \end{cases}. \tag{1.2}$$

In many situations, $\mathcal{R}$ is either implicitly or explicitly restricted to the range $[0, 1]$; cf. Remark 1.4.1.

Remark 1. *The one-hot encoding is right inverse to the argmax operation.*

Classifier

The classifier $\mathbf{F}$ itself is first of all a function $\mathcal{I} \to \mathcal{R}$.

together well with the channel architecture of convolutional neural networks; see Section 5.3.

[2] Equation 1.2 uses *curryed* notation for multi-argument functions, retaining the $\mathfrak{L} \to \mathcal{R}$ type which is in this case isomorphic to $\mathfrak{L} \times \mathfrak{L} \to \mathbb{R}$, and $\ell \mapsto \tilde{\ell} \mapsto \square$ corresponds to $(\ell, \tilde{\ell}) \mapsto \square$.

One property of classifiers that is suggested by the discrete nature of the ground-truth labels is that the model output should be approximately discrete too, i.e. for most inputs x there should be exactly one label ℓ such that $(\mathbf{F}(x))_\ell \approx 1$, with $(\mathbf{F}(x))_{\tilde{\ell}} \approx 0$ for all other $\tilde{\ell}$. This property is encouraged by training $\mathbf{F}$ to match exact categorical labels: certainly for x from the training dataset, and at least for the typical architectures (based on ReLU activations and a softmax final layer) this also tends to hold even for many inputs very different from the training data. It is this near-discreteness that allows analysing $\mathbf{F}$ in terms of class domains and their boundaries; see Remark 1.4.1.

For the purposes of saliency, it is most often only the projections $(\mathbf{F}(\cdot))_\ell$ which are pertinent, where ℓ labels the class of the fixed *target input* x_{Tg} whose classification is to be explained. The notation F is employed for this projection:

$$\begin{aligned} \mathsf{F}\colon\ \mathcal{I} &\to [0,1] \\ \mathsf{F}(x) &= (\mathbf{F}(x))_\ell\,. \end{aligned} \tag{1.3}$$

1.1.2 *Attribution*

Remark 2. *The terms "saliency" and "attribution" are used with somewhat inconsistent meanings in the literature. Often they are synonymous, or distinguished in the opposite sense of the one used here (cf. Section β.1). The use of the term "saliency" for explanation of classifiers seems to stem from Simonyan, Vedaldi, and Zisserman [91], who still wrote specifically "class saliency", whereas later works typically write either "attribution" or "saliency" for the same concept.*

Earlier use of similar terms include the Salience Distance Transform *[79], which is only weakly related to the subject discussed here.*

The previous section describes pure black-box classification, i.e. the user of $\mathbf{F}$ never learns more about it and how it applies to an input x_{Tg} than the concrete prediction $\mathbf{F}(x_{\mathrm{Tg}})$. Explainability is the attempt to gain information beyond that. This could mean many different things, even without bringing any implementation details about $\mathbf{F}$ into the picture: it could mean insights about the mathematical properties of $\mathbf{F}$ (beyond being simply a function), it could mean insights about the training data and what about it is relevant for classifying x_{Tg}, or insights about x_{Tg} itself and what about it is relevant for $\mathbf{F}$.

Saliency is particularly concerned with the lattermost. Again, there are sub-distinctions: it is possible to perform pre hoc analysis about x_{Tg}, finding properties that are likely important. This can be augmented by taking the single classification $\mathbf{F}(x_{\mathrm{Tg}})$ into account, which is still quite limited information to work with. More powerful methods generally also evaluate $\mathbf{F}$ with *different* inputs, and the generation of those inputs entails a substantial part of the effort.

What, as widely agreed, saliency should *not* be is a mere reiteration of the input information, an aspect emphasized by Adebayo et al. [2]. Neither should it provide results that have little to do with x_{Tg}. In particular a saliency method should not only procure so-called adversarial inputs of the classifier, i.e. extreme sensitivities of a neural network [99], which – albeit relevant for the study of a classifier – do not actually have much to do with the mechanism of the classification for x_{Tg}.

1.2 Existing work on Saliency

This section introduces a selection of methods from the literature. The mathematical definitions have been translated as far as appropriate to the terminology used in the Ablation Path method presented in chapter 2, and in some cases details simplified.

1.2.1 *Ante-hoc*

Even without knowing anything about $\mathbf{F}$ or $\mathbf{F}(x_{\mathrm{Tg}})$, one can often make some estimates as to what about an input x_{Tg} may be salient for the classification. As an extreme example, for an image with uniform-coloured background and only a confined region with much stronger value fluctuations, it is reasonable to assume that this region corresponds to the foreground object which is classified. This line of thinking is not much discussed in the saliency literature, and perhaps for good reasons: it is simply not true that classifiers behave in general accordingly. For an opposite extreme, consider a picture of a lake where the water itself may appear relatively featureless compared to the strong contrast of trees near the shore. Regardless of that, the water should clearly be more important for a classification of that image as "lake".

Besides this, the idea of classifier-oblivious saliency is also counter to the very concept of attribution, as per [2]. It is for two reasons that this branch of methods nevertheless deserves mention:

1. Most of the saliency methods discussed in the next sections are, to different degrees, *implicitly* sensitive to such image-intrinsic pseudo-saliency. This is most evident in the integrated gradients method [96], which contains a factor $x_{\mathrm{BL}} - x_{\mathrm{Tg}}$ (Equation 1.6) that is proportional to the local signal strength, regardless of what the classifier does.

2. Many classifiers are as a result of their architecture particularly sensitive or insensitive to kinds of features that can be named a priori. This is trivially true for classifiers which operate on features extracted from the images by human-coded preprocessing, but also for ones which directly process pixels.

Both of this is important to keep in mind when discussing saliency methods, independently of whether or not they explicitly take image-intrinsic information into account. Such information can be for example: intensity, edges and similar (compare Section 3.1.3), or multiscale properties (compare Section 3.4).

1.2.2 *In situ*

What we mean here are methods that evaluate one single input x_{Tg}, feeding it to the classifier once. This reveals, at the very least, the class-probabilities assigned to that image, though these come without explanation.

1.2.2A *Tack-on model*

The bare classification result does provide the information *what* should be explained. One may use this as a key to more specialised analysis of x_{Tg}. A simple approach here

would be to evaluate cross-correlations with training images of the top class, based on the assumption that the target image contains a mostly unchanged copy of parts of a training image. Such an assumption is valid in some applications, but these are the ones where cross-correlation could also be used directly for classification[106][3], which is a well-interpretable technique and therefore makes saliency methods quite unnecessary.

The same idea can be extended to the more general case where the assumptions of known interpretable techniques do not hold: as long as *some* classifier architecture has been shown to succeed on the task (i.e., high validation accuracy), it is likely that a similar architecture can also explain the classification. This leads to an approach of training such a model specifically on the explanation task. The training data here could be based on segmentation annotations, saliency from another method, or a combination. The approach is workable [21], and it sidesteps many of the problems of purpose-built saliency methods, but its main drawback is that it explains a black box with another black box. In other words, it provides opaque explanations. Not only does this limit the depths of insights that can be taken, it is also prone to alignment issues [38][39][32]: the explainer will tend to, both through its training and model selection, confirm what humans think the classification *should* be based on, with little guarantee that this corresponds to what the classification actually *is* based on.

An extreme case of opaque explanation would be to literally ask a generic AI system about the reasons for a decision that either the same system or another one made. The latest generation of Large Language Models are able to produce answers to such questions, but in a way that foremostly achieves a plausible appearance.[17] They happily explain even an outright wrong decision in a way that might trick an unprepared human into trusting it. And although it is foreseeable that future iterations of such models will also become better able to handle such cases, perhaps by actively criticising the original decision, this will still remain hard to verify. Indeed, as such models get more powerful, refuting their mistakes will become only harder: both will their capability for forging convincing explanations grow further, as will their decision-making capabilities themselves increasingly exceed those of humans (if not of field experts, then at least of layman users) who could catch out the mistakes.

All this is not to say that auxiliary explainer models are without merit, even black-box ones. Especially when used only as one among a suite of explanation tools – so that they can at least occasionally be double-checked – trusting them most of the time may be acceptable. The advantages include flexibility to learn a variety of explanation formats (e.g. particulary convenient visualisations), and non-necessity of evaluating $\mathbf{F}$ on more inputs and/or with special tooling. The latter can make this approach particularly fast and efficient [21], especially compared with the ones from Section 1.2.3 and the Ablation Path method presented in this thesis. They should then however rather be considered as usability-optimised approximations to a more principled, transparent saliency method, than as methods of their own right. Transparent saliency methods remain the topic deserving the most research attention.

Another use for explainer models could be to feed information back towards the development of better classifiers. Ideally, this could catch the mechanisms behind misaligned decisions and prevent them from happening in the first place. This is somewhat utopic at present, albeit related to some existing approaches for making AI

more well-behaved.

1.2.2B *Augmented evaluation*

Strictly speaking, for a black box, the evaluation $\mathbf{F}(x)$ is only a single value. But in practice it is usually still possible to extract some more information, without needing either detailed assumptions about the internal architecture nor explicitly probing $\mathbf{F}$ with more inputs.

The simplest information of this kind is the gradient[3] $\nabla\mathbf{F}(x)$, which describes the behaviour of $\mathbf{F}$ as an affine function approximating it in a whole neighbourhood (however small) around x. Although differentiability is not a trivial property, a gradient can – for the purposes of machine learning saliency – nevertheless be assumed to be available. Non-differentiable models exist, but they are unlikely candidates for deep learning because the training of those models normally also requires gradients. These are in practice computed by reverse-mode automatic differentiation; see Section 5.3 for details. In brief, it is sufficient for the model to be a composition of differentiable primitives.

Gradients can serve as saliency maps almost by themselves, and were used as such early on [7]. The intuition is that a gradient points towards the direction of strongest change; i.e. $v{:=}\nabla\mathbf{F}(x)$ is a vector such that $\mathbf{F}(x+\varepsilon{\cdot}v)$ differs substantially[4] from $\mathbf{F}(x)$ even for small ε, suggesting that v involves features important for the classification.

The main problem with gradient saliency is that a function's gradients are in general *only* locally representative of its behaviour, possibly confined to very small input regions. For an extreme example, consider a function with a slow- but steadily varying contribution plus a low-amplitude but high-frequency "noise" component on top. The noise would dominate the gradients (making them essentially random), despite being largely irrelevant for differences between real-world inputs. For standard image classifiers, even raw gradients are more useful than random noise [91], but they are hardly stable either. Gradients are also the tool used for obtaining adversarial examples (Section 1.4.3A), and thus at least as prone to their effects as other methods. The situation is exacerbated by the fact that well-trained classifiers are often near-constant on whole regions of the input space (Remark 1.4.1), such that in those regions noise / adversarial fluctuations are the *only* significant contributions to the gradient. As a result of all this, gradient saliency is usually highly grainy / noisy and unreliable. Some of the methods in Section 1.2.3 can be seen as addressing this specific problem while still conceptually following the affine-approximation idea behind the gradient method.

One can also obtain further information from $\mathbf{F}(x)$ beyond single-point gradients, still with only weak and structure-agnostic assumptions about the model. In particular a classifier made up only of piecewise linear layers can be considered as a network of linear inequality constraints. This is useful not just for saliency purposes, but moreso for assessing the classifier's robustness or even proving it to be well-behaved within a certain set of inputs [45][8]. This is a highly desirable research direction, but it does so far seem to be restricted to rather small, low-dimensional datasets. It is unclear whether

[3] Or, arguably, rather the *differential* or weaker notions; see Section 1.4.2 for the mathematical nuances.

[4] The phrasing is vague here for briefness; a proper treatment requires metrics on both input- and output spaces, see Section 1.4.2.

this is only a result of computational expense, or of more fundamental limitations.[5]

1.2.2C *Architecture-specific*

Taking the structure of a classifier network into account opens up far more possibilities for investigation. Such methods will be less generally applicable, but may still cover a large swath of models.

Sequential deep neural networks are built as a composition of simple functions. Even though the whole stack of them exhibits complex nonlinear behaviour, the individual layers can be easier understood. Because composition is associative, one may split up the network and explain only part of it. Most saliency methods can thus be run only only a suffix of final layers, even high-level ones that could also be used on the full network [27]. The saliency method itself will have an easier task then, but the price for it is that the explanation will be in terms of abstract learned features, not obvious pixels or otherwise inherently interpretable features (see chapter 3). There are ways of visualising even such abstract features by attempting to "invert" the preceding layers [62], so as to obtain again an image in $\mathcal{I}$. Such an inversion is in general ill-conditioned; it requires non-unique regularisation (in addition to regularisation that may be needed for the suffix-layer saliency). Even then it is debatable how natural the inputs really are, and whether they faithfully represent anything about the classifier decisions.

In the case of convolutional neural networks (cf. Section 5.3), one can exploit the additional property that at least part of the spatial confinement is carried through the layers (thanks to equivariance). As a result, one can a-priori map features in a layer far to the back of the network to regions. This is the idea behind the Class Activation Mapping [121], which essentially only computes saliency on the final layers of a CNN architecture, specifically on a global average pooling- and fully connected layer, and then uses this as a (potentially lower-resolution) saliency map for the input space. The crucial calculation in this algorithm is to spatially attribute the gradient-saliency of the fully connected layer across the pooling layer weighed according to the distribution of inputs that were actually present during classification of x_{Tg}.

Grad-CAM [87] generalizes this, using a gradient-based technique to propagate the localisation back to $\mathcal{I}$ even when not all layers are purely convolutional.

A main advantage of these methods is their simplicity and efficiency. They require very little extra computation over a mere classification forward pass, and are straightforward to understand operationally. Less clear is what this procedure means from a mathematical / statistical / data perspective. A main point of criticism is that the faithfulness is dubious: CAM-based saliency is not directly connected to any changes happening at the input. It makes the premise that the convolution layers are mere pattern detectors, and displays the locations where the most relevant pattern occurs. This works empirically well for many image classifiers, but it has no provisions for the case that the classifier encodes some nontrivial logic in its layers (e.g. that some pattern means something different depending on another pattern appearing in the neighbourhood). These concerns manifest in problems like the fact (raised in [80]) that

[5] Finite-dimensional linear constraints might not be a suitable framework for capturing the dynamics (diffeomorphisms etc., see Section 5.1) behind the data image classifiers deal with. In that case it could be that any sufficiently powerful classifier would also have adversarial examples that to the verification tool appear as errors.

Grad-CAM may localise two different classes in the same spot in an image, leaving the user unclear as to what is the difference between them.

1.2.3 *Interventional*

The remainder of the methods relevant to this thesis involve various kinds of purpose-generated inputs on which the classifier is probed, in addition to x_{Tg}. There are multiple reasons for doing this:

1. *Stability*: as discussed in Section 1.2.2B, deep classifiers tend to have local fluctuations that badly represent the real-world behaviour. Evaluating with different inputs allows seperating such fluctuations from actual representative behaviour, and ideally averaging them out.

2. *Counterfactual*: it is easier to assess the result for one input when being able to contrast it with results for other inputs, and comparison which changes do and which do not affect the classification.

3. *Validation*: concrete input-output pairs are at least for themselves beyond doubt of faithfulness. If a fully representative set of inputs could be found, it would guarantee that an explanation based on them is faithful.

None of these objectives are automatically reached by just throwing multiple inputs at the model. Only well-selected / -generated inputs achieve that (if at all). Indeed, none of the method discussed here evaluates F with purely synthetic inputs. This would require highly application-specific knowledge, or else the output of a deep-learned classifier would be so erratic as to make it hopeless to extract any attribution information from it. In cases where such knowledge is present, one should consider whether a black-box classifier is appropriate in the first place (cf. [80]). Instead, the inputs are at least partially based on modifications of existing data, mostly x_{Tg}.

1.2.3A *Baseline inputs*

A common notion in the following is that of a single input x_{BL}, which x_{Tg} is contrasted with. The idea is that x_{BL} should be a neutral input on a-priori grounds. Common choices include entirely black or grey images [28][19][95], random noise [28], and blurred versions of x_{Tg} [27][73]. In all cases, a requirement is that $\mathsf{F}(x_{BL})$ differs significantly from $\mathsf{F}(x_{Tg})$ – in other words, that x_{BL} is part of a different class.

Evaluation on x_{BL} by itself is relatively uninteresting, but x_{Tg} and x_{BL} together give rise to a fairly rich family of inputs.

In the following, we will always have x_{Tg} and x_{BL} fixed, as well as F. The saliency methods IntegratedGrads, RISE and MeaningPtrb defined below have these as implicit parameters.

1.2.3B *Interpolation*

The simplest combination-inputs that can be generated with very little additional assumptions are affine interpolations between x_{Tg} and x_{BL}. Specifically, this gives a

family of inputs parameterised by one argument, a concept called in the remainder of the thesis a *path*:

$$\begin{aligned} x_{\text{aff}} &: \ [0,1] \to \mathcal{I} \\ x_{\text{aff}}(t) &:= x_{\text{Tg}} \cdot (1-t) + x_{\text{BL}} \cdot t, \end{aligned} \tag{1.4}$$

or equivalently

$$x_{\text{aff}}(t) := x_{\text{Tg}} + (x_{\text{BL}} - x_{\text{Tg}}) \cdot t.$$

Notice that $x_{\text{aff}}(0) = x_{\text{Tg}}$ and $x_{\text{aff}}(1) = x_{\text{BL}}$.[6]

Equation 1.4 relies on (scalar) multiplication- and addition operations on $\mathcal{I}$. See Section 1.3 for discussion / generalisation. The rest of this section ignores the subtleties, taking the array-vector/tensor view.

The Integrated Gradient method from Sundararajan, Taly, and Yan [96] works by evaluating the gradient of F along x_{aff}, and multiplying it (in the sense of a pointwise scalar product) with the vector between the two end-point images:

$$\text{AverageGrads} := \int_0^1 dt \ \ \nabla_x (F(x))|_{x = x_{\text{aff}}(t)} \tag{1.5}$$

$$\text{IntegratedGrads} := (x_{\text{BL}} - x_{\text{Tg}}) \odot \text{AverageGrads} \tag{1.6}$$

(equation (1) in [96]). They presented this method as a unique choice following from supposedly self-evident axioms, which are not discussed here. Their relevance is put in some doubt by observations like those in Section 1.4.3B. Integrated Gradients do however tend to be less noisy than a single-input gradient. A simplistic reason would be that any integration tends to average out noise contributions, however the research presented in chapter 2 rather refutes this. Instead, a better justification seems to be that the path along which the evaluations happens necessarily crosses at least one decision boundary (cf. Remark 1.4.1), and that it is really the orientation of this boundary that the method evaluates.

The factor $x_{\text{BL}} - x_{\text{Tg}}$ is not motivated very convincingly in [96]. A technical motivation could be that the inner product sums over the colour channels, so one obtains a scalar, purely spatial saliency. That much could also be accomplished with a pointwise magnitude though. Empirically, the pointwise scalar product results in clearer-refined saliency maps, but this is arguably not a feature of the integrated gradient but rather of the fact that regions with high colour-difference between target and baseline are inherently more likely to have a strong influence on classification, as mentioned in Section 1.2.1. It has been argued [2] that this is a misfeature of Integrated Gradients as a saliency method: it makes it prone so showing heatmaps that visually resemble the target image, which makes for a convincing explanation but has little to do with the classification which one is actually interested in. It is not entirely unreasonable

[6] The direction in which a path is traversed a matter of convention. In this thesis the view is always that the path starts from the target image and leads to some other place. One of the reasons for choosing this convention is adaptability to a scenario of multiple baselines (Section 2.8.5B).

In the literature the convention is often the other way around, including [96]. When integrating over a path, the only difference is a flipped sign, which is cancelled by a flip of the difference it is multiplied with.

either, though: one could say Equation 1.6 provides the intersection of the pixels that the classifier is sensitive to, and those that actually have the possibility of affecting it.

In the work behind chapter 2, a third motivation for the $x_{BL} - x_{Tg}$ factor was found: IntegratedGrads corresponds to the gradient of the retaining path score and thus to the first step in an iterative optimisation of this score.

While the simplicity advantage of affine interpolation makes it attractive, it is dubious how much utility it adds over the original images x_{Tg} and x_{BL}. After all, for an essentially featureless baseline, all the $x_{aff}(t)$ contain still the same selection of features as x_{Tg}. In particular if the classifier includes some amplitude-normalisation (whether explicit or learned), it should respond to most of these image in much the same way. On the other hand, if x_{BL} contains features of itself then these could combine with the ones in x_{Tg} to entirely new features,[7] which could be classified in ways completely unrelated to either of them. This would then also show up in some of the gradients along the path, which seems inapproprate for a saliency method intended to explain the classification of x_{Tg}.

The latter concern is to some amount common to all the interventional methods; it is only one of a number of ways probing the classifier can go astray, see Section 1.4.3. Indeed affine interpolation may be among the less artifact-prone interventions. Its inability to probe different varations of weightings of the original input features is however a substantial limitation, as is the (related) fact that it still relies directly on the highly local classifier gradients.

A tool that enables probing such variation that is used by the methods presented in the following, and also in this thesis, are *masks* that select some regions of an image from x_{Tg}, some from x_{BL}. See Section 1.3 for a proper introduction of these. For now, think of a mask $\vartheta \in \mathcal{M}$ simply as a heatmap that can be applied to images with an operation $[\,\cdot\,]_\vartheta$ such that, locally (see Figure 1.1),

$$[{}^{x_R}_{x_L}]_0 = x_L \tag{1.7}$$

$$[{}^{x_R}_{x_L}]_1 = x_R. \tag{1.8}$$

1.2.3C *Exhaustive*

Revisiting the idea that saliency should describe which features are capable of changing the classification, it is sensible to search for actual examples of changes to the input that do or do not affect the classification.Evaluating the model with *all* possible inputs is a naïve ideal for interventions. All the possible $(\tilde{x}, \mathbf{F}(\tilde{x})) : \tilde{x} \in \mathcal{I}$ pairs together characterise exactly the function $\mathbf{F}$ itself. But even if this amount of evaluations was feasible, it would not give much more insight about the reasons behind the different outputs, as long as the $\tilde{x}$ can not be compared structurally.

Reducing it to only inputs of the form $\tilde{x} = [{}^{x_{BL}}_{x_{Tg}}]_\vartheta$ changes this. Thinking of the

[7] No particular notion of "feature" is intended here; a feature could be part of any representation learned by the classifier. For nonlinear classifiers, affine combination need not correspond to combination in its own feature representation. Pre-chosen features in image space, as discussed in chapter 3, are a different matter.

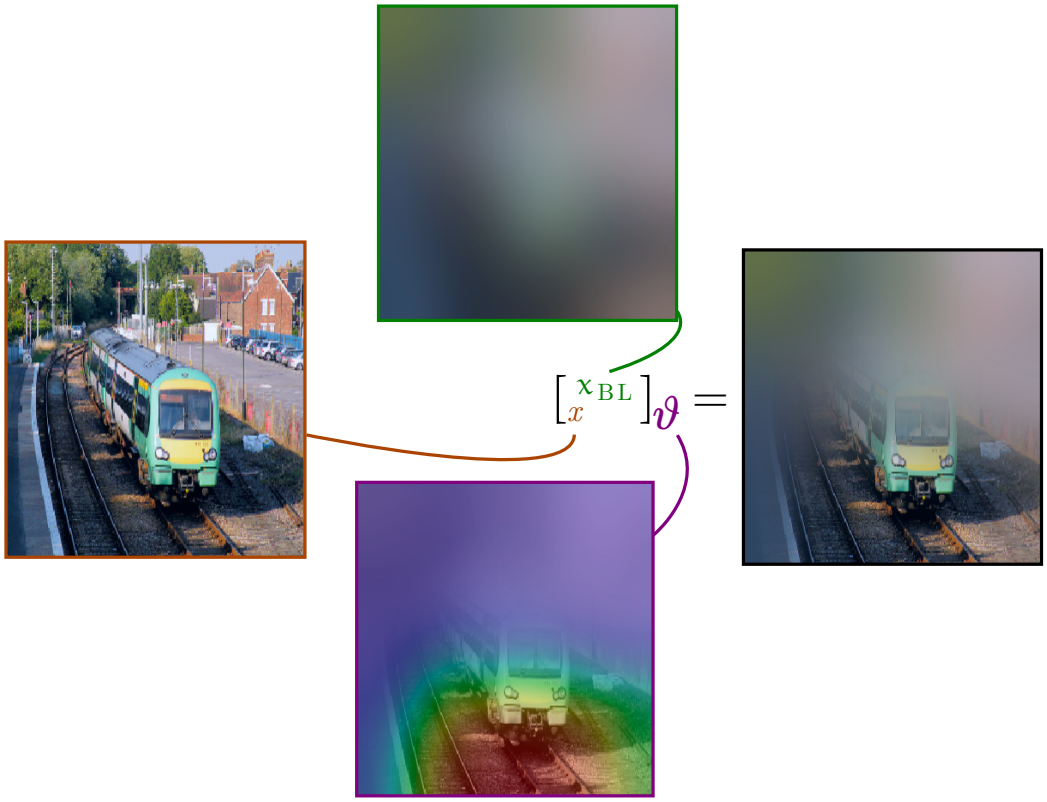


Fig. 1.1: Example of a masking-interpolation (the simple affine variaty, Equation 1.12), as used throughout this part of the thesis. Image: COCO [56]

mask ϑ as a way of selecting for each feature whether or not it is included in the input makes it possible to see this as a *game-theoretical* problem, where each feature represents a player who may or may not contribute to the coalition making up $\tilde{x}$. The problem of assessing how much of the value of the complete coalition should be ascribed to each of the players is a well-studied one in the field of game theory, and it has a solid theoretical solution in the form of Shapley values [89]. Direct computation of these values requires iteration over all possible coalitions, which are in the applications considered here far too many to be feasible. Lundberg and Lee [61] offered ways of approximating them, an approach called *SHAP*. This is an important branch of the explainability field, but its starting point from the view of features as discrete contributors makes it most suitable for applications where the features are already high-level individually interpretable ones. Although it has been demonstrated to work in some cases also with low-level features like pixels, it is not particularly robust in that role, largely falling victim to the same problems as other interventional methods (cf. Section 1.2.3E and following), which are simpler and more easily amenable to addressing these problems. SHAP or other game-theoretical approaches have not been followed in this work, though they could well be useful in combination with the methods developed here, particularly the SIFT technique of chapter 4.

1.2.3D *Random sampling*

The infeasability of evaluating for all possible inputs from a large set is a common problem across many domains, both theoretical and practical. A common way out is the

Monte Carlo approach: instead of probing with a well-controlled but small and likely not representative selection of inputs, or a representative but insurmountable exhaustive one, one probes the classifier with a substantial but still computationally handleable number of *random*-generated inputs. The intention being that such a sample can be representative already at much smaller size than if it were systematically structured, while hopefully avoiding strong biases in the sampling process. Although this may seem haphazard, Monte Carlo techniques have proven successful in many applications, ranging from statistical mechanics [24] to computer graphics [36]. More generally this is also the principle underlying e.g. randomised medical studies.

Petsiuk, Das, and Saenko [73] propose to use inputs arising from random masks to assess how *likely* certain features are to influence the classification. To be precise, they define the saliency of a pixel as the expected value of F over masked inputs, conditioned on this pixel being in the mask:

$$\begin{aligned}\mathsf{RISE} = \mathbf{r} \mapsto & \mathop{\mathbb{E}}_{\vartheta}\left[\mathsf{F}\left(\left[{}^{x_{\mathrm{BL}}}_{x}\right]_{\vartheta}\right) \,\middle|\, \vartheta\left(\mathbf{r}\right) = 0\right] \\ \approx & \frac{1}{\mathbb{E}_{\vartheta}\left[\vartheta\right]} \cdot \left(\sum_{\tilde{\vartheta}} \mathsf{F}\left(\left[{}^{x_{\mathrm{BL}}}_{x}\right]_{\tilde{\vartheta}}\right) \cdot \left(1 - \tilde{\vartheta}\right) \cdot \mathop{\mathbb{P}}_{\vartheta}\left[\vartheta = \tilde{\vartheta}\right] \right)\end{aligned} \tag{1.9}$$

(equations (2) and (5) in [73]).

Here, the expectations / probabilities over masks require pulling ϑ from an a-priori distribution, and the $\tilde{\vartheta}$ are from a fixed, finite sample off that distribution. The requirement to select a distribution is not necessarily a downside of the method, but it does pose some practical difficulties. The authors do it by sampling boolean masks on a lower-resolution grid and upsampling them to image resolution using bilinear interpolation. This way, even a evaluatably-small sample includes independent variations in most image regions, but there is a tradeoff to be made between the achievable fineness of localisation and the sampling efficiency. The bilinear interpolation is argued to avoid hard-edged masks (compare Section 1.4.3A and Section 3.1), but it also causes large parts of the images to be half-masked ($0 < \vartheta\left(\mathbf{r}\right) < 1$), with similar concerns as integrated-gradient. Furthermore, linear interpolation is known to have substantial grid-dependent artifacts, which might easily bias the results. Higher-level interpolation could address this to some extent, but brings its own considerations (such as overshoot phenomena).

Most of these issues are avoided by Chockler, Kroening, and Sun [19], whose method starts out by comparing the classification of very coarse random masked images to first determine only the general location of the salient parts, before then recursively narrowing them down to a more specific location. This combines in a sense the advantages of the purely statistical approach from Petsiuk, Das, and Saenko [73] and of the optimisation-based ones discussed in the next section. This combination approach is highly promising, but in its current mathematical form it is far more involved than the other methods, and has a mathematical structure that is both less easily understood and more tied to the image-classification application. Also, they use fully-boolean pixel masks for evaluation. While this certainly avoids any half-masking concerns, it does on the other hand introduce very strong artificial edges that might

easily have a substantial influence on the classifier. On the plus side, these edges are by construction narrowly confined, and the low amount of choice at each refinement step makes the method inherently less susceptible to adversarial traps than all of the others considered in this thesis.

If [19] had already been published and come to our attention earlier, it would probably have had a stronger influence on the research conducted for this thesis. The methods introduced here have not taken inspiration from the ideas of that line of work, but future research would likely benefit from this.

1.2.3E *Perturbation*

Instead, the general approach that is most similar to the Ablation Path method of chapter 2 is the Meaningful Perturbation method from Fong and Vedaldi [28], and its refinement in Fong, Patrick, and Vedaldi [27].

The idea here is to directly optimise a mask ϑ with the objective of achieving maximal $F([{}^{x_{BL}}_{x}]_{\vartheta})$. This, while intuitively a desirable goal if one wants to obtain saliency, is by itself hardly useful since the optimal mask will tend to be trivially $\vartheta \equiv 0$ or similar, i.e. the mask that more or less selects the full target image (which does, after all, by premise contain the object of interest). To make the task meaningful, one needs to impose at least an area- or mass constraint upon the mask, or alternatively an according penalty term to the optimised cost function. This essentially limits the number of features that can be selected from x_{Tg}, thus forcing the optimisation to prioritize the ones that are actually most salient.

Unfortunately, this is in general not sufficient, because optimisation-based saliency is extremely sensitive to adversarial effects. In fact, most methods of purpose-generating adversarial examples use very similar optimisation techniques [99][13]. What this means is that an optimised mask will typically highlight some individual pixels, such that $[{}^{x_{BL}}_{x}]_{\vartheta}$ is perhaps only subtly different from x_{BL} but classified completely different. In these cases, the mask is useless as a saliency map.

Fong and Vedaldi [28] address this by optimising with regularisation. Specifically, they impose a total-variation penalty on the masks, which is one way of punishing masks that are highly localised but have strong small-scale fluctuations. Additionally they introduce a "random jitter" of image-translations τ and take the expectation over these, the idea being to avoid depending on mask features that must be placed in a very specific way relative to the target image.

$$\mathsf{MeaningPtrb} = \arg\min_{\vartheta} \left(\lambda_1 \cdot \mathrm{mass}(\vartheta) + \lambda_2 \cdot \|\vartheta\|_{TV} + \mathop{\mathbb{E}}_{\tau} \left[F\left(\left[{}^{x_{BL}}_{\tau(x_{Tg})} \right]_{\vartheta} \right) \right] \right). \tag{1.10}$$

This regularisation approach, while being technically rather different, has an analogous effect to the upsampling from low-resolution masks[8] used by Petsiuk, Das, and Saenko [73]: it restricts the masks to actually work as selectors of already-present features, rather than new structures. The details, commonalities and differences are discussed in Section 1.4.2, Section 1.4.3A and Section 2.5.1.

[8] Fong and Vedaldi [28] do in fact also store the masks internally at moderately lower resolution, but more this seems to have less relevance compared to the TV regularisation.

Striking about particularly the Meaningful Perturbation method is that there is no obvious and universal recipe for interventions that lead to good saliency results. The mask-regularisation, respectively λ_2, is only one of several parameters that need to be somehow chosen. Although total variation by itself is well understood and parameter choices for a TV regularisation may in some applications be possible to make based on physical principles, this is even in confined scientific fields often done in a fairly ad-hoc manner [116]. For general-purpose image classification, or even more broadly data classification, there is little hope of deriving a one-size-fits-all strategy for optimisation and regularization from first principles. What is somewhat promising is to couple in domain-specific but still inherently understandable and classifier-independent data analysis to inform this strategy. This idea is explored in chapter 4. Methods like [19] could also be seen as implicitly generating an optimal mask with very specific properties, which is conceptually very similar but more difficult. There is much ongoing research in this direction; particularly related to ours is the work by [46], which is discussed in Section 3.4.

Neither [73], [28] nor [27] make such considerations, at least not explicitly. In fact these papers do not elaborate at all how they arrived at their hyperparameters. They merely give results for one parameter choice, and argue it to be good based on both direct visual examples, and the score in a benchmark, success in which is thought to be a way of validating that a saliency method behaves reasonably.

1.2.4 *Saliency validation*

Similarly to how a saliency method tests in a sense the classifier F, as a function of x, the saliency method itself may be considered as a (higher-order) function

$$\mathsf{S}\colon\ (\mathcal{I} \to \mathcal{R}) \times \mathcal{I} \to \mathcal{M}$$

which can in turn be tested. There are essentially two levels on which this can be done:

- Evaluating S with a given classifier input images, and comparing the results with those from other methods and/or a-priori expected characteristics.
- Checking properties of S as a function.

The latter is the more complete approach, but more involved too.

1.2.4A *Sanity properties*

Adebayo et al. [2] work in that property-check direction; their main concern is investigating whether a saliency method really depends on what classifier does. These are relevant checks, however they should rather be understood in terms of properties that a salience method should *not* have, rather than properties that it should have (thus the title "sanity checks".

1.2.4B *Expected results*

A more simplistic benchmark is to evaluate a classifier/saliency combination on a large number of inputs from a suitable dataset, and cross-checking the resulting saliency

maps against annotations about the data. For image classification, such annotations are often available in form of bounding boxes of the physical objects appearing in each image. More typically these would be used for training localising object detectors such as YOLO [76]. The premise behind the *pointing game*[118] is that a saliency method for image classification should at least typically highlight roughly the annotated objects.

It is human-intuitive that a good classifier should base its decision primarily on features spatially confined at the object associated with the class, and in that case a good saliency method would indeed also point in those regions. The premise is nevertheless problematic, since there is no inherent reason why a classifier trained only on labels would do this. As far as the cost function by which such a classifier is trained is concerned, it could just as well use any bias found in the data set. And indeed this has been shown to occur in real-world applications, the perhaps most infamous example being the tendency of skin cancer classifiers to classify surgical skin markings / rulers in the image, rather than the lesions in which the cancer might actually be developing. [112]

In such a case, a saliency method that correctly and importantly uncovers this misaligned behaviour would score badly in the pointing game. Meanwhile, a method that for whatever reason highlights the lesions instead would score high in the pointing game, but would have utterly failed to give a faithful explanation of the classifier.

This concern does not completely disqualify the pointing game as an assessment. For general-purpose image classification with many classes, it is reasonable to assume that such a extreme biases are not present, at least not consistently across most of the classes. In that case, a better saliency method should indeed perform at least as good in the pointing game as a worse method. Even when severe biases are present, a bad saliency method would not automatically have an advantage in the pointing game – after all, it would not only have to disregard the actual classifier behaviour, but also independently manage to match the annotations. Then again, it is not too far-fetched that it would accomplish this: the annotations themselves could be biased in a sense, for example it is typical for datasets to have the object of (human-) interest near the center. Simply preferring an explanation near the center would then a saliency method an edge that does not correspond to an improved accuracy in the classifier explanation. In other cases, it might be inherent image features that pull the saliency method's attention. Skin lesions for example have a particularly simple contrast and shape.

What the pointing game does quite unequivocally offer is a straightforward way of checking *stability* of a saliency method: a method that is disproportionally susceptible to near-random purely-local fluctuations and/or adversarial behaviour will have a correspondingly high fluctuation in where the saliency points. As a result, it will score worse than a more stable method. Therefore, using the pointing game to inform choice of hyperparameters of a given method that are specifically concerned with instability is reasonable.

Fong, Patrick, and Vedaldi [27] have run the pointing game, comparing their method with many other ones, on multiple commonly-used models and data sets. They apparently also based the tuning of their hyperparameters on this. See Section 2.7.3 for comparison of our method with this benchmark.

1.2.4C *Faithfulness post-check*

One thing that particularly the Meaningful Perturbation method [28] offers is to associate with the heatmap one concrete input, the classification result of which can be inspected by a human together with the image. This assures[9] to some extend the faithfulness of this explanation, in that an explanation which highlights only irrelevant features would correspond to an image that does *not* have the desired classification. This single input suggestion is not much data to go by, but at least the classification can be compared with that of the target and baseline. Most other methods do not have such a simple way of assessing relevance at all.

Petsiuk, Das, and Saenko [73] suggested a simple way to estimate faithfulness for *any* heatmap-yielding saliency method. This method, which we shall call *(ranked) Pixel Ablation*, has since become a standard test in the literature [27][95][19]. The following is a description of it in plain language. See Section 2.1 for the mathematical realisation.

The idea is to interpret the heatmap as a ranking of pixels from least important to most imporant. This can then be used for generating masks that contain a given number m of either only the most important pixels, or only the least important ones. By "contain" it is meant here that for the pixel in question the value that it has in x_{Tg} is chosen, whereas non-included pixels will take the value from the baseline x_{BL}. Doing this for all possible m, i.e. between 0 and $h \times w$ gives a fine-grained (if not quite continuous) sequence of mask-image-classification tuples. The F classifications can be plotted as a curve, called *deletion curve* when only the allegedly least important pixels are included (in other words, the m most important ones deleted) and *insertion curve* for the opposite case (the m most important pixels inserted).

The premise, then, is that a deletion curve for a successful saliency heatmap should drop to a low value already for small m, since the important parts of the image would be removed early on which should presumably result in images that are not classified like x_{Tg} anymore. Vice versa, an insertion curve should rise early, or equivalently not drop quickly traversing it right-to-left, since that corresponds to removing only unimportant pixels so that F should continue treating the images largely like the original. How good each of this holds up can be summarised as a single number, called variously *ablation score* or simply *area-under-curve*: it is simply the average of the classifier outputs across all the pixel-ablated images. Low deletion-AUC and high insertion-AUC are desirable.

These scores are a compelling check for a saliency method, and have been a major inspiration behind the Ablation Path method.

1.3 Features, sets, modules, rings

All the above, in line with most of the literature, has not delved into the mathematical structure behind the notion of features and masks. The simplest perspective on this is that the input space I is a finite-dimensional free Euclidean space (in other words, that x_{Tg} is an array of real numbers), that each dimension in this space corresponds to one feature, and applying a mask (also an array) means multiplying each entry by an individual gain factor:

$$(\vartheta \odot x)_i = \vartheta_i \cdot x_i \tag{1.11}$$

[9] Caveats to this in Section 1.4.3A and Section 1.4.3B.

where $\mathfrak{i}$ is a suitable multi-index. In the case where the array stores pixel-brightness values, this corresponds to the intuitive operation of shading some regions in the input into darkness.

The fixed choice of black (or other arbitrary reference point, can also be half-grey) as "featureless" is problematic: after all in many images a black region would be quite a strong artificial contrast. This is one of the reasons to employ a selectable baseline input, and use the masked interpolation operation, which can be based on the pointwise product to obtain a pointwise affine interpolation:

$$[\,{}_{x_\mathrm{L}}{}^{x_\mathrm{R}}\,]^{\mathrm{aff}}_{\vartheta} = (1-\vartheta)\odot x_\mathrm{L} + \vartheta\odot x_\mathrm{R}. \tag{1.12}$$

The space of masks $\vartheta \in \mathcal{M}$ does not need to be the same as that of images $x \in \mathcal{I}$: for colour images the latter will generally be of the form $\mathbb{R}^{h\times w\times 3}$, but the three colour dimensions would be hard to interpret as a saliency mask, so it is standard to adapt only the spatial dimensions to get $\mathcal{M} = \mathbb{R}^{h\times w}$. The pointwise multiplications are then pointwise scalar-by-vector multiplications.

All of this can be directly generalised by only requiring $\mathcal{M}$ to be a ring and $\mathcal{I}$ a (left) $\mathcal{M}$-module. In chapter 3 it is detailed why such generalisation is useful, but to mention just one aspect here: the need for regularization / upsampling in the literature methods demonstrate that it is not really appropriate to consider arbitrary multiplication of pixels, and that $\mathcal{M}$ should actually be a considerably smaller / more regular space than $\mathcal{I}$ (or a greyscale version of it). This is awkward to express with array-vectors, but readily allowed for by the module formulation.

The generalised form can be used not only for the explicitly mask-based methods [73][28][27], but also for Integrated Gradients [96]: with rings that have an embedding $\vartheta_\mathrm{hom} : \mathbb{R} \to \mathcal{M}$ (which would in the pixel case correspond to spatially constant / homogeneous masks) Equation 1.4 can be rewritten as

$$x_\mathrm{aff}(t) = [\,{}_{x_\mathrm{Tg}}{}^{x_\mathrm{BL}}\,]^{\mathrm{aff}}(\vartheta_\mathrm{hom}(t)) \tag{1.13}$$

– or simply $[\,{}_{x}{}^{x_\mathrm{BL}}\,]_t$, with implicit "broadcasting".

1.4 Geometric observations

1.4.1 *The class-domain picture*

As mentioned before, the classifier $\mathbf{F}$ may have a continuous space $\mathcal{R}$ as its codomain, but at least conceptually it approximates a discrete-valued function assigning each input x (at least each in-distribution one, cf. Section 1.4.3) one and only one label from $\mathfrak{L}$. In reality, this is not exactly true, but the output of a typical image classifier on validation-set inputs does indeed tend to be approximately one-hot, i.e. one class is assigned a softmax probability of 90-100% and all others negligibly low.[10]

[10] This observation is to some extent vacuous, because the softmax function maps *any* sufficiently large-magnitude vector (which not happens to have multiple equal maximal entries) to an approximately 1-hot output. What is not trivial is that this also happens with the particular amplitudes coming from the trained pænultimate layer of a deep-NN classifier. See Section 5.3 for more on this topic.

An explicit way of obtaining such a discrete function which $\mathbf{F}$ approximates[11] is to only consider the single highest-scoring class, i.e.

$$\begin{aligned} \hat{\mathbf{F}} &\colon \mathcal{I} \to \mathfrak{L} \\ \hat{\mathbf{F}}(x) &= \arg\max\left(\mathbf{F}(x)\right). \end{aligned} \tag{1.14}$$

Here, $\mathcal{R}$ was treated as the free space $\mathfrak{L} \to \mathbb{R}$ like in Equation 1.2.

Specifically for saliency-understanding purposes one can then consider the preimages of individual labels under $\hat{\mathbf{F}}$ as a proxy for $\mathbf{F}$. We call these preimage sets $\hat{\mathbf{F}}^{-}(\ell)$ *class domains.*

Consider for the sake of argument the oversimplified case that $\hat{\mathbf{F}}^{-}(\ell)$ is a convex set[12]. This leads to a simple result[13] about the Integrated Gradient method. It is somewhat of a "spherical cow" model not necessarily very representative for real-world data, but is still empirically close to true in many situations and has been another major inspiration for the devolopment of the path method of chapter 2.

Lemma 1. *For a saturated classifier* $\mathbf{F}$ *with smooth level sets, the average-gradient saliency (Equation 1.5) corresponds to the orientation of the normal on the decision boundary where it is crossed by the path of evaluations:*

$$\mathsf{AverageGrads} \propto \hat{\mathbf{n}}_{\partial\left(\hat{\mathbf{F}}^{-}(\ell)\right)}\left(x_{\mathrm{aff}}\left(\mathrm{t}_{\mathrm{trans}}\right)\right), \tag{1.15}$$

where $\ell = \hat{\mathbf{F}}(\mathrm{x}_{\mathrm{Tg}})$*, the crossing point is located by* $\mathrm{t}_{\mathrm{trans}}$ *that fulfills*

$$\hat{\mathbf{F}}\left(x_{\mathrm{aff}}\left(\mathrm{t}_{\mathrm{trans}}+\delta\right)\right) = \begin{cases} \ell & \text{for } \delta < 0, \\ \ell_{\mathrm{BL}} & \text{for } \delta > 0, \end{cases} \tag{1.16}$$

with $\ell_{\mathrm{BL}} = \hat{\mathbf{F}}(\mathrm{x}_{\mathrm{BL}})$*, and* $\hat{\mathbf{n}}_{\partial\left(\hat{\mathbf{F}}^{-}(\ell)\right)}\left(x_{\mathrm{aff}}\left(\mathrm{t}_{\mathrm{trans}}\right)\right)$ *denotes the unique vector that is orthogonal to all tangent vectors in the domain boundary* $\partial\left(\hat{\mathbf{F}}^{-}(\ell)\right)$*, cf. Figure 1.2.*

Proof. The saturated classifier can be modelled[14] as the limit of a family of smooth functions $\mathbf{F}_{\beta}$ converging to a piecewise const one. For this, Lemma 11 guarantees that any vector tangent to the decision boundary at the crossing point $x_{\mathrm{aff}}\left(\mathrm{t}_{\mathrm{trans}}\right)$ has a vanishing scalar product with AverageGrads. This is precisely the defining condition of a normal vector to that boundary. □

Remark 3. *The conditions for Lemma 1 are in practice (deep learning image classifiers) not fulfilled in the sense that the gradient is completely vanishing in the decision regions far from a boundary, as evidenced by the fact that the gradient at* x_{Tg} *itself has been used for saliency purposes [91]. The sense in which it is true is that the gradients near a decision boundary tend*

[11] The approximation is in the sense of following up $\hat{\mathbf{F}}$ with the explicit one-hot encoding again, so both functions have codomain $\mathcal{R}$.

[12] We do not really require convexity, but it is a simple way to ensure that paths cross a boundary only once.

[13] Due to Olivier Verdier, see appendix.

[14] This is what requires smooth boundary. We do not go into the details.

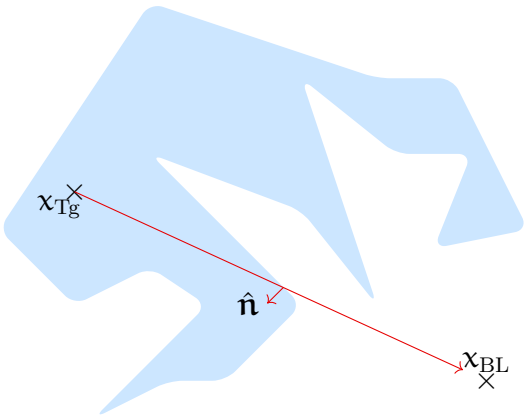


Fig. 1.2: Low-dimensional sketch of decision regions and -boundaries. The blue region represents the class-domain of the input x_{Tg}. A linear-interpolation path is shown, and the point where it crosses the decision boundary with a normal on that boundary. N.B.: this represents only very crudely the behaviour in real image classification applications, as inevitable with low-dimensional visualisations.

to be much stronger than far away, so that they dominate the integral. But the very concept of distance from a boundary is not without its challenges, since the domain is high-dimensional. It often seems that a fractal in which most points are very close to a boundary is a better model for the geometry of the class domains, rather than the easily visualisable sets with manifolds as their boundary as which they are often presented.

1.4.2 *Differentials and gradients*

Differentiation is employed for saliency in two senses: as a direct result contribution or -propagation, for example in Integrated Gradients [96] and Grad-CAM [87], or as a tool for optimisation purposes. The machine learning literature does usually not make a fundamental distinction between these, but mathematically there are some subtleties that merit discussion.

Conceptually, (strong[15]) differentiation is concerned with finding a local linearization to a function, i.e. given $F\colon \mathcal{I} \to \mathcal{R}$ and $x \in \mathcal{I}$ to find a linear map $F'_x\colon T_x(\mathcal{I}) \to T_{F(x)}(\mathcal{R})$ fulfilling

$$F(x + \Delta x) \approx F(x) + F'_x(\Delta x) \tag{1.17}$$

for sufficiently small Δx; more precisely it should converge quadratically:

$$F(x + \varepsilon\cdot\Delta x) = F(x) + \varepsilon\cdot F'_x(\Delta x) + \mathcal{O}\left(\varepsilon^2\right). \tag{1.18}$$

In case of scalar-valued functions, the space of linear maps reduces to the dual space of the tangent space, i.e. $F'_x \in T^*_x(\mathcal{I})$, which in case of a vector space domain is the same as the dual of the entire space. In case of a Hilbert space (including Euclidean spaces), it is furthermore isometrically isomorphic to $\mathcal{I}$ itself, giving rise to the common view of differentials as gradients, which are elements of the domain space fulfilling in place of

[15]The issue discussed here also carries over to weak differentiation.

Equation 1.17

$$\mathbf{F}(x + \Delta x) \approx \mathbf{F}(x) + \langle \nabla_x (\mathbf{F}(x)), \Delta x \rangle \tag{1.19}$$

with the Hilbert space's scalar product $\langle , \rangle$. Since $\nabla_x (\mathbf{F}(x)) \in \mathcal{I}$, one can then carry out operations of the form $x + h \cdot \nabla_x (\mathbf{F}(x))$ which are the basis of gradient descent algorithms. This is not possible with $\mathbf{F}'_x \in \mathcal{I}^*$. The flip side is that $\mathbf{F}'_x$ is uniquely determined from only $\mathbf{F}$, also when a scalar product is not available. Although it is usually possible to define a scalar product on any space that can be used for computations at all, there is not always one obvious choice. Using the Euclidean scalar product on the pixel representation of images corresponds to working in a discretized form of the $\mathcal{L}^2(\Omega)$ Hilbert space; cf. chapter 3. Empirically, this is often a good starting point, but $\mathcal{L}^2(\Omega)$ has some properties that encourage problematic behaviour like the adversarial effects discussed in Section 1.4.3A. In particular, narrowly localised features have a small $\mathcal{L}^2$ norm even if they contain e.g. visually striking edges.

A practical consequence of all this is that gradient-descent depends on the choice of metric. The reason this can in some cases be disregarded is that a proper minimum is still unambiguous. In particular, when optimising a strictly convex function on a compact domain, gradient descent with sufficiently small step size always converges on that minimum, regardless of the choice of metric.

This is however not very representative of the kind of optimisation problems considered here: $\mathbf{F}$ is highly nonlinear, and GD-optimisation (whether during training of a classifier, or for saliency purposes) is understood to not yield an exact global minimum, but rather approximates some local one. Precisely finding a narrow minimum may not even be desirable, as it tends to yield solutions that generalise poorly. Arguably this is a symptom that not the right optimisation problem was solved, but short of knowing a better one the standard approach is to use stochasticity, momentum and/or large step sizes, all of which encourage finding instead shallow, stable minima. This also voids the argument of minima being metric-agnostic, though.

Specifically an $\mathcal{L}^2$ metric will tend to encourage steps towards high-frequency features, as these often have a strong influence on the classifier but small norm. This can be appropriate for training internal representations, but specifically for saliency purposes it is rather undesirable (see next section).

A way to circumvent these difficulties is to carry out the optimisation in a space that can legitimately considered to be finite-dimensional. This is at least part of the idea behind the SIFT decomposition presented in chapters 3 and 4.

1.4.3 *On- and off manifold*

As mentioned above, the space of technically realisable inputs $\mathcal{I}$ is not the same as the space $\mathcal{I}_{\text{Ph}}$ of inputs that could actually arise in intended use. The remainder $\mathcal{I}_{\text{Ph}} \setminus \mathcal{I}$ constitutes what are called *off-manifold inputs*. An alternative view is that such inputs are still possible but extremely unlikely. This arises from the stance that every input encountered in practice is sampled from some distribution, which features off-manifold values with, if not zero, at least such low probability that they can be considered out-of-distribution.

As concrete examples one could name things such as submarine images used as input for a classifier trained on house cat images, but of more interest here are images

generated from digital manipulation that is in some sense unnatural. This includes particularly pixel-wise modifications such as addition of excessive noise, introduction of artificial colours, but also interpolation between different images or complete blurring.

There is no universal way to characterise what inputs are on-manifold; it depends on the application. In some cases there is a fairly good theoretical understanding of the physical process behind the input sampling (for example Cryo-EM images, chapter 6, are generated by sending an electron beam through a particular kind of ice sample) and then it can be possible to formulate rigorous criteria that would detect at least some kinds of off-manifold inputs (in Cryo-EM e.g. large uniform-intensity regions are exponentially unlikely due to the noise present). For the generic image datasets like COCO [56], heuristics are harder to come by (see Section 3.1 for some attempts), but a human would still discard many possible images immediately as nonsensical. This is where there is an important difference between human perception and that of computer vision systems: the standard deep-CNN based (Section 5.3) classifiers do not pause even when given completely mutilated images as input, but associate them to labels with similar confidence as images from the original training dataset. It is debatable whether this difference has anything to do with a fundamentally different way humans process images, or just with the fact that we humans have been trained with more varied inputs and have additional meta-classifiers that distinguish e.g. photos from pixel collages.

The following sections discuss some concrete challenges that various interventional saliency methods have in keeping the generated test images on-manifold.

1.4.3A *Adversarial phenomena*

A particularly relevant kind of off-manifold input are adversarial ones, first reported by Szegedy et al. [99]. These are inputs that differ from a realistic one in only a very small way, often indeed so small that a human cannot tell them apart, yet are classified in a completely different way. The perturbation itself (difference between adversarial image and original) tends to be not human-plausible, appearing like random noise or unrelated pixel defects. Adversarial attacks are most often discussed for images, but exist also for other applications such as text [30].

Though there are numerous attempts to make deep learning systems robust against adversarial attacks, these do generally not prove absence of adversarial examples and typically attacks are found little later [101] with different methods.Theoretically it might be possible to build a classifier that is provably robust to small adversarial perturbations [45], but such attempts are so far restricted to applications much simpler than those addressed by state-of-the art machine learning models. Built-in adversarial defense also often comes at the price of reduced performance on real test data [4]. It is thus currently necessary to admit that adversarial attacks are a possibility for all machine learning systems that one might want to explain with saliency methods.

Remark 4. *An interesting questing is whether better machine learning / AI systems will inherently tend to become more- or less susceptible to adversarial phenomena, compared to worse ones. Even before these effects were disseminated for neural networks [99], effects that might be called adversarial were discussed by analogy with human "counterfeit utility". Omohundro [70] gave examples like drug addiction, but argued that AI would develop "protective mechanisms"*

against this.

Adversarial inputs are generated by solving an optimisation problem, namely finding a small perturbation to the input that nevertheless effects a change to a different class. Notice that this is almost the same optimisation problem as the Meaningful Perturbation saliency method (Section 1.2.3E), except it skips the use of a mask for the generation of a perturbed input. Although mask-interpolating between only two images provides intuitively a much more restricted set of possible perturbed images, the achievable small-scale changes are actually almost equally powerful if using full-resolution masks, since a small contrast between target and baseline is enough to introduce small-scale but arbitrarily high-frequency information.[16] As such it is not surprising that the rawest form of perturbation saliency gives not explainable heatmaps, but rather produces masks that are just as inscrutable as the classical adversarial attacks.

The established way of avoiding this is regularization, as with the total-variation penalty in Equation 1.10. This works, but it is a fairly blunt tool in the sense that it restricts simultaneously not only the capability of masks introducing adversarial information, but also to mask out finely separated features. The latter can to some degree be amended with the techniques like multilevel area constraints [27], which are however quite ad-hoc and parameter-reliant. An alternative is a recursive strategy as used by Chockler, Kroening, and Sun [19], who use very restricted optimisation at each step of the algorithm and avoid finding adversarial examples this way, but then refine the search to nevertheless get good fits to narrowly confined features.

Both strategies, whilst empirically successful at avoiding adversarial masks, do not solve the problem of keeping perturbations on-manifold; indeed they rather worsen the distribution faithfulness, considering the effects discussed in the following.

1.4.3B *Blending; (non-)convexity of domain*

The premise behind all the methods involving paths is that it is possible to connect the target image to the baseline in a continuous way, and evaluate the classifier along the whole path. Specifically the affine path used in the integrated gradient methods enforces this by using linear interpolation. The essential property required for this is a *convex* domain. But while a vector space $\mathcal{I}$ is convex by construction, there is in general no reason to assume that the subset $\mathcal{I}_{\mathrm{Ph}}$ is convex as well. For some specific applications this may be warranted for physical reasons; for example, sound has a natural amplitude-scaling operation associated with distance between the source and microphone, as well as a natural addition operation associated with simultaneous play of multiple sources.

But specifically images are *not* of this character: whilst value-scaling is reasonable for them due to the possibility of darker lighting conditions in a scene, one would need to come up with quite contrived setups to implement addition in a physical way (e.g. a half-reflecting glass pane). See Section 3.1 for more discussion.

This does not necessarily mean that the domain is disconnected (though it could be), just that interpolation in the vector-space representation is not a very appropriate

[16] There is a restriction in that no artificial *colours* can be introduced, but colour tends to play a rather lesser role compared to luma-texture, a matter discussed in Section 4.1.2.

strategy for constructing paths. This casts doubt particularly on the integrated gradient method [96], but any method using $[\,\cdot\,]^{\mathrm{aff}}_{\vartheta}$ with masks ϑ that are not perfectly boolean is subject to the same criticism. Indeed, most of them favour – either explicitly or implicitly – the use masks that are mostly and/or approximately boolean:

- The Extremal Perturbation method [27] (unlike these authors' earlier work [28]) uses optimisation with limits on the range of $\vartheta \in [0, 1]$, the boundaries of which are booleans, and dedicated smoothing operators designed to preserve these values.

- RISE [73] starts by randomly sampling purely-binary (but low-resolution) masks, and only smoothens these afterwards to attain continuity.

- Occluded Explanations [19] use true binary masks with only few hard edges.

In all of these cases, the restriction to binary applies only to the generation of the images used for classifier-evaluation. It does not mean the result of the saliency methods is binary – which would only be little information, providing no relative importance within the "in" and "out" parts. Rather, evaluation happens for multiple such masks and a non-binary heatmap is generated based on the results (in each case this involves averaging of some appropriate kind, weighted by the classification), which provides more graduated information.

The downside of this is that the assessment capabilities are extenuated: while each of the input-output pairs is verifiable and human-interpretable, their statistical analysis is not. Avoiding this dilemma is perhaps the main advantage of the Ablation Path method of chapter 2.

1.4.3C *Cuts and blobs*

Well-regularised[17] binary masks are in principle reasonable for photos, since analogous masking occurs in reality whenever one object is partially obscured by another one in the scene. Unfortunately, even such masks can nevertheless lead to quite egregious off-manifold inputs, because the transition edges between the regions belonging to x_{Tg} and those belonging to x_{BL} can act as distinctive features all by themselves. Such features are however completely artificial, and if the classifier responds to them it is just as detrimental to the purpose of the saliency method as classical adversarial fluctuations are. In both cases, this is actively encouraged by an optimisation strategy that includes the classification in its objective function.

Some methods introduce cutting edges only in very specific ways, for example through using rectangular masks at the outset [19]. This restricts the potential for an optimiser to invent new features to achieve high score, but is not beyond doubt either because the rectangles introduce very hard corners which classifiers could plausibly respond to strongly.

Particularly problematic is that these issues are aggravated by the preference for binary masks – continuous masks could avoid adding distinctive edges, at the price of the

[17] Meaning, neither excessively segmented, with ragged edges or other small-scale fluctiations typical for adversarial attacks.

issues discussed in Section 1.4.3B, but near-binary masks need near-instantaneous transitions. This leads to another dilemma: the necessity to find a sweet spot (Section 2.8.3) for mask regularization parameters that provides a compromise between sufficient avoidance of both artificial edges and large-scale blending. A possible solution is to use a different notion of masking altogether. One such notion, based on multiscale decomposition, is the topic of chapter 4. Related techniques have also recently been published by Kolek et al. [46][47].

CHAPTER 2

OPTIMISED ABLATION PATHS

This chapter represents the largest piece of the work done for this thesis. We published that newly developed method in [85]; the following reiterates[1] that paper with some more focus on certain technical / mathematical details.

2.1 Introduction

The concept of a mask-based interventional path was already shown in Section 1.3, the simplest example being the affine path x_{aff}, continuously connecting the inputs x_{Tg} and x_{BL}. That path is used in the Integrated Gradients method [96] as a pre-defined way to obtain a whole family of inputs for which to evaluate the classifier f, instead of only the two given ones.This has several advantages, but as argued in Section 1.4.3B affine interpolation is of dubious merit for the intermediate points. Also, in this method the interventions are without information; the actual result of the method is the namesake integral over the classifier output (Equation 1.6), which provides no assurance regarding faithfulness of the produced explanation.

By contrast, methods like Meaningful Perturbation [28] and Occluded Explanation [19] generate masks that act as both tool for generating the interventions, as well as saliency heatmaps (either directly or statistical contribution). This offers a certain amount of faithfulness since the masked input with its classification can be directly inspected by a human.

What it does not offer is much context. These individual interventional inputs will generally differ from x_{Tg} in a major way, with differences in many features simultaneously. This makes it hard to estimate how stable the explanation is, what features are only relevant in a particular combination, and which are standalone contributors. This is arguably alleviated by using multiple such interventions, which most of these methods do in some way [27][19][73]. But when the interventions are obtained individually, they can differ amongst each other just as substantially as from x_{Tg}, which does not solve the problem of many-features-at-once. It only offers more information that may or may not form a consistent picture of the classifier behaviour. What is at any rate not provided this way is a continuous relationship between interventions and scores, the like of the affine path behind Integrated Gradients.

Another way of obtaining at least a near-continuous path connecting x_{Tg} to x_{BL} is the Pixel Ablation technique [73]. It is a very different path from the affine-interpolation

[1] Some parts of this chapter appear as verbatim in the paper.

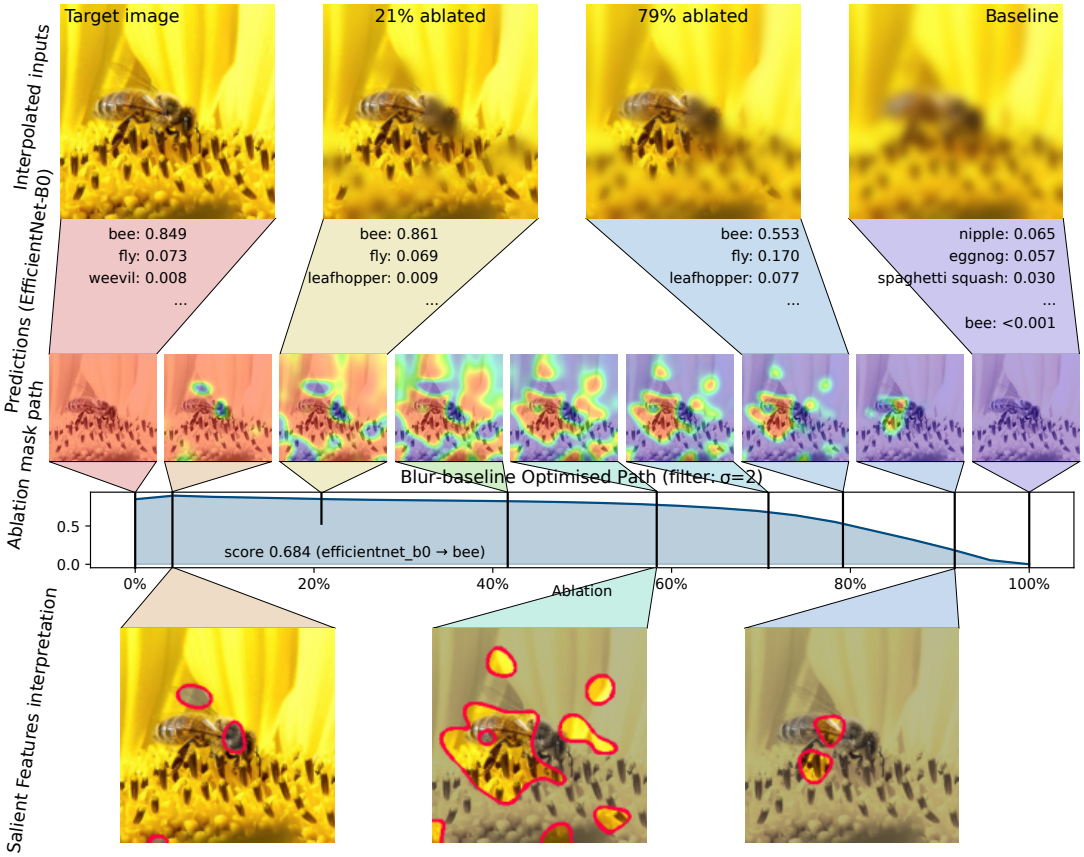


Fig. 2.1: Example of how an ablation path (sequence of masks, middle row) gives rise to a transition between a current target (a bee from ImageNet [82]) and a baseline (blurred version of the same image).

one, yet both can be seen as special cases of the same formalism:

- A sequence, or parametrization[2], of masks, called an *Ablation Path*

$$\varphi\colon [0,1] \to \mathcal{M}. \tag{2.1}$$

- The corresponding parametrisation of images, obtained by using the masks for interpolating between the inputs x_{Tg} and x_{BL}.

$$t \mapsto [x_{\mathrm{Tg}}{}^{x_{\mathrm{BL}}}](\varphi(t)). \tag{2.2}$$

  The interpolation is in this chapter always understood as $[x_{\mathrm{Tg}}{}^{x_{\mathrm{BL}}}]^{\mathrm{aff}}$ viz. Equation 1.12, though this can be generalised as in Section 4.3.

This can then be composed with the classifier F to obtain also a parametrization of classifications, cf. Figure 2.1.Specifically, the Integrated Gradients method can be formulated as using $\varphi^{\mathrm{aff}}(t) = \vartheta_{\mathrm{hom}} \cdot t$ to generate the evaluations $\nabla_x (F(x))|_{x=\left[x_{\mathrm{Tg}}{}^{x_{\mathrm{BL}}}\right](\varphi(t))}$ which are integrated (Equation 1.6).

[2]The argument, denoted with the symbol t, may be pronounced as "time" but there is no deep reason for this choice; it only resembles physical time in the sense that it happens to be a linear dimension that complements the spatial dimensions of the images / heatmaps.

The Pixel Ablation (insertion curve) of a heatmap is based on the masks of a path

$$\varphi^{\mathrm{pxIns}}\colon\ [0,1] \to \underbrace{\mathcal{M}}_{\{0,1\}^{h\times w}}$$

$$\left(\varphi^{\mathrm{pxIns}}(t)\right)_{j,k} = \begin{cases} 1 & \text{if pixel } (j,k) \text{ among the top } t \text{ most salient ones} \\ 0 & \text{else} \end{cases} \tag{2.3}$$

where "the top t most salient" is the set of $\lfloor t\cdot h\cdot w\rfloor$ pixels with highest values in the heatmap. The path φ^{pxIns} is a stepwise function, but it is easy to see that in the limit of high resolution it becomes a continuous injection to $\mathcal{L}^2(\Omega)$ if the heatmap is sufficiently smooth.

This path, like the affine one in the Integrated Gradients method, is used by evaluating F for the images $[\,x_{\mathrm{Tg}}{}^{x_{\mathrm{BL}}}]\left(\varphi^{\mathrm{pxIns}}(t)\right)$, the result being the deletion curve. And the area under this curve is, again reminiscently of Integrated Gradients, calculated as an integral:

$$\mathrm{AUC}^{\mathrm{Ins}} = \int_0^1 \mathrm{d}t\ \left(F\left([\,x_{\mathrm{Tg}}{}^{x_{\mathrm{BL}}}]\left(\varphi^{\mathrm{pxIns}}(t)\right)\right)\right). \tag{2.4}$$

As discussed in Section 1.2.4C, a faithful saliency heatmap is expected to have high values $\mathrm{AUC}^{\mathrm{Ins}}$. At its most basic, what our Ablation Path method does is to take this condition, which would otherwise only be checked post facto, as the goal itself: it optimises the expression of Equation 2.4 (or related ones, Section 2.3), with not an already given path φ^{pxIns} but instead a variable φ as the optimisation parameter. This simple idea is not quite as simple to realise, but it is possible to build a working saliency method out of it. The following sections show how.

2.1.1 *Assumptions*

A brief reiteration of the setting, as introduced in Section 1.1:

Inputs

The image $x_{\mathrm{Tg}} \in \mathcal{I}$ is being classified, which the saliency method shall explain. The baseline image x_{BL} is given to contrast it.

Classifier

The function $F\colon \mathcal{I} \to [0,1]$ (Equation 1.3) satisfies $F(x_{\mathrm{Tg}}) \approx 1$ and $F(x_{\mathrm{BL}}) \approx 0$.

Remark 5. *Alternatively,* $F(x_{\mathrm{Tg}}) \not\approx 1$ *is also possible, which means a label is explained other than the one* x_{Tg} *is classified as.*

Masks

The space $\mathcal{M}$ permits an injective operation

$$[\,x_{\mathrm{Tg}}{}^{x_{\mathrm{BL}}}]\colon \mathcal{M} \to \mathcal{I}$$

fulfilling $[{}_{x_{Tg}}{}^{x_{BL}}]\mathbf{0} = x_{Tg}$ and $[{}_{x_{Tg}}{}^{x_{BL}}]\mathbf{1} = x_{BL}$. Furthermore we require here a norm on that space, called *mass* $\overline{|\cdot|}\colon \mathcal{M} \to \mathbb{R}^+$, satisfying

$$\begin{aligned} \overline{|\mathbf{0}|} &= 0 \\ \overline{|\mathbf{1}|} &= 1. \end{aligned} \tag{2.5}$$

In practice, $\mathcal{M}$ is a cone of nonnegative-valued functions over the domain Ω, and – assuming a measure in which Ω has unit area – the mass is computed as

$$\overline{|\vartheta|} = \int_\Omega \mathrm{d}\mathbf{r}\ (\vartheta(\mathbf{r})), \tag{2.6}$$

which, given that ϑ is non-negative, is its $\mathcal{L}^1$ norm.

2.2 Axiom-based path notion

The concept of paths φ of masks will now be made precise. A path is a function of a time parameter t, yielding a mask. Since masks themselves are functions of a spatial parameter $\mathbf{r} \in \Omega$, we write interchangeably $\varphi(t, \mathbf{r}) \in \mathbb{R}$ or the curried form $\varphi(t) \equiv \varphi(t, \cdot) \in \mathcal{M}$, depending on context.

Directly optimising Equation 2.4 for arbitrary "paths" $\varphi\colon [0,1] \to \mathcal{M}$ would have a trivial solution $\varphi(t) \equiv \mathbf{0}$. This guarantees a high score because $[{}_{x_{Tg}}{}^{x_{BL}}]\mathbf{0} = x_{Tg}$ has by assumption a classification near 1, which is the highest possible one. In other words, an image's classification could always be explained by the whole image as it is, but that is utterly uninformative. It is thus necessary to define a more restricted notion of ablation path over which the optimisation is carried out. What we propose are the following axioms.

Definition 1. *The set of* ***ablation paths*** *is denoted $\mathcal{A} \subset \{\varphi\colon [0,1] \to \mathcal{M}\}$. Each path in it fulfils:*

Boundary conditions
$\varphi(0) = \mathbf{0}$ *and* $\varphi(1) = \mathbf{1}$.

Monotonicity
$t_1 \leqslant t_2 \implies \varphi(t_1) \leqslant \varphi(t_2)$ *for* $t_1, t_2 \in [0,1]$.

Constant speed
$\overline{|\varphi(t)|} = t$ *for all* $t \in [0,1]$.

The ordering in the monotonicity condition is understood pointwise (aka pixel-wise), i.e.

$$\varphi(t_1) \leqslant \varphi(t_2) \iff \forall(\mathbf{r} \in \Omega) : \varphi(t_1, \mathbf{r}) \leqslant \varphi(t_2, \mathbf{r}).$$

How these axioms can be fulfilled in practice is topic of Section 2.6. Therein, another, weaker notion of path will be useful for auxiliary purposes:

Definition 2. *Paths that obey the boundary conditions and monotonicity like in Definition 1, but do not necessarily have constant speed, are called* ***monotone paths****.*

Any monotone path gives rise to an ablation path in a canonical way; see Lemma 7.

2.2.1 *Rationale*

The boundary conditions ensure that

$$\begin{aligned}[{}_{x_{Tg}}{}^{x_{BL}}](\varphi(0)) &= x_{Tg}\\ [{}_{x_{Tg}}{}^{x_{BL}}](\varphi(1)) &= x_{BL},\end{aligned} \tag{2.7}$$

meaning the path starts at x_{Tg} and ends at x_{BL}, and therefore intuitively "connect" them. Without additional restrictions, this would not have much effect though, because it would allow contrived paths that e.g. stay at $\mathbf{0}$ for all $t < 1$ and only jump at $t = 1$ or at least at $t = 1 - \varepsilon$. In either case, scores like Equation 2.4 could be arbitrarily high without meaningfully highlighting anything. They would however not have the constant speed property, which ensures that the path has to "move away" from x_{Tg} in due time.

Monotonicity ensures that this movement cannot be erratically back-and-forth, highlighting entirely different features at different t. Without this condition, the Ablation Path technique reduces to computing a family of independent perturbation-optimisations like Fong, Patrick, and Vedaldi [27].

2.2.2 *Mathematical properties*

The boundary conditions together with monotonicity are sufficient to restrict the pointwise range of any ablation path:

$$\varphi(t, \mathbf{r}) \in [0, 1] \quad \forall (t \in [0, 1], \mathbf{r} \in \Omega). \tag{2.8}$$

The paths have thus bounded sup-norm; consequently, when ignoring null sets,

$$\mathcal{A} \subset \mathcal{L}^\infty([0, 1] \times \Omega). \tag{2.9}$$

Recall that $\mathcal{L}^\infty([0, 1] \times \Omega)$ is isomorphic to the dual space of the $\mathcal{L}^1([0, 1] \times \Omega)$ Banach space.

At a glance, it could be thought that both the boundary conditions and monotonicity follow from constant speed, but constant speed only implies the boundary conditions if the pointwise range is explicitly restricted to $[0, 1]$, which is not necessary since it also follows from the combination of boundary conditions and monotonicity.

Monotonicity is not implied by constant speed at all, since that makes no statement about pointwise growth. The combination of monotonicity and constant speed however imply a useful stronger property, that of continuity.

Lemma 2. *If φ is an ablation path as per Definition 1 whose masks have $\mathcal{L}^1$ mass (Equation 2.6), then*

$$\|\varphi(t_1) - \varphi(t_0)\|_{\mathcal{L}^1} = |t_1 - t_0|. \tag{2.10}$$

In particular, $t \mapsto \varphi(t)$ is continuous as a function $[0, 1] \to \mathcal{L}^1(\Omega)$ (as this equation witnesses in the limit $t_1 \to t_0$).

Proof. Choose t_0 and t_1 in $[0, 1]$. Without loss of generality, assume $t_1 \geqslant t_0$. Then,

$$\|\varphi(t_1) - \varphi(t_0)\|_{\mathcal{L}^1} = \int_\Omega d\mathbf{r}\ |\varphi(t_1) - \varphi(t_0)| = \int_\Omega d\mathbf{r}\ (\varphi(t_1) - \varphi(t_0))$$

due to monotonicity. Furthermore, by linearity of integration and Equation 2.6,

$$\int_\Omega d\mathbf{r}\ (\varphi(t_1) - \varphi(t_0)) = \int_\Omega d\mathbf{r}\ (\varphi(t_1)) - \int_\Omega d\mathbf{r}\ (\varphi(t_0)) = \overline{|\varphi(t_1)|} - \overline{|\varphi(t_0)|}.$$

Thanks to the constant speed axiom, this is simply $t_1 - t_0$, which is positive by assumption. It follows that

$$\|\varphi(t_1) - \varphi(t_0)\|_{\mathcal{L}^1} = t_1 - t_0 = |t_1 - t_0|,$$

from which we conclude that $\varphi(t_1) - \varphi(t_0)$ is in $\mathcal{L}^1$ and fulfills Equation 2.10. □

Continuity in $\mathcal{L}^1$ may seem a slightly obscure property, but the statement can be generalised to include, amongst others, the more familiar $\mathcal{L}^2$ notion (though not $\mathcal{L}^\infty$):

Theorem 3. *If φ is an ablation path like in Lemma 2 and $p \geqslant 1$ finite, then $t \mapsto \varphi(t)$ is continuous as a function $[0, 1] \to \mathcal{L}^p(\Omega)$.*

Proof.

$$\begin{aligned}\|\varphi(t_1) - \varphi(t_0)\|_{\mathcal{L}^p}^p &= \int_\Omega d\mathbf{r}\ |\varphi(t_1, \mathbf{r}) - \varphi(t_0, \mathbf{r})|^p \\ &= \int_\Omega d\mathbf{r}\ \left(|\varphi(t_1, \mathbf{r}) - \varphi(t_0, \mathbf{r})| \cdot |\varphi(t_1, \mathbf{r}) - \varphi(t_0, \mathbf{r})|^{p-1}\right).\end{aligned}$$

Because φ is bounded to the range $[0, 1]$, we have $-1 \leqslant \varphi(t_1, \mathbf{r}) - \varphi(t_0, \mathbf{r}) \leqslant 1$ and thus $|\varphi(t_1, \mathbf{r}) - \varphi(t_0, \mathbf{r})|^{p-1} \leqslant 1$, such that

$$\|\varphi(t_1) - \varphi(t_0)\|_{\mathcal{L}^p}^p \leqslant \int_\Omega d\mathbf{r}\ (|\varphi(t_1, \mathbf{r}) - \varphi(t_0, \mathbf{r})| \cdot 1) = \|\varphi(t_1) - \varphi(t_0)\|_{\mathcal{L}^1},$$

which is equal to $|t_1 - t_0|$ by Lemma 2. Therefore,

$$\|\varphi(t_1) - \varphi(t_0)\|_{\mathcal{L}^p} \leqslant |t_1 - t_0|^{\frac{1}{p}},$$

proving continuity. □

Remark 6. *The case $p = \infty$ is not only not covered by Theorem 3, but has real counterexamples, cf. Section 2.2.4.*

Arguably, it would be more natural to directly require continuity as part of the axioms, instead of monotonicity. The advantage of monotonicity, apart from being a stronger condition enforcing more easy interpretability on the path[3], is that it remains an exactly stateable condition also when the path is represented with discrete t steps,

[3]This restriction can also have downsides, in that it encourages interpolation-like paths even when this requires passing through off-manifold regions, cf. Section 1.4.3B.

whereas continuity requires an arbitrary-resolution limit. Continuity could only be emulated with additional regularisation, essentially smoothening out the t-valued function. Regularisation has its own problems though, generally requiring a parameter that trades off accuracy against stability; the choice of such a parameter already poses problems for the regularisation in the spatial directions (Section 2.5.1).

Monotonicity has no such difficulties. It is somewhat more challenging to implement, but we were able to find a solution that works without parameters or other problematic side effects, see Section 2.6.2B.

2.2.3 *Equivalent formulations*

An alternative representation to ablation paths (Definition 1) deserves brief mention. It stores not a sequence of masks, but instead "updates", or "patches", between masks. In essence, these represent the time-derivative of an ablation path[4],

$$\psi(t) = \frac{\partial \varphi(t)}{\partial t}. \tag{2.11}$$

Definition 3. *A **doubly-stochastic update sequence** $\psi\colon [0,1] \to \mathcal{M}$ fulfils*

Nonnegativity
$\psi(t, \mathbf{r}) \geqslant 0$ *for all* $t \in [0,1]$, $\mathbf{r} \in \Omega$.

Complementation
$\int_0^1 \mathrm{d}t\ (\psi(t)) = \mathbf{1}$.

Constant speed
$\overline{|\psi(t)|} = 1$ *for all* $t \in [0,1]$.

Remark 7. *Complementation and constant speed can also be written in a way that shows a striking symmetry between the time and spatial directions:*

$$\int_0^1 \mathrm{d}t\ (\psi(t, \mathbf{r})) = 1 \quad \forall\, \mathbf{r} \in \Omega$$
$$\int_\Omega \mathrm{d}\mathbf{r}\ (\psi(t, \mathbf{r})) = 1 \quad \forall\, t \in [0,1]\,.$$

These conditions are sufficient so that the path $\varphi(t) = \int_0^t \mathrm{d}t'\ (\psi(t'))$ is an ablation path as per Definition 1. Vice versa, if φ is an ablation path which is differentiable in time, then $\psi = \partial_t \varphi$ fulfils the axioms of Definition 3.

The constant speed requirement can also here be omitted; in this case, the path $\varphi(t) = \int_0^t \mathrm{d}t'\ (\psi(t'))$ will only be a monotone path, and may require reparameterization (Lemma 7).

[4]This requires of course a path that is differentiable, which not all ablation paths are.

2.2.4 *Saturated paths*

Definition 1 does *not* require any point-wise continuity. This is pertinent not only because it means the axioms can be used without needing to ensure such continuity, but also because there is a class of masks of particular interest that necessarily fail to be spatially continuous: *binary masks* have local values of either 0 or 1, with hard jumps at the boundary between each region. Such masks avoid the blending problematic (Section 1.4.3B), making it desirable to support them.

A concrete example are the masks in the pixel ablation method (Equation 2.3) – or their extension to continuous heatmaps $\Theta\colon \Omega \to [0,1]$, which can be formed thus:

$$\varphi^{\tilde{\text{pxIns}}}(t, \mathbf{r}) = \begin{cases} 1 & \text{if } \Theta(\mathbf{r}) > 1 - t, \\ 0 & \text{else.} \end{cases} \tag{2.12}$$

This is a monotone path after (Definition 2). Notice that the time variable is used in the opposite orientation compared to Equation 2.4, which is inconsequential for the optimisation problem but necessary to conform to the (arbitrary) convention of monotone increase rather than monotone decrease.

$\varphi^{\tilde{\text{pxIns}}}$ is an example of a path that is not continuous pointwise / as a function $[0,1] \to \mathcal{L}^\infty(\Omega)$ (and neither is its reparametrized ablation path by Lemma 7). Consequently it is also not differentiable in t, and does not permit a ψ-based representation (Equation 2.11), though in a discretised implementation this may be ignored since any finite difference is nevertheless well-defined.

2.3 Score functions

As already said, the idea behind our method is to optimise an ablation path with respect to a score like the pixel-ablation AUC (Equation 2.4). We call this the

Retaining score function

$$\begin{aligned} P_\uparrow &\colon \mathcal{A} \to \mathbb{R} \\ P_\uparrow(\varphi) &:= \int_0^1 dt \ \left(F\left([x_{Tg}{}^{x_{BL}}](\varphi(t)) \right) \right). \end{aligned} \tag{2.13}$$

This corresponds to optimising the insertion[5] metric of Petsiuk, Das, and Saenko [73]. Likewise, the converse can be optimised – their deletion metric, which we call

Dissipating score function

$$P_\downarrow(\varphi) := 1 - \int_0^1 dt \ \left(F\left([x_{Tg}{}^{x_{BL}}](\varphi(t)) \right) \right). \tag{2.14}$$

[5]One might ask why we deviate from the established terminology, i.e. "insert" vs "retain" and "delete" vs "dissipate". The reason is that the terms used by Petsiuk, Das, and Saenko [73] may describe well the procedure employed in the pixel ablation, but not so much the optimisation problem. And "insertion"

Intuitively, the first features to be deleted in a $P_\downarrow$-optimal path $\varphi_\downarrow$ should correspond roughly to the ones longest preserved in a $P_\uparrow$-optimal path $\varphi_\uparrow$, meaning that a feature that is potent at retaining the classification should be removed early on if the objective is to change the classification. More generally, one would expect $\varphi_\downarrow$ to be similar to the converse of $\varphi_\uparrow$ in opposite direction, i.e.

$$\varphi_\downarrow(t) \approx 1 - \varphi_\uparrow(1-t)?$$

We observe this to be often *not* the case: specifically, there are many examples where either the classification is so predominant that it is almost indeterminate what features should be preserved longest (because any of them is be sufficient to retain the classification), or vice versa the classification is so brittle that it is indeterminate which ones should be removed first. It is however possible to *enforce* features to be considered simultaneously in a sense of their potency to preserve the classification when they are kept, and changing it when removed. This is achieved by optimising a path with the combined objective of retaining for the path itself and dissipating for its opposite: this is expressed by optimising the

Contrastive score

$$P_\updownarrow(\varphi) := P_\uparrow(\varphi) + P_\downarrow(1-\varphi)\,. \tag{2.15}$$

This too corresponds to ideas already used by other authors, called "hybrid game" or "symmetric preservation" [28][21].

A related possibility is to train both a retaining and a dissipating path in tandem, but with additional constraints to keep them in correspondence. Here, it is most useful to keep them not opposites of each other, but rather to keep them as similar as possible. This is achieved by a score of the form

Boundary-straddling score

$$\begin{aligned} P_{\uparrow\downarrow}&:\ \mathcal{A}\times\mathcal{A}\to\mathbb{R} \\ P_{\uparrow\downarrow}(\varphi_\uparrow,\varphi_\downarrow) &:= P_\uparrow(\varphi_\uparrow) + P_\downarrow(\varphi_\downarrow) + \lambda_\pm\cdot\|\varphi_\uparrow - \varphi_\downarrow\|\,, \end{aligned} \tag{2.16}$$

where $\|\cdot\|$ could refer to various distance notions on the space of paths, and $\lambda_\pm$ parameterizes the degree to which this distance is penalized. We call the corresponding optimisation problem the *boundary-straddling method*, since (in the ideal of a classifier with exact decision boundaries) it rewards $\varphi_\uparrow$ staying in the domain of x_{Tg} as much as possible and $\varphi_\downarrow$ in the domain of x_{BL} as much as possible, i.e. on the opposite side of the decision boundary but as close as possible (Figure 2.2). Thus, $\varphi_\uparrow$ and $\varphi_\downarrow$ in effect pinch the decision boundary between them.

2.4 Optimisation strategies

Ideally speaking, optimisation should simply find the global maximum of a function. There are two main reasons why this is not doable for many applications that are

only makes sense when the path is traversed from x_{BL} to x_{Tg}, whereas we traverse by convention always from x_{Tg} to x_{BL}.

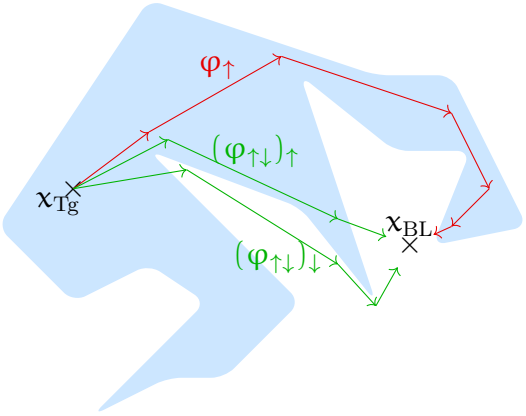


Fig. 2.2: Low-dimensional sketch of decision regions as in Figure 1.2, with a possible retaining path and a pair of boundary-straddling ones.

formally optimisation problems:

- Computational infeasibility: whilst schoolbook problems allow analytically solving for all the local minima, which can the exhaustively be compared to find the highest one, this is not possible for most practically interesting one, so that numerical approximation is a necessity.

- Even if there is a clear global maximum and it could be found with sufficient resource expense, it can be unstable, far-off, or even outside of the space one conceptually wants to work in. This is technically speaking a symptom that the problem setup was not right in the first place, but a better one may simply not be known. It can nevertheless be possible to use an algorithm that, despite according to theory only giving an approximation to the optimal solution, is in practice *better* behaved than the exact maximum would be.

Training of deep neural networks is a well-known example subject to both of these points. The standard approach to training such models is to randomly select a start state and apply a finite number of stochastic gradient descent steps, a strategy that has little in the way of theoretical guarantees but is vindicated where results empirically outperform more principled approaches.

Gradient descent per se is a rigorous enough technique when used for specific types of problem. For convex optimisation problems, there is only one unique extremum and gradient descent can be made to approach it (though not necessarily with good speed; there may be oscillations and other problems). Also many more complex nonlinear problems behave convex in the vicinity of their local extrema, making gradient descent usable to narrow down local optima (which may however be far from globally optimal. The stochastic element of SGD is the primary means of avoiding to converge too soon on an insufficiently good local extremum. The mechanism is related to well-understood instances like simulated (or real metallurgic) annealing, and has also in its concrete form some theoretical justification from Bayesian statistics [94], though whether it works in a given situation like training a particular CNN architecture on a given dataset is up to experiment.

Gradient descent also has precedent for saliency purposes, being used in the Meaningful Perturbation method [28], which is after all the method most closely related to ours. While we also considered gradient-free methods such as simulated annealing, these generally require far more evaluations and/or scale badly with dimension of the domain, which is a severe challenge when both the dimension is high and the evaluations expensive, as they are in our case (path of image-like masks; evaluation of deep classifier along the whole path).

2.4.1 *Iteration and start state*

An iterative procedure like gradient descent needs to start with a pre-selected state, in this case an initial path φ_0. This should not introduce anything that biases the result, nor should it prevent the algorithm from proceeding – specifically an already saturated path would make this challenging for the reasons in Section 2.4.2A.

The choice of start state that fulfills this is the affine path

$$\varphi_0(t, \mathbf{r}) = \varphi^{\text{aff}}(t, \mathbf{r}) = \vartheta_{\text{hom}} \cdot t(\mathbf{r}) = t, \tag{2.17}$$

for largely the same reasons as Sundararajan, Taly, and Yan [96] also argued it to be the canonical choice for Integrated Gradients. Unlike in that method, the start state in ours does not commit the saliency result to depend only on images interpolated with homogeneous blending mask (cf. Section 1.4.3B), because later iterations can and will be carried out with stronger or even binary masks.

2.4.2 *Constrained gradient descent*

Remark 8. *We continue using the term "gradient descent" for consistency, though since our problem is formulated as maximising the score functions it is actually a gradient* ascent.

Unlike with Meaningful Perturbations, our method demands optimisation not in a straightforward slice of a vector space, but in the space $\mathcal{A}$ with hard constraints in form of Definition 1. Of these, the boundary conditions and speed constraints by themselves are harmless (essentially linear projection), but monotonicity is nontrivial. There are broadly speaking three approaches by which to use gradient descent in such a setting:

2.4.2A *Exact manifold*

If the space $\mathcal{A}$ could directly be parameterised as a manifold, then a gradient would lie in the tangent space, and a suitable exponential map[6] could be used to apply the update step. This is closely related to numerical integration on manifolds [42].

Unfortunately, $\mathcal{A}$ is not a manifold, as witnessed by the fact that different ablation paths have very different degrees of freedom: an unsaturated path like φ^{aff} (cf. Section 2.1) permits any sufficiently small, smooth $[0, 1] \to \mathcal{M}$ function as a perturbation[7]

[6] This is a rigorous notion on a manifold with affine connection, in particular a Riemannian manifold. Exponentials are particularly useful on Lie Groups (which are briefly discussed in Section 5.2.1).

[7] The word "perturbation" here used in the sense of small-magnitude change around the mask-path $\varphi + \delta$, not as in Meaningful Perturbations where the masks themselves are considered pertubations to an image.

because adding something of small slope to the constant-slope φ^{aff} still leaves a positive slope (preserving monotonicity). On the other hand, a fully saturated path like φ^{pxIns} is constant at most points $(t, \mathbf{r})$ and will therefore cease to be monotone if perturbed with a function that is even slightly decreasing there.

2.4.2B *Simpler, dense subset*

Even though $\mathcal{A}$ itself is not a manifold, its interior is, specifically those paths that are pointwise *strictly* monotone. Unlike $\mathcal{A}$, this is not a compact space, which leads to challenges of its own (particularly concerning convergence/termination). But at least in principle it is possible to carry out gradient descent steps in this noncompact space and use the embedding into $\mathcal{A}$ as the end result.

This is one approach where the ψ-representation (Equation 2.11) was considered as potentially preferrable to φ, since the domain-interior has a more convenient description: strictly monotone paths φ correspond to strictly positive updates ψ, and those can be represented as a pointwise exponential (or other homeomorphism $\mathbb{R}^+ \leftrightarrow \mathbb{R}$) of an unrestricted function $\tilde{\psi}$,

$$\psi(t, \mathbf{r}) = \exp\left(\tilde{\psi}(t, \mathbf{r})\right). \tag{2.18}$$

Non-strictly monotone paths, including saturated ones, could still be approximated arbitrarily well by way of $\tilde{\psi}(t, \mathbf{r}) \ll 0 \Longrightarrow \psi(t, \mathbf{r}) \approx 0$.

This might work well if the paths of interest were mostly unsaturated / strictly-monotone, but this is not the case: solutions are expected to be *mostly* saturated, and/or saturation is desired for interpretability and the reasons listed in Section 1.4.3B. For such highly saturated paths, the large amplitudes required in a $\tilde{\psi}$ makes gradient descent highly unstable, specifically when also the other constraints and regularisations are taken into account. In our experiments, this approach largely failed to give usable results.

2.4.2C *Embedding—projection*

The score functions of Section 2.3 are defined not only for paths $\varphi \in \mathcal{A}$, but for any integrable functions $\varphi\colon [0,1] \to \mathcal{M}$. In the space $\mathcal{L}^2([0,1] \times \mathcal{M})$ (as implicit with the usual pixel representation), is is easy to compute the gradient and apply an update of step size h.

$$\varphi_{i+1}^{\mathcal{L}^2} = \varphi_i + h \cdot \nabla(P(\varphi))|_{\varphi=\varphi_i}. \tag{2.19}$$

It is understood that this will result in a path $\varphi_{i+1}^{\mathcal{L}^2} \notin \mathcal{A}$, violating one or more of the path axioms. This can still be useful though, if one can project it back to $\mathcal{A}$; conceptually

$$\varphi_{i+1} = \operatorname*{arg\,min}_{\varphi \in \mathcal{A}} \left(\left\| \varphi - \varphi_{i+1}^{\mathcal{L}^2} \right\| \right), \tag{2.20}$$

where various norms could be used. This will in general not have a unique solution, but any solution will only deviate as little as possible. In particular, in the limit of small step size h and consequently $\varphi_{i+1}^{\mathcal{L}^2}$ almost in $\mathcal{A}$, the updated and projected φ_{i+1} will

deviate arbitrarily little from φ_i plus its locally tangential contribution of the update.[8]

Similar projection techniques are well-established in convex optimisation, where *proximal operators* [66] have a solid theoretical backing. For the nonconvex problem we deal with here, this is not so much the case, but we have nevertheless been able to obtain good results in many cases, though it required some amount of experimentation and parameter-tweaking; more details in Section 2.6. A more mathematically principled technique would be desirable, but it is doubtful if that is possible without substantial assumptions about the classifier F.

The gradient itself in Equation 2.19 is easy to obtain for all the proposed score functions. It suffices to carry out the computation for the archetypical retaining score.

Lemma 4. *The gradient of the retaining score, with a differentiable classifier* F *and with respect to the* $\mathcal{L}^2(\Omega)$ *space, is*

$$\nabla(P_\uparrow(\varphi))(t,\mathbf{r}) = (x_{BL} - x_{Tg})(\mathbf{r}) \cdot \nabla(F(x))|_{x=\left[x_{Tg}^{x_{BL}}\right]_{(\varphi(t))}}(\mathbf{r}), \tag{2.21}$$

or, written in eta-reduced form of the spatial argument

$$\nabla(P_\uparrow(\varphi))(t) = (x_{BL} - x_{Tg}) \odot \nabla(F(x))|_{x=\left[x_{Tg}^{x_{BL}}\right]_{(\varphi(t))}}. \tag{2.22}$$

Proof. The purpose of the differential is to describe the behaviour of the score function for small deviations $\delta\varphi$,

$$\begin{aligned} P_\uparrow(\varphi+\delta\varphi) &= \int_0^1 dt \left(F\left([x_{Tg}^{x_{BL}}](\varphi(t)+\delta\varphi(t))\right)\right) \\ &= \int_0^1 dt \left(F(x_{Tg}+\varphi(t)\odot(x_{BL}-x_{Tg})+\delta\varphi(t)\odot(x_{BL}-x_{Tg}))\right). \end{aligned}$$

Taylor-expand the integrand, using differentiability of F:

$$\begin{aligned} &F(x_{Tg}+\varphi(t)\odot(x_{BL}-x_{Tg})+\delta\varphi(t)\odot(x_{BL}-x_{Tg})) \\ &= F(x_{Tg}+\varphi(t)\odot(x_{BL}-x_{Tg})) \\ &+ \int_\Omega d\mathbf{r} \left((x_{BL}-x_{Tg})(\mathbf{r}) \cdot (\nabla(P_\uparrow(\vartheta)))_{\vartheta=\delta\varphi(t)}(\mathbf{r}) \cdot (\delta\varphi(t,\mathbf{r}))\right) \\ &+ \mathcal{O}\left((\delta\varphi)^2\right). \end{aligned}$$

The middle term is the interesting one,

$$M(t,\delta\varphi) := \int_\Omega d\mathbf{r} \left((x_{BL}-x_{Tg})(\mathbf{r}) \cdot (\nabla(P_\uparrow(\vartheta)))_{\vartheta=\delta\varphi(t)}(\mathbf{r}) \cdot (\delta\varphi(t,\mathbf{r}))\right).$$

[8] In other words, the technique is *consistent*, if we adapt the language of ODE integrators.

We shall match this with the gradient's inner product with $\delta\varphi$:

$$\langle \nabla(\mathsf{P}_\uparrow(\varphi)), \delta\varphi \rangle_{\mathcal{L}^2} = \int_0^1 \mathrm{d}t \int_\Omega \mathrm{d}\mathbf{r}\ (\nabla(\mathsf{P}_\uparrow(\varphi))(t,\mathbf{r}) \cdot (\delta\varphi(t,\mathbf{r}))).$$

Inserting Equation 2.21 here, this is

$$\begin{aligned}
&= \int_0^1 \mathrm{d}t \int_\Omega \mathrm{d}\mathbf{r}\ \left((x_{\mathrm{BL}} - x_{\mathrm{Tg}})(\mathbf{r}) \cdot \nabla(\mathsf{F}(x))|_{x=\left[x_{\mathrm{Tg}}\ x_{\mathrm{BL}}\right](\varphi(t))}(\mathbf{r}) \cdot (\delta\varphi(t,\mathbf{r})) \right) \\
&= \int_0^1 \mathrm{d}t\ (\mathsf{M}(t,\delta\varphi)).
\end{aligned}$$

Taken together, we have

$$\begin{aligned}
\mathsf{P}_\uparrow(\varphi+\delta\varphi) &= \int_0^1 \mathrm{d}t\ (\mathsf{F}(x_{\mathrm{Tg}} + \varphi(t) \odot (x_{\mathrm{BL}} - x_{\mathrm{Tg}}))) + \int_0^1 \mathrm{d}t\ (\mathsf{M}(t,\delta\varphi)) + \mathcal{O}\left((\delta\varphi)^2\right) \\
&= \mathsf{P}_\uparrow(\varphi) + \langle \nabla(\mathsf{P}_\uparrow(\varphi)), \delta\varphi \rangle_{\mathcal{L}^2} + \mathcal{O}\left((\delta\varphi)^2\right),
\end{aligned}$$

as required by the defining condition for the gradient, Equation 1.19. □

A direct consequence of this calculation makes the connection between our method and the one by Sundararajan, Taly, and Yan [96] even clearer:

Theorem 5. *The time-integral over the gradient of the retaining score evaluated for the affine interpolation path is the integrated-gradient saliency.*

$$\int_0^1 \mathrm{d}t\ (\nabla(\mathsf{P}_\uparrow(\varphi))(t))\Big|_{\varphi=\varphi^{\mathrm{aff}}} = \mathsf{IntegratedGrads}. \tag{2.23}$$

Proof. This is the chaining of Equation 2.22 and Equation 1.6. □

2.4.3 *Stochasticity*

Using randomness to avoid getting stuck in suboptimal extrema is a fairly general idea, which can be carried out in very different ways. The main way it is usually done in machine learning is by pulling random batches from the available training set, which may additionally have randomized data augmentation applied to them. This emulates the actual random process that generated the training data in the first place. Likewise, many Monte Carlo algorithms use randomness emulating a concrete physical process, such as radioactive particle emmission.

2.4.3A *Disturbance*

The Ablation Path optimisation problem has no suggestive physical random process at hand. One alternative is to artificially add random disturbances to the paths between and/or after the gradient descent steps. This is also the notion of stochasticity used by Fong and Vedaldi [28] in their SGD optimisation for individual masks.

2.4.3B *Collapse*

To similar effect, the application of masks to images can be performed in a randomised fashion. This is particularly interesting from the stance that binary masks are preferred but binary paths are problematic for optimisation. The dilemma can be circumvented by interpreting the $\varphi(t, \mathbf{r})$ as *probabilities* that the pixel at $\mathbf{r}$ is taken from x_{BL} (and not x_{Tg}), rather than interpolation coefficients between these images. The result, when using a discretised t-axis with $n+1$ samples, is a sequence of images $(x_i)_i$ such that as before $x_0 = x_{Tg}$ and $x_n = x_{BL}$, but the sequence does not represent a continuous path in $\mathcal{I}$ anymore, though $x_i(\mathbf{r})$ will have a higher probability of being equal $x_j(\mathbf{r})$ when i and j are similar.

2.4.3C *Baselines*

Unlike the target image x_{Tg}, which is to be explained and therefore must be fixed, the baseline x_{BL} is only auxiliary and routinely chosen ad hoc to whatever facilitates the most successful explanations (in e.g. the pointing game sense, Section 1.2.4B). Since $x_{BL} \in \mathcal{I}$, it is suggestive to use another real image from a dataset; this has the advantage over a synthetic image that it is guaranteed on-manifold. This choice of baseline image is a natural candidate source for randomness, which not only introduces stochasticity but also avoids committing any single choice of baseline. Using a single image from a dataset would have disproportionate influence on the saliency result, which is why single choices of baseline in the literature generally try to be "neutral"; however that itself is only a heuristic notion. This makes random sampling of baseline an attractive alternative.

We implemented versions of all the above sources of stochasticity, but experimentally none of them proved to be an improvement over the deterministic-descent version. A main reason seems to be that any randomness in the update step Equation 2.19 leads to stronger violation of the path axioms than a pure interpolation update with neutral baseline. As a consequence, it also demands a more intrusive repair in the the projection step Equation 2.20, which in turn prevents the iteration from making any consistent progress, even when many steps are carried out and momentum [81] used.

2.5 Soft constraints

2.5.1 *Regularisation*

Although an ablation path has a notion of continuity built in by way of Theorem 3, this provides regularity only in the path's time direction, which is an advantage primarily for interpretability purposes, whereas hopes that it might also provide some inherent

protection against adversarial behaviour (Section 1.4.3A) have largely been disproven by experiment, see Section 2.7.3.

No spatial regularity is guaranteed, so that much like in case of the Meaningful Perturbation method [28], optimising with respect to any of the scores in Section 2.3 under only the constraints of Definition 1 leads to a path with an extremely high score (arbitrarily close to 1, meaning that even the smallest masks on the path lead to a near-perfect classification, indeed typically *higher* than the class archetype x_{Tg} itself). The standard strategy against such adversarial results is to impose additional conditions on the smoothness of the masks. Such conditions can be added to our method in a similar way as the Extremal Perturbation method [27]. The primary tool is a convolution that smoothens out high-frequency components of the mask before applying it to the images. We specifically use this as part of the projection back to the allowed set of paths after a gradient descent step is applied.

Fong, Patrick, and Vedaldi [27] replaced the convolution operator with a "smooth max-conv", whose purpose is to avoid (at least at most places) losing the binary character of highly-saturated masks. This specific operation is unfortunately badly suited for the ablation path method, because it involves masks of both very low and very high mass (Equation 2.6), but the max-conv treats these cases asymmetrically and in particular increases even extremely small masks to a substantial mass.

We use an ordinary Gaussian convolution instead, which does not have these problems. The observation by Fong, Patrick, and Vedaldi [27] that Gaussian filtering disrupts the binary property remains true, but in principle this should not be too problematic since a Gaussian kernel is fast-decaying so that regions[9] of substantial induced non-binariness are limited to a size on the order of few σ widths. The max-conv version has such an effect too affecting the value-0 regions; where it differs is in the regard that values of 1 are exactly preserved.

We apply the smoothening after each gradient-descent step Equation 2.19; this can be seen as a soft-projection onto the subspace of smooth masks (in addition to the hard projection to the space of ablation paths).

$$\varphi(t) \leftarrow \gamma_{\sigma_{\text{reguBlur}}} \star \varphi(t), \tag{2.24}$$

with the Gaussian kernel of dimension $n = \dim(\Omega) = 2$,

$$\begin{aligned} \gamma_\sigma &: \mathbb{R}^n \to \mathbb{R} \\ \gamma_\sigma(\mathbf{r}) &:= \frac{1}{\left(\sqrt{2\cdot\pi}\cdot\sigma\right)^n} \cdot e^{-\frac{\|\mathbf{r}\|^2}{2\cdot\sigma^2}}. \end{aligned} \tag{2.25}$$

The $\leftarrow$ notation is used in Equation 2.24 and onwards, as conventional, to express an in-place update of the current state for the ablation path under optimisation.

An alternative that was considered is to instead smooth the gradient itself before performing the update, which also has the nice interpretation in that the smoothing

[9] This assumes a mask that is not too fragmented. If it is fractal-like, with values both close to 0 and close to 1 near to each other everywhere, then Gaussian filtering will smooth them to everywhere some value in the middle. Then again, such fragmented masks are just what regularisation is supposed to prevent happening in the first place.

corresponds to the covariance operator corresponding to a more edge-sensitive metric on the space of masks that maps the differentials (dual vectors) to update-usable gradients as vectors, cf. Section 1.4.2. This too works, often with similar effect, but since a Gaussian filter does not completely suppress high frequencies it still allows these to accumulate over many iteration of the descent.

Applying the filter to the iterated optimisation state instead avoids this. It also has the effect that mask-contrasts introduced early during the optimisation which do however not appear in the gradients later on get progressively smoothed away completely, which may be seen as either an advantage or disadvantage: it is an advantage if the early gradients were sporadic, possibly caused by the blending (Section 1.4.3B) of the start state per Equation 2.17. It can be a disadvantage if the gradients become (at least locally) very small late in the iteration, which is not untypical because the classifier tends to be mostly class-saturated when evaluated along a well-optimised path.

We also considered using a sinc filter (brickwall in Fourier domain). Since that is a proper projection, the above questions do not arise, but it has its own problems that are more severe: it is much more delocalised than a Gaussian filter and lacks positivity, which is particularly problematic since it interferes with the projection of Equation 2.20.

See Section 2.8.3 for details on how regularisation affects optimised path results.

2.5.2 *Saturation*

Our method includes a dedicated means of preserving (near-) binary masks, and also attaining them in the first place: artificial saturation. This involves slightly modifying the signal such that values below 0.5 are progressively stronger pushed towards zero, and values above 0.5 towards one (Figure 2.3). The concrete expression we use for this is, defined pointwise,

$$\varphi\left(t,\mathbf{r}\right)\leftarrow\Pi_{\text{sat}}\left(\varphi\left(t,\mathbf{r}\right)\right):=\frac{1}{2}\cdot\left(\frac{\tanh\left(\left(\varphi\left(t,\mathbf{r}\right)\cdot 2-1\right)\cdot\zeta_{\text{sat}}\right)}{\tanh\left(\zeta_{\text{sat}}\right)}+1\right),\tag{2.26}$$

where the parameter ζ_{sat} determines how strongly binary values are encouraged. The exact formula for Equation 2.26 is uncritical; what matters is that it is a smooth, monotone function that has 0 and 1 as attractive fixpoints, $\frac{1}{2}$ as a repulsive fixpoint, and approches the identity in the limit $\zeta_{\text{sat}}\to 0$.

In practice, relatively low values $\zeta_{\text{sat}}<1$ are used, so that the first gradient-descent steps proceed almost unaffected and only mask-regions that are already significantly saturated are progressively further nudged towards the binary extremes. Selecting too high values for ζ_{sat} risks overwhelming the optimisation. In this case, even a slight contrast in an early state of the optimisation would get amplified to high saturation, which has the side-effect of largely "locking in" the first selected features. Since the first steps correspond to the Integrated Gradient method by Theorem 5, the consequence is that our method's result then degenerates to a binary-clipped version of that saliency method, rather than informing about the classifier beaviour along the more meaningful ablation path.

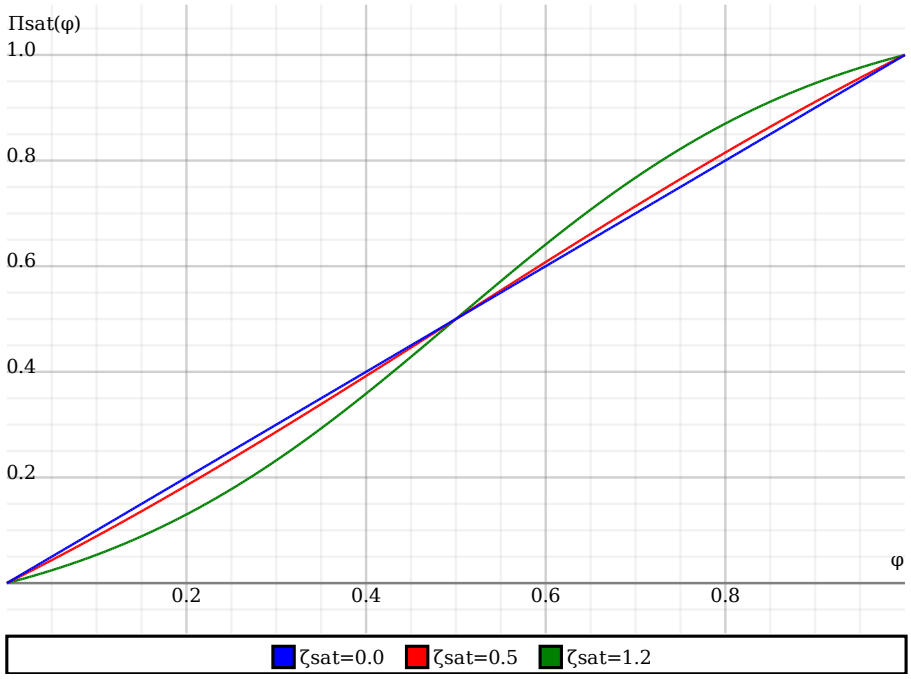


Fig. 2.3: The pointwise soft-projection function for artificial path saturation. The symbol φ here represents a single pointwise value $\varphi\,(t, \mathbf{r})$ of the ablation path.

2.5.3 *Boundary-pinching*

For the boundary-straddling method (Equation 2.13) there is another requirement: making $\varphi_\uparrow$ and $\varphi_\downarrow$ similar to each other can be achieved by explicitly penalizing their distance in the score function, but in our implementation this too is done by a dedicated algorithm step that manipulates the masks pointwise to become more similar. For interpretability purposes it is particularly desirable for $\varphi_\uparrow\,(t)$ to contain only few features that $\varphi_\downarrow\,(t)$ does not, since that allows direct comparison between two images showing how inclusion of these features bring the classification into the target class. The exact difference in strength of features meanwhile is less relevant (even when the masks themselves are not boolean). Accordingly, we suggest a *pinching tweak* that diminishes specifically the smaller positive differences between $\varphi_\uparrow$ and $\varphi_\downarrow$, in addition to any negative differences. The manifestation used in our experiments is of this form: (recall that values close to 1 correspond to masked-*away* features)

$$\varphi_\downarrow\,(t, \mathbf{r}) \leftarrow \varphi_\uparrow\,(t, \mathbf{r}) + \Pi_{\text{pinch}}\,(\varphi_\downarrow\,(t, \mathbf{r}) - \varphi_\uparrow\,(t, \mathbf{r})) \tag{2.27}$$

where

$$\begin{aligned} \Pi_{\text{pinch}}\colon\ [-1, 1] &\to [-1, 1] \\ \delta &\mapsto \Pi_{\text{pinch}}\,(\delta) \end{aligned} \tag{2.28}$$

is a continuous function with an attractive fixpoint at $\delta = 0$, and a repulsive one at $\delta = 1$ (see Figure 2.4). The former is responsible for squelching unsubstantial contrasts between $\varphi_\uparrow$ and $\varphi_\downarrow$. The latter allows the most salient features of $\varphi_\uparrow$ to stay absent from $\varphi_\downarrow$, as necessary for a high $P_{\uparrow\downarrow}$.The concrete definition of Π_{pinch} is again uncritical;

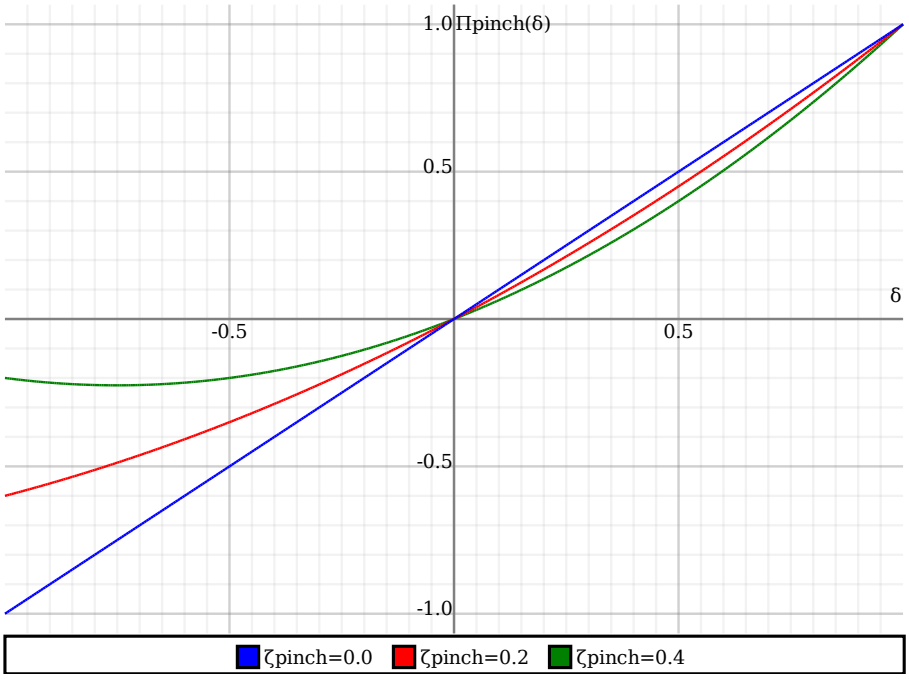


Fig. 2.4: The pointwise difference-pinching function for the boundary-straddling method.

in our experiments we used a simple polynomial expression:

$$\Pi_{\text{pinch}}(\delta) := \delta \cdot (1 - \zeta_{\text{pinch}}) + \delta^2 \cdot \zeta_{\text{pinch}}. \tag{2.29}$$

Notice that in Equation 2.27, $\varphi_{\uparrow}$ is not affected by $\varphi_{\downarrow}$, only vice versa. But conceptually, the update is performing a change to δ, i.e. the difference between the paths, rather than either of them individually.

2.6 Implementation

2.6.1 *Data structures*

The paths are in practice stored (for the image application in pixel representation) as 3-dimensional arrays aka tensors, with dimension 0 representing the time axis and dimensions 1 and 2 representing the spatial axes. Such arrays can readily be processed on both CPU and GPU, which is in our case largely taken care for by the PyTorch framework [29]. Concretely, this layout matches well the conventional data format of *batches of images*, which is how the mainstream deep-learning classifiers are also trained. The framework also provides the gradients required for optimisation, calculated with reverse-mode automatic differentiation (backprop).

The concrete shapes of the discretised φ depend on the images to be explained. Ideally one would sample the time axis very tightly and use the image resolution also for the masks, but this is expensive and unnecessary since (usable) paths have more regularity than general sequences of images. Particularly each sample on the t-axis is expensive, since it demands an entire evaluation of the classifier (forward and backward-gradient). While these can be batched on the GPU, we only have limited

GPU memory available.

As for the spatial resolution, how high this needs to be is mostly dependent on the regularisation. The masks can only have so much detail in them as permitted by the filtering. Consequently they can be represented at fairly low resolution, which means the implementation of the interpolation operation $[\,{}_{x_{Tg}}{}^{x_{BL}}\,]$ needs to also perform suitable resampling to carry out the pointwise multiplications. This also has precedent in the related saliency methods [73][28].

We carried out most of the tests with $14 \times 64 \times 64$ arrays, which seems to be appropriate for the Pascal [26] and COCO [56] datasets and many ImageNet [82] examples.

2.6.2 *Projections*

We will now explain how the correction step Equation 2.20 can be realised in detail, for each of the axioms from Definition 1.

2.6.2A *Boundary conditions*

Since the boundaries $\varphi(0)$ and $\varphi(1)$ have completely fixed prescribed values, reinstating is only a matter of literally writing zeroes and ones into the array, respectively. Indeed this is not even necessary: because these values are known a-priori, and would always give the same classifier output, we do not include them in the array in the first place, i.e. if we intend to sample n time slices, these will be at

$$t \in \left\{ \frac{1}{n+1}, \frac{2}{n+1}, \ldots, \frac{n}{n+1} \right\}, \tag{2.30}$$

a set which includes neither 0 nor 1.

Both the hard-enforcing and omitting strategies would not be so trivial if the axioms contained explicit continuity/regularity. But because the time-continuity is emergent via Theorem 3, it is possible to treat these entries separately from all the others. However, with the boundary-omitting array representation it needs to be ensured that the 0- and 1-values are still taken into account by the other processing steps. This amounts to letting these steps treat the $\varphi(0)$ and $\varphi(1)$ values as "read only parts" of the state array.

2.6.2B *Monotonicity*

The non-local monotonicity condition is the most challenging one for the projection step. We have attempted multiple ways of solving it, only one of which deemed to be usable.

Update clipping

Intuitively, this condition seems to be easier to fulfil in the ψ representation viz. Definition 3. There, it is a pointwise nonnegativity condition, which could be fulfilled by clipping the values to zero or larger. That would however interfere with the other

conditions, and though it is possible to repair these afterwards[10], this would be a non-local correction and in particular not be able to act even approximately local to where the original nonmonotonicity was. As a result, this approach is unstable except under unrealistic homogenity assumptions; specifically, we found counterexamples where an arbitrarily small noise perturbation leads to a substantial deviation in the result after both corrections, see Figure 2.6.

Descent flattening

The algorithm we decided on using instead manipulates the path φ near-locally, avoiding the above problems. This works for each $\mathbf{r}$ separately (in other words, pixel-wise), so the problem is reduced to the one of monotonising a $[0,1] \to [0,1]$ function. The

Algorithm 1 Make a function $p\colon [0,1] \to \mathbb{R}$ nondecreasing

$\cup_i [l_i, r_i] \leftarrow \{\, t \in [0,1] \mid p'(t) \leqslant 0 \,\}$ ▷ Union of intervals where p decreases
for i **do**
  $m_i \leftarrow \frac{p(l_i)+p(r_i)}{2}$
  $l_i \leftarrow \max\{\, t \in [r_{i-1}, l_i] \mid p(t) \leqslant m_i \,\}$
  $r_i \leftarrow \min\{\, t \in [r_i, l_{i+1}] \mid p(t) \geqslant m_i \,\}$
for i, j **do**
  if $[l_i, r_i] \cap [l_j, r_j] \neq \{\}$ **then**
    if $m_j < m_i$, merge the intervals and recompute m as the new center
return $t \mapsto \begin{cases} p(t) & \text{if } t \notin \cup_i [l_i, r_i] \\ m_i & \text{if } t \in [l_i, r_i] \end{cases}$

algorithm is easiest understood by example, see Figure 2.5. Working pixel-wise is not without disadvantages, both mathematical and technical. First, it makes the algorithm oblivious to whatever regularity the masks might have in $\mathcal{M}$. This turns out not to be too problematic in practice though, because such regularity also causes any changes performed by the monotonisation algorithm to be similar in nearby pixels, thanks to its stability.

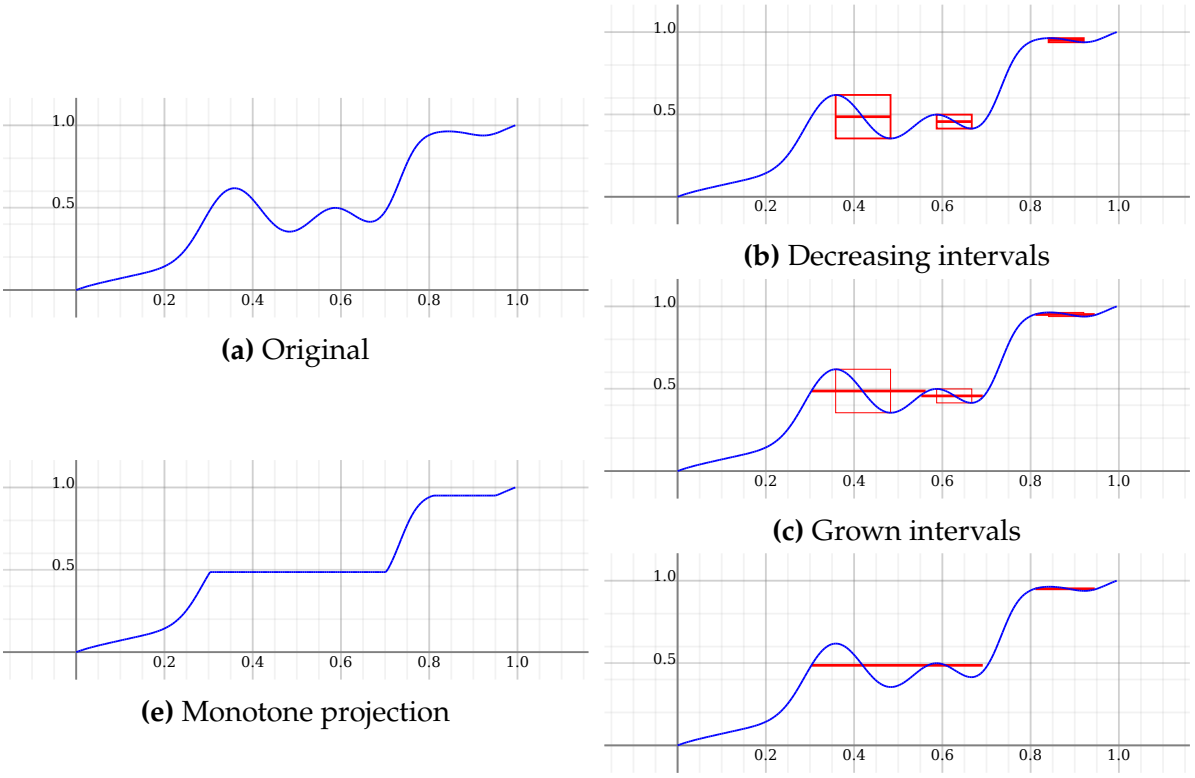
On the technical side, it has the disadvantage that the calculations cannot be expressed in terms of standard tensor operations provided by the GPU-capable framework. That is not to say that it could not be implemented on GPU, but it would require manual writing of a low-level language. We currently have only a CPU implementation[11], which is sufficiently fast not to be a bottleneck of Algorithm 2 (since the classifier evaluations dominate the computational demands), though this did require reasonably performant programming.

The most compelling argument for this method is that it is optimal in the following sense:

Lemma 6. *The result $\acute{p}$ of Algorithm 1 applied to a function $p\colon [0,1] \to [0,1]$ has the minimum possible $\mathcal{L}^\infty$ distance from p for a monotone function.*

[10] Without disrupting monotonicity again.

[11] Initially implemented in Haskell, then ported to Python to work together with the remaining PyTorch code. The vanilla Python version was too slow, but could be sped up by using Numba.



(a) Original

(b) Decreasing intervals

(c) Grown intervals

(d) Merged intervals

(e) Monotone projection

Fig. 2.5: Example view of the monotonisation algorithm in practice. (a) contains decreasing intervals, which have been localised in (b). For each interval, the centerline is then extended to meet the original path non-decreasingly (c). In some cases, this will cause intervals overlapping; in this case merge them to a single interval and re-grow from the corresponding centerline (d). Finally, replace the path on the intervals with their centerline (e).

Proof. Assume for the sake of contradiction that there is a function $\tilde{p}\colon [0,1] \to [0,1]$ with

$$\|\tilde{p} - p\|_{\mathcal{L}^\infty} < \|\acute{p} - p\|_{\mathcal{L}^\infty}.$$

That means that $|\tilde{p}(t) - p(t)|$ is everywhere smaller than the greatest value $\left|\acute{p}\left(\hat{t}\right) - p\left(\hat{t}\right)\right|$ attains. This necessarily corresponds to one of the distances $|m - p(t)|$, where

$$m = \frac{p(l) + p(r)}{2}$$

from the calculation in the algorithm. Here, $p(l)$ is a local maximum by construction of the descending-intervals, $p(r)$ a local minimum, and $l < r$. Notice that

$$|m - p(l)| = |m - p(r)|,$$

i.e. either of them alone would effect the known $\mathcal{L}^\infty$-norm, and we can select either $\hat{t} = l$ or $\hat{t} = r$. For both, the assumption, ensures $\left|\tilde{p}\left(\hat{t}\right) - p\left(\hat{t}\right)\right| < \left|\acute{p}\left(\hat{t}\right) - p\left(\hat{t}\right)\right|$. In

particular,

$$|\tilde{p}(l) - p(l)| < m - p(r) \qquad |\tilde{p}(r) - p(r)| < p(l) - m,$$

which due to $p(l) > p(r)$ implies that $\tilde{p}(l) > \tilde{p}(r)$, meaning $\tilde{p}$ is not monotone. It follows that any function which *is* monotone cannot satisfy the assumption of having a lower $\mathcal{L}^\infty$-distance to p. □

Recall that $\mathcal{L}^\infty$-optimality does not uniquely identify a solution. It is therefore possible that other algorithms could produce results that are both also optimal in the $\mathcal{L}^\infty$ sense and superior in other ways. Moreover, it is valid to ask whether $\mathcal{L}^\infty$ is the most relevant metric in the first place.

We do therefore not claim that Algorithm 1 is the canonical way of making general functions monotone, only that it has expedient properties and appears to work well for the intended use case.

Generic optimisation problem

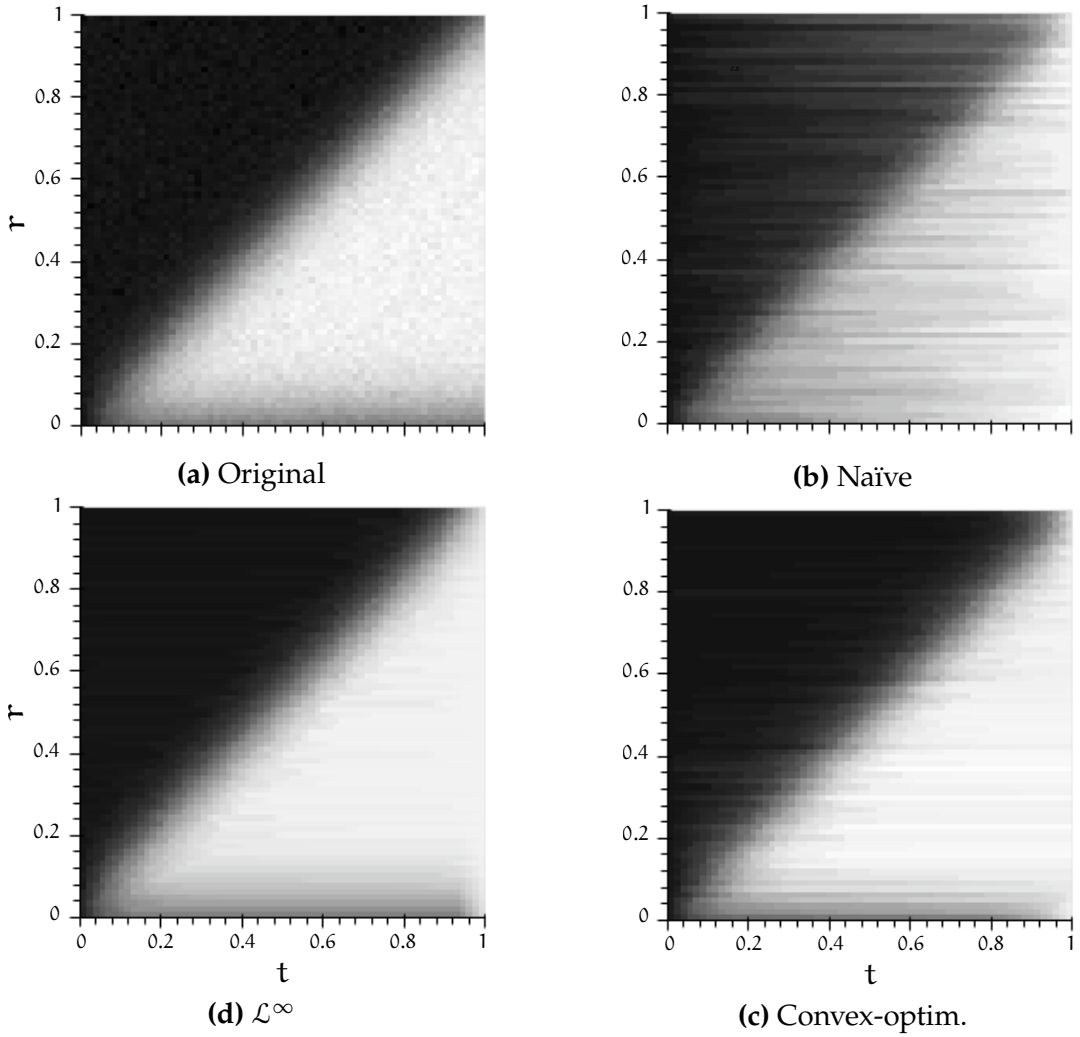
Instead of a bespoke algorithm that can be proven to fulfil a property like Lemma 6, one can also use an off-the-shelf solver to find a monotone function nearest to the current state. The main advantages of this are much higher flexibility (any norm can be used, rather than a specific one like $\mathcal{L}^\infty$) and being able to tap into existing work. The main disadvantage is that such a solver can not exploit the domain knowledge of this highly specific problem.

It does fall in the well-researched category of convex optimisation, because the set of monotone functions is a convex subset of the set of general functions on an interval and norms are convex functions. Convex optimisation is tractable compared to nonlinear optimisation (like the path-optimisation), but still generally requires iterative methods. Particularly rigid, non-smooth constraints can be challenging for them, and this includes the pointwise monotonicity condition. We essayed the odl library [34] for this purpose. It includes several solver algorithms, which will not be discussed here in detail. It was straightforward to set this up to solve the problem iteratively using the Primal-Dual Hybrid Gradient [18] using an $\mathcal{L}^2$ or H^1 norm. The latter (Sobolev space) were particularly interesting because they subsume also regularity, but we found this to be of little use (or indeed counterproductive[12]) when applied to a distance $\|\varphi - \acute{\varphi}\|$.

Without going into further details, we summarise that the experiments with convex solvers were not a failure; still we concluded that their use is not worthwhile, adding mostly complexity compared to the evidently satisfactory one-step Algorithm 1. It adds more parameters (of which there are already more than desirable anyway, cf. Section 2.8), and is also simply slower due to the iterative nature, despite making better use of hardware than our pixel-wise implementation.

Remark 9. *This is not to say that using existing solvers could not have merits over our ad-hoc implementations, but this would only pay off if it encompassed also at least the mask-regularisation in a more principled way, and ideally the whole path-optimisation iteration. To*

[12] H^1 distance forces $\acute{\varphi}$ to "copy" high-frequency components from φ, but those are in practice mostly undesirable/adversarial anyway, and only interfere with the more important goal of stability.



(a) Original **(b)** Naïve **(d)** $\mathcal{L}^\infty$ **(c)** Convex-optim.

Fig. 2.6: Comparison how the different monotonisation approaches behave on an input that is noisy, but otherwise already monotone. (a) noisy input (b) nonnegative-clipping of ψ-representation and rescaling (c) $\mathcal{L}^2$-optimal monotone approximation according to convex solver (d) our custom Algorithm 1.

our knowledge no available solver has the necessary combination of features to accomplish this, though.

2.6.2C *Constant speed*

Despite being in some sense the most specific amongst the path axioms of Definition 1, the constant speed property is easier to ensure than monotonicity, and there is a canonical way of doing it.

Lemma 7. *Every monotone path φ after Definition 2 has a representative ablation path $\Pi_{CS}(\varphi)$. If φ is continuous, then $\Pi_{CS}(\varphi)$ is faithful to φ in the sense that all its masks occur also in exact form in φ.*

Proof. The function $m\colon t \mapsto \overline{|\varphi(t)|}$ is at least monotonically non-decreasing, because φ is pointwise monotone and $\overline{|\cdot|}$ monotone in each of the point-values. Therefore, for each $w \in [0,1]$ the preimage $m^-(w)$ is convex. To define $\Pi_{CS}(\varphi)(w)$, we need to distinguish the three possible cases for convex subsets of $[0,1]$:

- $\mathfrak{m}^{-}(w) = \{t\}$, which is guaranteed in the continuous and strictly monotone case by the intermediate value theorem and injectivity of strictly continuous functions. In this case, let ("faithfully"!)

$$\Pi_{CS}(\varphi)(w) := \varphi(t).$$

This satisfies, by definition of the preimage,

$$\overline{|\Pi_{CS}(\varphi)(w)|} = \overline{|\varphi(t)|} = w$$

as required by the constant speed axiom.

- $\mathfrak{m}^{-}(w) = \{\}$, which can only happen if φ is discontinuous. This is the only case where φ contains no mask of suitable mass, but one can be constructed by interpolation: let

$$t_{<} := \sup\left\{t \in [0,1] : \overline{|\varphi(t)|} < w\right\} \qquad t_{>} := \inf\left\{t \in [0,1] : \overline{|\varphi(t)|} > w\right\}.$$

These exist because the boundary conditions guarantee $\overline{|\varphi(t)|} = 0$ and $\overline{|\varphi(t)|} = 1$ are covered. Let then

$$w_{<} := \overline{|\varphi(t_{<})|} \qquad\qquad w_{>} := \overline{|\varphi(t_{>})|},$$

and

$$\eta := \frac{w - w_{<}}{w_{>} - w_{<}}.$$

This allows defining

$$\Pi_{CS}(\varphi)(w) := \varphi(t_{<}) \cdot (1-\eta) + \varphi(t_{>}) \cdot \eta.$$

This satisfies, by linearity of $\overline{|\cdot|}$,

$$\begin{aligned}
\overline{|\Pi_{CS}(\varphi)(w)|} &= \overline{|\varphi(t_{<})|} \cdot (1-\eta) + \overline{|\varphi(t_{>})|} \cdot \eta \\
&= w_{<} \cdot (1-\eta) + w_{>} \cdot \eta \\
&= w_{<} \cdot \frac{w_{>} - w_{<} - w + w_{<}}{w_{>} - w_{<}} + \frac{w_{>} \cdot (w - w_{<})}{w_{>} - w_{<}} \\
&= \frac{w_{<} \cdot w_{>} - w_{<} \cdot w}{w_{>} - w_{<}} + \frac{w_{>} \cdot w - w_{>} \cdot w_{<}}{w_{>} - w_{<}} \\
&= \frac{w_{>} \cdot w - w_{<} \cdot w}{w_{>} - w_{<}} \\
&= w.
\end{aligned}$$

- $\mathfrak{m}^{-}(w) = S$ with $|S| > 1$. In this case, due to monotonicity of φ and strict monotonicity of $\overline{|\cdot|}$, φ must be constant for all $t \in S$, therefore one can arbitrarily choose any of them and again define faithfully

$$\Pi_{CS}(\varphi)(w) := \varphi(t).$$

□

In the discretised representation, the preimage queries are realised by an index search. This can be done efficiently because a monotone path is constructed from another monotone path, so that the read- and write locations can be moved in tandem (rather than having to use an e.g. binary search). Continuity is only ever approximated, so that the interpolation in performed unconditionally. Interpolation in general does again incur blending concerns (Section 1.4.3B), but this is tolerable especially since in the case of already almost-constant speed, the coefficients are mostly $\eta \approx 0$ or $\eta \approx 1$, meaning that paths that were almost binary will stay so. Furthermore, interpolation only happens between neighbouring masks in the path, which are already similar[13] and specifically in the saturated case across most of the domain *identical*, so that even aggressive interpolation only leads to intermediate values at the mask-region edges, and there they are incurred by regularisation anyways.

2.6.3 *Complete algorithm*

We proceed to show how all the elements above are used together, as an algorithm that optimises one of the score functions P from Section 2.3.

Algorithm 2 Projected Gradient Descent

$\varphi \leftarrow ((t, \mathbf{r}) \mapsto t)$ ▷ Start with linear-interpolation path
while φ is not sufficiently saturated **do**
  for t in $[0, 1]$ **do**
    $x_{\varphi,t} := (1 - \varphi(t))\, x_{\mathrm{Tg}} + \varphi(t)\, x_{\mathrm{BL}}$
    compute $P(x_{\varphi,t})$ with gradient $\mathbf{g} := \nabla P(x_{\varphi,t})$
    let $\hat{\mathbf{g}} := \mathbf{g} - \int_\Omega \mathbf{g}$ ▷ ensure $\hat{\mathbf{g}}$ does not affect mass of $\varphi(t)$
    update $\varphi(t) \leftarrow \varphi(t) - h\, \hat{\mathbf{g}}$
    (optional) apply a regularisation filter to $\varphi(t)$
  (optional) adjust learning rate h according to size of the actual step performed
  (optional) apply saturation to φ (Section 2.5.2)
  (optional) apply pinching to the paths $\varphi_\uparrow$, $\varphi_\downarrow$ (Section 2.5.3)
  for $\mathbf{r}$ in Ω **do**
    re-monotonise $t \mapsto \varphi(t, \mathbf{r})$, using Algorithm 1
  clamp $\varphi(t, \mathbf{r})$ to $[0, 1]$ everywhere
  re-parametrise φ, such that $\int_\Omega \varphi(t) = t$ for all t (using Lemma 7)

In case of $P_{\uparrow\downarrow}$ (Equation 2.16), all occurances of φ in the algorithm concern in fact the two paths $\varphi_\uparrow$ and $\varphi_\downarrow$, which are optimised independently of each other with respect to $P_\uparrow$ and $P_\downarrow$, respectively, and only interact with each other via the pinching correction.

The termination condition does not necessarily have to be "sufficient saturation", however we found this to be the most consistently applicable one. It might be more intuitive to base termination on the running path-score, however this is fraught with problems:

[13] Assuming sufficiently high resolution of the time axis, which may not always be feasible.

- A constant value for what score is good enough would not work across many different examples, because the realistically achievable score differs vastly from case to case.[14]

- Saturation-effects in the classifier can result in a (legitimately) high and stable score already early on in the optimisation, when however the path is not saturated at all yet, making for a poorly interpretable saliency result. Running the algorithm longer gives the artificial saturation time to achieve a more usable result; often this happens whilst the path score stays nearly constant.

In our experiments we generally terminated on a saturation level of 0.8, meaning

$$\int_0^1 dt \int_\Omega d\mathbf{r} \ (2 \cdot |\varphi(t, \mathbf{r}) - 0.5|) > 0.8. \tag{2.31}$$

This too is not always feasible: in some cases the classifier gradients may actively oppose the artificial saturation, meaning the algorithm gets stuck in a low-saturation state. In such cases it is most prudent to not rely on the result as a classification-explanation, since this is an indication that the issues discussed in Section 1.4.3B are at play. A change of hyperparameter may help, or a switch to an entirely different saliency method.

Alternatively, one can make the artificial saturation progressively stronger as the algorithm proceeds, and thus force a near-binary final state; however that will typically have only a low path score (since the saturation had to "fight" the classifier). Or one can "give up" and terminate the algorithm also at exceeding a preset iteration count and try to make the best use possible of the undersaturated masks. This is not recommendable for critical use cases, but it is what we did for the pointing game comparisons Section 2.8 in order to have at least *some* results for those examples, to allow taking statistics.

Many details are omitted in the high-level view above, such as the way batch processing is employed to make good use of GPU capabilities. The real implementation also contains several more processing options that we tried to tackle the various difficulties encountered in practice, but that did not have noteworthy success in their current form. See [84] for the full code[15].

2.6.4 *Performance*

There is no way to escape the fact that our method is computationally quite expensive. We found that roughly 50 iterations of the algorithm are necessary to obtain a useful result, quite often more (the exact number varies strongly between examples, even within a single dataset). Each of the iterations requires several classifier evaluations along the path (at least $\approx$ 10, better 20 or more), and though these can be batched on GPU there is for any deep neural networks a limit to how fast one evaluation can be carried out. Combined with the other processing steps, best-case wall-times for

[14] Unless insufficient regularisation gives way to adversarial paths, in which case the score is always 100% but the result unusable.

[15] https://github.com/leftaroundabout/ablation-paths-pytorch/blob/xAI-paper/experiments/ablation_paths.py#L762

obtaining an optimised path are around $\frac{1}{2}$ minute, but realistically one should expect having to wait for 10 minutes when computing it on consumer-available hardware.

The custom parts of our algorithm could possibly be sped up substantially, but this is not to be expected for the classifiers where vast amounts of effort have already been put into those to make them fast to train. Investing work into performance-optimising the other parts would therefore have diminishing returns.

2.7 Evaluation

While it is possible to get an impression of a saliency method's quality by trying it on individual examples, looking at the resulting heatmaps and pondering how reasonable they are, insights obtained this way often do not generalise well to other examples. This necessitates the use of assessment metrics that can be computed automatically, to summarise performance over a larger selection of examples. One such metric, the pixel ablation [73](Section 1.2.4C), is not usable for our method because it is almost literally part of the algorithm itself, and achieving arbitrarily high scores is possible through adversarial masks.

The main alternative we relied on is therefore the pointing game [118], whose definition is completely detached from anything the optimisation could subvert. It gives an estimate about both to what extent the method can compete with existing ones from the literature, as well as an aid for deciding which variations of the method work better or worse, in a sense that is not just single-image sporadic behaviour. All this should be weighed with the caveats from Section 1.2.4B.

2.7.1 *Baseline choice*

The choice of the baseline image x_{BL} is somewhat orthogonal to the Ablation Path method, so we did not put very much focus on its investigation. What is imperative is that x_{BL} lies in a different class from x_{Tg}; apart from that it is sensible to keep it simple and similar to x_{Tg} to avoid influences from entirely different features. An established option is the blurred baseline, specifically the convolution of x_{Tg} with a single Gaussian:

$$x_{\mathrm{BL}} = \gamma_{\sigma_{\mathrm{BLblur}}} \star x_{\mathrm{Tg}}, \tag{2.32}$$

where σ_{BLblur} is chosen to a minimum size that ensures $\arg\max\left(\mathbf{F}\left(x_{\mathrm{BL}}\right)\right) \neq \arg\max\left(\mathbf{F}\left(x_{\mathrm{Tg}}\right)\right)$, and at least 4 pixels, at most 100 pixels.

2.7.2 *Heatmap reduction*

The result of the Ablation Path method in its variations is one or multiple paths, whereas the pointing game expects a single heatmap. There are multiple ways of reducing to such a map:

2.7.2A *Averaging*

One can simply average over all the masks in a $P_{\uparrow}$-optimal path. This operation is (modulo a time renormalisation) left inverse to the pixel ablation of a saliency map

(Equation 2.3).

$$\Theta_{\text{avg}}(\varphi) := \int_0^1 \mathrm{d}t \ (\varphi(t)) . \tag{2.33}$$

This works well in some cases, but the result can be disproportionally affected by low-discriminate contrasts of masks generated far from a decision boundary, which are unstable in a similar way to plain gradient methods.

2.7.2B *Class transition*

Taking the point of view that the decision boundary is what matters, one can seek the position where the path crosses it by tracking the classifier outputs along the path.

Empirically, this gives better results than averaging (both for the pointing game and, to our eyes, ease of interpretation), but it hinges on the assumption of there being a single boundary-crossing (as in the assumption to Lemma 1). In general, there may be multiple crossings, or the classifier might have a far more gradual transition, or (in case an explanation for a class different from the prediction for x_{Tg} is sought) it might not cross a boundary at all.In our implementation, we therefore make a case distinction:

- If there exists t such that $\mathbf{F}(\varphi(t))$ is dominated by the target class, then we select the largest of these t as t_{clt}. In other words, we select the most confined mask that results in the classification of interest. Here (unlike the rest of the chapter) we consider the full multi-class output of $\mathbf{F}$, and by "dominate" we mean that the target class ranks higher than all others.

- If no such t exists, we select simply $t_{\text{clt}} = \arg\max_t (\mathbf{F}(\varphi(t)))$. This may not be the best selection strategy in all applications, but it does guarantee always getting a result that can be compared in the pointing game. In critical applications it is likely better to discard paths that do not cross a boundary, and consult a different method in such a case.

In either case the heatmap is then the single mask from the path at the selected place,

$$\Theta_{\text{clt}}(\varphi) := \varphi(t) . \tag{2.34}$$

Class transition heatmaps do not have the problem of considering many indeterminate masks far from a boundary, but the somewhat opposite problem of considering only a single heatmap, which may also be unstable (in particular if the domain boundary is not very sharp) in the sense that it is poorly defined where the boundary is but a change in location can have strong influence on the result. The two following reductors provide a compromise.

2.7.2C *Influence-weighted averaging*

A third method averages multiple heatmaps like Equation 2.33, but weighted differently. Specifically, it weighs *updates* by how strongly they affect the classification:

$$\Theta_{\text{iwa}}(\varphi) := -\int_0^1 \mathrm{d}t \left(\frac{\partial \varphi(t)}{\partial t} \cdot \frac{\partial F(\varphi(t))}{\partial t} \right). \tag{2.35}$$

This has the advantage compared to class transition that it is not based on a single mask, but takes any change between masks into account that influences the classification. Note that this can also be changes that influence the classification negatively, which can give some hints about not only the features belonging to the target class but also features belonging to *other* classes that work against the target classification; see Figure 2.7.

2.7.2D *Contrastive averaging*

For the two paths optimizing $P_{\uparrow\downarrow}$, the property of interest is that they pinch the decision boundary between them. That means that for each t, the normal direction of the boundary is approximated by $\varphi_\uparrow(t) - \varphi_\downarrow(t)$ (at least coarsely, cf. Figure 2.2). This suggests averaging between these values, i.e.

$$\Theta_{\text{cav}}(\varphi_\uparrow, \varphi_\downarrow) := \int_0^1 \mathrm{d}t \ (\varphi_\uparrow(t) - \varphi_\downarrow(t)). \tag{2.36}$$

Indeed this appears to give comparatively good, stable results in practice. Our interpretation is that on any indiscriminate parts of the path, the pinching tweak Equation 2.27 reduces $\varphi_\uparrow(t) - \varphi_\downarrow(t)$ so these parts do not contribute to the result like they would in Equation 2.33. The reason for this behaviour is that indiscriminate parts do not have a consistent F-gradient that would keep $\varphi_\uparrow(t)$ and $\varphi_\downarrow(t)$ apart during optimisation. On the other hand, stably-salient differences do keep them apart and therefore prevail in Θ_{cav}. Contrastive averaging therefore fulfills the goal of taking only classifier-affecting changes into account, but without being excessively singular like the class transition or requiring differential operators like the influence-weighted averaging.

2.7.3 *Pointing game*

With the paths reduced to single heatmaps, these can be used in saliency benchmarks just like any other. We used the benchmark included in the TorchRay repository [103], which was part of the work of Fong, Patrick, and Vedaldi [27] and used by them to evaluate several literature methods on the entire PASCAL VOC [26] test set (4952 images) and COCO [56] validation set ($\approx$ 50k images). In all cases, they run an explanation with respect to not only the top class, but each of the classes human-annotated for some object visible in the image. In the COCO set, these are often quite many. Two classifiers were used for all of this, VGG [91] and ResNet [37].[16]

[16]These models are by now quite dated. Arguably it would be more appropriate to do comparisons with newer models, but we stuck to the existing literature.

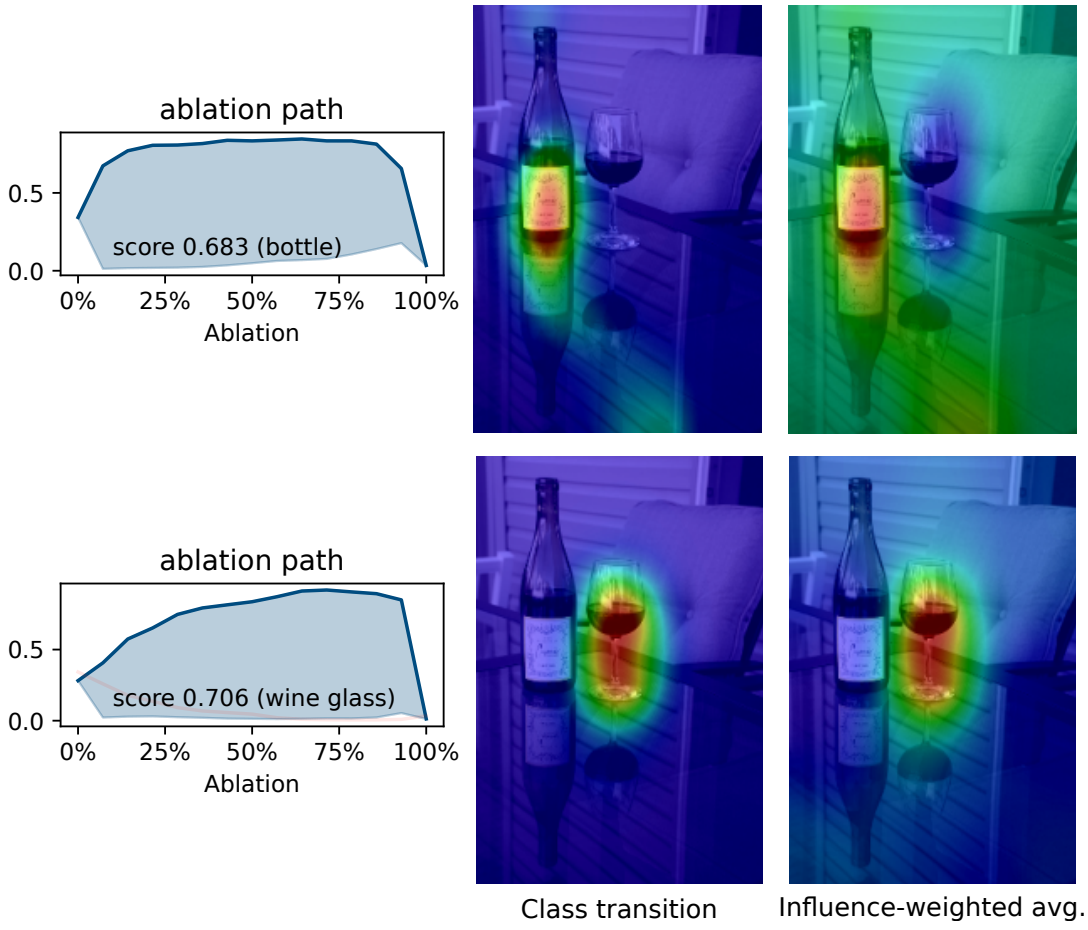


Fig. 2.7: Example how different classes within a single image (COCO) can be explained.

Method	*VOC07 Test* (All%/Diff%)	*COCO14 Val* (All%/Diff%)	Method	*VOC07 Test* (All%/Diff%)	*COCO14 Val* (All%/Diff%)
Ctr.	70.9/41.9	26.0/15.4	RISE	86.4/78.8	54.7/50.0
GCAM	90.5/80.4	57.1/49.2	GCAM	90.4/82.3	57.3/52.3
Ours	84.3/64.8	49.3/41.0	Extr	88.9/78.7	56.5/51.5

Table 2.1: *Left*: the highest-scoring results for the pointing game over 1000 images with ResNet50 classifier, for comparison with the state of the art. *Right*: excerpt from table 1 in [27] (theirs is the "Extr" method), which contains the scores of more methods from the literature for the complete datasets.
The "All/Diff" refer to a "difficult" subset of the data chosen in [118]. RISE is from [73], GCAM is Grad-CAM [87]. The "Ctr." method does not compute saliency but always points at the center, as a trivial null-score.

Due to the expensiveness of our method, we did not carry out the full benchmark, but only ran it for the subset of the first 1000 images from each of the datasets (with all the classes). Using the best parameter setup we found (see next section), Table 2.1 shows our method getting close to the state-of-the-art scores, but it does not quite reach them.

This demonstrates that the Ablation Path method can to a large degree do what existing methods can, in addition to giving considerably more information to interpret in the form of a browsable path. It is unclear what causes the gap that remains between our method and the top state of the art. Three plausible reasons are:

- We have not found the best settings / hyperparameters for the respective datasets. Though we have spent considerable effort in search (Section 2.8), it is not exhaustive.
- Our algorithms still lack features that are necessary for some examples, such as the max-convolution operation [27].
- The use of paths puts a fundamental limitation on how well 2D heatmap reductions can condense the information.

Section 2.9 discusses.

2.8 Variations / hyperparameters

As the previous sections lay out, the Ablation Path method is not so much a single method but a whole family. This can be seen as good in terms of flexibility of use, but for the unprepared user it may present rather a disadvantage, since it will not be obvious which manifestation to use. The differences between them have both subjective and objective aspects. The former include the choice of score function: these correspond to different settings for the classification process, all of which are useful in their own different way (Section 2.8.1). Similarly, some saturation/regularisation combination may be more useful for deciding between smaller or bigger features, etc..

The objective aspect is that many parameter combinations simply do not give an informative explanation at all, but only adversarial masks, unspecific over-smooth ones,

or prematurely saturated and suboptimal, etc.. Unfortunately, estimation of when one of these is happening in a given examined image is risky: what looks like a fragmented, nonsensical saturation artifact could also be a precise highlight of a classification based on biases in the dataset (which is bad from the classifier perspective, but good for the saliency method). A smooth mask on the other hand can invite the human investigator to apply confirmation bias, focusing on the object they expect the classification should be based on and happens to be within the fuzzy highlight, but is not actually relevant to the classifier's decision.

Vice versa, a fragmented binary mask can also have a disproportionate influence (essentially adversarial) on the classifier via its introduced artificial edges, wheras a fuzzy one can simply be testament to the fact that the classifier is reacting to the entire scene of the image, with no individual object being singled out as particularly critical nor sufficient when taken standalone.

Against all this confusion, our method provides some benefits through offering a whole path, compared to single-heatmap methods. This helps in the sense that the investigator can look at the sections of the path both with and without context, form hypotheses to be tested, and check small changes by moving back and forth along the path. It still leaves a significant risk of misinterpretation though.

At any rate it can be constated that a good choice of hyperparameter is important, and the choice cannot reliably be made based on only a given example. In the following, we attempt to make some statements regarding good/stable parameter choice, based on statistics about path scores and the pointing game. This can be seen critically for both the inherent reasons in Section 1.2.4B as well as quite practical that for many kinds of applications, something analogous to the pointing game is simply not available. Nevertheless we decided to rely on it here.

After all, the pointing game – in spite of all criticism – remains the most reliable independent assessment for how consistently the saliency method highlights something that has with high probability to do with what the classifier bases its decision on. Due get insights in reasonable time, we ran these experiments on only 100 images.

In Table 2.2 we show the same experiments as compared in Table 2.1 but with only 100 images and part of the used configuration shown. Note that the best results are achieved with slighty different configurations for both of the dataset; Table 2.1 shows only the better one for each dataset, labelled in Table 2.2 "Ours^{V}" for the best in VOC and "Ours^{C}" for the best in COCO.

Remark 10. *The astute reader may also notice that on only the first* 100 *images, the results are systematically worse than on the first* 1000 *or all of the datasets. This is less due to these images being more difficult, than artifact of the way the TorchRay benchmark gathers the results: specifically, it counts success rate for each class separately and averages in the end, but rates classes that are not even present in the smaller subset as* 0% *success.*

2.8.1 *Score functions*

The score functions serve somewhat separate purpuses, for example a high-scoring retaining path will show how a region of the image achieves the target classification with minimal other features from the image x_{Tg}, which can be seen as more generalised.

Method	opt.cr	ζ_{sat}	σ_{reguBlur}	postproc	*VOC07 Test* (All%/Diff%)	*COCO14 Val* (All%/Diff%)
Ctr.					71.4/36.6	26.5/11.2
GCAM					90.4/64.2	48.9/35.2
Ours$^{\text{V}}$	$\mathrm{P}_{\updownarrow}$	0	7.0px	window	80.5/46.1	46.2/31.0
Ours$^{\text{C}}$	$\mathrm{P}_{\uparrow\downarrow}$	0.8	7.0px		72.2/47.5	48.3/34.8

Table 2.2: The same experiments as in Table 2.1, but with only 100 images.

Method	opt.cr	ζ_{sat}	σ_{reguBlur}	Hm.red	*COCO14 Val* (All%/Diff%)
Ret.	$\mathrm{P}_{\uparrow}$	0.8	8.0px	Θ_{clt}	37.8/24.0
Ret.	$\mathrm{P}_{\uparrow}$	0.8	8.0px	Θ_{iwa}	38.9/25.7
Contr.	$\mathrm{P}_{\updownarrow}$	0.8	7.0px	Θ_{clt}	40.6/26.7
BndStr	$\mathrm{P}_{\uparrow\downarrow}$	0.8	7.0px	Θ_{cav}	48.3/34.8

Table 2.3: Pointing games with the different score functions and heatmap reductions.

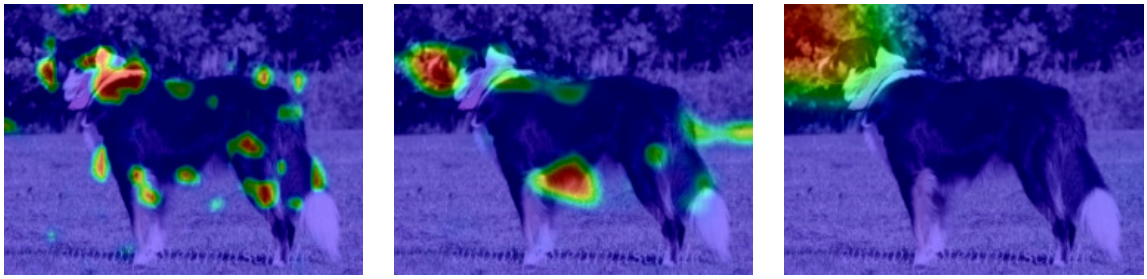
A high-scoring dissipating path on the other hand will show specifically how a region can be removed to most effectively change the classification away from the target one, which tells more about how the corresponding object is classified in the particular context of x_{Tg}. Both are useful in their own way.

All of them can also be used for the pointing game, via suitable heatmap reductions from Section 2.7.2. We observe that the boundary-straddling method gives rise to the most consistently good scores, presumably because it takes advantage of evaluating the classifier at many different points with relevant behaviour, and averages these stably. Particularly for the COCO dataset this is very effective. This dataset annotates in many images both big objects that are hard to obscure and small objects that are hard to highlight; all the other score functions have trouble with either category, but the boundary-straddling method can adapt by localising the divergence between $\varphi_{\uparrow}$ and $\varphi_{\downarrow}$ accordingly to the size of the objects. For the VOC dataset meanwhile, the single-path contrastive score can achieve slightly better score, probably because in simple images the main object does have one mask size corresponding both to object size and class transition, which is then used as the sole heatmap. All the following experiments are based on ResNet50 and 100 images from the COCO14 dataset (the more difficult dataset).

Even if the boundary-straddling method tends to fare best in the pointing game, this need not be reason to consider it the best one for practical use. The twin paths in this method make its result a bit less ergonomic to inspect, though it can certainly be also informative in ways the other varieties are not.

2.8.2 *Step size*

For the multiplier of applying the gradient to the current path state, there is at least one upper bound: it should not result in so strong violation of the path axioms that their repair leads to an entirely different state. A particularly simple consequence of



$\sigma_{\text{reguBlur}} = 0.5$ $\sigma_{\text{reguBlur}} = 2$ $\sigma_{\text{reguBlur}} = 8$

Fig. 2.8: Example of how both too little and too much regularisation can be detrimental for interpretability. Image from VOC2007 test set; saliency is class transition of a $P_{\updownarrow}$-optimal path.

the axioms is that the range is limited to $[0, 1]$. As such, there is a natural size scale for steps that work together with the projection: the change should have an $\mathcal{L}^\infty$-norm on the order of 1. We generally ensure this by normalising the step taken accordingly.

It is possible that smaller steps would in some cases be preferable, because even an axiom-conforming step can already correspond to a traversal of too much of the classifier domain to allow the gradient descent to work optimisingly. We tried smaller steps occasionally with different variations of the algorithm, but found that they only make the optimisation even slower than it is already otherwise. It can also make the artificial tweaks like saturation overpower the classifier updates, though these can always be downscaled accordingly.

2.8.3 *Regularisation*

The spatial regularisation is the hyperparameter we share with many literature methods, but surprisingly the literature does not say much about how it can be chosen. One has to assume it is generally based a lot on benchmarks like the very pointing game. Since the blurring parameter σ_{reguBlur} is not quantised in any way, it lends itself well for closer invesigation of how it interacts with other parameters and result characteristics.

Readily apparent is that both too weak and too strong regularisation is detrimental. Too weak, it will not protect against adversarial behaviour. Too strong, and it will not only (obviously) restrict how precise spatial features can be localised, but can also introduce severe biases. E.g. in Figure 2.8, the strongly regularised saliency is not only condensed to a single location, but also specifically to a corner of the image. Our interpretation is that this happens because it reduces the total variation (since 75% of the mask's gradient contributions lie outside of the image frame). And although the mask in this example still contains enough of the dog's head to keep the classification, its maximum lies misleadingly in front of its nose, outside of the dog's silhuette. This specific phenomenon – condensation of the highlighted regions near the boundaries – occurs quite often in the experiments when strong regularisition is applied. The pointing game scores reprimand this misbehaviour (Table 2.4), since the annotated objects are seldom located at the boundary.

This particular problem is not really specific to the Ablation Path method but generally regularised mask optimisation, but in individual-mask methods it can simply

Method	σ_{reguBlur}	Hm.red	*COCO14 Val* (All%/Diff%)	postproc Method	*COCO14 Val* (All%/Diff%)
$P_{\updownarrow}$	1.0px	Θ_{clt}	40.3/25.9	window	37.6/21.8
$P_{\updownarrow}$	2.0px	Θ_{clt}	38.8/26.4	window	39.6/22.9
$P_{\updownarrow}$	3.0px	Θ_{clt}	36.3/25.5	window	40.2/24.2
$P_{\updownarrow}$	5.0px	Θ_{clt}	32.0/22.1	window	40.9/24.7
$P_{\updownarrow}$	7.0px	Θ_{clt}	32.5/23.1	window	40.6/26.7
$P_{\updownarrow}$	10.0px	Θ_{clt}	24.0/16.8	window	37.5/23.6

Table 2.4: Pointing games with different regularization strengths and postprocessing.

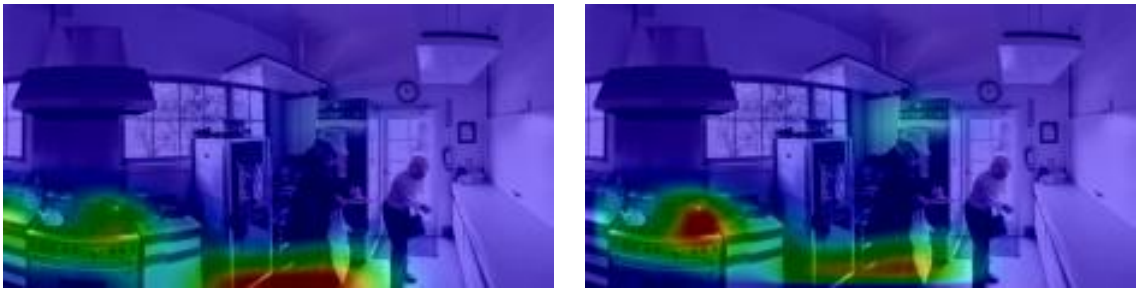


Fig. 2.9: Two heatmap extractions (class transition) of the same ablation path; the objective is to explain "oven". *Left*: the optimised mask contains a strong regularisation artifact in form of the highlighted band at the bottom. *Right*: applying the window postprocessing of Equation 2.37 (and normalising) shifts the maximum to the originally less prominent region considered correct by the pointing game.

be avoided (and is! [27]) by constraining the masks to vanish at the boundaries.Perhaps the best argument justify this constraint is to see it as prior knowledge that important objects seldom occur at the boundaries; it has two problems of its own though:

- It introduces another bias. The extremal perturbation method cannot detect at all important features which really are at the boundaries.
- It is irreconcilable with the ablation path axioms, specifically the boundary conditions.

An even more crude way of achieving a similar effect is to remove the boundary parts of a reduced heatmap as a postprocessing step (Figure 2.9), by multiplying with a window function

$$\Theta^{\text{window}}(\mathbf{r}=(x,y)) := \Theta(\mathbf{r}) \cdot \sqrt{\sin(\pi \cdot x) \cdot \sin(\pi \cdot y)} \tag{2.37}$$

– which we find does indeed improve the pointing-game score in strongly regularised cases, see Table 2.4 and Figure 2.10.

The boundary-straddling method does not have the boundary problem, because it only takes differences between $\varphi_{\uparrow}$ and $\varphi_{\downarrow}$ into account, and both of these paths are attracted to the boundary by regularisation in the same way.

Even if the boundary-condensation phenomenon can be circumvented, this does not avoid the need to select appropriate regularisation strengths. The pointing game

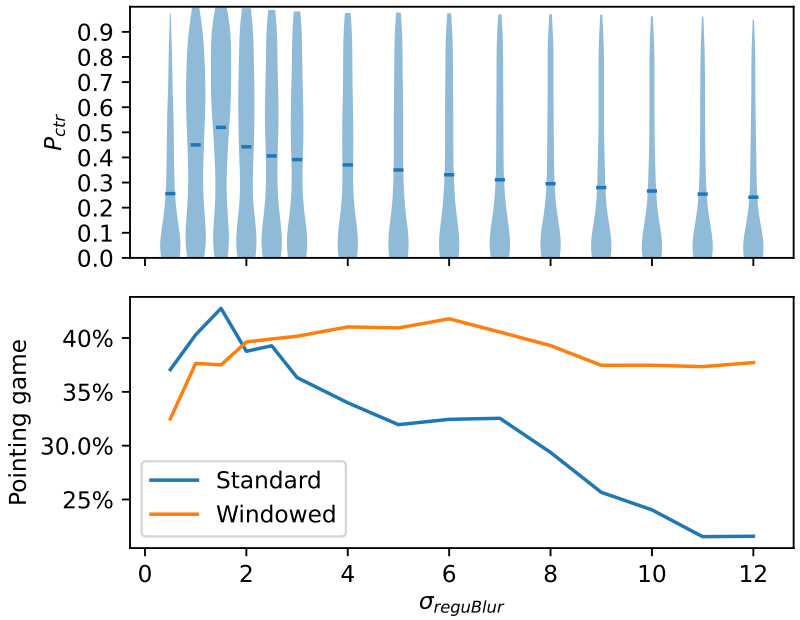


Fig. 2.10: Dependence on the size of the regularization filter, for both the distribution of boundary-straddling ablation-path scores and the ("All"-) pointing game score. Paths optimised for the contrastive score; pointing game evaluated on both standard class-transition masks (Equation 2.34), and with or without a boundary-suppressing window (Equation 2.37) applied.

shows that there is a certain sweet spot, but without it, with less well understood data, it is unclear how to find it. The ablation path method give one additional hint that may be at least somewhat helpful in this situation: the path-score after optimisation. This should for good saliency operation be expected to alway come out somewhere in the middle of the range $[0, 1]$, because the objects of interest have some finite size in the image and after ablating into them the classification should drop even for an optimal retaining path. Only if due to underregularisation the path has developed adversarial masks is it to be expected that scores of almost perfectly 1 are achieved, and Figure 2.10 shows this happening quite often when $\sigma_{\text{reguBlur}} < 4$, in form of the violin plots reaching a flat top with $P_{\updownarrow} = 1$. This a clear indication that regularisation $\sigma_{\text{reguBlur}} > 4$ is necessary, and indeed the pointing game confirms this with its peak at $\sigma_{\text{reguBlur}} = 6$. The conclusion is that one should choose regularisation slightly stronger than required to avoid the indication for adversarial behaviour, but not much stronger since that would unnecessarily reduce precision and introduce other disadvantageous effects of smoothing. Statistics about the path-score for this purpose can be obtained without need for the pointing game or similar independent validation.

Note that this heuristic is more difficult to use with the boundary-straddling method: it seldom attains scores close to 1 even when regularisation is very weak (Figure 2.11), because it has additionally the boundary-pinching mechanism (Equation 2.28) working against this.Still, also in this case at least a drop in average score at $\sigma_{\text{reguBlur}} = 4$ is visible, followed by a peak in pointing-game score at $\sigma_{\text{reguBlur}} = 6$.

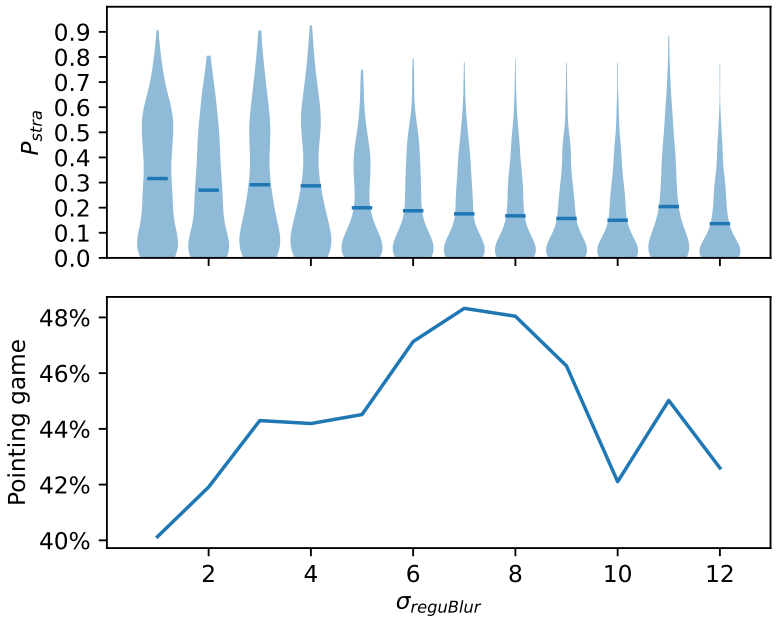


Fig. 2.11: Like Figure 2.10, but evaluated with contrastive averaging (Equation 2.36).

Method	opt.cr	ζ_{sat}	*COCO14 Val* (All%/Diff%)	postproc Method	*COCO14 Val* (All%/Diff%)
Contr.	$P_{\updownarrow}$	0	39.9/28.4	window	45.5/30.1
Contr.	$P_{\updownarrow}$	0.8	29.4/21.1	window	39.3/27.9
BndStr.	$P_{\uparrow\downarrow}$	0	46.8/32.8	window	48.0/32.5
BndStr.	$P_{\uparrow\downarrow}$	0.8	48.0/35.7	window	46.4/30.8

Table 2.5: Pointing games with different saturation in different contexts.

2.8.4 *Saturation*

The artificial saturation of Section 2.5.2 was introduced mainly to avoid the blending problematic of Section 1.4.3B, as well as to get a clearer, less nebulous and therefore easier to interpret ablation path. Unfortunately, saturation also can easily have detrimental side effects.In Table 2.5 we see that it can quite dramatically lower the score, as for the contrastive path. In this specific example this is largely avoided with windowing, indicating that the problem is again the boundary behaviour (which saturation generally exacerbates), but other times there is no such explanation, like for the boundary-straddling where windowing actually reduces the score but increases it when no saturation is applied.

We have no satisfying justification for why such behaviour happens, but what is clear is that saturation is counterproductive when it overpowers the classifier's gradient and becomes the dominating contribution to the optimisation procedure. This must be avoided, but unfortunately is hard to control since it depends on how the gradient updates interact with both the saturation and the axiom projections. Figure 2.12 seems to show such effects, but it is uncertain what exactly is happening there.

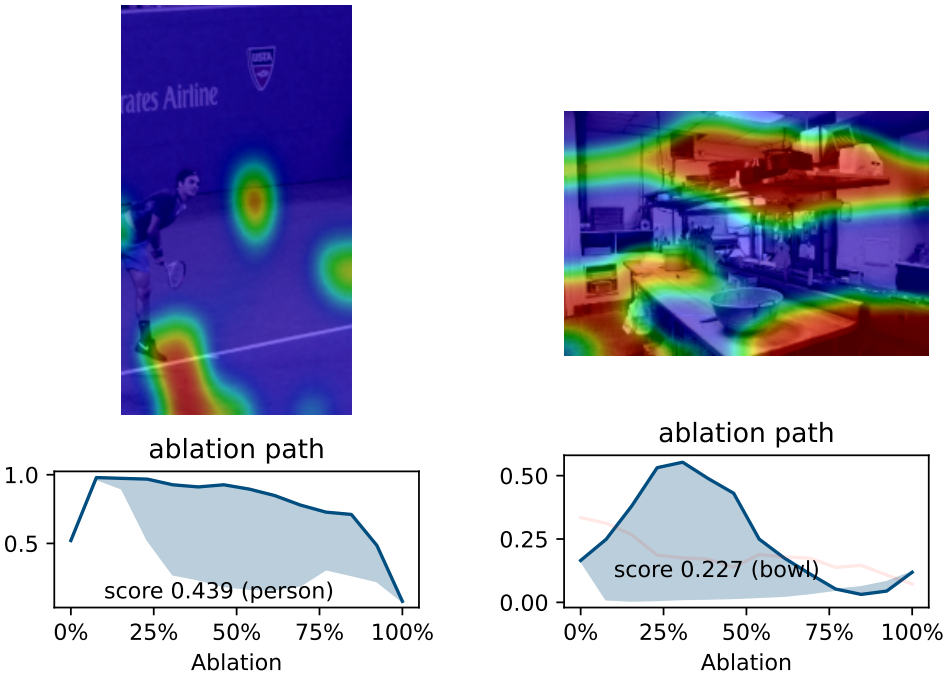


Fig. 2.12: Examples of plain nonsensical ablation path results. These may be a result of overzealous saturation, or some other problem of parts of the algorithm getting in each other's way.

2.8.5 *Others*

The parameter space is too large to investigate every possible combination, especially if quantitative assessment is desired, given the high computational cost for a pointing game run. Some more variations of the method deserve at least brief mention, though we have not (as yet) been able to obtain satisfying results with them.

2.8.5A *Blur pyramids*

All the results shown were obtained with a blurred baseline Equation 2.32. Many other kinds of baseline would be possible, but if one is chosen then there are good arguments for using a Gaussian filtered one. The fixed size however is not necessarily the best. A way to avoid the size choice is already used by Fong, Patrick, and Vedaldi [27]: they use an interpolation operator that chooses *locally* between different-filtered baselines. This can also be used with the ablation path method, which we implemented, but it did not improve results over the linear interpolation to a single select baseline. It is quite possible that this could be overcome by adjusting other parts of our algorithm.

This *blur-pyramid* is essentially a multiscale method. chapter 4 develops another such method, one that embraces the concept more consequently.

2.8.5B *Stochastic baselines*

In Section 2.4.3C we also touched on the possibility of using not one artificial, neutral baseline, but instead many different ones from a real dataset. We tried this, but failed to achieve any consistent convergence. That is not entirely surprising since these

baselines have strong classifier responses of their own, which push the optimisation in very different directions. Other applications of stochastic gradient descent manage to cope with this though, through various minibatch, momentum etc. techniques. The main hurdle appears to be that our algorithm in its current form is quite stiff, with its projections interspersed between the update steps. Changing this would require fundamentally restructuring the algorithm, and it is not clear whether it would work at all then.

2.8.5C *Dynamic parameters*

In principle the parameters do not need to stay fixed during the optimisation. For some of them, the best values towards the end of the optimisation are not necessarily the best at the start. In particular the artificial saturation can be detrimental if it is applied too strongly early on in the optimisation, whereas later on high values can be necessary to achieve interpretably saturated masks. Both can be facilitated by progressively ramping up ζ_{sat} during the optimisation.

2.9 Discussion

We have presented a novel kind of saliency method and demonstrated its feasibility. In many examples, the results are convincing, and the pointing game score demonstrates that these are not mere flukes (though several literature method score slightly better). The form of the results as paths is informative and friendly for human inspection with interactive tools like the HoloViews/Panel widgets we developed for the purpose[17].

Nevertheless, the method leaves many wishes. Even with the (presumed-) best parameter choices, many inputs lead to cryptic ablation paths, and with suboptimal parameter choices the results are prone to suffer undesired effects of the implementation. Because the algorithm combines many interacting components, parameter choice and debugging are difficult, if not outright hopeless for data lacking the possibility of checks like the pointing game. This problem is not exactly new to our method, with others also having more hyperparameters than desirable, but the complexity of ours certainly exacerbates it.

On the plus side, the path also offers some unique diagnostic opportunities like discussed in Section 2.8.3. Building on that together with more principled optimisation techniques could still improve the method a lot. Perhaps this would require writing the algorithm from scratch, perhaps incremental improvements would be sufficient.

But after already having spent great efforts on such improvements, our verdict is that these improvements mostly suppress the symptoms of a more general issue that is not so much about ablation paths but about perturbation-based saliency in general: that the pixelwise modification of images is too crude a tool to be reliably used for probing image classifiers. chapter 3 takes a step back to survey possible approach directions for alternative kinds of intervention, one of which we formalise and implement in chapter 4.

Another topic of interest, which we did not have time to cover, is the use of Ablation

[17] https://github.com/leftaroundabout/ablation-paths-pytorch/blob/xAI-paper/experiments/ablation_saliency_plotting.py#L548

Paths for classifiers of entirely different kinds of data. Nothing in the method's formalisation is specific to the image application. It is wide open in which kind of data the method would work as good as on images, or worse, or better (because the conditions are more concilatory than e.g. the rather conflicting fulfilment of mask saturation and image regularity).

Part III

FROM POSITIONS TO SCALES

CHAPTER 3

FEATURES, WHAT ARE THEY?

So far in this thesis, two notions of what features can be have been considered:

- The basis expansion already used for the inputs. This has upfront the advantage that the features themselves are completely understandable, so the explainability task is reduced to one of attribution. This attribution is however fraught with difficulties, as the previous chapters have touched upon. More discussion in Section 3.1.
- Abstract representations of some machine-learned nature. These are in the opposite position: they can empirically be attributed more easily, but even a perfect attribution provides not a very satisfying explanation. More in Section 3.3.2.

This chapter tries to pin down the difficulties with the former approach, and why the latter approach does not suffer from those. Then, chapter 4 proposes an alternative feature expansion that (specifically for images) attempts to avoid the attribution difficulties of a purely spatial / pixel expansion similarly to how abstract representations can, while still being completely transparent in the construction of the features.

In these two chapters, $\mathcal{I}$ can always be taken to be a space of images/photos. Unlike in chapter 2 (which used image classification for all examples but did not fundamentally rely on that choice), the feature expansions discussed here would hardly be applicable to other kinds of data, at least not without major changes.

This chapter explores mostly how features can be designed that allow the idea of ablation (or pertubation in [28]) to be carried out such that removing features only removes information. Before getting into solutions, it will be necessary to ponder what that should even mean. But before even that, we set aside both AI explainability and mathematical abstraction a bit and discuss more about the concrete manifestation of data, how it relates to the real world.

3.1 Some intuitive / naïve approaches

3.1.1 *Pixels of light*

When all inputs come in form of multidimensional arrays, it seems the obvious thing to formulate as much as possible of a method's implementation in terms of the entries of these arrays. For some operations this is completely uncontroversial; in particular linear combinations of elements of a vector space can be expressed with element-wise

scalar-multiplications and additions and the results are covariant with respect to any possible basis expansion. For more complicated operations this does not hold though, and it is worth pondering a while why e.g. masking on pixels works at all in a usable way.

Photographic images are a representation of a physical phenomenon, concretely the irradiance upon the sensor of a camera. Ignore for the moment the fact that the sensor is already divided into pixels; it could after all also be a frame of analogue film instead. A physicist might abstract this irradiance as a scalar field, i.e. as a continuous function; ignoring colour,

$$\begin{aligned} E\colon\ & [0,1]^2 \to \mathbb{R}^+ \\ E \in\ & \mathcal{C}^0\left([0,1]^2\right). \end{aligned}$$

Remark 11. *When getting to sufficient small scale, even analogue physical manifestations are also subject to effects that could be described as discretisation – at the very least, when getting down to the atomic scale and/or when quantum effects enter the picture. Specifically analogue film has a finite average grain size, entailing that the continuous description breaks down even earlier. Nevertheless, this sort of discretisation is of a rather different nature compared to the pixels in a digital image, and its characteristic scale still much smaller than the features in the image.*

The domain does not need to be a rectangle. The following treats it as a general compact Lipschitz domain Ω.

The continuous functions form a vector space, and scalar-valued ones furthermore a ring, with addition and multiplication defined pointwise

$$\begin{aligned} +, \odot\colon \mathcal{C}^0(\Omega) \times \mathcal{C}^0(\Omega) \to\ & \mathcal{C}^0(\Omega) \\ (\vartheta_0 + \vartheta_1)(\mathbf{r}) &= \vartheta_0(\mathbf{r}) + \vartheta_1(\mathbf{r}) \\ (\vartheta_0 \odot \vartheta_1)(\mathbf{r}) &= \vartheta_0(\mathbf{r}) \cdot \vartheta_1(\mathbf{r}). \end{aligned}$$

Vector-valued functions are a module over this ring, again with pointwise operations

$$\begin{aligned} +\colon \mathcal{C}^0(\Omega, V) \times \mathcal{C}^0(\Omega, V) \to\ & \mathcal{C}^0(\Omega, V) \\ (x_0 + x_1)(\mathbf{r}) &= x_0(\mathbf{r}) + x_1(\mathbf{r}) \\ \circledast\colon \mathcal{C}^0(\Omega) \times \mathcal{C}^0(\Omega, V) \to\ & \mathcal{C}^0(\Omega, V) \\ (\vartheta \circledast x)(\mathbf{r}) &= \vartheta(\mathbf{r}) \cdot x(\mathbf{r}). \end{aligned}$$

Treating the RGB colour space simply as $V = \mathbb{R}^3$, this is already sufficient to formulate masks and ablation paths for image classification directly on these continuous functions, without any mention of pixels. The correspondence with the pixel case only stops holding true when it comes to gradients, because unlike in the finite-dimensional case the dual space of $\mathcal{C}^0(\Omega)$ is not isomorphic to the space itself (though it can be extended to a Hilbert space which is again self-isomorphic).

The above view lends some legitimacy to the use of the pixel basis, as more than just an arbitrarily-chosen basis of a vector space. In particular, it is possible to think of

the features as being conceptually patches of continuous images, and merely use pixels as an approximate representation of these, although there are some technical subtleties with this (Section 3.2) which are often glossed over.

3.1.2 *3D scenes*

However, it is worth questioning whether light intensity at the sensor is actually what the image classification is concerned with. The objects the class labels refer to are, after all, not physically located on the camera sensor, but rather in the three-dimensional space before the camera. And what arrives at the sensor is affected not only by the objects themselves but also by lighting conditions and possibly obstructions.[1]

An ideal feature basis for a saliency method could be envisioned as starting with full inverse modelling to infer from the input photo the 3D scene it was taken from, with discrete solid objects in it that could be reordered or removed at will and a new image rendered from this. 3D scene reconstruction is indeed possible in some applications. This thesis does not directly deal with it, although this is one of the goals of the Cryo-EM techniques (chapter 6) to which it contributes – but these are very much application specific.

Reconstructing a scene from a single image is still an open problem, and in general quite ill-conditioned because depth information is simply not available in the 2D projection; it can at best be deduced from prior knowledge about relative sizes, aided by shading clues. In addition, parts of the scene that are obscured by an object of interest might need to be guessed if they would become visible with that object removed. This is essentially an inpainting problem.

Humans are capable of this (though by no means infallible), and deep learning has made strong advances in the field too. But even if these techniques worked with good reliability for the inputs of interest, they would still be wanting of explanation at least as much as the basic image classifiers that this thesis attempts to explain. Inpainting, while by itself a quite well-explored problem with capable solutions [115], adds another issue since it is literally designed to add information not present in the original image. Artificial information is already a problem in the sense of adversarial attacks as discussed before, and would probably become even harder do keep in check in this approach, though it would likely help if this was integrated in the 3D reconstruction (also an active research topic [22]), and correspondingly constrained in what it could do spatially.

Nevertheless, this is a promising direction for future research. Even if such a method would involve deep-learned 3D reconstruction, the use of an explicit and physically motivated feature space would add a large amount of understanding and credence compared to completely abstract representations learned by a black box (Section 3.3.2), in particular if the reverse direction is a rendering engine based on well-interpretable techniques like raytracing. This would still leave several technical hurdles to be overcome though. For one, such 3D rendering remains computationally expensive and not readily combinable with an optimisation strategy that requires differentiability.

[1] In fact, the masking away of features could be interpreted as literally putting a black object between the camera and part of the image. This raises the delicate point that it is now in a quite rigorous sense the obstruction that stands in the foreground, rather than any of the objects in the original image.

3.1.3 *Edges and shapes*

A similar but less ambitious approach could be to separate the image in 2D into discrete objects. Again, this is well doable in an interpretable way for specific kinds of images but far more problematic in the general or photographic case.

The simplest possible way this could be tackled is edge-detection based segmentation. This works easily for clean 2D graphics, where regions can readily be found as "watersheds" of a colour-gradient magnitude, but this has considerable problems in the presence of noise. Ways to address these have been proposed [86], with successes even for large photographic images, but ultimately there still remains the necessity of hard, arbitrary cutoffs, which mean small changes in parameters as well as lighting and/or noise in the image can lead to topological changes in the decomposition. That would entail a completely different feature basis and correspondingly explanation.

There also is still the question of, if the features are image segments, what removing them should mean. This is perhaps even more problematic if they have exactly confined contours, because then any means of removing the interior would still leave those exact contours, albeit with different contrast. But a classifier would plausibly use the very shape of the contours to a strong degree for its decision. Replacing a rabbit with a rabbit-shaped hole seems like a bad way of making an image non-rabbitlike. The only way to avoid this effect when removing hard-confined regions seems to be inpainting [107], with the problems mentioned in the previous section.

Perhaps more promising use of edges could be to treat them as features themselves.[2] This has indeed been a common tenet in the field of mathematical morphology. However, the edges alone do not contain sufficient information to reconstruct a full image, as would be necessary for explaining an image classifier along the lines of Section 1.2.3/chapter 2. There are ways to essentially associate pixels to edges [79] which might allow this.

None of this helps with the problem that pure 2D analysis is oblivious to much of the inherent structure of photographic images. It might be argued that this is a fundamental limitation and that anything short of 3D reconstruction falls foul of 3D phenomena. However at least some aspects of the original 3D scene composition *can* readily be described purely in 2D terms.

- Translation of objects in 3D manifest as 2D translations and/or scaling of their size in the projection.[3]
- Lighting changes map to local changes in lightness in the photo. Typically (though not always), these changes have less spatial varation than the variations corresponding to actually material/paint colour of physical objects in the scene.

Both of these facts can be seen as motivation for the SIFT method [59], whose use for saliency purposes is established in chapter 4. But before delving into that specific method, it is worth to visit some of the theoretical underpinnings of both SIFT and

[2]Edge location can even be used as a representation for real-world image classification, at least for specific applications [77], though this does not seem to have reached noteworthy success for general photo datasets [82] (possibly for similar reasons as those mentioned above).

[3]Here we disregard translations which result in one object obscuring another one (or not-anymore obscuring it).

other methods, including the ones presented in chapter 5 and chapter 7.

Even if they are not used as tweakable features themselves, edges (specifically, their prevalence) can still be considered as a proxy for complexity of an image. It has been suggested to use this as the basis of a "hallucination score" [47], which essentially attempts to measure how adversarial a perturbed input is. A good interventional saliency method should not generate artificial edges, and thus have a low hallucination score.

3.2 Signal theory

A term that has been mentioned multiple times so far but not been properly defined is *information*. This is nowadays an intuitive concept due do the ubiquitousness of digital files, but measuring information in bits or megabytes assumes that the data is already represented in a discrete form. But while this given in the sense that all the computation happens in the digital realm, the previous section has emphasized that especially for explainability concerns it is helpful to look back at the continuous signal form that is closer to the nature of the physical origin of the data samples.

Remark 12. *This section uses the term "signal" as a shorthand for data from any function-space like* $\mathcal{C}^0(\mathbb{R})$ *or* $\mathrm{H}^1\left([0,1]^2\right)$*, i.e. not just signals in the sense of continuous time series but also scalar- and vector fields. We do assume the domain to be a cartesian product of intervals though.*

Naïvely, such a signal would seem to contain an infinite amount of data: a general function needs to be evaluated on all points of its domain to be reproduced, and if that domain is a continuum like the real line or a square those are uncountably many. The function being continuous allows reducing this to countably many, but still not a finite number (even with stronger regularity, like multiple times differentiable). As such, it is somewhat remarkable that signals can be represented digitally at all, but evidently this is possible and has been done for a long time. A pragmatic reason is that an approximate reconstruction is sufficient, however one needs to be specific about what sense this approximation is to be understood in.

A basic framework in which this can be discussed is as follows: let $\mathcal{I}^{\mathrm{C}}$ the space of signals and $\mathcal{I}^{\mathrm{D}}$ a finite-dimensional space intended to contain the discretised form of the signals. Then we consider the pair of operators

$$\begin{aligned} \mathfrak{d}&\colon \mathcal{I}^{\mathrm{C}} \to \mathcal{I}^{\mathrm{D}} \\ \mathfrak{c}&\colon \mathcal{I}^{\mathrm{D}} \to \mathcal{I}^{\mathrm{C}}. \end{aligned} \tag{3.1}$$

One would generally require that $\mathfrak{d}$ is left inverse to $\mathfrak{c}$, but for it to be also right inverse is more elusive, and not surprisingly because this would immediately entail that both spaces are isomorphic.

This is however indeed the case for *bandlimited* signals, i.e. those signals whose Fourier transform is compactly supported. Such signals can be discretised by sampling on equal-spaced points, and then be reconstructed exactly by the Whittaker interpolation formula [109] (convolution with a sinc kernel). This result, also known as Shannon-Nyquist sampling theorem, is often considered the basis of all digital signal processing. The superficial paradox of isomorphy between a continuous and a discrete

space is perhaps not the most remarkable consequence of it – after all, it is only a particular subspace of the function space on which this holds, and one could always choose explicitly the image of an interpolation operator as the subspace.

What *is* remarkable is that this subspace has particularly maths- and physics-friendly properties. In particular, a large class of wideband signals can be projected to an approximately bandlimited form by purely analogue filtering, which is explicitly done in analogue-to-digital converters. For photos, this is less obvious, but indeed the sharpness limitations of a camera's optics also provide at least some filtering, though not always enough to prevent the aliasing / Moiré effects that can arise when attempting to sample a signal that is not in the bandlimited subspace.

The physics-compatibility turns Shannon-Nyquist into an obvious choice, or even *the* canonical way of sampling signals. This, in combination with quantization of the samples to digital numbers, is called *pulse-code modulation* (PCM) and gives rise to one way of measuring the information capacity of a signal. Is is made precise by the Shannon-Hartley theorem [88]:

$$C = B \cdot \log_2 (1 + \mathrm{SNR}) \tag{3.2}$$

i.e. the achievable bitrate is proportional to the highest frequency[4] and the number of bits needed to enumerate every possible instantaneous signal-level, modulo changes within the expected amplitude of noise fluctuations. This formula is however often misunderstood: the original motivation is concerned with how much information can be *transmitted* via a signal over an analogue, Gaussian-noisy channel.

This does not mean that all such signals actually contain this amount of information: they contain typically much less, as witnessed by the fact that compression algorithms such as JPEG [100] can routinely reduce storage requirements by more than an order of magnitude. It is sometimes argued that the lossy nature of such algorithms disqualifies them as a way of measuring information, but this is dubious since Shannon-Nyquist – despite its aforementioned advantages – also is lossy, if the required bandlimiting is taken into account. Even lossless algorithms can achieve a substantial reduction in the size of typical photo files (though these, for example PNG [1], generally perform much better on graphics like plots).

One way this could be interpreted is that the local signal-to-noise ration is lower than the global one – which can sometimes be quite obvious, for example in audio files that contain only occasionally loud noises (e.g. timpani hits in an orchestral recording) but over most of the time far lower levels, and are thus in a spatial sense sparse. A purely spectral consideration of the signal would not be able to take that into account, since even (or particularly!) highly localised peaks have a broadband frequency spectrum. Vice versa, the purely-spatial PCM representation disregards any sparsity in frequency space, which is also highly common in both audio and image data.

Successful compression algorithms take both kinds of sparsity into account, which can be accomplished either by the use of constant-size windows as in JPEG, or (often more effective) by multiscale techniques (Section 3.4) which adapt to the fact that different spatial locality typically occurs in different frequency ranges.

[4] Strictly speaking, the bandwidth B is the difference between the highest and lowest frequency, but for signals like images the spectrum reaches down to the constant level, i.e. zero frequency.

A consequence of the non-optimality of PCM is that a change performed in this representation – even one that seemingly only removes information (like replacing a pixel of the target image with this pixel from a neutral baseline) – can easily *add* information instead. That is in some cases obvious enough (if the thus changed pixel forms a strong contrast with its surrounding), in others subtle but just as consequential (low-amplitude adversarial examples).

It is debatable whether information is the best point of view from which to capture the possibility of adversarial masks versus true saliency. It is in a profound sense impossible to prove that a masking-change does not add information, because one can never be sure that the representation one is working on was optimal, information wise. If x (in PCM-digital form) is considered as a generic digital string, finding the minimal representation would amount to computing its Kolmogorov complexity [49], but this is known to be uncomputable.

It should also be remembered that every practical notion of information will be relative to some context. The classifier may respond excessively to some particular kinds of input changes, but it may also ignore many changes even if they add lots of information. The extreme example is adding literal white noise. This has a *high* information content in the Shannon sense, but might reasonably be filtered out in the early layers of a classifier, either because it was explicitly designed this way or because denoising turned to be a useful instrumental goal during training.[5] Confer also Remark 4.

Nevertheless, it does match experimental evidence that low-information masks fare better in the adversarial aspect of the saliency problem: in Meaningful Pertubation [28] and Ablation Paths [85], information is kept low through regularisation; in RISE [73] through the lower-resolution random masks that are only then upsampled; and in Occlusion Explanation [19] through the recursive refinement algorithm with only minimal choice at each stage. In all of these cases, the means of "compression" is by itself interpretable, which is the other aspect that makes low information desirable specifically for a saliency method.

Such ad-hoc restrictions on the possible masks lead to hyperparameter tradeoffs though, to which metrics like the pointing game can provide only unsatisfactory guidance (Section 2.8). Even the best possible parameter choices might be adequate only for some inputs. For others, the regularisation might completely miss how information is present in that particular instance. For example, in a photo picturing several people gathered in a small spot of the scene, none of the coarse-oversampled random-sampled masks in a RISE shootout might sufficiently separate them, but simultaneaously they could still manage to add artificial information through inpainting into the previously featureless sky.

3.3 Learned representations

Some readers may be surprised that the previous section did not mention the term "entropy". This was a deliberate choice, to make the point that information can usefully be

[5] This would be analogous to how the denoising in chapter 7 is a preprocessing step to the Cryo-EM reconstruction problem (chapter 6).

discussed without any reference to probability distributions. Probability distributions are, however, the standard approach to the subject and deserve discussion as well. In fact it could be argued that all the discussion about compression algorithms hinges on some suitable distribution of the signals that could possibly be encountered.

In that approach, information of each sample x from a distribution Ψ is equated with entropy of the distribution:

$$H(\Psi) = -\mathop{\mathbb{E}}_{x\leftarrow\Psi}\left[\log_2 \mathop{\mathbb{P}}_{\tilde{x}\leftarrow\Psi}\left[\tilde{x} = x\right]\right]. \tag{3.3}$$

For some quantities, one can come up with reasonable prior distributions, often of some Gaussian form. For x as images from photo datasets, this is however all but hopeless. *Estimating* the distribution from the dataset can be feasible to some extent, though it requires considerable care.

3.3.1 *Simple statistics*

In low dimensions, approximating the distribution by a histogram[6] can be appropriate. This works only if sufficiently many samples are available for each of the bins, which in practice necessitates quantization.

Without sufficient binning, the estimated distribution would be a sparse one, assigning equal probability to the exact samples encountered in the data set and zero to anything else. Designing a coding based on that distribution would amount to assigning each sample from the dataset a discrete number. This is certainly very "efficient" (only a few bytes), but it completely fails to encode any new inputs that were not already in the dataset, though such inputs are the ones that actually matter.

Moreover, the discrete numbers do not meaningfully split into features to which the classification of this example could be attributed, even for training-set inputs. Each input would consist of only one "feature". This problem would also persist if a sufficient discretisation could be found for histogram bins that would allow for frequency statistics, though that is largely hopeless anyway when starting out in a space with as high dimensionality as pixel images: rasters defined on a high-dimensional space have exponentially many bins, so even a vast dataset would not be representative.

It is not entirely clear what properties of a distribution would be required to facilitate a split into features, but a sufficient condition is if the probability is separable, i.e. if there are spaces $\left(\hat{\mathcal{I}}_j\right)_j$ such that $\mathcal{I} = \bigoplus_j \hat{\mathcal{I}}_j$ and a distribution $\hat{\Psi}_j$ on each $\hat{\mathcal{I}}_j$ such that

$$\mathop{\mathbb{P}}_{\tilde{x}\leftarrow\Psi}\left[\tilde{x} = \bigoplus_j \hat{x}_j\right] = \prod_j \mathop{\mathbb{P}}_{\hat{\tilde{x}}\leftarrow\hat{\Psi}_j}\left[\hat{\tilde{x}} = \hat{x}_j\right].$$

This is another way of saying that the $\hat{x}_j$ are independent of each other. In that case the entopy separates as

$$H(\Psi) = \sum_j H(\Psi_j), \tag{3.4}$$

[6]Most of what is said in this section about histograms would also apply for similar tools such as Kernel Density Estimation, which is in several ways more usable but not quite as easy to discuss.

where each summand can be understood as a "chunk of information" and each chunk in the representation can be modified independently.

The lower-dimensional $\hat{\mathcal{I}}_j$ also make it more feasible to estimate the individual distributions. The extreme case, for images, would be that all pixels are completely uncorrelated; in that case, a simple 1D histogram could be obtained for each pixel. But that is only really given for pure noise images, or (statistically equivalent) images which are being used as an optimal transmission channel for unknown data. Then we would be back to allowing individual tweaking of pixels. This provides another way of looking at adversarial examples: if the dataset has some correlation built in that the feature basis does not take into account, tweaking the features independently can lead to perturbed inputs outside of the distribution.[7] For such inputs, the classifier behaviour is effectively indeterminate as they would not have occurred during training.

Ensuring all correlations are preserved in feature-tweaking would prevent this. But even relatively large regions of an image would still have correlations between them, though they would be less obvious and hardly possible to pin down reliably, certainly not with traditional statistical tools.

Points to take away here are

- Even a feature-basis where the features are somewhat correlated can be useful for explainability. The smooth masks used in the literature clearly fall in that category, but this does not necessarily incur adversarial-style non-explanations – so long as the classifier itself does not respond too erratically to these unnatural inputs.

- The distribution approach to information makes it clear that a linear decomposition is not really at the heart of the concept. Direct sums are merely a particularly simple decomposition which is highly convenient (Equation 3.4), but should, apart from the simplest applications, not be expected to be ideal as an encoding (information-wise) or as the features of a saliency method.

To the latter point it should also be remarked that even a direct-sum decomposition does not have to be of a spatial nature at all. The obvious counterexample are spectral decompositions that were already discussed in Section 3.2 but are for different reasons hardly suitable for explanation; these reasons and possible compromises are the subject of Section 3.4.

3.3.2 *Autoencoding*

Estimating the distribution Ψ from which a dataset has been sampled is by no means only of interest for explainability purposes. On the contrary, this could be seen as the underlying task behind all machine learning. Particulary unsupervised learning can do litte else but find a representation that allows fulfilling tasks like generation of new values as if they were sampled from Ψ. If the task is literally synthesizing such samples, it is called a generative model. This already suggests a parallel with the intervention approach to saliency. Generating unrelated data points is not enough:

[7] For distributions like Gaussians, there is no hard cutoff what is "outside", but points sufficiently far from the mean do become so unlikely that they may as well be considered impossible.

even for a method like RISE [73] one needs for each random sample also the information how it is similar and/or different from the target image. A fully black-box generative model would not provide such information.

Most such models are anyways not structured as random generators per se though, but decouple at least the nondeterminism injection from the representation aspects. Furthermore, they are usually equipped to take also inputs x, if only because these are used during training. The general architecture in which this is most evident are *autoencoders*: they are built from a pair of functions

$$\begin{aligned} \mathsf{E}_\phi &\colon \mathcal{I} \to \Lambda \\ \mathsf{D}_\theta &\colon \Lambda \to \mathcal{I} \end{aligned} \tag{3.5}$$

where Λ is a *latent space* of encodings without prescribed meaning; one would normally attempt for this to be as low-dimensional as possible. Both the encoder E and the decoder D are parameterised in some suitable way, and these parameters ϕ and θ are what is trained.

What concrete function architectures are used for E and D is subject to choice. A particulary simple option are linear functions, which would mean the synthetic images are a simple weighted superposition of fixed basis functions. Such a basis can be searched with simple Principal Component Analysis. A linear basis would largely avoid the concerns about use of autoencoders for saliency purpose that are discussed below. Unfortunately, in case of images this is workable only for some specific applications such as already aligned face pictures [93]. For general image datasets, the basis would either be so low-level as to offer little advantage over single pixels, or else just copy training images verbatim.

In practice, the go-to candidate for an image autoencoder are neural networks, most typically convolutional and can be structurally very similar to the ones used for image classification or the denoising task of chapter 7. The reasons such networks work well for those tasks are much the same as why they are good candidates for autoencoders. See Section 5.3.

The objective to the training of an autoencoder is to achieve $\mathsf{D}_\theta\left(\mathsf{E}_\phi\left(x\right)\right) \approx x$ for x from the dataset/distribution. This if of course very similar to the operators in Equation 3.1, except that in this case even $\mathsf{E}_\phi \circ \mathsf{D}_\theta = \mathrm{id}$ is only approximated through training, and there are not necessarily clear subspaces on which the two functions act as isomorphisms.

That notwithstanding, if the approximations are good and the latent space low-dimensional then it is a promising representation for performing the masking in. This could be done with

$$\begin{aligned} \left[{}^{x_R}_{x_L}\right]^{(\mathsf{E}_\phi,\mathsf{D}_\theta)} &\colon \Lambda \to \mathcal{I} \\ \left[{}^{x_R}_{x_L}\right]^{(\mathsf{E}_\phi,\mathsf{D}_\theta)}\vartheta &= \mathsf{D}_\theta\left(\left(1-\vartheta\right) \odot \mathsf{E}_\phi\left(x_L\right) + \vartheta \odot \mathsf{E}_\phi\left(x_R\right)\right). \end{aligned} \tag{3.6}$$

This, combined with standard gradient-based explanation techniques, is essentially the approach taken by Bordt et al. [12], who understand the autoencoder as a way of staying in the data manifold, or equivalently projecting any gradients into its tangent space. We have not yet attempted using something akin for the ablation path saliency

or one of the other interventional methods. One practical hindrance is that the latent space is completely abstract and a "heatmap" in it would not provide any explanation at all. This would be dispensable for an ablation path though, because of its ability to highlight small changes (regardless of what nature) together with their impact upon the classification; for the other methods it might also be possible to retrieve a spatial heatmap through some additional gradient-based attribution transfer.

A more fundamental issue is that the encoding is not interpretable itself. This is perhaps not quite as problematic as with a single mask-producing explainer black-box model (Section 1.2.2A), because one still has input-output pairs that offer some faithfulness check. But it is nevertheless far from ideal. In particular, it is conceivable that both the classifier and the autoencoder would learn a bias in the dataset, with the result that the synthetic inputs would all share a trait (perhaps delocalised and nearly invisible to humans) that makes the classifier behave more as it did on the training and validation set. This might result in innocuous-looking saliency maps even though the classifier is actually much more erratic on real data that do not have this bias.

None of this is necessarily disqualifies autoencoders for feature generation, but it does at any rate demand careful research into the influence of such issues and possibilities to mitigate them. This might be worth it for a classifier that cannot stably be explained in any other way, but if an explanation using only inherently interpretable features is possible than this does seem strongly preferrable. For this thesis, it was decided to focus the effort into the latter direction, as detailed in the next section and chapter 4.

It is also worth noting that a deep neural network classifier itself contains a learned representation in its intermediate layers, which can be quite similar to the one of an autoencoder trained on the same dataset.[8] For these internal learned features, Grad-CAM [87] already provides an efficient attribution technique. Comparing this to an independent autoencoder would be interesting regardless, but it might not provide much new insights.

3.4 Multiscale methods

The pixel basis is maximally localised (for a given bandwidth). The opposite is the Fourier basis, which is maximally localised in frequency space (for a given image size), which entails being maximally *de*localised in position space. Locality per se is advantageous for a saliency method, and at least some degree of locality is required for methods like RISE to give usable results at all. On the other hand, it is quite nonsensical for a saliency method to locate e.g. individual spots in a homogeneous surface in an image. The only way this kind of locality can matter is in adversarial masks. Therefore it it sensible to relax the locality demand. Smoothness regularizations have this as a side effect, but they are a fairly blunt tool that also precludes finely locating true small physical objects in a photo.

It was discussed above that the capability for adversarial interventions is related to information introduced by the masking process. Fourier analysis demonstrates a simple way how even a large image can have low information content: if it is sparse in

[8] It will not be similar in the sense that there are co-aligned baseis in the latent spaces, but it may still describe a very similar distribution.

frequency space; being homogeneously coloured is a special case of this. In that case, most pixel-wise masking will effect an increase in information content. It is uncommon for images to be globally sparse in Fourier space, though: most do contain at least some sharp edges / transients or otherwise broadband components, and even if these are themselves spatially confined the Fourier decomposition has no way of keeping them apart from the frequency-sparse regions. This, combined with the non-locality, makes a simple Fourier decomposition quite useless for explainability.

A compromise between pixel space and Fourier space is a promising way of resolving the dilemma. This can be approached from either space, and both are deployed in image compression:

- Grouping together pixels within each homogeneous or near-homogeneous region is one way to interpret PNG compression [1]. This is not literally how it works, but the use of delta coding and run-length coding has a similar effect. In the literal sense, such a grouping requires edge detection, watersheds etc. as in [86]. PNG compression is highly efficient for simple graphics, but not very effective for photos (at least without dedicated quantization [54]); this can be explained as the region decomposition being unstable under lighting subtleties and noise. As long as the goal is only to reconstruct the original image, the only consequence is large file size. But if the corresponding features are supposed to be used for interventional saliency, such instability is hardly tolerable. Anyway this would be difficult to carry out; at least PNG is constructed with purely discrete concepts and it is doubtful how it would permit any notion of *attenuating* features.

- Splitting an image into small regions (typically square blocks) and Fourier-transforming these allows strongly compressing only the ones that are frequency-sparse, while mostly keeping the local information of broadband regions. This is the main principle behind JPEG [100] and many other compression algorithms. Audio compression too usually employs short-time Fourier transform. Such transforms are purely linear decompositions, which makes them easy enough to use for interventions. However, they require a fixed choice of block size, and modification of the coefficients (including the quantization needed to use this for actual compression) is known to manifest in substantial ringing- / blocking artifacts.

Both approaches are successful at capturing some of the regularity inherent to images, but neither is really suitable for our purposes. Though the concrete problems are quite different, part of the underlying cause is shared: slow-varying, long-distance pixel correlations are only captured when not disturbed by higher-frequency contributions, or only under a tradeoff of accuracy of these contributions. A solution is to treat different frequency ranges differently, such that high-frequency contributions can stay localised whereas low-frequency ones can stay long-ranged.

That is the idea behind *wavelet transforms*. Like the Fourier transform, these represent any signal as a superposition of basis functions from an orthogonal set with respect to the $\mathcal{L}^2$ scalar product, or at least a set of biorthogonal pairs. In particular, they are true transforms in the sense of isomorphism.

Unlike with the Fourier transform, most of the basis functions are still localised in the sense of being supported within a small area of the domain. The more localised

elements of the basis inhabit a higher frequency band than the more extended ones; this is typically ensured by using a single "mother wavelet" and scaling it down, which simultaneously scales up the frequency components.

All of this makes wavelets fairly good candidates as features for saliency. Again, there is a choice between multiple versions, including:

- The Haar wavelet [35] is a simple pair of adjacent, opposite-polarity boxcar functions. It has advantages of simplicity and lack of ringing, but a disadvantage of discontinuity. Particularly troubling is that changing the amplitude of a single long-range wavelet would in general imprint the hard edges of its boxcar halves into the image.

- Daubechies wavelets [23] generalise and smoothen these. The price for this is a considerably more complicated shape and loss of shift invariance, though they are still efficient to compute.

- Morlet / Gabor wavelets [67] are simple and have good properties in the sense of both translation invariance and continuity. They are literally defined as spacially confined versions of sinusoidals (i.e. Fourier basis functions). They are computationally somewhat less efficient, though this would hardly matter. More problematic is that they do ring/oscillate multiple times within the window, albeit with exponential decay.

At least some sign change is unavoidable for wavelets, since in order to form an orthogonal system (or even just a well-conditioned invertible transformation matrix) the inner products between different base functions need cancelling contributions. This could potentially be a problem for saliency interventions because removing a feature can cause the maximum to increase, i.e. leave the allowed amplitude range, though this seems relatively easy to address and wavelets still remain an interesting option for saliency purposes. This combination has recently been realised [46], and also extended to *shearlets* [47] which are more suitable for capturing in particular edges.

Key insight for the method presented in the next chapter is that orthogonality, or indeed the notion of basis in the linear algebra sense altogether, is not necessary for features. This allows constructing the method based on only Gaussian kernels, avoiding any negativity as well as information addition, while still having similar frequency-dependent locality as a wavelet expansion.

3.5 Linear vs nonlinear

One aspect common to Fourier and wavelet representations, and also to the Shannon-Nyquist interpretation of PCM, is that the complete image is considered to be a linear superposition of multiple components. This can in many cases be argued to be unproblematic; for example sound waves do also in the physical reality superimpose in a way that simply adds the pressures (relative to ambient). In this case it is exploited that the appropriate physical model (compressible Euler equations) can as a good approximation be linearised, at least for relatively low-amplitude signals.At sufficiently high amplitudes, linear approximations generally break down. For some physical

systems like the vacuum Maxwell equations that would mean so high field strengths as to be unachievable anyway, but for pixel brightness variations a linearised view already breaks down very quickly, since there is a hard-limited range of permittable pointwise values and certainly no such thing as negative brightness. This limitation is somewhat ironic since light is a phenomenon. Moreover it is often dubious whether addition in a vector space is an appropriate framework at all. Particularly for photographs, a view of overlapping patches seems a more appropriate paradigm.

Even when accepting superpositions as in principle reasonable, it is still important to consider that whilst they commute with arbitrary basis transformations on the vector space, they do *not* commute with nonlinear transformations. Particularly relevant for image applications (unlike audio[9]), the actual pixel values are usually not stored as numbers proportional to physical intensities, but rather in a nonlinearly mapped colour space like sRGB [41] or CIELAB [83]. These nonlinearities are intended to more closely match human perception, which has advantages including more efficient use of the bits (this could incidentally be seen as a form of entropy-coding compression), and are possibly also benefitial when using the representation for saliency intervention (particularly for metrics on the mask space, cf. Section 4.3.3).

The prevalence of linear-based representations certainly has at least as much to do with computational practicality as with properties of the application domains. Even if a dataset results from a highly nonlinear process, linear tools can still have their uses here[10], but they will always be limited.

It may well be one of the reasons of the success of deep learning that it better embraces nonlinearity than traditional methods. It is possible, too, that an explanation technique based on any linear basis expansion can not hope to properly capture deep learning decisions, but in absence of interpretable nonlinear explanation techniques it seems necessary to try making the best of the linear approaches.

[9] Formats like μ-law aside, which are of mostly historical relevance.

[10] An example from physics is the use of Fourier analysis to study turbulence [48][114].

CHAPTER 4

SIFT-BASED ABLATION

The Scale-Invariant Feature Transform was developed by Lowe [59] as a technique for extracting information from images that is practical to use and stable with respect to commonly encountered disturbances that hamper many other techniques. Concretely,

- It separates large-scale, low-frequency features from small, high-frequency ones – much like the wavelet transforms discussed in Section 3.4. This is particularly relevant for stability under lighting conditions and noise.
- It avoids depending on ad-hoc choices for rasterisation or basis-functions. The only consequential fixed choice is that of Gaussian filters, for which there are very compelling reasons.[58]
- Translations, rotations and scalings of the input manifest directly in corresponding transformations, thus the name "scale-invariant". This makes it stable under changes in the exact manner a photo is taken.
- It is efficient to compute. This was more important in 1999 than with the GPU resources available nowadays, but it is still useful. The adaptation presented here only exploits some of the efficiency tricks of the original; this is enough to avoid it being a computational bottleneck.

Remark 13. *Arguably, the method should not be called SIFT but SEFE: Scale-Equivariant Feature Extraction. Were it actually invariant, the outputs would not change at all under scaling (see chapter 5 for terminology). The name "transformation" is not very fitting either since, unlike Fourier- or wavelet transforms and many others, SIFT results in only discrete keypoints. It does not depend on the entire input information, and certainly is not invertible – although, as this chapter demonstrates, it is still reconstructible in a different sense that turns out to be sufficient for saliency purposes.*

A nutshell description of how SIFT works is that it extracts the extrema of an image in its scale-space representation. These extrema or *keypoints* are used as the features.

The nutshell description of our extension to it is that associates each of these keypoints with the actual signal information most pertaining to it, and devises a way to reconstuct the entire image from that, with the possibility to attenuate each of the features independently.

Several reasons for using the SIFT keypoints as features for image classification were given in chapter 3; the arguments are elaborated in the following sections. Another

reason which is perhaps even more compelling is that SIFT was once itself a main tool within the state of the art of image classification. Most entries to the ImageNet Large Scale Visual Recognition challenge 2010 [82][11] used SIFT in some way[1], before deep learning pushed such methods out of competition starting in 2012. SIFT may thus provide the most promising *interpretable decomposition*, inasmuch as its notion of information has proven sufficient for a good part of the classification task, while still being simple and sparse.

4.1 SIF-*Transform*: a reconstructible formulation

Recall the general notions from Section 1.1 of the space $\mathcal{I}$ of inputs/images, which is some function space on a domain Ω with values in a space V that usually represents colour. This section always considers a single given image $x \in \mathcal{I}$ with $\mathcal{I} = \mathcal{L}^2(\Omega, V)$ and $\Omega = [0,1]^n$. In the image application it is $n = 2$, whereas in the figures below demonstrating the concept $n = 1$ is used for visibility. The image x can be treated as a function

$$x : \Omega \to V.$$

To be pendantic, elements of $\mathcal{L}^2$ are equivalence classes of functions, but they can for the purposes relevant here be treated like single functions. The main reason for working in $\mathcal{L}^2$ is that Fourier transform is well-behaved on that space, which is useful for the theory employed here.

4.1.1 *To scale space and back*

The first processing step of SIFT is to transform this image into its scale-space representation, specifically its *Difference-Of-Gaussians Pyramid*.

4.1.1A *The space of scales*

Scale space was introduced by Witkin [113]. The idea is to avoid choosing any particular length scale or frequency range, or grids in either position- or frequency space, but instead using *all* the scales. This amounts to adding a dimension to the signal's domain, the *scale dimension*, which is indeed physically a length dimension. In that combined domain

$$\mathcal{Z} := \Omega \times \mathbb{R}^+, \tag{4.1}$$

define the progressively low-pass filtered image:

$$\begin{aligned} \Gamma\colon\ \mathcal{Z} &\to V \\ \Gamma(\mathbf{r}, \sigma) &:= (\gamma_\sigma \star x)(\mathbf{r}), \end{aligned} \tag{4.2}$$

with the ($\mathcal{L}^2$-normalized) Gaussian kernel (Equation 2.25).

[1] According to the ImageNet publications, the winning entry to ILSVRC2010 is among those using SIFT. The paper associated with that method [57] does not mention this, though.

4.1.1B *Fourier analysis*

We defer to textbooks for the basic definitions of Fourier theory. Details such as absolute amplitudes are inessental for the following discussion; what is important is mostly the standard results for Gaussian functions, and that a normalisation convention is used where the convolution theorem holds as

$$\begin{gathered}\mathcal{FT}\left(f \star g\right)\left(\mathbf{k}\right) = \mathcal{FT}\left(f\right)\left(\mathbf{k}\right) \cdot \mathcal{FT}\left(g\right)\left(\mathbf{k}\right) \\ \forall f, g : \mathcal{L}^2\left(\mathbb{R}^n\right).\end{gathered} \tag{4.3}$$

The Gaussian-lowpass filtered cascade Γ is a highly redundant representation of the image x: all sufficiently low-frequency information will be present at full strength in all the $\Gamma\left(\cdot, \sigma\right)$ slices at sufficiently small σ. This can be seen in its Fourier transform:

$$\begin{aligned}\mathcal{FT}\left(\Gamma\left(\cdot, \sigma\right)\right)\left(\mathbf{k}\right) &= \mathcal{FT}\left(\gamma_\sigma\right)\left(\mathbf{k}\right) \cdot \mathcal{FT}\left(x\right)\left(\mathbf{k}\right) \\ &= e^{-\frac{\|\mathbf{k}\|^2 \cdot \sigma^2}{2}} \cdot \mathcal{FT}\left(x\right)\left(\mathbf{k}\right).\end{aligned} \tag{4.4}$$

Specifically, the exponential approaches constant 1 whenever $\sigma \cdot \mathbf{k} \ll 1$, which makes the result approximately independent of the concrete value σ within that domain. By contrast, the *differential of Gaussians* representation

$$\begin{gathered}\Delta\colon\ \mathcal{Z} \to V \\ \Delta\left(\mathbf{r}, \sigma\right) := -\sigma \cdot \frac{\partial}{\partial \sigma}\left(\Gamma\left(\mathbf{r}, \sigma\right)\right)\end{gathered} \tag{4.5}$$

retains for each σ only that information which is added as decreasing size of the Gaussian kernel increases the bandwidth (cf. Figure 4.1).

Remark 14. *The factor $-\sigma$ in Equation 4.5 is largely unmotivated at this point, except for the superficial benefit of avoiding a 1/length contribution in the physical dimension of Δ. The factor is not necessary for the purposes of the current section, but it does make the differential formulation match up with the practical* difference *of Gaussians formulation; see Lemma 9.*

The Fourier transform shows the bandpass property explicitly:

$$\begin{aligned}\mathcal{FT}\left(\Delta\left(\cdot, \sigma\right)\right)\left(\mathbf{k}\right) &= -\sigma \cdot \left(\frac{\partial}{\partial \sigma}\left(\mathcal{FT}\left(\gamma_\sigma\right)\left(\mathbf{k}\right)\right)\right) \cdot \mathcal{FT}\left(x\right)\left(\mathbf{k}\right) \\ &= \|\mathbf{k}\|^2 \cdot \sigma^2 \cdot e^{-\frac{\|\mathbf{k}\|^2 \cdot \sigma^2}{2}} \cdot \mathcal{FT}\left(x\right)\left(\mathbf{k}\right).\end{aligned} \tag{4.6}$$

Here, the lowest frequencies approach zero gain (instead of unity gain as in Equation 4.4) due to the $\|\mathbf{k}\|^2$ factor, but the magnitude still depends on σ^2.

The bandpass filtering can also directly be expressed as convolution with a different kernel:

$$\Delta\left(\mathbf{r}, \sigma\right) = \left(\mathrm{BPK}_\sigma \star x\right)\left(\mathbf{r}\right) \tag{4.7}$$

where the kernel BPK_σ (Figure 4.2) is obtained either by inverse Fourier transform, or

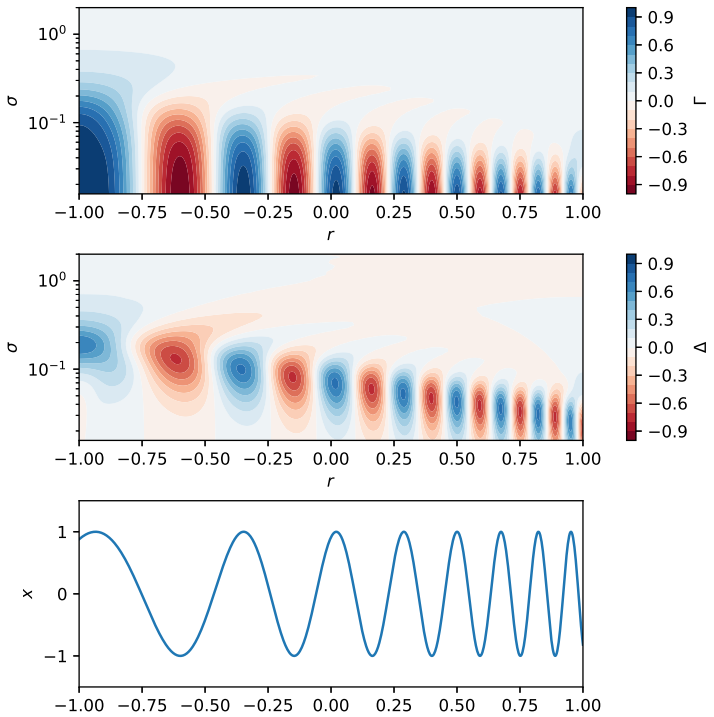


Fig. 4.1: The Gaussian lowpass and differential-of-Gaussians representations of a simple sinusoidal sweep, demonstrating the bandpass character. Observe that the minima and maxima of the original signal manifest in extrema of Δ that are isolated in both of the $\mathbf{r}$ and σ dimensions.

by differentiation of the Gaussian kernel:

$$\mathrm{BPK}\left(\mathbf{r}\right) = -\sigma \cdot \frac{\partial}{\partial \sigma}\left(\gamma_\sigma\left(\mathbf{r}\right)\right). \tag{4.8}$$

4.1.1C *Original image reconstruction*

Each band pass filter discards[2] most of the information in the image, but it is still preserved in the collection of all of them together:

Lemma 8. *The image $x \in \mathcal{L}^2\left(\mathcal{I}\right)$ can be computed exactly from the scale-space representation $\Delta \in \mathcal{L}^2\left(\mathcal{Z}\right)$:*

$$x\left(\mathbf{r}\right) = \int_0^\infty \mathrm{d}\sigma \, \frac{\Delta\left(\mathbf{r}, \sigma\right)}{\sigma}.$$

Proof. This is essentially only application of the fundamental theorem of calculus. The

[2]Technically speaking, the information is not completely discarded: unlike in a Fourier transform, each layer also lets some of the neighbouring frequencies through and technically speaking even faraway ones – but with over-exponential attenuation, so that reconstructing from a single bandpass-filtered version would be highly unstable. The leakiness in frequency space is a tradeoff for the much better locality in position space of a Gaussian kernel, compared to the sinc kernels that correspond to hard frequency cutoffs.

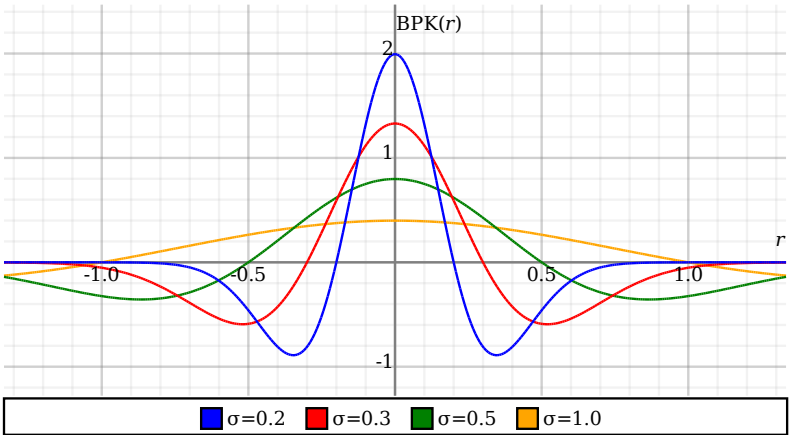


Fig. 4.2: The kernel corresponding to the band-pass interpretation of Δ. This kernel is not explicitly used in the implementation, only shown for reference. Notice that this could also be considered as a wavelet (cf. Section 3.4).

improper integral is defined as a limit of proper ones, to which the theorem applies:

$$\begin{aligned}
\int_0^\infty \mathrm{d}\sigma\, \frac{\Delta(\mathbf{r},\sigma)}{\sigma} &= \lim_{R\to\infty}\left(\int_0^R \mathrm{d}\sigma\, \frac{\Delta(\mathbf{r},\sigma)}{\sigma}\right)\\
&= \lim_{R\to\infty}\left(\int_0^R \mathrm{d}\sigma\, \left(-\frac{\partial}{\partial\sigma}\left(\Gamma(\mathbf{r},\sigma)\right)\right)\right)\\
&= -\lim_{R\to\infty}\left(\left[\Gamma(\mathbf{r},\sigma)\right]\Big|_{\sigma=0}^{R}\right)\\
&= \Gamma(\mathbf{r},0) - \lim_{R\to\infty}\left(\Gamma(\mathbf{r},R)\right).
\end{aligned}$$

Both contributions can only be understood in a limit sense. The Fourier transform is a bounded[3] linear operator and thus continuous, meaning it commutes with the limit and can directly be evaluated from Equation 4.4:

$$\mathcal{FT}\left(\Gamma(\mathbf{r},0)\right)(\mathbf{k}) = e^{0}\cdot\mathcal{FT}(x)(\mathbf{k}) = \mathcal{FT}(x)(\mathbf{k}),$$

and thus $\Gamma(\mathbf{r},0) = x(\mathbf{r})$; and

$$\mathcal{FT}\left(\lim_{R\to\infty}\left(\Gamma(\mathbf{r},R)\right)\right)(\mathbf{k}) = \lim_{R\to\infty}\left(e^{-\frac{\|\mathbf{k}\|^2\cdot R^2}{2}}\right)\cdot\mathcal{FT}(x)(\mathbf{k}) = 0,$$

for any $\mathbf{k} = 0$. Since $\{0\}$ is a null set and values on a null set do not matter for Lebesgue integration, this means $\mathcal{FT}\left(\lim_{R\to\infty}\left(\Gamma(\mathbf{r},R)\right)\right) = 0$ in the $\mathcal{L}^2$ sense. It follows that $\lim_{R\to\infty}\left(\Gamma(\mathbf{r},R)\right) = 0$.

[3] In fact, it is unitary (Parseval's theorem).

Putting it together, one obtains

$$\int_0^\infty \mathrm{d}\sigma \, \frac{\Delta(\mathbf{r}, \sigma)}{\sigma} = x(\mathbf{r}) - 0 = x(\mathbf{r}) \,.$$

□

The above result demonstrates exact reconstruction is possible from the fully-continuous, unlimited-scale representation. In practice it is neither feasible nor necessary to integrate all the way to infinity, instead one simply goes to a finite size σ, generally chosen as similar to the whole image size R, and stores the single low-pass representation from there on, which is added to the remainder of the integral.

As a justification, the information in $\Delta(\mathbf{r}, \sigma)$ for $\sigma \gg R$ is not relevant as features of x. In particular, there can be no local maxima here (see next section) because of the monotonic decay with σ. Thus it is fully suffcient to store it only up to a maximum on the order of R, and basing the reconstruction on that finite range.

4.1.2 *SIFT keypoints – idealized*

The idea behind the Scale-Invariant Feature Transform [60] can be summarized as tracking all the extrema of Δ, i.e. all pairs

$$\kappa_i = (\mathbf{r}_i, \sigma_i) \in \mathcal{Z}$$

for which $\Delta(\kappa_i)$ is either a local minimum or local maximum. These κ are called *keypoints*.

A principal, and namesake[4], property of SIFT is that the output is equivariant under several input transformations; cf. chapter 5. Specifically, translations, rotations and scalings of the input images map to corresponding translations of the keypoints. This property was considered important for use of SIFT in its original applications, it ties in to the symmetry investigations in the next chapters, and more concretely it is also precondition for the notion that the keypoints are located *inside* an image's scale-space expansion, which is needed for the following construction.

It has several advantages to take the extrema in the $n+1$-dimensional space $\mathcal{Z}$, instead of the extrema of x in Ω (which would still be equivariant). One of them is that significant extrema from the signal are decoupled from those of noise contribution, as demonstrated in Figure 4.3. The (typically lower-frequency and/or higher-amplitude) features of interest can thus be used uninterfered by the sporadic noise ones.

A caveat that needs to be made here is that a continuous, physical signal has an unbounded noise spectrum.[5] That means there are an infinite number of extrema at values of σ smaller than the features of the actual signal, which cannot even be processed (making the point moot of whether they would interfere with the signal ones). In practice, this situation analogous to the classical ultraviolet-catastrophe is

[4] Barring Remark 13

[5] Being even more pedantic, the spectrum *is* bounded due to quantum mechanics at finite temperature. For most applications this cutoff is not of relevance though, since it occurs many orders of magnitude above the frequencies that are actually captured.

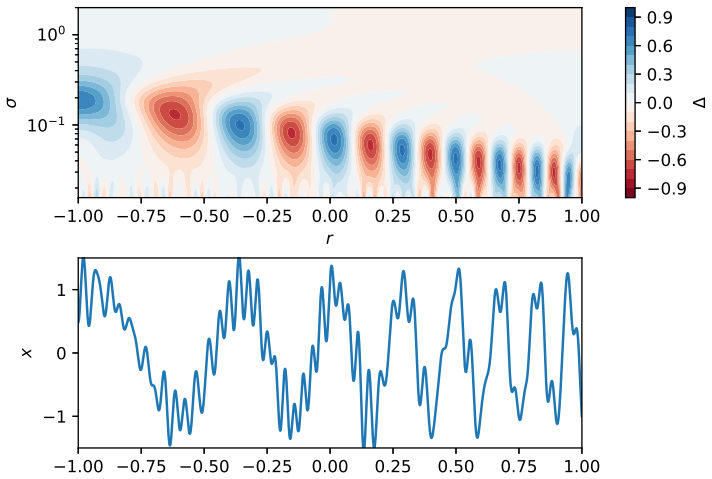


Fig. 4.3: DOG of a sinusoidal sweep as in Figure 4.1, but with (bandlimited) noise superimposed. In this case the extrema in Δ, and thus SIFT keypoints, do not coincide with those in the noisy signal x anymore.

avoided by a combination of limiting the smallest σ and a minimum contrast between minima and maxima, in addition to other conditions.

Remark 15. *Critical readers may remonstrate that when SIFT is used for saliency, such cutoffs have an analogous effect to the regularization employed in the literature (Section 1.2.3). This cannot be dismissed entirely, but a crucial difference is that for SIFT the cutoff is directly concerned with noise in the image itself. The properties of this noise can be obtained in quite reliable and transparent ways (ideally from e.g. camera metadata, but it is also doable with interpretable heuristics on the pixel data [75]). By contrast, e.g. the undersampled masks used in RISE [73] preclude an entire frequency band from the masks, and the target image will in general contain not only noise but also many signal features in that high-frequency band. TV-regularization has slightly different, but still analogous problems.*

The criticism may nevertheless have some relevance in practice, because what should be considered (or at least, what the classifier considers) as signal vs noise can be quite different from the purely physical image noise.

The original version of SIFT [59] computed the extrema directly on the chosen discretized form of Δ, see Section 4.2.2. These can differ considerably from the extrema in the continuous form. The version that is now considered the standard SIFT [60] uses however a cubic interpolation [14] that gives a much better approximation to the true continuous extrema, so much that it is appropriate to consider them as interchangeable and base the discussion here entirely on the continuous case.

Another caveat is that the notion of minima and maxima only makes sense for scalar fields, but Δ is vector valued with colours in $V \simeq \mathbb{R}^3$. There are extensions of SIFT that work directly with colour images [15], but for this work the keypoints were simply obtained on a greyscale reduction of the image. This is justified insofar as the luma component of an image generally holds the most information. Arguments in support of this include the fact that image encodings (not only digital ones like JPEG [100], also analogue ones like PAL) assign most of the bandwidth to the luma, more than both chroma channels combined; this is based largely on human perception, which

does not uniformly respond to RGB changes [16]. Another one is the possibility to reconstruct convincingly all the colour of many photographic images from only the greyscale-reduced version [117], though this does not always work and relies on deep learning.

It is clear that there are examples where greyscale-based keypoints are not appropriate, but often they appear to be sufficient.

For the purposes of this chapter, the keypoints are all that is needed. Most applications employing SIFT use also additional orientation information computed from image gradients. These could potentially be useful also for a feature decomposition similar to the one proposed in the following, but this possibility has not been explored yet.

Inclusion of colour and/or directional information are interesting possibilities for future research.

4.1.3 *Feature cells*

In Section 4.1.1 we showed that the differential-of-Gaussians Δ contains all information to reconstruct the signal, which does in principle allow using it as a latent space on which masks could be applied, like the ones considered in Section 3.3.2. This would however not be usable for saliency because Δ is even less information-efficient than x in pixel representation.

On the other hand, the SIFT keypoints κ are a highly efficient representation in the sense of information: for the typical images from datasets like ImageNet [82] or COCO [56] with sizes around 500×400 pixels, the SIFT with typical cutoff parameters yields on the order of 1000 keypoints. But as said before, these points are (notwithstanding their usefulness for image processing tasks) not sufficient to accurately reconstruct x. They inhabit the domain of Δ, but only meagrely.

Remark 16. *There exist also attempts to reconstruct images directly from only their SIFT keypoints or similar representations, but the only workable solutions use deep learning [62]. Apart from the inherent concerns about interpretability, bias susceptibility etc. that this raises again, it is also fundamentally only able to generate approximations to x which have significant errors without guaranteed bounds. The method presented here can meanwhile do it exactly (up to numerical rounding).*

The solution we propose combines the advantages of Δ and κ: it applies masks only as gain factors ϑ_i to each of the manageably few κ_i, but then reconstructs entire images under use of additional information from Δ. The trick to this is determining what information of Δ pertains to each of the κ_i, which is feasible since these keypoints are scattered in the domain of Δ.

In terms of encoder/decoder signatures as in Section 3.3.2, a type like the following would be suggestive if the keypoints were all that is needed:

$$\begin{aligned} \mathrm{E}_{\mathrm{SIFT}} &\colon \mathcal{I} \to \mathcal{P}(\mathcal{Z}) \\ \mathrm{D}_{\mathrm{SIFT}} &\colon \mathcal{P}(\mathcal{Z}) \to \mathcal{I} \end{aligned} \tag{4.9}$$

Here, sets of keypoints κ would be the intermediate representation, so that the latent space is the power set of the scale space $\mathcal{Z}$. As said above, this is not practical:

- The keypoints are *not* sufficient. Our solution is to add the entire DOG expansion $\Delta \in \mathcal{L}^2(\mathcal{Z}, V)$ to the intermediate representation as well, i.e. making the latent space a cartesian product with $\Lambda := \mathcal{L}^2(\mathcal{Z}, V)$.

- Working with sets is impractical. Although finite sets can be computer-represented with tree data structures or hash maps, they are not efficient for GPU computations. Those require flat data storage (readily achievable by storing the points in an array, modulo ordering) with predictable dimensions (requiring to keep track of the number of keypoints and using correspondingly-sized arrays). More fundamentally, even a set of sets of fixed size does not have an easily usable topology on it, but the K-fold cartesian product $\mathcal{Z}^K$ has the usual inherited geometry.

The full signature can be expressed with the following (dependent, à la Martin-Löf [65]) types:

$$\begin{aligned} E_{\mathrm{SIFT}} &: \mathcal{I} \to \sum_{K:\mathbb{N}} \Lambda \times \mathcal{Z}^K \\ D_{\mathrm{SIFT}} &: \prod_{K:\mathbb{N}} \left(\Lambda \times (\mathcal{Z} \times \mathbb{R})^K \to \mathcal{I} \right) \end{aligned} \tag{4.10}$$

The pi- and sigma types express that the number of keypoints is image-dependent; K is the number of keypoints and κ are the keypoints themselves. The decoder accepts in addition to the data given by the encoder a gain factor associated with each keypoint.

Remark 17. *A simpler, weaker-typed formulation is that* E_{SIFT} *yields a* list *of keypoints, and* D_{SIFT} *takes a list of keypoint-gain pairs.*

$$\begin{aligned} E_{\mathrm{SIFT}} &: \mathcal{I} \to \Lambda \times \mathcal{Z}^* \\ D_{\mathrm{SIFT}} &: \Lambda \times (\mathcal{Z} \times \mathbb{R})^* \to \mathcal{I}. \end{aligned} \tag{4.11}$$

This signature is however badly suited when the decoder will be regarded as a differentiable function of the gain factors.

E_{SIFT} and D_{SIFT} represent an encoding in the sense that, if

$$E_{\mathrm{SIFT}}(x) = (K, (\Delta, \kappa)),$$

then

$$(D_{\mathrm{SIFT}})_K (\Delta, [(\kappa_i, 1) \,|\, i < k]) = x, \tag{4.12}$$

i.e. setting the gain of each keypoint to 1 reconstructs the original image. Simplifying this notation, write (with implicit K)

$$D_{\mathrm{SIFT}}(\Delta, \mathrm{zip}(\kappa, \mathbf{1})) = x \tag{4.13}$$

or even

$$D_{\mathrm{SIFT}}^{\Delta,\kappa}(\mathbf{1}) = x. \tag{4.14}$$

4.1.3A *Encoder*

E_{SIFT} was essentially defined already: it simply maps the image x to its scale-space, differential-of-Gaussians expansion Δ and the SIFT keypoints κ in an array with arbitrary ordering. None of this is novel.

4.1.3B *Decoder version 1 (partition)*

A main part of the decoder D_{SIFT} is the recomposition derived in Section 4.1.1, but that by itself does not provide a way to apply the gain factors.

The simplest way to accomplish this is to partition the whole domain $\mathcal{Z}$ into K sectors $\check{\mathcal{Z}}_i$, one for each feature. A natural requirement is that each keypoint should be "within" its associated feature:

$$\kappa_i \in \check{\mathcal{Z}}_i. \tag{4.15}$$

Each such sector gives rise to a restricted function

$$\begin{aligned} \check{\Delta}_i &: \check{\mathcal{Z}}_i \to V \\ \check{\Delta}_i &:= \Delta|_{\check{\mathcal{Z}}_i}. \end{aligned} \tag{4.16}$$

Since a partition is disjoint and covers the entire space, the full Δ can be re-assembled from that:

$$\Delta(\mathbf{r}, \sigma) = \check{\Delta}_{\check{\imath}_{\mathbf{r},\sigma}}(\mathbf{r}, \sigma) \tag{4.17}$$

where $\check{\imath}_{\mathbf{r},\sigma}$ is the unique index such that $(\mathbf{r}, \sigma) \in \check{\mathcal{Z}}_{\check{\imath}_{\mathbf{r},\sigma}}$.

Remark 18. *In practice, it is sufficient for the $\check{\mathcal{Z}}_i$ to cover only* almost *the entire space $\mathcal{Z}$, because Δ is continuous and can therefore be reconstructed from its restriction to any dense subset. See Remark 19 for why this is important.*

Since the scale space $\mathcal{Z}$ is just a direct-sum space with in total 3 length-like dimensions (two from $\mathbf{r} \in \Omega$, one from σ)[6], it is suggestive to use the Euclidean $\mathbb{R}^3$ metric for constructing the partitions. This has several desirable properties, in particular it is equivariant under scaling $(\mathbf{r}, \sigma) \mapsto (\mu{\cdot}\mathbf{r}, \mu{\cdot}\sigma)$ and invariant under translations and rotations. The behaviour under translations and scalings is a key property of the SIFT algorithm, and it makes sense to preserve these properties also for the decomposition. Rotations however are only meaningful in the spatial components, i.e. within the $\mathcal{I} \subset \mathcal{Z}$ slices. Rotations outside of the spatial planes (in other words, with a rotation axis not parallel to the σ-axis) would mix localization with frequency information, which is nonsensical. Although the filtering parameter σ is physically length-like, it is not directly comparable to lengths in the sense of distances between pixels. This is because although σ parameterizes the width of the Gaussian peak γ, it is not the unique way of measuring it. These peaks are after all not sharply delimited – in a sense their size is infinite. More pragmatically, the size could be defined as the radius where γ_σ vanishes in the noise floor.

These considerations lead to a Euclidean-like distance function, but with a weighing

[6] One fewer dimension in the example figures with 1-dimensional Ω.

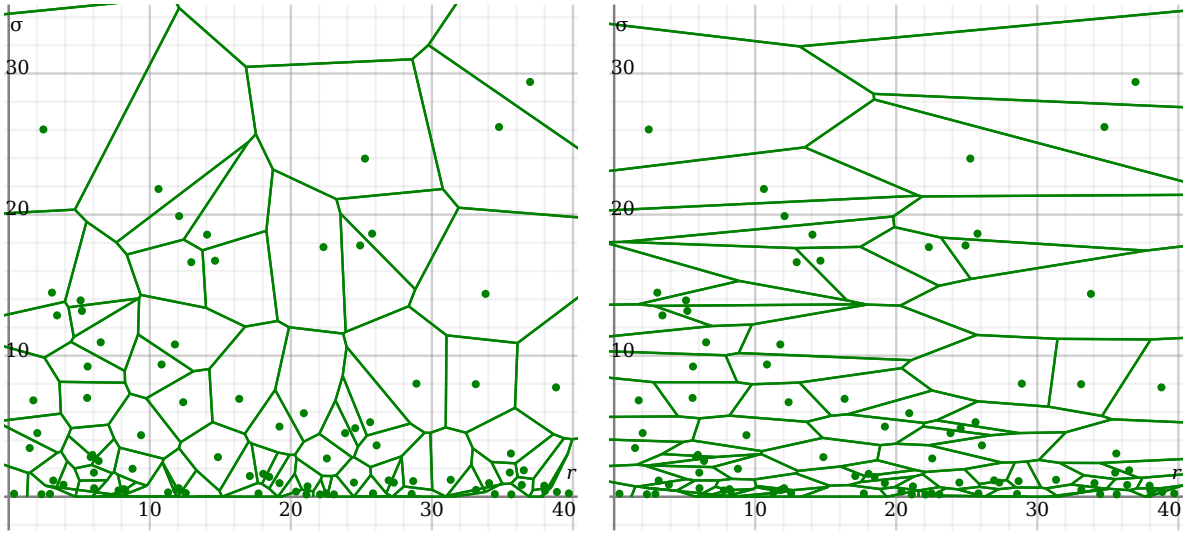


Fig. 4.4: Two Voronoi tesselations for the same selection of random-synthetic points, with different parameters $\eta = 1$ (left) and $\eta = 4$ (right). The σ-coordinates are sampled from a truncated exponential distribution, emulating the fact that most SIFT keypoints belong to high-frequency, narrow-localised features.

hyperparameter η that distinguishes the different coordinates:

$$\begin{aligned} \mathrm{d}_{\mathcal{Z},\eta} \colon\ & \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_{\geqslant 0} \\ \mathrm{d}_{\mathcal{Z},\eta}\left((\mathbf{r}_1, \sigma_1), (\mathbf{r}_0, \sigma_0)\right) &:= \sqrt{\left(\mathrm{d}_{\mathbb{R}^2}\left(\mathbf{r}_1, \mathbf{r}_0\right)\right)^2 + \eta^2 \cdot \left(\sigma_1 - \sigma_0\right)^2}. \end{aligned} \tag{4.18}$$

Effects of the choice of η are discussed below and in Section 4.3.4.

This distance function (or any other) can then be used for creating a partition. The most straightforward way of doing that is to associate each point in $\mathcal{Z}$ with the $\mathrm{d}_{\mathcal{Z},\eta}$-nearest keypoint (respectively, its index). The result is a *Voronoi tesselation*, a standard tool [5] for extending discrete data points to a metric space they are embedded in.

$$\begin{aligned} \mathfrak{V} \colon\ & \prod_{K:\mathbb{N}} \left(\mathcal{Z}^K \to \left(\mathcal{P}\left(\mathcal{Z}\right)\right)^K\right) \\ \left(\mathfrak{V}_K\left(\kappa\right)\right)_i &= \left\{\lambda \in \mathcal{Z} \colon\ \arg\min_j \left(\mathrm{d}_{\mathcal{Z},\eta}\left(\kappa_j, \lambda\right)\right) = i\right\}. \end{aligned} \tag{4.19}$$

Remark 19. *The Voronoi tesselation is not strictly speaking a partition: points that are equidistant from multiple keypoints cannot be unambiguously assigned to any of them. This does – at least in principle – not pose a problem for the recomposition, because these points form a null set in $\mathcal{Z}$; in particular a set whose complement is dense, so that Remark 18 applies. It would nevertheless lead to unbalanced results if the $\check{\imath}_{\mathbf{r},\sigma}$ are only assigned to the finitely many pixels and some of these (or, their centroids) are equidistant between keypoints (which would happen if the keypoints themselves have exact pixel centroid coordinates). In this case, the set of ambiguous assignments would not be a null set. This is another reason for using a Brown and Lowe [14] version of SIFT, because its cubic-estimated maxima almost never coincide with pixel centroids, whereas discrete-extremum keypoints [59] lie per definition always on a centroid.*

Remark 20. *For the discrete pixel case, it would arguably be better to not assign each pixel to exactly one Voronoi cell, but instead consider the pixel as a rectangle intersecting possibly multiple cells, and assign it to all of them weighted by size of overlap. The reason this was deemed unnecessary here is that the filtering applied later (Section 4.1.3C) has a similar effect already, smoothening out cell boundaries and assigning pixels at a boundary to a combination of the adjacent cells.*

Taking into account only a single cell simplifies the computation; indeed it is not necessary to construct the Voronoi tesselation as a collection of cells, but only to query for each $\mathcal{Z}$-voxel–centroid the nearest neighbour among the κ_j.

The effect of the η parameter in Equation 4.18 is to select how de-localised the Voronoi cells should be. Notice in Figure 4.4 that with $\eta = 1$, even the keypoints with $\sigma > 20$ still have cells fairly localised in space, though they correspond to Gaussian filters that would smear almost the entire image to a single uniform colour. With a higher value like $\eta = 4$, the cells stretching over almost the entire Ω domain embody this fact better.

The idea for building a decoder in the sense of Equation 4.10 based on the Voronoi decomposition is to perform the recomposition as in Equation 4.17, but with each $\check{\Delta}_i$ scaled by a gain factor ϑ_i. In other words, we define a kind of pointwise- or rather cellwise-multiplication operator $\circledast$, which applies each gain factor to the corresponding cell:

$$(\vartheta \circledast \Delta)(\mathbf{r}, \sigma) := \vartheta_{\check{\imath}_{\mathbf{r},\sigma}} \cdot \check{\Delta}_{\check{\imath}_{\mathbf{r},\sigma}}(\mathbf{r}, \sigma)\,. \tag{4.20}$$

In this case, the "mask" ϑ is only a vector / 1D-array of real numbers, and all spatial association it has is stored separately in the keypoints.

An equivalent formulation is as an entry-wise product with weighted characteristic functions[7] for each cell:

$$(\vartheta \circledast \Delta)(\mathbf{r}, \sigma) = \sum_i \vartheta_i \cdot \chi_i(\mathbf{r}, \sigma) \cdot \Delta(\mathbf{r}, \sigma)\,, \tag{4.21}$$

with

$$\chi_i(\mathbf{r}, \sigma) := \begin{cases} 1 & \text{if } \check{\imath}_{\mathbf{r},\sigma} = i \\ 0 & \text{else.} \end{cases} \tag{4.22}$$

The above defined SIFT-Voronoi decomposition already provides a notion of feature-basis that can be used for interventional saliency, albeit without a baseline as used previously (Section 1.2.3A). Namely, one can sandwich the cell-weighing Equation 4.20 between the computation of Δ (Equation 4.5) and the reconstruction of a feature-ablated version of x from it (Lemma 8). This does indeed work, however the resulting reconstructed images do not, in general, have the desirable properties suggested from the scale-space construction. Specifically, every σ-slice of $\vartheta \circledast \Delta(\mathbf{r}, \sigma)$ has hard edges where two Voronoi-cells (i.e., the slices of them) with different weights from ϑ meet. As a consequence, the resulting ablated image is not continuous, even if x is. For example, in

[7] Apologies for the use of the near-lookalike symbols x and χ in this chapter. The former is based on saliency literature, whereas χ (chi) is conventional notation for characteristic functions. Beware the distinction, as well as the threefold distinction between Σ (being the capital version of sigma) for sets of

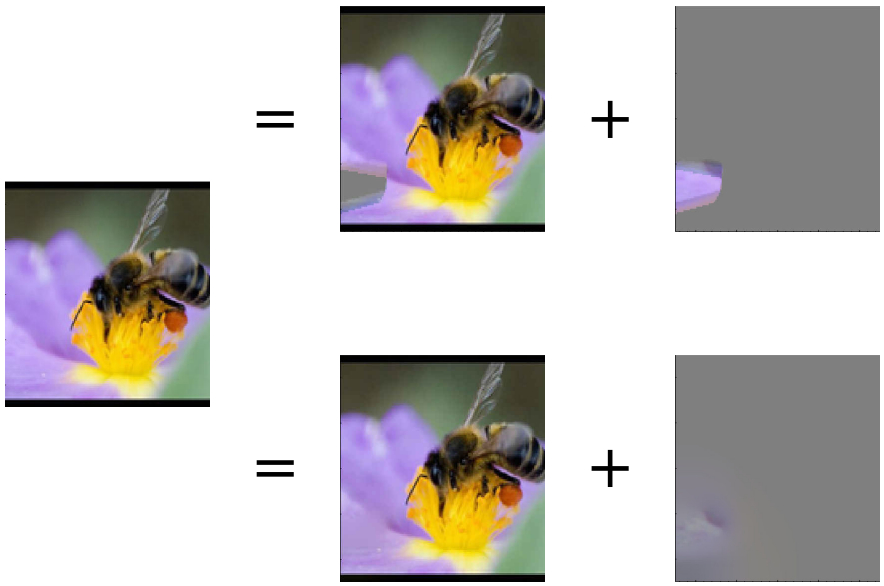


Fig. 4.5: One SIFT-Voronoi cell feature isolated from an image. Top: without smoothening; bottom: with smoothening (lowpass-after-gain, Equation 4.25). Photo: ImageNet [82]. Scaling-factor $\eta = 1$ (cf. Equation 4.18); N.B. larger η-values tend to avoid such strong edges turning up in the first place since they let the feature-separation take place more in the σ-dimension, but this does not completely avoid the phenomenon.

Figure 4.5 the removed feature in the flower petal introduces an almost perfectly sharp polygonal patch of different colour, which constitutes an obvious artificial addition to the image. This is reminiscent of artifacts that can arise from direct pixel modification (though these are usually far less clearly structured), and problematic as explained earlier (the newly introduced edges are prone to affecting the classifier directly, perhaps even adversarially).

For this reason, it is preferrable to use a slightly different notion of recomposition.

4.1.3C *Decoder version 2 (smooth cells)*

The discontinuity issue is particularly paradox since each σ corresponds not only to a particular part of the input information, but also to a particular degree of smoothness, induced by the Gaussian lowpass. It would be natural for this imposed smoothness to also apply to the ablated version $\vartheta \circledast \Delta$, but Equation 4.17 does not achieve that: the cell-wise weighing disrupts all smoothness previously introduced by the lowpass filtering. This order of operation arose naturally from the definition of Δ as a differential of pure low-pass filtering, but as per Equation 4.7 this is equivalent to directly applying bandpass filters, and these in turn are (by the convolution theorem) equivalent to a sequence of low pass and highpass filtering, in arbitrary order. Naïvely, one might rearrange Equation 4.6 into said form, separating the low-cutting and high-cutting

σ-values vs. the uses of the $\sum$ notation for both sums and dependent types.

factors:

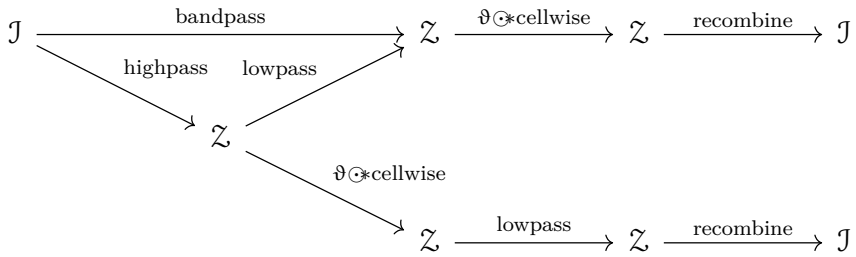
$$\begin{aligned}\Delta(\cdot,\sigma) &= \mathcal{FT}^{-1}\left(\mathbf{k}\mapsto \|\mathbf{k}\|^2\cdot\sigma^2\right)\star\mathcal{FT}^{-1}\left(\mathbf{k}\mapsto e^{-\frac{\|\mathbf{k}\|^2\cdot\sigma^2}{2}}\right)\star x\\ &= \mathcal{FT}^{-1}\left(\mathbf{k}\mapsto e^{-\frac{\|\mathbf{k}\|^2\cdot\sigma^2}{2}}\right)\star\mathcal{FT}^{-1}\left(\mathbf{k}\mapsto \|\mathbf{k}\|^2\cdot\sigma^2\right)\star x.\end{aligned} \tag{4.23}$$

Remark 21. *That simple decomposition is strictly speaking not well-founded since* $\mathbf{k}\mapsto\|\mathbf{k}\|^2\cdot\sigma^2$ *diverges by itself. A proper factorization into high-pass and low-pass filters is possible by adding cancelling factors to both of them, in a way that makes the low-cut limited to gains below unity, for example thus:*

$$\mathcal{FT}^{-1}\left(\mathbf{k}\mapsto e^{-\frac{\|\mathbf{k}\|^2\cdot\sigma^2}{2}}\cdot\left(1+\|\mathbf{k}\|^2\cdot\sigma^2\right)\right)\star\mathcal{FT}^{-1}\left(\mathbf{k}\mapsto\frac{\|\mathbf{k}\|^2\cdot\sigma^2}{1+\|\mathbf{k}\|^2\cdot\sigma^2}\right)\star x. \tag{4.24}$$

It is not necessary to use such a construction explicitly because in practice the filter is anyway not computed according to Equation 4.7, but rather as a finite difference of only lowpass filters that approximates Equation 4.5.

The trick to retaining smoothness of the ablated recomposition is, then, to reorder the computation such that the lowpass filtering is performed *after* the cell-wise weighing à la Equation 4.20.



(4.25)

In the case of a constant $c=\vartheta_i\ \forall i$, this reordering does not make a difference. In particular, for $\vartheta_i\equiv 1$, one still retains the exact original x after recombination. On the other hand, any edges introduced on the boundary between cells with different ϑ_i will get smoothened out by the subsequent filtering, and this at the appropriate scale (Figure 4.5 bottom).

The image reconstruction from the ablated scale space reconstruction is carried out as before. Lemma 8 still holds, since in the unablated case all the processing steps commute and the derivation via the differential remains equivalent to the split-up bandpass filter.

This trick can also be expressed in the style of Equation 4.21, and that corresponds more closely to how it is computed in practice:

$$\begin{aligned}(\vartheta\tilde{\odot}\Delta)(\mathbf{r},\sigma) &= \sum_i \vartheta_i\cdot\tilde{\chi}_i(\mathbf{r},\sigma)\cdot\Delta(\mathbf{r},\sigma)\\ &=: \sum_i \vartheta_i\cdot\zeta_i^{\Delta}(\mathbf{r},\sigma)\end{aligned} \tag{4.26}$$

with

$$\tilde{\chi}_i(\mathbf{r}, \sigma) := (\gamma_\sigma \star \chi_i(\cdot, \sigma))(\mathbf{r}). \tag{4.27}$$

I.e., instead of delaying the lowpass filtering in Δ until after the pointwise multiplication, one applies the same lowpass filter also to each cell's characteristic function (by itself). This is another way of preventing the cell boundary from imprinting hard edges into the recomposed image. $\tilde{\chi}$ is now not a partition in the sense of subsets, but rather a *partition of unity* as used in signal processing.

Remark 22. *Applying the lowpass filter to signal and characteristic function separately is not exactly equivalent to applying it after the pointwise multiplication of both. We see no inherent reason for preferring one over the other, though it might be interesting to investigate the differences between them more closely.*

The way Equation 4.26 is used in practice (details in the next section) is that all the ζ_i^Δ are precomputed and stored as a form of matrix. Then, applying a mask ϑ just amounts to applying the linear operator defined by this matrix to ϑ in its Euclidean vector form. Written compactly,

$$\begin{aligned} \zeta^\Delta &: \mathbb{R}^K \to \mathcal{L}^2(\mathcal{Z}, V) \\ \vartheta \tilde{\odot} \Delta &= \zeta^\Delta(\vartheta). \end{aligned} \tag{4.28}$$

Because this mapping is linear, it is in particular also differentiable and can thus easily be used with optimisation algorithms.

4.2 Implementing the feature decomposition

The previous section introduced a mathematical method for using the SIFT keypoints as ablatable features. But in its given form, it not only deals with images as signals in an infinite-dimensional space, but even adds an extra scale dimension as well as the Voronoi split. Continuous signals cannot be directly stored or processed digitally, but that is not fundamentally different from the situation for the original images.

What *is* different is that 2D images can still be quite easily handled in the common pixel / PCM representation, whereas for the scale-space expansion this is almost completely infeasible. More sophisticated discretisation schemes need to be used. Nevertheless, the homogeneous voxel view is a good starting point for the discussion.

4.2.1 *Discretization for* σ

Like the input images are given in the format of a homogeneous 2D pixel array, so could also the σ dimension be sampled uniformly. Δ would then be a 3D array (4D if counting colour channels. Although such sampling is usually understood in the Shannon-Nyquist sense (Section 3.2), this relies too heavily on the uniformity and it is here more useful to think of the discrete σ-slices as subintegrals of the integral used in Lemma 8, since that is what generates the recombined images.

This has in particular the advantage that the additional term $\Gamma(\mathbf{r}, R)$ as an (infinite) integral, so that all can be phrased as a single sum of $M + 1$ sub-integrals over intervals

Σ_ι that tile $\mathbb{R}^+$:

$$T_\iota := \int_{\Sigma_\iota} d\sigma \, \frac{\Delta(\mathbf{r}, \sigma)}{\sigma}, \tag{4.29}$$

$$\sum_\iota T_\iota = \int_0^\infty d\sigma \, \frac{\Delta(\mathbf{r}, \sigma)}{\sigma} = x. \tag{4.30}$$

Here, all the Σ_ι for $\iota < M$ are bounded intervals of the form $[\sigma_\iota, \sigma_{\iota+1}[$, whereas the final one $\Sigma_M = [R, \infty[$, so that.

$$T_M = \int_R^\infty d\sigma \, \frac{\Delta(\mathbf{r}, \sigma)}{\sigma} = -\left[\Gamma(\mathbf{r}, \sigma)\right]|_{\sigma=R}^{\infty} = \Gamma(\mathbf{r}, R). \tag{4.31}$$

The same partition is used also for storing the (semi-) discretized versions of $\tilde{\chi} \mapsto \tilde{Y}$ and $\zeta^\Delta \mapsto Z^\Delta$, with the Voronoi split as discussed in Section 4.1.3C. The latter satisfies

$$Z^\Delta_{i,\iota} = \int_{\Sigma_\iota} d\sigma \, \frac{\zeta_i^\Delta(\mathbf{r}, \sigma)}{\sigma} \tag{4.32}$$

and is computed, analogously to Equation 4.26, from

$$\left((\vartheta \tilde{\circledast} T)_\iota\right)(\mathbf{r}) = \sum_i \vartheta_i \cdot \tilde{Y}_{i,\iota}(\mathbf{r}) \cdot T_\iota(\mathbf{r}). \tag{4.33}$$

$\tilde{Y}$ itself is prepared directly in the discretised form, by first computing the non-smoothed form Y from voxel-wise cell-membership queries, and then convolving each $Y_{i,\iota}$ with a Gaussian kernel.

In this setting, combined with also a discretisation of the spatial dimensions (not explicitly shown here; standard pixel/PCM basis), the decoder D_{SIFT} boils down to the matrix multiplication of Z with ϑ, followed by summation over the index of the discretized σ dimension. In the style of Equation 4.14,

$$D^{\Delta,\kappa}_{\text{SIFT}}(\vartheta) = \sum_\iota Z^\Delta_{\cdot,\iota}(\vartheta). \tag{4.34}$$

Thanks to linearity, this is the same as

$$D^{\Delta,\kappa}_{\text{SIFT}} = \sum_\iota Z^\Delta_{\cdot,\iota}; \tag{4.35}$$

intuitively it should be more efficient to pre-sum this, but that would interfere with the optimisations below.

As already said, using a homogeneous grid for the entire scale space (even the R-limited version) is not actually feasible, so that alternatives are developed in the next sections. To wit: homogeneously covering the σ-dimension space for a 500×500 image would require another array axis for the $M \approx 500$ scale-layers. If the image contains 1000 keypoints and correspondingly many $\check{\Delta}_i$, that requires a total of 125 billion voxels in Z^Δ, or 1.4 terabytes if using RGB and single-precision floats. Machines with capability

to handle such amounts of data exist, but they are expensive and impractical to use for a single image of not even high resolution.

That problem is not new to this application, and it can be solved with a combination of standard techniques and purpose-crafted ones.

4.2.2 *Nonuniform grid*

The first and most straightforward memory optimisation is to sample the σ axis logarithmically, in other words have the intervals of each layer grow exponentially:

$$\begin{aligned} \Sigma_l &= [\sigma_l, \sigma_{l+1}] \\ \sigma_l &= 2^{\frac{l}{\rho_{8ve}}}, \end{aligned} \tag{4.36}$$

where the parameter ρ_{8ve} determines how many scales are sampled per octave (i.e. doubling of scale).This is reasonable because filtering with e.g. $\sigma = 30$ and $\sigma = 31$ gives nearly the same result (unlike e.g. $\sigma = 3$ vs $\sigma = 4$), and storing both of them independently would be mostly redundant.[8] Logarithmic sampling of σ is standard practice and used internally by most, if not all, implementations of scale-space algorithms. This allows in practice reducing the number of layers by a factor of ca. 10–20, which still leaves memory need at around 100 gigabyte.

A related technique is to also reduce the spatial resolution as σ increases. This is even more intuitive, since low-pass filtering directly removes high-frequency components. Ignoring the very low amplitude remainders after the exponential cutoff, this means the bandwidth is not lower therefore allows downsampling without loss of information.

The standard SIFT implementation [60] achieves both the logarithmic σ and the downsampling in a way that is highly efficient on CPUs. It first calculates the sequence of *low*pass filtered images, starting at the unfiltered, full resolution one, and compute a small number (3–6) of versions filtered with small kernels, specifically only within one "octave" of σ values – i.e. up to a doubling. Then it decimates the most filtered of these versions to half the resolution, and uses this to compute the next octave of σ values. Since this happens at the lower resolution, again only small kernels are used, which makes the computation cheap.

There are two reasons why this cannot be used for the method proposed here:

- The decimation at the octave is not quite lossless: after the Gaussian filters there is still some content left at frequencies above the new Nyquist frequency. Discarding those may be harmless when SIFT is only used to find the keypoints, but it would prevent the original from being exactly reconstructed.

- The sequential resampling is ill-suited for GPU-parallelization and differentiable computation, both of which is necessary when using the SIFT feature basis for saliency purposes.

On the plus side, the availability of GPUs means that at least the speed aspect of using smaller filters in the decimated versions is less crucial now than it was in 1998.

[8] This argument arises in essence from the fact that σ appears in an exponential in $\mathcal{FT}(\Delta(\cdot, \sigma))$; see Equation 4.6.

Resolution	σ
$h \times w$	$]0, \frac{9}{2}]$
$\lfloor \frac{h}{2} \rfloor \times \lfloor \frac{w}{2} \rfloor$	$]\frac{9}{2}, 9]$
$\lfloor \frac{h}{4} \rfloor \times \lfloor \frac{w}{4} \rfloor$	$]9, \infty[$.

Table 4.1: The chosen default settings of the resolutions/ranges in scale space for an image of resolution $h \times w$. All the σ values are measured relative to the pixel size of the original image.

This suggests using a compromise: the high-σ layers are stored in a downsampled form, but not in the sense of sequential decimation at every octave but rather in a fixed and conservative downsampling (using a standard resizing routine with bilinear interpolation) of several of the layers. Table 4.1 shows concrete parameters that were found useful.

This is a compromise in which the various disadvantages are largely avoided: the decimation artifacts are kept very small due to the Gaussian filters' exponential HF rejection, memory usage is significantly reduced since most of the layers are at least somewhat downsampled, and there is no need for sequential resampling or filtering. Going to even lower resolutions would have diminishing saving returns and only make worse use of the GPU capabilities. The exact σ ranges used are uncritical, so long as it is ensured the resolution is not decimated by more than the blur size (nor close to it). With the ranges in Table 4.1, the minimum σ for each sub-resolution level corresponds to 2.25 pixel sizes in that resolution.

This filtering also means that the resampling algorithm is uncritical, with high-frequency artifacts being suppressed afterwards. Similarly, the filters smoothing over the boundaries of the Voronoi cells allows cell-assignment to be carried out in a simple/efficient manner (nearest keypoint to the centroid of each voxel) without concerns that aliasing will have strong influence on the results.

The grid defined this way brings the memory consumption down enough so low-resolution images can just barely be de- and recomposed on consumer-grade hardware (ca. 10 gigabytes). It is still too much for practical saliency use (where the classifier needs to be kept in memory too, with backpropagation records for batches of inputs, and the input resolution can be higher).

The logarithmic sampling is also the reason[9] for the so-far mysterios factor $-\sigma$ in the formula for Δ. This does not need to be multiplied explicitly, because it arises from the discrete difference in the logarithmic sampling.

Lemma 9. *In the limit of $\rho_{8ve} \to \infty$, the discrete difference of the Gaussian lowpass-filtered signals Γ (with a suitable factor to avoid vanishing difference) approaches the differential of Gaussians Δ (Equation 4.5).*

Proof. Expand the logarithmic sampling,

$$\sigma_{l+1} = 2^{\frac{l+1}{\rho_{8ve}}} = \sigma_l \cdot 2^{\frac{1}{\rho_{8ve}}}.$$

[9]Regardless of historical reason, the factor also simply benefitial for getting suitable minima and maxima as the SIFT keypoints.

Because we consider the limit of large ρ_{8ve}, we can use the first Taylor terms

$$2^{\frac{1}{\rho_{8ve}}} = 1 + \frac{1}{\rho_{8ve}} + \mathcal{O}\left(\frac{1}{(\rho_{8ve})^2}\right),$$

so, choosing the factor ρ_{8ve} itself for scaling the differences,

$$\begin{aligned}
\lim_{\rho_{8ve}\to\infty} \left(\rho_{8ve}\cdot\left(\Gamma\left(\mathbf{r},\sigma_l\right) - \Gamma\left(\mathbf{r},\sigma_{l+1}\right)\right)\right) &= \lim_{\rho_{8ve}\to\infty} \left(\rho_{8ve}\cdot\left(\Gamma\left(\mathbf{r},\sigma_l\right) - \Gamma\left(\mathbf{r},\sigma_l + \frac{\sigma_l}{\rho_{8ve}}\right)\right)\right) \\
&= -\lim_{h\to 0} \left(\rho_{8ve}\cdot\left(\Gamma\left(\mathbf{r},\sigma_l\right) - \Gamma\left(\mathbf{r},\sigma_l + \sigma_l\cdot h\right)\right)\right) \\
&= -\sigma_l\cdot\frac{\partial\Gamma\left(\mathbf{r},\sigma\right)}{\partial\sigma|_{\sigma=\sigma_l}} \\
&= \Delta\left(\mathbf{r},\sigma_l\right).
\end{aligned}$$

□

4.2.3 *Sparsity*

Apart from the excessive resolution, a naïve 4D-array sampling (scale space times keypoint cells) also has the inefficiency that every cell is stored on the complete domain $\mathcal{Z}$, although from the definition via the Voronoi tesselation it is evident that each of them is confined to a rather small region around its corresponding keypoint. Particularly for the decoding as per Section 4.1.3B/Equation 4.20, this means that most of the entries of ζ as a matrix are zeroes. There are standard routines for handling of such sparse matrices, which also occur in many other applications such as finite elements analysis.

Remark 23. *Arguably, sparse matrices are a symptom indicating that an algorithm should not be using matrices at all but rather a direct computation of the linear function. For a CPU implementation, this would likely be the best choice here, but testing the Voronoi-cell membership directly on a GPU is considerably more difficult. It was thus most pragmatic to compute the sparse matrix from the Voronoi cells on the CPU, and use that sparse matrix for the linear function on the GPU.*

The problem is that the post-filtered version (Equation 4.26) does not have this exact sparsity: the $\tilde{\chi}_i$ are in most of the domain *close to* zero, but nowhere exactly zero. One way this could be addressed is by using a filter with compactly supported impulse response instead of the Gaussian in Equation 4.27. This would probably work well in practice, but it requires choosing such a filter; the advantage of the Gaussian[10] is that its choice is obvious since it is already used for the signal as well.

Another option, which was chosen here, is to first compute $\tilde{\chi}$ with Gaussian lowpass, and then truncate entries below a threshold $\mathfrak{s}$ to zero. This causes two new problems though.[11]

[10] Another advantage is that Gaussian filtering has efficient implementations available, but even generic convolution filters could probably be used without creating a performance bottleneck in the intended path-saliency application, since the $\tilde{\chi}_i$ are anyway only computed once, stored (as $^{\mathfrak{s}}\tilde{\chi}_i$) and then reused in the computationally heavy path-optimisation steps. The bottleneck is GPU memory rather than -time.

[11] It is not clear whether these problems would have really mattered in practice so long as $\mathfrak{s}$ is small

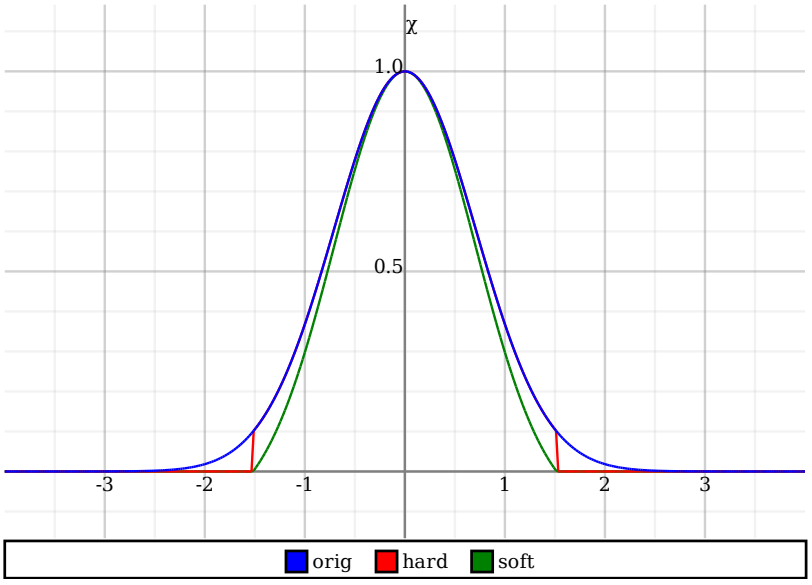


Fig. 4.6: How a hard cutoff or a soft one (Equation 4.37 with $\mathfrak{s} = 0.1$) makes a Gaussian kernel compactly supported. In practice, the threshold is typically chosen (much) lower than 0.1, making both tweaks less perceptible.

4.2.3A *Cutoff edges*

The main reason for filtering $\tilde{\chi}$ in the first place was to avoid the discontinuity of the Voronoi-cell boundaries from appearing in the recomposed image. Filtering blurred these edges away, but any cutoff would introduce new discontinuities, albeit with smaller value-jump.

This can be avoided by applying the cutoff not as a hard threshold, but instead a continuous tweak that moves small values progressively closer to zero and eventually to exactly zero. Such tweaks are known in the design of window functions for signal processing, which is also related to the design of compact-support filter kernels. Here, the following is chosen:

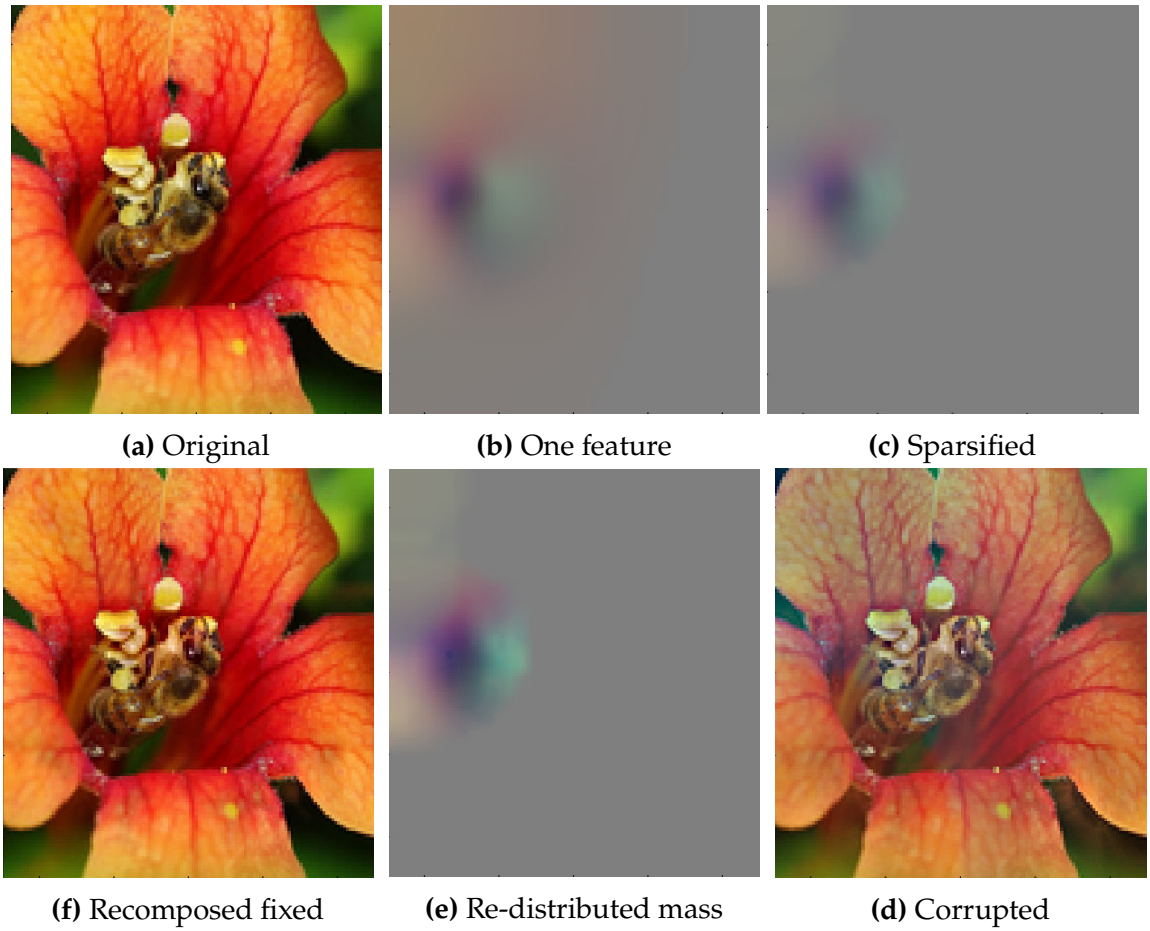
$$\overset{\mathfrak{s}}{\tilde{\chi}}_i(\mathbf{r},\sigma) := \begin{cases} \frac{\tilde{\chi}_i(\mathbf{r},\sigma)-\mathfrak{s}}{1-\mathfrak{s}} & \text{if } \tilde{\chi}_i(\mathbf{r},\sigma) > \mathfrak{s}, \\ 0 & \text{else.} \end{cases} \tag{4.37}$$

Notice that this particular tweak only guarantees continuity, but no higher-order regularity. Also notice that it has 1 as the only fixpoint (apart from 0). That means it keeps the parts that are fully inside one Voronoi cell at 100%, but it modifies everything at $\tilde{\chi}_i(\mathbf{r},\sigma) < 1$, including even parts with relatively high amplitude that would have retained their strength with the hard cutoff (Figure 4.6). This could have been avoided with more sophisticated formulas than the simple affine stretch in Equation 4.37, but it was decided against this to avoid complexity (which would again raise questions about ambiguous choice that might influence classification later on) and because it would have not prevented the fundamental issue of the next section, whose solution also undoes part of this amplitude-sagging.

enough. Possibly this section is over-engineered, but it was considered important to minimise the influence of a pure computation-optimisation like sparsity on the actual behaviour of the method.

4.2.3B *(Non-) partition of unity*

Since Equation 4.37 systematically reduces the values of $\tilde{\chi}$, the resulting $\overset{s}{\tilde{\chi}}_i$ do not sum to 1 anymore. As a consequence, using this as in Equation 4.26 would not give back the original x even when $\vartheta_i \equiv 1$. This would be quite unacceptable for saliency purposes, since one wishes after all to make statements about interventions around x.



(a) Original **(b)** One feature **(c)** Sparsified

(f) Recomposed fixed **(e)** Re-distributed mass **(d)** Corrupted

Fig. 4.7: How (exaggerated) sparsity cutoff prevents exact original image from being reconstructible, and how this can be fixed again. (a): original image; (b): reduced to the feature corresponding to a single SIFT-keypoint; (c): that feature with an aggressive sparsity cutoff applied; (d): how the image would get (not-) reconstructed from such trimmed features; (e): the example feature after mass-redistribution (Equation 4.39); (f): near-perfect original-image reconstruction using the redistributed masses. — Image: ImageNet

One solution is to re-distribute the "missing mass"

$$\tilde{w}(\mathbf{r},\sigma) := \sum_i \left(\tilde{\chi}_i(\mathbf{r},\sigma) - \overset{s}{\tilde{\chi}}_i(\mathbf{r},\sigma) \right) = 1 - \tilde{m}(\mathbf{r},\sigma) \tag{4.38}$$

$$\tilde{m}(\mathbf{r},\sigma) := \sum_i \overset{s}{\tilde{\chi}}_i(\mathbf{r},\sigma)$$

to other $\overset{s}{\tilde{\chi}}_j$. This way, the complete sum will again be 1. It only requires some care to

ensure that

1. The re-distribution should keep the locality of ${}^{s}\tilde{\chi}_j$. It must be avoided that a decision is attributed to ϑ_j but actually originates in a part of the scale space far away from κ_j, which would be highly misleading in a saliency result.

2. It should not distribute back to parts that were already truncated to zero. That would undo the sparsity, which was the whole point of truncating in the first place.

Both can be achieved by making the redistribution dependent on the value ${}^{s}\tilde{\chi}_j(\mathbf{r}, \sigma)$ itself:

$$\overset{s,1}{\tilde{\chi}}_j(\mathbf{r}, \sigma) := \overset{s}{\tilde{\chi}}_j(\mathbf{r}, \sigma) + \frac{{}^{s}\tilde{\chi}_j(\mathbf{r}, \sigma) \cdot \tilde{w}(\mathbf{r}, \sigma)}{\tilde{m}(\mathbf{r}, \sigma)}. \tag{4.39}$$

Intuitively, that means the lack of mass at each point in scale space is taken over by the features which already have the most significance at that point anyway. This way, the relative order of feature-importance is preserved for each point too.

Lemma 10. *If* $\tilde{m}(\mathbf{r}, \sigma) > 0$ *everywhere, then*

$$\sum_j \overset{s,1}{\tilde{\chi}}_j(\mathbf{r}, \sigma) = 1.$$

It follows that ${}^{s}\tilde{\chi}$ *gives rise to an exact reconstruction of the original image.*

Proof.

$$\begin{aligned}
\sum_j \overset{s,1}{\tilde{\chi}}_j(\mathbf{r}, \sigma) &= \sum_j \left(\overset{s}{\tilde{\chi}}_j(\mathbf{r}, \sigma) + \frac{{}^{s}\tilde{\chi}_j(\mathbf{r}, \sigma) \cdot \tilde{w}(\mathbf{r}, \sigma)}{\tilde{m}(\mathbf{r}, \sigma)} \right) \\
&= \sum_j \overset{s}{\tilde{\chi}}_j(\mathbf{r}, \sigma) \cdot \left(1 + \frac{1 - \tilde{m}(\mathbf{r}, \sigma)}{\tilde{m}(\mathbf{r}, \sigma)} \right) \\
&= \sum_j \overset{s}{\tilde{\chi}}_j(\mathbf{r}, \sigma) \cdot \frac{1}{\tilde{m}(\mathbf{r}, \sigma)} \\
&= \left(\sum_j \overset{s}{\tilde{\chi}}_j(\mathbf{r}, \sigma) \right) \cdot \left(\sum_i \overset{s}{\tilde{\chi}}_i(\mathbf{r}, \sigma) \right)^{-1} \\
&= 1.
\end{aligned}$$

□

The condition $\tilde{m}(\mathbf{r}, \sigma) > 0$ is not strictly speaking guaranteed, but should be fulfilled with any reasonable choice of s. The converse would mean the sparsification was so aggressive as to completely remove parts of the scale space from the representation, in which case it is not surprising that a reconstruction is not possible anymore. This is the case in Figure 4.7, though the reconstruction is still very good, much better than without the mass redistribution.

The implementation contains a guard for the zero case, replacing the denominator with $\max\left(\tilde{m}\left(\mathbf{r}, \sigma\right), 10^{-3}\right)$, which has the effect that parts with very low combined mass are simply omitted the reconstructions. In practice this then still gives close approximations to x even with extreme sparsity thresholds.

With reasonable s, recomposing with ${}^{s,1}\tilde{\chi}$ gives results almost indistinguishable from the fully-smooth $\tilde{\chi}$ but takes much less memory. Because the weight re-distribution contains the factor ${}^{s}\tilde{\chi}_j$, it preserves zeroes so that ${}^{s,1}\tilde{\chi}$ is no less sparse than ${}^{s}\tilde{\chi}$.

4.2.4 *Differentiable recomposition*

Equation 4.34 gives the linear-map form of the decoder that reconstructs full images. This would be a matrix if all the σ-layers had the same spatial resolution.

Each $Z^{\Delta}_{i,l}$ is an image, but their resolutions differ (Section 4.2.2). There are still (by design) many layers that share resolution though: if $\bar{\Sigma}^{\rho}$ is the interval of σ values associated with resolution ρ, then $\zeta^{\Delta|_{\sigma\in\bar{\Sigma}^{\rho}}}$ has a proper matrix representation. Therefore, when calling L^{ρ} the set of indices such that $\bigcup_{l\in L^{\rho}} \Sigma_l = \bar{\Sigma}^{\rho}$,

$$\bar{Z}^{\Delta,\rho}_{l} := \sum_{l\in L^{\rho}} Z^{\Delta}_{\cdot,l} \tag{4.40}$$

is a well-formed sum and generates a (still sparse) matrix representing the mapping of ϑ to the whole frequency range of the correspondingly weight-recomposed image which can be efficiently stored at resolution ρ. Putting these together after each matrix multiplication still requires resampling, but only for a small number of images, which has therefore little performance impact. The complete operation, as implemented, is thus

$$D^{\Delta,\kappa}_{\mathrm{SIFT}}\left(\vartheta\right) = \sum_{\rho} \mathfrak{r}_{\rho\mapsto(h,w)}\left(\bar{Z}^{\Delta,\rho}_{l}\left(\vartheta\right)\right). \tag{4.41}$$

The resampling operator $\mathfrak{r}$ uses again simple and efficient bilinear interpolation, sufficient since the lower-resolution layers have ample Gaussian filtering applied to them.

Remark 24. *If the decomposition were to be used with much higher-resolution images than the ones from the considered datasets, it might become necessary to downsample more aggressively to still stay with the memory limitations, and that would mean more care needs to be taken with respect to aliasing and other artifacts. It would require substantial further efforts to get this right. For the present work, it was unnecessary since the sparsity and downsampling settings already cause the cost of the SIFT recomposition to be much less than that of the subsequent network evaluations, making further performance optimisation of this component largely futile.*

Because the whole decoder is linear, its derivative is the same as the operation itself. Practically speaking, what is needed is the transpose of the Jacobi matrix, to carry out reverse-mode automatic differentiation together with the classifier network that receives $D^{\Delta,\kappa}_{\mathrm{SIFT}}\left(\vartheta\right)$ as its input. All of this is handled automatically by the PyTorch framework [29], given the $\bar{Z}^{\Delta,\rho}_{l}$ as sparse matrices as well as the resampling specifications. Typically a whole batch of applications $D^{\Delta,\kappa}_{\mathrm{SIFT}}\left(\vartheta^{b}\right)$ is computed, which is equivalent to treating ϑ also as a matrix and computing matrix-matrix products in Equation 4.41.

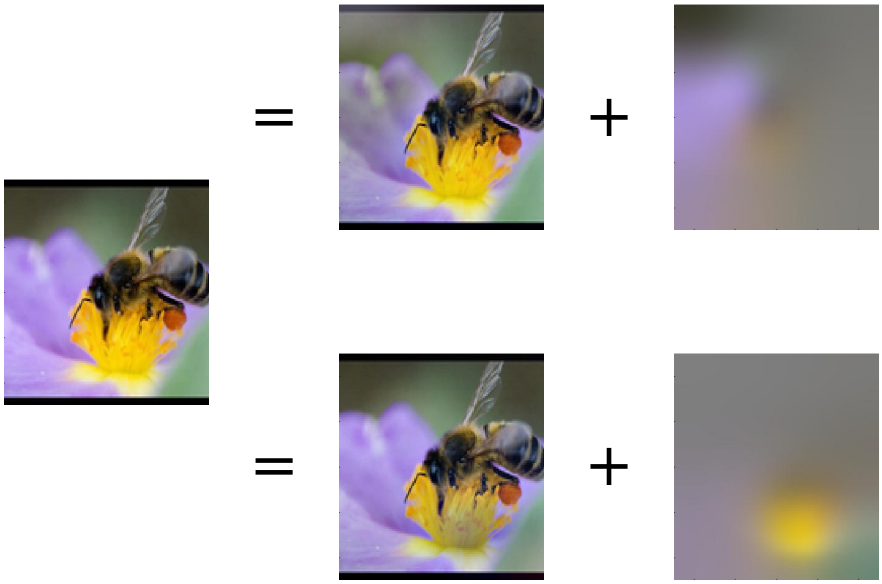


Fig. 4.8: The two SIFT features for the highest-σ keypoints in this image essentially constitute the background colours, which could also be considered as a neutral baseline. Photo: ImageNet [82]

4.3 Saliency use

In the form of Equation 4.41, the SIFT recomposition can directly be used in the input stage of most interventional saliency methods (Section 1.2.3). This involves an evaluation of the form

$$\mathsf{F}\left(\mathrm{D}^{\Delta,\kappa}_{\mathrm{SIFT}}\left(\vartheta\right)\right),$$

in other words, one is dealing with a simple function composition of type

$$\mathsf{F}\circ\mathrm{D}^{\Delta,\kappa}_{\mathrm{SIFT}}\colon \mathbb{R}^{K}\to\mathcal{R}. \tag{4.42}$$

The main changes compared to chapter 1 are that ϑ does not represent a spatial mask anymore, that there is no explicit baseline, and that the space of masks $\mathcal{M}$ is not fixed anymore.

4.3.1 *Implicit baseline*

The case $\vartheta_i \equiv 0$ corresponds (by linearity) also to an all-0 input to the classifier. This may be considered problematic, especially when thinking of $\mathcal{R}$ as only an affine space, not a vector space. Concretely, zero could just as legitimately represent black or white or (in the reference implementation) middle grey. Unlike with spatial masking, this is not much of an issue for the SIFT version though, since there are only very few keypoints in the lowest-frequency regions of $\mathcal{Z}$. These effectively encapsulate the background colour as one or a few features of their own right (Figure 4.8), which in itself is interesting and useful since these features can well have significance to the classification. It would

also be possible to fix the layers above a certain σ as constant and this way have again a blur baseline like used in chapter 2. On the other hand, not having to choose a baseline even in the sense of a filter cutoff is also desirable.

For uniformity of formulation and code reuse, the implementation of the following does employ explicit dummy target and baseline values in form of all-one and all-zero arrays, which serve no purpose but to be used in the familiar masked-interpolation construct, which then yields the mask verbatim. This is finally passed through $D^{\Delta,\kappa}_{\mathrm{SIFT}}$, which is what introduces the proper x-targeting.

$$D^{\Delta,\kappa}_{\mathrm{SIFT}}\left([\,{}^{1}_{0}\,]_{\vartheta}\right) = D^{\Delta,\kappa}_{\mathrm{SIFT}}\left(\vartheta\right) \tag{4.43}$$

4.3.2 *Interventions*

The SIFT decomposition should be usable for saliency methods akin to RISE [73], Meaningful Perturbation [28] and Ablation Paths (chapter 2). It is perhaps particularly interesting for RISE, since the discrete nature of the features means the random sampling is much more straightforward than for a spatial representation where it required very ad hoc choices of subsampling and interpolation.

So far, only the Ablation Path version has been implemented and tested, since that method is anyway part of this thesis. The dummy target and baseline are multiplied with the masks within the ablation path

$$\varphi\colon\ [0,1] \to \mathbb{R}^{K}. \tag{4.44}$$

The fact that $\mathbb{R}^K$ is not a fixed type like $\mathcal{M}$ in chapter 2 might at first be considered an aspect in favour of a dynamic language like Python, but actually this makes it rather more complicated. In particular, initializing a path requires knowledge of K as the dimension of each $\varphi\,(t)$. A static language with rank-2 polymorphism could handle this automatically, but in a dynamic language the dependent type in Equation 4.10 needs to be manually unwound. This amounts to first evaluating $E_{\mathrm{SIFT}}\,(x)$ separately, which provides both the information for D_{SIFT} and for path initialization. Then D_{SIFT} is treated as a curried function to obtain the partially evaluated form $D^{\Delta,\kappa}_{\mathrm{SIFT}}$, and also the initial affine path generated. Only with all that in place is the optimisation started.

The optimisation itself works much like in chapter 2, except that the filtering steps which relied on the function-space nature of $\mathcal{M}$ do not make sense anymore – but neither are they necessary in the way as before, since the SIFT basis enforces to a considerable extent regularity by itself. Notice how in Figure 4.9 no artificial graininess or edges are visible, despite the quite narrowly confined selection of information patches of the target-class giraffe.

4.3.3 *Geometry*

One aspect that nevertheless remains subtle is that of a metric on the mask space. Since $\mathbb{R}^K$ is legitimately a finite-dimensional, Euclidean space, the standard isotropic $\mathcal{L}^2$ metric is a plausible enough choice, arguably more so than in the pixel case. On the other hand, the SIFT features are not a priori equiponderous: the main condition for a keypoint is just to be local extremum in $\mathcal{Z}$, but the prominence and isolation of these

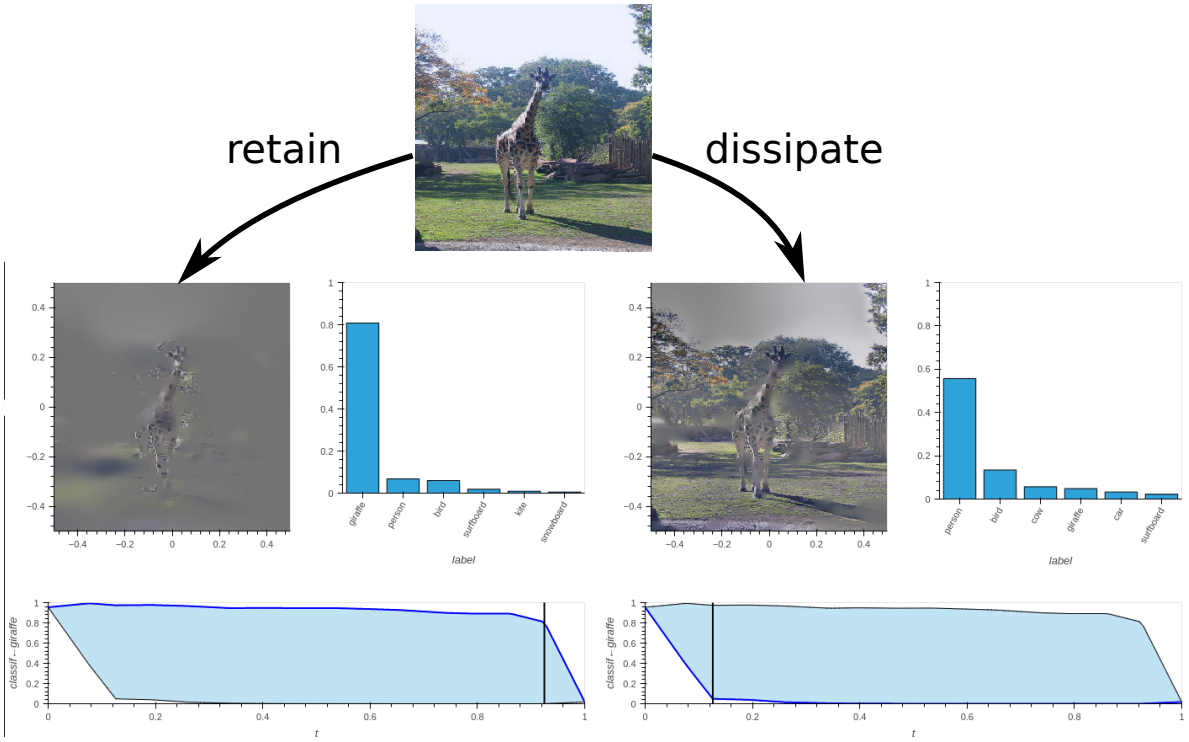


Fig. 4.9: Example slices through a straddle-optimised ablation path using the SIFT feature basis. Photo: COCO [56]; classifier: ResNet [92].

extrema can vary, apart from the ones too minor for the cutoff criterion. It is reasonable to weigh them accordingly in the metric. Such weight can be obtained by evaluating $D^{\Delta,\kappa}_{\mathrm{SIFT}}$ once for each feature in isolation (i.e. for the canonical basis inputs ϑ^j satisfying $\left(\vartheta^j\right)_i = \delta_{i,j}$).

$$W_j := \left\| D^{\Delta,\kappa}_{\mathrm{SIFT}}\left(\vartheta^j\right) \right\| . \tag{4.45}$$

Any choice of norm could be used here; so far tried was the usual $\mathcal{L}^2$ one.

For the Ablation Path method, the metric in which these weights can be used has an effect upon both the time-normalisation and the computation of the gradient from the differential. Empirically, this has rather little influence on the actual results though, in the experiments carried out so far; Figure 4.9 and Figure 4.10 show almost identical slices. The only effect that seems to cause a small difference is that spatially extended low frequency features tend to have higher mass, which means that turning only few such features on/off can still correspond to a fairly large time-step in the ablation path. This may be the reason for the slightly lower ablation-path scores.

4.3.4 *Interpretation and comparison*

While the SIFT de- / recomposition is inherently interpretable, it is not quite as obvious what is going on as when looking at purely spatial heatmaps. The easiest way to inspect the results is by looking at small changes and how they affect the classification, for which Ablation Paths provide an excellent framework. For example, in Figure 4.9 one can immediately see some aspects that would not have been obvious in the pixel-based version, including that the colours are not very relevant, that the distribution of fur-patterns and shades on the giraffe is sufficient to classify as such, but also that small

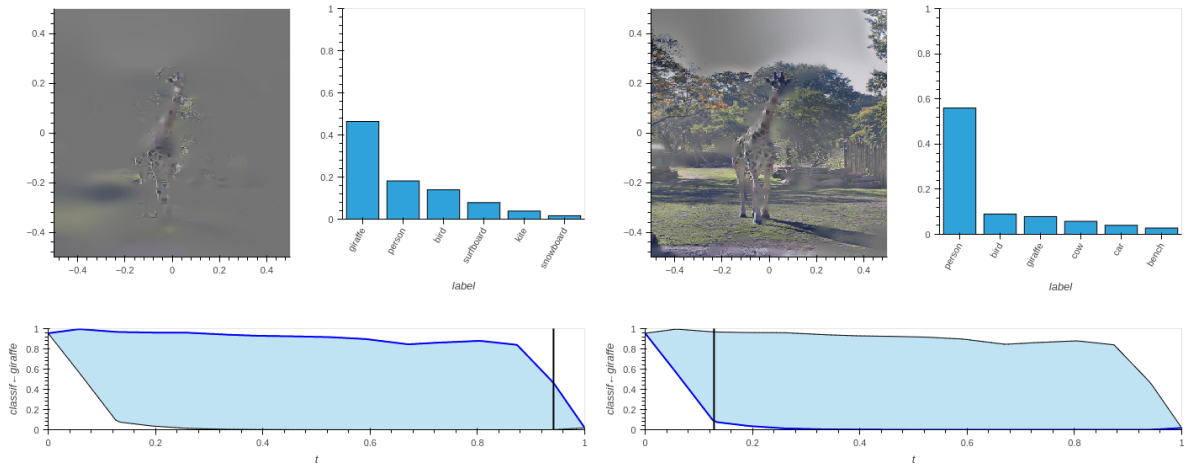


Fig. 4.10: Like Figure 4.9, but with $\mathcal{L}^2$-weights in the metric on $\mathcal{M}$.

changes in the lighting conditions such as the darker region on the giraffe's breast are sufficient to dissipate this classification. This would not have been possible to notice with a purely spatial basis.

Such observations can not always be made reliably; indeed it is still quite often the case that strong classification changes occur from inoccuous changes that seem nonsensical to a human. This may again be considered as adversarial effects, however due to the principled construction of the SIFT features and avoidance of information-adding it could then be argued that these are more due to aberrant classifier extrapolation than an unrealistic intervention space.

4.3.4A *Heatmaps*

At any rate, what is not directly possible anymore is to compare the SIFT-based saliency method with others, like in the pointing game [118] as used by Fong, Patrick, and Vedaldi [27]. This is fully based on a spatial heatmap. It is however still possible to extract only spatial aspects of the SIFT saliency, although this betrays in a sense the very point of using it. One way of doing that is to use the same keypoints and mask-amplitudes as for the intervention-modified classifier input, but with an artificial homogeneous scalar field in the scale space instead of a Gaussian-derived field Δ:

$$I^{\kappa}_{\mathrm{SIFT}}(\vartheta) := D^{C_{\mathcal{Z}},\kappa}_{\mathrm{SIFT}} \tag{4.46}$$

where $C_{\mathcal{Z}} : \mathcal{Z} \to \mathbb{R}$ is spatially symmetric, i.e.

$$C_{\mathcal{Z}}(\mathbf{r}, \sigma) = C(\sigma) \tag{4.47}$$

for some function C that may weigh different scale-length differently. Note that a field such as $C_{\mathcal{Z}}$ could *not* have arisen from the differential of Gaussians construction of an image, because that entails high-pass filtering and does therefore surpress constant contributions in all the σ-slices with $\sigma \ll R$ (as is the case for most of the $Z^{\Delta}_{i,l}$). Because of this, I does not benefit from the artifact avoidance implemented in Section 4.1.3C but has the Voronoi cell boundaries imprinted quite visibly (Figure 4.11).

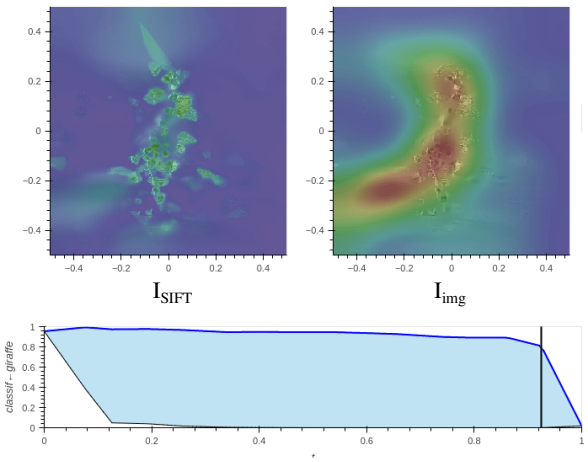


Fig. 4.11: Different heatmap extractions for the example in Figure 4.9.

An entirely different approach for generating heatmaps is to disregard the details of the feature generation entirely, and instead post-analyse the images $\mathrm{D}^{\Delta,\kappa}_{\mathrm{SIFT}}(\vartheta)$ as they are also fed to the classifier. There are many ways this could be done, a simple one being a local detection of amplitude of colour-fluctuations. This can be implemented as a rectifier sandwiched between a highpass- and lowpass filter:

$$\mathrm{I}^{\sigma_{\mathrm{LP}},\sigma_{\mathrm{HP}}}_{\mathrm{img}}(x) := \gamma_{\sigma_{\mathrm{LP}}} \star |x - \gamma_{\sigma_{\mathrm{HP}}} \star x| \tag{4.48}$$

where the rectifier $|\cdot|$ is understood pointwise.

This approach inherits all the regularity properties guaranteed by the image reconstruction technique, but its problem is that the result is strongly biased by the original image, favouring regions of high contrast over highly classifier-salient but low-contrast ones. For example, in Figure 4.11, the shadow of the tree in the background is quite prominent (lower left). Though this shadow does appear in the "last retaining" slice in Figure 4.9, it is hardly the most striking there. In this particular case this bias could have been avoided by selecting a lower σ_{HP} (so that relatively low-frequency features do not even reach the rectifier). That would however only exacerbate the problem in images that have localised high-frequency contrast.

4.3.4B *Pointing game*

Combining the above mask-to-heatmap extraction with an appropriate path-to-heatmap one from Section 2.7.2 allows the SIFT-Ablation-Path saliency to compete in the pointing game (even though this is somewhat unnatural, since the saliency result inhabits a much more complicated space than the ordinary 2D heatmaps provided by most other saliency methods). Table 4.2 shows some concrete results. We will not discuss these in great detail. Evident is that not as high scores are reached as with the pixel-based Ablation Path method, let alone the state-ef-the art scores. This is disappointing, but not too surprising since the method is not really designed to "point" in space.

Where it appears to have an advantage is in stability: although the methods adds even more parameters, we do not observe the pointing game scores to be as sensitive to their choice as in the pixel-based case. In particular it avoids the delicate interaction

Method	opt.cr	η	ζ_{sat}	Heatmap	Hm.red	*COCO14 Val* (All%/Diff%)
Contr.	$P_{\updownarrow}$	1.0	0.8	Voronoi	Θ_{clt}	32.5/18.9
Contr.	$P_{\updownarrow}$	2.0	0.8	Voronoi	Θ_{clt}	31.8/19.1
Contr.	$P_{\updownarrow}$	4.0	0.8	Voronoi	Θ_{clt}	35.5/22.0
Contr.	$P_{\updownarrow}$	4.0	0.8	$\sigma_{\text{LP}} = 8$, $\sigma_{\text{HP}} = 5$	Θ_{clt}	42.0/28.1
BndStr.	$P_{\uparrow\downarrow}$	4.0	0.8	Voronoi	Θ_{cav}	42.7/29.7

Table 4.2: Pointing games for the SIFT saliency.

between regularisation and saturation parameters: the optimisation take place simply in a Euclidean space without any further structure imposed on it, and all signal regularity is already determined when the SIFT decoder is setup. This decouples it from the saliency generation, making both aspects easier to inspect.

4.4 Discussion

We demonstrated that it is possible to use SIFT features as a per-image re-composable basis through which it is possible to apply interventions in a very controlled and theoretically understood manner. Although the method was developed as a tool for making better use of the Ablation Path method, many other applications are conceivable, including non-path saliency but also entirely unrelated signal processing. Indeed, the Ablation Path method may not even be a particularly good showcase for the SIFT decomposition, since the optimisation algorithm is known to have problems which may overshade the quality of the SIFT basis.

Notwithstanding, the combination of Ablation Paths and SIFT features does work, and does fit together well in some ways: the Ablation Paths (with an interactive tool) provide a useful setting in which to visually observe small SIFT changes (although they are, in spite of their elegance, still much less self-explanatory as pixels), and the SIFT method avoids one of the problems of the current Ablation Path method, namely the tug-of-war behaviour between constraints that are hard to reconcile with one another.

For future work it will be interesting to attempt making even better use of the mathematical properties of SIFT. Furthermore, different use cases for the method are inviting, in particular a scheme similar to RISE [73] which entirely avoids an iterative optimisation and interpolation; unlike in the pixel case where a somewhat dubious subresolution choice had to be made the SIFT feature could be directly turned on or off individually.

Part IV

FROM TRANSLATIONS TO ROTATIONS

CHAPTER 5

SYMMETRIES

The previous chapters were concerned with explaining black-box models. This represents the bulk of the research done for this thesis. But although some progress was made there, it was overshadowed by recurrence of mysterious results – inputs that a network apparently classifies correctly, but with saliency pointing to neither a human-reasonable object in the image nor anything indicative of a plausible dataset bias. To some extent this may still be due to insufficient stability of the representations and algorithms used. But the whole approach of black-box explanation is, if not outright fraught as argued by Rudin [80], then at least limited and unreliable. When a black box is all one has the pragmatism may require making this compromise to get any insights at all, but in the long run such explanations remain unsatisfactory.

The remainder of the thesis is therefore rather concerned with interpretability instead – not in the sense that it develops a fully interpretable alternative to black boxes image classifier, but in that it summarizes an aspect in which current models are already interpretable, identifies an application where this aspect is particularly significant, and then presents a new model designed specifically towards it. That aspect is *symmetry*.

This chapter provides a general overview, whereas chapter 6 introduces the specific application to which we present a symmetry-based solution in chapter 7.

5.1 The physical world

Most data of interest for machine learning arise in some way from measurements.[1] A measurement is a physical process that results in gain of knowledge about some real-world system. For many applications, a data scientist might not give much thought to these physical processes, but that does not mean they have no relevance upon the data and the ways one can learn from them.

Particulary for the case of photographic images, some aspects of the physics behind their generation were already discussed in Section 3.1. Many of them are application-specific, but there are also some physical principles that are relevant for nearly all applications. Symmetry may be the most important of these, turning up prominently in everything from fundamental physical theories to concrete engineering challenges. Historically, symmetries have often played a central role in the development of theories,

[1] There are certainly also counterexamples, perhaps the most important being (written) natural language. For these applications, symmetries may still be present in some way, but they are at least much less evident.

with Galilean invariance having paved the way for Newtonian mechanics, Lorentz invariance prepared special relativity, and particle physics being rooted in various gauge theories.The result that best typifies the importance of symmetries is Noether's theorem [69], which demonstrates that every[2] symmetry gives rise to a conserved quantity. This thesis is not primarily concerned with the physical phenomena themselves, and therefore does not go into any of these details.

What it is concerned with instead are machine learning systems dealing with such real-world data. It makes some evident sense for such a system to be invariant under a symmetry of the experimental setup, but this is actually a somewhat problematic notion, if not outright meaningless. If one would understand e.g. translational symmetry (translation being part of Galilean transformations) in the sense that both the camera and all visible objects (and light sources) are moved by the same displacement, then this leaves already the measurement data (i.e., the photos) completely invariant, and a classifier on these photos could not possibly be anything but invariant (short of being *indeterministic*). If on the other hand one understands it in the sense of translating only the camera or only select objects in the scene, then this can in general *not* leave the classification invariant: if the translation is such that the object previously in center and focus completely leaves the image frame, then it would be odd to expect the classifier to still classify the image based on this now-invisible object. If anything, it would demonstrate a strong bias in the background or other parasitic features of the scene. Anyways, physics is not symmetric under translation of only part of a system to begin with.

What is in practice relevant are instead transformations such as changing the camera angle slightly, changing the brightness at which a scene is lit, zoom, etc..

5.2 Mathematical formulation

5.2.1 *Basic concepts*

In the most general terms, a symmetry is the property of a function $f\colon A \to B$ to change when its inputs are modified by a transformation $\alpha_g\colon A \to A$ as if the result was modified by another transformation $\beta_g\colon B \to B$, i.e.

$$f(\alpha_g(x)) = \beta_g(f(x)) \quad \forall x \in A, \tag{5.1}$$

or compactly $f \circ \alpha_g = \beta_g \circ f$, or in category-theory notation that the diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \downarrow{\scriptstyle \alpha_g} & & \downarrow{\scriptstyle \beta_g} \\ A & \xrightarrow{f} & B \end{array} \tag{5.2}$$

commutes. Two important special cases are

Definition 4. *When* $\beta_g = \mathrm{id}_B$, *then* f *is said to be* invariant under the action α_g.

[2]The theorem only applies to differentiable symmetries. Every symmetry discussed herein is differentiable / Lie-group-action, though that is not to say that discrete ones are without importance.

Definition 5. *When* $A = B$ *and* $\beta_g = \alpha_g$*, then* f *is said to be* equivariant under the action α_g.

Usually, one considers not only a single pair of transformations (α_g, β_g) but a whole family. Indeed, from symmetry of f under α_g it follows that $f \circ \alpha_g$ has the same property, and therefore f is also symmetric under $\alpha_g \circ \alpha_g$ as well as, by induction, any higher powers α_g^n. This fact is deeply ingrained to the mathematical formalism of symmetry, but it also means that notions like "symmetric only under sufficiently small translations" do not really fit the framework, since one could always extrapolate from this to larger transformations.

Two possibilities to avoid this conflict could be

- Restricting α_g to a subdomain of A, such that $f \circ \alpha_g$ is less capable for further transformation than f.
- Allowing for a certain small discrepancy between $f(\alpha_g(x))$ and $\beta_g(f(x))$, which would increase through compounding of transformations.

Neither of these is explicitly formalised here, but the ideas were influential on the development of the technique in Section 7.3.

The standard treatment is to abstract the composition of transformations into the acting elements g:

$$\alpha_{g \cdot h} = \alpha_g \circ \alpha_h \tag{5.3}$$

where g and h inhabit the monoid G with identity e_G such that $\alpha_{e_G} = \mathrm{id}_A$, and the associative product $\cdot$. Usually one assumes also invertibility, which makes G a *group*. In that case, equivariance can also be expressed as

$$\alpha_{g^{-1}} \circ f \circ \alpha_g = f. \tag{5.4}$$

Remark 25. *Invertibility is natural for many symmetries; for example a translation is inverted by simply translating in the opposite direction. The term "group" is sometimes used also more generally for anything with an action. But some examples like the renormalization group in physics are in fact only monoids, since their action does not preserve all information and can thus not be exactly invertible.*

Translations have in fact a much stronger structure: they form a vector space, with the group product being addition of displacements and additionally the scalar product that changes the magnitude of displacement. Such a vector-space structure is not very common for symmetry groups, but an only slightly weaker one is: that G is a manifold and the product continuous in both arguments, in which case G is called a Lie group [55]. This provides at least a way to *discuss* "smallness" of actions: as a manifold, G has a tangent space $\mathfrak{g} := T_{e_G}(G)$ around its identity called the Lie algebra. It contains in a sense infinitesimally small transformations.

Much more could be said about the theory of Lie groups, but it has beyond these basic concepts not found use in the research presented in the following, and therefore will not be layed out further here. We defer to Munthe-Kaas [68] for a treatise on their use in related applications. That is not to imply that the deeper subjects of the theory could not be useful for the goals pursued here, which is left for future work.

5.2.2 *Actions on function spaces*

The spaces of interest here are those containing signals / images as their elements. This topic was covered from several perspectives in chapter 3, but should be seen a bit more generally and abstractly in the following.

The simplest manifestation of a signal space is literally as functions (e.g. continuous ones) x on a domain Ω. In that setting, an invertible action on Ω lifts to an action on $\mathcal{C}(\Omega)$ as

$$\hat{\alpha}_g(x)(\mathbf{r}) = x\left(\alpha_{g^{-1}}(\mathbf{r})\right). \tag{5.5}$$

For $\mathcal{L}^2$, an equivalent (for the continuous representatives) lift is obtained by having α act *non*-inverted before the test functions one integrates against, provided the integration measure is invariant under the action:

$$\begin{aligned}
\langle \hat{\alpha}_g(x), y\rangle_{\mathcal{L}^2} &= \int_\Omega d\mathbf{r}\ \left(\hat{\alpha}_g(x)(\mathbf{r}) \cdot y(\mathbf{r})\right) \\
&= \int_\Omega d\mathbf{r}\ \left(x\left(\alpha_{g^{-1}}(\mathbf{r})\right) \cdot y(\mathbf{r})\right) \\
&= \int_{\alpha_g(\Omega)} d\mathbf{r}\ \left(x(\mathbf{r}) \cdot y(\alpha_g(\mathbf{r}))\right) \\
&= \int_\Omega d\mathbf{r}\ \left(x(\mathbf{r}) \cdot y\left(\hat{\alpha}_{g^{-1}}(\mathbf{r})\right)\right) \\
&= \left\langle x, \hat{\alpha}_{g^{-1}}(y)\right\rangle_{\mathcal{L}^2}.
\end{aligned}$$

Note that $\hat{\alpha}_{g^{-1}}(y)$ could be computed *without* an actual inverse action on Ω.

Remark 26. *The reason for using the inverse g^{-1} in Equation 5.5 is that this has the effect of moving a localised feature in the function x at position $\mathbf{r}$ to the position $\alpha_g(\mathbf{r})$. One way of seeing this is integrating against a correspondingly localised test function.*

All of the following will be concerned with actions on function spaces, i.e. $A = \mathcal{L}^2(\Omega)$ or similar. The models whose symmetries are of interest are functionals of the type $\mathcal{L}^2(\Omega) \to \mathcal{L}^2(\Omega)$, and symmetry is understood in the sense of $\hat{\alpha}$. The codomain of these function spaces can vary, but is in practice a generic $\mathbb{R}^n$ Euclidean space.

5.2.3 *The simple case and its generalisations*

The vector space structure of the group of translations includes a commutative addition / group-product, and the translation group is isomorphic to the domain with $\alpha_g(\mathbf{r}) = \mathbf{r} + g$. This allows a dramatic reduction in complexity of, specifically, *linear* functionals of the above type which are equivariant; in signal processing these are called "linear time-invariant [sic]". All of these can be computed as a single convolution with some suitable kernel ψ, or in other words in Fourier space as frequency-wise multiplication with the kernel's Fourier transform.[3]

The idea of Cohen and Welling [20] is[4] to generalise these convolutions to other Lie

[3] This standard result follows from the fact that equivariant functionals must have complex exponentials as eigenfunctions.

[4] In [20], the formula is written for discretized representations, i.e. with sums instead of integrals,

groups, using an analogous definition

$$\begin{aligned} \cdot \star \cdot : \mathcal{L}^2(\Omega) \times \mathcal{L}^2(\Omega) &\to \mathcal{L}^2(G) \\ (x \star \psi)(g) &= \int_\Omega d\mathbf{r} \left(x(\mathbf{r}) \cdot \psi\left(\alpha_{g^{-1}}(\mathbf{r})\right)\right). \end{aligned} \tag{5.6}$$

Remark 27. *A detail that is often glossed over, and also here, is the terminology between "convolution" and "correlation". Equation 5.6 is more convention-adheringly called "correlation", whereas a convolution would use the kernel with flipped inputs. The distinction is in practice unimportant because the kernels are freely learned, which works the same with or without flip.*

Cohen and Welling [20] also offer a version operating on inputs as functions in the group itself, the motivation being to compose multiple convolutions (since Equation 5.6 does not give back a $\mathcal{L}^2(\Omega)$signal as its result):

$$\begin{aligned} \cdot \star \cdot : \mathcal{L}^2(G) \times \mathcal{L}^2(G) &\to \mathcal{L}^2(G) \\ (x \star \psi)(g) &= \int_G dh \left(x(h) \cdot \psi\left(g^{-1} \cdot h\right)\right). \end{aligned} \tag{5.7}$$

Both of these constructions are equivariant in the sense that the action of $u \in G$ on the input ($\hat{\alpha}_u$ in case of Equation 5.6) corresponds to an action of u on the result (with simple group multiplication as the position-action).

These group-convolutions work well for discrete groups (truely discrete, not just discretized), but they are in the original form from [20] not amenable for, particularly, continuous rotational symmetry, because the integral could not be implemented efficiently enough.

Numerous special versions of the general idea have been published since, but all that we were aware of either diverge too far from the ideal of a general-purpose linear equivariant mapping, or have their own computation challenges, which is why chapter 7 develops a new scheme instead. It is very likely that some existing method could also be adapted, but this project did not have time to try this.

5.3 Convolutional neural networks

Although linear mappings are useful for many purposes, most real-world relationships are at least moderately nonlinear, and often completely nonlinear (in the sense that they can not satisfyingly be described by a linear term with added perturbations). Attempting to directly fit a completely general function to any measured data is however hopeless. Classical approaches include fitting special classes of functions such as polynomials, but this tends to only work for sufficiently low-dimensional data.

5.3.1 *Neural networks*

The currently highly popular approach of Deep Learning can be summarized as exploiting the simplicity and well-behavedness of linear mappings within an architecture that

can model nonlinear relationships. In the broadest terms, a deep neural network is a composition of linear functions ("layers") which are highly parameterized (but linearity severely reduces the possibilities; these parameters amount to matrix entries), and nonlinear but fixed ones (generally a selected $\mathbb{R} \to \mathbb{R}$ function applied to each vector entry). More specifically, a (fully connected) feedforward neural network alternates between such linear layers plus biases (in other words, affine layers) and pointwise-nonlinearity layers (called activations).

The activation is regularly just a single function chosen once for the entire network. The most popular choice are historically sigmoidal functions, i.e. smoothened versions of a Heaviside function (the motivation being a "either activate or not" behaviour, inspired by the namesake biological neurons), or nowadays predominantly ReLU functions, i.e. $\max(\cdot, 0)$. Considerations in the choice of activation will not be discussed here.

Such networks were originally called (multilayer-) *perceptrons*. Part of the reasoning often cited for this particular layout is that it satisfies a universal approximation property [40], meaning any sufficiently wide feedforward neural network can to arbitrary accuracy model any continuous function; this argument should be taken with some care though because much the same could be said about many other classes of models, which nevertheless have failed to achieve similar success to deep learning architectures. Existance of an approximation does after all not prove anything about that an approximation can also be constructed efficiently and stably for a relationship to be inferred from limited data.

5.3.2 *Convolutional*

Indeed, the fully connected version of feedforward network is hardly usable for applications like image processing either, having still too much freedom and being prone to overfitting – much like more traditional highly-parameterised models. What has brought such networks to today's success in these applications is the use of convolutions as the linear layers, in combination with efficient gradient-based training, with the gradients being computed by reverse-mode automatic differentiation (also known as backpropagation).[52]

The most interesting aspect of these architectures for the present discussion is that the convolutional layers are equivariant under translations, and the pointwise activations are trivially equivariant too. An entire fully-convolutional deep neural network, as a composition of only equivariant functions, is therefore also as a whole equivariant under translations in the input.

It would be suggestive to implement the convolutions in Fourier space, where they simply amount to a frequency-wise multiplication. The problem with this is that if the activations are to work in position space, an entire Fourier transform would need to be performed in between each linear and nonlinear layer. While FFT algorithms make this in principle feasible ($\mathcal{O}(n \cdot \log n)$), it would still incur a substantial computational cost compared to the very cheap $\mathcal{O}(n)$ and embarrassingly parallelizable nonlinearity and multiplication steps.

which fits the implementation and their examples, but not so much the general concept and the symmetries relevant here.

Direct (integral-as-sum) computation of a general convolution meanwhile has a cost on the order $\mathcal{O}(n^2)$, which disqualifies it altogether. This can be dramatically reduced though for the special case of (very) compactly supported kernels, such that the integral/sum only needs to be carried out over a few pixels. It can then be unrolled by an optimizing compiler, runs in $\mathcal{O}(n)$ and almost as fast as the activation functions on GPUs.

This can also be motivated from more deep reasons: *locality* is another far-reaching[5] principle in physics. Physical interactions always originate on the microscopic level, and all long-range interactions can be seen as propagation of chains of local-scope interactions. This motivates also why the activations should happen in position space: a nonlinearity in Fourier space would be highly non-local in position space. In the mathematic formulation of classical physics, this locality manifests mostly in the fact that systems are usually described by differential equations. That provides another motivation for the significance of small convolutions in particular: in a finite-differences discretisation of differential equations[6], differential operators are represented by compactly supported convolutions.

Remark 28. *The term "local" can have two related yet distinct meanings relevant for the present context: a local symmetry may be understood as a symmetry under only small or even infinitesimal input transformations. This is often a more tractable notion of symmetry than global in the sense of e.g. arbitrary large translations [63]. The above paragraph however refers to locality in the spatial structure of image-like inputs instead.*

5.3.3 *Symmetry breakers*

Perfect equivariance is not always desired (Section 5.1), or potentially even nonsensical. An image classifier for example does not even have any capability of expressing spatial information in its output, the output space being not a function space over a domain like Ω but only a space spanned by finitely many labels. This is why image classifiers generally have at least one fully connected layer at the end, mapping from a spatial space to an abstract one. Having such a layer fully parametric in all $h \times w \geqslant 10^5$ inputs would however still be prone to overfitting and loss of all the spatial symmetry preserved so far in the convolutional parts.

This is less of an issue if the resolution is reduced beforehand. To this end, image classifiers typically employ *pooling* layers, which group together several neighbouring pixels using a mathematical operation which may be linear or nonlinear, a popular choice being a maximum. Such pooling layers are strictly speaking neither local nor equivariant, but do satisfy both properties in an approximate sense: translations of a sufficiently oversampled (or lowpass-filtered) signal by $2 \cdot k$ pixels in the input to a 2×2 pooling operation results approximately in a translation by k pixels in its output. Notice that very small translations map to even smaller ones (relative to the respective pixel scales). Suitable lowpass filtering for this to work can be provided by preceding convolutional layers, though this is not perfect, the convolutions do in principle not have

[5] Pun not intended.

[6] More precisely, an *explicit* discretisation. In an implicit scheme, the operators can in effect have infinite support.

to act as lowpasses at all, and even if they do then the nonlinear activations introduce again harmonics which can subvert this.

Indeed, the nonlinearities themselves are not equivariant under sub-pixel translations, even though they conceptually represent perfectly equivariant post-composition of continuous functions. This could be reinstored through antialiasing techniques [119], but that is seldom done in practice.

Another effect that prevents convolutional networks from being truely symmetric is that the bounded rectangular domains they work on cannot even express translations except of signals which are compactly supported on sufficiently a small subdomain. This is seldom given for photos (and anyways requires a suitable convention of zero colour). Equivariance does still hold generally for a convolutional layer in the sense that the restriction of the output to a co-translated subdomain is invariant under the translation; however again in practice no such domain-restriction is part of the architecture.

For these reasons, saying image classifiers are invariant under (even small) translations is problematic, if not not outright wrong; indeed it is easy to observe that very small translations can strongly change their outputs [6] in a way almost reminiscent of the adversarial effects discussed in Section 1.4.3A and Section 2.5.1. How important the architecture-imposed symmetry properties of deep neural networks' components nevertheless are is still a matter of debate, which this thesis can not hope to settle for the general case. Instead, the next chapters focus on a specific application that is particularly amenable to investigations of symmetry. This can however also include symmetries that are *not* pre-determined by architecture.

5.3.4 *Semi-intrinsic symmetry*

Even a fully-connected network can in principle learn any symmetry, if it is sufficiently exhibited by training data. Such learned symmetry can be artificially encouraged by the use of *data augmentation*, i.e. by representing each sample in the original data set with not just itself but multiple symmetry-transformed copies of it. This could be seen as increasing the dataset to arbitrary size. Alternatively one could also use completely synthetic data, which can include the symmetry action systematically or randomly. Making a fully connected network e.g. translationally invariant in tis way is still impractical, since it would require that it had seen every possible pixel shift (or at least a large fraction of them) for each image. For a data set that is by itself of large size, and a high resolution, this would even on modern hardware require excessive training time.

For discrete symmetries, this is far more promising, but empirically it also works for making a model that already obeys a translational symmetry (in particular, convolutional neural networks) symmetric under another group. It is common practice to use random rotation as data augmentation for image classifiers; this may not be sufficient for making the classifier perfectly invariant under rotations [31], but can still succeed in making it sufficiently robust under the rotations encountered in testing (or application) data to increase the accuracy there, which is often worth some additional training time.

Likely, a main reason this works is that local translational symmetry already subsumes much of the combinational complexity of rotational symmetry. To wit, the effect

of any small rotation upon a localised feature is approximated by the action of a translation on that feature. For larger rotations this is not true anymore, but any rotation can be decomposed into one of only a few large rotations (which could be learned brute-force) and a smaller one that is sufficiently approximated by local translational symmetry.

5.3.5 *Diffeomorphisms (excursion)*

A notion of input-transformation that generalises local translations as well as rotations is that of *diffeomorphisms* on the input domain, i.e. smooth deformations that transport different parts of an image in ways that need not be globally connected at all. Diffeomorphism invariance has been argued to be a more powerful and likely more relevant notion of symmetry [63], instead of rigid translations. It approximates the effect of more changes that could happen to the physical scene depicted in a photo, and also sidesteps some of the issues that make translations inapplicable (in particular, boundary limitations).

Diffeomorphisms are locally a combination of translations and linear stretch/shear/rotations. Specifically for diffeomorphisms whose displacement field has small partial derivatives[7], the latter are near-identities so that the local translations dominate the effect of the diffeomorphism, much the same arguments as for translations and rotations apply also for why convolutional architectures are good candidates when diffeomorphism equi- or invariance is desired.

Also, the same caveats apply as in Section 5.3.3. Even an ideal (infinite-resolution) convolutional layer is not exactly diffeomorphism equivariant, and a realistic deep convolutional classifier certainly is not exactly diffeomorphism invariant. Whether it even is benefitial to have approximate invariance is not completely settled either, but Petrini et al. [72] have observed that specifically the relative equivariance under diffeomorphisms compared to general input changes correlates to performance, and increases during training of a classifier. In other words, trivial invariance through general insensitivity is (unsurprisingly) not benefitial, but dedicated invariance to diffeomorphisms in particular is.

This subject is of high interest for future research, but made difficult by the inexactness and need for dedicated regularisations. This thesis does not contribute results on diffeomorphism properties but sticks to rigid transformations, appropriate to the particular application of Cryo-EM.

[7] This is a natural condition: being differentiable is half of the definition of a diffeomorphism, and slow-changing displacement is a sufficient condition for also being invertible with differentiable inverse, which is the other half.

CHAPTER 6

CRYO-EM

In chapter 5 a very general overview was given about the topic of symmetries and their relevance and use possibilities for deep learning. This chapter introduces the concrete application in which we apply this theory, in form of the purpose-built network architecture we present in chapter 7.

6.1 Introduction

Much of the discipline of chemistry is concerned with the structure of molecules, which is crucial for their physical, chemical and biological properties. While in some simple cases the structure is uniquely determined by the atomic stochiometry, for most nontrivial molecules it needs to be determined by dedicated means. They are too small to be seen in a conventional sense, since the wavelength of visible light is larger than the patterns that need to be resolved. One option are shorter-wavelength photons, in particular X-rays. These are indeed used[1], but they cannot be focused the way light can; this makes it necessary to rely on wave-vector based analysis, which can only be applied to periodic structures (such as crystals).

An alternative are electrons, which can be focused thanks to their charge, while having De Broglie wavelengths short enough to resolve features on scales down to few ångströms. They have their own challenges, but are now used in several different microscopy technologies. The particular one that is the topic here, *single-particle cryogenic electron microscopy*, applies transmission electron microscopy to samples in the form of e.g. protein molecules dissolved in a layer of vitreous ice. These show up in the images taken (the *micrographs*) as a kind of shadowcast.

Remark 29. *Some details of the physical process are glossed over here. It is intuitive to think of the molecules as casting shadows, but in fact the amplitude of the electron beam is not greatly affected by passing through the target molecules, compared to the surrounding medium. What is affected instead is the* phase *of the electrons' wave function. Making this detectable requires a slight defocus of the beam, such that the phase-shifted contributions interfere destructively, which is what causes a shadow-like picture as the result.*

Furthermore, in practice the images taken are not static in spite of the frozen state of the specimens, requiring additional frame-alignment [78].

[1] X-ray spectroscopy was the method behind, for example, the discovery of the helical structure of DNA [105].

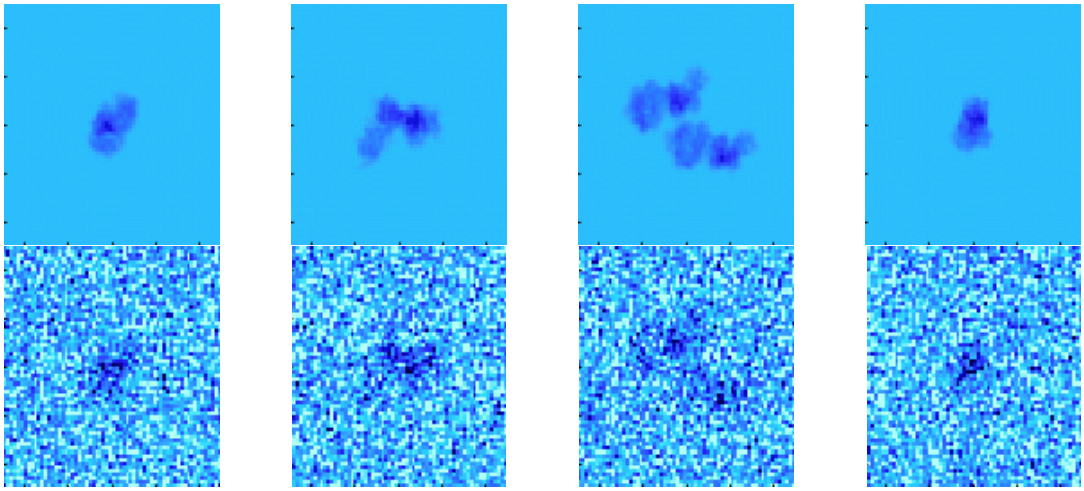


Fig. 6.1: Some examples of how Cryo-EM micrographs would look in a hypothetical noiseless projection, and how they rather look with a realistic amount of (artificial) noise.

The synthetic data used for the experiments discussed in the following were generated from a simple static density projection. This may not be very representative of the real defocus-interference mechanism, but should sufficiently reflect the symmetry-related challenges that the algorithms need to deal with.

The main problem with the cryo-EM method is a very low signal-to-noise ratio [10], as visualised in Figure 6.1. This is an consequence of two fundamental physical factors (limited contrast from the interference technique, and limited electron dose lest the probe suffer damage). It can therefore not be much improved with e.g. better sensors, as might be done in other applications.

There are essentially two approaches to deal with this; in practice a combination/-compromise is employed.

- Seeing the individual-molecule micrographs themselves as the signal to be obtained, one should apply denoising to them as individual images. This can hardly be done with traditional general-purpose image denoising algorithms (which rely on the fact that the signal normally dominates and the noise is only a perturbation), but is possible to some extent with dedicated methods. This is the approach to which the next chapter contributes, although this is fundamentally limited by the smallness of the information amount (in the Shannon noisy-channel sense; cf. Section 3.2) contained in a single projection.

- Since there are in practice many copies of the same molecule in a given sample, better results can be obtained by combining the information from all (or at least several) of them. This would be easy if the molecules were present in a regular pattern and with predictable orientation. In the extreme case of them being all aligned in the same way, they could simply be *stacked*, averaging out the noise. But both location and orientation of the molecules are in fact random, adding much complexity to the challenge. This aspect deserves some brief discussion.

6.2 Tomography

Reconstructing three-dimensional information from projections is a common problem. When the experimentalist has freedom to choose the projection directions, this has a straightforward solution thanks to the Fourier-slice theorem: the collection of all[2] directions corresponds to the density distribution's Radon transform; taking the Fourier transform of each slice and arranging them radially is equivalent to the entire distribution's Fourier transform, which can readily be inverted. Even when not all directions are available, this still provides a linear inverse modelling problem which can be solved with various regularization means. This is relevant for many applications, ranging from many medical ones to geophysics. Traditional regularizations can work well but often do not allow for satisfyingly high resolution. Machine learning can be applied to improve the resolution [104], though with the usual caveat that this may cause unknown training-data biases to infest the results.

In either case, the tasks of denoising and reconstruction would generally be tackled as one single inverse problem, taking raw measured projections as input and giving a noise-stable 3D reconstruction as the result. The denoising therein can for the case of Gaussian noise be interpreted as achieving a least-squares optimal solution; such a method may still also work for other types of noise. Since sets of many images are used as the source material, the denoising performance benefits from the stacking effect.

Even in applications where the projection directions cannot be chosen (yet *are* known), a similar approach to the above can still be carried out, by treating the available sample directions as points in a mesh without a-priori structure, which can nevertheless be used as a discretisation in e.g. a finite-element sense.

In the case of cryo-EM, the challenge is that the directions are not even known, since the orientation of each molecule cannot be determined except from the image itself. And with the initial high level of noise, an estimation based on e.g. edge / keypoint based techniques would be too unstable. This leaves two possibilities: first performing single-image denoising and then "classifying" them according to orientation, or obtaining a representation that is invariant under the unknown rotations and performing the stacking-denoising in that representation. Such representations exist [120], but they are mathematically and computationally involved and lose much of the locality, simplicity and interpretability of the original input format.

In the present work, the tomography problems are not addressed per se but only the single-image denoising. It is nevertheless to be noted that this denoising has a particular responsibility which would not be obligate in most other applications: each denoised image needs to be suitable for accurately estimating the orientation of the molecule contained therein. In particular, the denoising should not introduce any biases towards particular orientations, since that would systematically misalign the images to be stacked. A sufficient condition for avoiding such biases is if the denoiser is *equi*variant under rotations.

[2]Of course, in practice only a finite sample of directions is obtained, but they can be chosen along a regular grid, so what one gets is rather a (standard) discretisation of the Radon transform.

6.3 Noise

Noise was mentioned in Section 3.2 in information-theoretical aspects, but not treated explicitly.

A common view is to consider noise as an additive perturbation on top of a signal, i.e.

$$x = x_{\text{sig}} + x_{\text{noise}}.$$

This is certainly appropriate for the noisy-channel setting, specifically for a transmission through a linear medium (e.g. Maxwell equations) with independent interference. It is also a good model for many other scenarios, including Johnson noise from the resistors in analogue electric circuits (which appears in essentially all measurements in practice). The predominant noise source in Cryo-EM images however is technically *not* of this nature: it is *Poisson noise* arising from the quantization of the electrons. Though in the limit of many electrons per pixel, the Poisson distribution converges to a Gaussian one centered on the expectation value, this is not well approximated by the low-dose images used in Cryo-EM. In particular, the statistics depend on the pointwise intensity, i.e. the signal itself.

This detail, too, is not considered in the next chapter, and (artificial) additive Gaussian noise used instead, as this simplifies the analysis and comparison of SNR and equivariance concerns. The denoiser models based on neural networks do not explicitly assume additive and/or Gaussian noise, so it is reasonable to assume they would be usable for Poisson noise as well if they work for the Gaussian case. This is not to say that future investigation of the performance for the Poisson case is not important, though.

The additive model also makes the notion of signal-to-noise ratio straightforward, as the ratio of the amplitudes or norms of x_{sig} and x_{noise}.

Many different approaches exist for the attenuation of noise, i.e. estimation of x_{sig} when only x is known. The simplest rely on a known frequency-domain split: for typical e.g. audio signals, much the noise energy is in high-frequency components of the Fourier decomposition whereas the signal's energy is concentrated in the lower frequencies. As a result, applying a simple lowpass filter to x produces something closer (in the $\mathcal{L}^2$ sense) to x_{sig}. Except for specific use cases[3], the side effects of such crude filtering (such as the smearing out of transients) will however scarcely be acceptable.

Thanks to linearity, the filtering approach is very well understood theoretically and has for the Gaussian case an optimal solution regarding the tradeoff between noise supression fidelity, in form of the *Wiener filter* [110], whose theory will not be covered here.

Like with several applications mentioned earlier in this thesis, neural networks have in recent years become an alternative to such traditional methods also for the task of denoising signals / images. The idea is to use some deep, usually convolutional network and train it to approximate, as a generic regression problem, the hidden x_{sig} associated to training examples of x [43]. Alternatively it can be made to estimate x_{noise}, which appears to be an equivalent problem (this estimated x_{noise} can then be subtracted from x) but is claimed to have some practical advantages [118]. Curiously enough, it

[3]One use case where it is acceptable is indeed in Cryo-EM, but only for the preliminary step of determining the molecule positions, not for any structure investigations.

is even possible to use different noisy images as the training target, but still obtain a network that removes noise [53]; this has the advantage of only requiring realistic data with both signal and noise for training. In the following, the noise will however always be synthetic, so this does not have much advantage and direct signal estimation makes for clearer comparisons.

CHAPTER 7

EQUIVARIANT DENOISING

This chapter presents a particular architecture that can be used for denoising Cryo-EM images in an *approximately* equivariant way. This is part of a larger project on Cryo-EM which is still very much work in progress, so the following will not go into great depth but only show one idea and how it links to the topics covered previously.

7.1 Problem formulation

The concrete problem to be solved here is the estimation of x_{sig} from a given $x = x_{\text{sig}} + x_{\text{noise}}$,[1] where all three signals live in a space $\mathcal{I}$. In practice these are normally taken to be square pixel images, but conceptually it is rather a function space on a disc with the molecule's center of mass as its midpoint. Assume therefore

$$\mathcal{I} = \mathcal{L}^2\left(D^2\right), \tag{7.1}$$

with

$$D^2 = \left\{\mathbf{r} \in \mathbb{R}^2 \middle| \|\mathbf{r}\|_2 \leqslant 1\right\}, \tag{7.2}$$

for which we use the standard parametrisation with $r \in [0, 1]$ and $\vartheta \in [-\pi, \pi]$ under the equivalence relation identifying all ϑ when $r = 0$, and only $\vartheta = \pi$ with $\vartheta = -\pi$ else.

The rotation group $\mathrm{SO}(2)$, which is a Lie group homeomorphic to the circle S^1 that we parameterise by $\delta\vartheta \in [-\pi, \pi]$, acts on D^2 by way of

$$(r, \vartheta) \mapsto (r, \vartheta + \delta\vartheta) \tag{7.3}$$

(mod $2{\cdot}\pi$), and in the correspondingly induced way (Equation 5.5) on $\mathcal{I}$.

Remark 30. $\mathrm{SO}(2)$ *is only a small subgroup of the rotations a molecule can undergo, namely those whose axis is aligned with the projection direction (z-axis). Furthermore, since the measurement setup cannot distinguish flips along the z-axis, the actual symmetry of relevance is* $\mathrm{O}(3)$. *But the plane rotations are already a nontrivial and useful special case.*

The objective of an equivariant denoiser $\mathsf{F}\colon \mathcal{I} \to \mathcal{L}$ then is to

1. Denoise, in the sense that $\|\mathsf{F}(x) - x_{\text{sig}}\|$ should be small
2. Be equivariant in the sense of Definition 5.

[1] See additivity caveats in Section 6.3.

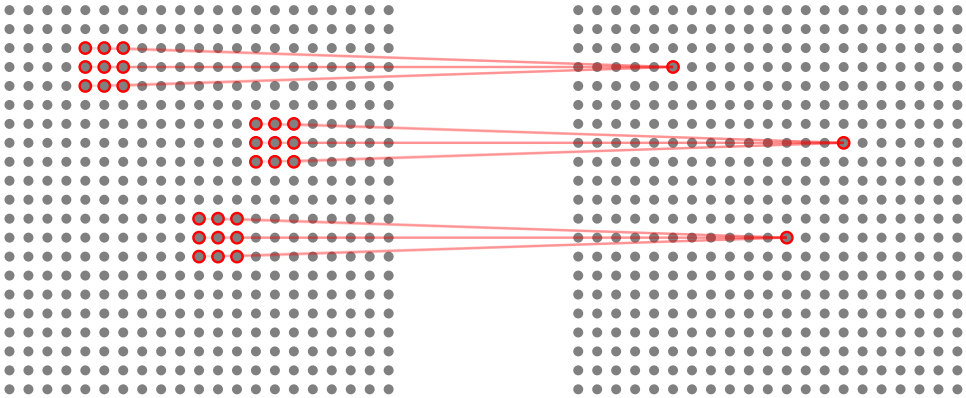


Fig. 7.1: The fields of reception in an ordinary convolution are simply translated copies.

7.2 Existing approaches

Standard convolutional neural networks have translationally equivariant layers[2]. This is a useful property for many image applications but not really appropriate for this denoising problem as formulated, because translations of a molecule would co-shift its center of mass and thus actually leave the representation *in*variant rather than equivariant. Convolutional neural networks are not inherently rotation-equivariant at all. It turns out that they can readily learn to be approximately equivariant when given enough data, though at least in the case of $\mathrm{SO}(3)$ acting on the sphere it was found [31] that such data-augmentation-based equivariance can not take it up with built-in equivariance.[3]

As such, using a convolutional network on the cartesian representation of the noisy image is a possible strategy, but it is not ideal when equivariance is important as it is in the first stages of a Cryo-EM pipeline.

An alternative is to use them on the *polar* representation, in which case one of the axis of the network's equivariance becomes the desired angular equivariance of the images. The problem with this is that it is badly compatible with another central feature of convolutions, that of *weight-sharing*: in a normal convolution, a given filter kernel is used in the same way everywhere across the image (Figure 7.1).If the convolution is performed in polar coordinates however, the real shape of each kernel as applied to the image varies dramatically (Figure 7.2). Particularly critical is that the fields of reception get very narrow in angular direction as the radius is decreased. Such a narrow field is not enough to gather sufficient data to perform the kind of local averaging required for denoising (or to gain enough information to perform statistical lookup with the internal representation learned from the training data). Another way of looking at this is is that the frequency accuracy of the filters that the convolution kernels implement get poorer and poorer towards the center. There are ways of addressing these problems, but this is mathematically challenging and requires foregoing the simple local computation nature of the convolution operations. We will not address any of these techniques here,

[2]This is a simplification not entirely true due to the finite resolution, unless antialiasing is employed [119].

[3]They found that *in*variant models trained by augmentation can reach the performance of inherently invariant ones, though. Both of these observations are empirical and may not hold for all applications.

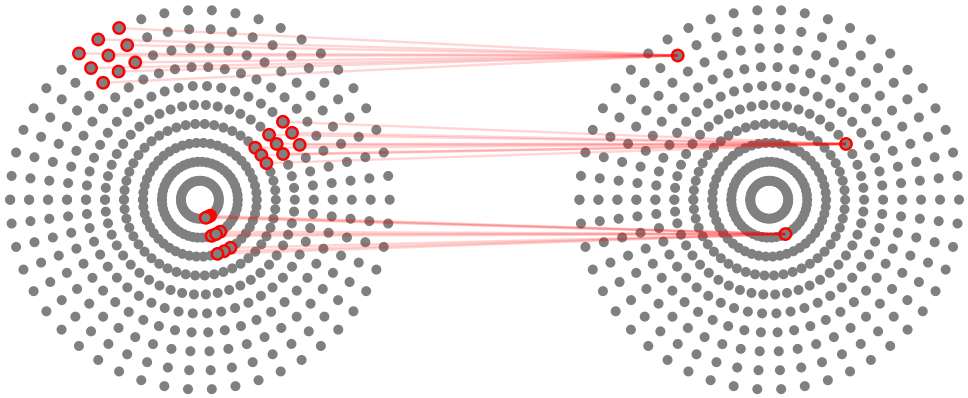


Fig. 7.2: How the fields of reception thin out towards the center in a polar version of convolution.

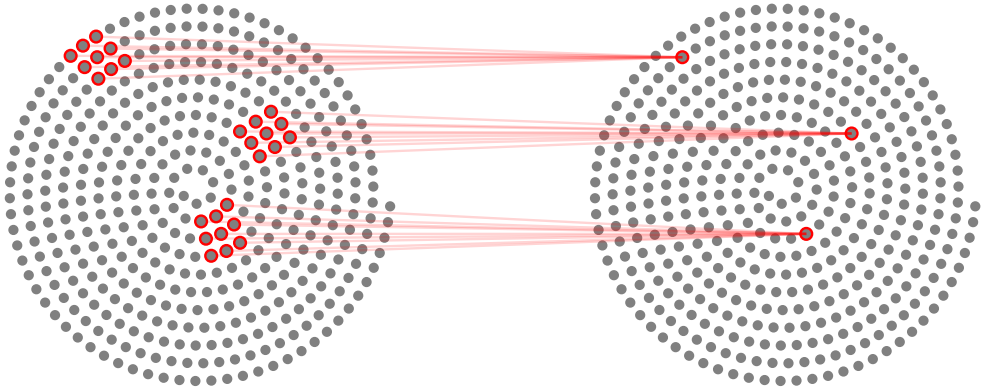


Fig. 7.3: The spiral sampling with approximately constant point density in both radial and angular direction.

but instead suggests a rather simple alternative that allows using standard convolution implementations and still get a nearly rotation-equivariant operation with fields of reception that have suitable homegeneity.

7.3 Spiral convolutional neural network

The idea is to use instead of a grid of concentric rings one made up from a single *spiral* path, specifically an Archemedean spiral.

7.3.1 *Spiral sampling*

The Archimedean spiral has the property that each winding is equidistant from the next, measured in the radial direction. When many windings are used (as they will be to realise a typical image resolution), they are also asymptotically orthogonal to the radial direction and therefore then equidistant in the sense of the $\mathbb{R}^2$ they are embedded into. Because the windings, unlike the rings in a polar grid, do not have to close upon themselves with an integer count, they can also at every location be spaced out arbitrarily, allowing in particular making the angular spacing also equidistant in the

$l^2(\mathbb{R}^2)$ sense. Figure 7.3 demonstrates how this by itself makes the fields of reception behave more like in the cartesian case.

Normally, the Archimedean spiral is presented by the radius parameterised by the angular coordinate:

$$r(\vartheta) = b \cdot \vartheta.$$

Our construction can be understood as inverting the dependency and then sampling ϑ so that at each radius an appropriate spacing is reached. In the limit of an infinitely tight sampling this is equivalent to parameterising r so its derivative is inversely proportional to the circumference at that radius:

$$\frac{\partial r(t)}{\partial t} = \frac{1}{2 \cdot \pi \cdot r(t)}, \tag{7.4}$$

the solution of which is (modulo time shift)

$$r(t) = \sqrt{\frac{t}{\pi}}. \tag{7.5}$$

The steepness b is now chosen to facilitate that the disc (conventionally of radius 1) is covered by a choosable number of windings n_{wind}:

$$\vartheta(t) = 2 \cdot \pi \cdot n_{\text{wind}} \cdot r(t) = 2 \cdot n_{\text{wind}} \cdot \sqrt{\pi \cdot t}. \tag{7.6}$$

This is then sampled homogeneously to cover the entire range $t \in [0, \pi]$ with a choosable number n_{spls} of total sample points:

$$t_i = \frac{i \cdot \pi}{n_{\text{spls}}}. \tag{7.7}$$

7.3.2 *Spiral sampling*

So far, this construction only samples a one-dimensional path. That would at most give rise to a convolution operation with kernels extending *only* in (near-) angular direction. But a convolution that expresses kernels covering an open set in $\mathbb{R}^2$ requires taking also the topology of the neighbour winding in radial direction into account.

To this effect, we define the convolution not on the spiral sampling itself but rather on a spiral *ribbon* of desired thickness, which is then (validly) convolved down to the 1D path on which the result is represented (Figure 7.4). With the ribbon being a $3 \times n_{\text{spls}}$ array (or $5 \times n_{\text{spls}}$, depending on what size of convolution one wishes to use), the convolution on it can be carried out with standard routines (PyTorch `conv2d` in our implementation). But to obtain the values on the ribbon in the first place requires resampling from the representation used by the preceding layer. We chose to implement this based on a generic method: using the Delauney triangulation of the preceding layer's output sample point, and linear interpolation in the barycentric coordinates of the source triangle containing the point at which the representation is to be sampled. This is a fairly standard technique (corresponding e.g. to some finite element methods), so we will not go into the details.

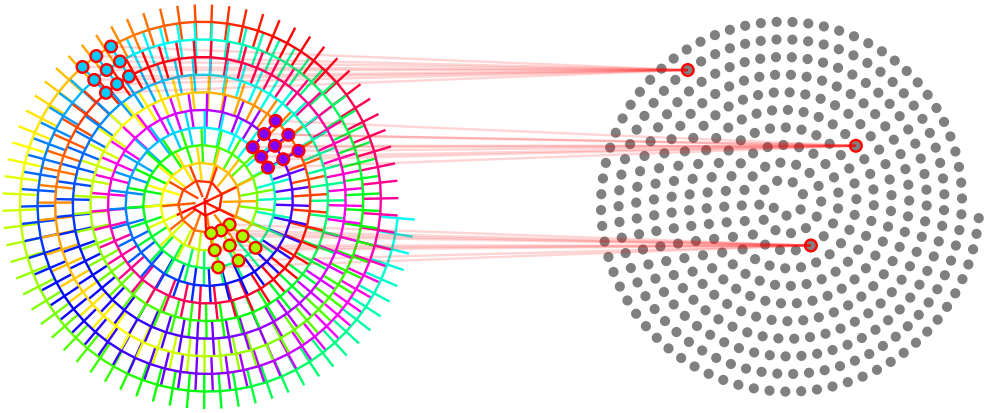


Fig. 7.4: The spiral ribbon on which the convolution actionally is carried out.

The thus defined convolution can then be applied to arbitrary images on the disc. There are some subtleties to take into account at the outer rim and near the center, but these are mostly analogous to questions that also arise for ordinary convolutional networks at the boundaries.

In Figure 7.5 we see that this convolution does, as desired, to good approximation rotate the patterns in the kernels. Furthermore it also shows a (less ideal) radial equivariance. It is not clear whether this is actually advantageous for Cryo-EM denoising, but at least for the processing of the noise it is sensible (certainly much more sensible than angularly squished kernels of the polar representation), and though the molecules in the images this is intended to be used for will be centered it is still plausible to see similar patterns at different radii, e.g. corresponding to chemically similar groups at the rims of different-size molecules. Radial equivariance is more sensible than cartesian equivariance in this regard.

Near the center (or when using very large convolutions or low resolutions, like in Figure 7.4) the radial symmetry breaks down due to the increasingly tight flexing of the convolution ribbon.

Remark 31. *Outside of the disc of interest the behaviour of the convolution is generated by extrapolation; this might best be considered as* undefined behaviour.

7.3.3 *Spiral deep network*

To tie it all together into an architecture that can be effectively trained requires precomputing all of the spiral geometry. This concerns *only* the step of resampling to the ribbon representation; the convolutions can be carried out without knowledge of what space they are assumed to be embedded in.

The resampling meanwhile is a fixed linear mapping, and therefore can be fully expressed by a single matrix. Due to the high locality, it is importantly a sparse matrix, making this feasible to compute even for higher resolutions. Unlike in Section 4.2.3, it is exactly sparse by construction, so no enforced cutoffs are needed (though they can still be benefitial to reduce the number of nonzero elements further[4]).

[4]The reasons is that the edges of the ribbon mostly lie right on the centerline of the neighbouring windings. As such, the interpolation takes in practice mostly place between only two, not three points.

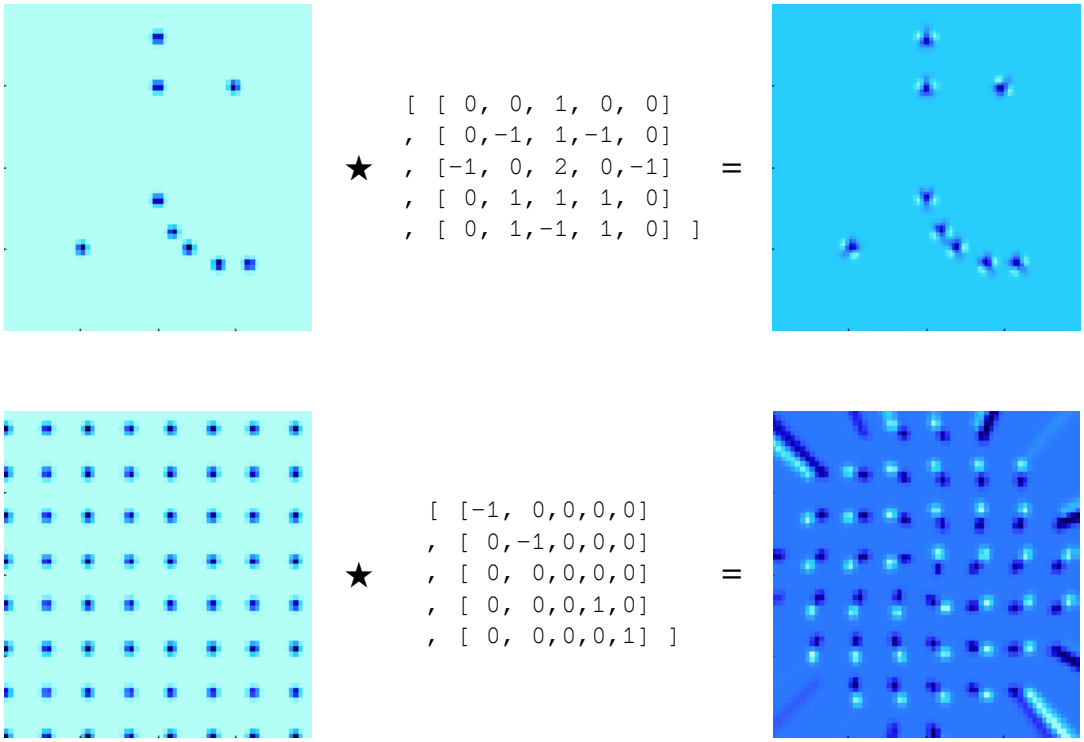


Fig. 7.5: Test patterns for the spiral convolution. The original images contain here only several Dirac peaks, whose place is after convolution taken by almost-copies of the convolution kernel.

In other regards, the networks we tried are fairly simplistic standard ones, with RELU activation between the convolutions and batch-normalisation. We tried various numbers of layers and sizes of the hidden ones.

7.4 Results

We trained this new architecture on a (simulated, to have noiseless ground truth) dataset that had also previously been used for other denoisers. This section only briefly summarizes some of the results that we find to hold quite consistently; our experiments included many more variations of network sizes, data selection etc.. than will be compared here. As Figure 7.6 shows, the new architecture works and is able to perform the task just as well as the conventional one – in fact they are hard to even distinguish.

Differences, apart from the *guarantee* of equivariance (not exactly but approximate) in the spiral version, can be found in the details, particularly with regard to how the properties develop during training. The most obvious difference is that the spiral denoiser is nearly (rotation-) equivariant already right from the start of the training, whereas the cartesian version has much higher non-equivariance at the start. When the signal-to-noise ratio is reasonably low, it can catch up almost perfectly though (Figure 7.7), apparently because there are enough instances of rotated near-copies in the dataset.

It is a different story at lower SNR (i.e., higher noise level, as it realistically will be): in this case, the cartesian version does not match the equivariance performance of the spiral even after longer train, and it also does not achieve as high denoising fidelity

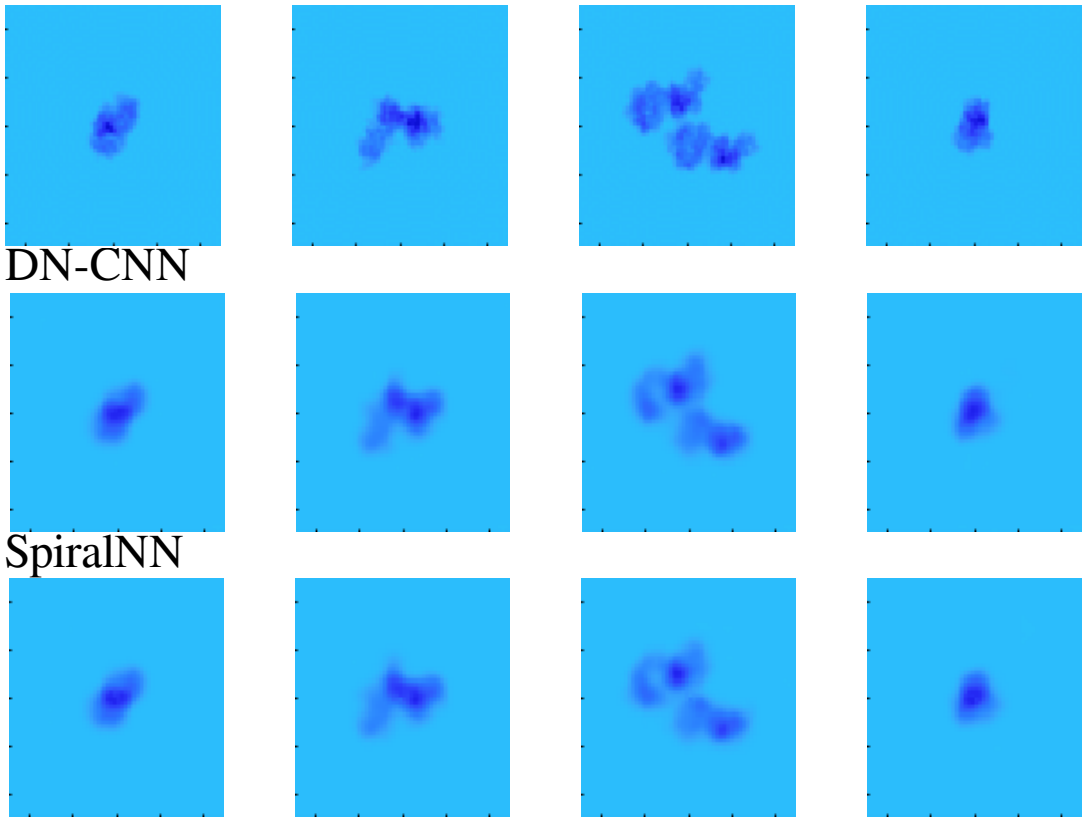


Fig. 7.6: How a normal cartesian convolutional network, and a spiral one as proposed here denoise the examples from Figure 6.1.

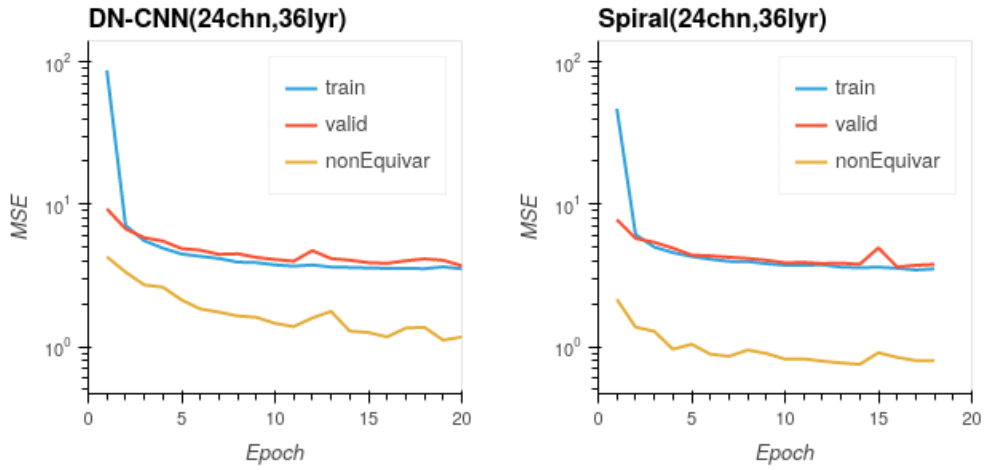


Fig. 7.7: Comparison of the two denoiser architectures' score- and equivariance performance, with images of signal-to-noise ratio $\frac{1}{4}$.

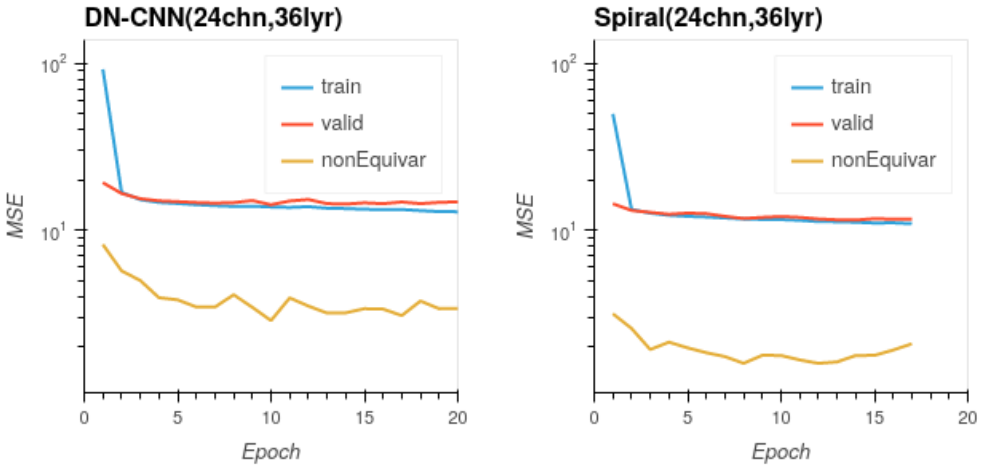


Fig. 7.8: Like Figure 7.7, but with signal-to-noise ratio $\frac{1}{64}$.

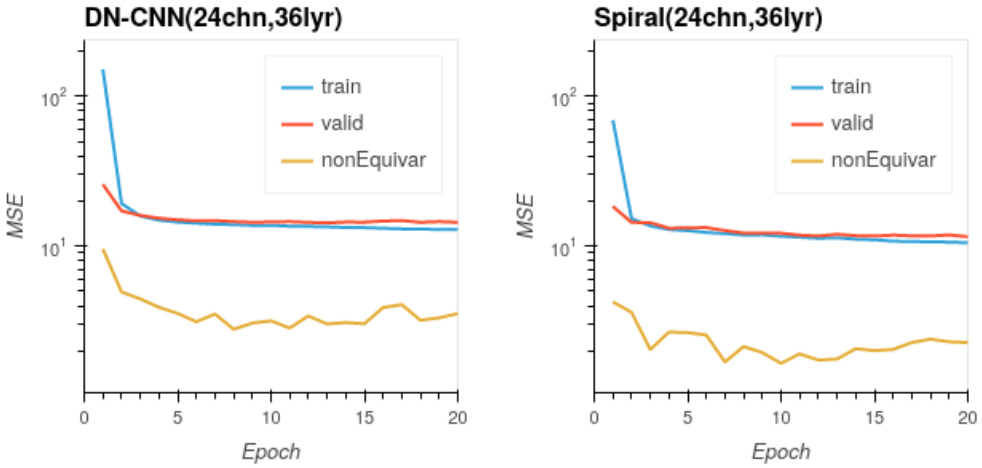


Fig. 7.9: Like Figure 7.8, but with random rotations applied to the images in each training epoch.

(Figure 7.8) – which is plausible, since it did not take the additional information from rotational equivariance into account. The performance difference is not big though, unlike the equivariance difference. It mostly shows up in the form of slightly more overfitting of the cartesian network.

This changes slightly when more rotation examples are supplied in the form of random data augmentation. In this case, the cartesian version comes closer to the spiral one in terms of equivariance, but still does not quite reach it (Figure 7.8). This matches the experience of reports concerning other use cases of equivariance [31].

Part V

CONCLUSIONS

CHAPTER ω

CONCLUSIONS

ω.1 Summary of Results

In Part II, we developed a novel formalism for saliency. The Ablation Path method has merits both theoretically and practically:

- It demonstrates how the Integrated Gradient method, Meaningful Perturbation methods, and ablation scores can be seen as more or less special cases or aspects of a single formalism.
- It proposes axioms for paths that are at least in principle sufficient to discuss the optimisation problem over them. Whilst the concept of paths per se is not new, this formalisation was required to use them as first-class citizens. Within that framework, we demonstrate that optimisation of the paths leads to a saliency method that comes close to established methods, even in a metric like the pointing game.
- It comes with a convenient interactive tool, letting users not only look at the saliency as a heatmap but actively investigate what it means in terms of input-output pairs of the classifier.

The last point, although it did not take up a lot of space in chapter 2, is the most valuable in practice, and also the one that justifies the claim that this method has forayed into making explanations reliable: it does this not by guaranteeing a good explanation, but by giving the user the best chance to assess how good the explanation is.

The flip side is that the explanations really are not guaranteed to be good. In fact, the results provide in many cases more confusion than clarification. Part of the reason for this seems to be to blame on the algorithm, or perhaps even the idea of tracing paths through their input domain of a classifier.

But we are confident that this is not the whole reason. Though the state-of-the-art literature methods outperform ours in the pointing game, they do not do so by a large margin, also producing in many cases heatmaps that seem rather nonsensical, so it is not just our algorithm that has these issues.

What the heart of the problem is, we cannot say with certainty. But it seems likely that it has to do with the whole notion of *pixels as features* being a flawed concept, for purposes of classification-explanation. We have highlighted in particular the dilemma that smoothing / regularisation is simultaneously necessary and detrimental. Different

methods use various tricks to work around this [27][19][46], but though this can work it makes the interventions idiosyncratic in their own way (blobby / straightedges / cartoon-look, for these respective answers). At best, this makes it somewhat of a gamble whether the classifier will respond sensibly to such inputs; at worst a malicous black-box could actively detect them and proceed to adjust its behaviour, appearing more innocuous than it really is (à la Volkswagen emissions scandal).

These considerations, amongst others, lead us to Part III. Therein we first broadly surveyed possible alternative notions of feature-interventions instead of pixels, before deciding on the SIFT decomposition. Although it was challenging to implement this decomposition in the required encoder-decoder fashion, we arrived at a solution which appears to solve the problem without reservations. The only point of worry is the still rather high memory demand, which has no bearing upon the mathematics (only makes use more hardware-demanding).[1]

We then succeeded in using the SIFT decomposition together with the Ablation Path method, and deem that it works – rather better than in the pixel basis. This would still benefit from more experimental backing; these are somewhat hindered by the fact that the SIFT attributions cannot entirely well be converted into spatial attributions for running the pointing game. We do *not* consider the non-spatialness a deficiency of the decomposition, only an (admittedly inconvenient) incompatibility. A human employing Ablation-Path+SIFT in search of explanations can after all look at the input/classification pairs without any need for the masks to be shown as heatmaps; the actual intervention images are anyways more informative, being also capable of expressing changes like background colour.

It would however also be dishonest to claim that the SIFT basis provides a wonder cure to the occurance of wrong/useless path optimisations. Examples where the saliency does not seem to make any sense can still readily be found. Nor does it count the possibility of volkswagening, although SIFT-manipulated images (with the right smoothing, Section 4.1.3C) tend to have far less obvious artifacts than pixel-manipulated ones.

And it is likely the fate of any black box explainability method to be vulnerable in such ways: there is always the possibility for the classifier to do something truely incomprehensible or malicious. This general truth about such methods became again and again clear during the work on the thesis: black-box explanations are often useful, but one should never take this for granted.

As such, it was fitting to conclude the thesis with the topic of Part IV. It is only weakly connected to that of the preceding parts, and does not deal with black box explanation at all. This part is more an excursion than a research project of its own.[2] It would be an exaggeration to call the spiral convolutional network developed therein an interpretable model, but what can be said is that it has an aspect that is advantageous in an interpretable way.

The thesis-relevant achievement here is not necessarily the specific denoising archi-

[1] Even this could be partly circumvented through more agressive sparsity cutoff and/or downsampling, which we eschewed for the time being – mostly to ensure we do not mistake issues arising from these implementation details for more fundamental problems.

[2] It is a small part of a larger and still ongoing project.

tecture with spiral convolutions. Rather, it is the gathering of domain knowledge about an application that is on one hand difficult enough to warrant use of deep learning, on the other hand unusually well understood and controlled. From there, the spiral convolution was more or less just a demonstration that this knowledge can be used to make a small and simple change to the architecture, that has as an effect a small improvement in performance and more importantly reliability.

ω.2 Takeaways

To wrap up these four years of research attempting to understand deep learning also requires to put it in the context of the time. The years 2019-2023 have been turbulent in many ways, and the progresses in the AI field are among them.

I reckon myself to have achieved success in my work, yet it makes only a very slight contribution to the state of the art in explainability. And that progress concerns models that were already available before I started the project.I would claim that my work was reasonably successful, yet it makes only a very slight contribution to the state of the art in explainability, and this regards models which were already available before I started the project. Other authors have of course made progress too, but we are still far from having a firm grip on the understanding of those models.

In the meantime, the machine learning industry has moved on in leaps and bounds with the development of models vastly more complex than were available in 2019. Extrapolations about an entire research field's future are risky, but it is hard to escape the feeling that the endeavour of explainability is falling further and further behind the goals it aspires to.

Clearly, not everybody agrees – with that assessment itself, or whether it is a bad thing. Many would say that explainability of neural networks has not really been necessary so far and progress in AI capabilities is always good (with negative outcomes only being a matter of humans using the systems in bad ways). There are many legitimate reasons to be enthusiastic about the progress in generative AI, and the opportunities it opens up. Yet, severe negative ramifications (aside from human malice) are plausible and just as numerous. And even if we understood these systems well, it would be hard to estimate how likely each such scenario is.

The experience I made during this work with probing image classifiers in many different ways and building denoising models has made me both more impressed of how deep neural networks can solve some tasks with unexpected autonomy (such as the rotation-equivariance developed by a cartesian convolutional network), as well as dismayed for how easily they flip to sudden, unexplained and drastic misbehaviour (as with the manifold adversarial-like responses in ablation-path intervention).

I actually do not know what conclusions to take from this, regarding black-box explainers. On one hand, they seem more needed than ever and in need of research to improve their reliability, on the other hand I am doubtful whether they ever will be reliable. A stance that I subscribe to more than ever is the one expressed by Rudin [80]: that critical decisions should not be left to black box models at all, and inherent interplainability is the way forward.

No doubt both will stay relevant, but inherent interpretability is perhaps the branch

deserving of more research effort and also political support to increase adoption. I do not believe this is possible without sacrificing performance in some cases (where deep neural networks manage to be "magically" good), but do think it is a price worth paying. For the uses where black boxes are the only option, we should not leave it to a probability game to make failure unlikely. Instead we should expect that failures, biases etc. can and do happen, and install safeguards to take care of this. The ideal use for such models seems to be exploratory tasks, where they suggest solutions that would be too tedious to find in other way. Then the solutions should however be validated by other means, whether with robust algorithms (ideally, formally verified ones) or through human quality control.

Remark 32. *On the subject of formal verification: one takeaway that is contestable, and only tangentially related to this thesis' subject, is that the Python programming language should be avoided. It is poorly suited for devoloping large systems in a reliable way, let alone validating them. One aspect to this is type safety*[3]*. Although several additions have been made to Python's typing capabilities, they are still mostly ad-hoc and not comparable in power to strong-statically typed languages. Since machine learning architectures are generally built declaratively, a functional language would be a very natural fit [25], such as the Haskell language.*

What on the other hand seems haphardous is to couple black boxes to fragile data systems such as web applications. Here, small misbehaviour could trigger cascades of failure. On a different level this is also something encountered during the Ablation Path experiments: the interaction between the classifier and the heuristic gradient descent algorithm often went astray.

The interactions between the classifier and more principledly designed components meanwhile, such as the SIFT decomposition and the monotonisation-projection, worked with far less troubles.

Another takeaway is that one should not be too quickly satisfied with an explanation to a black-box' decision. In my work, I often tried some modification of the algorithm, ran a handful of examples, got results that looked clear and plausible and though I had "finally fixed it". But there always turned out to be more problems which only surfaced with deeper investigation, and sometimes a plausible saliency map was really only plausible because of some bias effect that made it *less* faithful to the classifier behaviour. This ties to the following – which I consider paramount to keep in mind, so I leave it as the final sentence: *a wrong explanation is worse than no explanation.*

[3] Arguably the most important aspect, since via the Curry-Howard correspondence types are exactly what expresses theorems, with programs as their proofs.

BIBLIOGRAPHY

[1] *Portable Network Graphics (PNG): Functional specification*. Standard. 2004. URL: http://www.iso.org/iso/catalogue_detail.htm?csnumber=29581 (cit. on pp. 88, 94).

[2] Julius Adebayo et al. "Sanity Checks for Saliency Maps". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 9505–9515. URL: http://papers.nips.cc/paper/8160-sanity-checks-for-saliency-maps.pdf (cit. on pp. 17, 18, 23, 28).

[3] I. Ahmad and M.T. Ibrahim. "Image Classification and Retrieval Using Correlation". In: *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. IEEE. DOI: 10.1109/crv.2006.40. (Cit. on p. 19).

[4] Naveed Akhtar et al. "Advances in Adversarial Attacks and Defenses in Computer Vision: A Survey". In: *IEEE Access* 9 (2021), pp. 155161–155196. DOI: 10.1109/access.2021.3127960. (Cit. on p. 35).

[5] Franz Aurenhammer. "Voronoi Diagrams—a Survey of a Fundamental Geometric Data Structure". In: *ACM Computing Surveys* 23.3 (Sept. 1991), pp. 345–405. DOI: 10.1145/116873.116880. (Cit. on p. 107).

[6] Aharon Azulay and Yair Weiss. "Why do deep convolutional networks generalize so poorly to small image transformations?" In: *Journal of Machine Learning Research* 20.184 (2019), pp. 1–25. URL: http://jmlr.org/papers/v20/19-519.html (cit. on p. 136).

[7] David Baehrens et al. "How to Explain Individual Classification Decisions". In: *Journal of Machine Learning Research* 11.61 (2010), pp. 1803–1831. URL: http://jmlr.org/papers/v11/baehrens10a.html (cit. on p. 20).

[8] Stanley Bak. "nnenum: Verification of ReLU Neural Networks With Optimized Abstraction Refinement". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2021, pp. 19–36. DOI: 10.1007/978-3-030-76384-8_2. (Cit. on p. 20).

[9] Emily M. Bender et al. "On the Dangers of Stochastic Parrots". In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Mar. 2021. DOI: 10.1145/3442188.3445922. (Cit. on p. 3).

[10] Tamir Bendory, Alberto Bartesaghi, and Amit Singer. "Single-Particle Cryo-Electron Microscopy: Mathematical Theory, Computational Challenges, and Opportunities". In: *IEEE Signal Processing Magazine* 37.2 (Mar. 2020), pp. 58–76. DOI: 10.1109/msp.2019.2957822. (Cit. on p. 140).

[11] Alex Berg, Jia Deng, and Fei-Fei Li. *Results of the ImageNet Large Scale Visual Recognition Challenge 2010*. URL: https://image-net.org/challenges/LSVRC/2010/results.php (cit. on p. 98).

[12] Sebastian Bordt et al. "The Manifold Hypothesis for Gradient-Based Explanations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2023, pp. 3696–3701 (cit. on p. 92).

[13] Wieland Brendel, Jonas Rauber, and Matthias Bethge. "Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=SyZI0GWCZ (cit. on p. 27).

[14] M. Brown and D. Lowe. "Invariant Features from Interest Point Groups". In: *Procedings of the British Machine Vision Conference 2002*. British Machine Vision Association, 2002. DOI: 10.5244/c.16.23. (Cit. on pp. 103, 107).

[15] Matthew Brown and Sabine Süsstrunk. "Multi-spectral SIFT for Scene Category Recognition". In: *CVPR 2011*. IEEE, June 2011. DOI: 10.1109/cvpr.2011.5995637. (Cit. on p. 103).

[16] W. R. J. Brown and D. L. MacAdam. "Visual Sensitivities to Combined Chromaticity and Luminance Differences*". In: *Journal of the Optical Society of America* 39.10 (Oct. 1949), p. 808. DOI: 10.1364/josa.39.000808. (Cit. on p. 104).

[17] Sébastien Bubeck et al. "Sparks of Artificial General Intelligence: Early experiments with GPT-4". Mar. 2023. URL: https://www.microsoft.com/en-us/research/publication/sparks-of-artificial-general-intelligence-early-experiments-with-gpt-4/ (cit. on p. 19).

[18] Antonin Chambolle and Thomas Pock. "A First-Order Primal-Dual Algorithm for Convex Problems With Applications to Imaging". In: *Journal of Mathematical Imaging and Vision* 40.1 (Dec. 2010), pp. 120–145. DOI: 10.1007/s10851-010-0251-1. (Cit. on p. 61).

[19] Hana Chockler, Daniel Kroening, and Youcheng Sun. "Explanations for Occluded Images". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 1234–1243 (cit. on pp. 10, 22, 26–28, 30, 36, 37, 39, 89, 156).

[20] Taco Cohen and Max Welling. "Group equivariant convolutional networks". In: *International conference on machine learning*. PMLR. 2016, pp. 2990–2999 (cit. on pp. 132, 133).

[21] Piotr Dabkowski and Yarin Gal. "Real Time Image Saliency for Black Box Classifiers". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/0060ef47b12160b9198302ebdb144dcf-Paper.pdf (cit. on pp. 19, 47).

[22] Manuel Dahnert et al. "Panoptic 3D Scene Reconstruction From a Single RGB Image". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 8282–8293. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/46031b3d04dc90994ca317a7c55c4289-Paper.pdf (cit. on p. 85).

[23] Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992. DOI: 10.1137/1.9781611970104. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611970104. (Cit. on p. 95).

[24] J. R. Ehrman, L. D. Fosdick, and D. C. Handscomb. "Computation of Order Parameters in an Ising Lattice by the Monte Carlo Method". In: *Journal of Mathematical Physics* 1.6 (Nov. 1960), pp. 547–558. DOI: 10.1063/1.1703692. (Cit. on p. 26).

[25] Conal Elliott. "The Simple Essence of Automatic Differentiation". In: *Proceedings of the ACM on Programming Languages* 2.ICFP (July 2018), pp. 1–29. DOI: 10.1145/3236765. (Cit. on p. 158).

[26] Mark Everingham et al. "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2 (Sept. 2009), pp. 303–338. DOI: 10.1007/s11263-009-0275-4. (Cit. on pp. 15, 58, 68).

[27] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. "Understanding Deep Networks via Extremal Perturbations and Smooth Masks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 (cit. on pp. 9, 21, 22, 27–31, 36, 37, 39, 43, 54, 68, 70, 74, 77, 123, 156).

[28] Ruth C. Fong and Andrea Vedaldi. "Interpretable Explanations of Black Boxes by Meaningful Perturbation". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017. DOI: 10.1109/iccv.2017.371. (Cit. on pp. 22, 27, 28, 30, 31, 37, 39, 47, 49, 53, 54, 58, 83, 89, 121).

[29] The PyTorch Foundation. *PyTorch machine learning framework*. URL: https://pytorch.org/ (cit. on pp. 57, 119).

[30] Ivan Fursov et al. "A Differentiable Language Model Adversarial Attack on Text Classifiers". In: *IEEE Access* 10 (2022), pp. 17966–17976. DOI: 10.1109/access.2022.3148413. (Cit. on p. 35).

[31] Jan Gerken et al. "Equivariance versus augmentation for spherical images". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 7404–7421 (cit. on pp. 136, 146, 152).

[32] Leilani H. Gilpin et al. "Explaining Explanations: An Overview of Interpretability of Machine Learning". In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, Oct. 2018. DOI: 10.1109/dsaa.2018.00018. (Cit. on p. 19).

[33] Kurt Gödel. "Über Formal Unentscheidbare Sätze Der Principia Mathematica Und Verwandter Systeme I". In: *Monatshefte für Mathematik und Physik* 38-38.1 (Dec. 1931), pp. 173–198. DOI: 10.1007/bf01700692. (Cit. on p. 7).

[34] ODL group. *Operator Discretization Library*. https://odlgroup.github.io/odl (cit. on p. 61).

[35] Alfred Haar. "Zur Theorie Der Orthogonalen Funktionensysteme". In: *Mathematische Annalen* 69.3 (Sept. 1910), pp. 331–371. DOI: 10.1007/bf01456326. (Cit. on p. 95).

[36] Toshiya Hachisuka, Anton S. Kaplanyan, and Carsten Dachsbacher. "Multiplexed Metropolis Light Transport". In: *ACM Transactions on Graphics* 33.4 (July 2014), pp. 1–10. DOI: 10.1145/2601097.2601138. (Cit. on p. 26).

[37] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016 (cit. on p. 68).

[38] Dan Hendrycks et al. *Unsolved Problems in Ml Safety*. 2022. DOI: 10.48550/arXiv.2109.13916. arXiv: 2109.13916 [cs.LG]. (Cit. on p. 19).

[39] Bernease Herman. "The Promise and Peril of Human Evaluation for Model Interpretability". In: (2017). DOI: 10.48550/ARXIV.1711.07414. (Cit. on pp. 10, 19).

[40] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer Feedforward Networks Are Universal Approximators". In: *Neural Networks* 2.5 (Jan. 1989), pp. 359–366. DOI: 10.1016/0893-6080(89)90020-8. (Cit. on p. 134).

[41] *Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB*. Standard. 1999. URL: https://webstore.iec.ch/publication/6169 (cit. on p. 96).

[42] Arieh Iserles et al. "Lie-group Methods". In: *Acta Numerica* 9 (Jan. 2000), pp. 215–365. DOI: 10.1017/s0962492900002154. (Cit. on p. 49).

[43] Viren Jain and Sebastian Seung. "Natural image denoising with convolutional networks". In: *Advances in neural information processing systems* 21 (2008) (cit. on p. 142).

[44] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. 2020. DOI: 10.48550/ARXIV.2001.08361. (Cit. on p. 8).

[45] Guy Katz et al. "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks". In: *Computer Aided Verification*. Springer International Publishing, 2017, pp. 97–117. DOI: 10.1007/978-3-319-63387-9_5. (Cit. on pp. 20, 35).

[46] Stefan Kolek et al. "Cartoon Explanations Of Image Classifiers". In: *Lecture Notes in Computer Science*. Springer Nature Switzerland, 2022, pp. 443–458. DOI: 10.1007/978-3-031-19775-8_26. (Cit. on pp. 28, 38, 95, 156).

[47] Stefan Kolek et al. "Explaining Image Classifiers With Multiscale Directional Image Representation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 18600–18609 (cit. on pp. 38, 87, 95).

[48] A. N. Kolmogorov. "A Refinement of Previous Hypotheses concerning the Local Structure of Turbulence in a Viscous Incompressible Fluid at High Reynolds Number". In: *Journal of Fluid Mechanics* 13.1 (May 1962), pp. 82–85. DOI: 10.1017/s0022112062000518. (Cit. on p. 96).

[49] A. N. Kolmogorov. "Three Approaches to the Quantitative Definition of Information". In: *International Journal of Computer Mathematics* 2.1-4 (1968), pp. 157–168. DOI: 10.1080/00207166808803030 (cit. on p. 89).

[50] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009) (cit. on p. 15).

[51] Thomas S Kuhn. *The structure of scientific revolutions*. Vol. 111. Chicago University of Chicago Press, 1970 (cit. on p. 7).

[52] Y. LeCun et al. "Gradient-based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791. (Cit. on pp. 15, 134).

[53] Jaakko Lehtinen et al. "Noise2Noise: Learning Image Restoration without Clean Data". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 2965–2974. URL: https://proceedings.mlr.press/v80/lehtinen18a.html (cit. on p. 143).

[54] Kornel Lesiński. *pngquant, a command-line utility and a library for lossy compression of PNG images*. URL: https://pngquant.org/ (cit. on p. 94).

[55] Sophus Lie. "Theorie Der Transformationsgruppen I". In: *Mathematische Annalen* 16.4 (Dec. 1880), pp. 441–528. DOI: 10.1007/bf01446218. (Cit. on p. 131).

[56] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48. (Cit. on pp. 15, 25, 35, 58, 68, 104, 122).

[57] Yuanqing Lin et al. "Large-scale Image Classification: Fast Feature Extraction and SVM Training". In: *CVPR 2011*. IEEE, June 2011. DOI: 10.1109/cvpr.2011.5995477. (Cit. on p. 98).

[58] Tony Lindeberg. "Scale-space Theory: a Basic Tool for Analyzing Structures at Different Scales". In: *Journal of Applied Statistics* 21.1-2 (1994), pp. 225–270. DOI: 10.1080/757582976 (cit. on p. 97).

[59] D.G. Lowe. "Object Recognition from Local Scale-invariant Features". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. IEEE, 1999. DOI: 10.1109/iccv.1999.790410. (Cit. on pp. 10, 86, 97, 103, 107).

[60] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110. DOI: 10.1023/b:visi.0000029664.99615.94. (Cit. on pp. 102, 103, 113).

[61] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf (cit. on p. 25).

[62] Aravindh Mahendran and Andrea Vedaldi. "Visualizing Deep Convolutional Neural Networks Using Natural Pre-images". In: *International Journal of Computer Vision* 120.3 (May 2016), pp. 233–255. DOI: 10.1007/s11263-016-0911-8. (Cit. on pp. 21, 104).

[63] Stéphane Mallat. "Understanding Deep Convolutional Networks". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering*

Sciences 374.2065 (2016), p. 20150203. DOI: 10.1098/rsta.2015.0203. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2015.0203. (Cit. on pp. 135, 137).

[64] Pierre Marage and Grégoire Wallenborn. "The Debate between Einstein and Bohr, or How to Interpret Quantum Mechanics". In: *The Solvay Councils and the Birth of Modern Physics*. Birkhäuser Basel, 1999, pp. 161–174. DOI: 10.1007/978-3-0348-7703-9_10. (Cit. on p. 7).

[65] Per Martin-Löf. "An Intuitionistic Theory of Types: Predicative Part". In: *Logic Colloquium '73, Proceedings of the Logic Colloquium*. Elsevier, 1975, pp. 73–118. DOI: 10.1016/s0049-237x(08)71945-1. (Cit. on p. 105).

[66] Jean Jacques Moreau. "Fonctions convexes duales et points proximaux dans un espace hilbertien". In: *Comptes rendus hebdomadaires des séances de l'Académie des sciences* 255 (1962), pp. 2897–2899 (cit. on p. 51).

[67] J. Morlet et al. "Wave Propagation and Sampling Theory—Part I: Complex Signal and Scattering in Multilayered Media". In: *GEOPHYSICS* 47.2 (Feb. 1982), pp. 203–221. DOI: 10.1190/1.1441328. (Cit. on p. 95).

[68] Hans Z. Munthe-Kaas. "Groups and Symmetries in Numerical Linear Algebra". In: *Lecture Notes in Mathematics*. Springer International Publishing, 2016, pp. 319–406. DOI: 10.1007/978-3-319-49887-4_5. (Cit. on p. 131).

[69] Emmy Noether. "Invariante Variationsprobleme". ger. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1918 (1918), pp. 235–257. URL: http://eudml.org/doc/59024 (cit. on p. 130).

[70] Stephen M Omohundro. "The basic AI drives". In: *Proceedings of the First AGI Conference*. Vol. 171. 2008, pp. 483–492 (cit. on p. 35).

[71] *OpenAI ChatGPT*. https://chat.openai.com (cit. on p. 3).

[72] Leonardo Petrini et al. "Relative stability toward diffeomorphisms indicates performance in deep nets". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 8727–8739. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/497476fe61816251905e8baafdf54c23-Paper.pdf (cit. on p. 137).

[73] Vitali Petsiuk, Abir Das, and Kate Saenko. "RISE: Randomized Input Sampling for Explanation of Black-box Models". In: *British Machine Vision Conference (BMVC)*. 2018. URL: http://bmvc2018.org/contents/papers/1064.pdf (cit. on pp. 22, 26–28, 30, 31, 37, 39, 46, 58, 66, 70, 89, 92, 103, 121, 125).

[74] Karl Popper. *Logik der Forschung. Zur Erkenntnistheorie der Modernen Naturwissenschaft*. Vol. 9. Springer-Verlag Wien GmbH, 1935 (cit. on p. 7).

[75] S. Pyatykh, J. Hesser, and Lei Zheng. "Image Noise Level Estimation by Principal Component Analysis". In: *IEEE Transactions on Image Processing* 22.2 (Feb. 2013), pp. 687–699. DOI: 10.1109/tip.2012.2221728. (Cit. on p. 103).

[76] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016 (cit. on p. 29).

[77] P. Revathi and M. Hemalatha. "Classification of Cotton Leaf Spot Diseases Using Image Processing Edge Detection Techniques". In: *2012 International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET)*. IEEE, Dec. 2012. DOI: 10.1109/incoset.2012.6513900. (Cit. on p. 86).

[78] Z.A. Ripstein and J.L. Rubinstein. "Processing of Cryo-EM Movie Data". In: *Methods in Enzymology*. Elsevier, 2016, pp. 103–124. DOI: 10.1016/bs.mie.2016.04.009. (Cit. on p. 139).

[79] Paul L. Rosin and Geoff A.W. West. "Salience Distance Transforms". In: *Graphical Models and Image Processing* 57.6 (Nov. 1995), pp. 483–521. DOI: 10.1006/gmip.1995.1041. (Cit. on pp. 17, 86).

[80] Cynthia Rudin. "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead". In: *Nature Machine Intelligence* 1.5 (May 2019), pp. 206–215. DOI: 10.1038/s42256-019-0048-x. (Cit. on pp. 6, 10, 21, 22, 129, 157).

[81] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-propagating Errors". In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. DOI: 10.1038/323533a0. (Cit. on p. 53).

[82] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (Apr. 2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y. (Cit. on pp. 15, 40, 58, 86, 98, 104, 109, 120).

[83] *CIE Colorimetry - Part 4: 1976 L*a*b* Colour Space*. Standard. 1976. URL: https://cie.co.at/publications/colorimetry-part-4-cie-1976-lab-colour-space-0 (cit. on p. 96).

[84] Justus Sagemüller. *PyTorch implementation of the Ablation Path Saliency method*. https://github.com/leftaroundabout/ablation-paths-pytorch. 2023 (cit. on p. 65).

[85] Justus Sagemüller and Olivier Verdier. "Ablation Path Saliency". In: To appear in the proceedings of the xAI World Conference. 2023. arXiv: 2209.12459 [cs.CV]. URL: https://xaiworldconference.com/2023 (cit. on pp. 39, 89).

[86] A.D. Sappa and M. Devy. "Fast Range Image Segmentation by an Edge Detection Strategy". In: *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. IEEE Comput. Soc. DOI: 10.1109/im.2001.924460. (Cit. on pp. 86, 94).

[87] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017. DOI: 10.1109/iccv.2017.74. (Cit. on pp. 10, 21, 33, 70, 93).

[88] C.E. Shannon. "Communication in the Presence of Noise". In: *Proceedings of the IRE* 37.1 (Jan. 1949), pp. 10–21. DOI: 10.1109/jrproc.1949.232969. (Cit. on p. 88).

[89] L. S. Shapley. "17. A Value for N-Person Games". In: *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press, Dec. 1953, pp. 307–318. DOI: 10.1515/9781400881970-018. (Cit. on p. 25).

[90] Allan Silverman. "Plato's Middle Period Metaphysics and Epistemology". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Fall 2022. Metaphysics Research Lab, Stanford University, 2022 (cit. on p. 7).

[91] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6034 (cit. on pp. 17, 20, 32, 68).

[92] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*. 2015 (cit. on p. 122).

[93] L. Sirovich and M. Kirby. "Low-dimensional Procedure for the Characterization of Human Faces". In: *Journal of the Optical Society of America A* 4.3 (Mar. 1987), p. 519. DOI: 10.1364/josaa.4.000519. (Cit. on p. 92).

[94] Samuel L. Smith and Quoc V. Le. "A Bayesian Perspective on Generalization and Stochastic Gradient Descent". In: *International Conference on Learning Representations*. 2018 (cit. on p. 48).

[95] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. "Visualizing the Impact of Feature Attribution Baselines". In: *Distill* 5.1 (Jan. 2020). DOI: 10.23915/distill.00022. (Cit. on pp. 22, 30).

[96] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 3319–3328. URL: https://proceedings.mlr.press/v70/sundararajan17a.html (cit. on pp. 10, 18, 23, 31, 33, 37, 39, 49, 52).

[97] Teo Susnjak. *ChatGPT: The End of Online Exam Integrity?* 2022. DOI: 10.48550/ARXIV.2212.09292. (Cit. on p. 3).

[98] Richard Sutton. *The Bitter Lesson*. http://www.incompleteideas.net/IncIdeas/BitterLesson.html. 2019 (cit. on p. 8).

[99] Christian Szegedy et al. "Intriguing properties of neural networks". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6199 (cit. on pp. 17, 27, 35).

[100] *Digital Compression and Coding of Continuous-tone Still Images – Requirements and Guidelines*. Standard. 1992. URL: https://www.w3.org/Graphics/JPEG/itu-t81.pdf (cit. on pp. 88, 94, 103).

[101] Florian Tramer et al. "On Adaptive Attacks to Adversarial Example Defenses". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1633–1645. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/11f38f8ecd71867b42433548d1078e38-Paper.pdf (cit. on p. 35).

[102] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: https://homotopytypetheory.org/book, 2013 (cit. on p. 7).

[103] Andrea Vedaldi. *Understanding Deep Networks via Extremal Perturbations and Smooth Masks*. https://github.com/facebookresearch/TorchRay. 2019 (cit. on p. 68).

[104] Ge Wang, Jong Chul Ye, and Bruno De Man. "Deep Learning for Tomographic Image Reconstruction". In: *Nature Machine Intelligence* 2.12 (Dec. 2020), pp. 737–748. DOI: 10.1038/s42256-020-00273-z. (Cit. on p. 141).

[105] J. D. Watson and F. H. C. Crick. "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid". In: *Nature* 171.4356 (Apr. 1953), pp. 737–738. DOI: 10.1038/171737a0. (Cit. on p. 139).

[106] Miles N. Wernick and G. Michael Morris. "Image Classification at Low Light Levels". In: *Journal of the Optical Society of America A* 3.12 (Dec. 1986), p. 2179. DOI: 10.1364/josaa.3.002179. (Cit. on p. 19).

[107] Adam White et al. "Contrastive Counterfactual Visual Explanations With Overdetermination". In: *Machine Learning* 112.9 (May 2023), pp. 3497–3525. DOI: 10.1007/s10994-023-06333-w. (Cit. on p. 86).

[108] Alfred North Whitehead and Bertrand Russell. *Principia Mathematica to *56*. Cambridge University Press, Sept. 1997. DOI: 10.1017/cbo9780511623585. (Cit. on p. 7).

[109] E. T. Whittaker. "XVIII.—On the Functions Which Are Represented by the Expansions of the Interpolation-Theory". In: *Proceedings of the Royal Society of Edinburgh* 35 (1915), pp. 181–194. DOI: 10.1017/s0370164600017806. (Cit. on p. 87).

[110] Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. The MIT press, 1949 (cit. on p. 142).

[111] Eugene P. Wigner. "The Unreasonable Effectiveness of Mathematics in the Natural Sciences. Richard Courant Lecture in Mathematical Sciences Delivered at New York University, May 11, 1959". In: *Communications on Pure and Applied Mathematics* 13.1 (1960), pp. 1–14. DOI: 10.1002/cpa.3160130102. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160130102. (Cit. on p. 8).

[112] Julia K. Winkler et al. "Association Between Surgical Skin Markings in Dermoscopic Images and Diagnostic Performance of a Deep Learning Convolutional Neural Network for Melanoma Recognition". In: *JAMA Dermatology* 155.10 (Oct. 2019), p. 1135. DOI: 10.1001/jamadermatol.2019.1735. (Cit. on p. 29).

[113] A. Witkin. "Scale-space Filtering: A New Approach to Multi-scale Description". In: *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Institute of Electrical and Electronics Engineers. DOI: 10.1109/icassp.1984.1172729. (Cit. on p. 98).

[114] J C Wyngaard. "Measurement of Small-scale Turbulence Structure With Hot Wires". In: *Journal of Physics E: Scientific Instruments* 1.11 (Nov. 1968), pp. 1105–1108. DOI: 10.1088/0022-3735/1/11/310. (Cit. on p. 96).

[115] Hanyu Xiang et al. "Deep Learning for Image Inpainting: A Survey". In: *Pattern Recognition* 134 (Feb. 2023), p. 109046. DOI: 10.1016/j.patcog.2022.109046. (Cit. on p. 85).

[116] Peng Yong et al. "Total Variation Regularization for Seismic Waveform Inversion Using an Adaptive Primal Dual Hybrid Gradient Method". In: *Inverse Problems* 34.4 (Mar. 2018), p. 045006. DOI: 10.1088/1361-6420/aaaf8e. (Cit. on p. 28).

[117] Ivana Zeger et al. "Grayscale Image Colorization Methods: Overview and Evaluation". In: *IEEE Access* 9 (2021), pp. 113326–113346. DOI: 10.1109/access.2021.3104515. (Cit. on p. 104).

[118] Jianming Zhang et al. "Top-Down Neural Attention by Excitation Backprop". In: *International Journal of Computer Vision* 126.10 (Dec. 2017), pp. 1084–1102. DOI: 10.1007/s11263-017-1059-x. (Cit. on pp. 29, 66, 70, 123, 142).

[119] Richard Zhang. "Making Convolutional Networks Shift-Invariant Again". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 7324–7334. URL: https://proceedings.mlr.press/v97/zhang19a.html (cit. on pp. 136, 146).

[120] Zhizhen Zhao and Amit Singer. "Rotationally Invariant Image Representation for Viewing Direction Classification in Cryo-EM". In: *Journal of Structural Biology* 186.1 (2014), pp. 153–166. ISSN: 1047-8477. DOI: 10.1016/j.jsb.2014.03.003. (Cit. on p. 141).

[121] Bolei Zhou et al. "Learning Deep Features for Discriminative Localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016 (cit. on p. 21).

APPENDIX

Proof of equality between average gradient and boundary-normal

These are the technical details behind Lemma 1, written by Olivier Verdier.

Define the *logistic function* σ by

$$\sigma(y) := \frac{1+\tanh(y/2)}{2} = \frac{\exp(y)}{1+\exp(y)}$$

Consider a vector space V and a smooth real valued function $G\colon V \to \mathbb{R}$.

This function can be *saturated* with the logistic function by defining the function $F_\beta\colon V \to \mathbb{R}$ as

$$F_\beta(x) = \sigma(\beta G(x))$$

In the limit when $\beta \to \infty$, the function F_β takes only the values one and zero at points $x \in V$ whenever $G(x) \neq 0$, hence the name of "saturation".

Fix two points x_{-1} and x_1 in V. Define the segment

$$[x_{-1}, x_1]_t := ((t+1)x_1 + (t-1)x_0)/2$$

Define also for convenience the points $x_t := [x_{-1}, x_1]_t$, for $t \in [-1, 1]$.

For any smooth function F, we define the *average differential* $A(F) \in V^*$ to be

$$\langle A(F), v\rangle := \int_{-1}^{1} \langle dF, v\rangle_{x_t}\, dt \qquad v \in V$$

Define the function $g(t) := G(x_t)$. We now assume that the function G is such that $g(-1) < 0$ and $g(1) > 0$, that g crosses zero at only one point $t^* \in (-1, 1)$, and that $g'(t^*) > 0$.

The set $\{G(x) = 0\}$ is the *decision boundary*.

In terms of average gradient, the following result states that, in the saturation limit $\beta \to \infty$, *the average gradient is perpendicular to the decision boundary*.

Lemma 11. *With the definitions and assumptions above, if a vector v is tangent to the decision boundary, that is, if $\langle dG, v\rangle_{x^*} = 0$, then*

$$\lim_{\beta\to\infty} \langle A(F_\beta), v\rangle = 0.$$

Proof. First, we make the assumption that g is strictly increasing on $[-1, 1]$, if not, one can simply restrict to a smaller interval by choosing other points x_{-1} and x_1.

We now use the function G as one coordinate variable denoted by X and complete it to a full set of coordinates, collectively denoted by X, Y. We further choose it so that the points x_t have coordinates $(X_t, 0)$ (i.e, all coordinates other than the first vanish on the segment $[x_{-1}, x_1]$). By definition, in those coordinates, $G(X, Y) = X$. The crossing is thus at the point $(X_{t^*}, 0)$ in these coordinates.

The parallel transport of a vector v along the segment x_{-1}, x_1 in those variables is no longer trivial. It is a vector with coordinates $v_X(t), v_Y(t)$ at the point x_t.

In the coordinates X, Y, the function F writes $F(X, Y) = \sigma(\beta X)$, so the differential is just $dF = \beta\sigma'(\beta X)dX$, and the average differential thus

$$\langle A(F), v\rangle = \int_{-1}^{1} \beta\sigma'(\beta X_t)v_X(t)dt$$

Define the function $f(t) = v_X(x_t)$. The main assumption above amounts to $f(t^*) = 0$ and $X_{t^*} = 0$.

The result now follows from Lemma 12, if, without loss of generality, we assume that $t^* = 0$.

□

Lemma 12. *Suppose that a positive continuous function $C\colon \mathbb{R} \to \mathbb{R}$ has a finite integral. Consider a continuous function $f\colon [-1, 1]$ such that $f(0) = 0$, and a strictly increasing C^1 function $g\colon [-1, 1] \to \mathbb{R}$ such that $g(0) = 0$. Then we have*

$$\lim_{\beta\to\infty} \int_{-1}^{1} \beta C(\beta g(t))f(t)dt = 0$$

Proof. With the change of variable $u = \beta t$, rewrite the integral as

$$I(\beta) = \int_{-\beta}^{\beta} C(\beta g(u/\beta))f(u/\beta)du$$

and then for any fixed u_0 such that $0 < u_0 < \beta$, as

$$I = I_0(\beta, u_0) + I_1(\beta, u_0)$$

with

$$I_0(\beta, u_0) \coloneqq \int_{|u|<u_0} C(\beta g(u/\beta))f(u/\beta)du$$

and

$$I_1(\beta, u_0) \coloneqq \int_{u_0<|u|<\beta} C(\beta g(u/\beta))f(u/\beta)du$$

Focus first on the integral $J \coloneqq \int_{u_0<u<\beta} C(\beta g(u/\beta))f(u/\beta)du$. We assume that g' is strictly increasing, so g' is bounded below on $[-1, 1]$ by a positive $m > 0$, in other words, $g'(t) > m$, or, $g'(t)/m \geqslant 1$. Now since f is continous on $[-1, 1]$ it is bounded, so $|f(t)| \leqslant M$. We thus obtain

$$J \leqslant M \int_{u_0<u<\beta} C(\beta g(u/\beta))\frac{g'(u/\beta)}{m}du$$

Changing variable with $w = \beta g(u/\beta)$ now gives

$$J \leqslant \frac{M}{m} \int_{\beta g^{-1}(u_0/\beta)<w<\beta g^{-1}(1)} C(w)dw$$

which is further bounded by

$$J \leqslant \frac{M}{m} \int_{\beta g^{-1}(u_0/\beta) < w} C(w) \mathrm{d}w$$

Now, since $g(0) = 0$ and by integrating the inequality $g'(t) \leqslant 1/\lambda$ (since g' is bounded), we obtain for positive t that $t \geqslant \lambda g(t)$, so $\beta g^{-1}(u/\beta) \geqslant \lambda u_0$, and

$$J \leqslant \frac{M}{m} \int_{\lambda u_0 < w} C(w) \mathrm{d}w$$

Crucially, this bound in independent of β, so $I_1(\beta, u_0)$ goes to zero uniformly in β. This means that $I_1(\beta, u_0)$ is arbitrarily small for all $\beta > 0$, as long as $u_0 > U$ for some value $U > 0$.

Now use the bound on C, that is, $|C(x)| \leqslant C_0$ for $x \in \mathbb{R}$. We get $|I_0| \leqslant C_0 \int_{u \leqslant u_0} f(u/\beta) \mathrm{d}u$, and this goes to zero for any fixed u_0 since f is continuous and $f(0) = 0$. □