

Maskinell Transkribering av Snapchat Skjermopptak
Automatic Transcription of Snapchat Screen Recordings
Systemdokumentasjon

Versjon 3.3

Dokumentet er basert på Systemdokumentasjon utarbeidet ved NTNU. Revisjon og tilpasninger til bruk ved IDER, DATA-INF utført av Carsten Gunnar Helgesen, Svein-Ivar Lillehaug og Per Christian Engdal. Dokumentet finnes også i engelsk utgave.



**Høgskulen
på Vestlandet**

Fakultet for ingeniør- og naturvitenskap

Institutt for datateknologi, elektroteknologi og realfag

REVISJONSHISTORIE

Dato	Versjon	Beskrivelse	Forfatter
01/feb/22	1.0	Prosjektstruktur og klassediagram	Vilde, Thea, Anne
01/mar/22	1.1	Installasjon og kjøring	Vilde
15/mar/22	2.0	Sikkerhet, Databasemodell og Server-tjenester	Vilde
23/mar/22	2.1	Dokumentasjon av kildekode	Thea
01/apr/22	3.0	Arkitektur og prosjektstruktur	Vilde, Thea, Anne
17/apr/22	3.1	Oppdatert installasjon	Thea
15/mai/22	3.2	Oppdatering basert på nye krav	Vilde, Thea, Anne
20/mai/22	3.3	Endringer før siste levering	Anne, Thea, Vilde



INNHALDSFORTEGNELSE

1	INNLEDNING.....	1
2	ARKITEKTUR.....	2
3	PROSJEKTSTRUKTUR.....	4
4	KLASSEDIAGRAM.....	6
5	DATABASEMODELL.....	7
6	SERVER-TJENESTER.....	8
7	SIKKERHET.....	9
8	INSTALLASJON OG KJØRING.....	10
9	DOKUMENTASJON AV KILDEKODE.....	11
10	KONTINUERLIG INTEGRASJON OG TESTING.....	13
11	BRUKERMANUAL.....	14
12	REFERANSER.....	19

1 INNLEDNING

Systemdokumentasjon er skrevet for å gi en overordnet beskrivelse av systemet for systemutviklere. Systemdokumentasjonen kan brukes som et hjelpemiddel for å navigere i systemet og finne informasjon hvis deler av systemet er uklart. Det skal i systemdokumentasjonen beskrives hvordan systemet er designet og utviklet. Det vil også bli laget en brukermanual som vil gi en anvisning til hvordan programmet fungerer og hvordan bruker kan selv gjøre endringer.

2 ARKITEKTUR

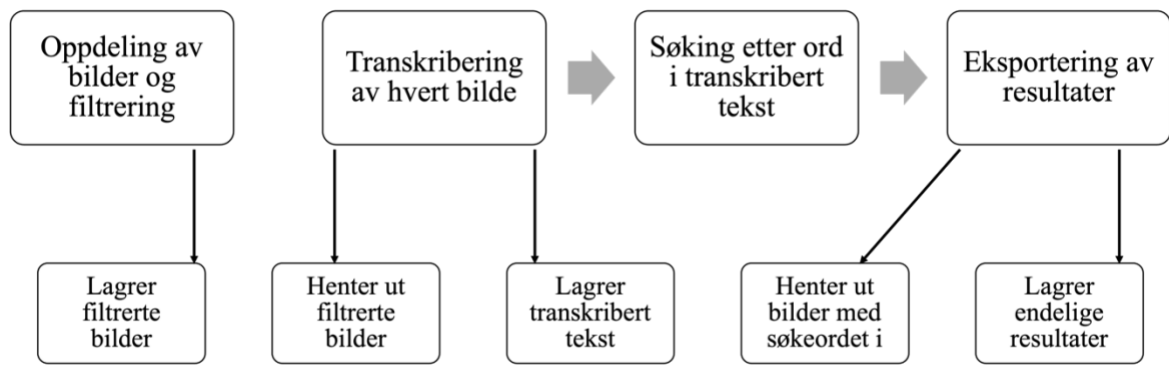
Programvarearkitekturen for programvaren er delt i tre lag, som vist i Figur 1. Lagene i programvarearkitekturen er brukersamhandling, prosesser i systemet og lagring av resultater og midlertidig informasjon.



Figur 1 Visualisering av programvarearkitekturen

Brukersamhandling består av to deler; valg av parametere og spesifisering av transkriberingsområde. Det første som skjer når programvaren kjøres er at brukeren må velge filen eller mappen som skal transkriberes og oppgi søkeordet som skal finnes igjen i videoen eller bildene. Før bakgrunnsprosessene starter å kjøre vil programvaren gi ut et bilde fra mappen eller første bilde i videoen, med mulighet for brukeren å velge område i. Området som velges av brukeren er det området av resten av bildene som brukes i bakgrunnsprosessene.

Bakgrunnsprosessene skjer i fire kategorier og samarbeider tett med lagringsdelen av arkitekturen. Samarbeidet mellom disse to lagene er beskrevet i Figur 2. Etter at filtreringsprosessen er ferdig lagres de utvalgte bildene i en egen mappe. Transkriberingsprosessen henter bildene fra denne mappen og transkriberer dem individuelt. Transkriberingsprosessen lagrer den transkriberte teksten fra alle bildene sammen i en tekstfil og sender en liste over transkribering fra hvert bilde videre til søkeprosessen. Søkeprosessen bruker transkriberingslisten for å finne hvilke bilder som inneholder søkeordet, og sender filnavnene videre til eksporteringsprosessen. Eksporteringsprosessen henter så ut bildene hva der filtreringsprosessen lagret dem og samler dem til en PDF-fil som også lagres i en mappe.



Figur 2 Kommunikasjon mellom bakgrunnsprosesser og lagring

3 PROSJEKTSTRUKTUR

Kildekode og filer er strukturert i en overordnet mappe som består av flere mapper ut fra funksjonalitet og andre filer, som vist i Figur 3. Mappene og filene lengst til venstre i figuren viser første nivået i den overordnede mappen. Her er kildekode og filer sortert videre i mapper for å holde det strukturert. Unntaket er de to filene ‘installModules.bat’ og ‘Main.py’. ‘installModules.bat’ er installasjonsfilen som må kjøres før programvaren kan startes og ‘Main.py’ er kildekode for å starte alle bakgrunnsprosessene.

Det neste nivået beskriver innholdet i mappene på det første nivået. Mappen ‘Results’ brukes for å lagre resultatene til programvaren, og de resterende mappen lagrer kildekode som er strukturert i klasser. Navnet på mappene i ‘Results’ opprettes basert på navnet på videoen eller mappen som skal transkribes, ordet som det søkes etter og datoen når denne instansen kjøres. Under denne mappen lagres den fullstendig transkriberte teksten, en PDF-fil med bildene som inneholder søkeordet og de filtrerte rammene med tilnærmet unikt innhold.

GUI	Crop.py	
	gui.py	
Results	<videonavn>_<søkeord>_<kjøredato>	fullTranscript.txt matchingframes_<søkeord>.pdf unikframe_0001.jpg ⋮
SplitVideo	Filter.py	
	Folder.py	
	Video.py	
Testing	TestCorr.py	
	TestWords.py	
Transcribing	transkrib.py	
	installModules.bat	
	Main.py	

Figur 3 Prosjektstrukturen til programvaren

Strukturen i kjørbare kode varierer basert på om klassen har statiske metoder eller objekt metoder. Hvis klassen har statiske metoder, trenger den ikke konstruktør og det er ikke nødvendig å opprette et objekt av klassen for å kjøre metodene. For klasser med objekt metoder er det motsatt.

Figur 4 viser generell struktur for de to strukturversjonene. Figur 4a viser strukturen for kildekode med statiske metoder. Før en metode defineres og handling implementeres må det festes en 'staticmethod' dekorator slik at programvaren skal forstå at dette er en statisk metode. I Figur 4b vises strukturen for kildekode med objekt metoder. De globale variablene er markert i grått da dette ikke er implementert i alle objekt klassene. Det må defineres en konstruktør som i Python defineres med ' __init__ '.

```
<import pakker>

class Klassenavn:
    @staticmethod
    def metodeNavn(*args):
        <kode>
```

(a) Struktur for klasser med statiske metoder

```
<import pakker>
<globale variabler>

class Klassenavn:
    def __init__(self, *args):
        <kode>

    def metodeNavn(args*):
        <kode>
```

(b) Struktur for klasser med objekt metoder

Figur 4 Generell struktur for kildekoden til programvaren

4 KLASSEDIAGRAM

Klassediagrammet brukes for å vise elementer og funksjoner i kildekoden og hvordan disse kommuniserer, som vist i Figur 5. Programvaren starter i GUI klassen som oppretter brukergrensesnittet og videre kjører Main som utfører prosessene i programvaren.

Main kommuniserer videre med de andre klassene for å løse ulike oppgaver som å dele video til bilder (utført av Video klassen), filtrering av bildene (utført av Filter klassen) og transkribering (utført av Transcribe klassen).



Figur 5 Klassediagram for programvaren

5 DATABASEMODELL

Ikke relevant for prosjektet.

6 SERVER-TJENESTER

Ikke relevant for prosjektet.

7 SIKKERHET

Ettersom programmet blir kjørt av etterforskere i DPA uten kommunikasjon med andre servere og programmet kan kjøres uten internett er det ikke behov for beskyttelse mot uautorisert tilgang.

Men etter ønske fra oppdragsgiver er det inkludert MD5 hash for fil før og etter transkribering slik at etterforsker kan sjekke om det er blitt gjort endringer i videofilen under kjøring av programmet.

8 INSTALLASJON OG KJØRING

Installasjonen tar utgangspunkt i at maskinen har Python 3 installert sammen med de inkluderte modulene `os`, `datetime` og `tkinter`. Prosjektet lastets ned fra github som en ZIP-fil og pakkes deretter ut. For maskiner med Windows som operativ system kan Python modulene som ikke er inkludert i Python 3 installeres ved å kjøre Batch-filen «`installModules.bat`» i ledeteksten. I ledeteksten må brukeren navigere til prosjektets plassering, slik at kjøring kommandoen ‘`dir`’ gir ut innholdet i prosjektmappen, inkludert batch-filen, som vist i Figur 6.

```
C:\Users\vilde\Bachlor-Mappe\Bachelor>dir
Volume in drive C is Windows
Volume Serial Number is 08D0-C11E

Directory of C:\Users\vilde\Bachlor-Mappe\Bachelor

20.05.2022  16:41    <DIR>          .
04.05.2022  12:07    <DIR>          ..
20.05.2022  16:41             14 340 .DS_Store
19.05.2022  19:00    <DIR>          DelOppFilm
19.05.2022  19:00    <DIR>          GUI
19.05.2022  16:42             400 installModules.bat
18.05.2022  19:58             8 208 Main.py
03.02.2022  15:46             142 README.md
22.05.2022  02:01    <DIR>          Results
20.05.2022  13:56             4 264 SearchMain.py
21.05.2022  20:12    <DIR>          Testing
22.05.2022  02:02    <DIR>          Transkribering
22.05.2022  02:01    <DIR>          __pycache__
                5 File(s)          27 354 bytes
                8 Dir(s)  306 748 166 144 bytes free
```

Figur 6 Kommando for å vise mappeinnhold i nåværende lokasjon

Videre kjøres denne filen ved å skrive inn navnet: ‘`installModules.bat`’ for så å trykke enter som vist i Figur 7. Da installeres pakkene som er definert i filen.

```
C:\Users\vilde\Bachlor-Mappe\Bachelor>installModules.bat
```

Figur 7 Kjøring av batch-fil i ledetekst

For maskiner med andre operativsystemer kan pip kommandoene fra «`installModules.bat`» kjøres enkeltvis i kommandolinje prompt.

Installasjon på Windows maskin krever et ekstra steg i installasjonen. Prosjektmappen inneholder en mappe ved navn “*Tesseract-OCR*”, denne mappen må flyttes til lokasjonen “*C:\Program Files*” slik at dens fullstendige sti blir “*C:\Program Files\Tesseract-OCR*”. For andre operativ systemer kan denne mappen slettes fra prosjektmappen.

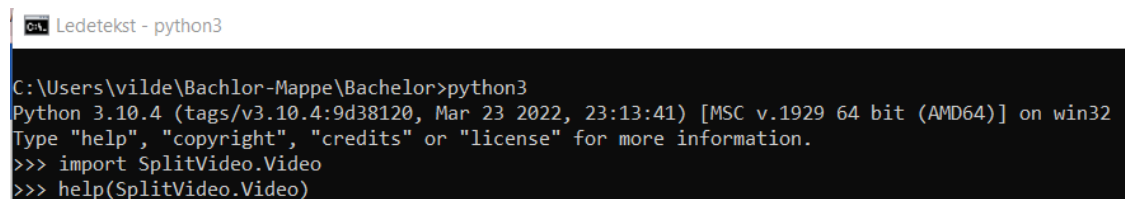
9 DOKUMENTASJON AV KILDEKODE

Kildekoden dokumenteres ved Python docstring med Google Inc formatering (Google, 2022).

For å generere Python Docstring for klasser og metoder i en modul må Python 3 være installert. I terminalen navigeres det til mappen hvor prosjektet befinner seg, så kjøres følgende kommandoene:

python3 – For å komme inn i Python miljøet
import <mappenavn.klassenavn> – For å importere modulen til miljøet
help(<mappenavn.klassenavn>) – For å generere dokumentasjonen til modulen.

Et eksempel på hvordan man får ut dokumentasjonen er vist i Figur 8 hvor dokumentasjonen for klassen 'Video' hentes ut. For at programvaren og henting av dokumentasjon skal fungere må brukeren befinne seg i prosjektmappen ved kjøring og henting av dokumentasjon. Dette medfører at klasser som er under mapper som beskrevet i Kapittel 3 krever en lengre sti, slik som i eksempelet.



```
CA: Ledetekst - python3
C:\Users\vilde\Bachlor-Mappe\Bachelor>python3
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import SplitVideo.Video
>>> help(SplitVideo.Video)
```

Figur 8 Kommandoer i ledeteksten for å generere Docstring

Når man henter ut Docstring dokumentasjonen får man ut informasjonen om metoder og variabler i klassen som vist i Figur 9. Det er mulig å bla videre i dokumentasjonen ved å trykke på 'Enter', og får å gå ut av dokumentasjonen brukes Ctrl+Z.

```

Help on module SplitVideo.Video in SplitVideo:

NAME
  SplitVideo.Video

CLASSES
  builtins.object
    Video

  class Video(builtins.object)
    Video(pathvideo, path_save)

    A class to represent a video,with operations for processing the frames of the video.

    Attributes:
    pathvideo (str):This is where the video is stored.
    path_save (str):

    Methods defined here:

    __init__(self, pathvideo, path_save)
        The constructor for Video class.

    Args:
    pathvideo (str) :This is where the video is stored.
    path_save (str) :This is where frames will be stored after filtering.
    rec (VideoCapture):VideoCapture object representing the video.
    fps (int) : The Frames/second rate.
    noOfFrames (int) : The number of frames in the video.
    duration (float) : The duration of the video.
    width (int) : Widt of frames in the video.
    height (int) : Heigth of the frames in the video.

    divideToFrames(self)
        Function that extracts all frames from the video and filter out frames with similar hash value,
        stores them according to the Video object's path_unique.

    doFiltering(self, dim)
        Method to perform filtering so unique frames can be extracted.

    getFirstFrame(self)
        Extract the first frame from the video.

    infoToJson(self)
        Saves information about the video to a json structure.

    -----
    Data descriptors defined here:
  
```

Figur 9 Docstring for Video-klassen

10 KONTINUERLIG INTEGRASJON OG TESTING

Det er ikke brukt kontinuerlig integrasjon i dette prosjektet

Metoder brukt i programvaren gir varierte resultater grunnet brukerinteraksjon og dynamiske metoder som gjør det vanskelig å forutse korrekte verdier, og på grunnlag av dette er det ikke blitt brukt enhetstester i programvaren. Det anvendes flere metoder fra godt dokumenterte Python pakker som ansees som pålitelige og derfor er det ikke skrevet enhetstester for disse heller.

Det er laget to klasser for å beregne korrekthet til transkriberingsresultatene. For å kunne kjøre disse må brukeren ha tilgang på korrekt transkribert tekst.

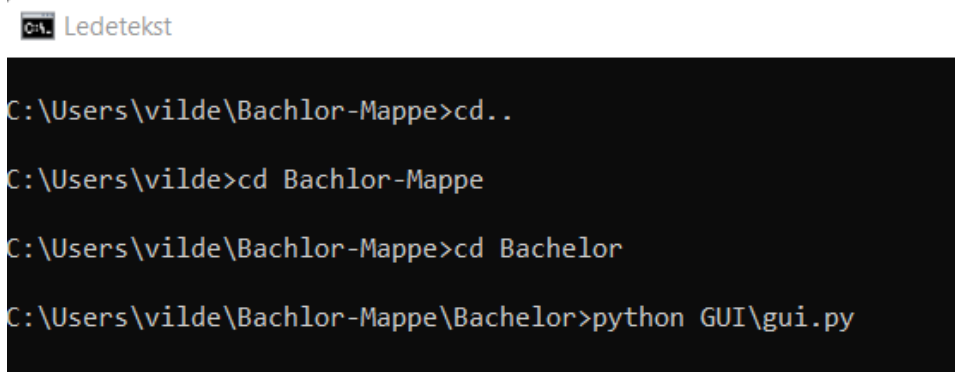
11 BRUKERMANUAL

Forutsetninger for å bruke programvaren:

- Python3 må være installert
- Prosjektet er lastet ned fra Github som en ZIP-fil og pakket ut.

0. Installer nødvendige pakker om de ikke allerede er installert.

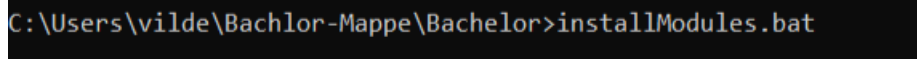
1. Gå inn i Ledetekst programmet (Ledetekst).
2. Naviger til prosjektets plassering i ledeteksten ved hjelpekommandoen `cd`, som vist i Figur 10.



```
C:\Users\vilde\Bachlor-Mappe>cd .
C:\Users\vilde>cd Bachlor-Mappe
C:\Users\vilde\Bachlor-Mappe>cd Bachelor
C:\Users\vilde\Bachlor-Mappe\Bachelor>python GUI\gui.py
```

Figur 10 Navigasjon til prosjektmappen i ledetekst

3. Installer pakkene ved å kjøre: `installModules.bat`, som vist i Figur 11



```
C:\Users\vilde\Bachlor-Mappe\Bachelor>installModules.bat
```

Figur 11 Kjøring av batch-fil

- For maskiner med andre operativsystemer kan pip kommandoene fra «`installModules.bat`» kjøres enkeltvis i kommandolinje prompt.
- Installasjon på Windows maskin krever et ekstra steg i installasjonen. Prosjektmappen inneholder en mappe ved navn “*Tesseract-OCR*”, denne mappen må flyttes til lokasjonen “*C:\Program Files*” slik at dens fullstendige sti blir “*C:\Program Files\Tesseract-OCR*”. For andre operativ systemer kan denne mappen slettes fra prosjektmappen.

1. Hvordan kjøre programmet

1. Gå inn i ledetekst og naviger til prosjektmappen.
2. Skriv inn `python GUI\gui.py`, og trykk enter.

3. Hovedmenyen kommer frem som sett i Figur 12:

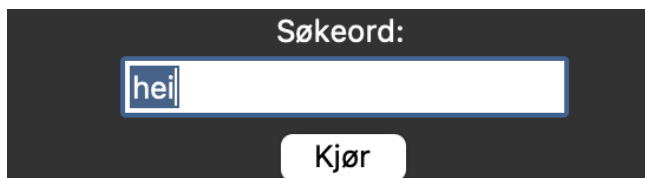


Figur 12 Bilde fra oppstart av programvaren

2. Hva de ulike knappene betyr

- Velg video
 - Her velger man hvilken video man vil transkribere.
- Velg mappe
 - Her velger man hvilken mappe med bilder man vil transkribere
- Kjør
 - Her kjører man selve programmet. Denne kan ikke kjøres før etter man har skrevet inn et søkeord.

3. Hvordan søke



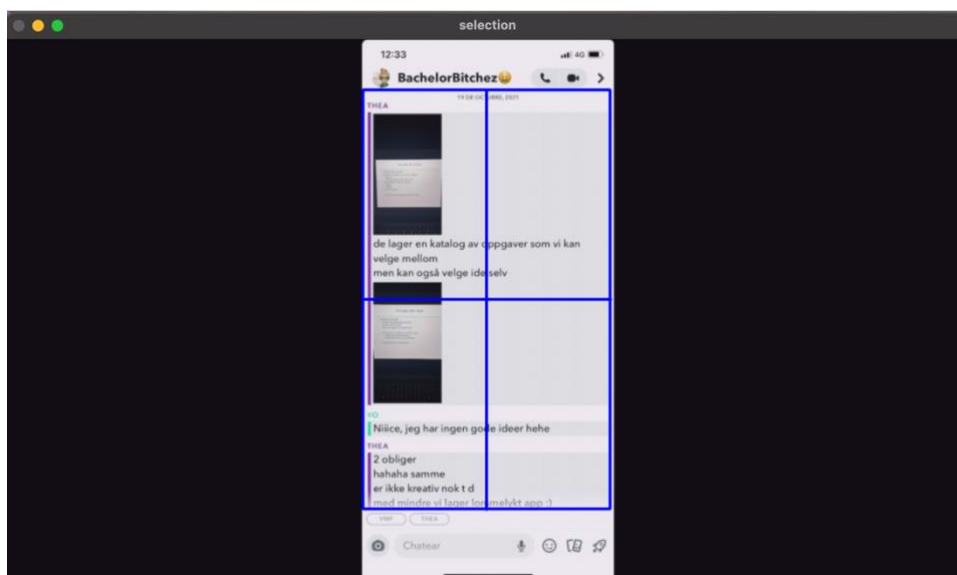
Figur 13 Søkefelt på brukergrensesnittet

- Etter man har valgt video / mappe, er det mulig å søke etter et ord man vil finne igjen i videoen / mappen.
- Dette utføres ved å skrive et ord inn i tekstfeltet som sett i Figur 13
- Trykk kjøør for å fortsette

4. Hvordan få best resultater

- Det første som kommer opp er første skjermbilde fra video / mappen som sett i Figur 14. Dette fordi at det er nødvendig å markere hvor på skjermen man

transkribere fra. Dette hjelper med at man ikke får med unødvendig tekst og/eller tastatur. Brukeren kan da velge området hvor teksten som skal transkriberes er



Figur 14 ROI markering for å definere området som skal transkriberes

- Snapchat bør ikke være i Dark Mode.
- Medium til rolig skrolle-hastighet når det blir tatt opptak av videoen.
 - Unntak om bilder per sekund (FPS) er høy nok
 - Programvaren er kun kjørt på videoer med maks FPS på 60.
- Søkordet inneholder mer enn 2 tegn.
 - Dette er grunnet det uklare regulære uttrykket som brukes
 - Det tillater feil i treff på halvparten av ord-lengden, rundet ned

5. Hvordan finne resultat

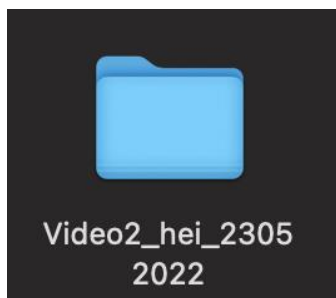
Man ser i ledetekst når programmet har transkribert og utført søking av søkeordet. Når det står "Finished matching!" er programmet ferdig og man kan finne resultatet. Et eksempel er vist i Figur 15. Når programmet er ferdig å kjøre vil en PDF-fil med resultatene åpne

seg automatisk.

```
Select a ROI and then press SPACE or ENTER button!  
Cancel the selection process by pressing c button!  
Filtering ...  
Finished filtering!  
Transcribing ...  
Finished transcribing!  
Matching the word with the text...  
Finished matching!  
TID: 0:04:19.930833
```

Figur 15 Ledetekst etter at programvaren har kjørt

Resultatet ligger som en mappe i Prosjektmappen under *Results*. Her vil navnet på resultatet være “*VideoNavn_Søkeord_Dato*” som synlig i Figur 16. I mappen finner man den fullstendige transkriberte teksten sammen med bildene hentet ut fra videoen og PDF-filen som viser resultatene fra søket, som vist i Figur 17.



Figur 16 Bilde som viser navnet på mappen hvor resultatene lagres



Figur 17 Innholdet i resultatmappen til en video

6. Tilpassing av programvaren:

Endre språk

For å endre språk vil det være nødvendig å endre selve koden. Dette må gjøres ved å gå i Python klassen `Main.py` på linje 127. Standard innstilling vises i Figur 18. Her er det ikke spesifisert språk fordi norsk brukes som standarden. Ved å sende inn et annet parameter i metoden på linje 127 vil alfabetet til dette språket brukes i transkriberingen. For eksempel vil endring av linje 127 til `transkr = t(result_path, «ara»)` gi transkribering av det arabiske alfabetet (tesseract-ocr, 2022). OBS: Noen språk kan kreve ekstra

installasjon.

```
126     print("Transcribing ...")
127     transkr = t(result_path)
128     print("Finished transcribing!")
```

Figur 18 Plassering i koden hvor det er mulig å endre språk

Endre skalering på ROI

Dette kan endres i crop.py klassen på linje 20, som vist i Figur 19. Standard i koden er at vinduet alltid settes til 1000 piksler, men ved å endre width=1000 kan man gjøre vinduet større eller mindre.

```
20     frame = imutils.resize(frame,width=1000)
```

Figur 19 Plassering i koden for å endre skalering av ROI vinduet

Hvordan legge til flere aksepterte filtyper

For å endre/legge til filtyper må man gå inn å endre selve koden. Dette må gjøres i Python klassen gui.py på linje 18 og 19. Standard er at kun '.mp4' filer er akseptert som sett i Figur 20. Det er mulig å liste flere video-filtyper i denne listen, programvaren bruker også ffmpeg i kombinasjon med andre pakker, som tillater en større mengde filtyper.

Her kan man legge til og endre hvilke filtyper man vil ha. Om man heller vil ha MOV eller en AVI type, kan man skrive .avi.

```
17     #Global variables
18     typer = (".mp4")
```

Figur 20 Lovlige filtyper for video

12 REFERANSER

- Google. (2022, April 19). *Comments and Docstrings*. Hentet fra Github:
<https://github.com/google/styleguide/blob/gh-pages/pyguide.md#38-comments-and-docstrings>
- tesseract-ocr. (2022, Mai 17). *Languages supported in different versions of Tesseract*. Hentet fra Tesseract documentation: <https://tesseract-ocr.github.io/tessdoc/Data-Files-in-different-versions.html>