```
from ast import ExtSlice
from asyncio.windows_events import NULL
import threading
from time import time
import cv2
import numpy as np
from matplotlib import pyplot as plt
import os
from pandas import notnull
import tensorflow as tf
from object_detection.utils import label_map_util
from object_detection.utils import visualization_utils as viz_utils
from object_detection.builders import model_builder
from object_detection.utils import config_util
import time
import serial

#Object detection, line(20-74), FOUND HERE:
https://github.com/nicknochnack/TFODCourse/blob/main/2.%20Training%20and%20Detection.ipynb

kameraValg = 1
#CUSTOM_MODEL_NAME = 'efficientdet_d1_coco17_tpu-32_V11_35000s'
#PRETRAINED_MODEL_NAME = 'efficientdet_d1_coco17_tpu-32'
#PRETRAINED_MODEL_URL =
'http://download.tensorflow.org/models/object_detection/tf2/20200711/efficientdet_d1_coco17_tpu-32.tar.gz'
#TF_RECORD_SCRIPT_NAME = 'generate_tfrecord.py'
#LABEL_MAP_NAME = 'label_map.pbtxt'


CUSTOM_MODEL_NAME = 'ssd_mobilenet_v2_fpnlite_640x640_V4_sveis_40000s'
PRETRAINED_MODEL_NAME = 'ssd_mobilenet_v2_fpnlite_640x640_coco17_tpu-8'
PRETRAINED_MODEL_URL =
'http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpnlite_640x640_coco17_tpu-8.tar.
gz'
TF_RECORD_SCRIPT_NAME = 'generate_tfrecord.py'
LABEL_MAP_NAME = 'label_map.pbtxt'

paths = {
    'WORKSPACE_PATH': os.path.join('Tensorflow', 'workspace'),
    'SCRIPTS_PATH': os.path.join('Tensorflow','scripts'),
    'APIMODEL_PATH': os.path.join('Tensorflow','models'),
    'ANNOTATION_PATH': os.path.join('Tensorflow', 'workspace','annotations'),
    'IMAGE_PATH': os.path.join('Tensorflow', 'workspace','images'),
    'MODEL_PATH': os.path.join('Tensorflow', 'workspace','models'),
    'PRETRAINED_MODEL_PATH': os.path.join('Tensorflow', 'workspace','pre-trained-models'),
    'CHECKPOINT_PATH': os.path.join('Tensorflow', 'workspace','models',CUSTOM_MODEL_NAME),
    'OUTPUT_PATH': os.path.join('Tensorflow', 'workspace','models',CUSTOM_MODEL_NAME, 'export'),
    'TFJS_PATH':os.path.join('Tensorflow', 'workspace','models',CUSTOM_MODEL_NAME, 'tfjsexport'),
    'TFLITE_PATH':os.path.join('Tensorflow', 'workspace','models',CUSTOM_MODEL_NAME, 'tfliteexport'),
    'PROTOC_PATH':os.path.join('Tensorflow','protoc')
 }
```

```
48
49  files = {
50      'PIPELINE_CONFIG':os.path.join('Tensorflow', 'workspace','models', CUSTOM_MODEL_NAME, 'pipeline.config'),
51      'TF_RECORD_SCRIPT': os.path.join(paths['SCRIPTS_PATH'], TF_RECORD_SCRIPT_NAME),
52      'LABELMAP': os.path.join(paths['ANNOTATION_PATH'], LABEL_MAP_NAME)
53  }
54
55  configs = config_util.get_configs_from_pipeline_file(files['PIPELINE_CONFIG'])
56  detection_model = model_builder.build(model_config=configs['model'], is_training=False)
57
58  ckpt = tf.compat.v2.train.Checkpoint(model=detection_model)
59  ckpt.restore(os.path.join(paths['CHECKPOINT_PATH'], 'ckpt-41')).expect_partial()  #The longer a model trains the more
    chekpoints it returns, important to chose the last checkpoint bechause its traind the most.
60
61  def detect_fn(image):
62      image, shapes = detection_model.preprocess(image)
63      prediction_dict = detection_model.predict(image, shapes)
64      detections = detection_model.postprocess(prediction_dict, shapes)
65      return detections
66
67  category_index = label_map_util.create_category_index_from_labelmap(files['LABELMAP'])
68
69  cap = cv2.VideoCapture(kameraValg)                    #Velger kamera som skal ta opp video
70
71  width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
72  height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
73
74  display_str_list = []
75  ##########################################################################################################
76  #Reading values from arduino serial port. line(77). FOUND HERE: https://www.youtube.com/watch?v=VN3HJm3spRE
77  ArduinoSerialData= serial.Serial('com3',9600)
78  time.sleep(1)
79  ##########################################################################################################
80  #Read last value from arduino serial port, not working correctly yet, line(81-98). FOUND HERE:
    https://stackoverflow.com/questions/1093598/pyserial-how-to-read-the-last-line-sent-from-a-serial-device
81  def serialReader():
82      ArduinoSerialData.flushInput()
83      #time.sleep(1)
84      while(ArduinoSerialData.inWaiting()==0):
85          pass
86       #time.sleep(2)
87      distanse=ArduinoSerialData.readline()
88      distanse = str(distanse,'utf-8')
89      distanse = distanse.strip('\r\n')
90      distanseInt = 0
91      try:
92          if(distanse != ''):
93              distanseInt = int(float(distanse))
94          #print(distanseInt)
95      except:
96          a=0
```

```
97          return(distanseInt)
98      ##########################################################################################################
99      #Object detection, line(100-132), FOUND HERE:
        https://github.com/nicknochnack/TFODCourse/blob/main/2.%20Training%20and%20Detection.ipynb
100     while cap.isOpened():
101         ret, frame = cap.read()
102         image_np = np.array(frame)
103         input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0), dtype=tf.float32)
104         detections = detect_fn(input_tensor)
105
106         num_detections = int(detections.pop('num_detections'))
107         detections = {key: value[0, :num_detections].numpy()
108                       for key, value in detections.items()}
109         detections['num_detections'] = num_detections
110
111         # detection_classes should be ints.
112         detections['detection_classes'] = detections['detection_classes'].astype(np.int64)
113
114         label_id_offset = 1
115         det_boxes = detections['detection_boxes']
116         det_classes = detections['detection_classes']+label_id_offset
117         det_scores = detections['detection_scores']
118
119
120         image_np_with_detections = image_np.copy()
121
122
123         viz_utils.visualize_boxes_and_labels_on_image_array(
124                     image_np_with_detections,
125                     det_boxes,
126                     det_classes,
127                     det_scores,
128                     category_index,
129                     use_normalized_coordinates=True,
130                     max_boxes_to_draw=1,  #Only the boundingbox with the highest estimated accuracy wil be showed, since
                        the estimated accuracy tends to drop at longer distances the weld closest to the robot is most likley
                        to be detected.
131                     min_score_thresh=.5,  #The estimated accuracy needs to be at lest 50% to draw a bounding box, this is
                        to secure that it is a weld that is detected
132                     agnostic_mode=False)
133
134
135
136         #cv2.imshow('object detection',  cv2.resize(image_np_with_detections, (800, 650)))
137
138         ##########################################################################################################
139         # Find boundingbox cordinates, line(139-164), FOUND HERE: https://github.com/tensorflow/models/issues/4682
140         boxes = detections['detection_boxes']
141         height, width = image_np_with_detections.shape[:2]
142         box= np.squeeze(boxes)
143         max_boxes_to_draw=box.shape[0]
```

```
144         scores=np.squeeze(det_scores)
145         min_score_thresh=0.5
146         #time.sleep(5)
147 
148         xmax = 0
149         xmin = 0
150         ymin = 0
151         ymax = 0
152 
153 
154         if scores [0] > min_score_thresh:  #If multible boxes are found it is bosible to only take the ones with highest
            estimated accuracy, in our case its not nesesary since we only detect one box at the time.
155              ymin = (int(box[0,0]*height))
156              xmin = (int(box[0,1]*width))
157              ymax = (int(box[0,2]*height))
158              xmax = (int(box[0,3]*width))
159              print (xmin,ymin,xmax,ymax)
160 
161         nPixelPipeDimension = xmax-xmin
162         nPixelWeldWidth = ymax-ymin
163         if(xmax == 0 and xmin == 0 and ymin == 0 and ymax == 0): #If nothing is detectecd, the program notifies it
164             print('Boundingbox not detected')
165         ##############################################################################################################
166         if(nPixelPipeDimension<nPixelWeldWidth):
167             nD = nPixelPipeDimension
168             nPixelPipeDimension = nPixelWeldWidth
169             nPixelWeldWidth = nD
170 
171 
172         DistanceToPipe = serialReader()
173         LengthPerPixelD = (24/16789)*(DistanceToPipe-40)+(12/163)
174         print(DistanceToPipe,' cm')
175         print('D: ', nPixelPipeDimension)
176         print('W: ',nPixelWeldWidth)
177 
178         RørDimensjon = nPixelPipeDimension*LengthPerPixelD
179         WeldWidth = nPixelWeldWidth*LengthPerPixelD
180 
181         RoundedPipeDimension = round(RørDimensjon/2.86) #From cm to closest whole inche
182         WeldWidth = round(WeldWidth,2)
183         WeldWidth = WeldWidth - 1
184         print(WeldWidth,'cm : Weld width')
185         print(RoundedPipeDimension,'inches : Pipe dimension')
186         print(' ')
187 
188         TextDistanceToPipe = 'Distance to pipe: '+str(DistanceToPipe)+' cm'
189         TextWeldWidth = 'Weld width: '+str(round(WeldWidth,2))+' cm'
190         TextPipeDimension = 'Pipe dimension: '+str(round(RoundedPipeDimension,2))+' inches'
191 
192         ##############################################################################################################
193         #Text display, line(194-197) https://github.com/techwithtim/OpenCV-Tutorials/blob/main/tutorial4.py
```

```
        font = cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(image_np_with_detections,TextDistanceToPipe, (25,30), font, 0.7, (0, 0, 240), 1, cv2.LINE_AA)
        cv2.putText(image_np_with_detections,TextWeldWidth, (25,60), font, 0.7, (0, 0, 240), 1, cv2.LINE_AA)
        cv2.putText(image_np_with_detections,TextPipeDimension, (25,90), font, 0.7, (0, 0, 240), 1, cv2.LINE_AA)
        ##########################################################################################################
        image_np_with_detections = cv2.circle(image_np_with_detections,(xmin,ymin),5,(0,0,200),-1)
        image_np_with_detections = cv2.circle(image_np_with_detections,(xmax,ymax),5,(0,0,200),-1)
        #cv2.imshow('object detection',  cv2.resize(image_np_with_detections, (800, 650)))
        cv2.imshow("Capture",frame)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            cap.release()
            cv2.destroyAllWindows()
            break
```