Exam 3

EEL 6990

University of West Florida

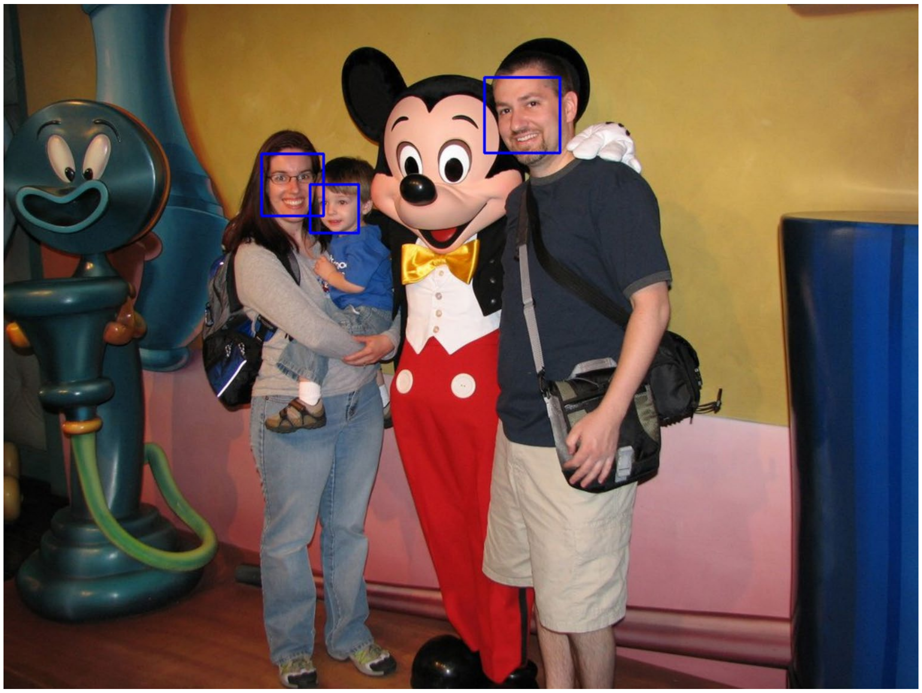
Problem Number		
1	30	
2	30	
3	40	
Total	100	

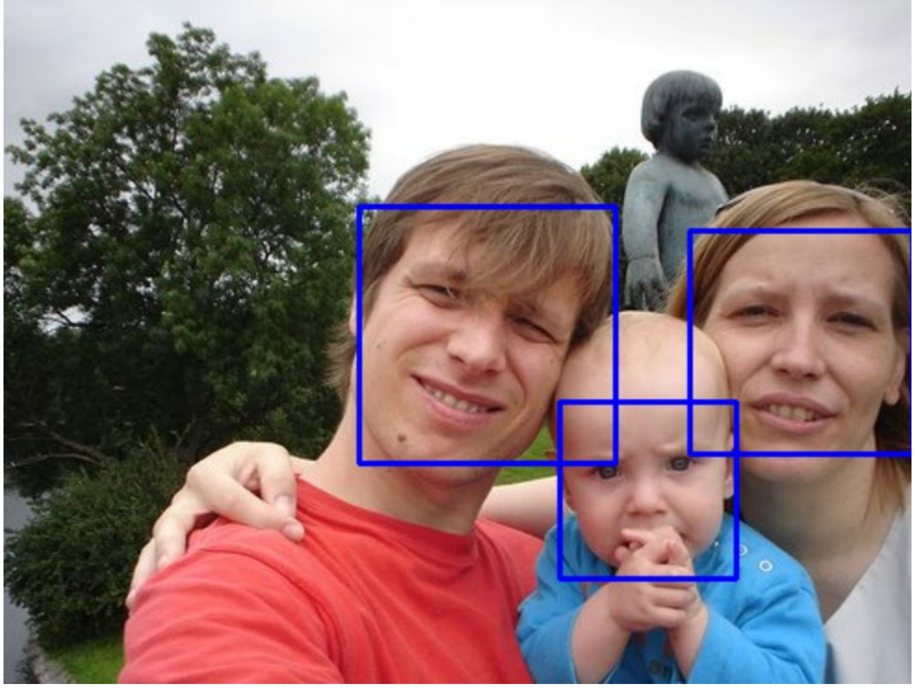
Tegan Shiver
EEL6990 – Exam 3
12/3/2021

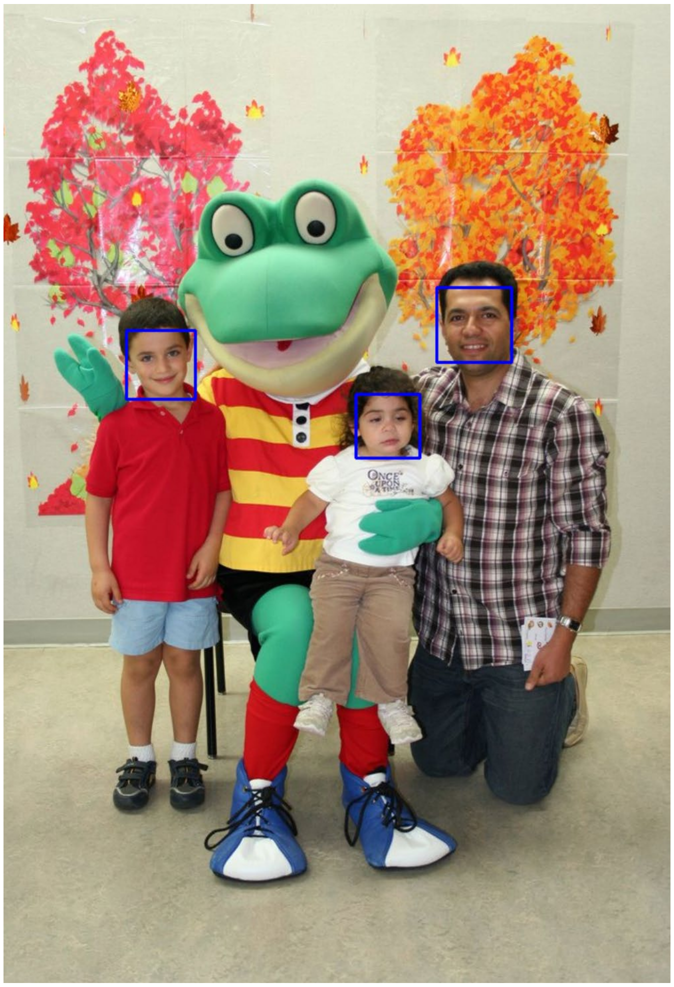
Problem 1) (30 Points)

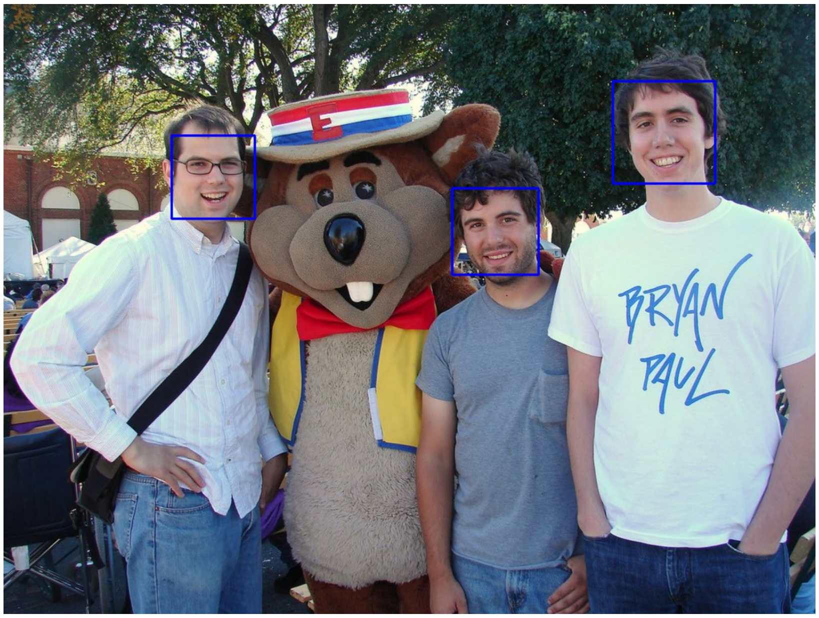
Using images titled "prob1-1.jpg" to "prob1-4.jpg" (4 images in total), implement a face detector algorithm of your choice. You can find code from online resources. Add results in your answer that include detected faces with bounding boxes. Tune the algorithm in such a way that it needs to put a bounding box for only human faces.

Used Haar-Cascade Classifier for Face Detection








E
BRYAN
PAUL

Problem 2) (30 Points)

For the video titled “prob2.mp4”, implement any detection-tracking algorithm of your choice for detecting-tracking fixed-wing aircraft. You can find code from online resources. The resulting video should have a bounding box around the aircraft. Upload your resulting video to YouTube channel associated with your UWF student email; upload it as “Unlisted” and put the link of video in your answer. Add detection rate calculation in your answer. How to: create a spreadsheet, add row for each frame of the video. First column will be if the aircraft visible in that frame, manually check frames and fill out that column. Second column will be if the algorithm detects aircraft in that frame with a bounding box. Detection rate will be equal to the ratio of number of frames that algorithm detects the aircraft over number of frames that aircraft is visible.

Used MIL Tracker – Output Video Link: https://youtu.be/Ir-uZPa4KN8

Detection Rate = 74%

- detected in frames 0-209
- not detected in frames 210-381
- present in frames 0-209 and 308-381

→ 210 / (210+74) * 100% = 74%

Problem 3) (40 Points)

For the video file titled “prob3.mp4”, implement any recognition algorithm of your choice. You can find code from online resources. The resulting video should have bounding boxes around three moving objects, 1) humanoid robot 2) ground vehicle 3) quadrotor, and there should be different labels for each object (for instance, label “robot” for humanoid robot, label “rover” for ground vehicle, label “drone” for quadrotor). Upload your resulting video to YouTube channel associated with your UWF student email; upload it as “Unlisted” and put the link of video in your answer.

Output Video Link using YOLOv5s model: https://youtu.be/RpbXQQfNNds

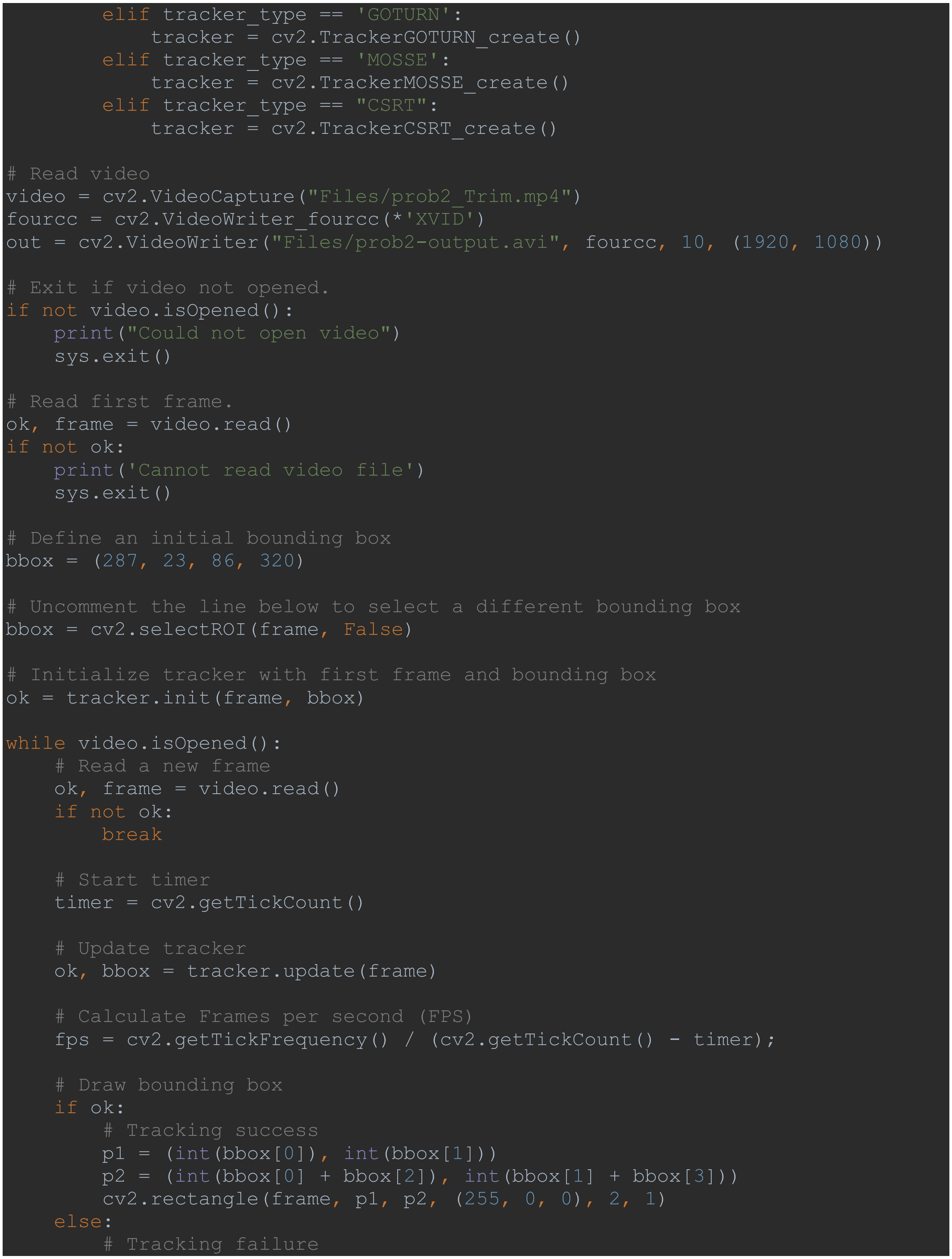
Code

Problem 1)

```
import cv2

# Load the cascade
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt.xml')
# Read the input image
img = cv2.imread('Files/prob1-3.jpg')
# Convert into grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Detect faces
faces = face_cascade.detectMultiScale(gray, 1.1, 4)
# Draw rectangle around the faces
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
# Display the output
cv2.imshow('img', img)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Problem 2)

```
import cv2
import sys

(major_ver, minor_ver, subminor_ver) = (cv2.__version__).split('.')

if __name__ == '__main__':

    # Set up tracker.
    # Instead of CSRT, you can also use

    tracker_types = ['BOOSTING', 'MIL', 'KCF', 'TLD', 'MEDIANFLOW', 'GOTURN',
'MOSSE', 'CSRT']
    tracker_type = tracker_types[1]

    if int(minor_ver) < 3:
        tracker = cv2.Tracker_create(tracker_type)
    else:
        if tracker_type == 'BOOSTING':
            tracker = cv2.TrackerBoosting_create()
        elif tracker_type == 'MIL':
            tracker = cv2.TrackerMIL_create()
        elif tracker_type == 'KCF':
            tracker = cv2.TrackerKCF_create()
        elif tracker_type == 'TLD':
            tracker = cv2.TrackerTLD_create()
        elif tracker_type == 'MEDIANFLOW':
            tracker = cv2.TrackerMedianFlow_create()
```



```
        elif tracker_type == 'GOTURN':
            tracker = cv2.TrackerGOTURN_create()
        elif tracker_type == 'MOSSE':
            tracker = cv2.TrackerMOSSE_create()
        elif tracker_type == "CSRT":
            tracker = cv2.TrackerCSRT_create()

# Read video
video = cv2.VideoCapture("Files/prob2_Trim.mp4")
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter("Files/prob2-output.avi", fourcc, 10, (1920, 1080))

# Exit if video not opened.
if not video.isOpened():
    print("Could not open video")
    sys.exit()

# Read first frame.
ok, frame = video.read()
if not ok:
    print('Cannot read video file')
    sys.exit()

# Define an initial bounding box
bbox = (287, 23, 86, 320)

# Uncomment the line below to select a different bounding box
bbox = cv2.selectROI(frame, False)

# Initialize tracker with first frame and bounding box
ok = tracker.init(frame, bbox)

while video.isOpened():
    # Read a new frame
    ok, frame = video.read()
    if not ok:
        break

    # Start timer
    timer = cv2.getTickCount()

    # Update tracker
    ok, bbox = tracker.update(frame)

    # Calculate Frames per second (FPS)
    fps = cv2.getTickFrequency() / (cv2.getTickCount() - timer);

    # Draw bounding box
    if ok:
        # Tracking success
        p1 = (int(bbox[0]), int(bbox[1]))
        p2 = (int(bbox[0] + bbox[2]), int(bbox[1] + bbox[3]))
        cv2.rectangle(frame, p1, p2, (255, 0, 0), 2, 1)
    else:
        # Tracking failure
        cv2.putText(frame, "Tracking failure detected", (100, 80),
cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0, 255), 2)

    # Display tracker type on frame
    cv2.putText(frame, tracker_type + " Tracker", (100, 20),
cv2.FONT_HERSHEY_SIMPLEX, 0.75, (50, 170, 50), 2);

    # Display FPS on frame
    cv2.putText(frame, "FPS : " + str(int(fps)), (100, 50),
cv2.FONT_HERSHEY_SIMPLEX, 0.75, (50, 170, 50), 2);
    out.write(frame)
    # Display result
    cv2.imshow("Tracking", frame)

    # Exit if ESC pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):  # if press SPACE bar
        break

video.release()
out.release()
cv2.destroyAllWindows()
```

Parse Output Video into Frames to Calculate Detection Rate

```
import cv2

cap = cv2.VideoCapture("Files/prob2-output.avi")
i = 0
while cap.isOpened():
    ret, frame = cap.read()
    if ret == False:
        break
    cv2.imwrite('Files/Prob2/Frame' + str(i) + '.jpg', frame)
    i += 1

cap.release()
cv2.destroyAllWindows()
```

Problem 3)

```
# install CUDA 11.1 & cuDNN v8.1.0
# https://developer.nvidia.com/cuda-11.1.0-download-archive
# https://developer.nvidia.com/rdp/cudnn-archive

# install PyTorch in Anaconda Prompt
# pip install torch==1.8.2+cu111 torchvision==0.9.2+cu111 torchaudio===0.8.2
-f https://download.pytorch.org/whl/lts/1.8/torch_lts.html

# Clone YOLOv5 from Github
# git clone https://github.com/ultralytics/yolov5

# install YOLOv5 dependencies
# cd yolov5 & pip install -r requirements.txt
# Clone labelImg from Github
# git clone https://github.com/tzutalin/labelImg

# Open labelImg in Anaconda Prompt

# Label Images from parsed video - make sure to save annotations in YOLO
format

# Create dataset.yaml file using classes file generated from labelImg

# Train custom model using Anaconda Prompt using yolov5 weights
# python train.py --img 320 --batch 16 --epochs 250 --data dataset.yaml --
weights yolov5s.pt --workers 1

import torch
import numpy as np
import cv2

# use custom model
model = torch.hub.load('ultralytics/yolov5', 'custom',
path='C:/Users/TVS/anaconda3/CV/yolov5/runs/train/exp2'
                                      '/weights/best.pt', force_reload=True)
cap = cv2.VideoCapture('Files/prob3.mp4')

while cap.isOpened():
    ret, frame = cap.read()

    # Make detections
    results = model(frame)
    cv2.imshow('YOLO', np.squeeze(results.render()))

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

cap.release()
cv2.destroyAllWindows()
```

References

Problem 1)

- Face Detection
  https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python-90f89d7c0f81

Problem 2)

- Object Tracking with OpenCV
  https://livecodestream.dev/post/object-tracking-with-opencv/

Problem 3)

- YOLOv5 Tutorial
  https://www.youtube.com/watch?v=tFNJGim3FXw

- Clone YOLOv5 from Github
  https://github.com/ultralytics/yolov5

- Clone labelImg from Github
  https://github.com/tzutalin/labelImg