Tegan Shiver

EEL6990 – Project 1

October 5, 2021

<u>Vehicle Detection from Aerial Footage Using a Haar Cascade Classifier</u>

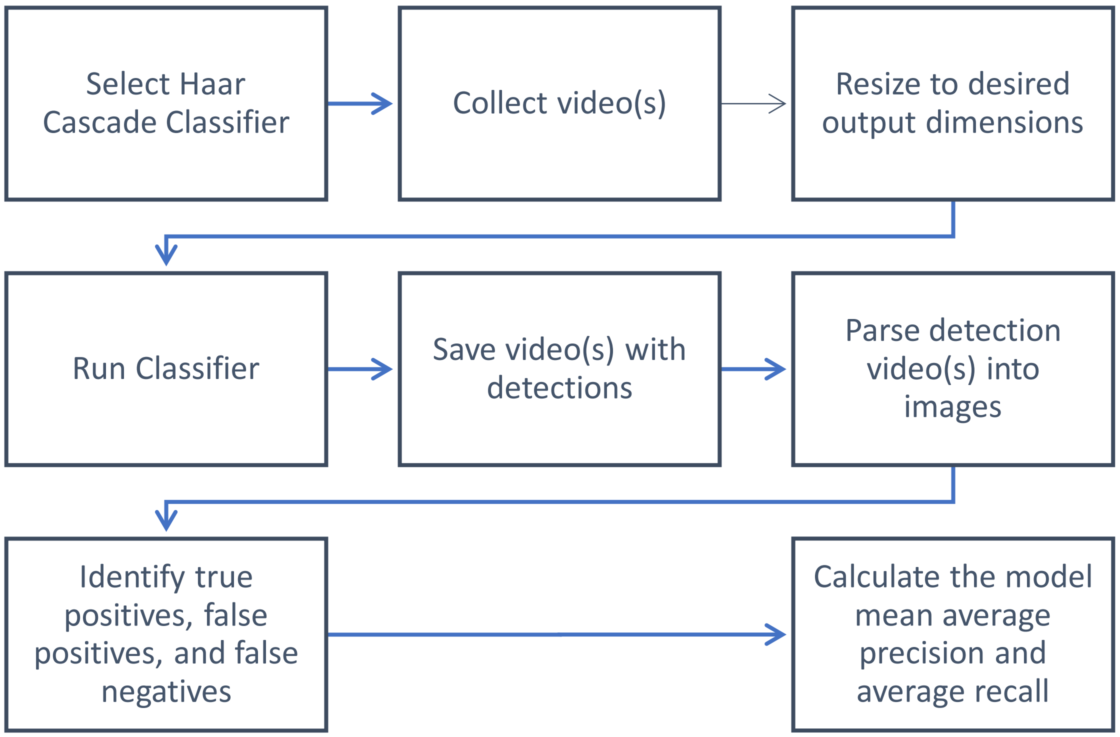
Description

A Haar Cascade Classifier is an effective object detection approach that was proposed by Paul Viola and Michael Jones in 2001 in their paper, *"Rapid Object Detection using a Boosted Cascade of Simple Features"*. These classifiers use a machine learning approach where a cascade function is trained from a collection of positive and negative images. Positive images are paired with an annotation file (.xml) that contains information about the object's location.

For this project, drone footage of UWF's campus was utilized to test a pre-trained Haar Cascade Classifier that was built for vehicle detection.

Flow Chart

Process using a pre-trained classifier:


Select Haar Cascade Classifier
Collect video(s)
Resize to desired output dimensions
Run Classifier
Save video(s) with detections
Parse detection video(s) into images
Identify true positives, false positives, and false negatives
Calculate the model mean average precision and average recall

Tegan Shiver

EEL6990 – Project 1

October 5, 2021

Select Haar Cascade Classifier

For this project, the Haar Cascade Classifier ‘cars.xml’ was chosen for vehicle detection. This classifier was pre-trained using 526 images of vehicles’ front and rear-ends on a freeway.

Collect Video(s)

Videos used to test this classifier were repurposed from another project at UWF where drone footage was taken over various areas of the campus after Hurricane Sally to detect damage. A video with a significant number of vehicles was selected to test the classifier.

Resize to Desired Output Dimensions

In order to run the classifier, it was necessary to resize the testing video to 1920x1080 pixels (dimensions were too large to view on screen). The dimensions also needed to match the desired output (1080p). (See code)

Run Classifier/Save Video(s) with Detections

Run the chosen classifier on the resized video. Save the video with the detections (bounding boxes). (See code)

Parse Detection Video(s) into Images

To count the true positives, false positives, and false negatives, the video was parsed into images by frame. (See code)

Identify True Positives, False Positives, and False Negatives

True positives (TP) are correct detections, false positives (FP) do not contain the object (incorrect detections), and false negatives (FN) occur when the object is present, but no detections are made at all.

Calculate the Model’s Mean Average Precision and Average Recall

The mean average precision is the proportion of detections that were correct,

$$\frac{TP}{TP + FP}$$

The average recall is the proportion of the actual objects that were captured,

$$\frac{TP}{TP + FN}$$

Tegan Shiver

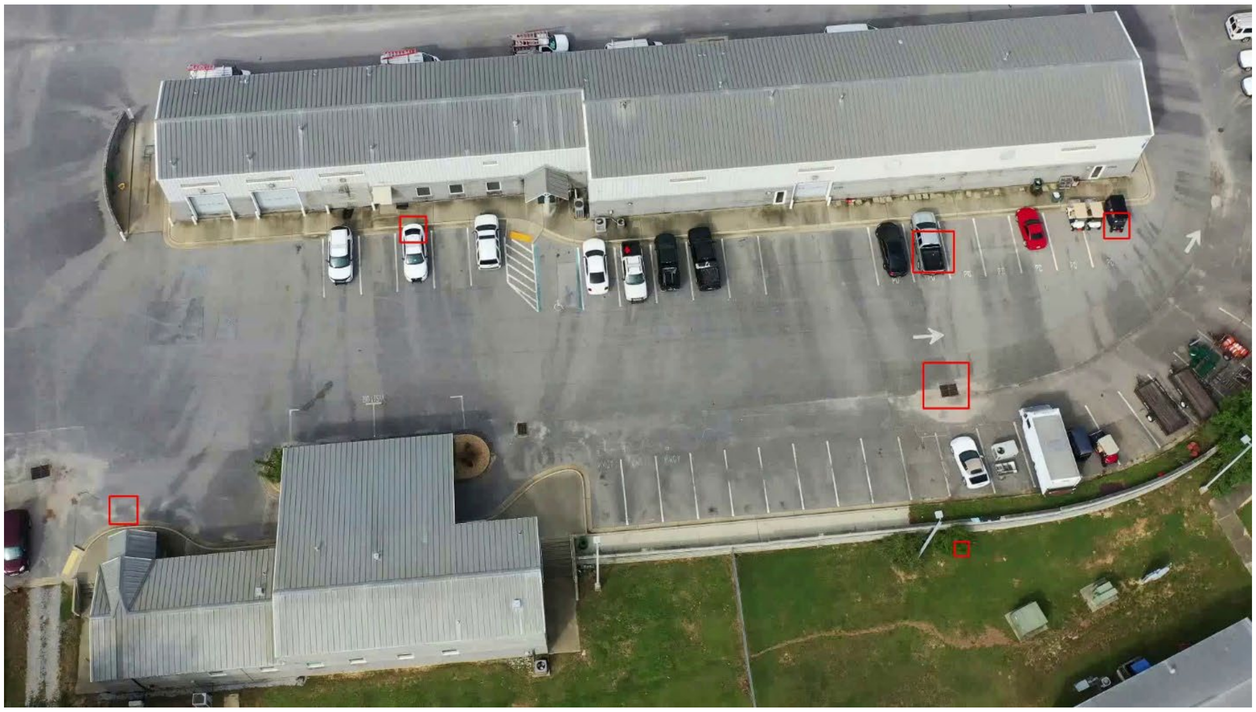
EEL6990 – Project 1

October 5, 2021

Results and Analysis

Below are sample images of the output detection video frames: 0, 165, and 327. The bounding boxes only changed every three frames. Therefore, every third image was reviewed to record true positives, false positives, and false negatives (110 images).

The mean average precision (mAP) was calculated to be 0.62 (62%), and the average recall was 0.26 (26%). This means that 62% of the detections made were of vehicles, but only 26% of the vehicles were detected in total. This performance was expected using a classifier that was trained on front and rear-ends of vehicles, as the drone footage is from an aerial view (top).

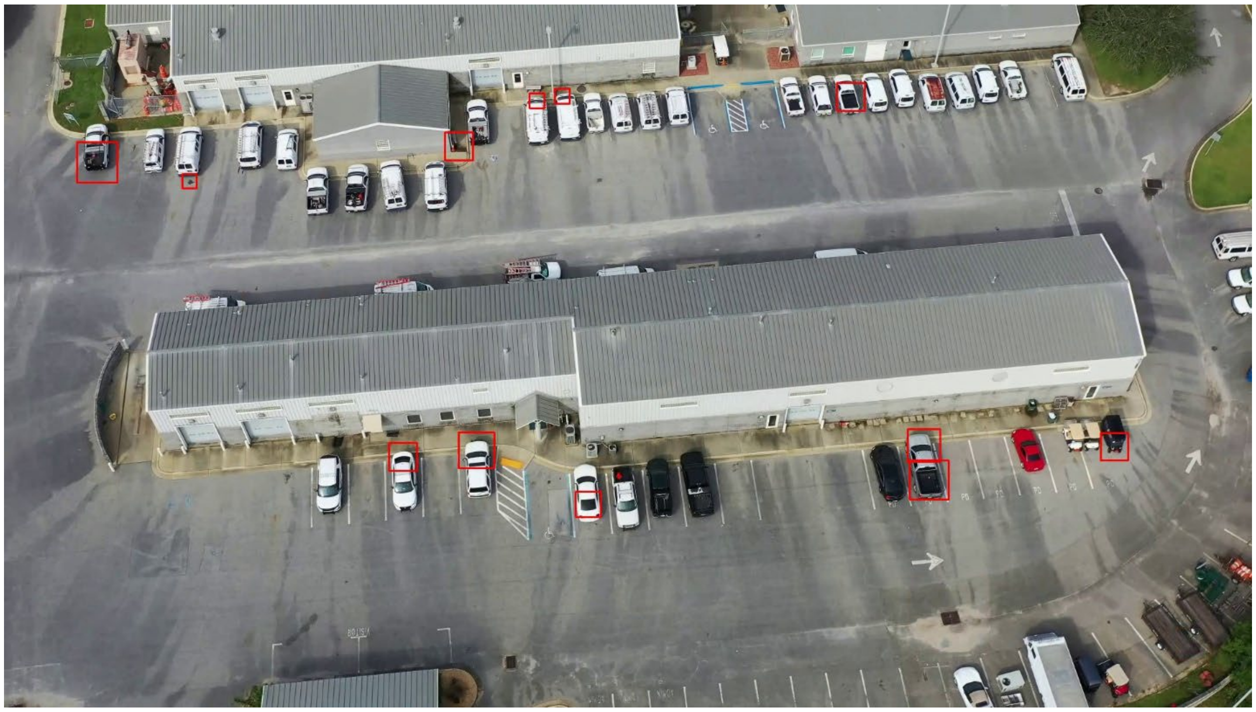


i. Frame0.jpg

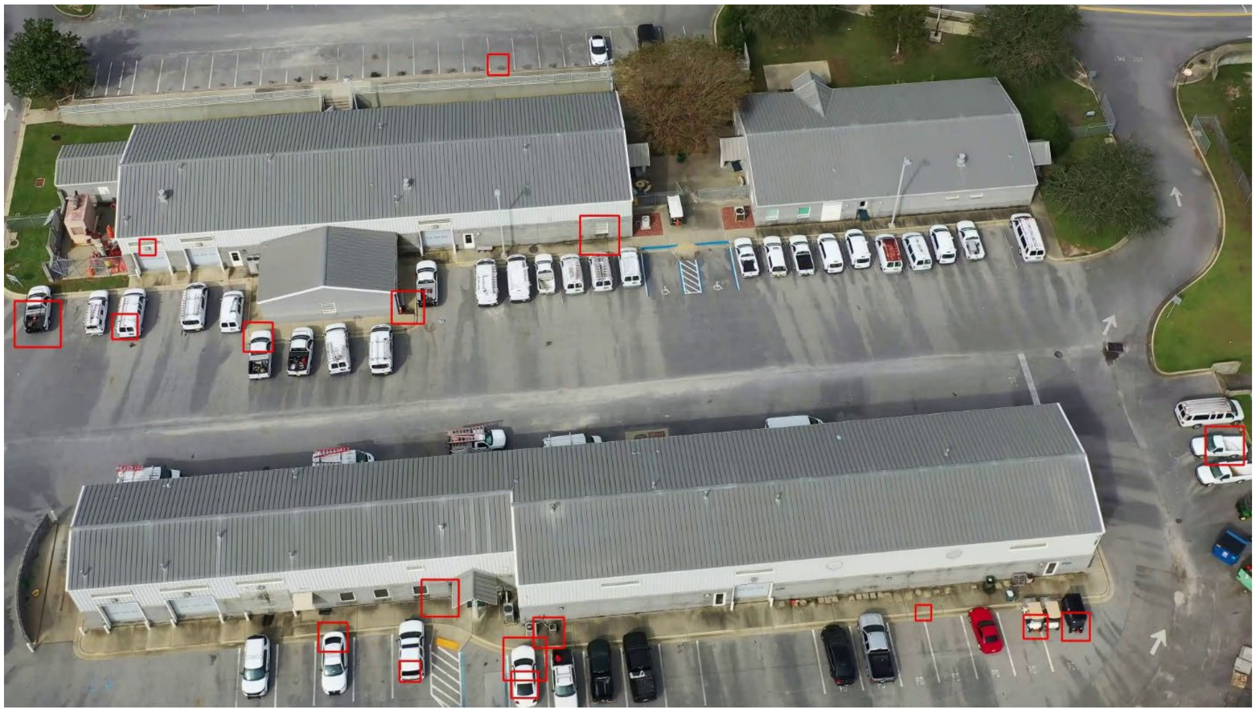
Tegan Shiver

EEL6990 – Project 1

October 5, 2021



ii. Frame165.jpg



iii. Frame327.jpg

Tegan Shiver

EEL6990 – Project 1

October 5, 2021

Code

Python/PyCharm used

Resize Original Video to Desired Output Dimensions

```
import cv2

# Code to resize video to match output dimensions
# Upload original video
cap = cv2.VideoCapture("Drone Footage/Parking Services_Trim.MP4")

# Resize video and save
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter("Drone Footage/Resized/Parking Services_Trim
1080p.avi", fourcc, 50, (1920, 1080))

while True:
    ret, frame = cap.read()
    if ret == True:
        b = cv2.resize(frame, (1920, 1080), fx=0, fy=0,
interpolation=cv2.INTER_CUBIC)
        out.write(b)
    else:
        break

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

Run Classifier/Save Video(s) with Detections

```
import cv2

# Code to save detection video with bounding boxes
# Upload resized video
cap = cv2.VideoCapture("Drone Footage/Resized/Parking Services_Trim
1080p.avi")

fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter("Drone Footage/Detection/Parking
Services_Trim_Detection 1080p.avi", fourcc, 10, (1920, 1080))

# Haar Cascade Classifier being used
carCascade = cv2.CascadeClassifier("haarcascade_car.xml")

while cap.isOpened():
    ret, frame = cap.read()
    if ret == True:
        if (type(frame)) == type(None):
            break
        imgGray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        cars = carCascade.detectMultiScale(imgGray, 1.1, 1)
        for (x, y, w, h) in cars:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
        out.write(frame)
        cv2.imshow('Detection Video', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break

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

Parse Detection Video(s) into Images

```
import cv2
cap = cv2.VideoCapture("Drone Footage/Detection/Parking
Services_Trim_Detection 1080p_Trim.mp4")
i = 0
while (cap.isOpened()):
    ret, frame = cap.read()
    if ret == False:
        break
    cv2.imwrite('Drone Footage/Detection/Video to Img/Frame' + str(i) +
'.jpg', frame)
    i += 1

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

Tegan Shiver

EEL6990 – Project 1

October 5, 2021

References

- Lowande, R., Clevenger, A., Mahyari, A., Sevil, H.E., "Analysis of Post-Disaster Damage Detection using Aerial Footage from UWF Campus after Hurricane Sally". *International Conference on Image Processing, Computer Vision, & Pattern Recognition (IPCV'21)*, Las Vegas, USA, 26-29 July 2021.

- Murtaza's Workshop - Robotics and AI, "Learn OpenCV in 3 Hours with Python | Including 3xProjects | Computer Vision." https://youtu.be/WQeoO7MI0Bs

- Nicholas Renotte, "Tensorflow Object Detection in 5 Hours with Python | Full Course with 3 Projects." https://youtu.be/yqkISICHH-U

- Khan, "Computer Vision - Detecting objects using Haar Cascade Classifier." https://towardsdatascience.com/computer-vision-detecting-objects-using-haar-cascade-classifier-4585472829a9

- Andrewssobral, "Vehicle Detection by Haar Cascades with OpenCV." https://github.com/andrewssobral/vehicle_detection_haarcascades