Tegan Shiver

EEL6990 – Project 2

November 9, 2021

Vehicle Detection from Aerial Footage Using PyTorch and YOLOv5

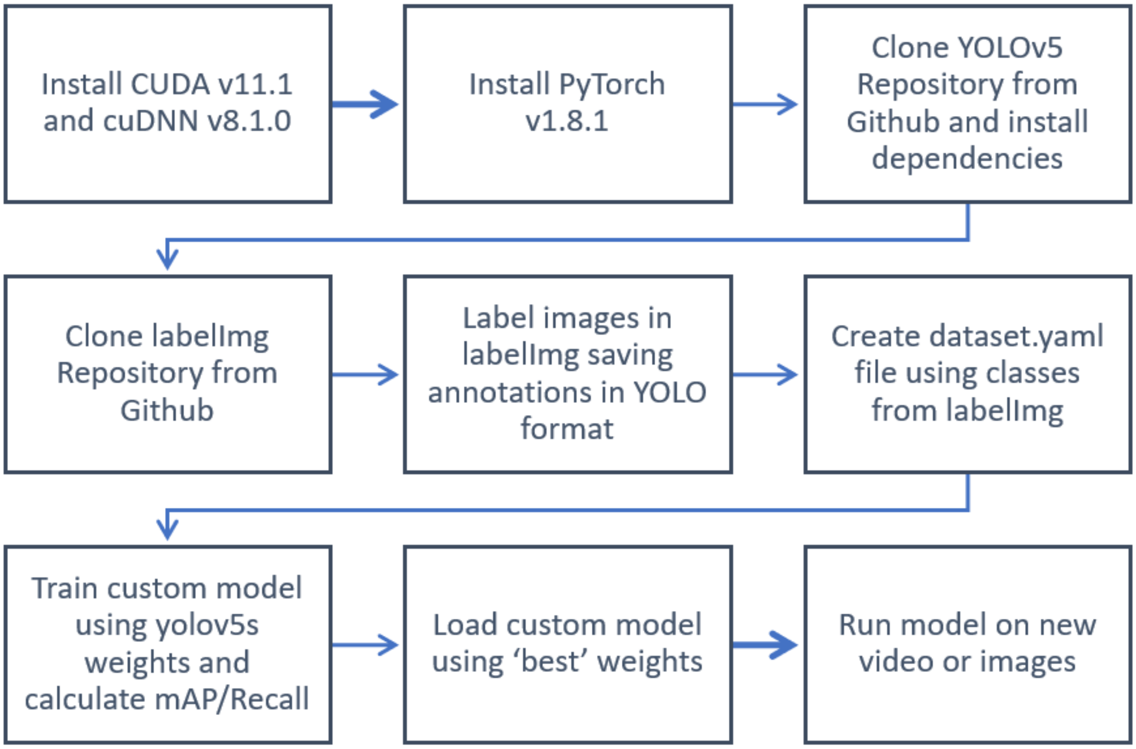
Description

YOLO (*You Only Look Once*) is an algorithm that employs convolutional neural networks (CNN) to detect objects in real-time. This algorithm is popular because of its speed and accuracy. YOLOv5 is the latest version of YOLO that utilizes the PyTorch framework and is pretrained on the COCO dataset. There are four different models to choose from for custom training – yolov5s (small), yolov5m (medium), yolov5l (large), and yolov5x (extra-large).

For this project, the yolov5s model was trained using image snips from drone footage of UWF's campus after hurricane Sally. These images were hand labeled in labelImg using the labels – 'car', 'truck', and 'suv'.

Flow Chart

Process using YOLOv5:


Install CUDA v11.1 and cuDNN v8.1.0
Install PyTorch v1.8.1
Clone YOLOv5 Repository from Github and install dependencies
Clone labelImg Repository from Github
Label images in labelImg saving annotations in YOLO format
Create dataset.yaml file using classes from labelImg
Train custom model using yolov5s weights and calculate mAP/Recall
Load custom model using 'best' weights
Run model on new video or images

Tegan Shiver

EEL6990 – Project 2

November 9, 2021

Install CUDA v11.1 and cuDNN v8.1.0

Install versions of CUDA and cuDNN that are compatible with PyTorch.

Install PyTorch v1.8.1

YOLOv5 requires PyTorch >= 1.7

Clone YOLOv5 Repository from Github and install dependencies

Clone the YOLOv5 repository from 'https://github.com/ultralytics/yolov5' and install all dependencies using 'cd yolov5 & pip install -r requirements.txt'

Clone labelImg Repository from Github

Clone labelImg repository from 'https://github.com/tzutalin/labelImg'.

Label images in labelImg saving annotations in YOLO format

Open labelImg and point the directory to image folder. Change annotation format to YOLO before saving labels, otherwise YOLOv5 will not work. 15 images were annotated for this project.

Create dataset.yaml file using classes from labelImg

LabelImg populates a 'classes.txt' file based on selected YOLO model (yolov5s). To train a custom model a YAML file must be created containing these class labels. Create a file names dataset.yaml in the same location YOLOv5 was cloned.

Train custom model using yolov5s weights and calculate mAP/ Recall

Train model using selected weights in the Anaconda Prompt – 'python train.py --img 320 --batch 16 --epochs 500 --data dataset.yaml --weights yolov5s.pt --workers 1'. Calculate mAP and Recall from generated confusion matrix.
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,

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

Load custom model using 'best' weights

model = torch.hub.load('ultralytics/yolov5', 'custom', path=' yolov5/runs/train/exp5/weights/best.pt', force_reload=True)

Run model on new video or images

Import media using OpenCV – run model (See code).

Tegan Shiver

EEL6990 – Project 2

November 9, 2021

Results and Analysis

Below is the Confusion Matrix generated by yolov5s (*Fig 1*), the Training Batch images (*Fig 2*), and Validation Batch images with Predictions (*Fig 3*) from the custom trained model described above.

The Correlation Matrix was utilized to calculate the mean average precision (mAP) and average recall – 65% and 96% respectively. This means that 65% of the detections made were correct, and 96% of the vehicles were detected in total.

The mAP here is very close to the pre-trained Haar Cascade Classifier used in Project 1 (62%), but YOLOv5 detected 70% more of the vehicles present (Cascade recall = 26%). This difference was not surprising since the YOLO model was trained using custom images/labels.

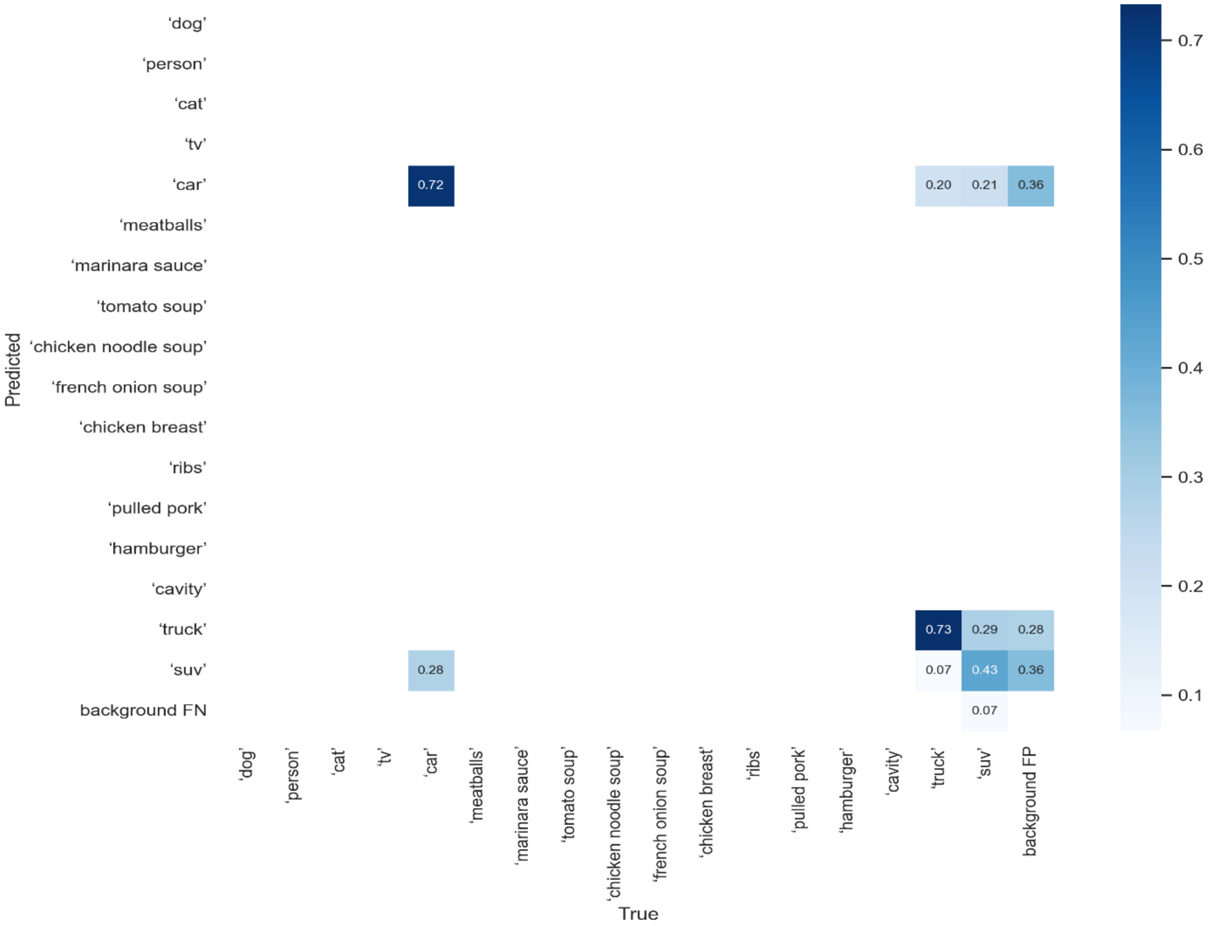


Fig 1. yolov5s Correlation Matrix

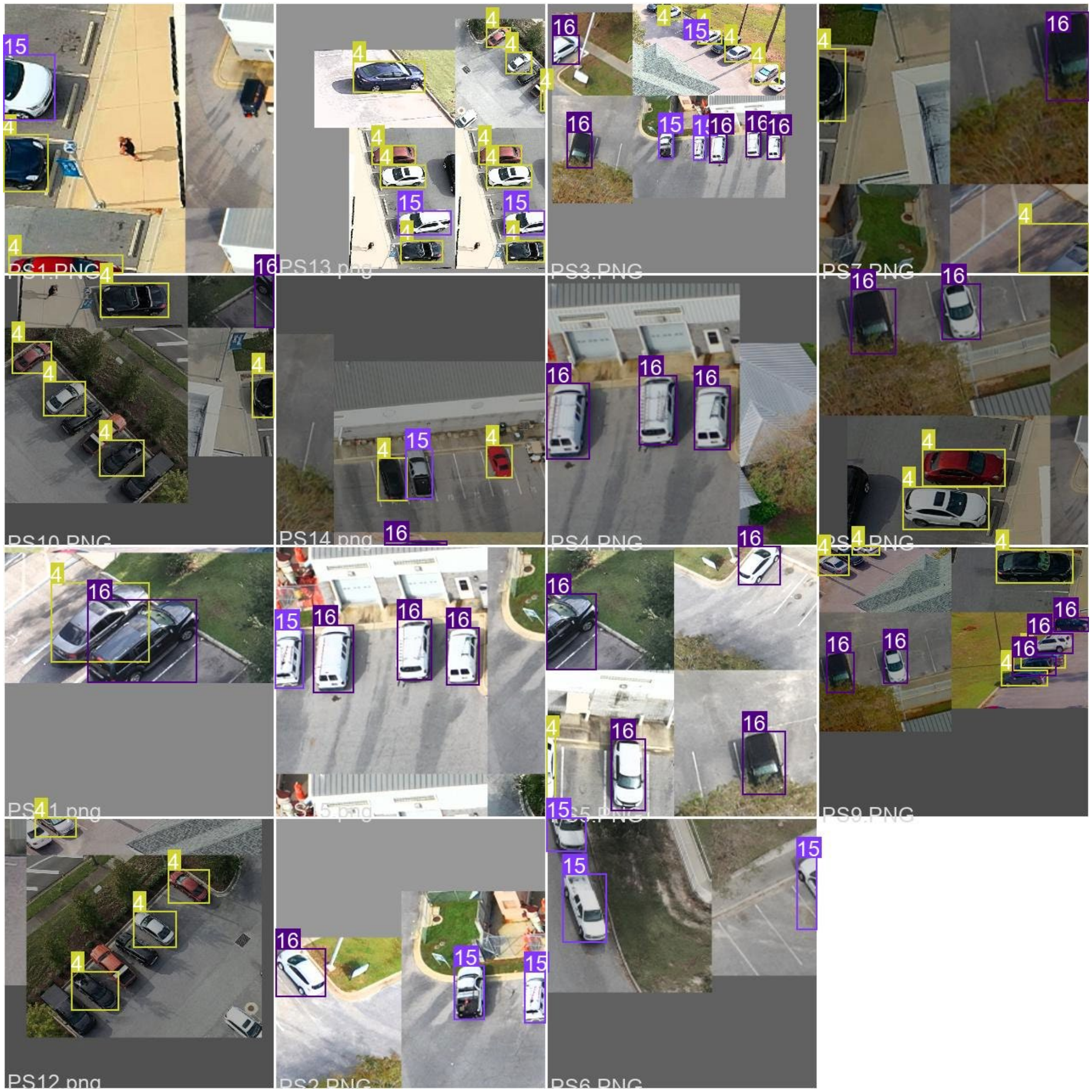


Fig 2. Training Batch

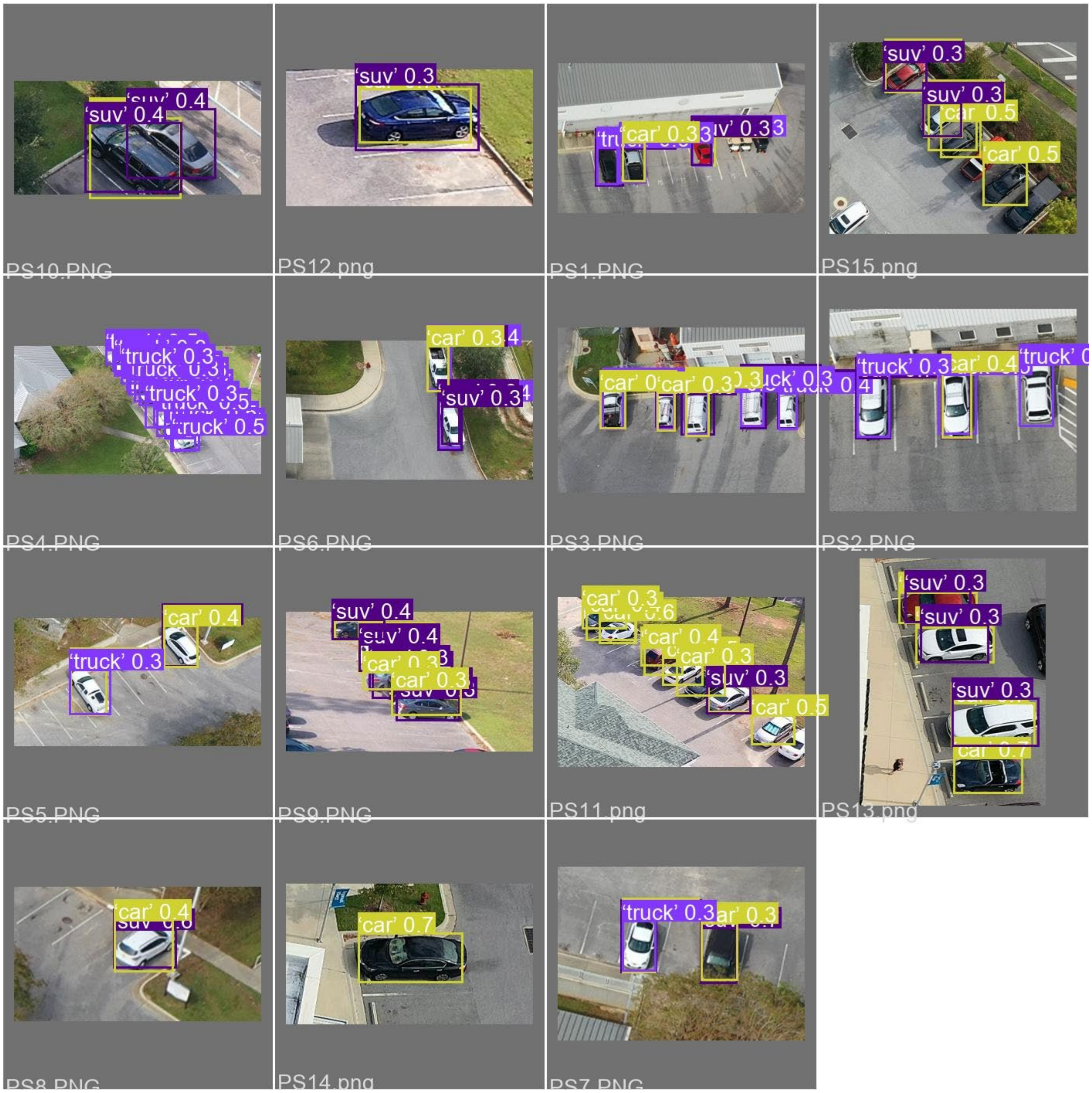


Fig 3. Validation Batch with Predictions

Tegan Shiver

EEL6990 – Project 2

November 9, 2021

Code

Python/PyCharm/Anaconda Prompt used

```
# 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==1.8.1+cu111 torchvision==0.9.1+cu111 torchaudio===0.8.1
-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 - make sure to save annotations in YOLO format

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

# Train custom model using Anaconda Prompt using yolov5s weights
# python train.py --img 320 --batch 16 --epochs 500 --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/tegan/anaconda3/YOLO/YOLO '
'Code/yolov5/runs/train/exp5/weights/best.pt',force_reload=True)

cap = cv2.VideoCapture('Videos/Argo Village 1080p.avi')
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()
```

Tegan Shiver

EEL6990 – Project 2

November 9, 2021

YAML Code – dataset.yaml

```yaml
path: ../data  # dataset root dir
train: images
val: images

# Classes
nc: 17  # number of classes
names: [ ‘dog’, ‘person’, ‘cat’, ‘tv’, ‘car’, ‘meatballs’, ‘marinara sauce’, ‘tomato
soup’, ‘chicken noodle soup’, ‘french onion soup’, ‘chicken breast’, ‘ribs’, ‘pulled
pork’, ‘hamburger’, ‘cavity’, ‘truck’, ‘suv’ ]  # class names
```

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.

- YOLO Algorithm Background
https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/

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

- Install CUDA v11.1
https://developer.nvidia.com/cuda-11.1.0-download-archive

- Install cuDNN v8.1.0
https://developer.nvidia.com/rdp/cudnn-archive

- Install PyTorch v1.8.1
https://pytorch.org/get-started/locally/

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

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