Object Detection in Aerial Drone Videos

Soumyadeep Banik

Instructor: Sujoy Kumar Biswas

Machine Learning Systems, 2021

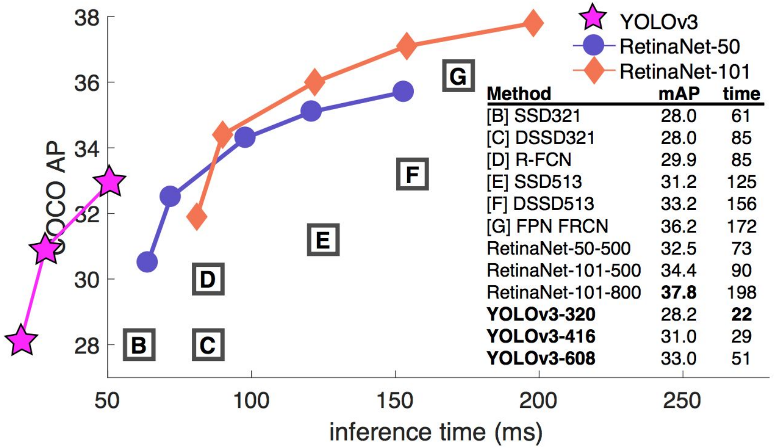
RKMVERI, Belur

Problem Statement and Related works on Object Detection:

To detect and recognize the visible objects in a particular instance of a video captured from drone/UAV.

Dataset used: **Stanford Drone Dataset** ([can be found here])

Other detectors:



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

Source: J. Redmon, A. Farhadi
YOLOv3: An Incremental Improvement

- *YOLOv3 at 320 × 320, 3 times faster than SSD*
- *compared to 57.5 AP_{50} in 198 ms by RetinaNet, similar performance but 3.8× faster*

Dataset Description:

- 60 videos(70 GB) from Stanford university campus captured from UAV/drone.
- Categorized into 8 unique scenes. Each video contains pedestrians, bicyclists, skateboarders, cars, buses, and golf carts.
- Ground Truth:

  2 million+ annotations in text format consisting :

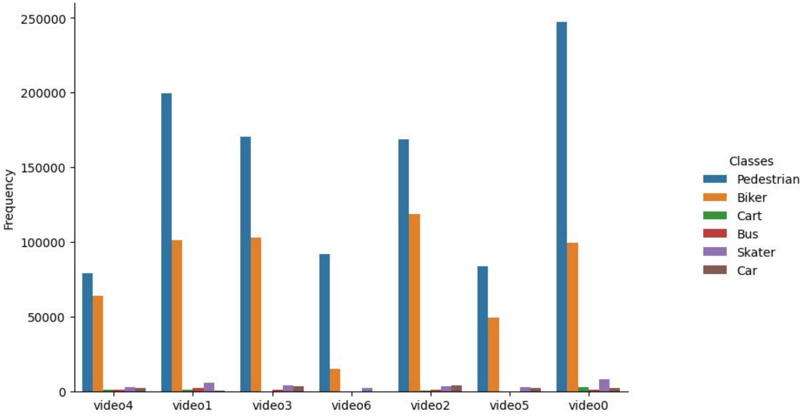
  [**track id, xmin, ymin, xmax, ymax, frame, lost, occluded, appeared, class**]

- Training data: 17 videos chosen from all 8 different locations.
- Converted into PASCAL-VOC format. 1 in every 30 frames and corresponding labels are selected and put into the new dataset.
  [**class_ID_1 X_CENTER_NORM Y_CENTER_NORM WIDTH_NORM HEIGHT_NORM**]

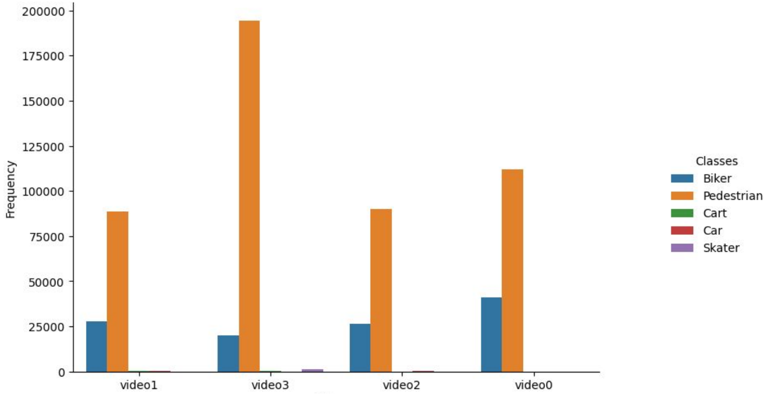
```
— dataset
  ├── videos
        └── categories {bookstore, quad, .. etc}
               └── {video1, video2, ...}
                      └── video.mov
        ├── annotations .......
```

```
— dataset
    ├── images -- category_videono_frame.jpg
    ├── labels  -- category_videono_frame.txt
    ├── train.csv
    ├── validation.csv
    ├── test.csv
```

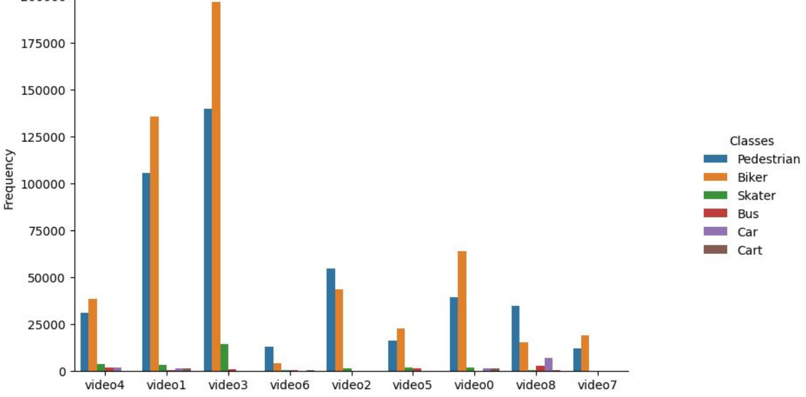
Analysis of the Dataset



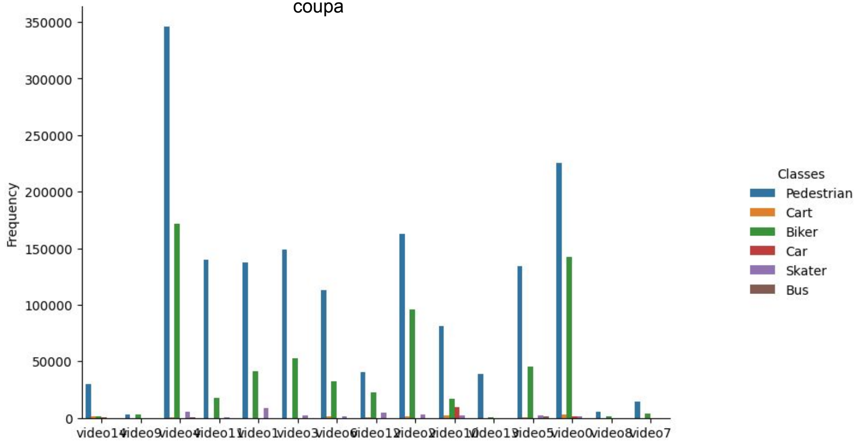
bookstore



coupa

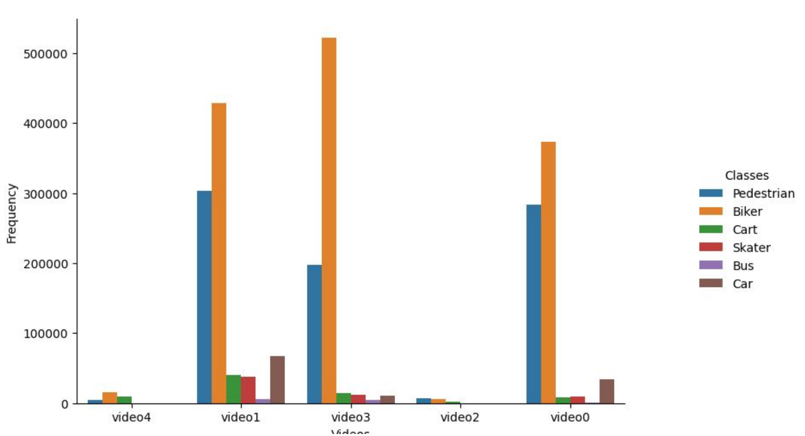


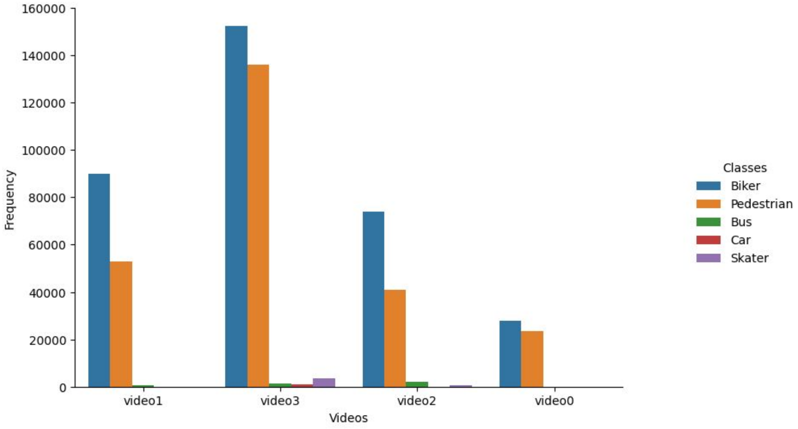
gates

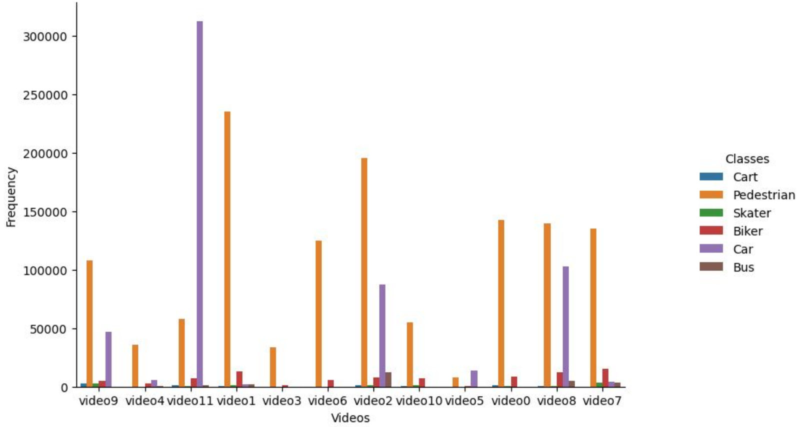


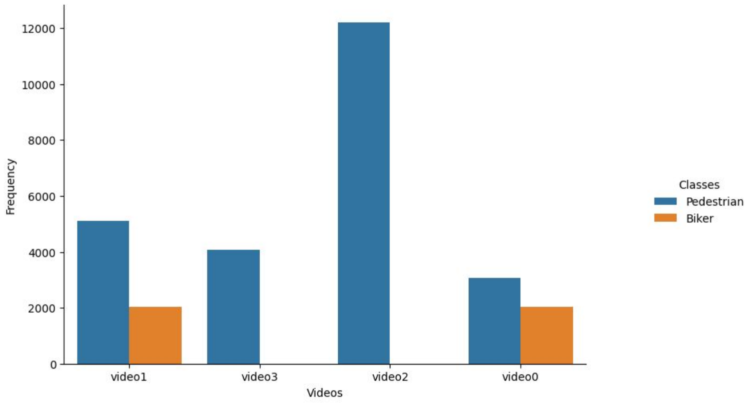
hyang

Analysis of the Dataset

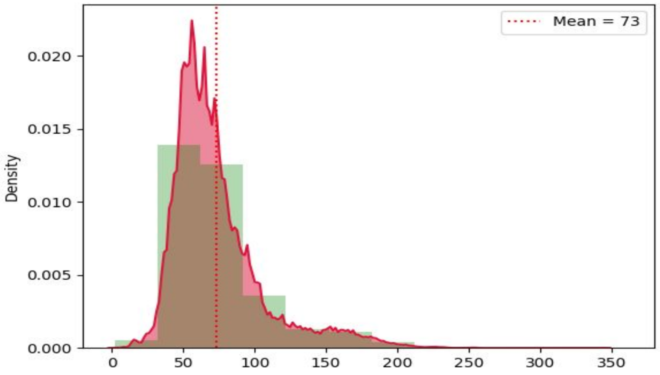


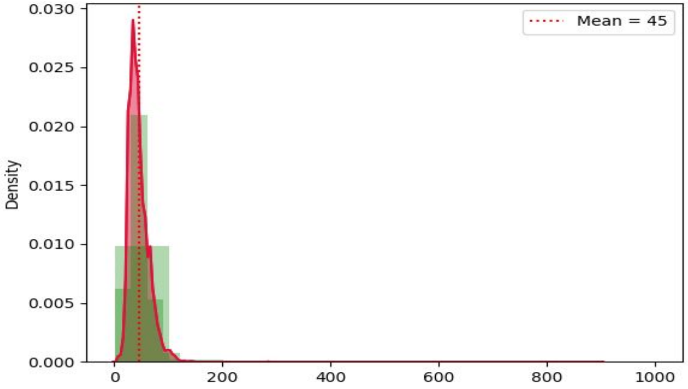




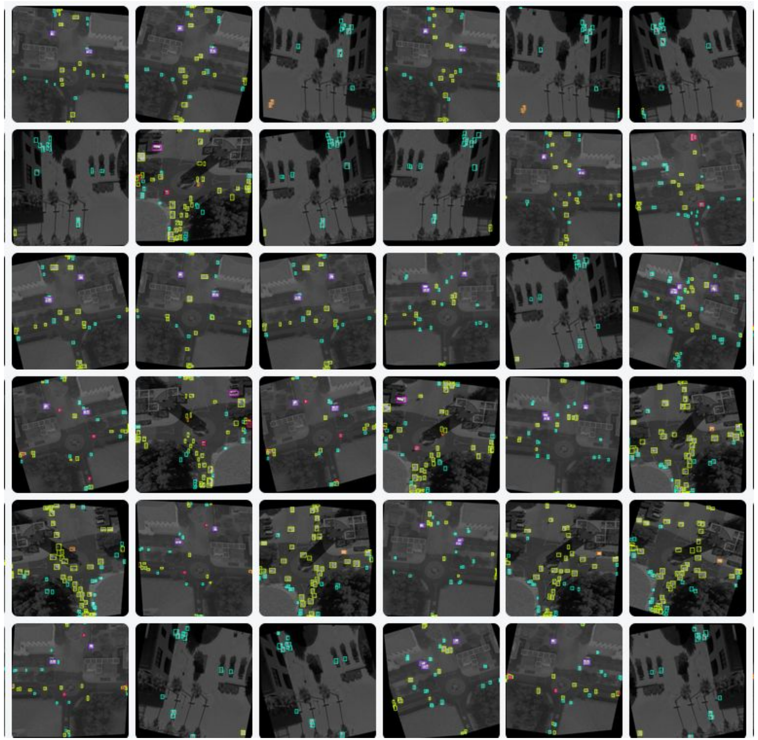


Analysis of the Dataset





Data Preprocessing & Preparation

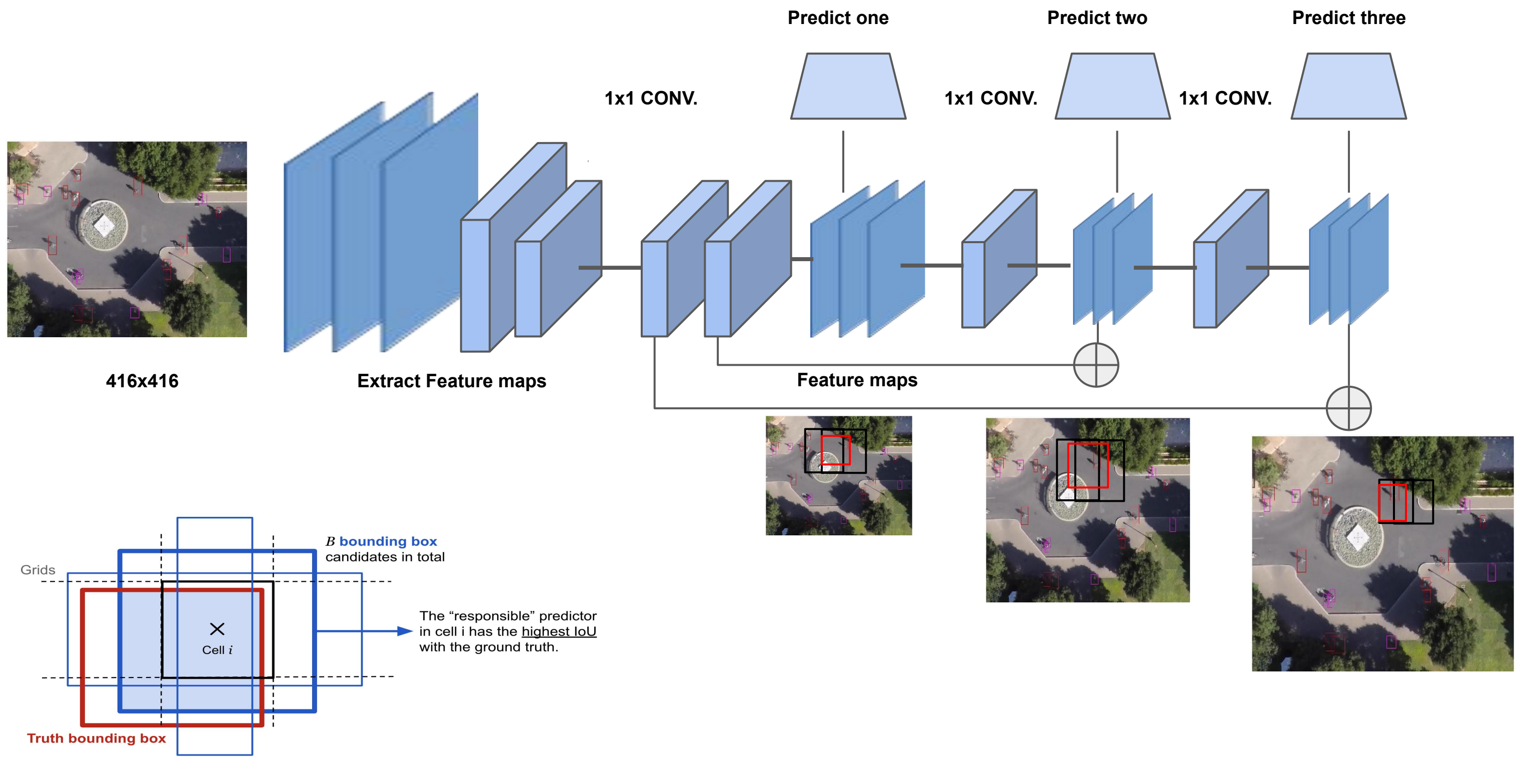


Flip: Horizontal
Rotation: Between -15° and +15°
Shear: ±15° Horizontal, ±15° Vertical

Resize: Stretch to 416x416
Grayscale: Applied

Dataset	No of images	Labels
Train set	4563	50,000+
Validation Set	1543	-
Test Set	1510	-
Total	7486 images	2,08000+

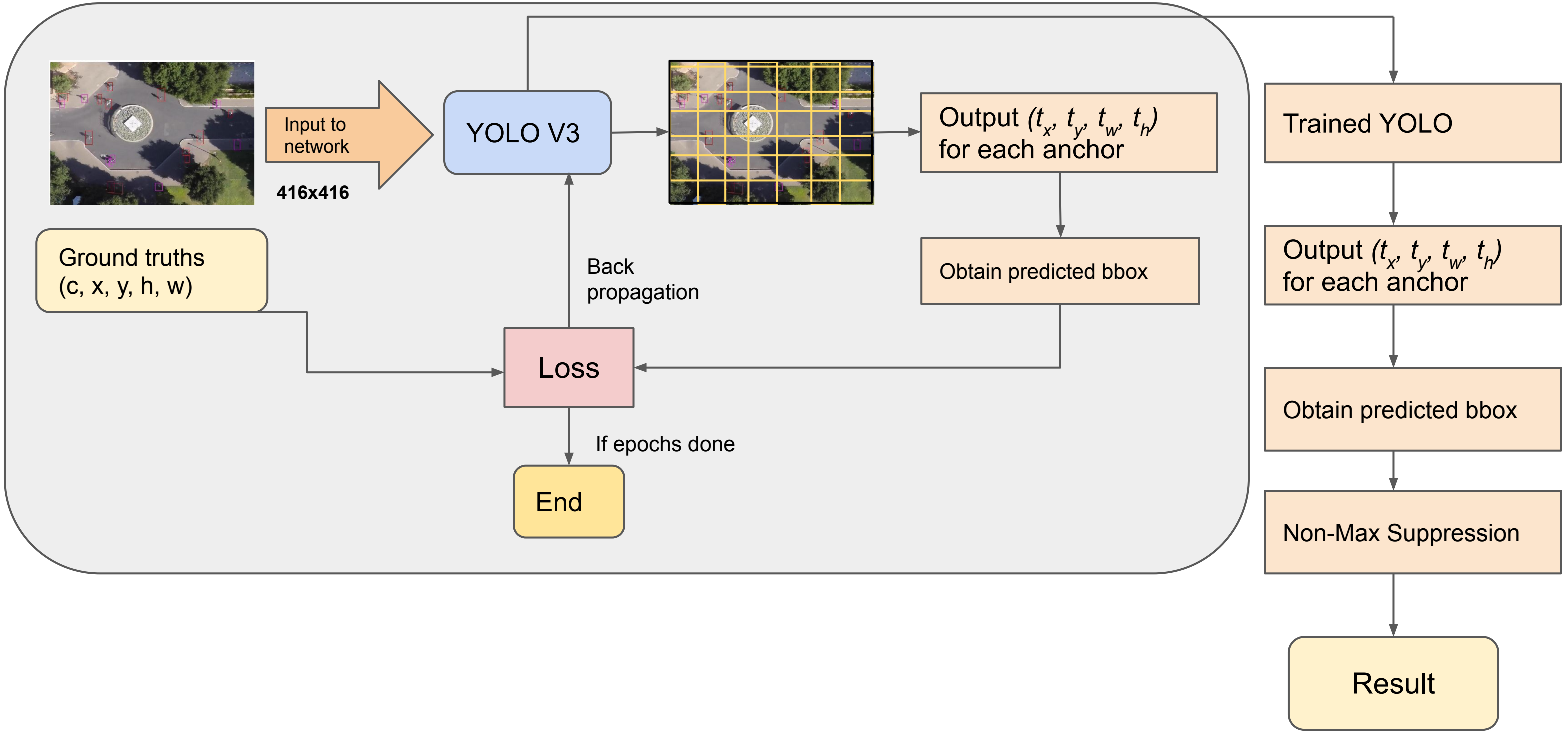
Network Architecture



Workflow

Training Phase

Inference



Loss and Other Maths

4 Losses:

1. MSE of center X, center Y, Width and Height of bounding box

2. BCE of objectness score of a bounding box

3. BCE of no objectness score of a bounding box

4. BCE of multi-class predictions of a bounding box

$$b_x = \sigma(t\,x) + c\,x$$

$$b_y = \sigma(t\,y) + c\,y$$

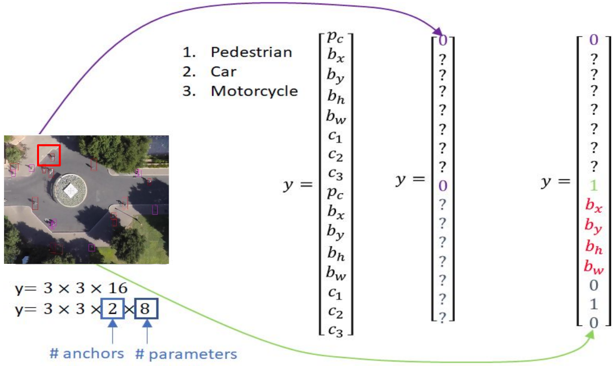
$$b_w = p_w * e^{tw}$$

$$b_h = p_h * e^{th}$$

$$Pr(class_i \mid object) * IOU\,(b, object) * Pr(object) = Pr\,(class) * IOU$$

t_x	t_y	t_w	t_h	p_0	p_1	....	p_5	p_6

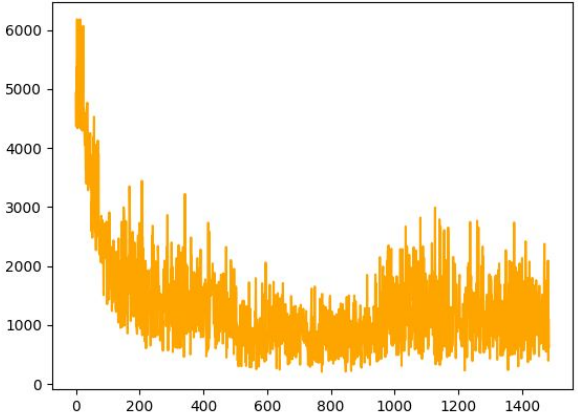
Confidence score



Source: https://towardsdatascience.com/

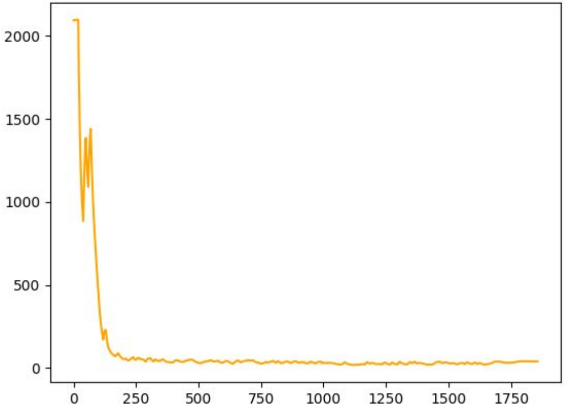
Experiments & Results

YOLO-V4 Pytorch Implementation (Training Loss)



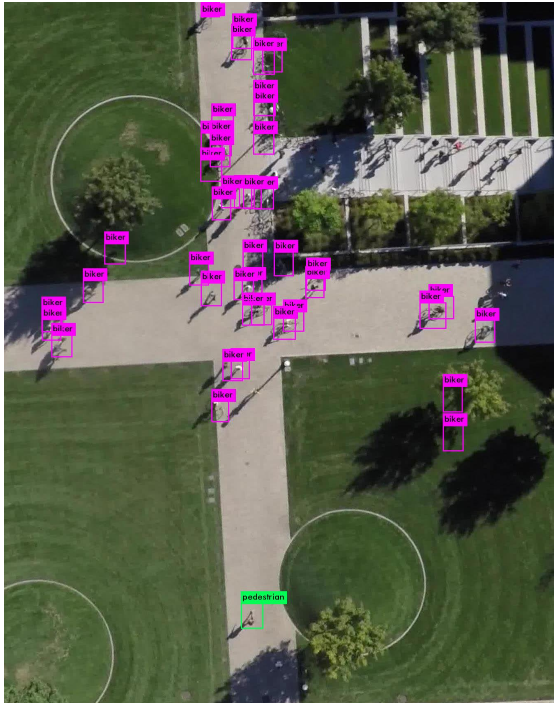
Learning rate	epochs	mAP	Batch size
5e-3	60/250	0.0000	4
1e-4	61/250	0.0000	8

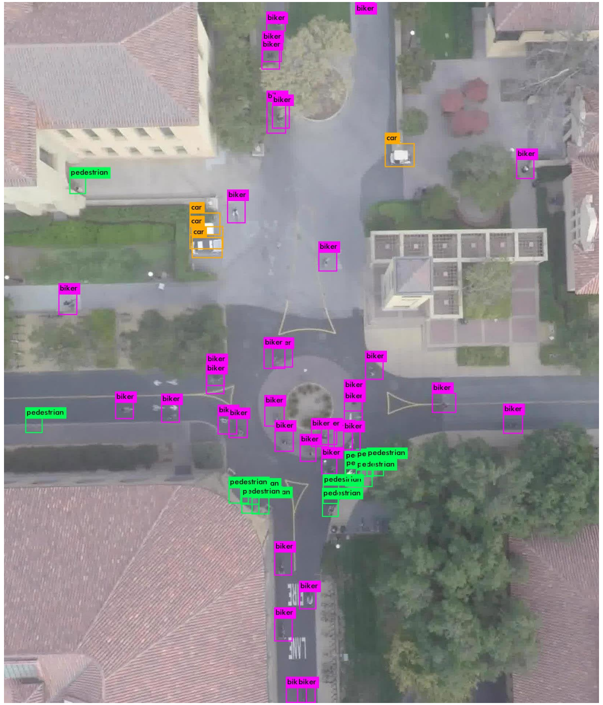
In DarkNet Framework (Training Loss)

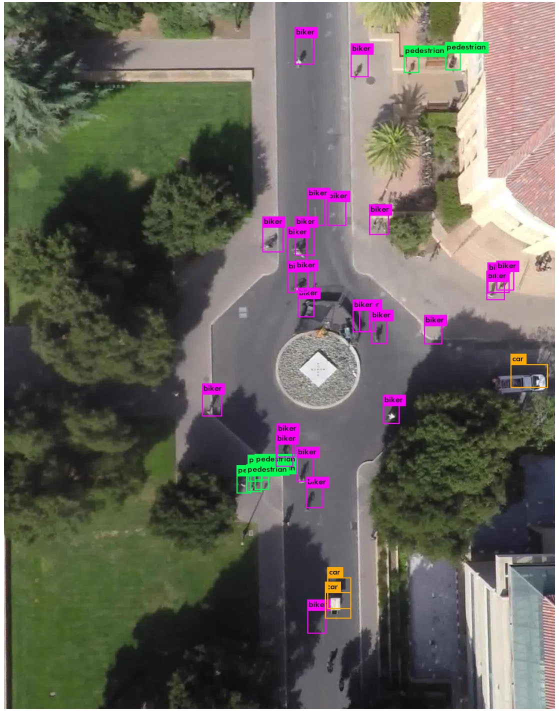


Learning rate	iterations	mAP	Train Set	Test Set
1e-3	2000	26.07%	1000+	200+

Visualization of Output(Demo)







To Dos

- ✓ Clean and Get the new Video data
- ✓ Convert into frames and take frames in 1 sec interval
- ✓ Get labels of each frames
- ✓ Split into train, test, validation
- ✓ Define the anchor values
- ✖ Train Yolo V3 network on it
- ✓ Train on DarkNet Framework
- ✓ Evaluation

References:

1. *A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese, Learning Social Etiquette: HumanTrajectory Prediction In Crowded Scenes in European Conference on Computer Vision(ECCV), 2016.*

2. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi *You Only Look Once: Unified, Real-Time Object Detection*

3. Joseph Redmon, Ali Farhadi *YOLOv3: An Incremental Improvement*

4. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao *YOLOv4: Optimal Speed and Accuracy of Object Detection*