INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH, BHOPAL

DSE 309

Python Final Project Report

Intruder detection using Object Detection Algorithm

Submitted By:

Name: Kunal Shaw	Name: Aman Mishra
Roll No: 19353	Roll No: 19026
Dept: EECS	Dept: CHM

Instructor- Dr. Parthiban Srinivasan

Acknowledgements

We both Aman and Kunal would like to express ours sincere gratitude to our instructor Dr. Parthiban Srinivasan providing their invaluable guidance, comments and suggestions throughout the course of the project.

Also I would like to thank ours friends and family for their support and time to work hard in this course and take out some learning which we could apply in real world.

Content

1. Introduction
2. Dataset
3. Methodology and algorithm

    3.1.Training

    3.2.Performance Analysis and Result

4. Discussion
5. Conclusion
6. References and Sources

Introduction

Privacy and security has become a major concern in today's time.

Especially in hostels where students often don't look their room privacy especially become a issue. This project also has wide application in others security areas.

This model can predict the intruder which can help to sent alert at the run time to the concern authority ,which can measure on time. The model in the project can perfectly detect person and hence help to eliminate others in access to Restricted areas.

The best model to fit in our requirement was to use a neural network with image processing. To be able to build a good task assistant system, it is necessary to implement a good detection method. The present work relies on a YOLOv5 deep neural network, which is one of the fastest and most reliable detectors available nowadays.

A crucial part of this project is to detect correctly the roommates and not to count any other person in them. Computer Vision algorithms have been implemented for various tasks in traffic monitoring for many years, e.g., traffic sign recognition ; intelligent traffic light system ; vehicle speed monitoring ; traffic violation monitoring ; vehicle tracking; vehicle classification ; vehicle counting system on streets and highways ; parking spot detection from the point of view of the car for parking assistants ; and parking spot monitoring .

As data science student, our focus was on the application of the algorithm, data acquisition and data annotation.

Dataset

Data for this project is collected from images of hostel room.

Video(60fps) are captured are captured from mobile phone. Images used for dataset is then extracted from the video by saving the still images frames. The above work is done using python code available as open source.

Images are collected from room from phone camera at different lighting conditions (Daylight, evening light, and night light with artificial lighting system).

The images are then annotated manually with labelling as follows:

- R1=Roommate 1
- R2=Roommate 2
- Unknown= Unknown

For annotation we have used online software (Make Sense) which directly convert the annotation into YOLOv5 types.

Dataset Type	Number of Images
Training	633
Test	91
Validation	181

Self-annotated images at different lighting condition

Devices Used(GPU)

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 495.44       Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:04.0 Off |                    0 |
| N/A   35C    P0    27W / 250W |      2MiB / 16280MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Methodology and Algorithm

Library Used and their versions:

```
# pip install -r requirements.txt

# Base ----------------------------------------
matplotlib>=3.2.2
numpy>=1.18.5
opencv-python>=4.1.2
Pillow>=7.1.2
PyYAML>=5.3.1
requests>=2.23.0
scipy>=1.4.1
torch>=1.7.0
torchvision>=0.8.1
tqdm>=4.41.0

# Logging -------------------------------------
tensorboard>=2.4.1
# wandb

# Plotting ------------------------------------
pandas>=1.1.4
seaborn>=0.11.0

# Export --------------------------------------
# coremltools>=4.1  # CoreML export
# onnx>=1.9.0  # ONNX export
# onnx-simplifier>=0.3.6  # ONNX simplifier
# scikit-learn==0.19.2  # CoreML quantization
# tensorflow>=2.4.1  # TFLite export
# tensorflowjs>=3.9.0  # TF.js export

# Extras --------------------------------------
# albumentations>=1.0.3
# Cython  # for pycocotools https://github.com/cocodataset/cocoapi/issues/172
# pycocotools>=2.0  # COCO mAP
# roboflow
thop  # FLOPs computation
```

Training

For this experiment, 4 different models for training our dataset which are

I. YOLOv5s
II. YOLOv5l
III. YOLOv5x
IV. YOLOv5m

Models for better performance prediction.

For training procedure consisted of 300 epoches (we have used google colab with P20 GPU for fast training).

We have used 633 of the 905 images for training and 181 were used for validation.

On training our dataset out of 633 annotated images used 50 images had annotation missing, corrupted annotation images:0, empty annotated images:0.

We have a batch size of 16. Hence no. of batches = 683/16 = 43.

Model evaluation indicators

In this model we introduce indicators like precision (P), which is precision rate, recall rate (R), and mean average precision (mAP) to evaluate the performance of the intruder detection model.

The expressions of P and R are as follows:

$$P = TP\ (TP + FP)$$

$$R = TP\ (TP + FN)$$

Among them, true positives (TP), false positives (FP), and false negatives (FN), respectively, represent positive samples with correct classification, negative samples with incorrect classification, and positive samples with incorrect classification.

AP is the average accuracy rate, which is the integral of the P index to the R index,

that is, the area under the P–R curve; mAP is the average accuracy of the mean, which

means that the AP value of each category is summed, and then divided by all categories,

i.e., the average value. They are defined as follows:

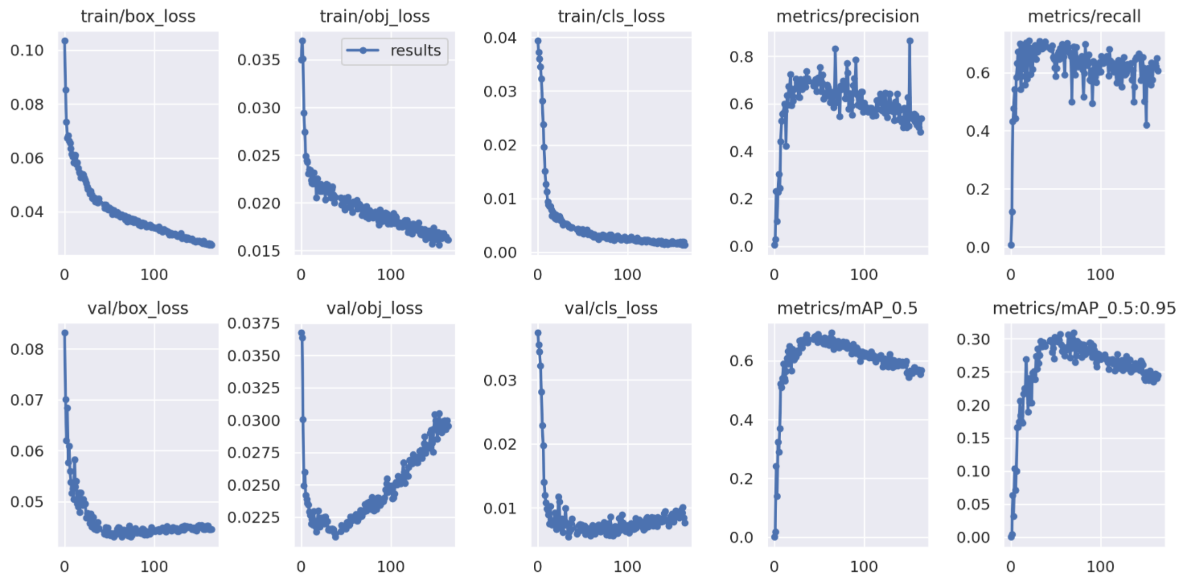
$$AP = \int_0^1 P(R)dR$$

$$mAP = \frac{1}{|Qr|}\sum_{q=Qr}^{q=1} AP(q)$$

Where Qr is the number of categories.

The improvement in our model can be seen in the graphs below,

which display different performance metrics for both the training and validation sets.



Graph shows box loss, objectness loss, classification loss, precision, recall and mean average precision (mAP) over the training epochs for the training and validation set.

We encounter three type of loss:

- Box loss(train/box_loss and val/box_loss):

  The box loss represents how well the algorithm can locate the centre of an object and how well the predicted bounding box covers an object

- Objectness loss(train/obj_loss and val/obj_loss):

  Objectness is essentially a measure of the probability that an object exists in a proposed region of interest. If the objectivity is high, this means that the image window is likely to contain an object

- Classifiaction loss(train/cls_loss and val/cls_loss):

  Classification loss gives an idea of how well the algorithm can predict the correct class of a given object.

Performance analysis

On training our model on train dataset on 4 different model(i.e YOLOv5s,x,l,m), we have prediction for our new and unseen pictures in our test.

The algorithm can detect roommate 1(R1), roommate 2(R2), and the intruder(Unknown) to a reasonable extent.

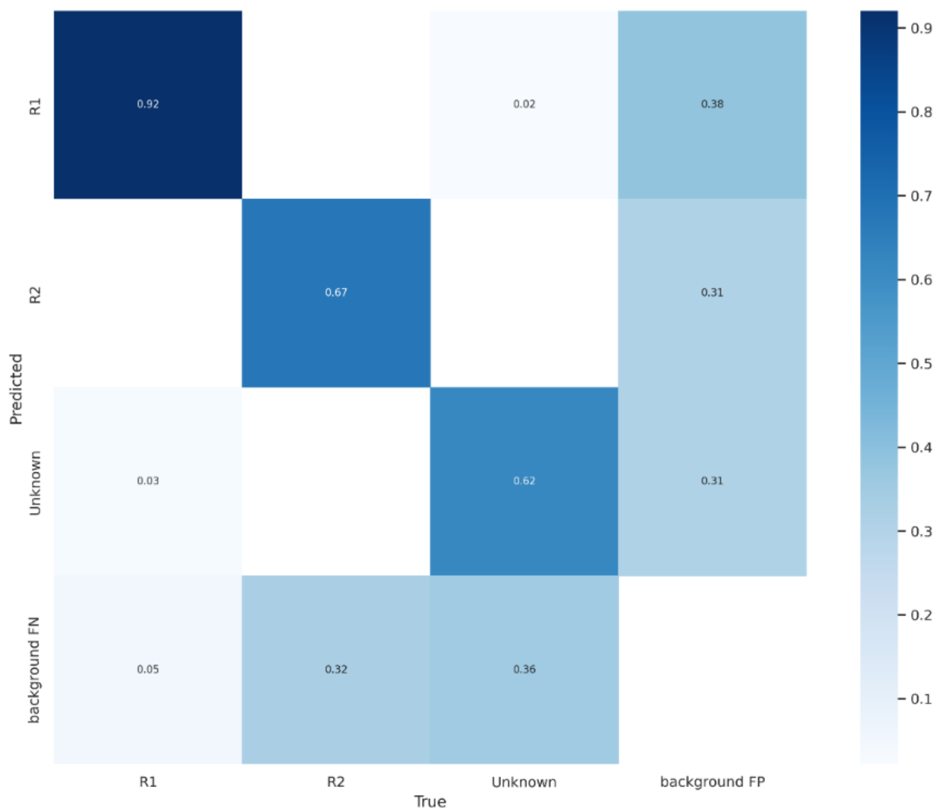
The model can detect the intruder and the roommates at different lighting condition and different position .

Performance of model based on first 20 epoches

epoch	train/box_loss	train/obj_loss	train/cls_loss	metrics/precision	metrics/recall	metrics/mAP_0.5	metrics/mAP_0.5:0.95	val/box_loss	val/obj_loss	val/cls_loss	x/lr0	x/lr1	x/lr2
0	0.098743	0.034858	0.037948	0.0016264	0.09725	0.0011536	0.00027158	0.0784	0.037522	0.037105	0.00042	0.00042	0.09622
1	0.078574	0.03753	0.036518	0.10978	0.2554	0.070068	0.016237	0.065573	0.036045	0.03569	0.00084998	0.00084998	0.09235
2	0.069297	0.03364	0.035256	0.12278	0.36038	0.089247	0.016078	0.066534	0.028057	0.03421	0.0012799	0.0012799	0.08848
3	0.065384	0.027222	0.033587	0.19916	0.41783	0.21787	0.070939	0.066691	0.025621	0.031629	0.0017096	0.0017096	0.08461
4	0.066693	0.026156	0.030295	0.28417	0.41295	0.29035	0.077332	0.06152	0.02475	0.024261	0.0021392	0.0021392	0.080739
5	0.064447	0.024207	0.023398	0.6887	0.33402	0.40467	0.10397	0.059434	0.023678	0.016971	0.0025684	0.0025684	0.076868
6	0.064434	0.023489	0.016786	0.4629	0.65067	0.52573	0.1596	0.05419	0.023364	0.011801	0.0029973	0.0029973	0.072997
7	0.060504	0.023227	0.012406	0.27562	0.59007	0.28034	0.068124	0.059861	0.022147	0.011603	0.0034259	0.0034259	0.069126
8	0.059125	0.023233	0.010606	0.61842	0.69679	0.64257	0.24416	0.048417	0.022334	0.0083761	0.0038539	0.0038539	0.065254
9	0.056306	0.02249	0.0087613	0.57914	0.68938	0.61495	0.24209	0.050258	0.021964	0.0075974	0.0042814	0.0042814	0.061381
10	0.054441	0.021756	0.0072208	0.57327	0.65185	0.60751	0.23747	0.049393	0.021671	0.0071962	0.0047084	0.0047084	0.057508
11	0.055583	0.021214	0.0065962	0.51963	0.5713	0.52444	0.1843	0.051152	0.021726	0.0084845	0.0051346	0.0051346	0.053635
12	0.05598	0.021045	0.0070892	0.59546	0.5534	0.57068	0.20402	0.053311	0.021387	0.0070623	0.0055602	0.0055602	0.04976
13	0.055711	0.021077	0.0061215	0.5176	0.66932	0.5607	0.18737	0.053597	0.021456	0.0067551	0.005985	0.005985	0.045885
14	0.054089	0.020649	0.0063528	0.63363	0.64552	0.6363	0.24617	0.047488	0.021402	0.0065353	0.0064089	0.0064089	0.042009
15	0.051229	0.021173	0.006171	0.58441	0.63546	0.58445	0.22656	0.04978	0.021347	0.006902	0.0068319	0.0068319	0.038132
16	0.050826	0.020687	0.0056807	0.69216	0.66431	0.66524	0.2652	0.047153	0.021745	0.0067596	0.007254	0.007254	0.034254
17	0.04982	0.02008	0.0056239	0.64755	0.67055	0.64501	0.25925	0.047819	0.021352	0.0060159	0.007675	0.007675	0.030375
18	0.050403	0.020055	0.0049762	0.69431	0.70299	0.66386	0.25613	0.047098	0.021244	0.0062892	0.008095	0.008095	0.026495
19	0.049491	0.020132	0.005375	0.51143	0.53113	0.49088	0.14336	0.055776	0.020862	0.0067851	0.0085137	0.0085137	0.022614
20	0.052779	0.020059	0.0057594	0.73417	0.69799	0.67517	0.27783	0.047437	0.02128	0.0067954	0.0089313	0.0089313	0.018731

Here we can see that mAP value increases with increases in the number of epochs , indicating our prediction is going well.

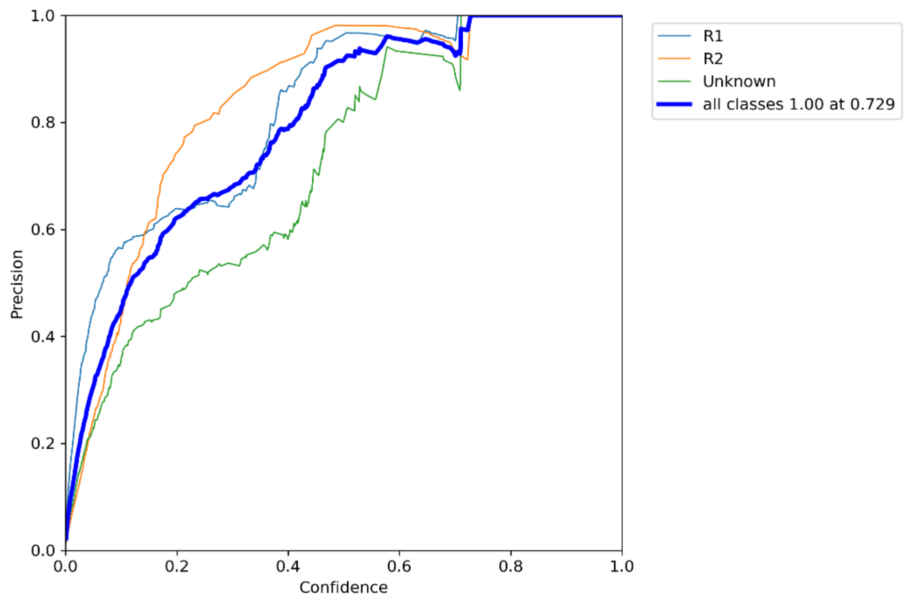
Various data analysis of model prediction



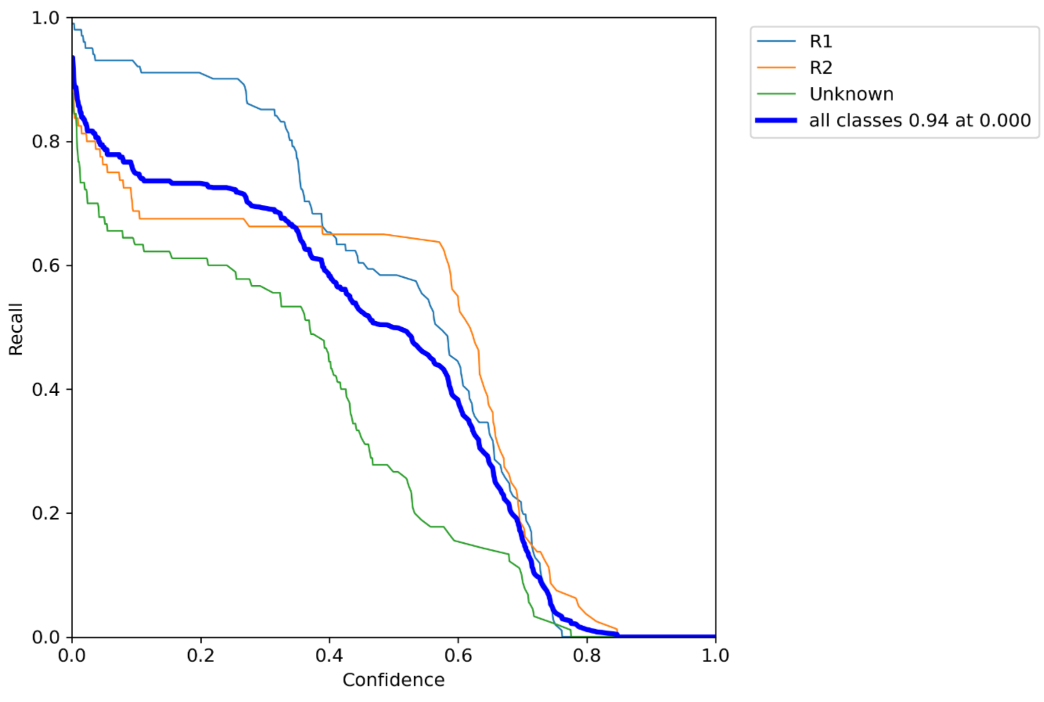
Confusion matrix of the predicted and actual class

Here we can observed the positive predicted and true predicted for R1=0.92, for R2=0.67, Unknown=0.63

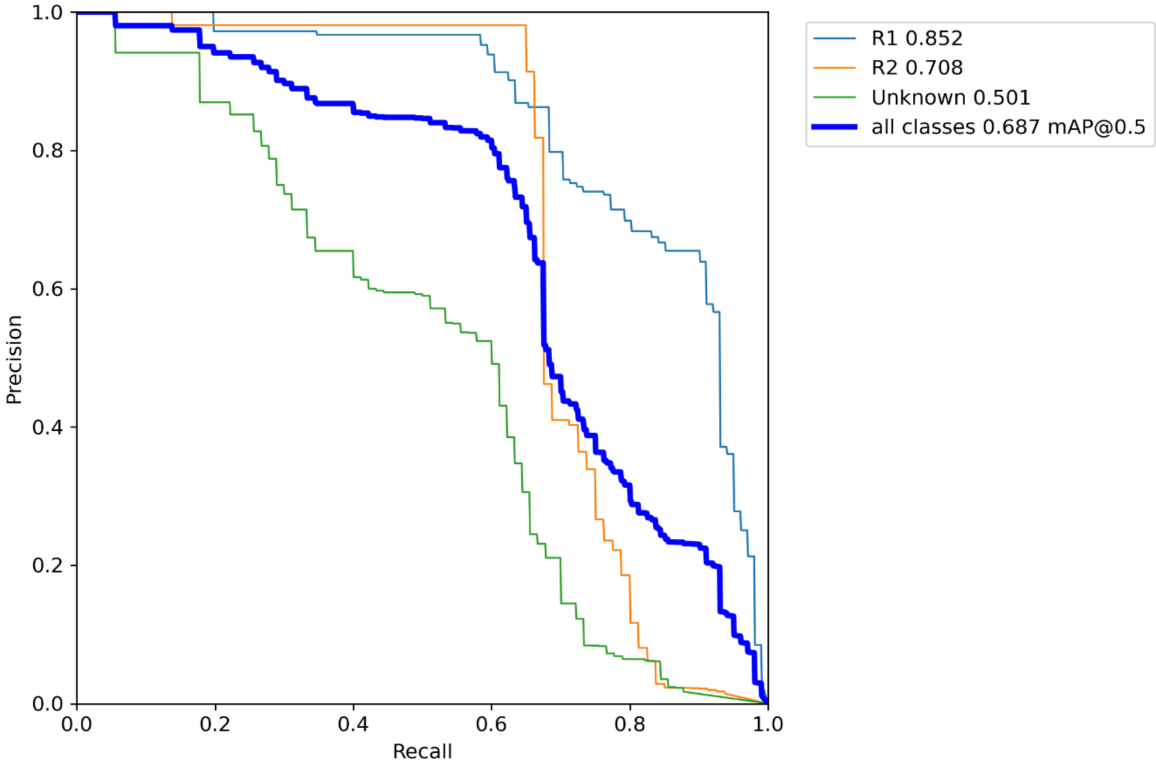
Also background prediction of positive true is also very good.



Precision curve of various classes



Recall curve for various class



Precision vs recall value for three class

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 3/3
all	91	150	0.65	0.706	0.678	0.287
R1	91	59	0.699	0.915	0.865	0.374
R2	91	34	0.677	0.588	0.624	0.26
Unknown	91	57	0.574	0.614	0.545	0.226

Performance on test set for yolov5l

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 3/3
all	91	150	0.923	0.517	0.66	0.283
R1	91	59	0.905	0.644	0.81	0.375
R2	91	34	0.951	0.573	0.617	0.251
Unknown	91	57	0.912	0.333	0.554	0.222

Performance on test set for yolov5m

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 3/3
all	91	150	0.668	0.639	0.66	0.296
R1	91	59	0.694	0.814	0.833	0.39
R2	91	34	0.692	0.559	0.621	0.264
Unknown	91	57	0.618	0.544	0.525	0.233

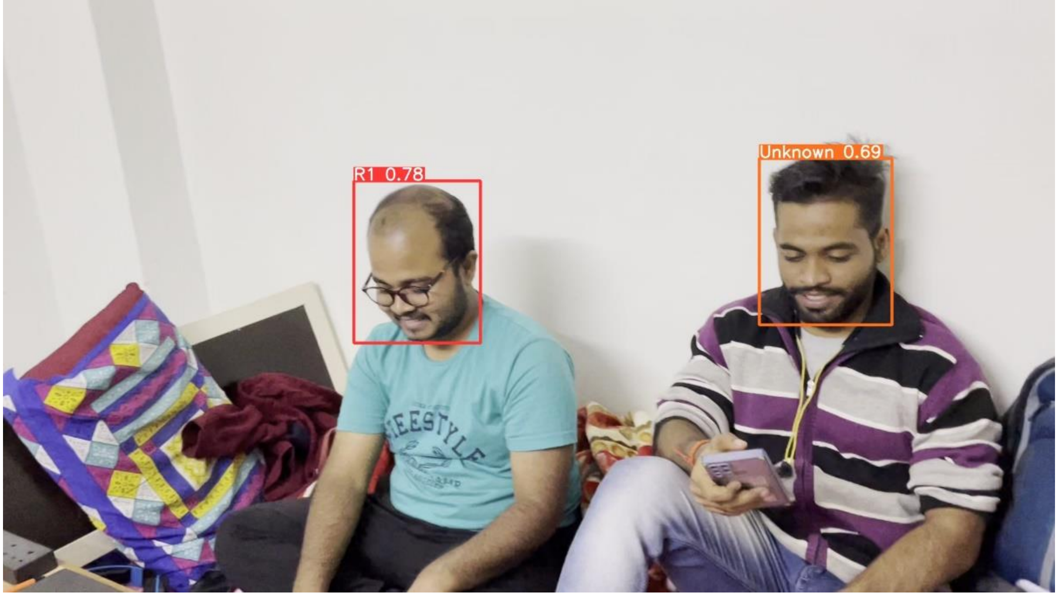
Performance on test set for yolov5s

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 3/3
all	91	150	0.69	0.655	0.673	0.295
R1	91	59	0.75	0.729	0.824	0.368
R2	91	34	0.743	0.588	0.658	0.296
Unknown	91	57	0.577	0.649	0.537	0.22

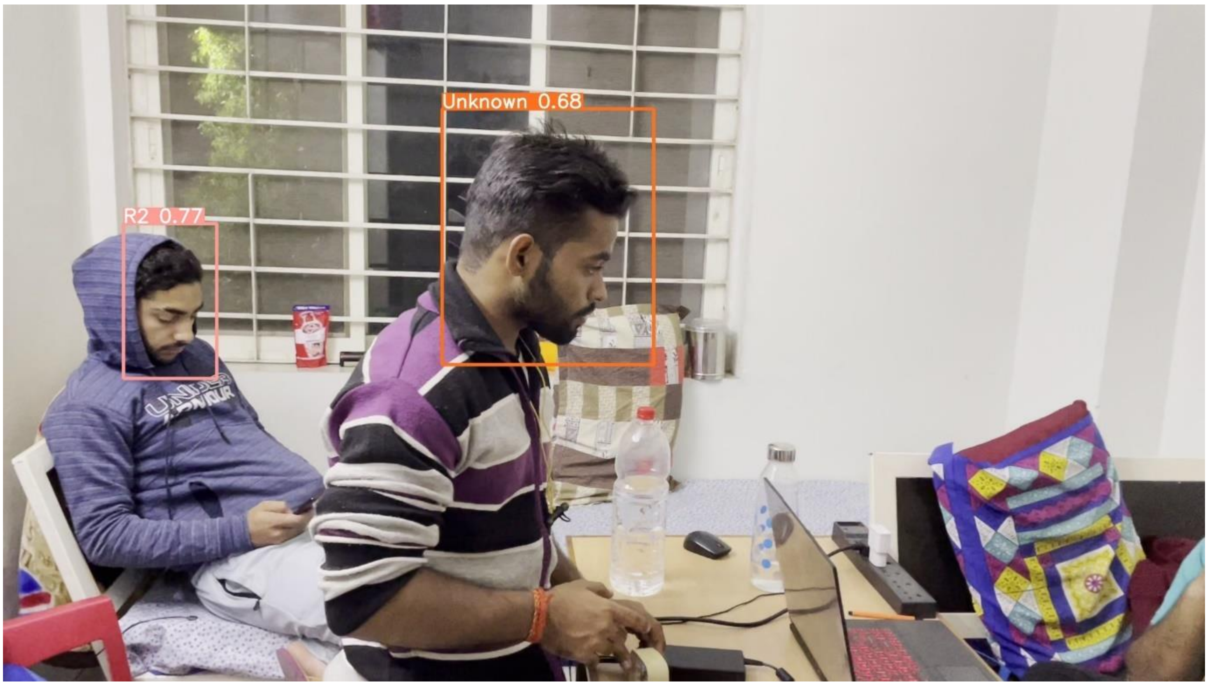
Performance on test set for yolov5x

Output Images with model detecting roommate 1, roommate 2 and intruder

Detection of R1(as roommate 1) and Unknown (as intruder)



Detection of R1 (as roommate 1) and Unknown (as intruder)



Detection of R2 (as roommate 2) and unknown(as intruder)

Our model can predict roommate 1 ,roommate 2 and all other person other than them as unknown which are intruder in our case.

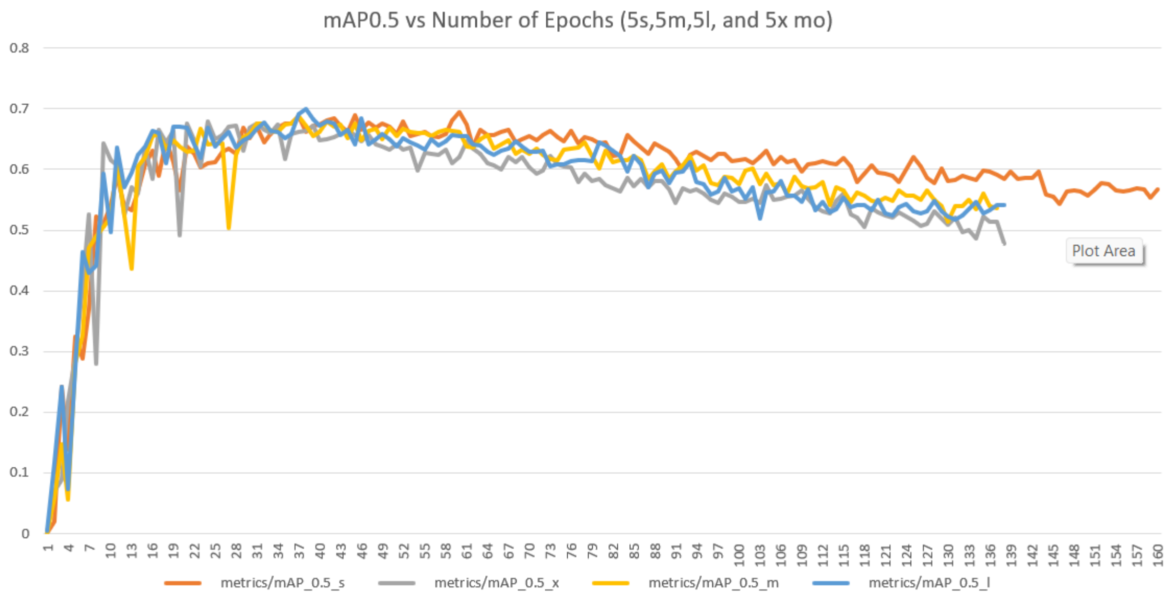
The percentage across various square check boxes represent the probability of detection of person.

Some error are there in the prediction of intruder due to less number of image dataset to train. Also, sometime background error are also there in the prediction.

Comparison of Model YOLOv5s, Model YOLOv5m, Model YOLOv5l and Model YOLOv5x

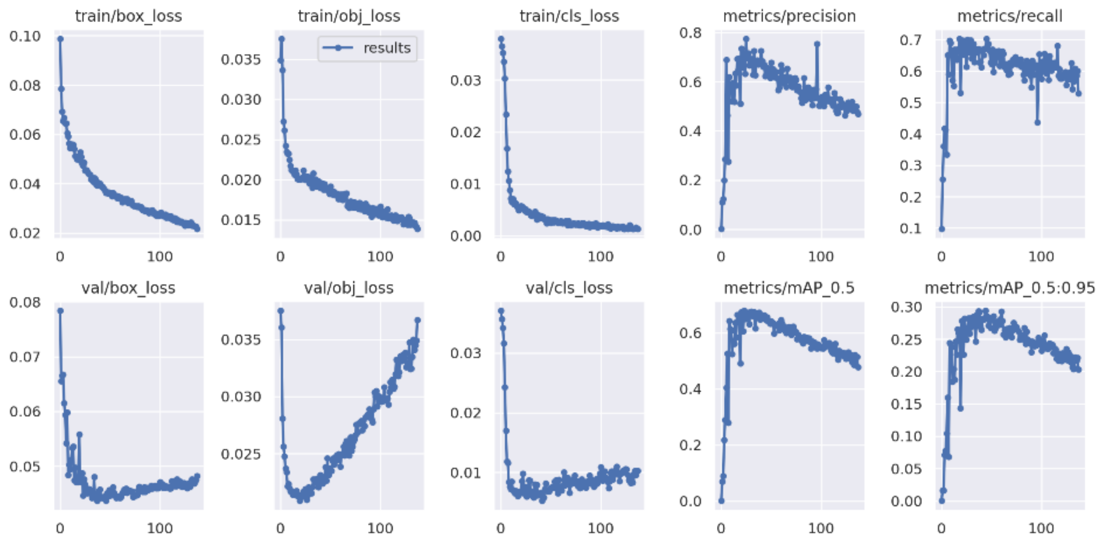
In terms of speed YOLOv5x is fastest among all.

The mAP0.5 value are in the order of YOLOv5l> YOLOv5s> YOLOv5x> YOLOv5m

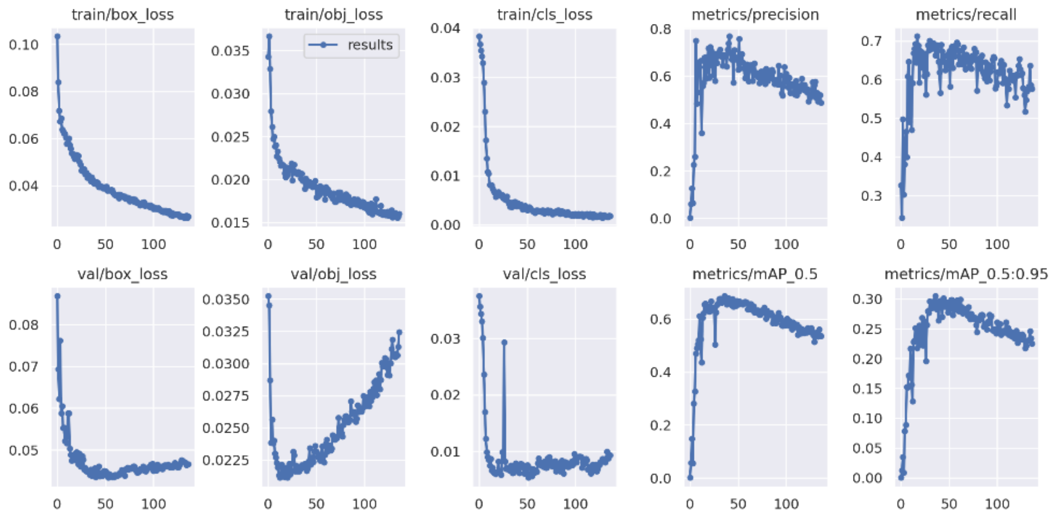


Comparison of various model on the basis of mAP0.5 value

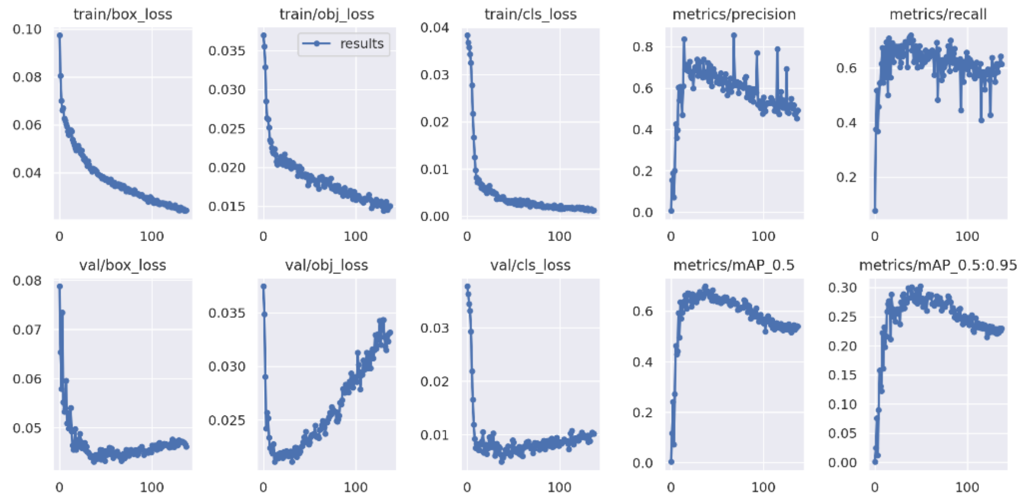
The loss function shows the performance of a given predictor in classifying the input data points in a dataset. The smaller the loss, the better the classifier is at modelling the relationship between the input data and the output targets. There are two different types of loss shown in Figure 6. The loss represented at the top is related to both the predicted bounding box and the loss related to the given cell containing an object during the training. The graphs of val Box and val Objectness represent their validation scores. Training loss is measured during each epoch while validation loss is measured after each epoch



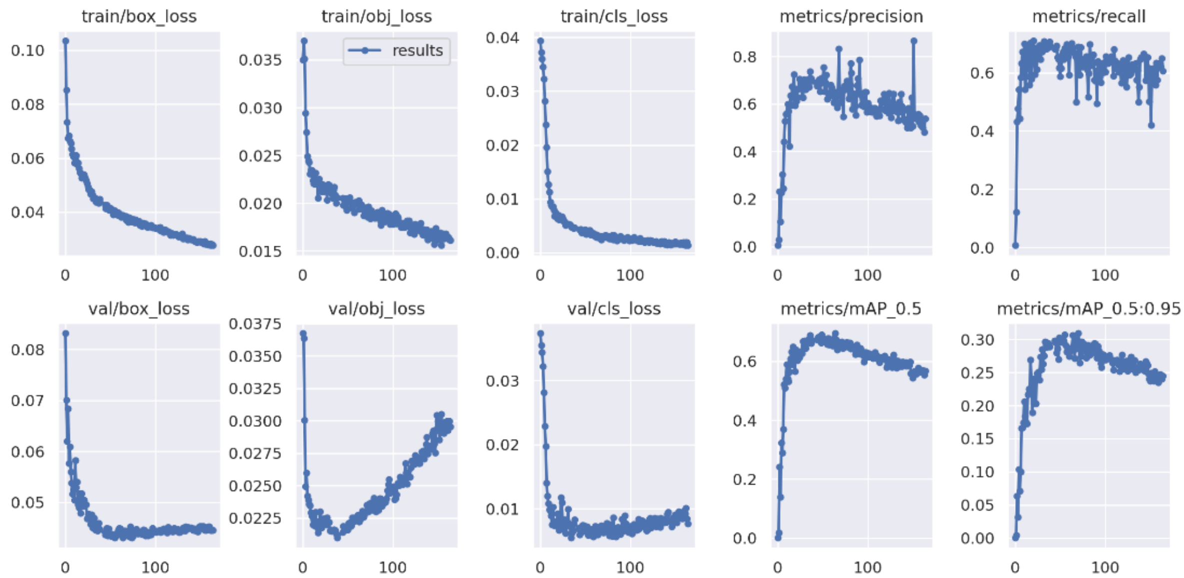
YOLOv5x model result



YOLOv5m model result



YOLOv5l model result



YOLOv5s model result

Here, we can observe the 4 different models and compare them on various parameters like loss functions, precisions, mAP_0.5,recall etc.

Discussion

Using a YOLO deep network for object detection we have detect the intruder in hostel room. The model of object detection trained learns quickly and gives a prediction in a fraction of a second, making it suitable for use in real time.

For most applications, changes to the physical data collection cannot be influenced.

There are some also failure in detection which may be due to small dataset of images. Quite a few time the detection for R2(roommate 2) is getting corrupted with unknown.

The background detection also sometime detect non-human(such as water bottle, blanket etc) as human .

A sample of above false detection can be seen below

Prediction of R2 is incorrect (as roommate 2 is not in the image) and also background object are detect as human

The prediction can be increase by increasing the dataset by at least 1000 annotated images of each of the three class(R1,R2 and Unknown)

In terms of processing accuracy, the dataset of this study is manually captured images; hence, the background information is relatively simple. In slightly complex background conditions, the accuracy may be reduced. Thus, there will be no complex background in practical application.

In terms of model generalization ability, YOLOv5 uses a mosaic data enhancement strategy to improve the model's generalization ability and robustness.

Based on the above discussion, we believe that the method we proposed is an effective exploration and can be used to detect intruders in room when both roommates are not present.

Conclusions and Future Work

The main goal of the project is to detect intruder inside the hostel room using a neural network for deep learning. To train the model, four different models were used. Four versions of YOLOv5 were also tested, and it was determined that YOLOv5s can be sufficient for the intended detection problem.

However, we already showed that when analysing images from small angle cameras to detect persons have a high number of overlaps and cut-offs, the model can be improved by detecting certain characteristic features instead of the whole object. Furthermore, the usage of thermal network cameras at night can enhance the prediction.

We are confident that with a bigger training set and the implementation of the changes suggested in discussion section, the algorithm can be improved even further.

We will implement this prediction with message sending system so that roommate can get notification and alert about intruder.

References and Sources

Model creation and source:-

I. https://github.com/ultralytics/yolov5
II. https://ultralytics.com/
III. https://wandb.ai/site

Reference papers and articles:-

- https://www.mdpi.com/1192268
- https://www.mdpi.com/1054952
- https://www.mdpi.com/1119922
- Evaluating Object Detection Models Using Mean Average Precision (mAP)(Link)
- https://hcis-journal.springeropen.com/articles/10.1186/s13673-020-00219-9