Person Detection in Aerial Images

Name & Roll No. (Member 1): **Pranav Deep Indrakanti** (**S20201010007)**

Abstract:
To detect human presence in aerial imagery. Variety of applications available, particularly in the area of Search and Rescue Operations(SAR ops)

Motivation:

- India ranks 89th in disaster management according to the World Risk Index (WRI) 2020, even lower than Sri Lanka, Bhutan & Maldives.
- Search and Rescue Missions (SAR) missions in India are generally limited to Coast Guard or military operations only. (ISRO's intervention in SAR missions has been a very recent trend, but mostly in case of high profile military operations like Indo-China border rescue missions etc.)
- Local bodies(say Municipality) which have more statistics and information about the region have very limited resources and/or access to execute SAR missions during calamities or otherwise.
- But all is not gloomy, as the procurement & setup cost of UAVs has significantly decreased.
- However, automation in this domain is more/less not explored(Indian context).
- To start with, I want to detect persons in aerial images in imagery that has been captured during calamity-like scenarios.

Datasets Details (HERIDAL DATASET) [1]

Patch Data
Positive and Negative Samples
Image Size : 81 * 81
Channels : 3
+ve Images : ~29,000
-ve Images : ~32,000
Images are taken with a high resolution camera on the DJI Phantom 3, vertically at a 50 m altitude with the selected

Actual Data
Image Size : 4000 * 3000
Channels : 3
No. of Images : ~1500
Note : The dataset originally contains Train and Test images - But I've merged all of them and made my own splits as the splits they gave were skewed

Train, Val, Test - 64%, 18%, 18%

Models Used :

1) **Classification** (using Patch Data)
   a) Machine Learning - **SVM**
   b) Deep Learning - **Custom Network**
2) **Detection -** using Actual Images
   a) YOLO v5

Deep Learning Network [3]

Figure 1. Classification network architecture.

Tweaks to the above model

Changed final Activation fn. To sigmoid instead of softmax
Trainable params: 522,561
Epochs = 20
Batch Size = 32
Loss Fn. = Binary Cross Entropy
Optimizer = Adam
Used Dropout Layer - 0.5
This network is used for classification of patch data

YOLO v5 [2]

Q: **Why this model?**

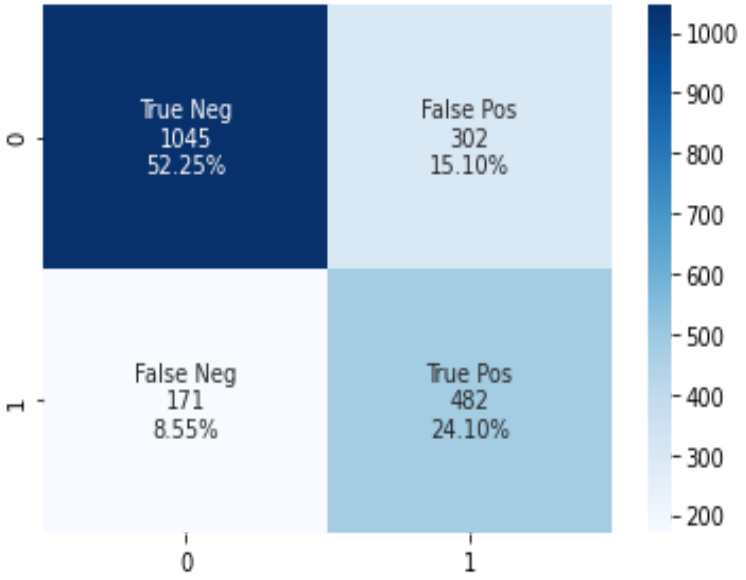
a) It has a **distributed aspect ratio** which makes it more robust for various aspect ratios
b) **Automatic new anchors** - The model can generate new anchors if the given anchors < threshold
c) It uses models like K-Means, Genetic Algorithm to generate anchors
d) Uses a **Dense Network** with Body, Head and Neck
e) **Head** captures most of the major features
f) **Neck** captures the minor features
g) **Data Augmentation**:
h) a) High degree of scaling augmentation
i) b) Good Color-Space Augmentation
j) Image Size (pixel) is the same as that of the image size - thereby increasing the volume of the batch.
k) This increases **variation**
l) Overall, this architecture makes the computation(time+memory) very efficient.

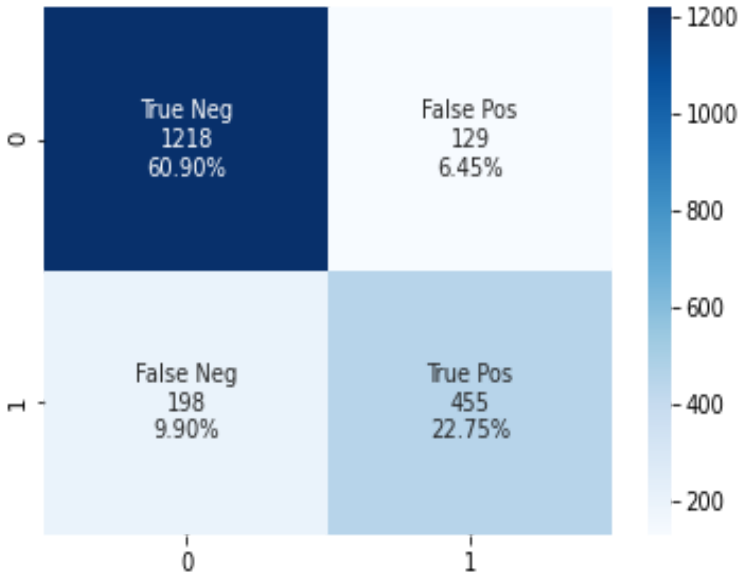
Results (Classification)

	Linear SVM	Polynomial SVM	DL Classifier
Training Accuracy	100%	96.7 %	98.769 %
Validation Accuracy	NA	NA	98.86 %
Testing Accuracy	76.35 %	83.65 %	98.86 %
Precision	61.5 %	77.9 %	98.68 %
Recall	73.8%	69.7 %	97.36 %

SVM Confusion Matrices

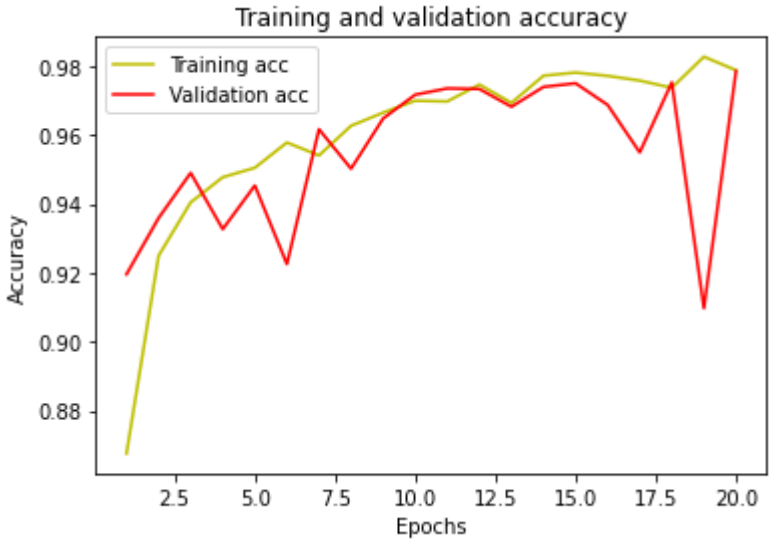
Linear SVM

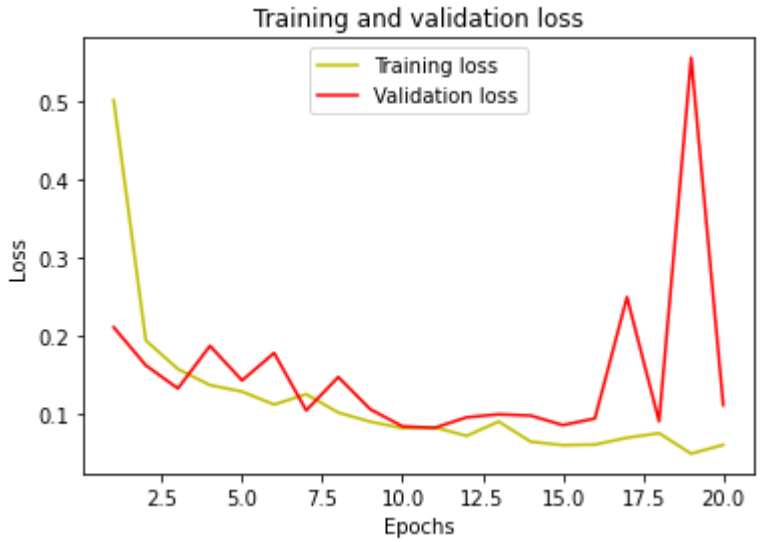




Polynomial SVM

DL Model's Plots



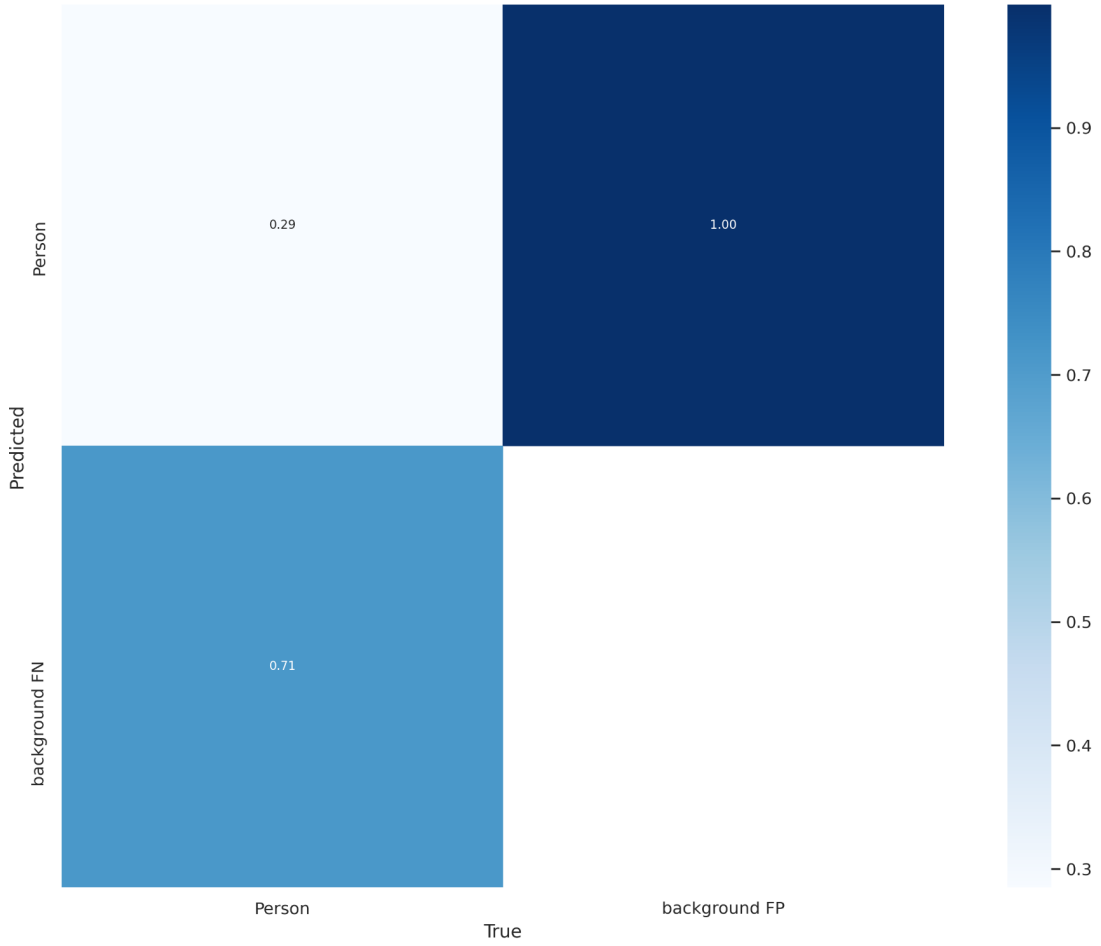


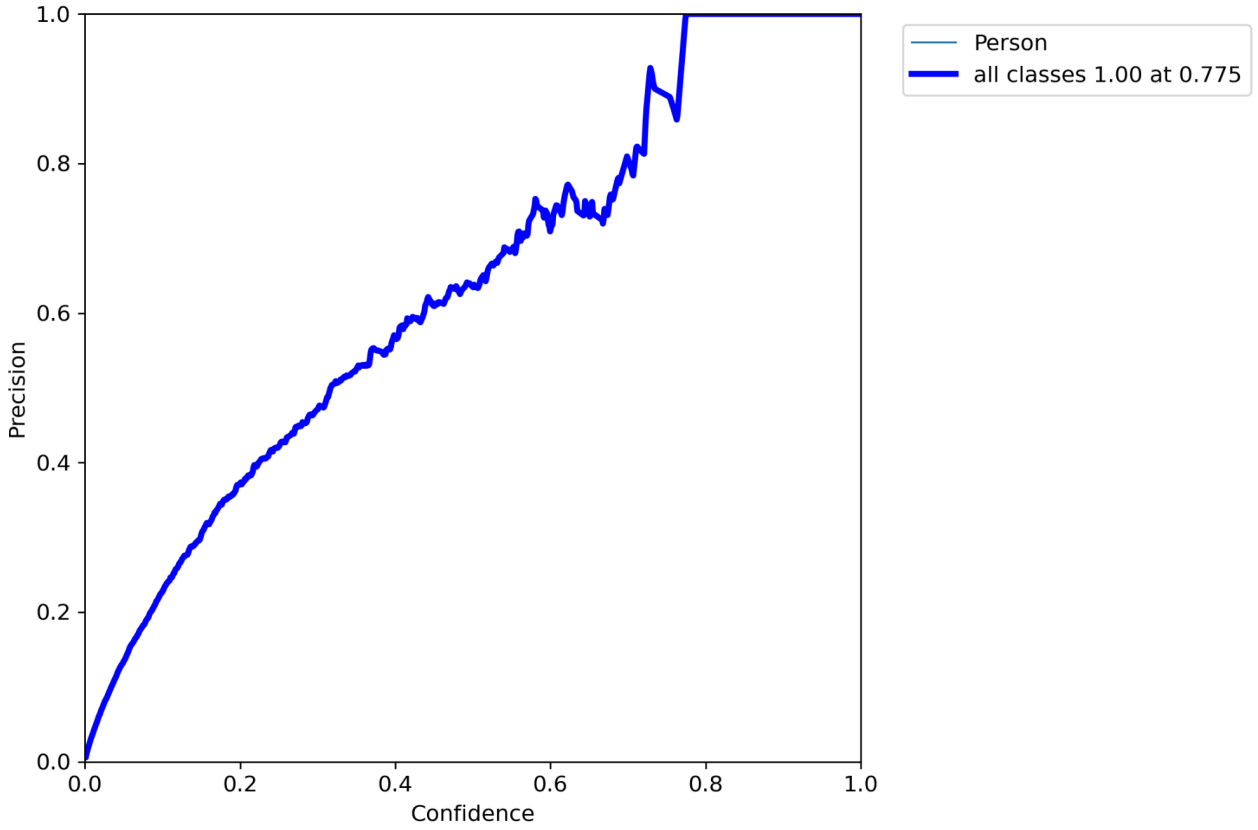
YOLO v5

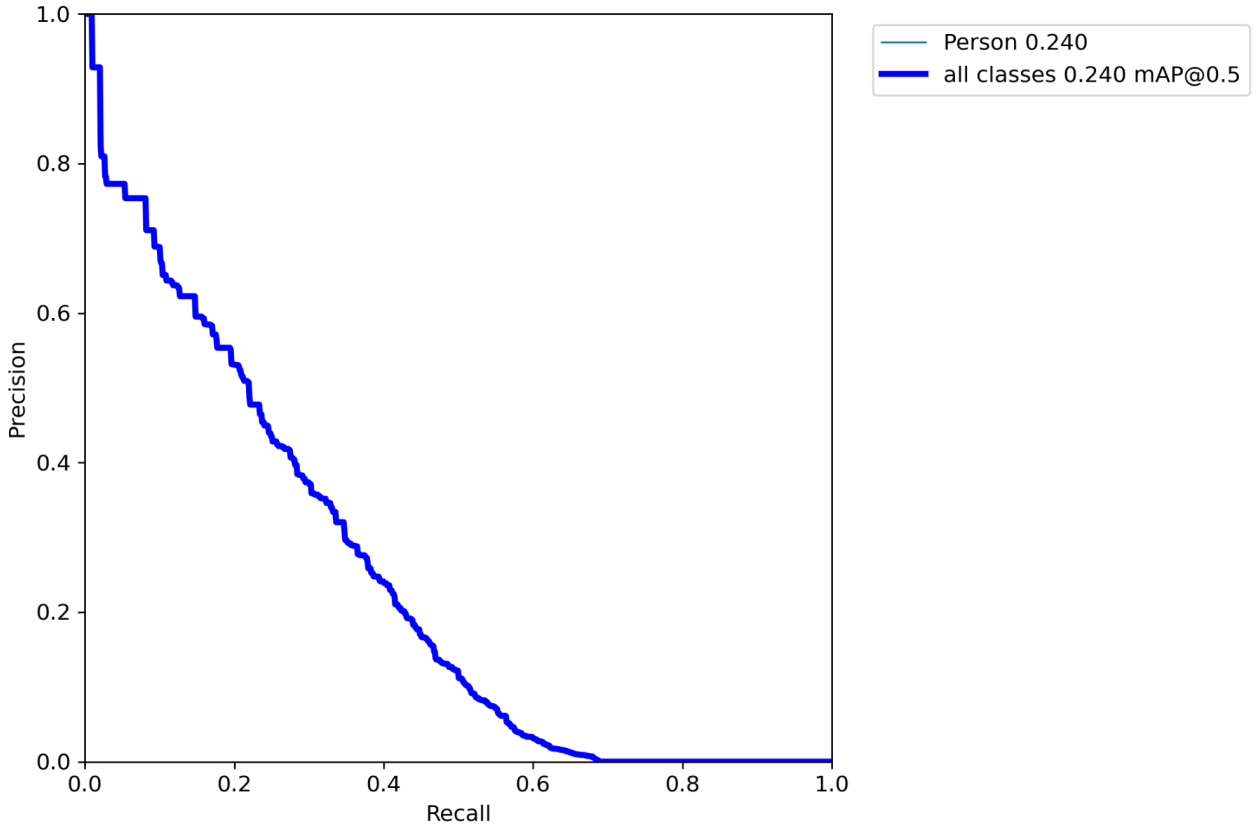
Hyper Parameters for Yolo v5

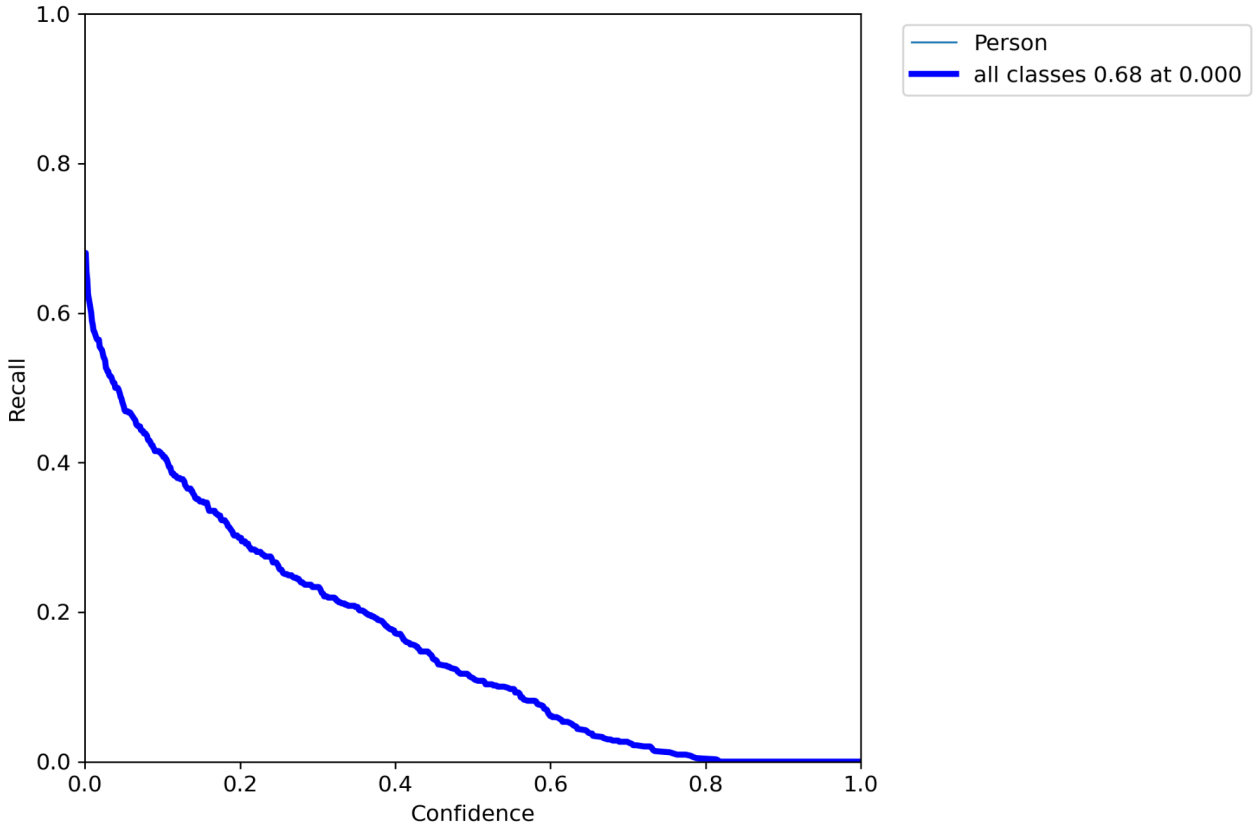
Learning Rate	0.01
No. of Epochs	200
Batch_Size	16
Optimizer	SGD (+Momentum)
Momentum	0.937
Weight Decay	0.0005

Plots:


0.29
1.00
0.71
Person
background FN
Predicted
Person
background FP
True
0.9
0.8
0.7
0.6
0.5
0.4
0.3


Person
all classes 1.00 at 0.775
Precision
Confidence
0.0
0.2
0.4
0.6
0.8
1.0


Person 0.240
all classes 0.240 mAP@0.5
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Person
all classes 0.68 at 0.000
Recall
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Outputs:

Conclusion:

Very good performance for the classifier. Objection Detection's performance is performing poorly. Image size (4000,3000) was a huge problem. Dividing the image into blocks of 4-5 and implementing RetinaNET can be a future scope

References:

1) Dunja Božić-Štulić, Željko Marušić, Sven Gotovac: Deep Learning Approach on Aerial Imagery in Supporting Land Search and Rescue Missions, International Journal of Computer Vision, 2019. - Link
2) Official Implementation of YOLO v5 - Link
3) Kundid Vasić, M.; Papić, V. Multimodel Deep Learning for Person Detection in Aerial Images. Electronics 2020, 9, 1459. - Link