VEHICLE DETECTION AND LOCALIZATION IN DHAKA ROAD IMAGES

Team : one_man_army

Abid Ahsan Samin (Islamic University of Technology)

Dhaka AI 2020

INTRODUCTION :

The capital city of Dhaka has only **7%** traffic roads (compared to **25%** urban standard) in presence of approximately **8 million** commuters a day within **306 sq km** area. The scenario of Dhaka traffic is unique which poses complex new challenges in terms of automated traffic detection. To solve the problem, I'm using **Deep Convolutional neural networks** out of any other existing methods

The problem statement says,We need to detect and localize **21 classes** of vehicles in challenging scenarios . To solve it we had to consider :

- Lighting conditions
- Occluded objects
- Dataset problems
- Confusing classes
- Class Imbalance issue

Deep learning is the best option to solve this challenge .

OTHER APPROACHES :

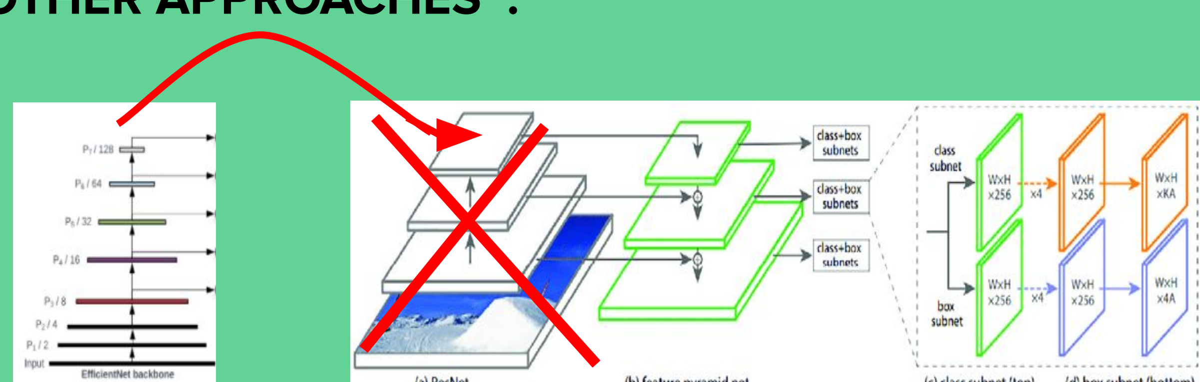


Figure: Retina Net with Efficient-Net Backbone feature extractor

Retina net uses special loss function **Focal loss** built in its network , other Sota(state of the Art) Models also try to add this loss function to handle **class imbalance problem**

Retina net is well designed to perform better on

- **Occluded**
- **Dense objects**

in images/videos . That's why I selected it at the beginning.

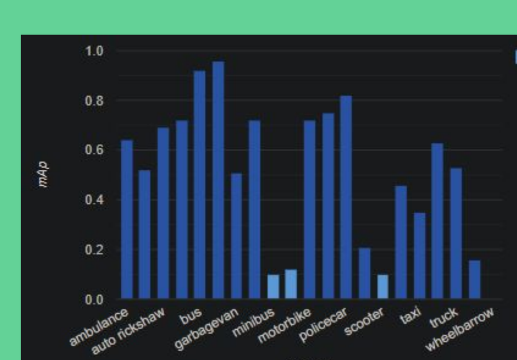


Figure: Class wise mAp score

From the bar chart we can see model has a quite a good score detecting objects with higher number of examples .

I got score of **0.142** in 1st round without TTA or ensembling .

I was using it in the 1st round and as i couldn't implement TTA and ensembling ,I switched to **YOLOV5** .

MAIN APPROACH :

As manually coding TTA (test time augmentation) and Ensembling is quite hard and no Convolutional neural net model uses built in **TTA , Augmentation , Ensembling** other than **Yolov5** or at least I couldn't find.

The YOLO network consists of three main pieces.

1) Backbone - A convolutional neural network that aggregates and forms image features at different granularities.

2) Neck - A series of layers to mix and combine image features to pass them forward to prediction.

3) Head - Consumes features from the neck and takes box and class prediction steps.

YOLOV5

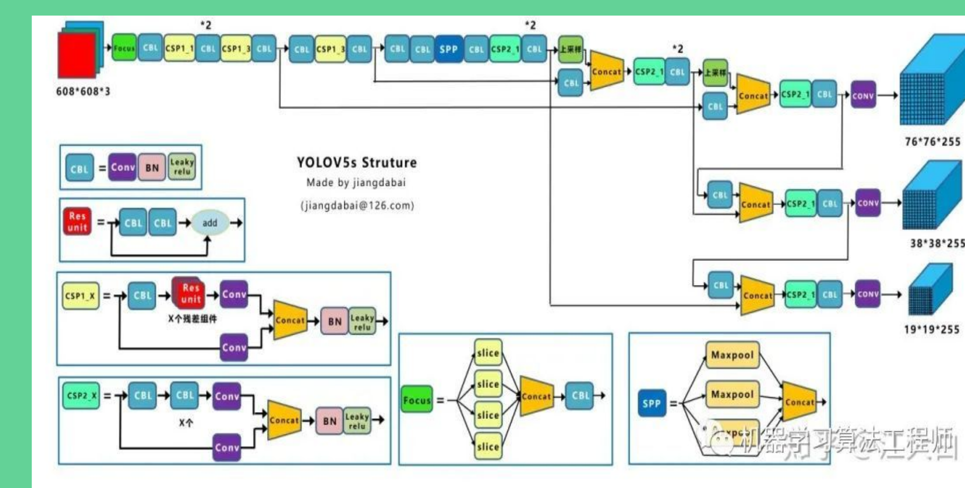
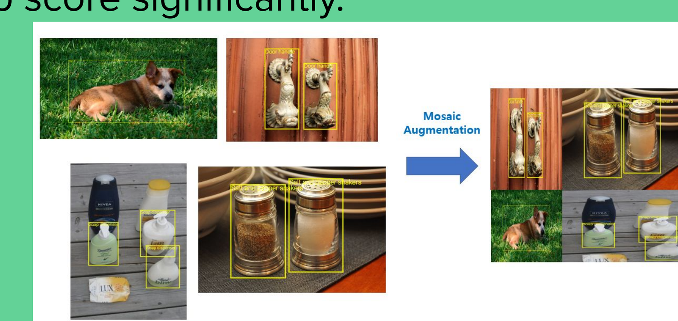


Figure: YOLO V5 Architecture

MOSAIC AUGMENTATION: A special augmentation technique introduced by author which improves mAp score significantly.



TECHNIQUES AND FINDINGS :

- **Added Cutout augmentation** :

  It's a **Regularization** and a **Advanced augmentation** technique that helps model **not to get biased** towards some specific features of a sample object rather helps to pay attention to every part of the sample . for example : Ambulance and Van has exactly similar features . only distinguishable feature is Ambulances **siren**. Cutout helps to focus on that as well

- **Model Size and Accuracy trade off**:

  Bigger model learns better features but when the dataset is small it easily overfits , so I chose **yolov5m** model considering the trade off .

- **Day model and Night Model**:

  Trained 3 day and 1 night model separately and **ensembled** them considering the **lighting condition** of test image .

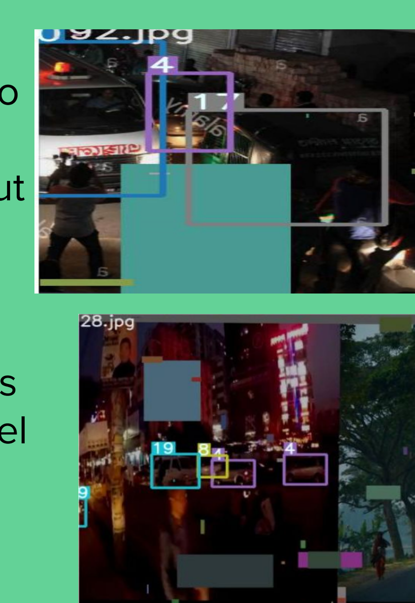


Figure: Cutout Augmentation

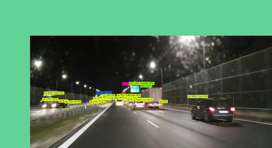


Figure: Night model prediction

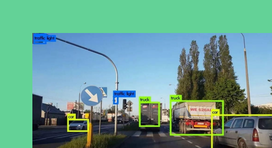


Figure: Day model prediction

- **Unannotated objects :**

  Found almost **500+** unannotated objects .

- **Mislabeled objects :**
  - Wheelbarrow as Rickshaw
  - Suvs as van/minivan
- **Inconsistent labels:**
  - Probox car
  - Noah
  - Van/Minivan/Suv
- **Label smoothing :**

  It deals with human annotation error , but it wont help On this highly incorrect dataset
- **Extra data**
- **Pseudo labeling :**

  It increases accuracy significantly as the data set is small . I only used 3-4 images and got quite high accuracy.

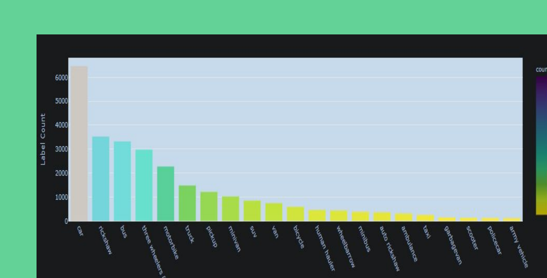


Figure: class imbalance

RESULTS :

After applying every techniques and fixing training data annotation i got a little bit boost in accuracy.

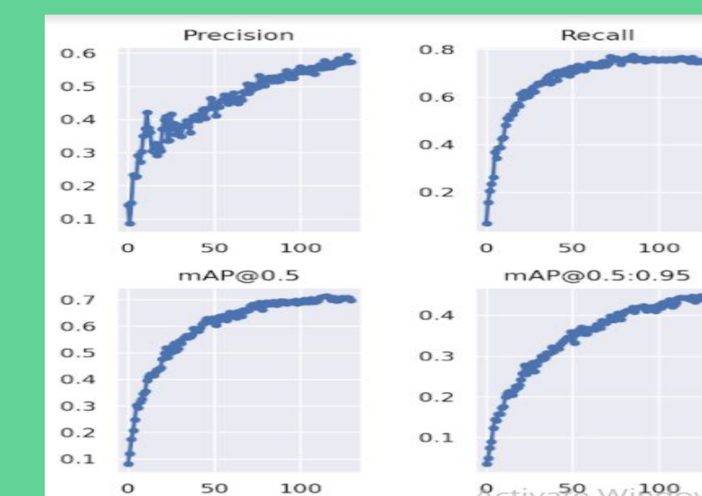


Figure: Precision Recall & mAp scores

Overall mAp was **.72+** which is significantly better than 1st rounds .63 .

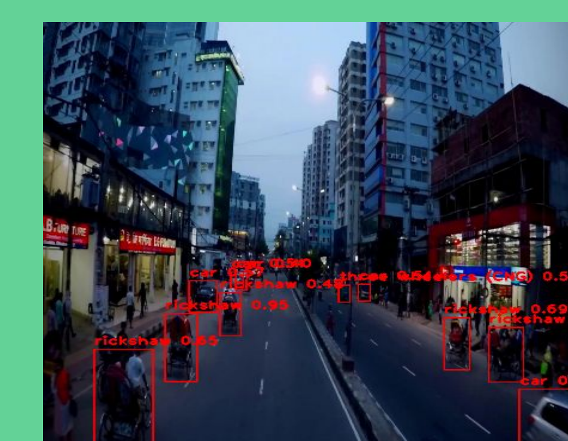


Figure: Good detection of night model

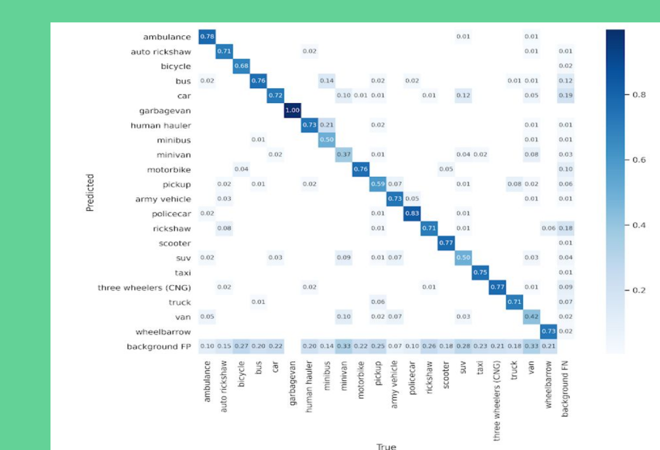


Figure: Confusion matrix

Conclusion :

As there are many state of the art models, it becomes a tough job to select a model. My observation is , we should select a model which is easily customizable and the user has deep understanding of internal building blocks . speed is also a big factor as to get better performance we need to ensemble several models which slows down inference speed. Many tradeoffs need to considered while working on a object detection task.

The dataset is too small and problematic which caused low accuracy in test set. I hope next time dataset would be more consistent and bigger

As 21 classes need at least 50k images . 3000 images can never be enough.