Report

This report concerns with the work that our team, i.e **TEAM-F**, has done until now on the given problem statement.

Team Members: Ujjwal Upadhyay, Rishabh Banga, Divyansh Jha

What is done until now?

Our solution is based on the modified retinanet based model. I decided to use retinanet as it's much simpler comparing to Faster-RCNN like models or SSD while having comparable results, this allows much easier experiments and debugging/tuning of model.

Credits to pytorch-retinanet implementation my solution is based on:

1. https://github.com/yhenon/pytorch-retinanet
2. https://github.com/fizyr/keras-retinanet

I scaled the original images to 512x512 resolution, with 256 resolution I have seen results degradation and using the full resolution was not as practical with heavier base models.

Modifications I have done to the original implementation:

- tested different base models, se-resnext101 worked the best, se-resnext50 slightly worse
- added an extra output for smaller anchors (level 2 pyramid layer) to handle smaller boxes
- added another classification output predicting the class of the whole image ('Person only', 'Truck only' or 'Both'). I have not used the output but even making the model to predict other related function improved the result.

- As classification outputs overfit much faster comparing to anchors position/size regression outputs, I added dropout to anchor and the whole image class outputs. In addition to extra regularisation, it helped to achieve the optimal classification and regression results around the same epoch.

Augmentation used:

Mild rotations (up to 6 deg), shift, scale, and h_flip, for some images random level of blur and noise and gamma changes. I limited the amount of brightness/gamma augmentations as it was hard for me to verify if it does not invalidate labels. To reduce the impact of rotation to bounding box sizes, instead of rotating the corners I rotated two points at each edge, at 1/3 and 2/3 edge length from the corner, 8 points in total and calculated the new bounding box as min/max of rotated points.

Currently, What's going on?

We are trying to distill our model so as to make it faster on CPU and it will, therefore, reduce inferencing time. We are making use of implementation provided at Intel Neural Network Distiller https://github.com/NervanaSystems/distiller and blogs on intel website like this https://software.intel.com/en-us/articles/knowledge-distillation-with-keras.

We are trying to convert the trained model to the IR model which can run on NCS. But the inference speed of the trained model is our priority.

Some of the results that we got from our model:


person 1.000
person 1.000
person 1.000
motorcycle 0.664
rider 1.000


truck 0.999
autorickshaw 0.995
truck 1.000
bus 1.000
car 1.000

Next Course of Action

We are looking forward to testing our model on i7 & i5 processors, Intel GPU 620 & 640 and myriad VPU i.e NCS-v1 (We dont have NCS-2).

Challenges and Concerns

The only challenge that we are facing until now is to convert our trained model to IR model since we have made use of custom layers in our model. If the problem persists we might have to shift to F-RCNN as its conversion is seamless in IR.