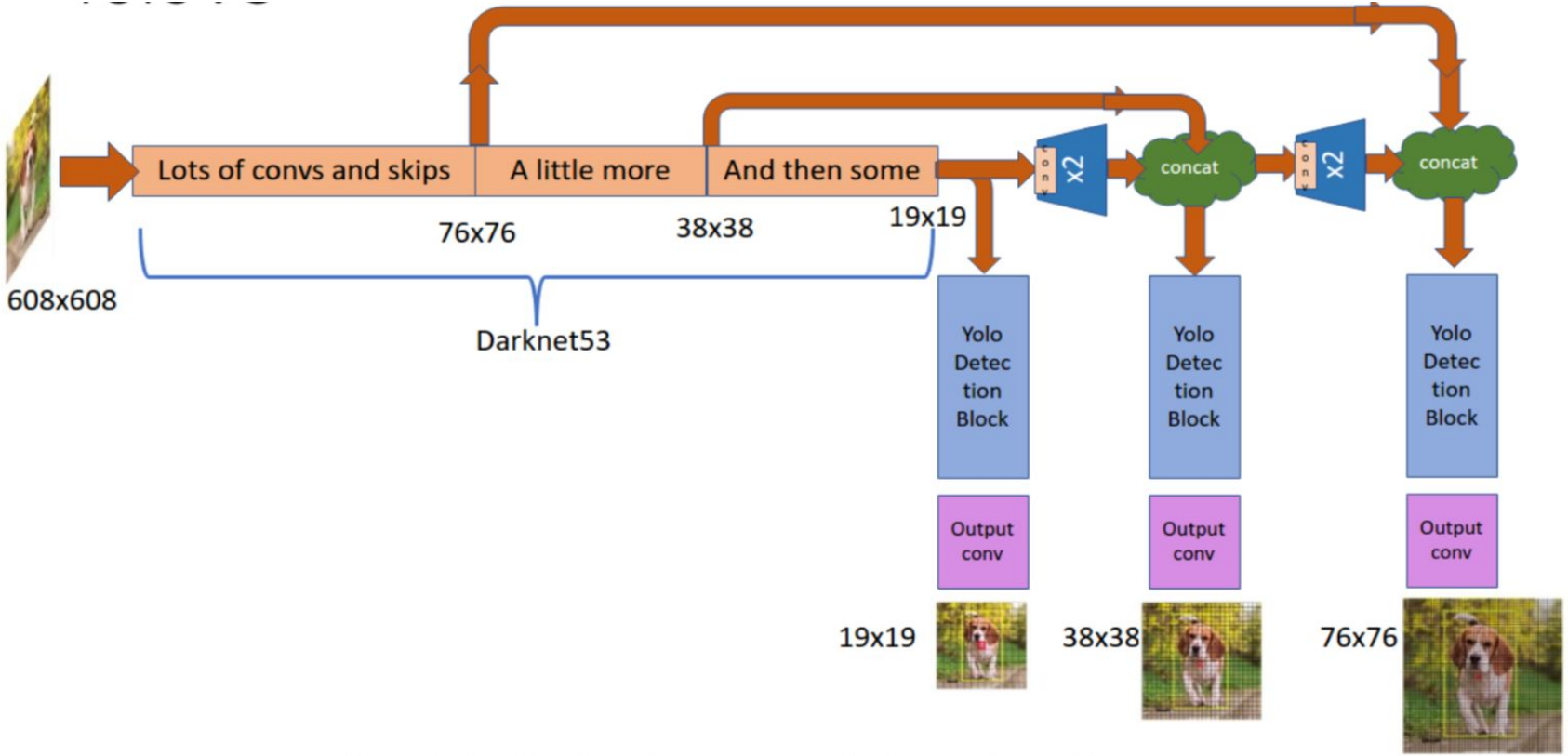
Real-time Courier service provider detection system using YOLO framework

Phase- 3

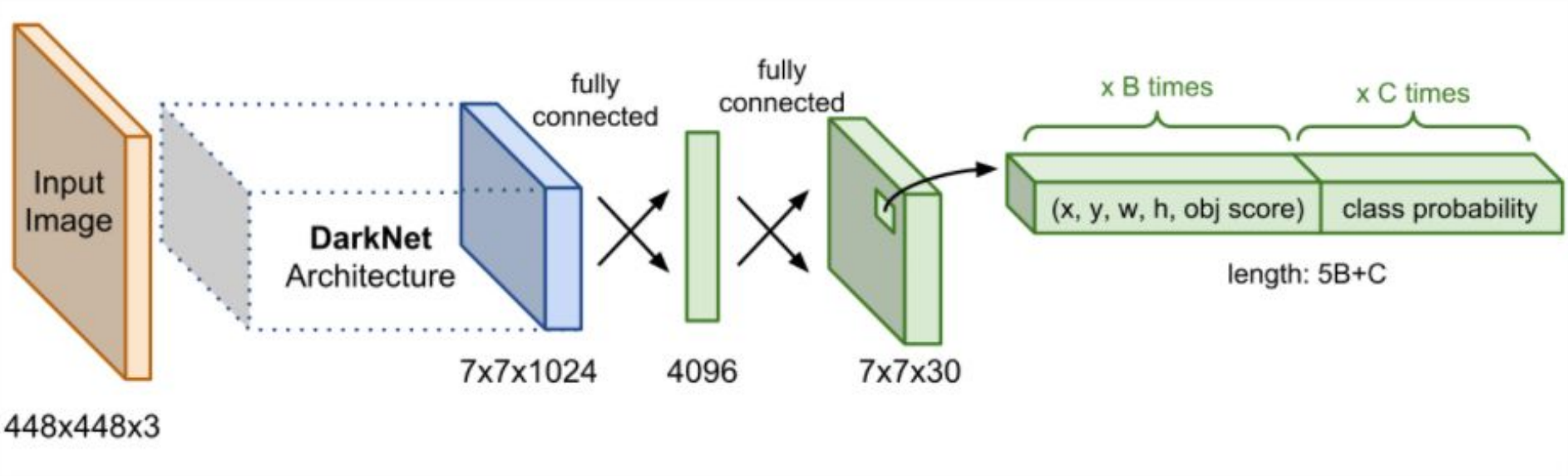
DATA 606 - Capstone Project

Shrimanth Ajjamane Manohar

YOLOv3 Architecture



YOLO model


Input Image
448x448x3
DarkNet Architecture
7x7x1024
fully connected
4096
fully connected
7x7x30
x B times
x C times
(x, y, w, h, obj score)
class probability
length: 5B+C

Training

I trained the datasets on YOLOv3, YOLOv4, YOLOv3 tiny, and YOLOv4 tiny models.

Each of the models was trained on 4000 epochs.

1. Yolo_V3 - Colab Notebook

   trained model weights file - drive.google.com/file/d/1FBFf_6thDNevuc9qzQS_-LU6mWWgjuln/view?usp=sharing

2. Yolo_V3 tiny - Colab Notebook

   trained model weights file - drive.google.com/file/d/1MfVpxBrd0GIAaF0Shf-zufVi-g-_8zEp/view?usp=sharing

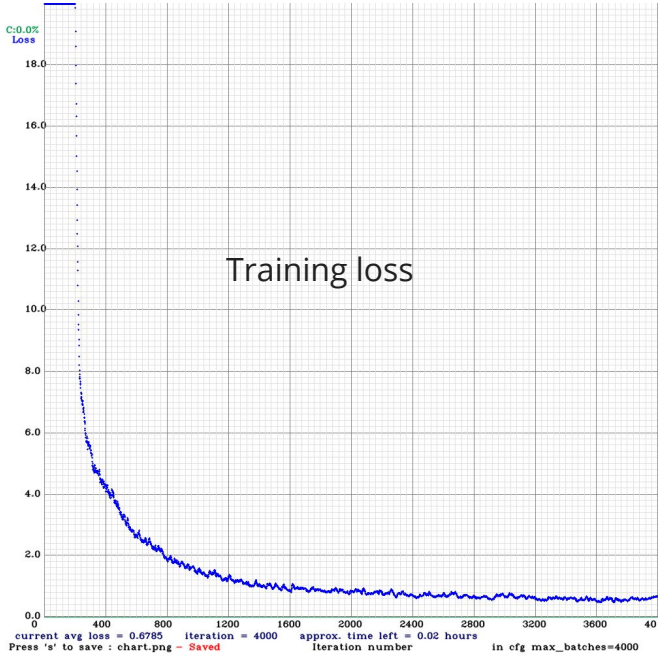
3. Yolo_V4 - Colab Notebook

   trained model weights file - drive.google.com/file/d/16bQwz1DsTbDlfdJMsWnGwGdnkyyWmdB1/view?usp=sharing

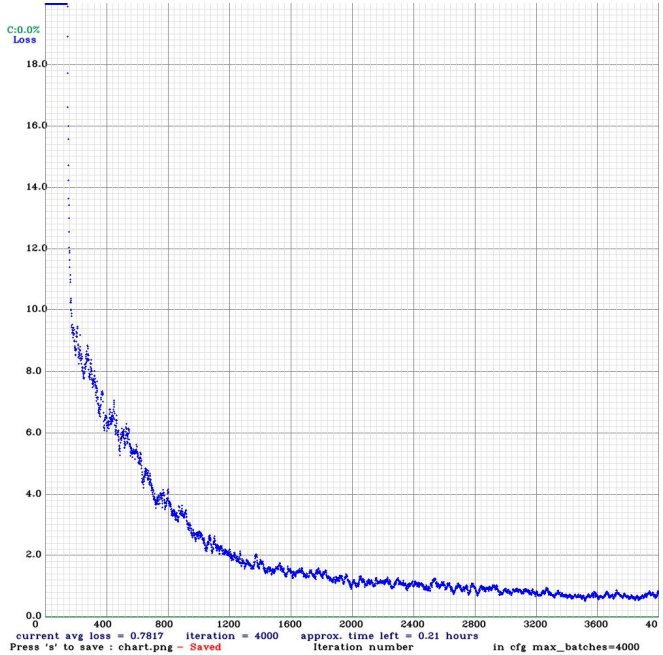
4. Yolo_v4 tiny - Colab Notebook
   trained model weights file - drive.google.com/file/d/1ArUkn6WvakV_7mdRzVjCw6FAlv9I1-Jw/view?usp=sharing

Results

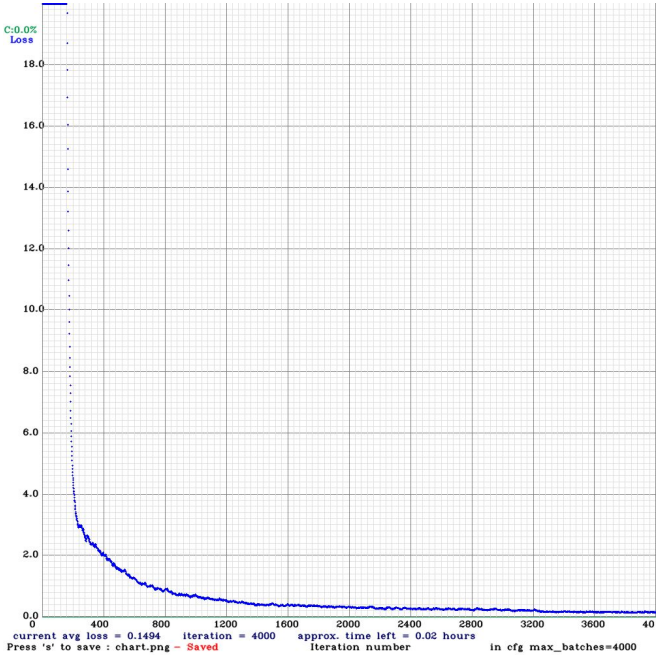
YOLOv3 training loss:



YOLOv4:



YOLOv4 tiny:



Results on video clips:

YOLOv3:

Results on video clips:

YOLOv4:

Conclusion and Future work:

Object detection is the core component for most computer vision systems, the current advancements have been used in many real time applications.

As we know training the model is computer intensive, but attaining the inference on low compute devices not requires much of resources. Hence the future is towards predicting inference on edge devices which have limited or no access to cloud or high compute resources. Deploying the trained model's weights on edge devices and detecting the images captured on the go by running a forward pass along with the pre-trained weights will surely lead to many more applications.