REAL TIME SIMULTANEOUS OBJECT DETECTION

Problem Statement

- ★ The main objective of our project is the Real time detection of objects in a Household scenario.
- ★ Our application can detect objects using live video(using webcam) which includes locating the objects with a bounding box, detecting the object with a confidence score and tracing the object in a video.
- ★ The result of this project can be used in applications involving Home automation.

Methodology & Implementation Details

Architecture

- ★ Yolov3 algorithm is used as it predicts the objects fast at real time.
- ★ Resnet50 is used as the base model for our Neural Network with Detnet used in few of the Bottleneck layer.
- ★ Detnet layer has two 1x1 convolution blocks, one 3x3 dilated convolution block and a skip connection

Dataset & Training

- ★ COCO 2017 images Dataset is used for training and testing, customizing the labels to suit a household environment.
- ★ 78606 images are used for training and 3357 images for testing with 10 class labels.
- ★ All training images are annotated with its labels and bounding boxes.

Losses

- ★ Classification loss,bounding box coordinate loss, no object loss are added to compute total loss.
- ★ Mean squared error of targets and predictions is used as the loss function.
- ★ After loss computation backward propagation is handled by default pytorch function (loss.backward())

Evaluation Metrics

- ★ IoU(Intersection over Union) is computed to find the amount of area of intersection of target and predicted bounding boxes.
- ★ Precision and Recall are calculated with help of TP & FP(obtained based IoU threshold values)
- ★ mAP(mean Average Precision) is calculated by averaging all the Average precisions obtained for individual class labels.

Real Time Integration

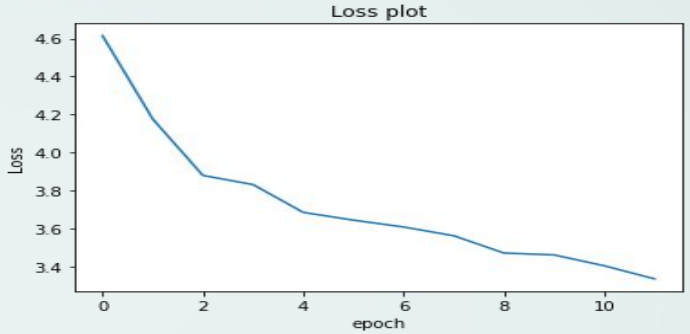
- ★ The trained model is saved and loaded to make predictions at real time.
- ★ Flask framework is used to stream the video in browser, which enables the webcam.
- ★ Each frame of the video is sent to the model which predicts the objects of that frame using predict function.

Experimental Details

- ★ Dataset is trained by changing different hyper parameters and losses are observed.
- ★ Entire dataset training carried out for 12 epochs, which takes 2 hrs time for one epoch.
- ★ The final hyper parameters used are :
  - Learning rate (α) = 0.001
  - IoU threshold = 0.5
  - Non maximum suppression threshold = 0.3
  - We used B = 2, S = 14
- ★ Model will be saved for each epoch and best detector path can be traced back.
- ★ The final trained model is evaluated with the test samples(3357 Images).

Results and Observations

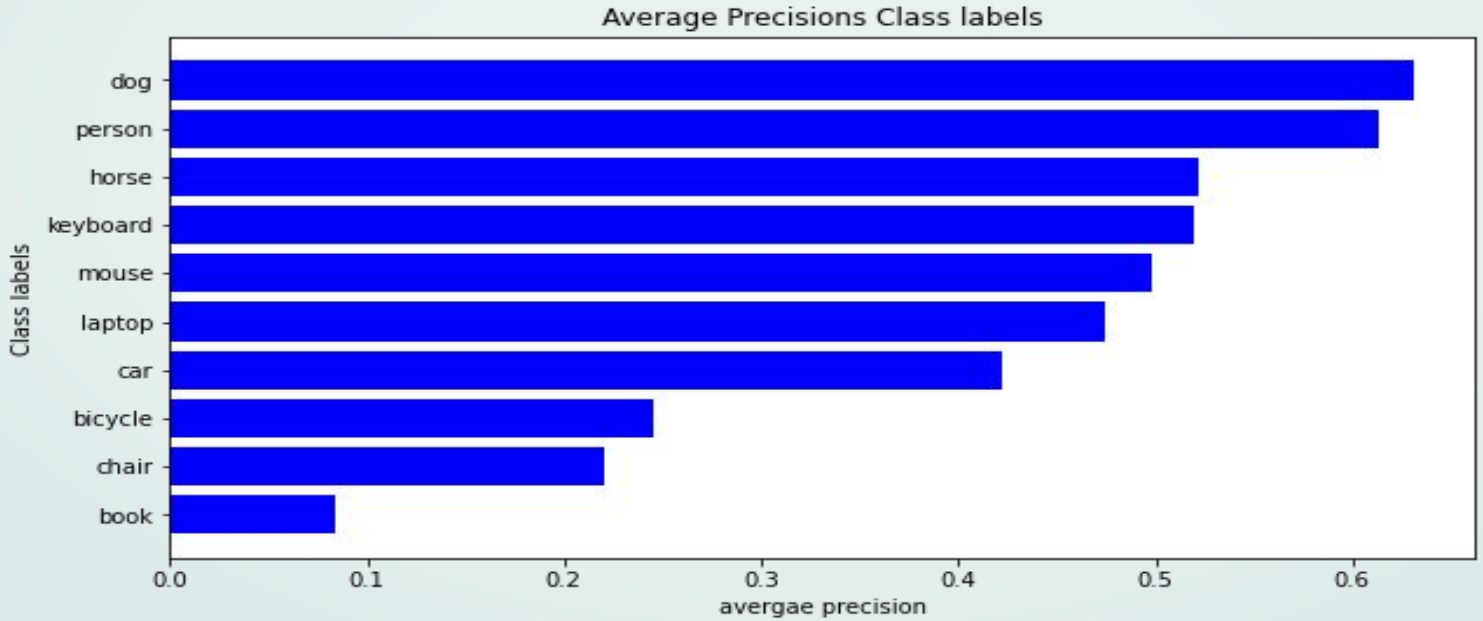
- ★ Overall Loss(Regression and Classification) is computed for 12 epochs, it is decreasing accordingly.
- ★ The minimum loss obtained is for 12th epoch i.e 3.33560072524207



- ★ Class scores and average precisions for each class label are significant for our parameters.
- ★ The final mAP is 0.42 and most the true detections is obtained for dog class.

```
---Evaluate model on test samples---
100%|██████████| 3357/3357 [01:38<00:00, 33.98it/s]
---class person ap 0.6133291528940201---
---class bicycle ap 0.2459286332226221---
---class car ap 0.4227326767676697---
---class dog ap 0.630862406067567---
---class horse ap 0.5223068134709519---
---class chair ap 0.22081246939048---
---class laptop ap 0.47451647912123934---
---class mouse ap 0.4980084319530861---
---class keyboard ap 0.5198584606941563---
---class book ap 0.08428747982999552---
---map 0.4232643003411788---
```

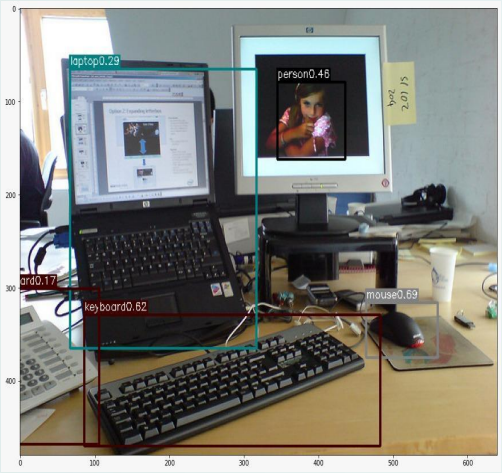
Graph to visualize Average precisions

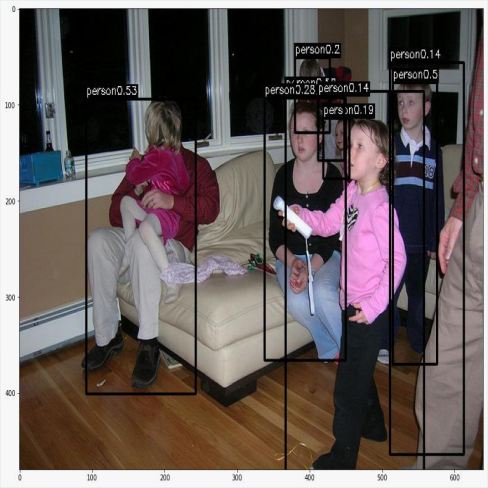


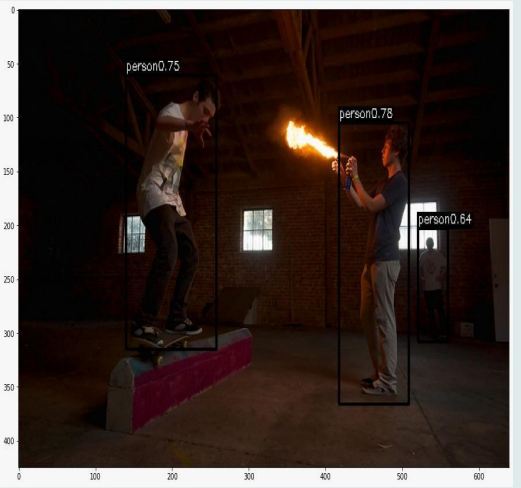
Realtime DEMO

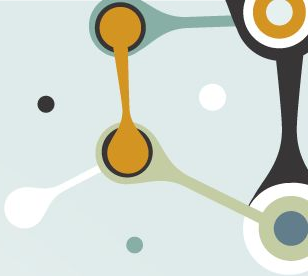
Video Streaming Demonstration

Some detections on Test set







THANK YOU