EE5179: Deep Learning for Imaging

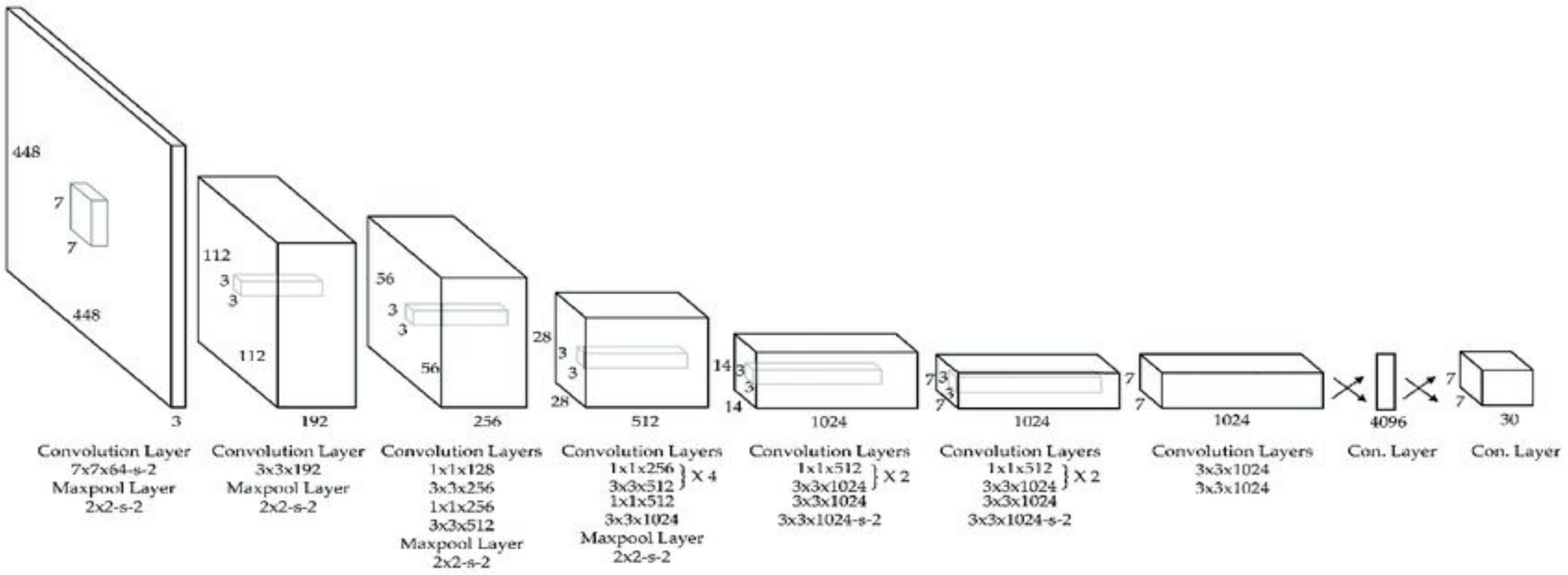
Low Light Object Detection

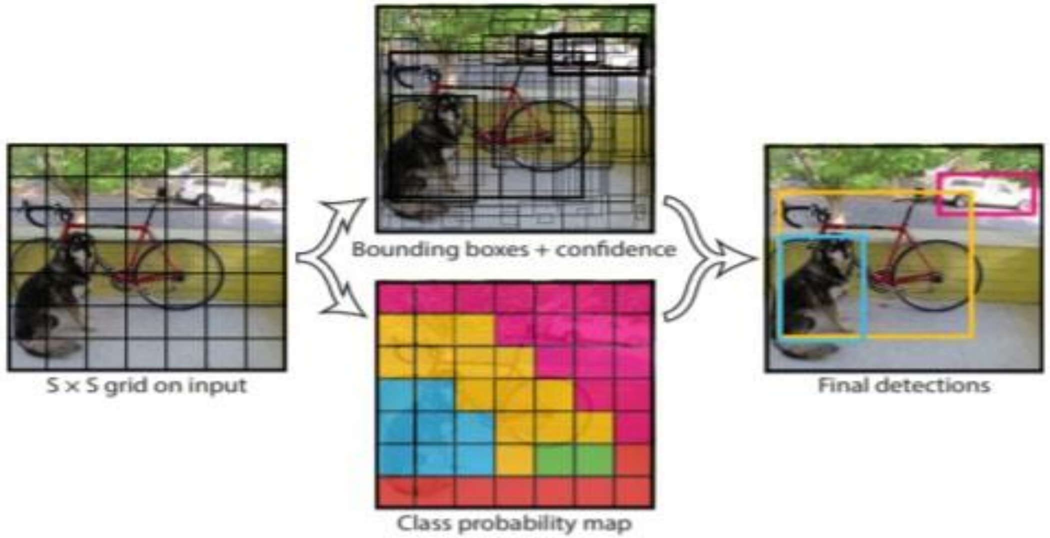
S Siva (EE22S044)

E M Sandeep Murthy (EE22M013)

Ashish Srivastava (ME21B036)

YOLO : You Only Look Once

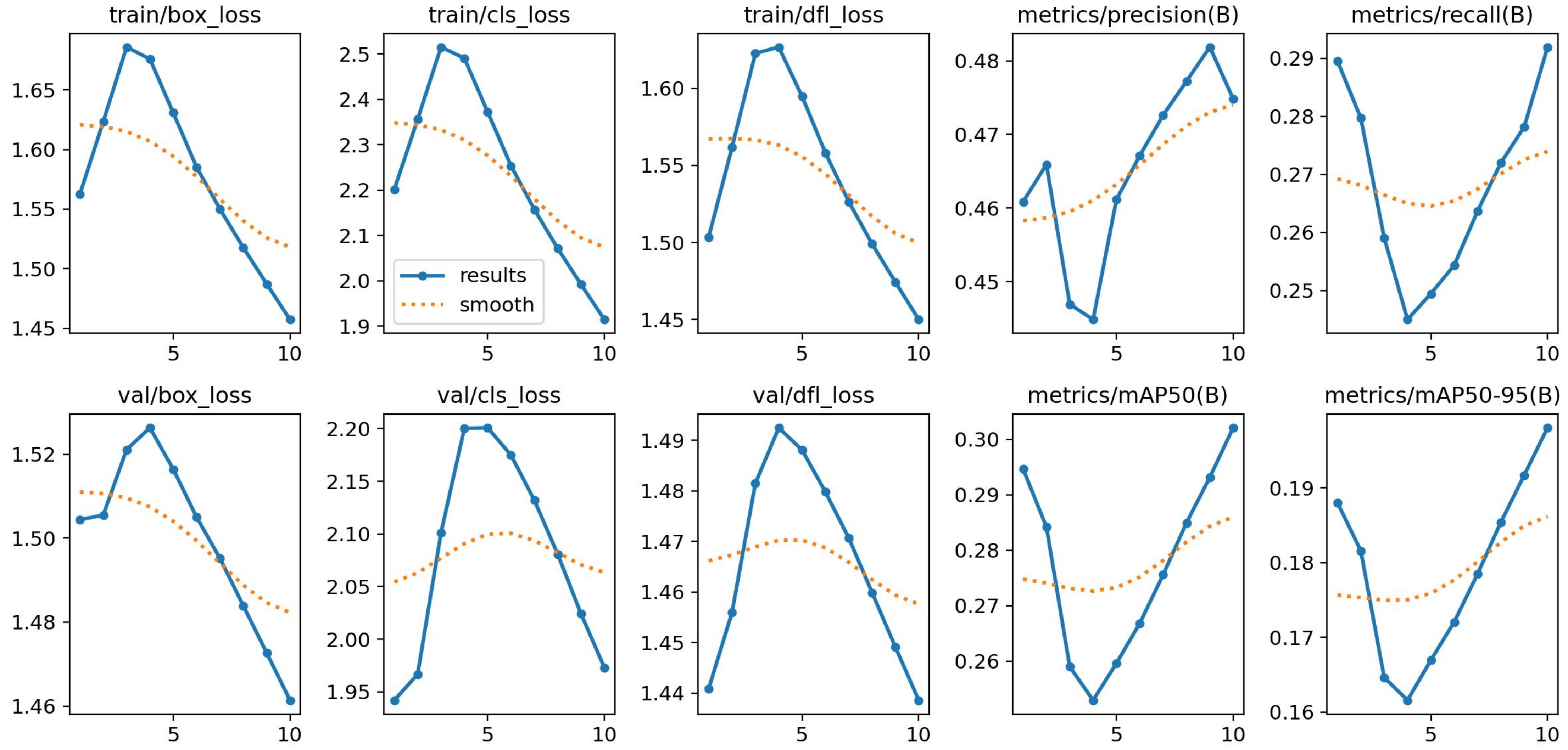




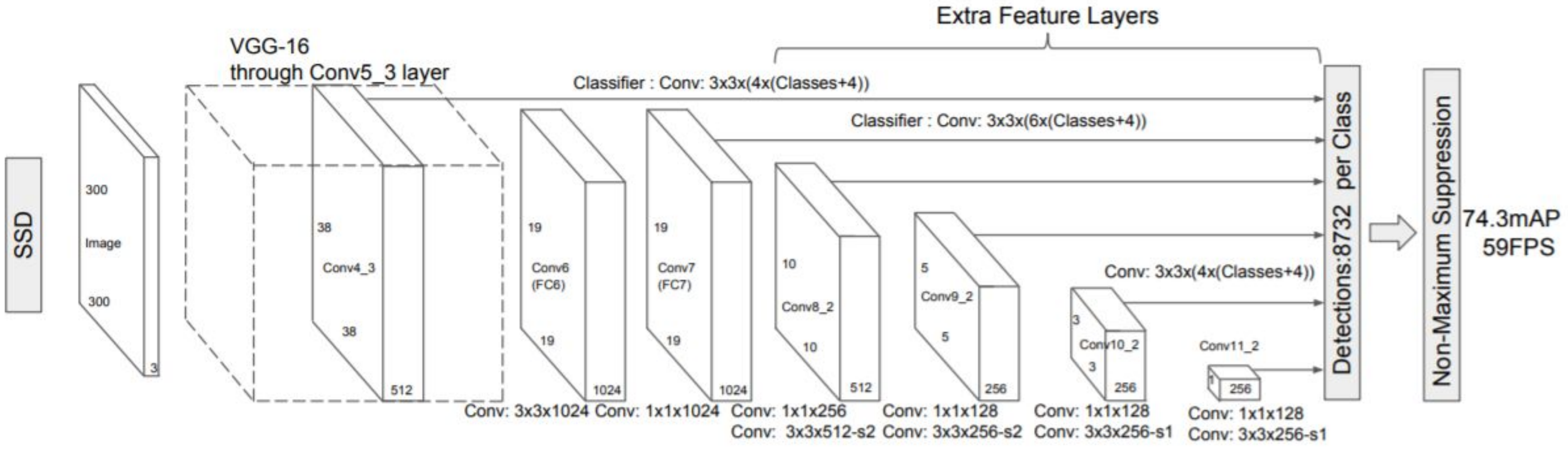
MODEL :

- **Grid-based Detection:** YOLO divides the input image into a grid, typically, S x S. Each cell in the grid is responsible for predicting bounding boxes and class probabilities.
- **Bounding Box Prediction:** Each grid cell predicts multiple bounding boxes along with confidence scores. These bounding boxes represent potential locations of objects within the cell.
- **Class Prediction:** Alongside each bounding box, YOLO predicts the probability distribution over the classes of objects. This is done for each bounding box independently.
- **Single Forward Pass:** YOLO performs object detection in a single forward pass through the neural network, which makes it highly efficient and suitable for real-time applications.
- **Non-maximum Suppression:** After the initial predictions, YOLO applies non-maximum suppression to filter out duplicate detections and retain only the most confident ones. This helps in eliminating redundant bounding boxes for the same object.

YOLO Results :

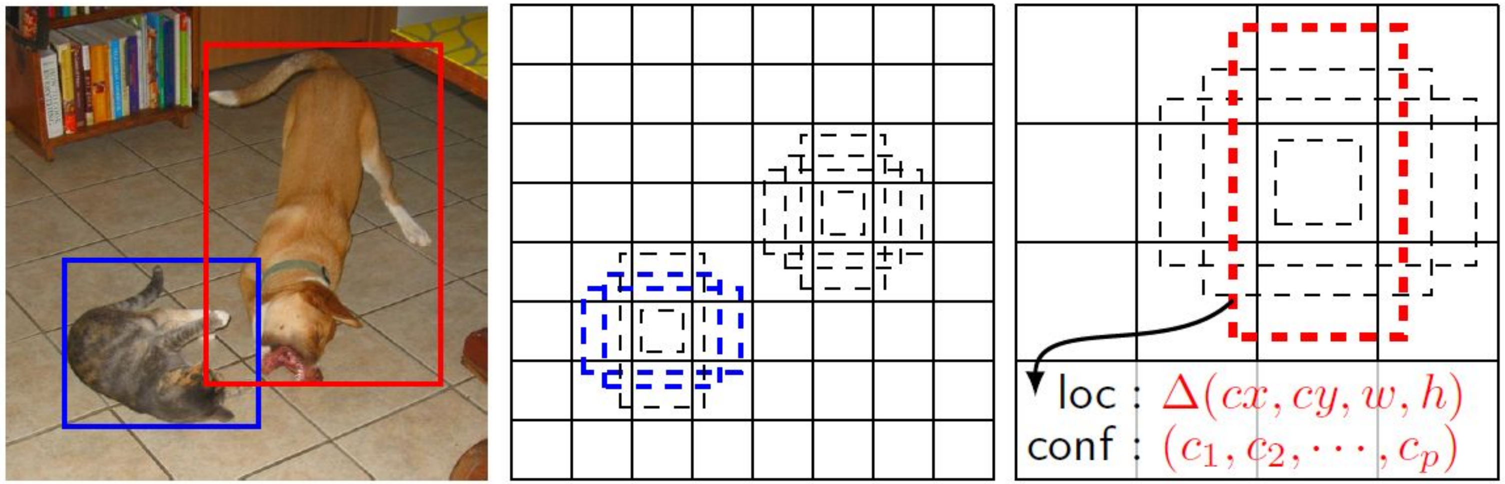


SSD : Single shot multi-box detector



MODEL :

- It is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections.

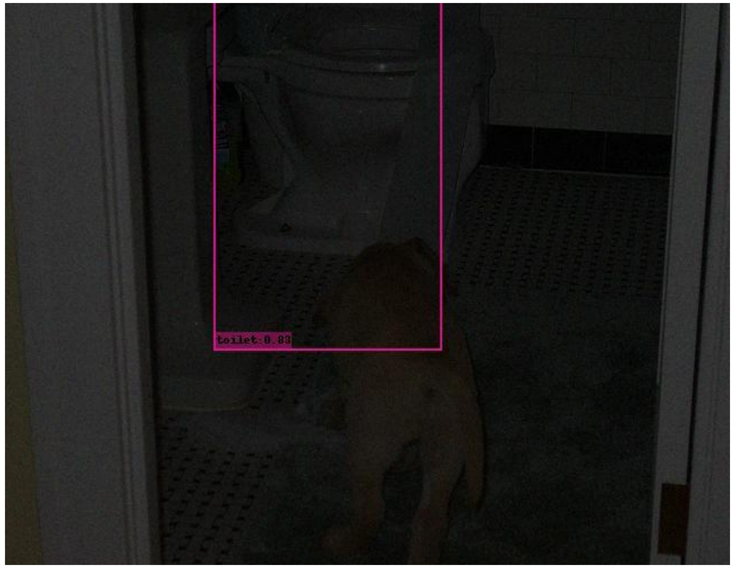


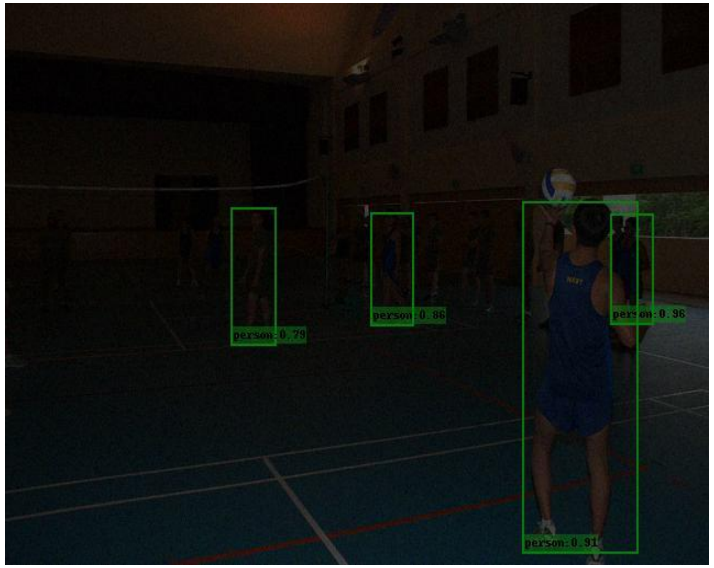
(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

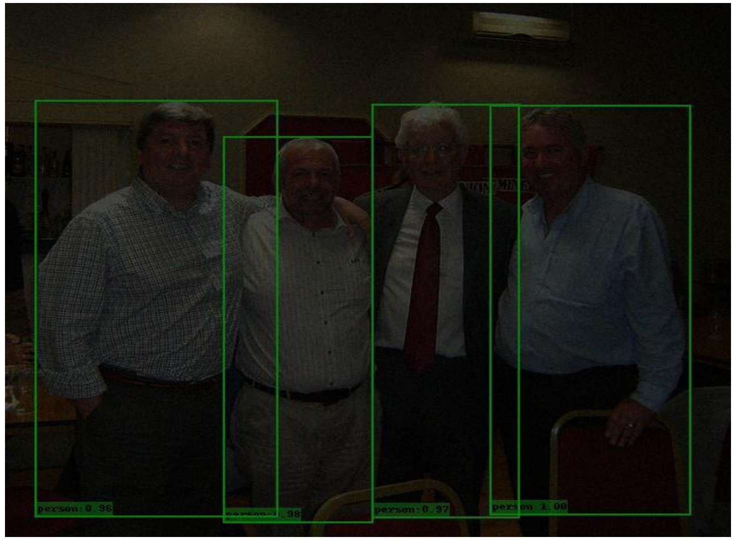
- We add convolutional feature layers to the end of the truncated base network(VGG16). These layers decrease in size progressively and allow predictions of detections at multiple scales.
- The convolutional model for predicting detections is different for each feature layer.

SSD Results :

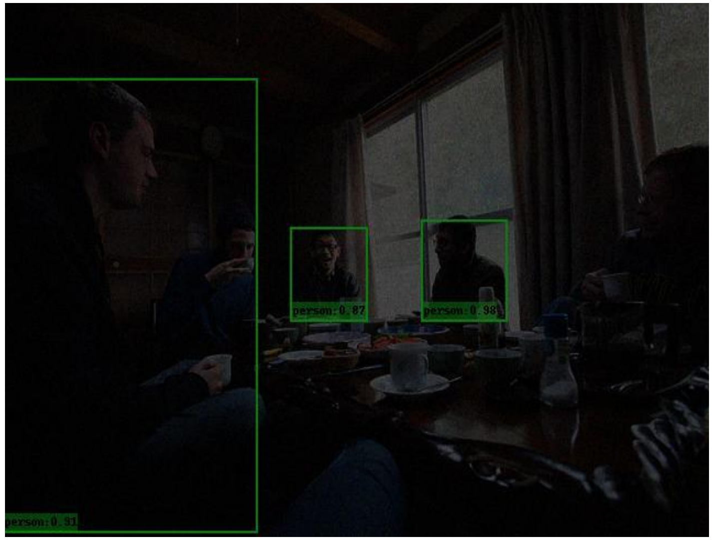

person:0.99


toilet:0.83

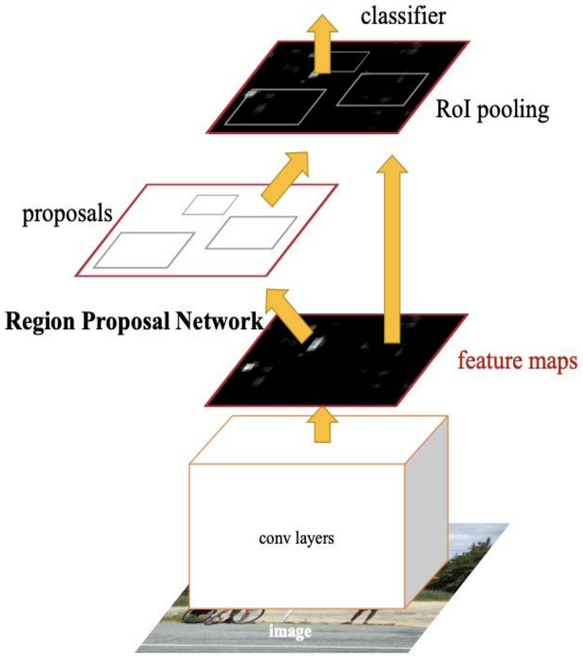

person:0.79
person:0.86
person:0.96
person:0.91

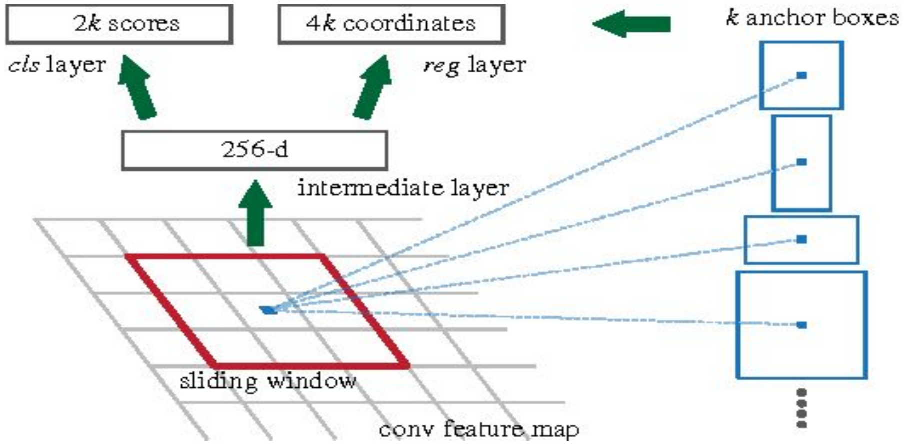

person:0.96
person:0.97
person:1.00

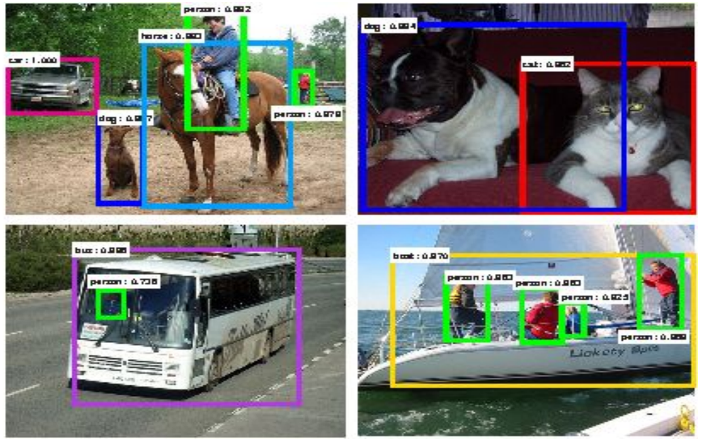

cat:0.90


person:0.87
person:0.98
person:0.91

Faster R-CNN :







MODEL :

-> Region Proposal Network (RPN): It is responsible for proposing potential regions in the image that may contain objects.

-> Anchor Boxes: These are pre-defined bounding boxes of various scales and aspect ratios, that are used to propose regions.

-> Region of Interest (RoI) Pooling: This is used to extract fixed-size feature vectors from each proposed region.

-> Classifier and Regressor: The classification branch determines the object class, while the regression branch refines the bounding box coordinates.

Comparison :

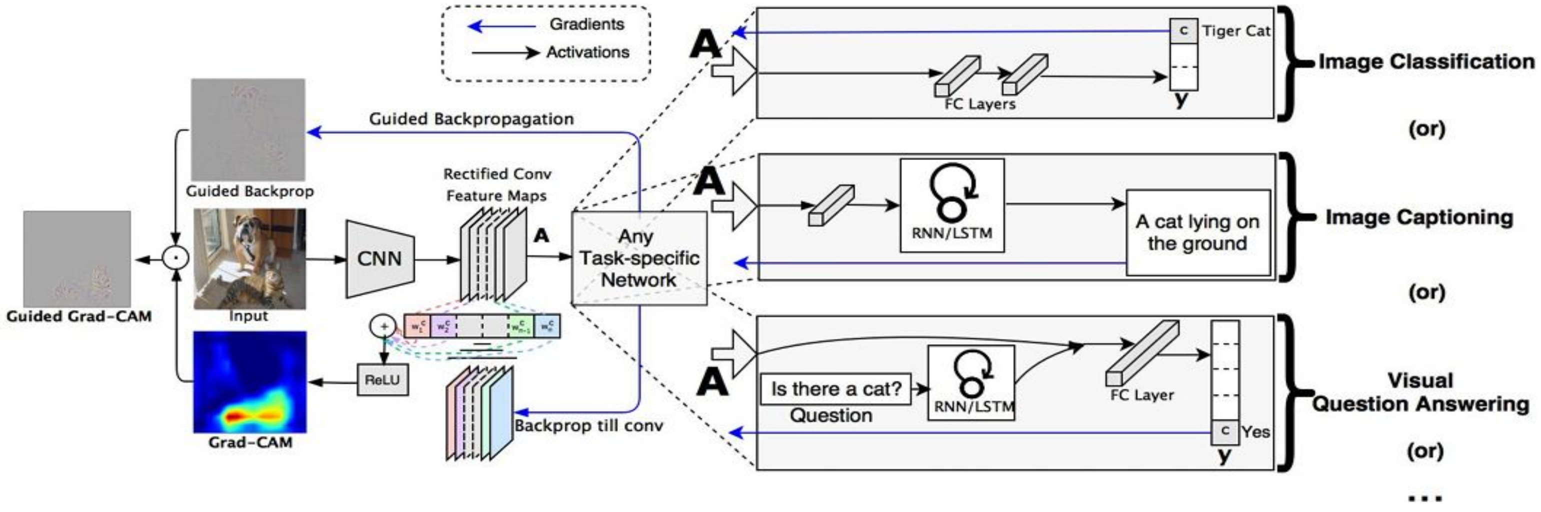
Metrics / Model	mAP50	mAP50-95	Frame Rate
YOLO	0.33791	0.24214	~ 160 FPS
Faster R-CNN			~ 10 FPS
SSD	0.324	0.215	~ 50 FPS

- YOLO on being trained with further dimmed images, gave the following results.
  * **mAP50** – 0.29872
  * **mAP50-95** – 0.20306
- Accuracy reduced along with brightness.

Contributions

- YOLO – Siva
- SSD – Sandeep
- Faster R-CNN - Ashish

Grad CAM Paper :



$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

- In grad cam, we pass the image and compute the necessary score. We then set the gradients for all the other scores to zero and perform backpropagation till just before the CNN layers.
- We then calculate the neuron importance function which is the global average pooled backprop of the score with respect to the activation maps where A is the activation map and y is the score of class c.
- Alpha is the neuron importance. We then do a weighted of all the activation maps with the corresponding importance function and perform a ReLU operation to get back the same size heatmap as the last convolutional feature map.