Object Detection

CV Project - Team “Kuch bhi”

Team Members:
2018101033 - Jay Sharma
2018102021 - Tanmay Garg
2018102040 - Shantanu Agrawal
2020900019 - Anirudh Polatpally

RCNN - Method - Overview

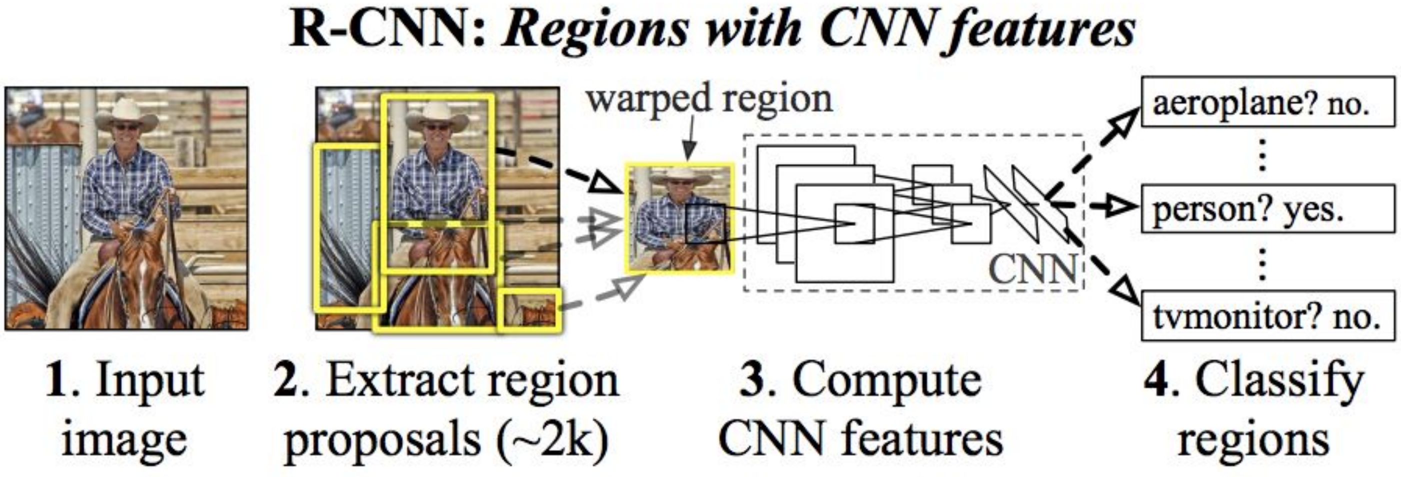


Fig 1: RCNN workflow
Source: [1]

Region Proposals

- Used Selective Search Algorithm
- Foreground: IoU greater than 0.8
- Background: IoU lesser than 0.3
- Proposal dimensions - 224, 224, 3

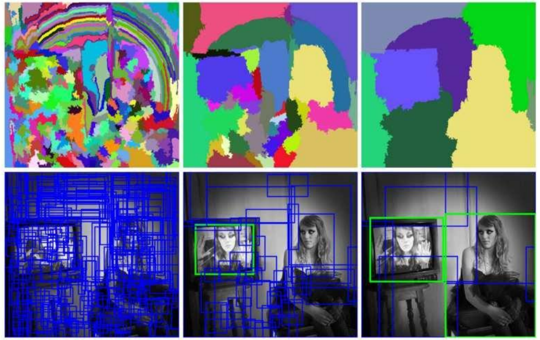


Fig 2: Selective Search at different scales
Source: [2]

Feature Extraction - CNN

- Used VGG-16 as backbone feature extractor.
- *3×3* convolutional layers stacked on top of each other
- Max-pooling to reduce volume in between
- Fully Connected layers at the end followed by softmax.
- Final layers changed to perform classification and localisation.

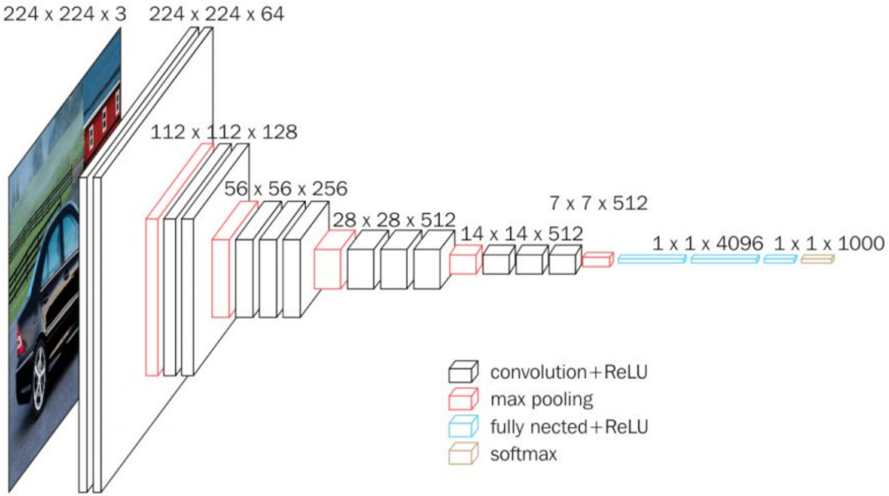


Fig 3: VGG16, Source: [3]

Drawbacks of R-CNN

- Extremely large amount of time
- 2 hours per epoch
- ~8 days for 200 epochs
- Prediction time very high (1-2 mins per image)

Fast R-CNN

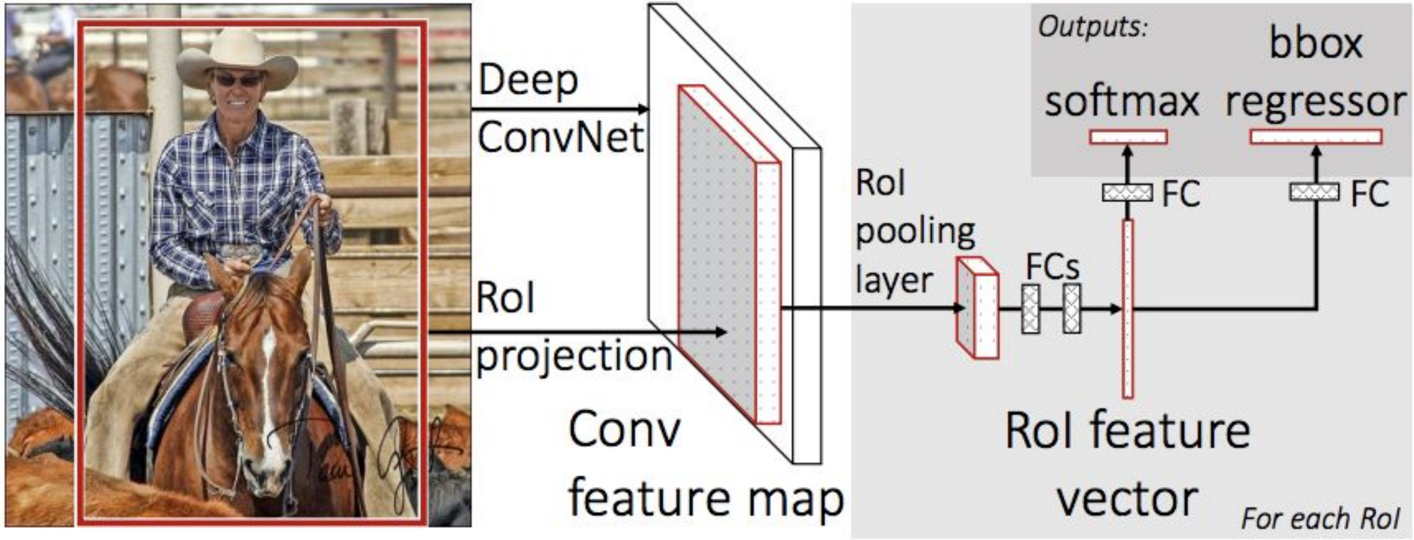


Fig 4: Fast R-CNN, Source: [4]

Fast R-CNN

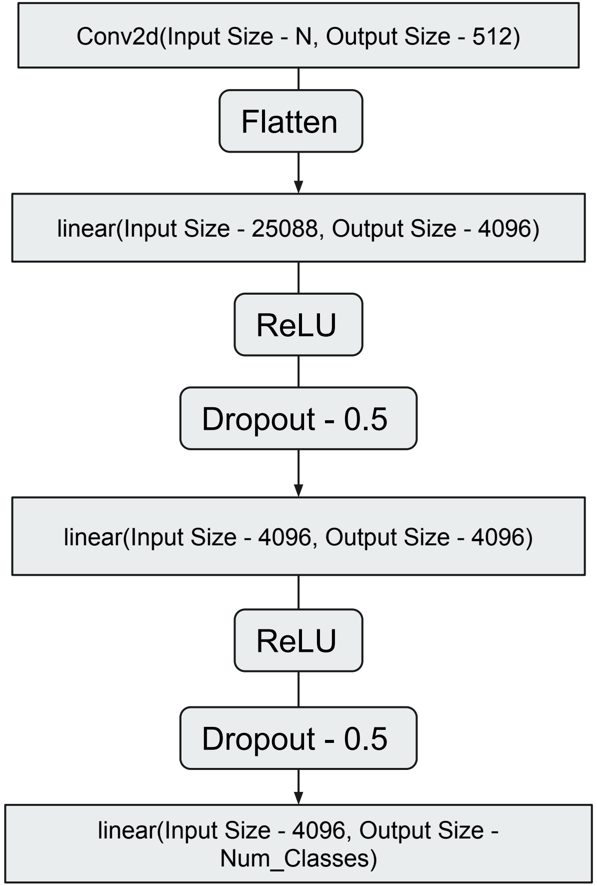
- Directly feed Input image to CNN
- Obtain Feature Map
- Identify and warp features
- Perform RoI Pooling for fixed shape
- Predict Class label using Softmax layer
- Predict Bounding Box using Bounding Box Regression

Improvement:

- Each epoch takes around 60 seconds now
- i.e. 3.33 hours per model.
- i.e. 20 hours for classification and regression for all 3 models (2*3 = 6 models).

Classification Head

- Perform Classification
- Final output - scores for each of the classes in consideration (21 in our case)



Conv2d(Input Size - N, Output Size - 512)

Flatten

linear(Input Size - 25088, Output Size - 4096)

ReLU

Dropout - 0.5

linear(Input Size - 4096, Output Size - 4096)

ReLU

Dropout - 0.5

linear(Input Size - 4096, Output Size - Num_Classes)

Fig 4: Classification Head

Regression Head

- Perform Bounding Box Regression
- Final output - Predicted Bounding Box (4 values)

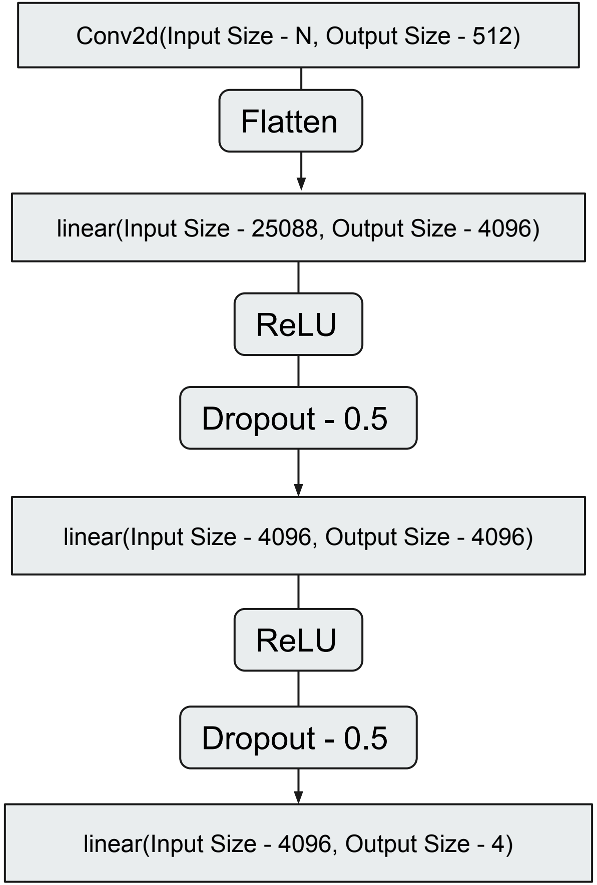

Conv2d(Input Size - N, Output Size - 512)
Flatten
linear(Input Size - 25088, Output Size - 4096)
ReLU
Dropout - 0.5
linear(Input Size - 4096, Output Size - 4096)
ReLU
Dropout - 0.5
linear(Input Size - 4096, Output Size - 4)

Fig 5: Regression Head

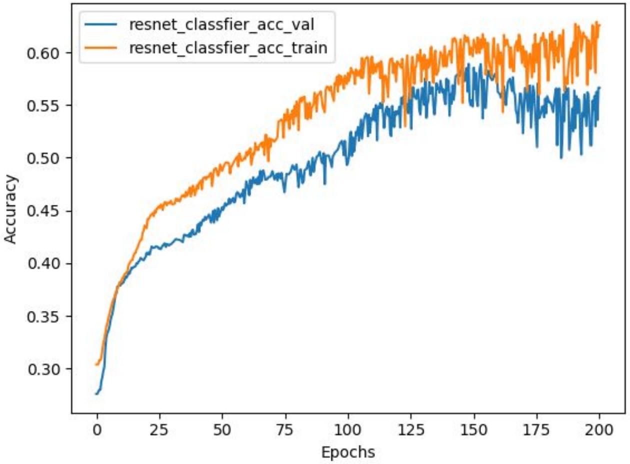
Non Maximum Suppression

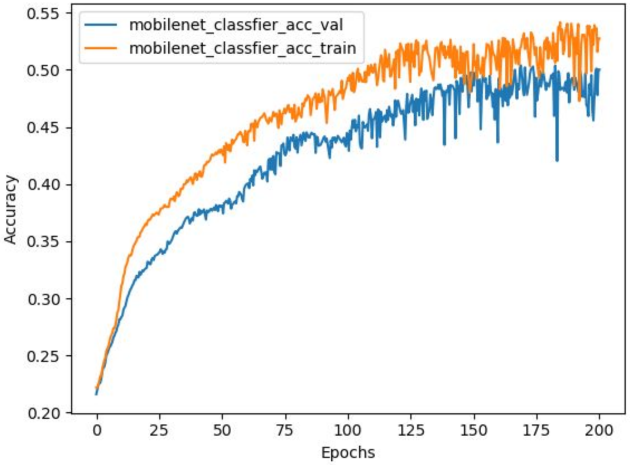
- Remove multiple bounding boxes
- Remove boxes with low confidence
- Select a box with highest confidence
- Removes lower scoring boxes which have an IoU greater than iou_threshold with the highest scoring box

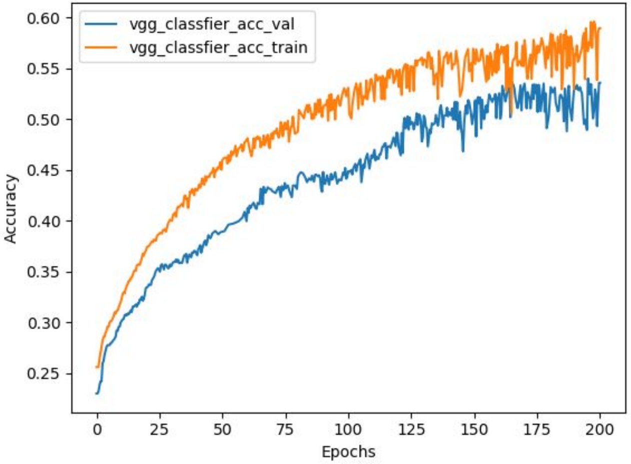
Training

- Obtained feature vectors for each image in training set
- Obtained proposals from these vectors using selective search
- Trained model using these proposals
- Initially trained classification head
- Subsequently trained regression head.

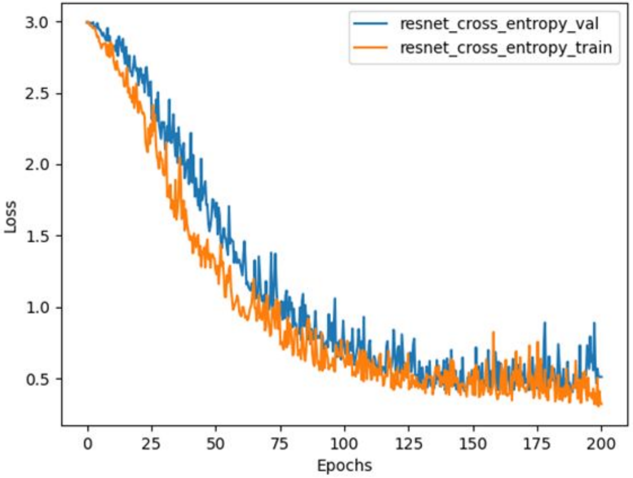
Classification Accuracy Curves

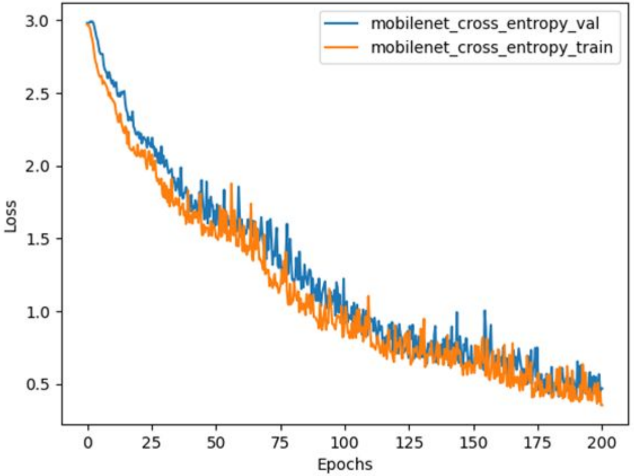


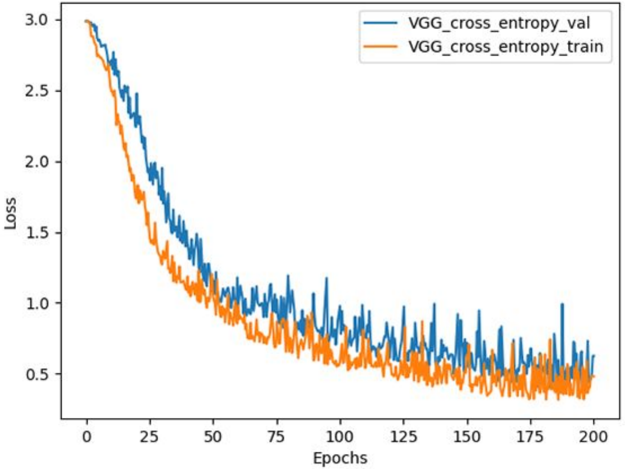




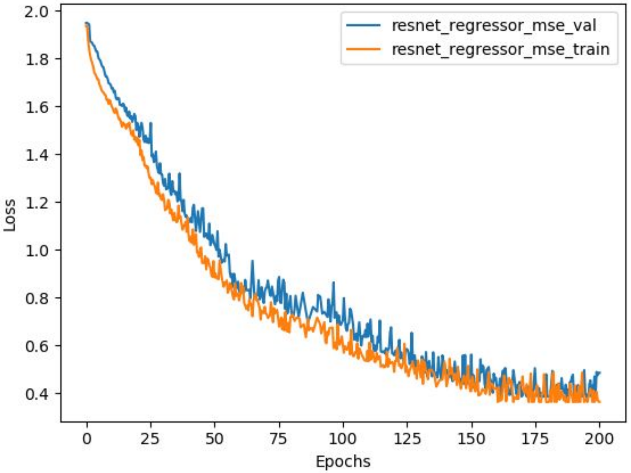
Classification Loss Curves

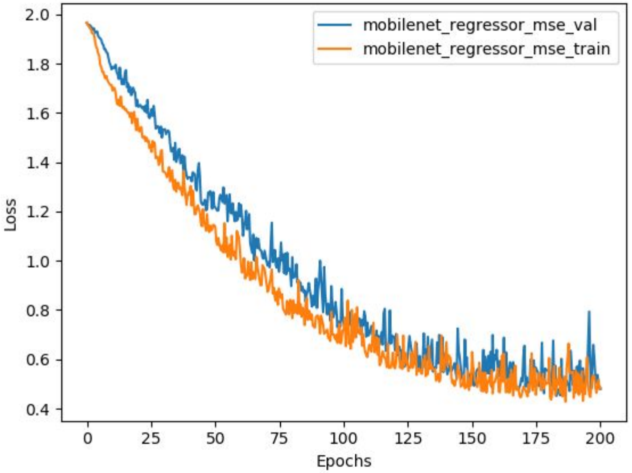


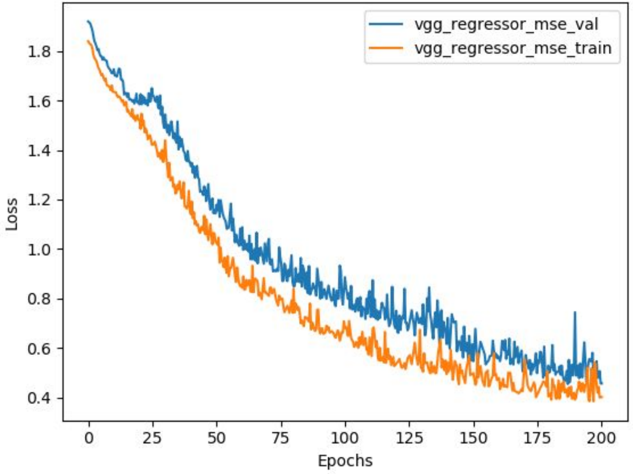




Regression Loss Curves



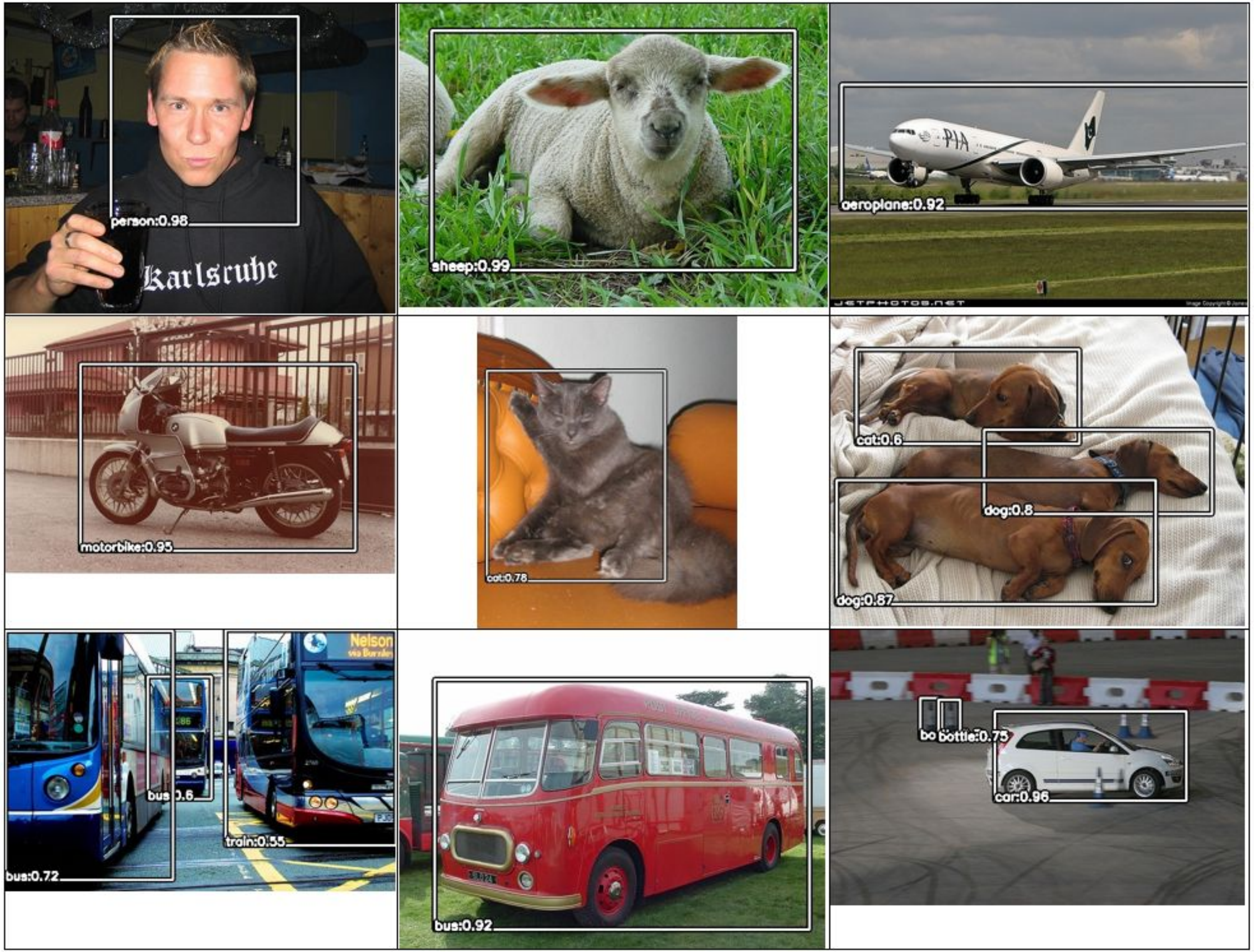




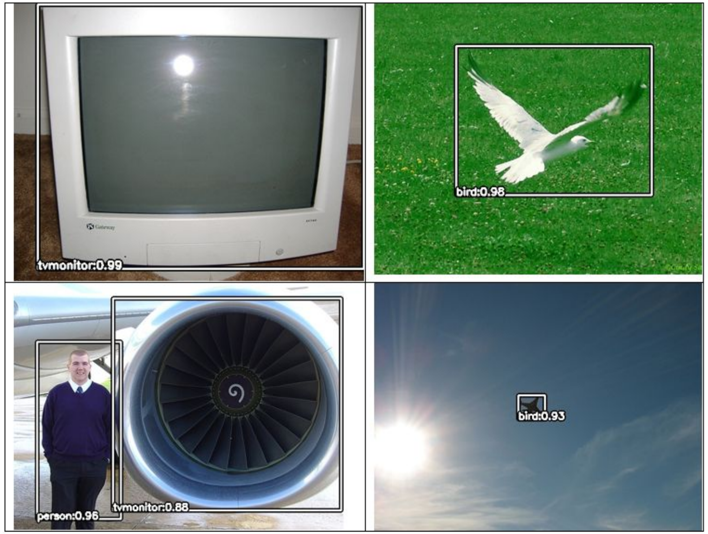
Metrics

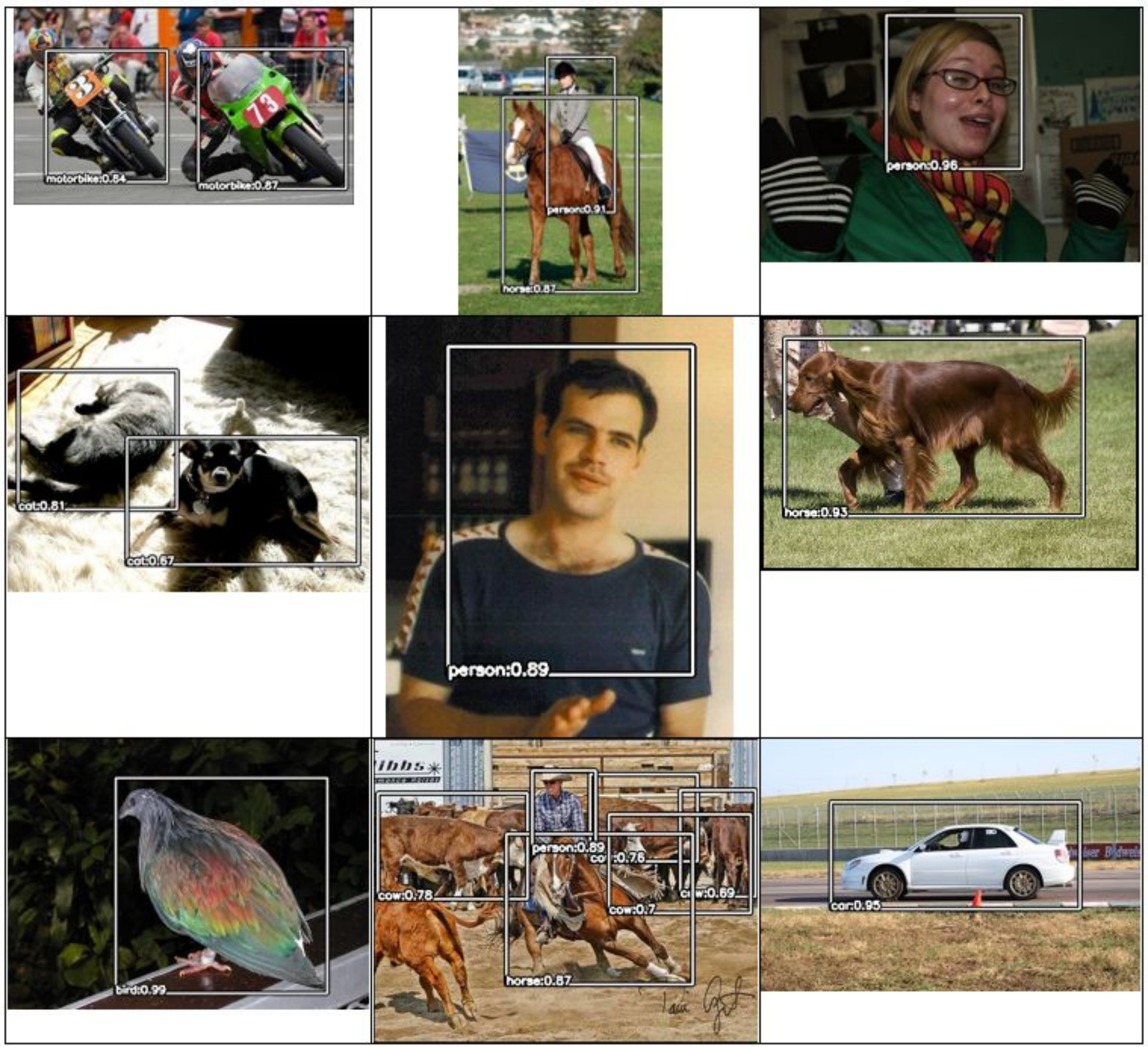
Model Backbone	Train Accuracy	Val Accuracy	Test Accuracy	Classifier Loss (Train)	Classifier Loss (Val)	Regressor Loss (Train)	Regressor Loss (Val)	mAP
Resnet50	64.5%	59.9%	57.1%	0.297	0.335	0.362	0.385	62.8%
VGG16	60.1%	56.4%	53.7%	0.311	0.369	0.384	0.438	59.4%
MobileNet V2	55.1%	52.3%	48.6%	0.323	0.371	0.419	0.441	55.9%

Results


person:0.98
sheep:0.99
aeroplane:0.92
motorbike:0.95
cat:0.78
cat:0.6
dog:0.8
dog:0.87
bus:0.6
train:0.55
bus:0.72
bus:0.92
bo
bottle:0.75
car:0.96

Results


tvmonitor:0.99
bird:0.98
person:0.96
tvmonitor:0.88
bird:0.93


motorbike:0.84
motorbike:0.87
person:0.91
horse:0.87
person:0.96
cat:0.81
cat:0.67
person:0.89
horse:0.93
bird:0.99
person:0.89
cow:0.76
cow:0.78
cow:0.69
cow:0.7
horse:0.87
car:0.95

Inferences

- We could not not achieve the results as good as the original paper, which can be attributed equally to having less experience with hyper-parameter tuning and limited computational resources.
- In general, the results are in the order: Resnet50 > VGG16 > MobileNetV2
- Fast RCNN is much more feasible to train compared to RCNN, with no significant hit to accuracy.
- Faster RCNN is supposed to even better, but we could not get it to work.

Reference(s)

1. Rich feature hierarchies for accurate object detection and semantic segmentation - Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik (CVPR 2014) - Link
2. Selective Search for Object Recognition - Uijlings, Jasper & Sande, K. & Gevers, T. & Smeulders, A.W.M. (IJCV 2013) - Link
3. Very Deep Convolutional Networks for Large-Scale Image Recognition - Karen Simonyan, Andrew Zisserman (ICLR 2014) - Link
4. Fast R-CNN - Ross Girshick (ICCV 2015), 2015 - Link

The End