Deep Learning with Convolutional Neural Networks

- Evaluating performance of object detection models

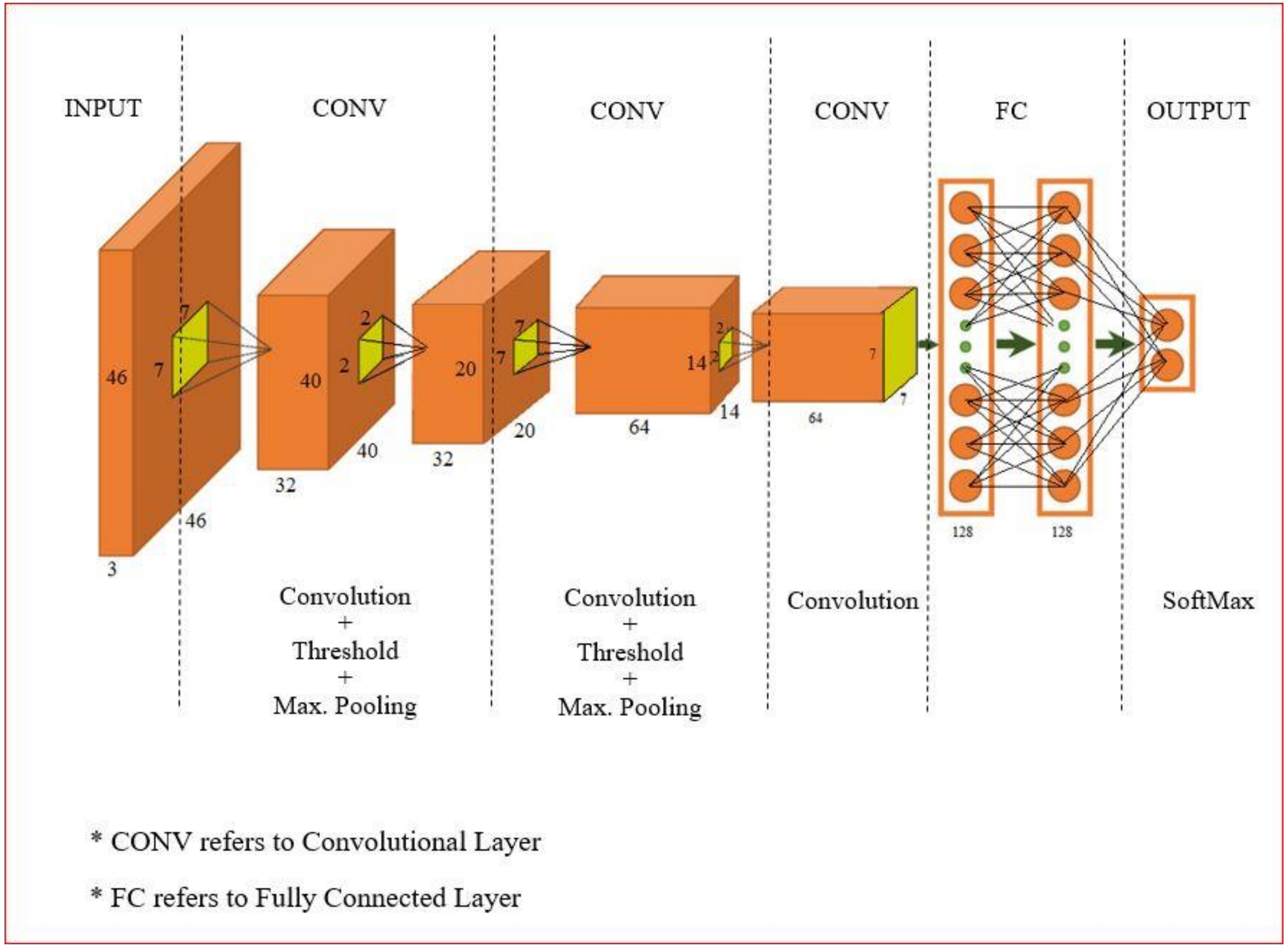
Presented by Nikhil Sharma

CSC 720 – AI 2 || Term Project || Spring 2017 || nsharm12

Objective

- Train a Human Upper Body Detector
- Train an Object Detector with pre-defined set of classes
- Combine the detections and evaluate the performance
  - How difficult does it get for a single detector to classify, as the number of classes increases

Upper Body Detection



CNN architecture

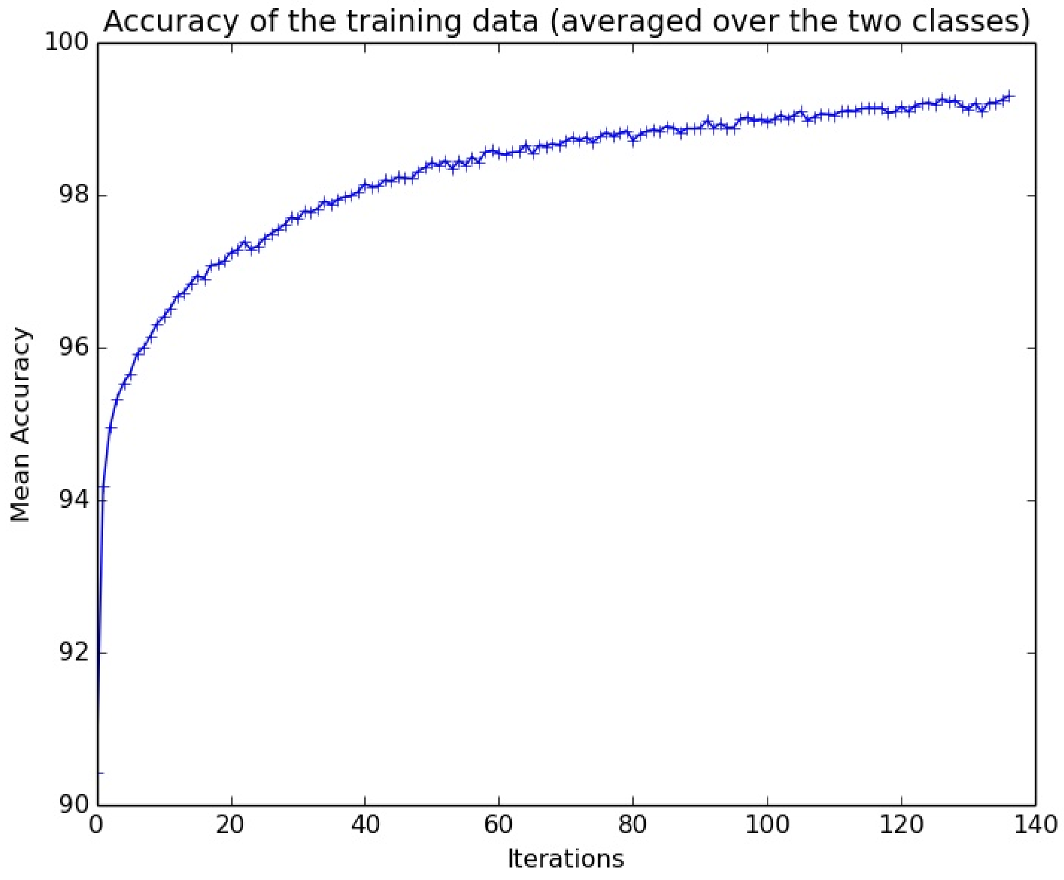
Data

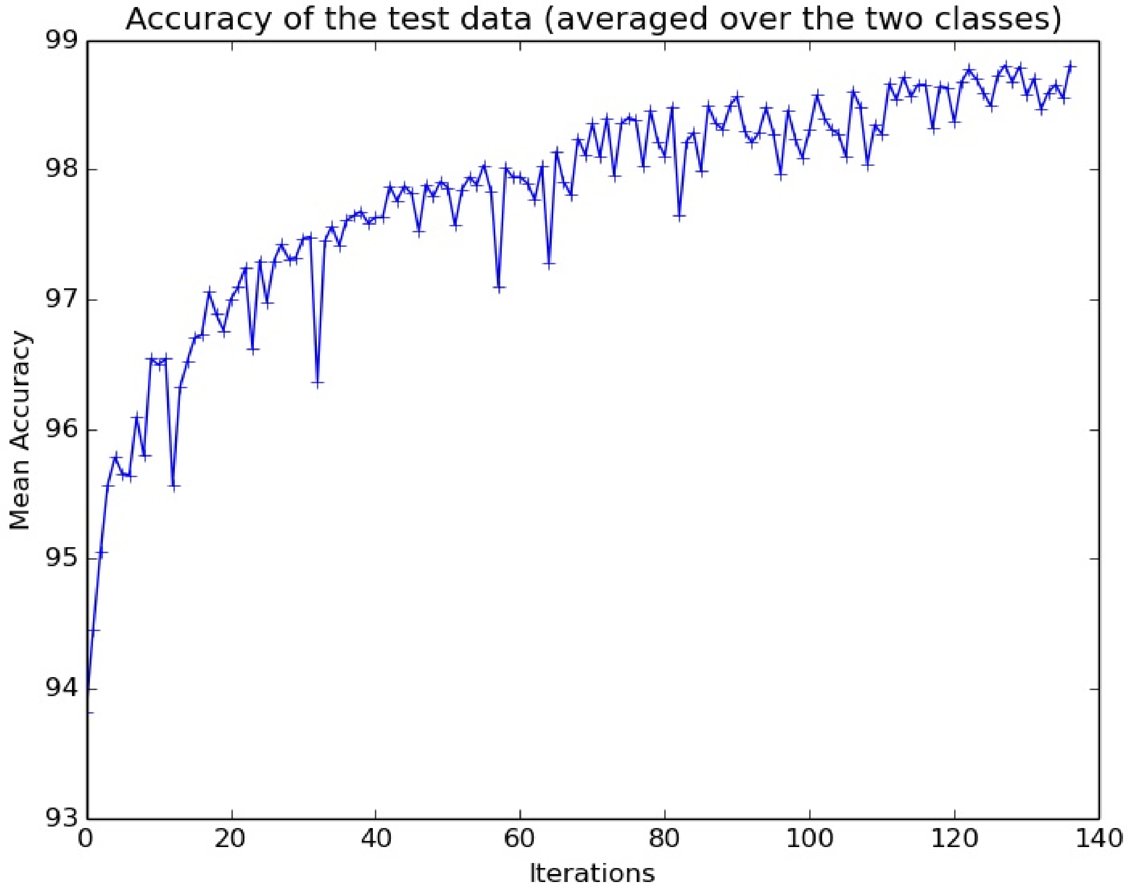
- INRIA Person Dataset
- CUHK 03 Dataset (Chinese University of Hong Kong)
- MIT Scene Detection Dataset

- #Upper Body Images = 10K
- #Background = 32K

- Training : Testing split = 80 : 20

Results of training the model





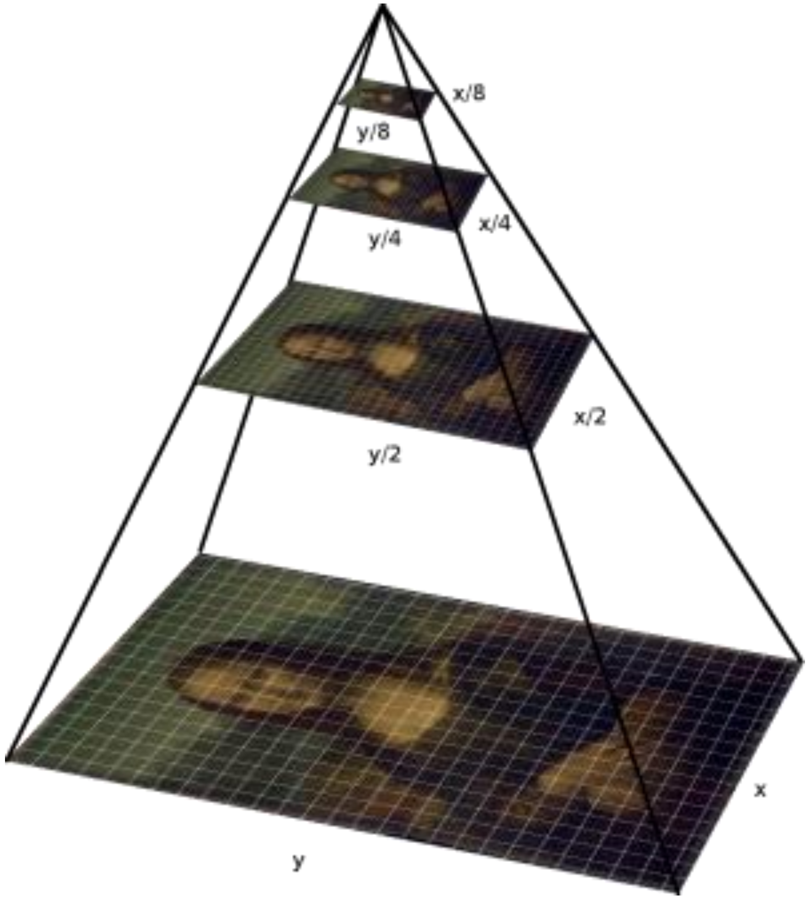
Confusion Matrix

```
==> doing epoch on training data:
==> online epoch # 137 [batchSize = 128]
 [==================== 35201/35312 ===========>.]  ETA: 906ms | Step: 8ms
==> time to learn 1 sample = 8.1482129057555ms
ConfusionMatrix:
[[    8415      132]   98.456%   [class: body]
 [     116    26537]]  99.565%   [class: backg]

==> testing on test set:
 [==================== 8705/8829 =============>.]  ETA: 0ms | Step: 23h59m
==> time to test 1 sample = 2.1952222300651ms
ConfusionMatrix:
[[    2141       45]   97.941%   [class: body]
 [      59     6459]]  99.095%   [class: backg]
```

Detecting Upper Bodies

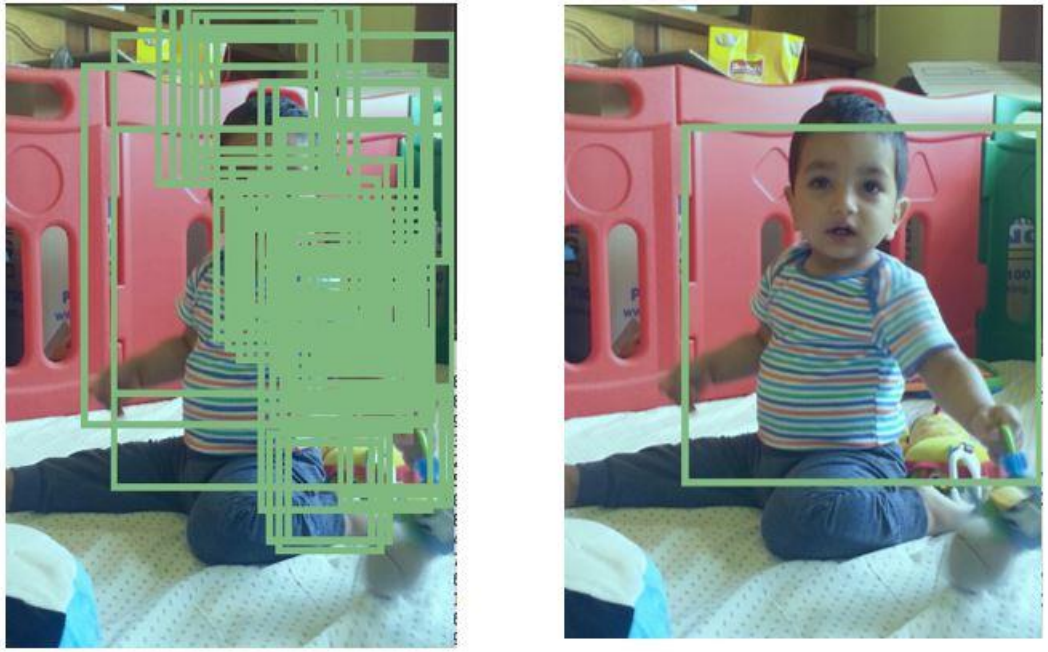
1. Image Pyramid



- For most of the images, scales = {0.3, 0.15, 0.1} worked

Detection when 0.5 was included in the list of scales

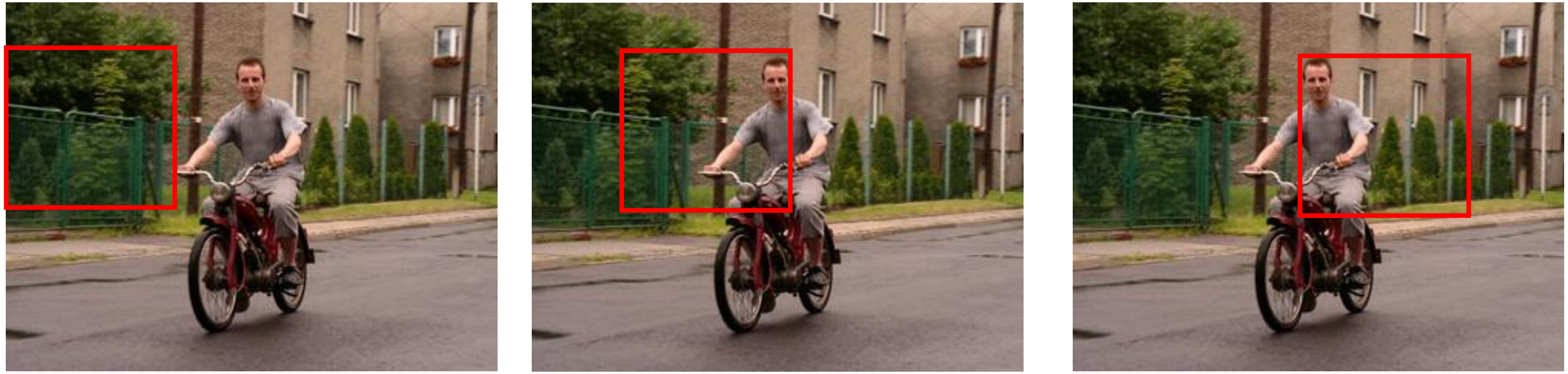
Detection when 0.4 was included in the list of scales



Detection when 0.5 and 0.4 were included in the list of scales

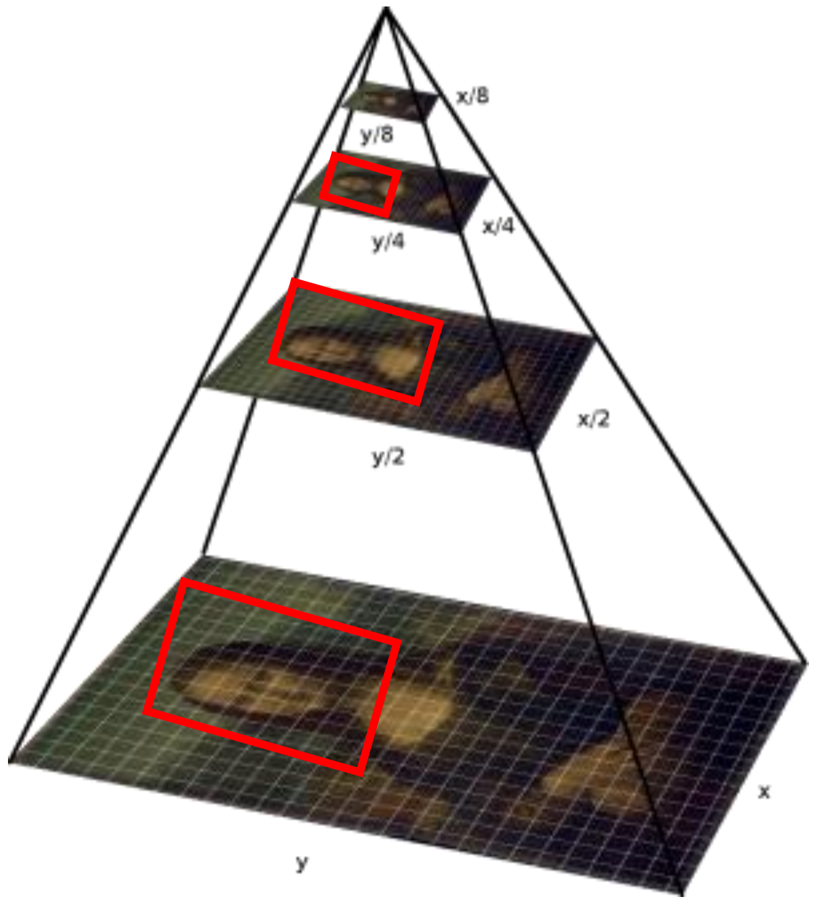
Detection when scales were just 0.3, 0.15, 0.1

2. Sliding Window

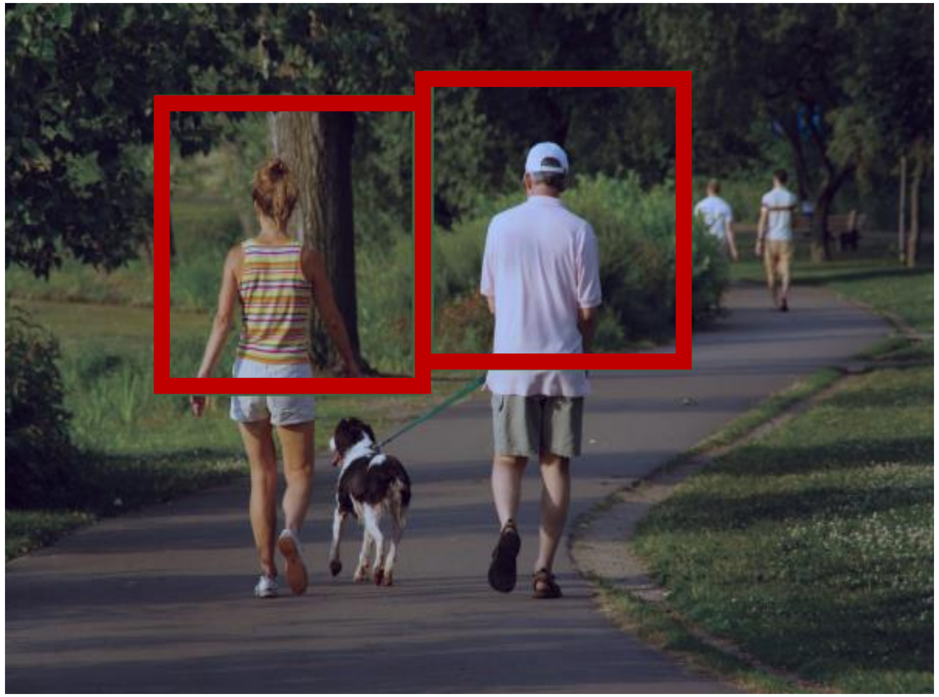


3. Non-maximum Suppression

- To reduce multiple detections of the same object at different scales

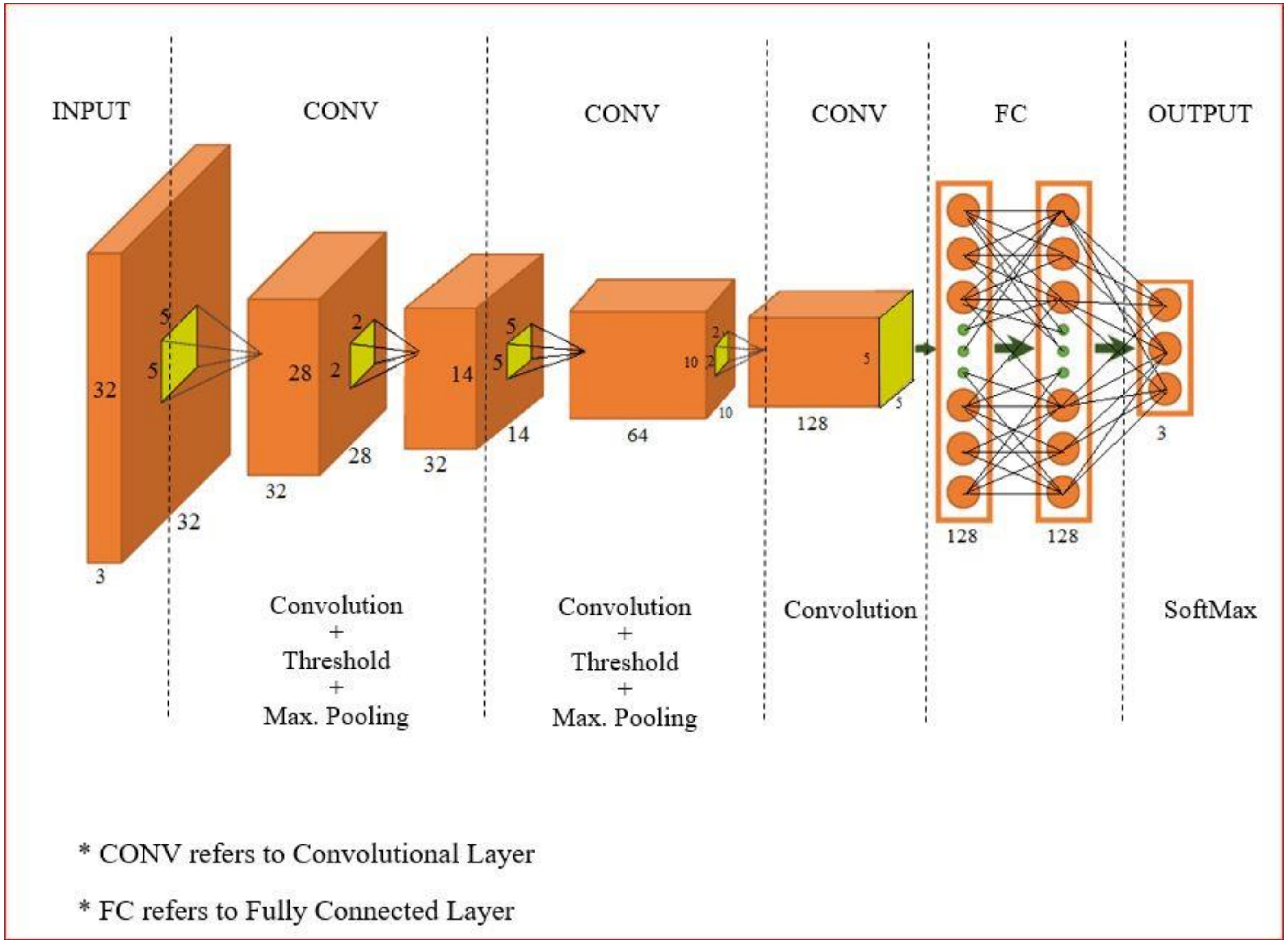


$$\textit{Overlap score of two detections A and B} = \frac{Area\ (A\ \cap\ B)}{Min(Area\ of\ A,\ Area\ of\ B)}$$



It is easy to fool the detector!

Object Detection

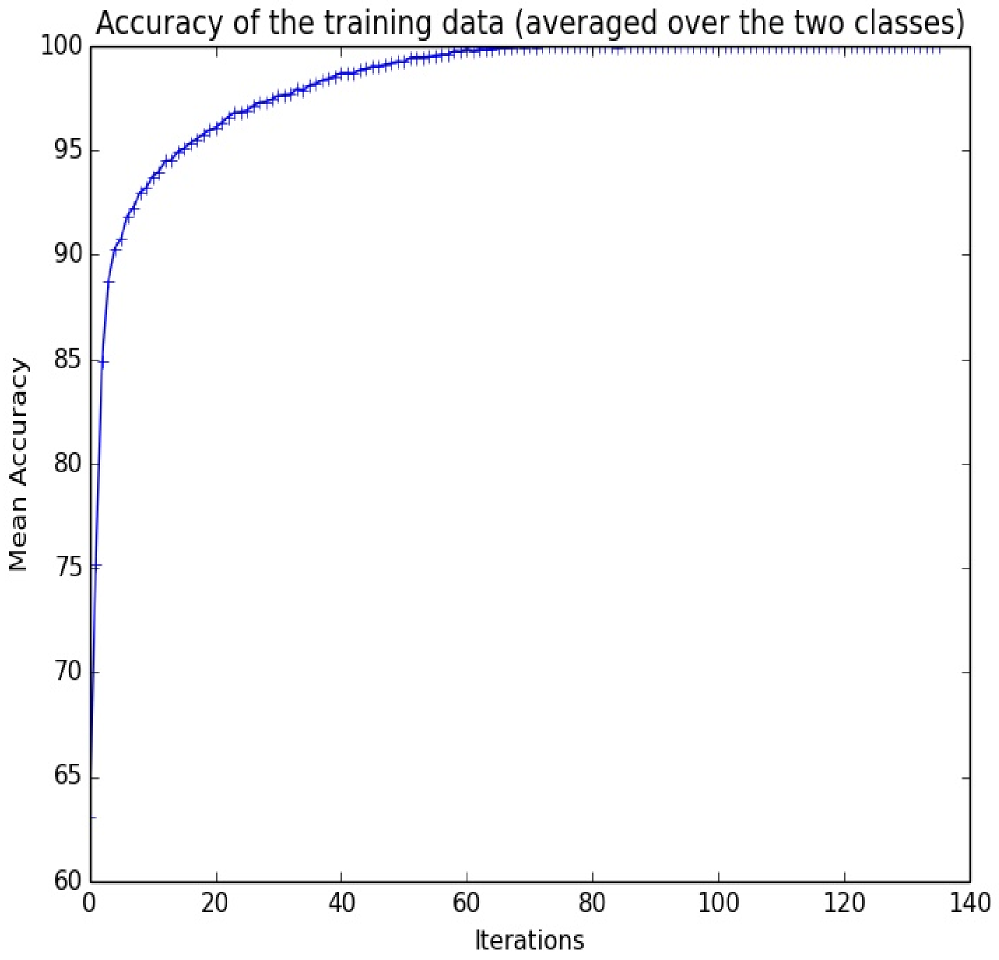


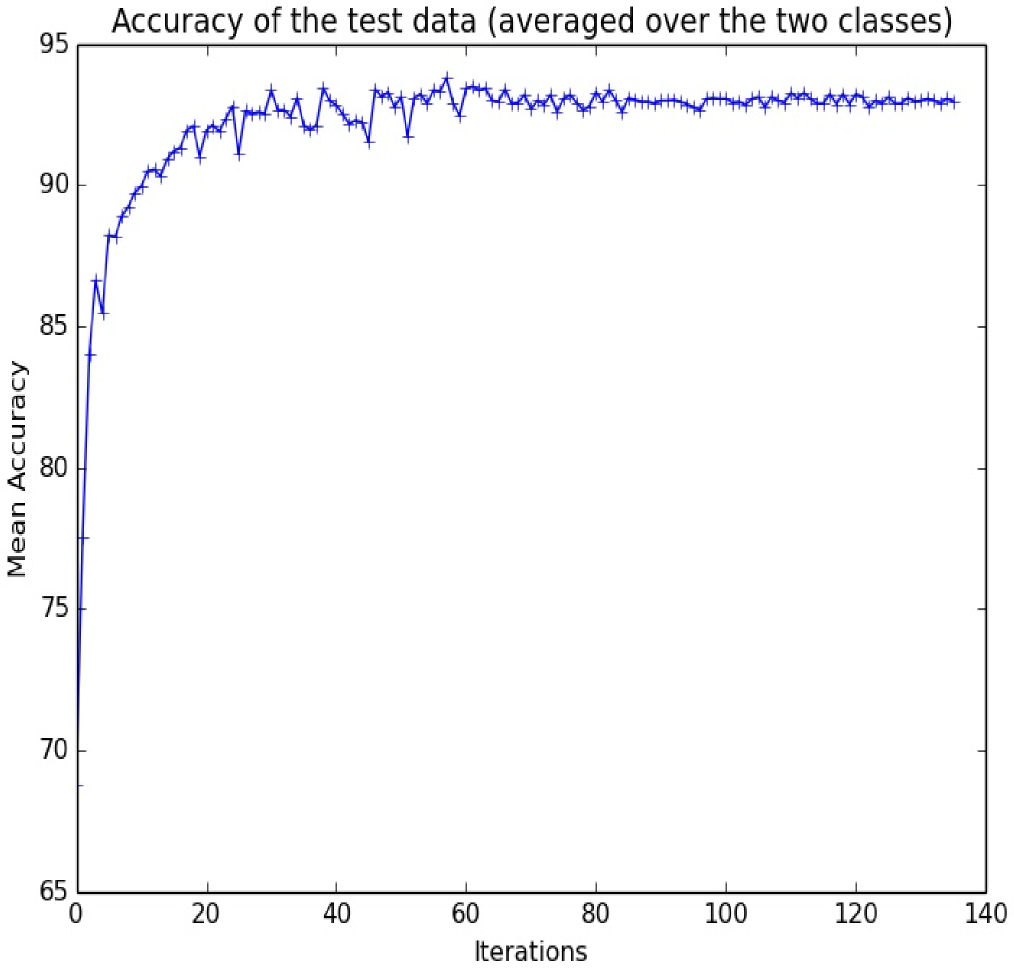
CNN architecture

Data

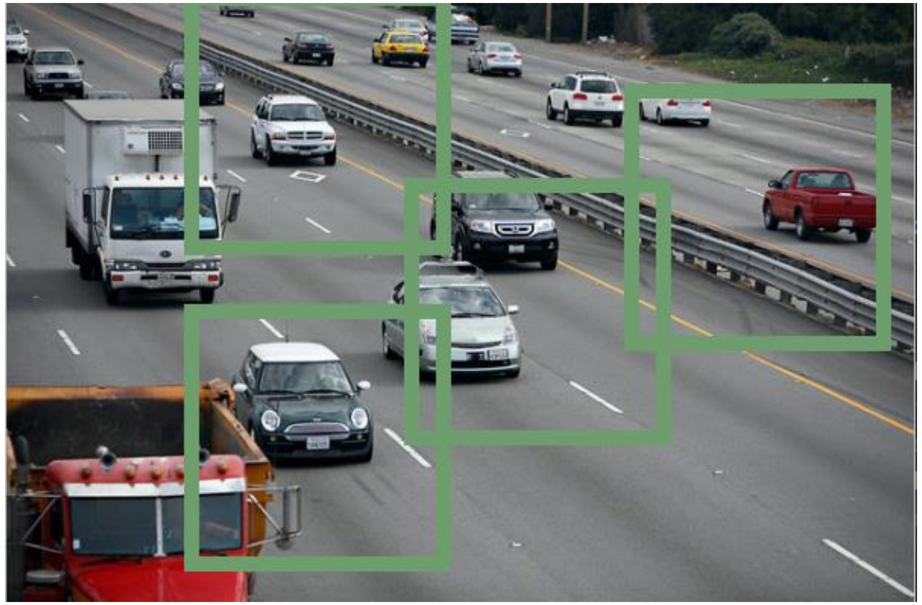
- CIFAR – 10 Object Recognition Dataset (6K images each of 10 classes)
- Classes used: Automobile, Dog, and Background
- Training : Testing ratio = 5 : 1

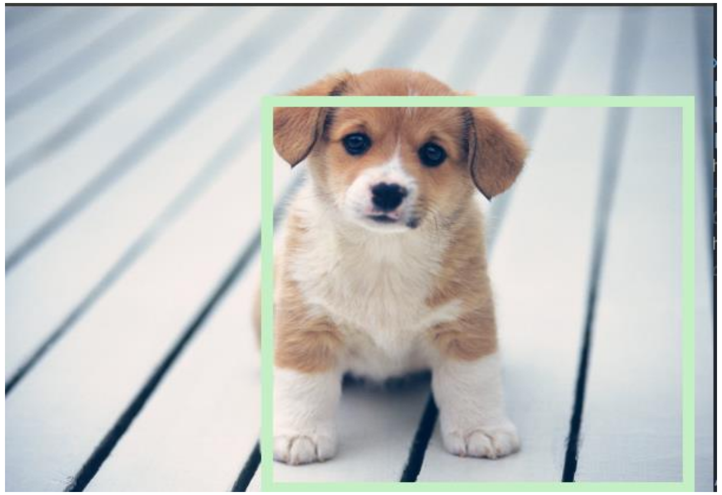
Results of training the model



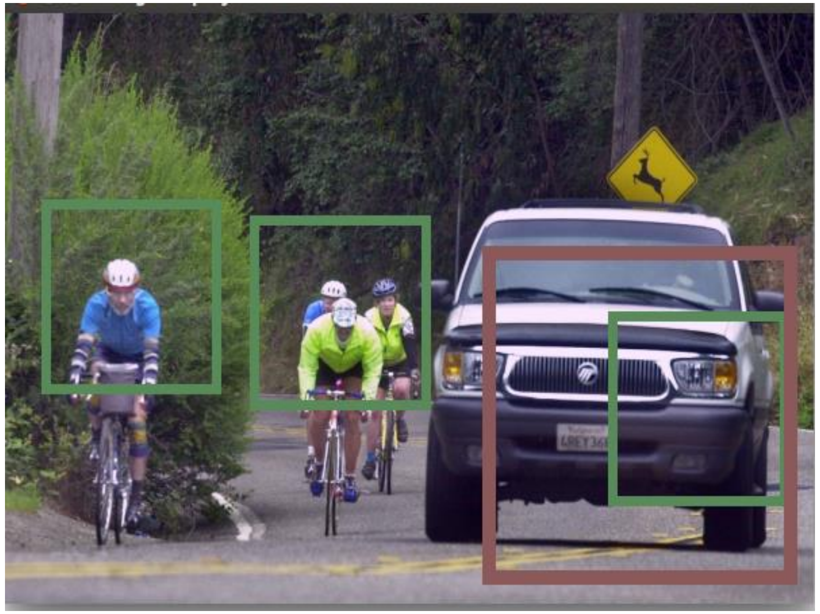


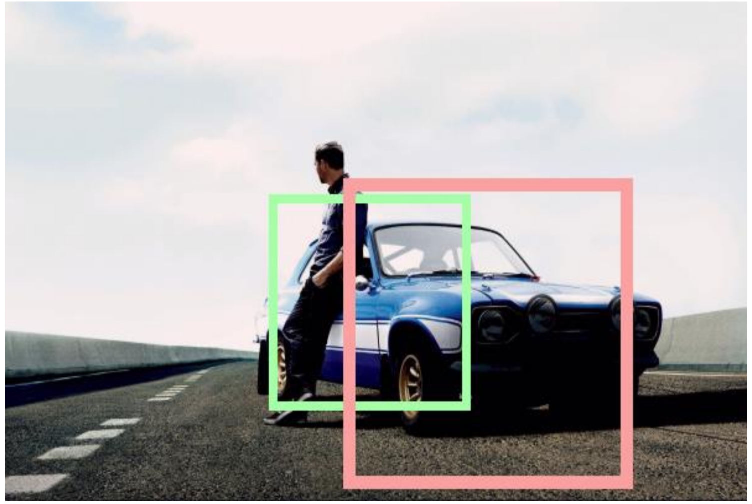
Detections using Image Pyramids, Sliding Windows, NMS

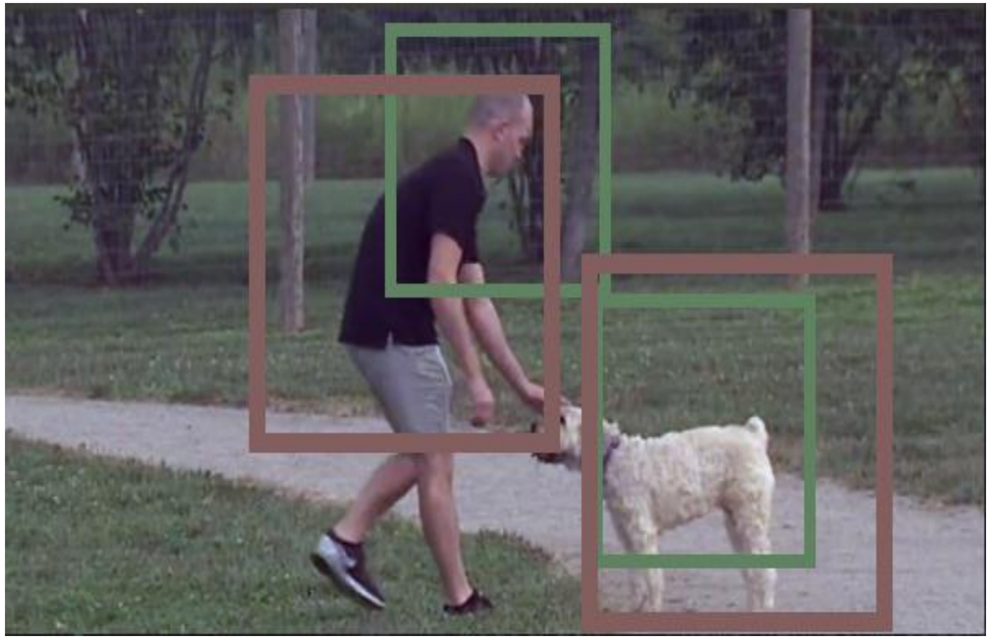




Combining Detections



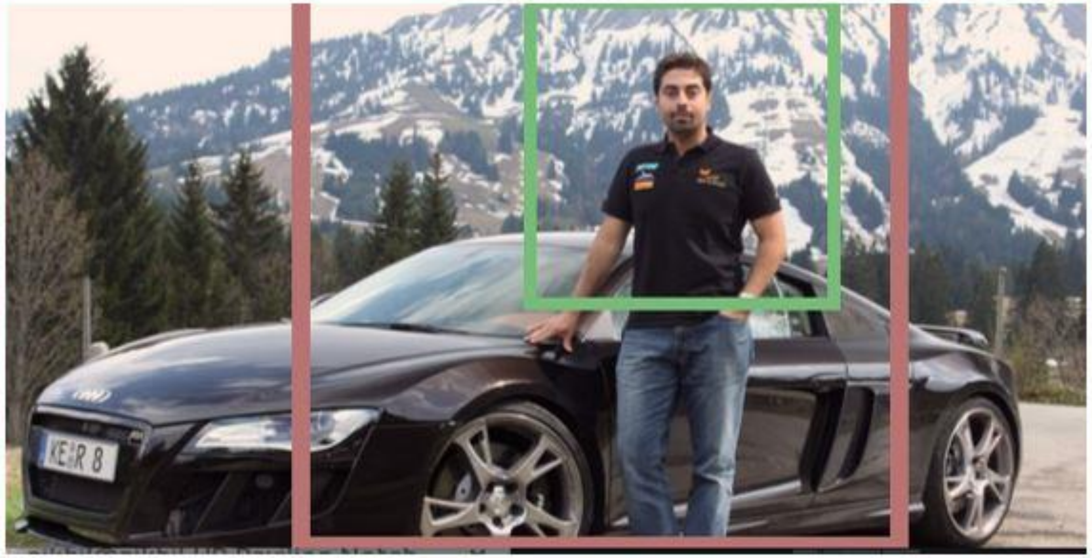




- Confidence score that the detector outputs for each detection is actually its belief w.r.t the other categories, and not independent of them

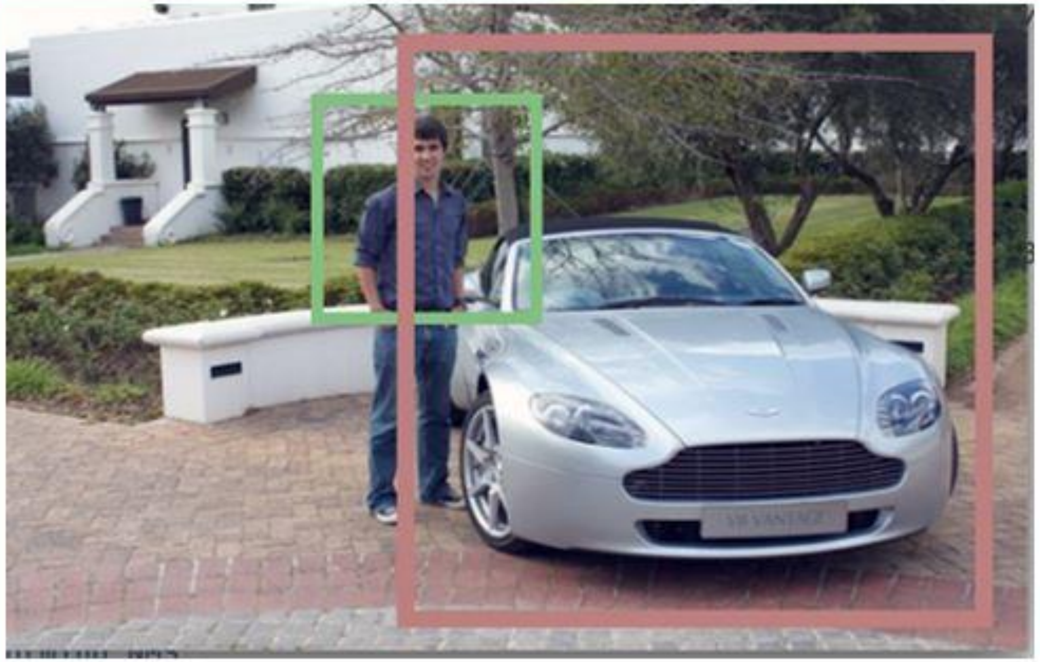
Further Detections

- Color extraction
  - K-means to cluster the colors
  - Convert RGB values to LAB and compare using Euclidean distance



Upper Body and Car Detected
Color Tag: Black

Upper Body and Car Detected
Color Tag: Black



Upper Body and Car Detected
Color Tag: Blue

Identifying multiple objects

- Train individual detectors and run them in parallel

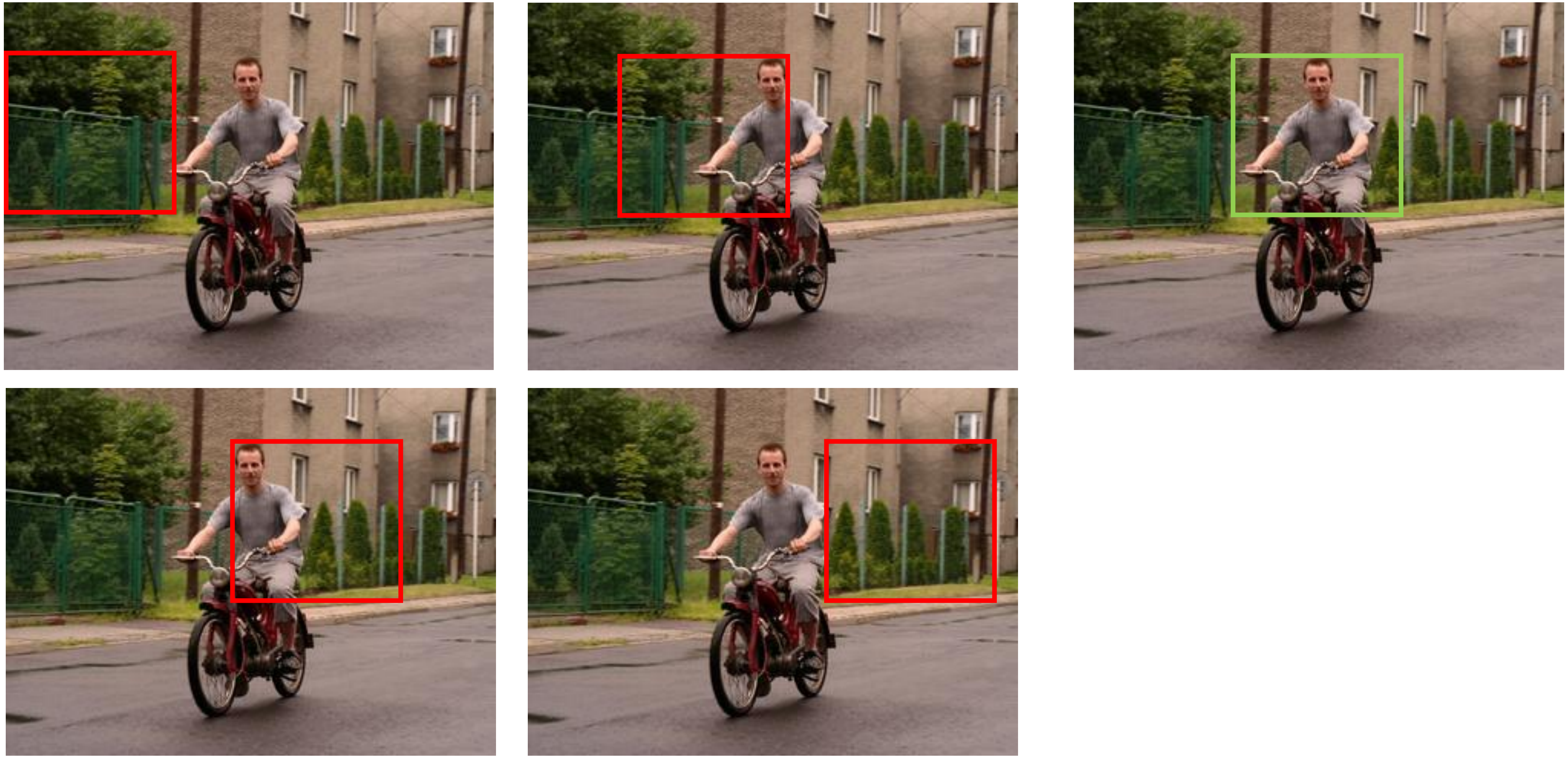
  OR

- Train a single detector with multiple classes

```
==> testing on test set:
 [==================== 4996/5000 =============>.]  ETA: 5ms | Step: 1ms
==> time to test 1 sample = 1.4713657855988ms
ConfusionMatrix:
[[     888      39      40      14      19]   88.800%   [class: airplane]
 [      34     909      30      14      13]   90.900%   [class: automobile]
 [      41      20     654     218      67]   65.400%   [class: cat]
 [      23      11     211     681      74]   68.100%   [class: dog]
 [      30       8      58      95     809]]  80.900%   [class: horse]
 + average row correct: 78.819999694824%
 + average rowUcol correct (VOC measure): 66.328020691872%
 + global correct: 78.82%
```

Future Scope

- Use sliding window while training itself



- Dropout
- PCA with SVM
- Re-training with hard examples
- Regularization

Thank You