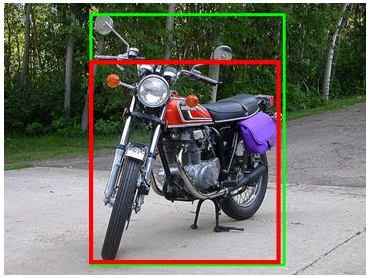
Object Detection

Tejus Gupta



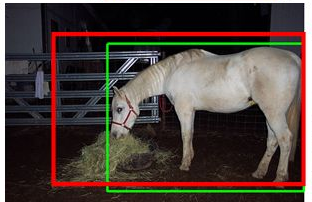











Simplest Method: Sliding Window Approach

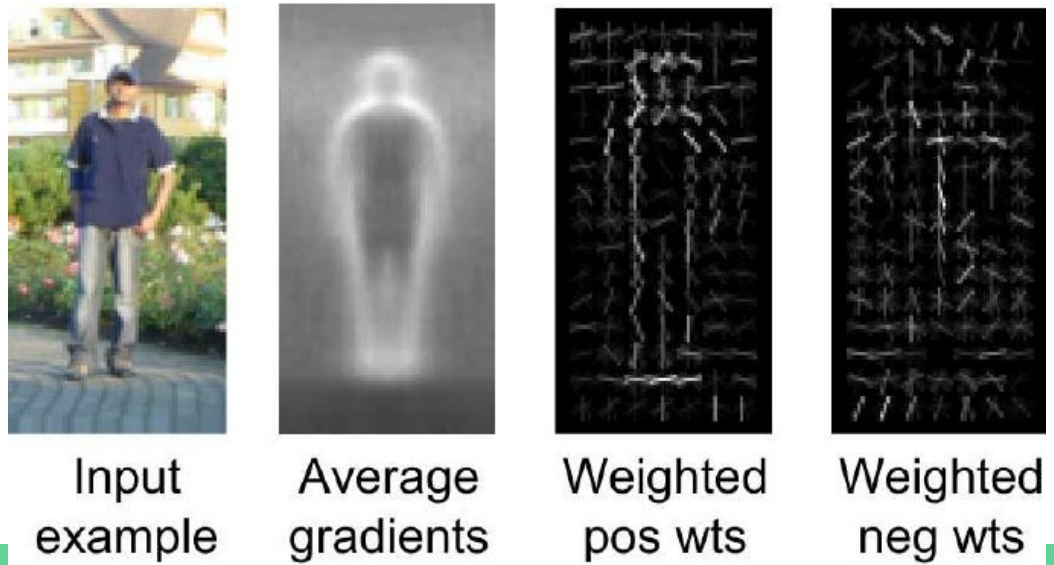
- Use a classifier that can identify if an image has an object.
- Use boxes with different scales, aspect ratios and positions to detect object.
- Be careful: Non maximal suppression

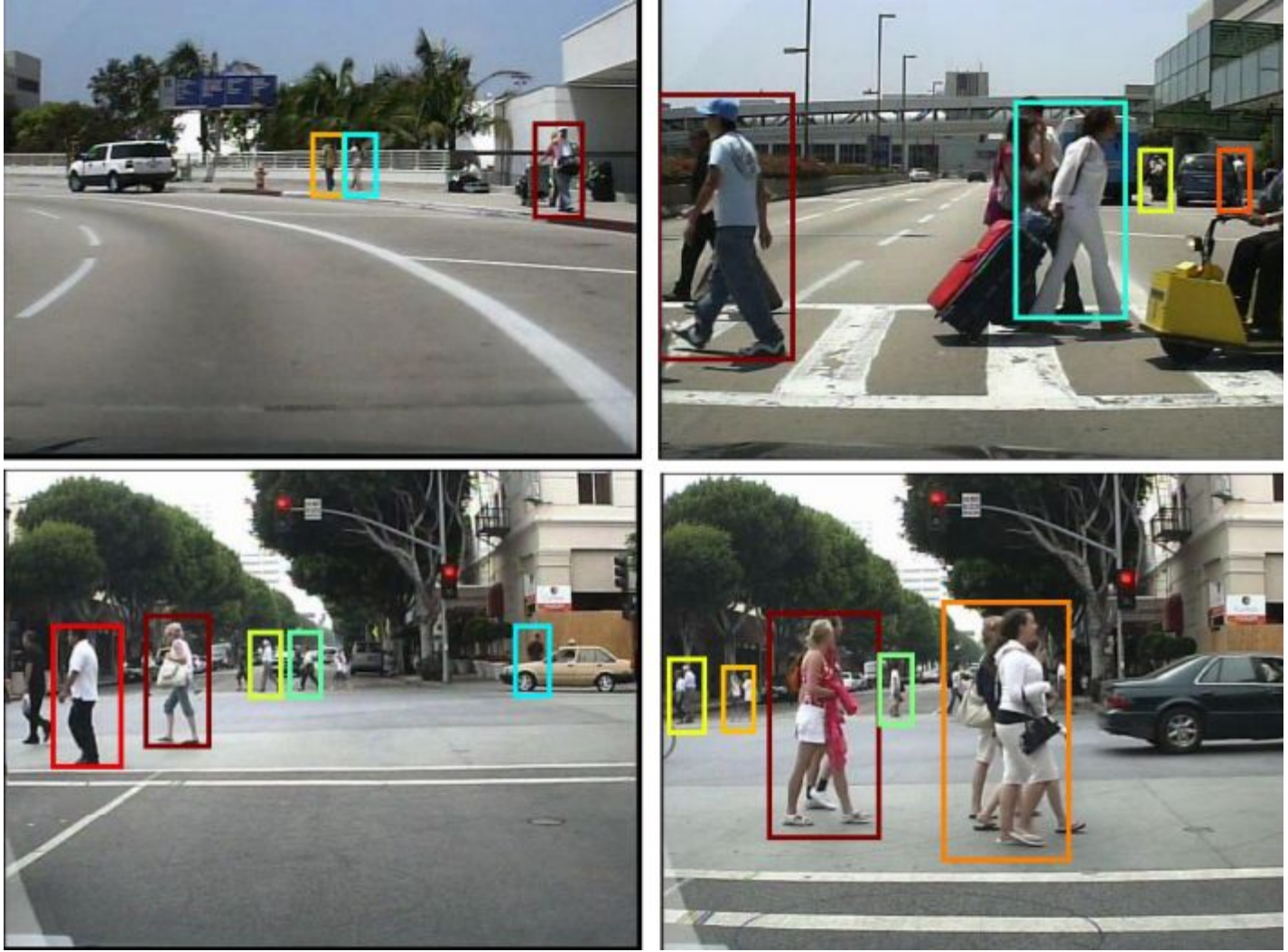
What classifier to use?

- Template matching

What classifier to use?

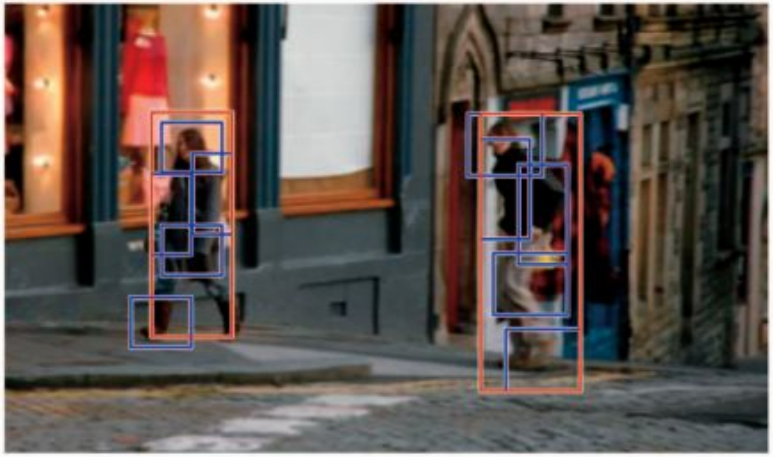
- Histogram Of Gradient (Dalal and Triggs)
  - (Was so good, authors created new dataset INRIA)
  - Local appearance and shape characterized by edge direction, not precise location
  - Divide image into cells, compute 1d histogram, normalize
  - For humans, coarse spatial sampling, fine orientation sampling and strong normalization best
  - Use (linear)SVM

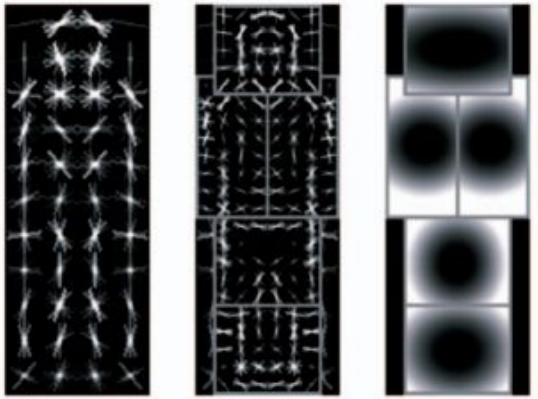




What classifier to use?

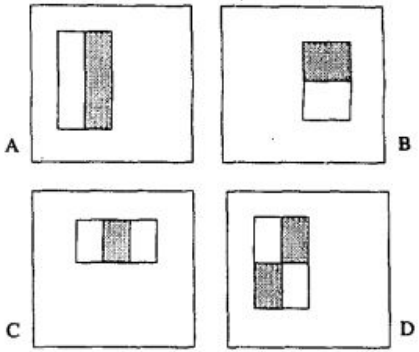
- Deformable Parts Model (Girshick et al - Univ. of Chicago)

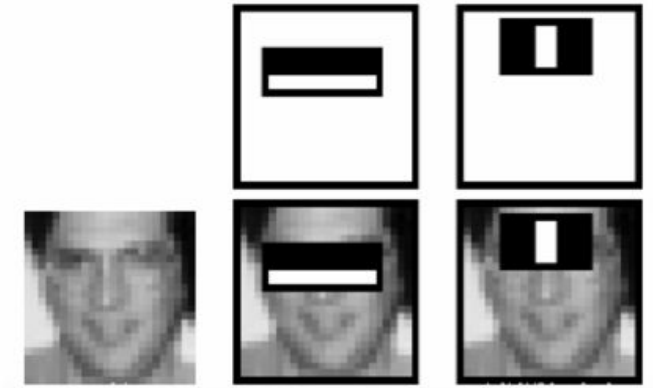




What classifier to use?

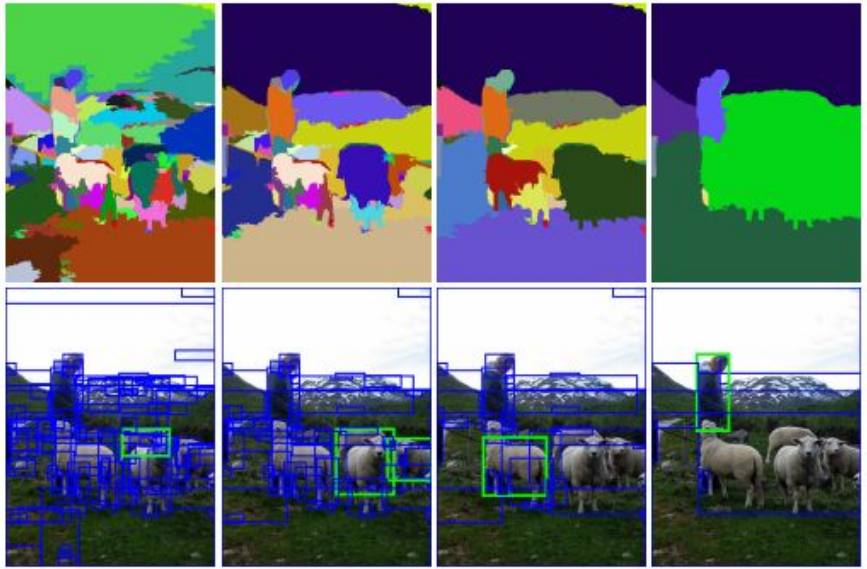
- Boosted cascade of Simple features(Viola, Jones - Cambridge)
  - Integral image allows very fast feature evaluation
  - Just 3 kinds of features
  - Over 180,000 rectangle features for a window, much more than pixels
  - Apply each feature, choose best threshold and select features with min error rates(->6000 features)
  - Train a decision tree to select which features to evaluate

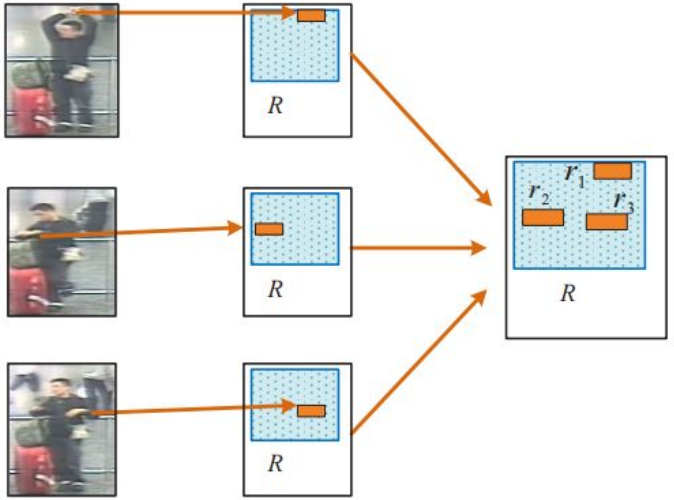




Segmentation as selective search + Regionlets features

- Use hierarchical segmentation to guess object location.
- Different strategies for deformable and rigid objects.
- Small regions - vulnerable to variations, big regions - poor localization.
- Regionlets are designed capture same appearance in different locations due to deformations.





Build a 3D Model

What classifier to use?

- Convert image to 1D array.
- Or use hand engineered feature extractors
- Train a classifier (kNN, ...) based on some distance metric.

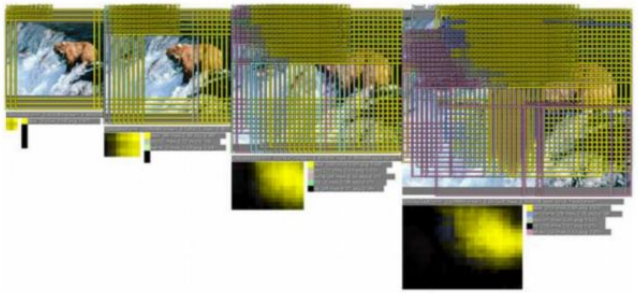
Pascal VOC 2007 MAP

- DPM(2011) - 33.7
- Regionlets - 41.7
- R-CNN + AlexNet - 54.2
- R-CNN + BBox regression - 58.5
- R-CNN + VGG16 - 66
- Fast R-CNN - 69 (0.5 FPS)
- Faster R-CNN - 73 (VGG16, 7FPS) and 62 (ZF, 18FPS)
- YOLO - 63.4 (45 FPS)

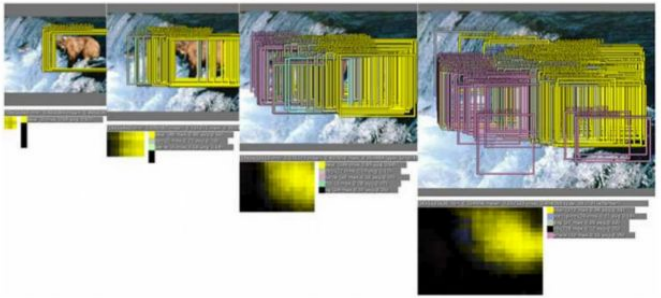
Convolutional Neural Networks

Try every box, but smartly

Window positions + score maps

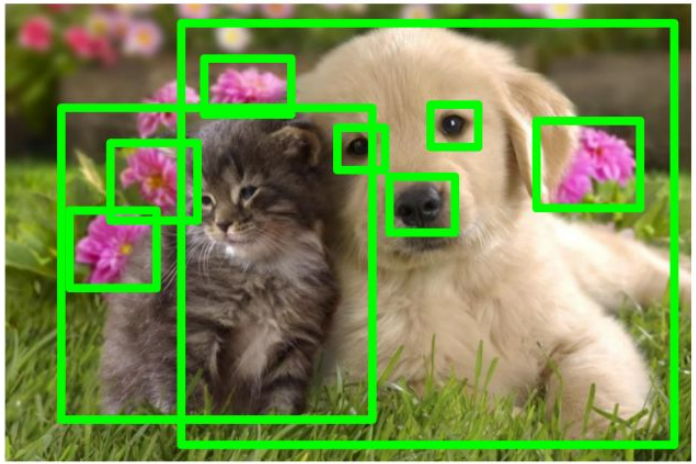


Box regression outputs

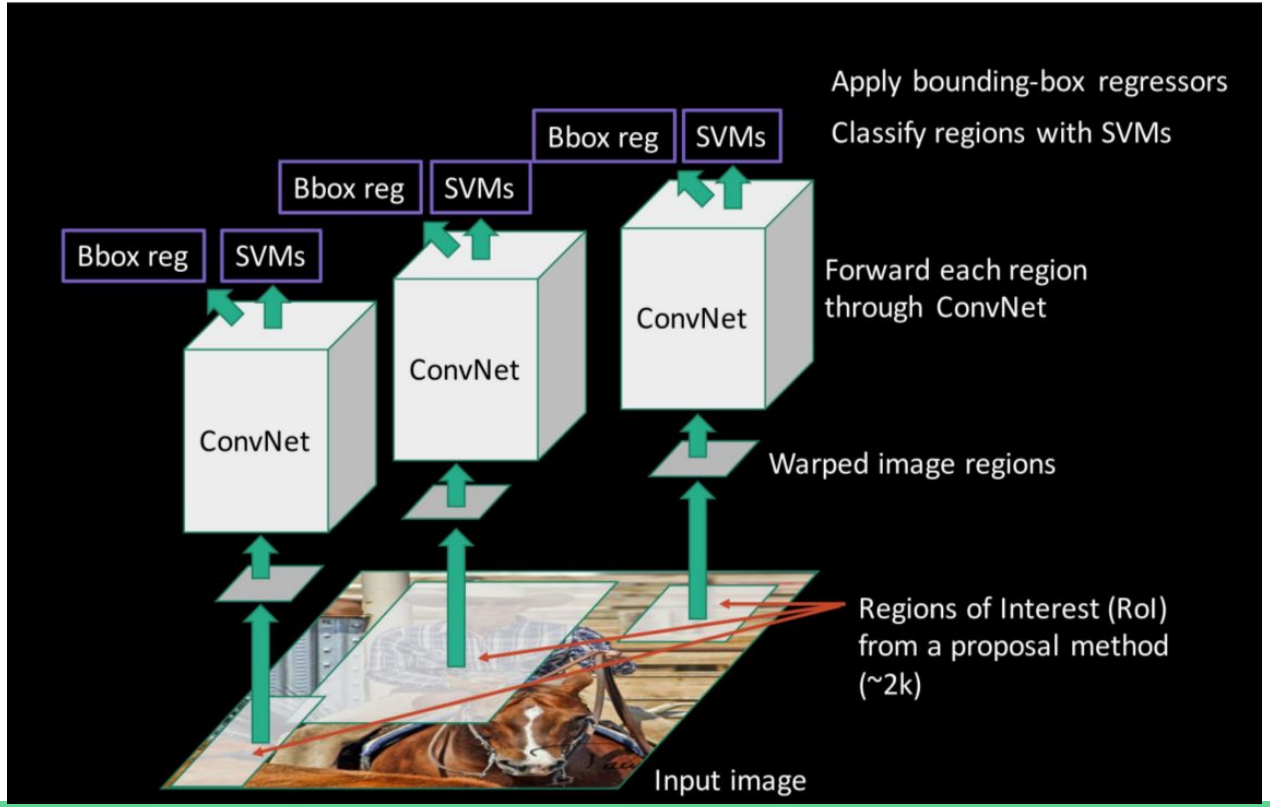


Final Predictions

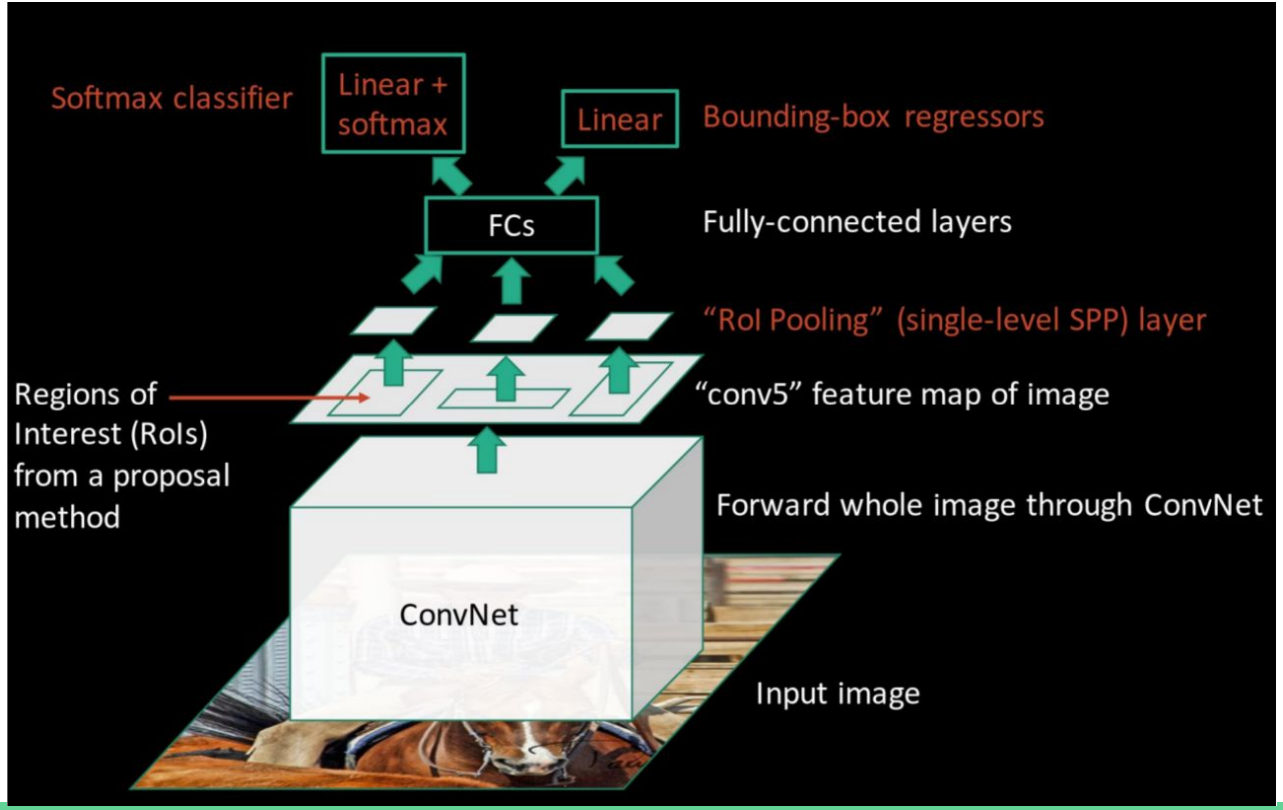
Region Proposal



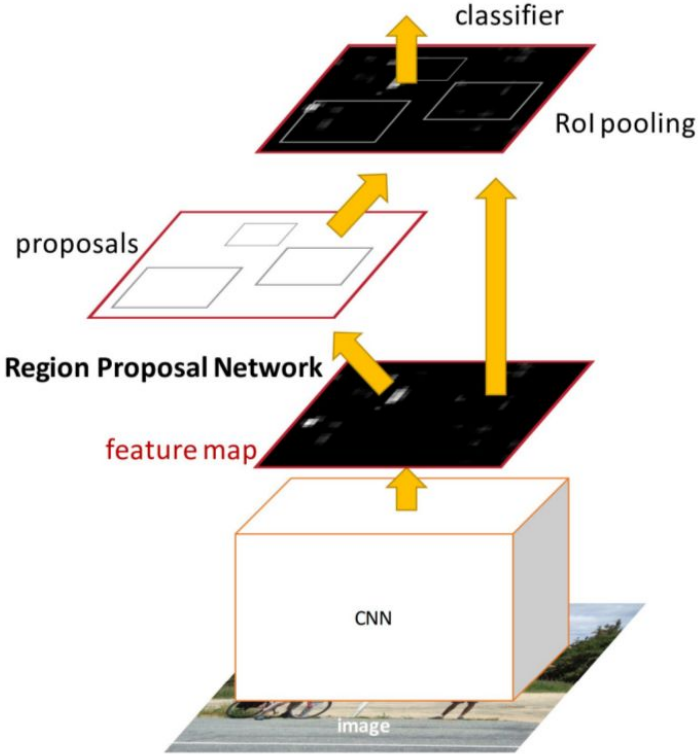
R-CNN


Apply bounding-box regressors
Classify regions with SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Forward each region through ConvNet
ConvNet
ConvNet
ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

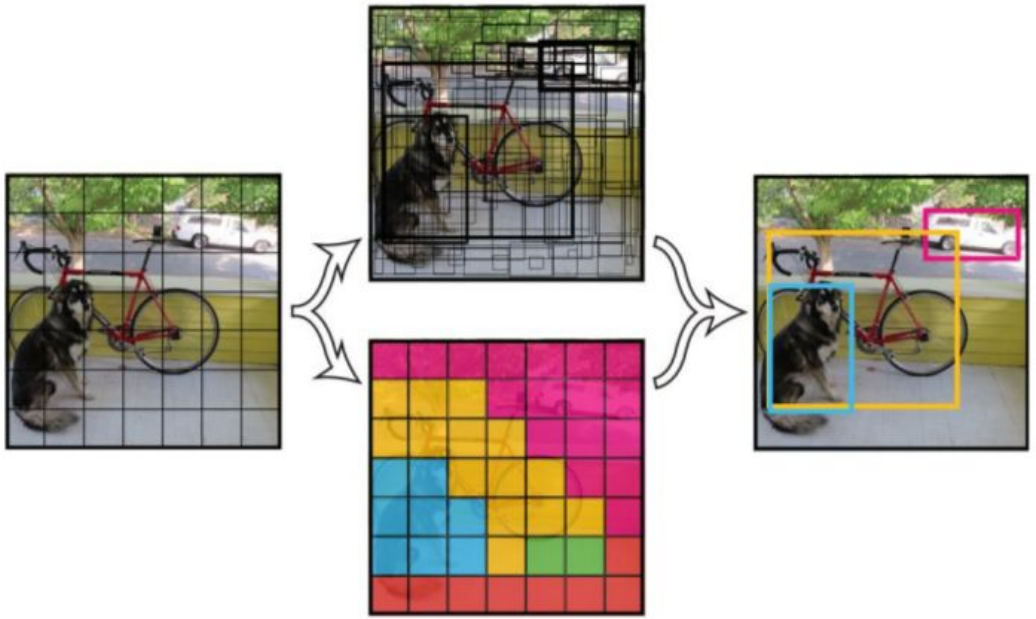
Fast R-CNN


Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
FCs
Fully-connected layers
"RoI Pooling" (single-level SPP) layer
Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image

Faster R-CNN


classifier
RoI pooling
proposals
Region Proposal Network
feature map
CNN
image

YOLO



Andrea Vedaldi's Tutorials

Decision tree trained on VGG16 features

Bag of Visual Words

Thank You

Tejus Gupta

tejusgupta14@gmail.com