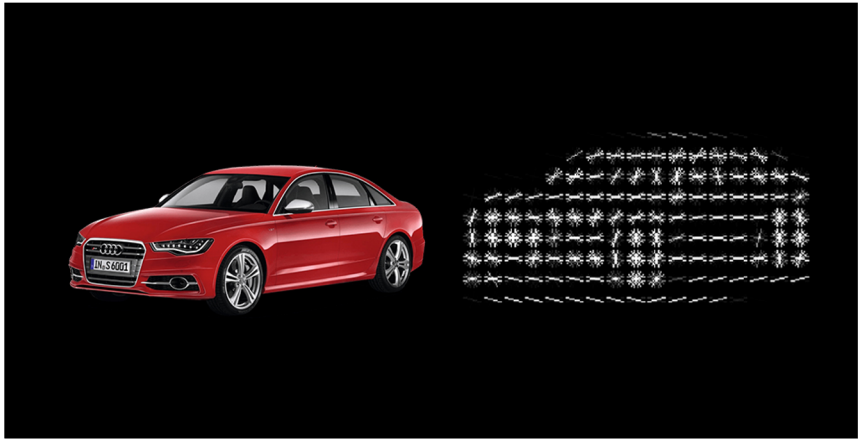
R-CNN: Region Based Convolutional Neural Networks

Duke

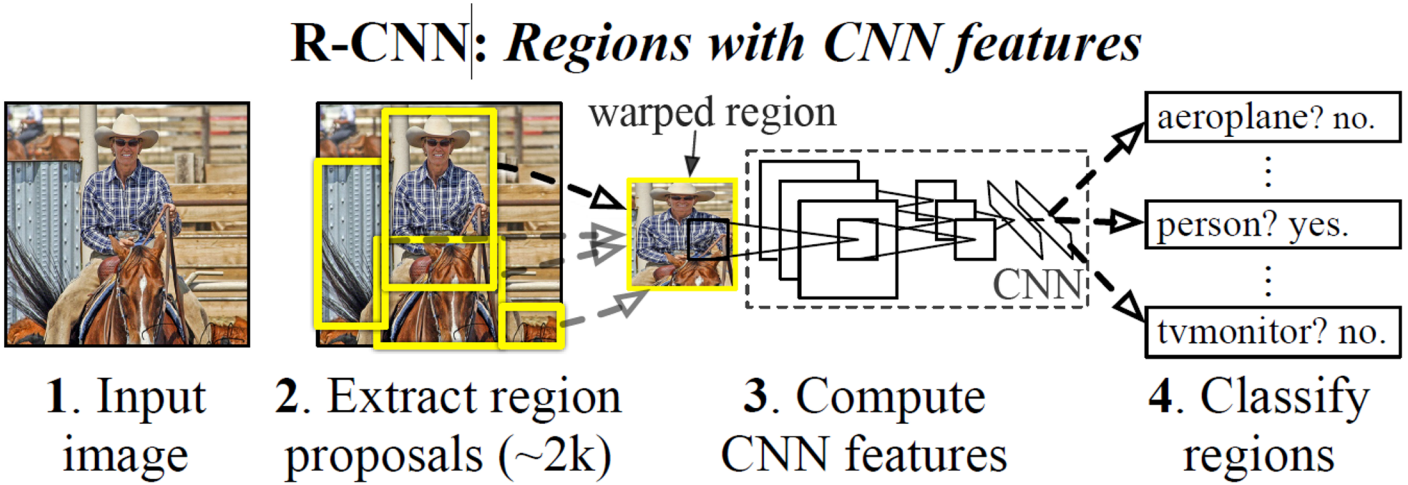
Context

- Previous object detection used methods such as HOG and SIFT
  - Extract features from image using gradients
  - Computationally expensive
  - Ensemble models
- Object detection requires better localization within the image
- Use of deep learning (CNNs) can provide more efficient feature extraction
  - Improved object detection



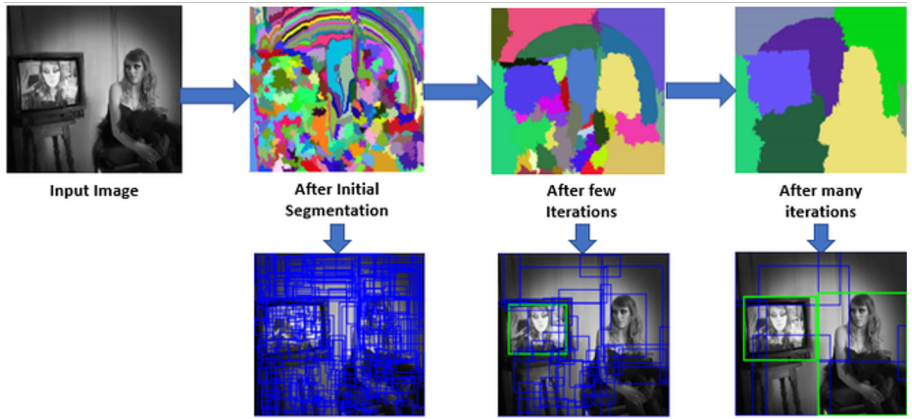
Duke

R-CNN (2014) Overview



Key Network Features

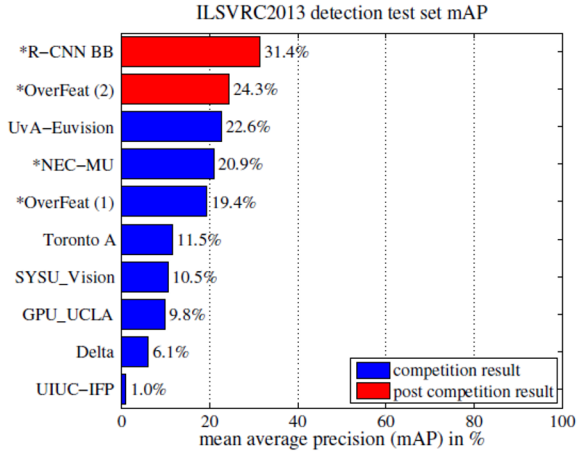
- Uses selective search for generating region proposals
  - Can work with other methods
- Efficiency
  - CNN parameters shared across all classes
  - Low dimensional feature vectors compared to prior models (360k vs 4k dimensions)
  - Several orders of magnitude faster

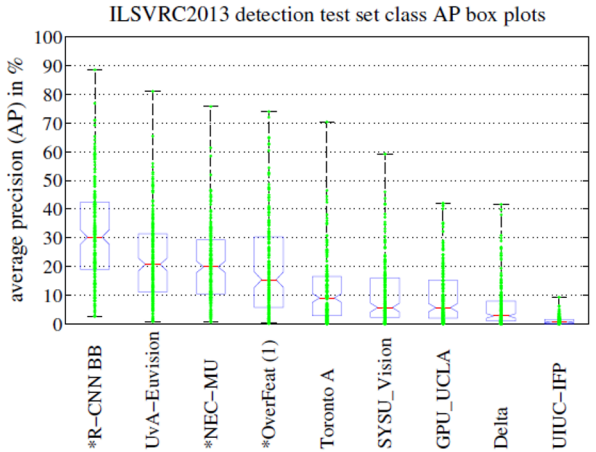


Duke

Results

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [20]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [39]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [41]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [18]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**





Duke

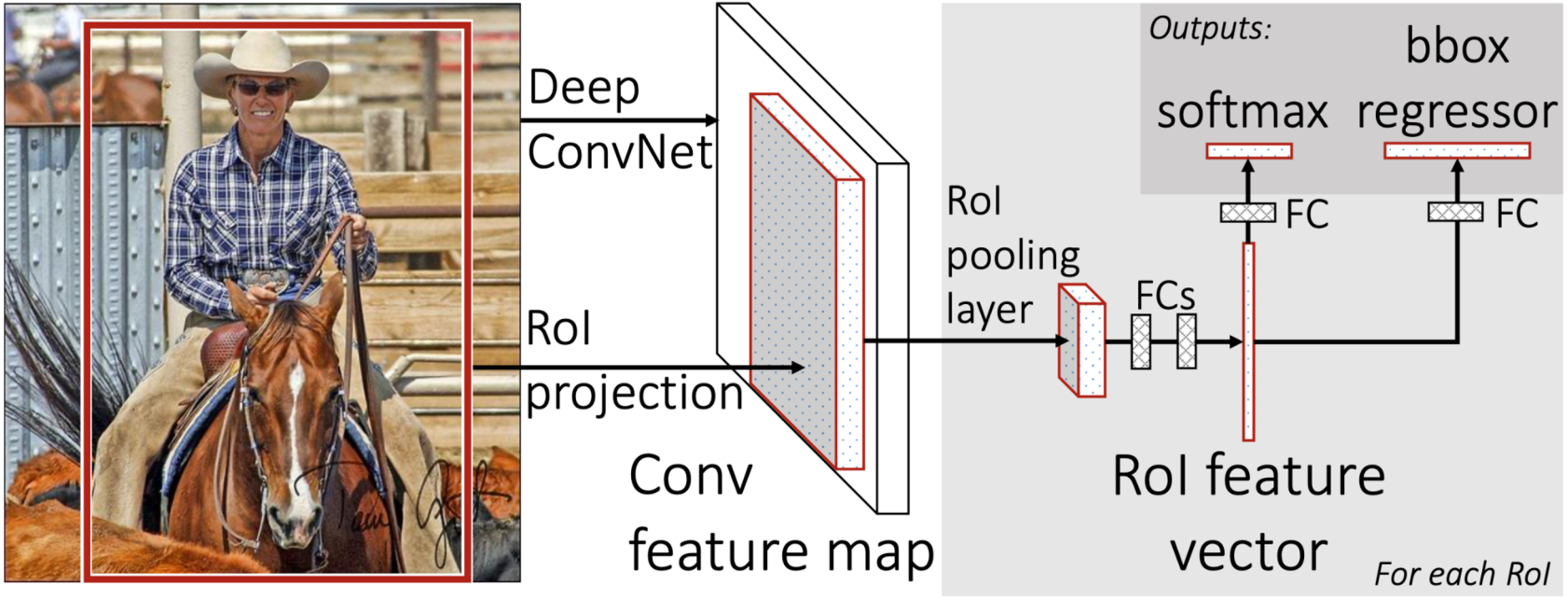
Limitations of R-CNN

- Major limitations of R-CNNs:
  1. Training is a multi-stage pipeline
  2. Training is expensive in space and time
  3. Object detection is slow (47 seconds on test data)
- R-CNN is slow primarily because it performs a convolutional network forward pass for each object proposal, without sharing computation.
- R-CNN also generates 2,000 region proposals per full image making it slow to train and test.

From R-CNN to Fast R-CNN

- Fast R-CNNs improve on R-CNNs in terms of space, time, and accuracy.
  1. Training is single-stage, using a multi-task loss function
  2. Training can update all network layers
  3. No disk storage is required for feature caching
  4. Higher detection quality (mean average precision) than R-CNNs
  5. Comparable performance to SPPnet (immediate successor to R-CNNs)
- Generates region proposals on the convolutional feature maps instead of on the full image. This drastically reducing training time.

Fast R-CNN Architecture



* RoI: Region of Interest
* bbox: Bounding Box
* FC: Fully Connected

Duke

Information Sharing

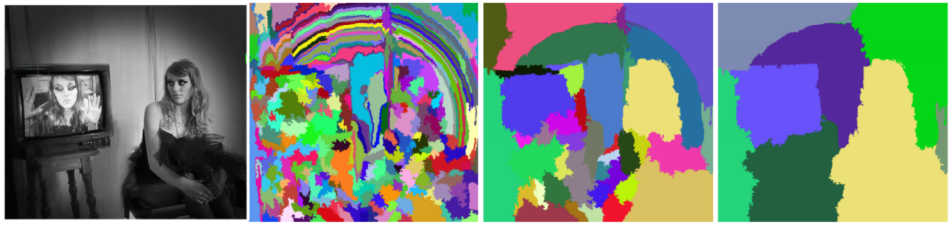
- Both the regressor and classifier share two fully connected layers.
- Information is shared between both parts of the models.
- Learning happens sequentially due to the multi-task loss function.
- Improves performance overall and makes softmax predictions marginally better the R-CNN's support vector machine with big gains in speed.

Fast R-CNN Benchmark Performance Gains

- Trains very deep VGG16 network 9x faster than R-CNN and is 213x faster at test time.
- 65.7% mAP on PASCAL VOC 2012 compared to 62% achieved by R-CNN.
- Performance in general is better than R-CNN and comparable to SPPnet but training and testing is much faster than both of these methods.

Drawbacks of Fast R-CNN

- Test Region Proposal is slow
  - Selective Search (2s/img)
  - EdgeBoxes (0.2s/img)



Input Image | Initial Segmentation | After some iterations | After more iterations

- Despite Fast-RCNN giving real-time object detection, obtaining region proposals in the beginning is a ‘bottleneck’

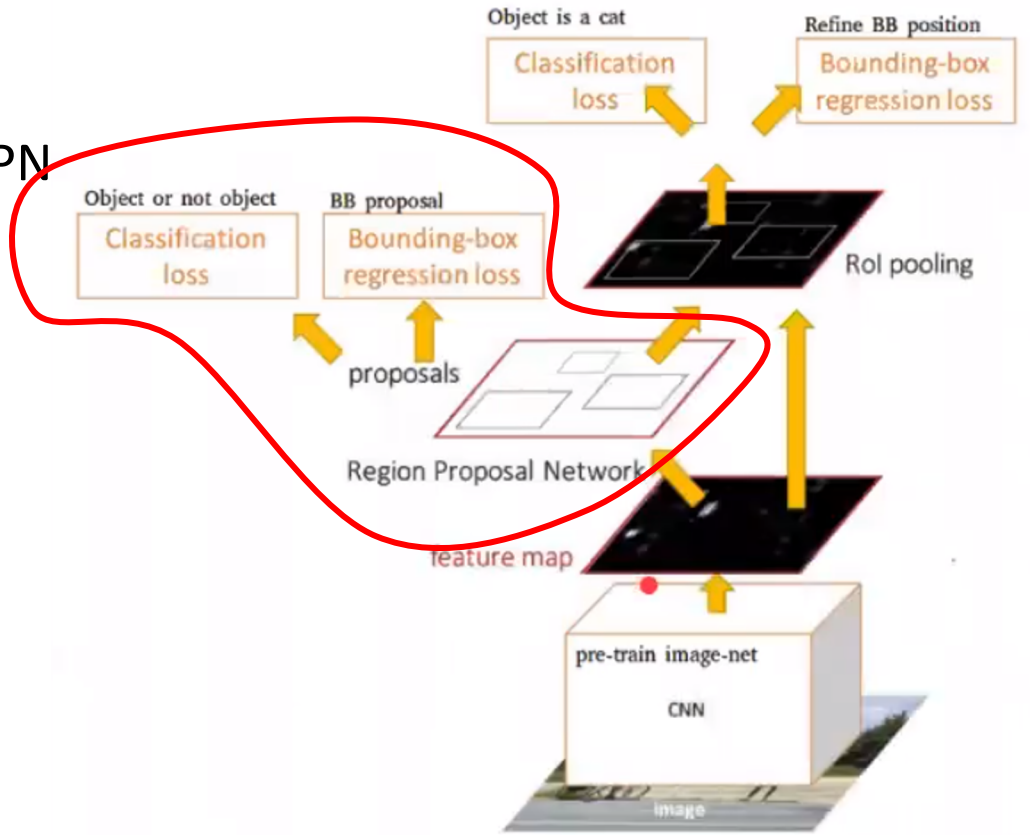
- Region Proposal Time ~ Detection Time

Duke

Faster R-CNN (2016)

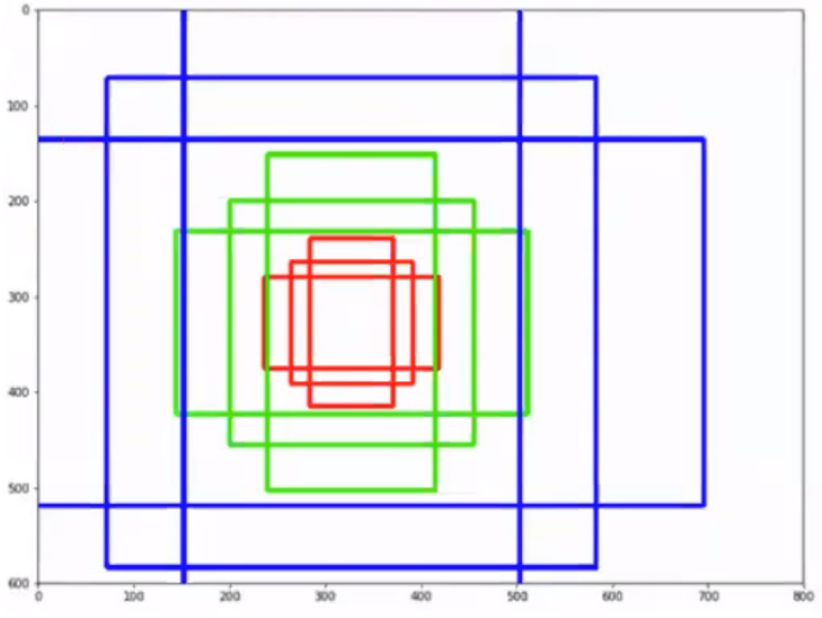
- Region Proposal Network (RPN)
  - **Initial Guess**
  - Fully Convolutional Network
  - Outputs :
    - Rectangular Object Proposal
    - Objectness Score

- **RPN + Fast R-CNN**



How does RPN Work?

- Anchors
  - **Initial Initial guess** (Input to RPN)
  - Anchors are shapes and aspect ratios we place at every pixel on image
  - They are not meant to be close to the ground truth bounding box. They are just a starting point **proposal.** Hence Region Proposal Network (RPN)



k = *9* anchor boxes per pixel
Total Anchor Boxes = W * H * k

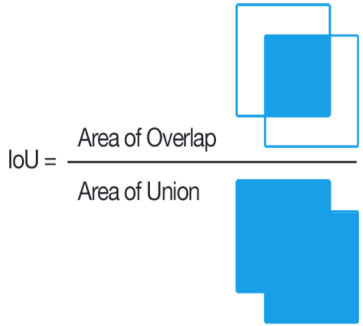
- **RPN prunes anchor boxes**

Duke

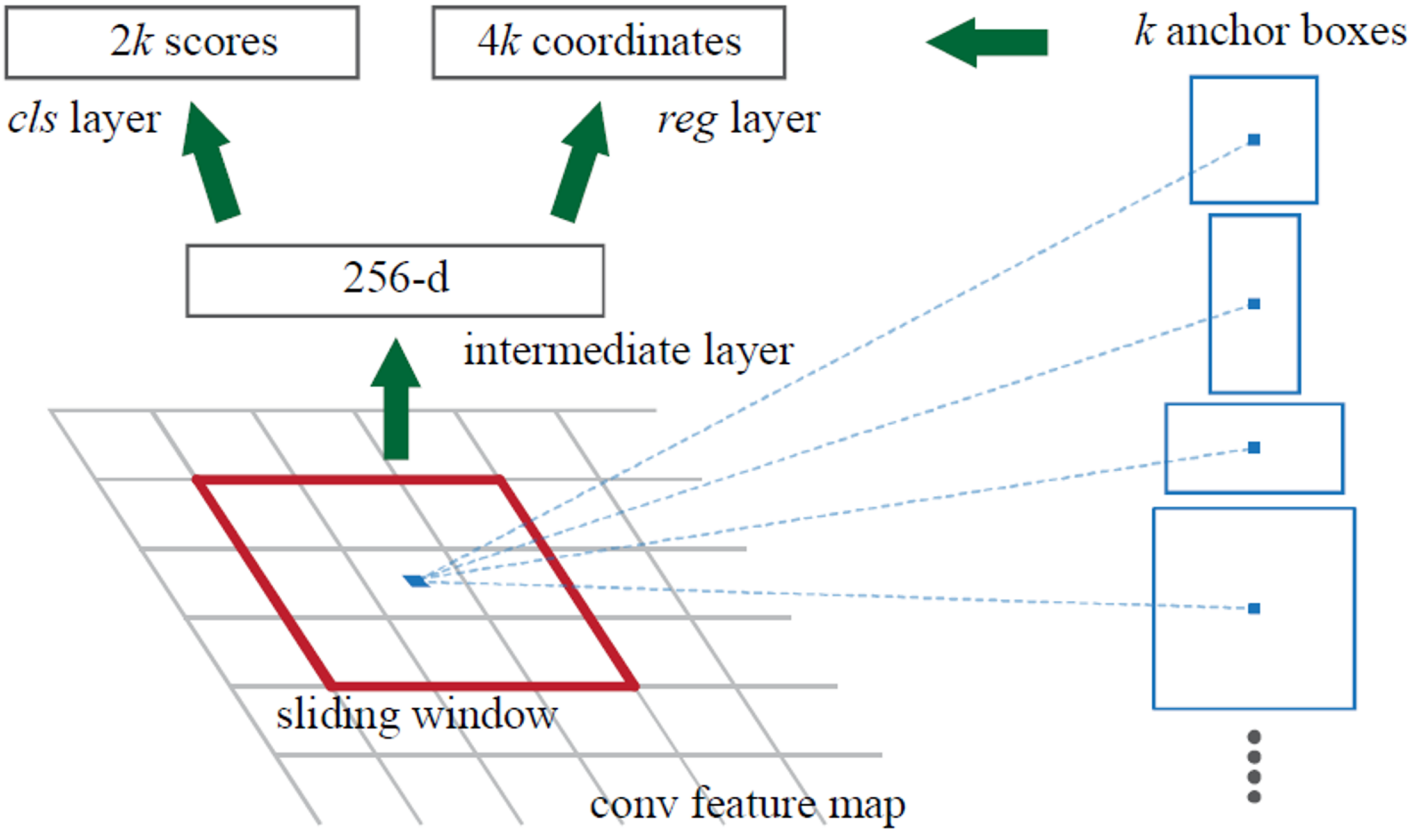
How does RPN Prune Anchor Boxes?

- Gives binary label to each anchor box **(Object or No Object)**
  - **Condition for positive label:**
    - Anchor with an IoU > 0.7 with the ground-truth box **or** anchor with the highest IoU (could be as low a 0.5).
  - **Condition for negative label:**
    - Anchor with an IoU < 0.3 with the ground-truth box

- Single ground truth box can give positive label to multiple anchor boxes



Duke


2k scores
4k coordinates
k anchor boxes
cls layer
reg layer
256-d
intermediate layer
sliding window
conv feature map

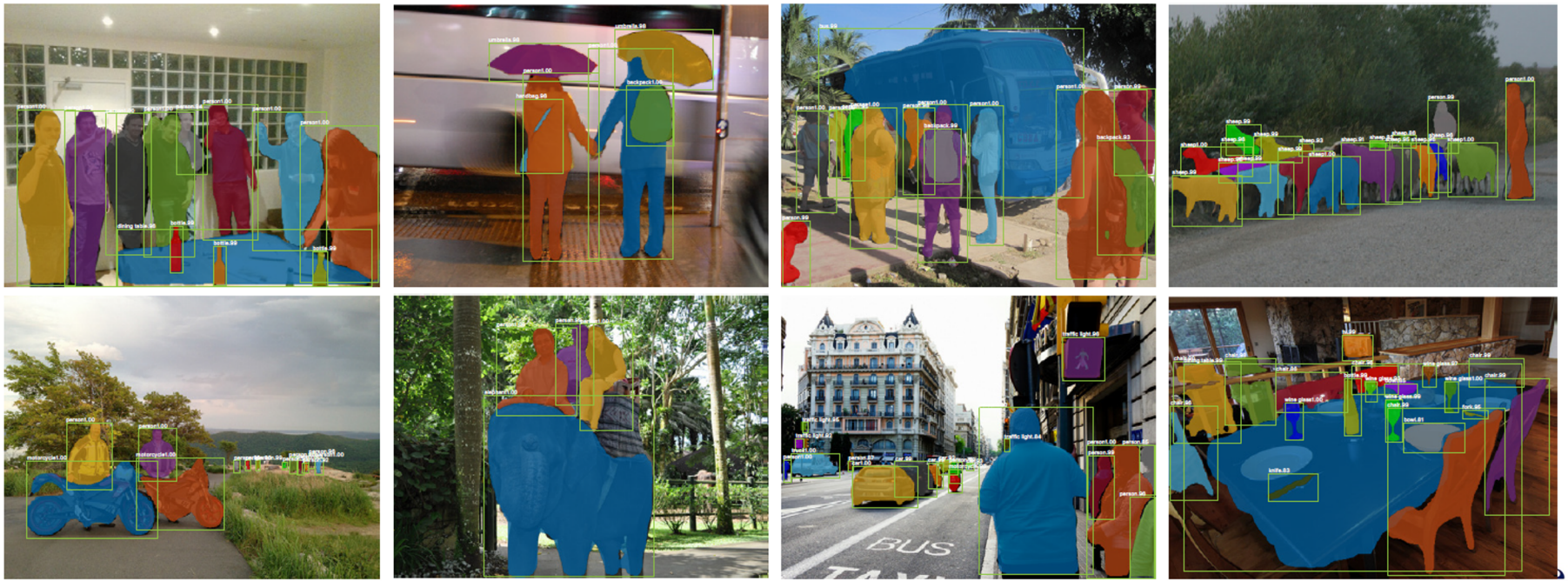
Selective Search (2s/img)

EdgeBoxes (0.2s/img)

RPN (0.01s/img)

Mask R-CNN (2018)

- Not viable for our problem



YOLO: You Only Look Once

Duke

Limitations of YOLO

- Lags behind state-of-the-art algorithms in terms of detection accuracy. Faster R-CNN has 73.2 mAP on PASCAL 2007 while YOLO has 63.4 mAP.
- Struggles to identify smaller objects (which may be an issue in terms of logo identification).
- Produces more localization errors than Fast R-CNN.
- Finds it difficult to detect objects of varying sizes.

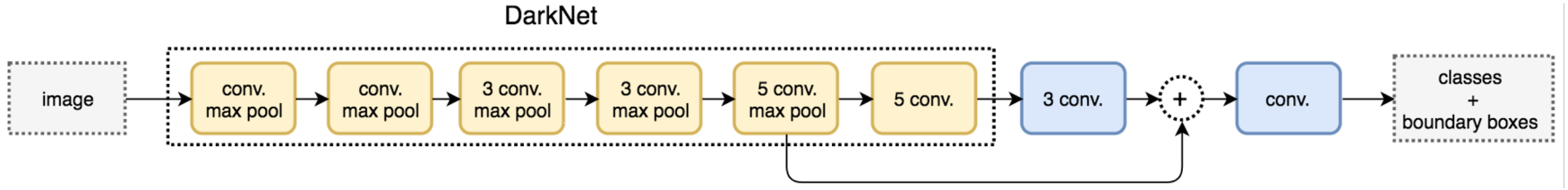
From YOLO to YOLO v2

- **Higher Resolution Classifier**: Train on ImageNet at 224x224 but resize to 448x448 for additional training. +3.5% mAP.
- **Anchor Boxes**: use k-means to generate dimension clusters. Predict location coordinates (instead of offsets) from dimension clusters rather than hand picked anchor boxes. +5% mAP.
- **Multiscale Training**: training at different aspect ratios. +1.5% mAP.
- **Darknet 19**: maintains YOLO's speed but improves mAP. +0.4% mAP.
- **Batch Normalization**: stabalizes training. +2% mAP.
- **YOLO9000**: training is done jointly for detection and classification using both COCO and Imagenet. Model learns to predict novel detection labels due to its hierarchical structure.

Duke

YOLO v2 Architecture

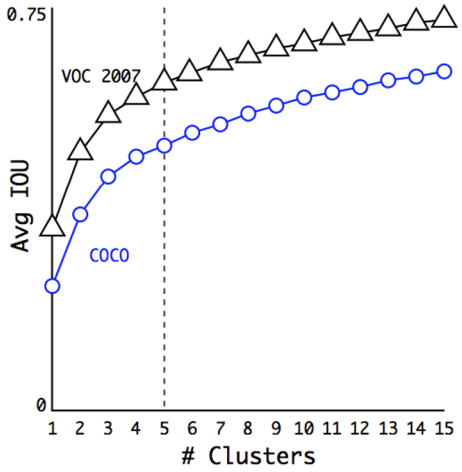
- Fully Convolutional.
- Moves class prediction from cell level to boundary box level.
- Removes one max pooling layer to make spatial output of the network 13x13 instead of 7x7.
- Uses passthrough to help with small object detection.

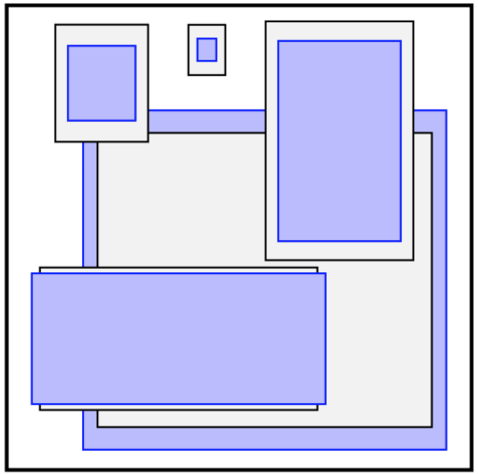


Duke

Anchor Boxes Using K-Means Clustering

- Cluster the bounding boxes in the training data using K-Means.
- Use the clusters to obtain K anchors.
- Predict location coordinates instead of offsets.
- Use custom distance metric to ensure bounding box size does not affect clusters.
- YOLO v2 goes with 5 clusters to balance Avg IOU and prediction time.





Duke

PASCAL VOC 2007 Performance

Detection Frameworks	Train	mAP	FPS
Fast R-CNN [5]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[15]	2007+2012	73.2	7
Faster R-CNN ResNet[6]	2007+2012	76.4	5
YOLO [14]	2007+2012	63.4	45
SSD300 [11]	2007+2012	74.3	46
SSD500 [11]	2007+2012	76.8	19
YOLOv2 288 × 288	2007+2012	69.0	91
YOLOv2 352 × 352	2007+2012	73.7	81
YOLOv2 416 × 416	2007+2012	76.8	67
YOLOv2 480 × 480	2007+2012	77.8	59
YOLOv2 544 × 544	2007+2012	**78.6**	40

Duke