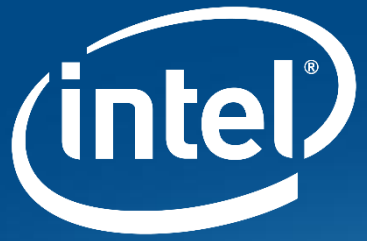


DL Object Detectors

Alexander Kozlov

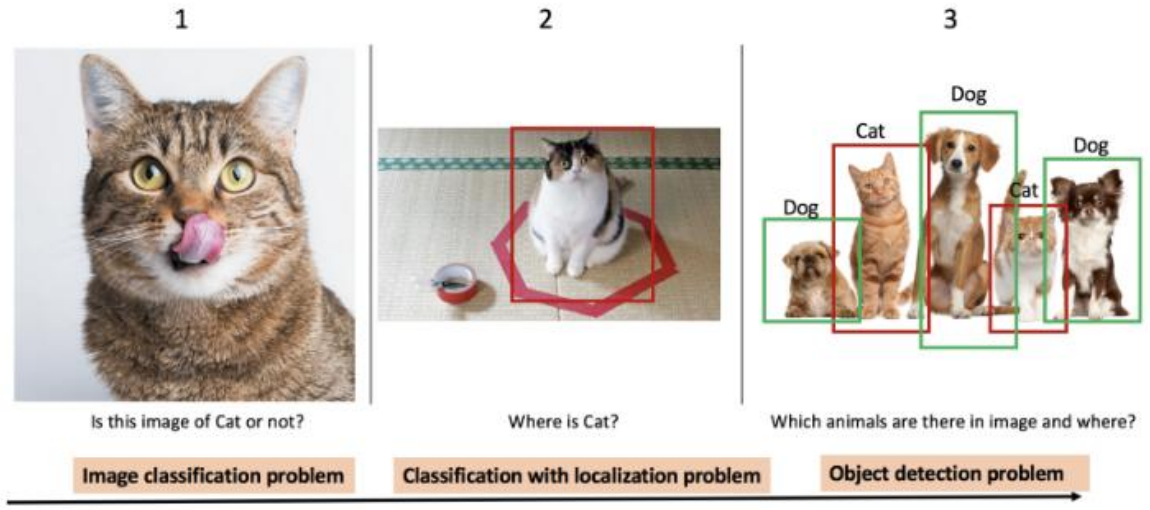
Internet of Things Group

Agenda

- Problem definition
- Classical approaches
- Deep Learning approaches

Problem Definition

- Find objects on the image
  - Coordinates of bounding boxes
  - Class probabilities
- Basically used in conjunction w/ other algorithms



Metric

- Intersection over Union (IoU) of two bounding boxes can be computed as:

`(a & b).area() / (a.area() + b.area() - (a & b).area())`, not `(a & b).area() / (a | b).area()`

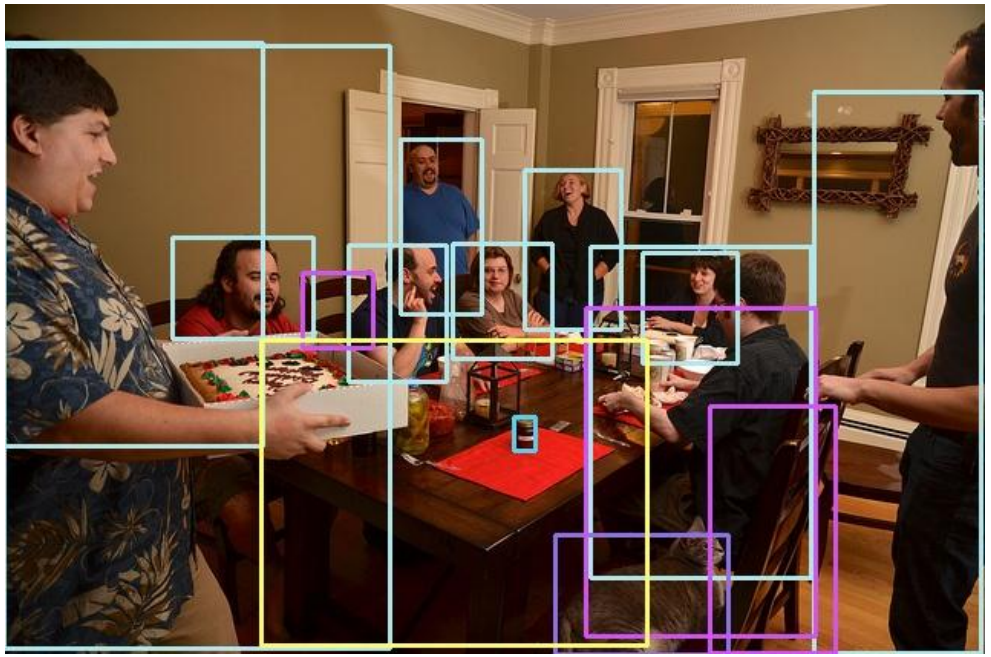
- Average precision

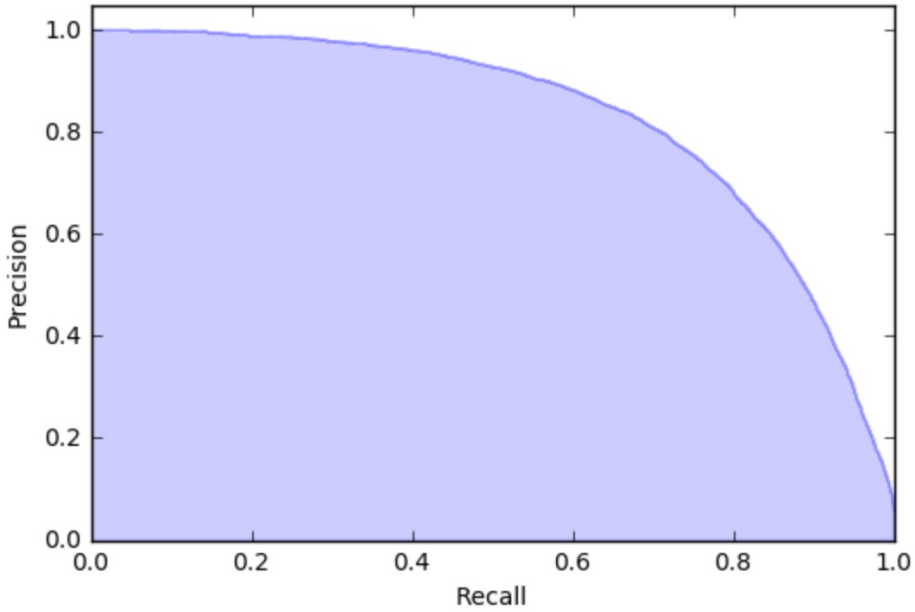
TP = True positive
TN = True negative
FP = False positive
FN = False negative

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

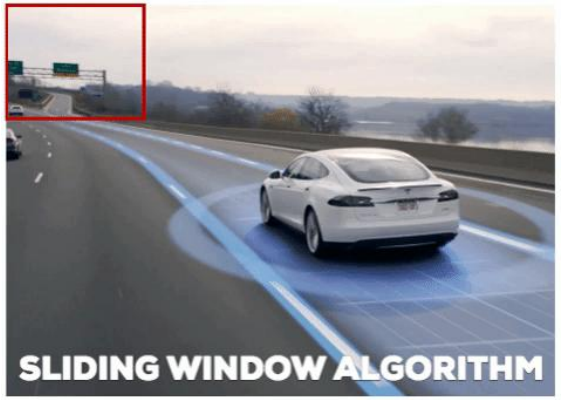
$$AP = \frac{1}{11} \times \left(AP_r(0) + AP_r(0.1) + \ldots + AP_r(1.0)\right)$$



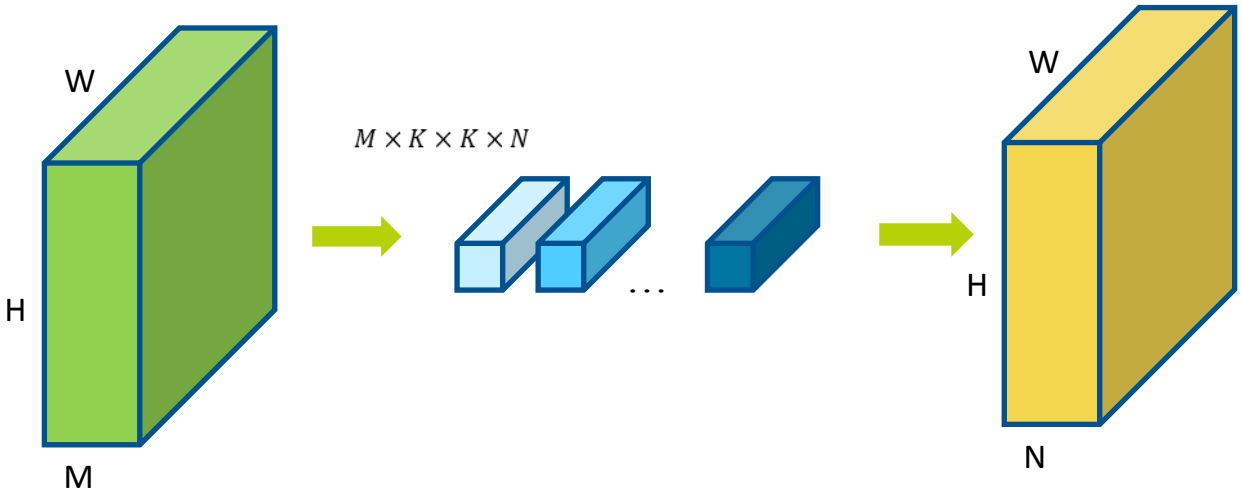


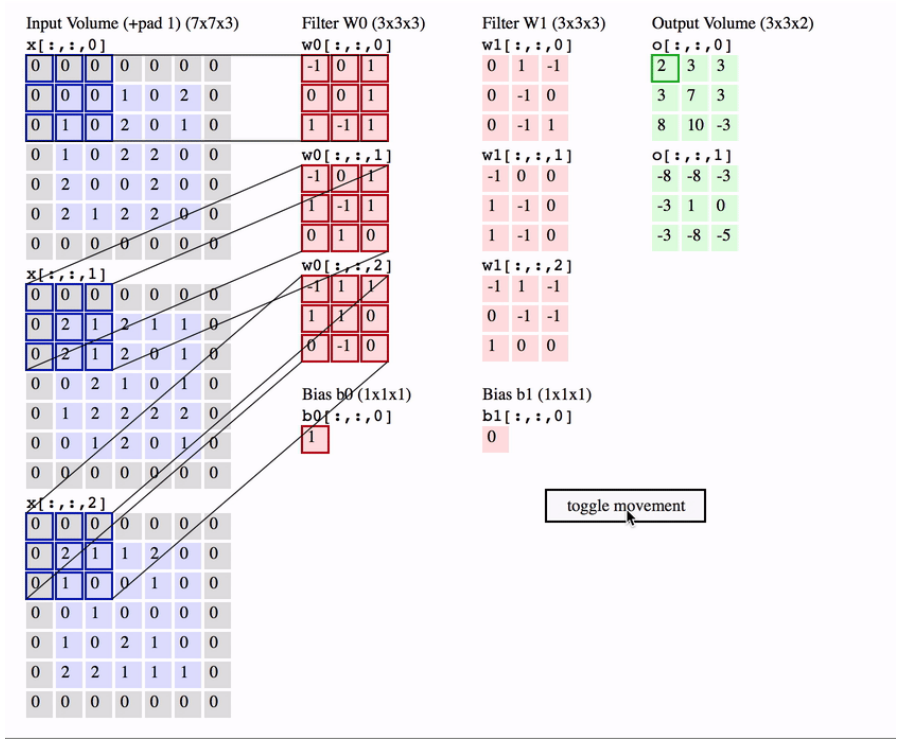
Classical methods

- Hand-crafted features: LBP, HOG, SIFT, etc.
- Sliding window approach
- Pyramid of images (features)
- Classifier: SVM, Adaboost, Random Forest, etc.

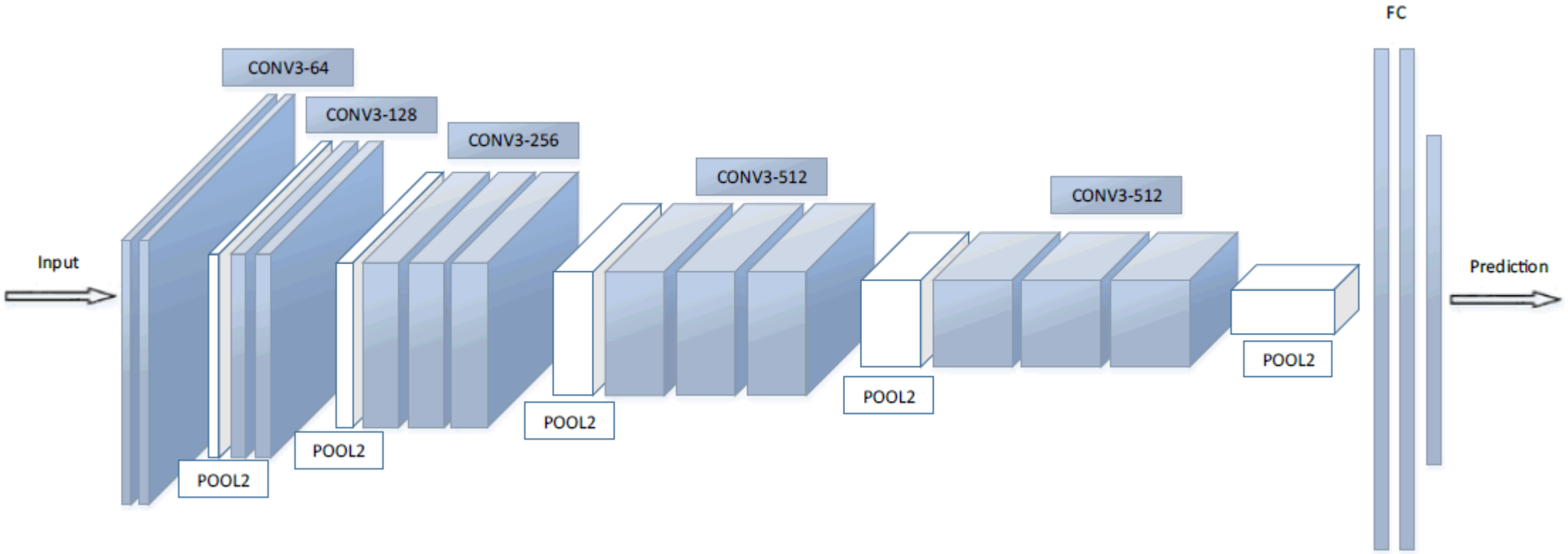


Convolution



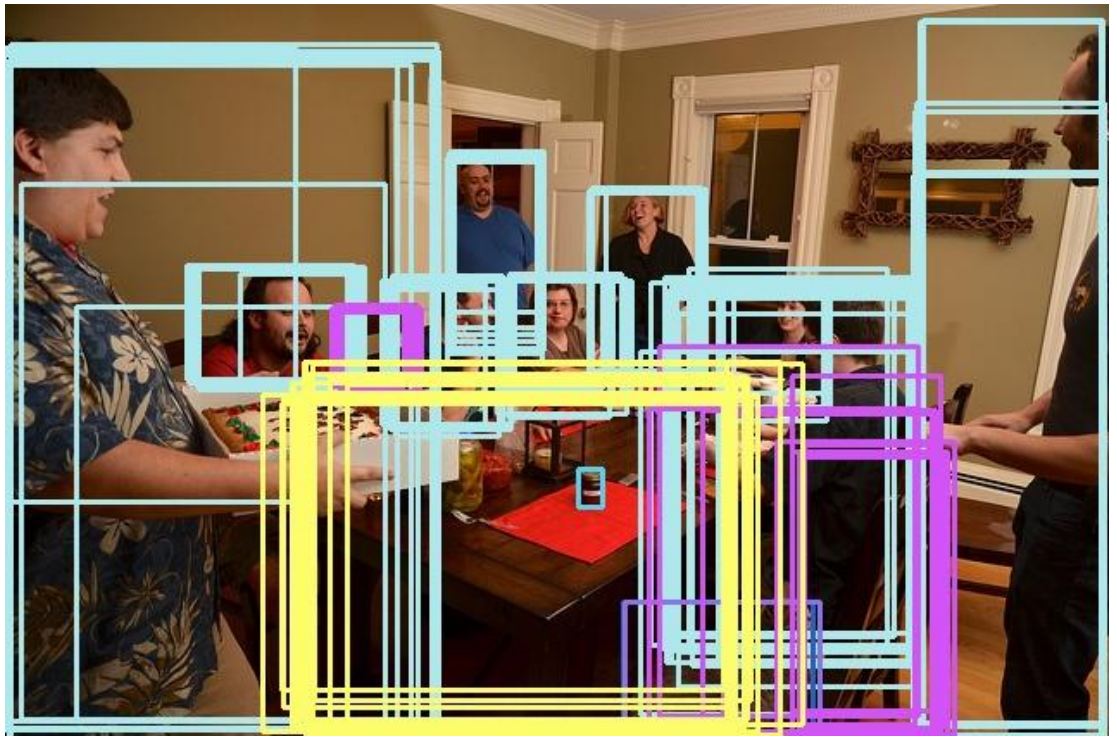


VGG CNN


FC
CONV3-64
CONV3-128
CONV3-256
CONV3-512
CONV3-512
Input
Prediction
POOL2
POOL2
POOL2
POOL2
POOL2

Non-Maximum Suppression (NMS)

- Detector outputs multiple detections for the same object
- NMS discards all except one with best features, e.g. with highest score
- Can be learnable



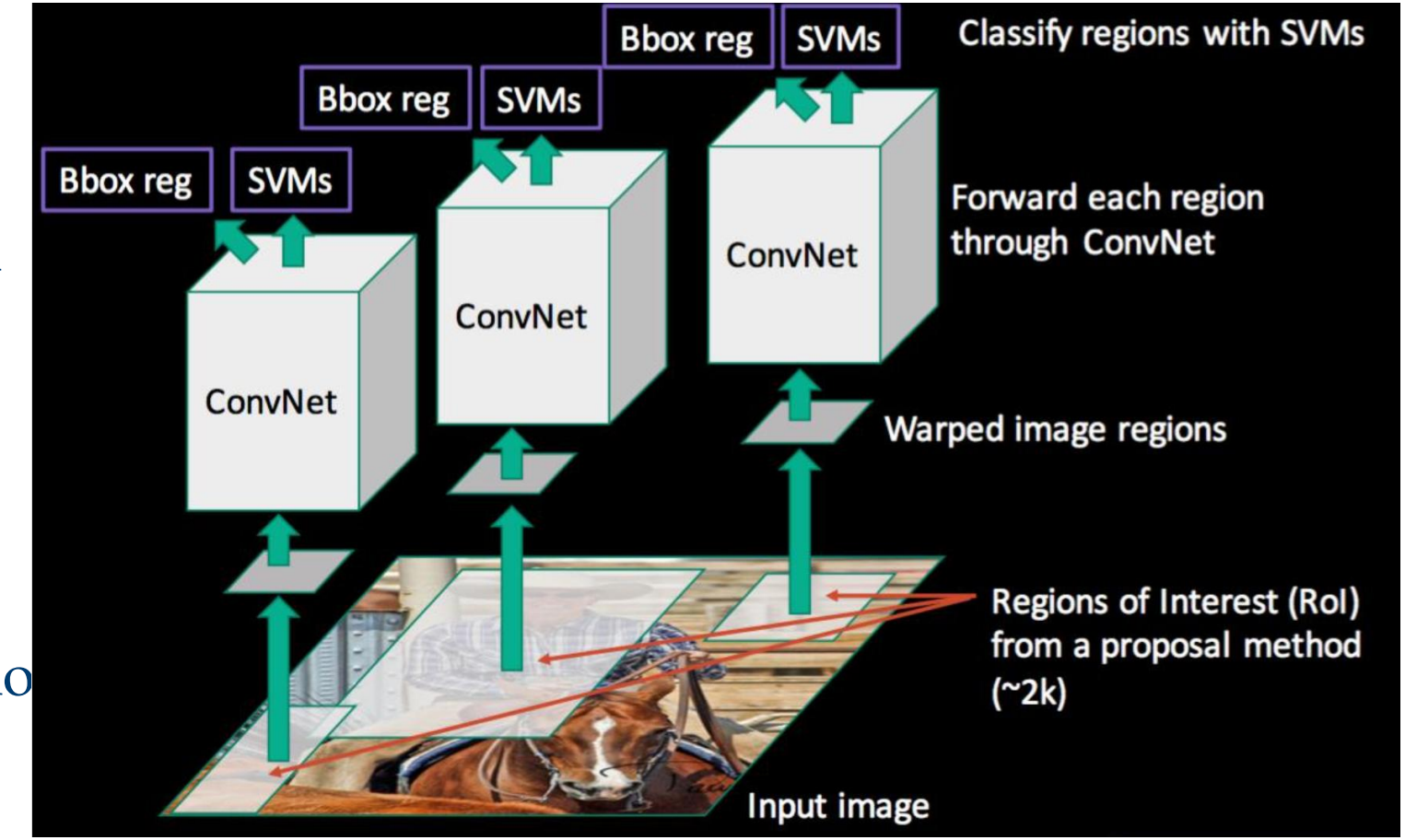
R(egion based)-CNN (CVPR, 2014)

Motivaion:

- Deep learning-based classifiers are known to be powerful but quite slow
  - Operating in a sliding window fashion seems to be very slow
  - Good proposal generation stage can potentially resolve the issue

R(egion based)-CNN (CVPR, 2014)

- Region proposal algorithm (Selective Search) is used to get RoIs
- Off-the-shelf image classification net (AlexNet) is used to extract features for every RoI
- SVM classifier is used to classify RoIs as objects or background
- Linear regression is applied to localize bounding boxes inside Ro



Fast R-CNN (2015)

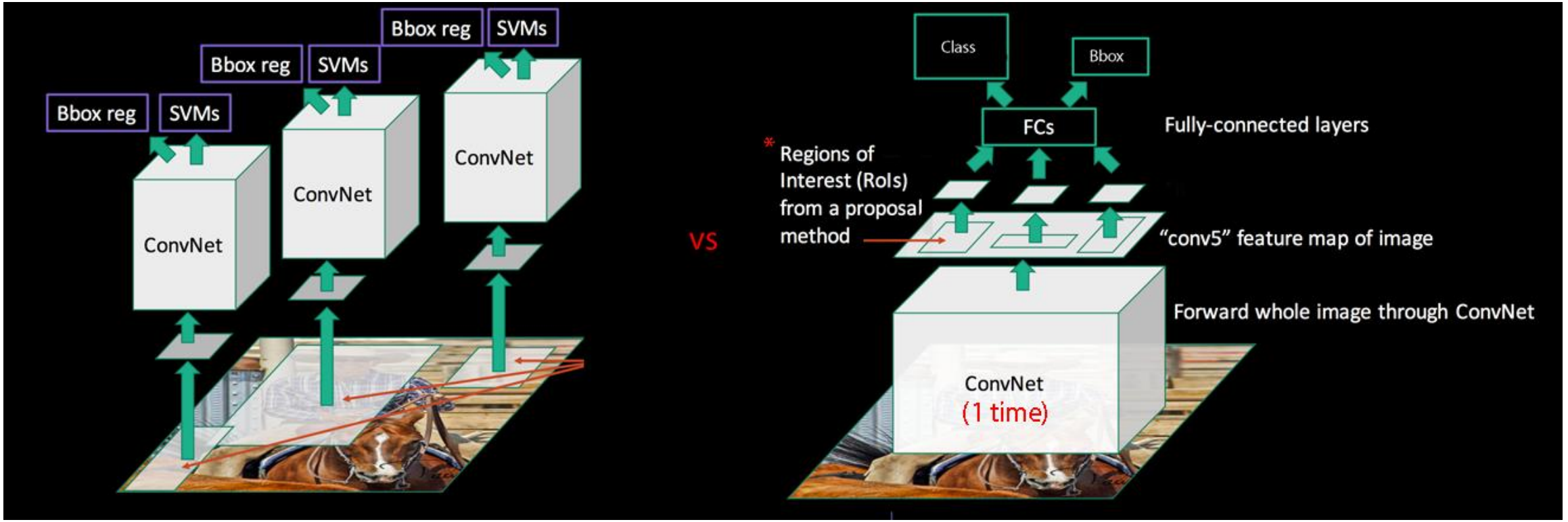
Drawbacks of R-CNN:

- Complicated multi-stage model is hard to train
- Despite the use of Selective Search is slow at test time

Solution – Fast R-CNN:

- Merge classifier and regressor to the convnet itself to train it end-to-end
- Apply convnet to the whole image and crop RoIs on high-level feature map to make detection faster

Fast R-CNN (2015)



Deep features compute once per image, not per proposal

Faster R-CNN

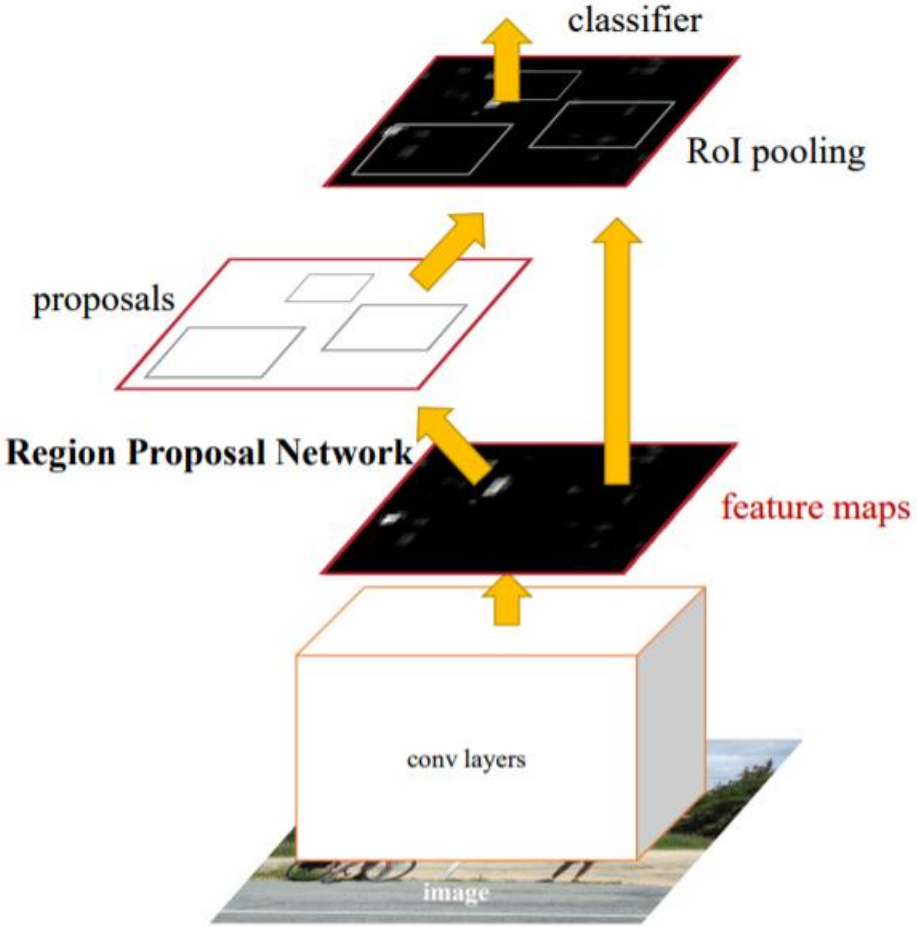
Drawbacks of Fast R-CNN:

- 25x faster then R-CNN, but still too slow. Mostly because of the Selective Search now (~2s per VGA image)

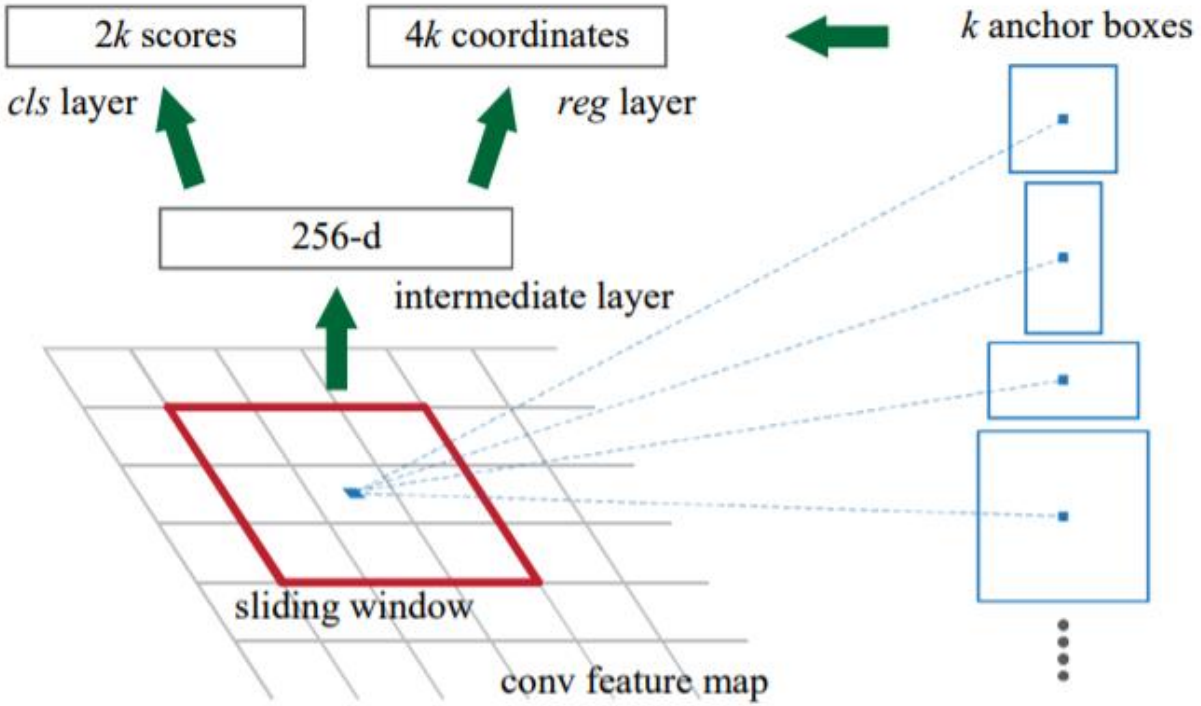
Solution – Faster R-CNN:

- Merge region proposal stage into the net as well

Faster R-CNN (2015)

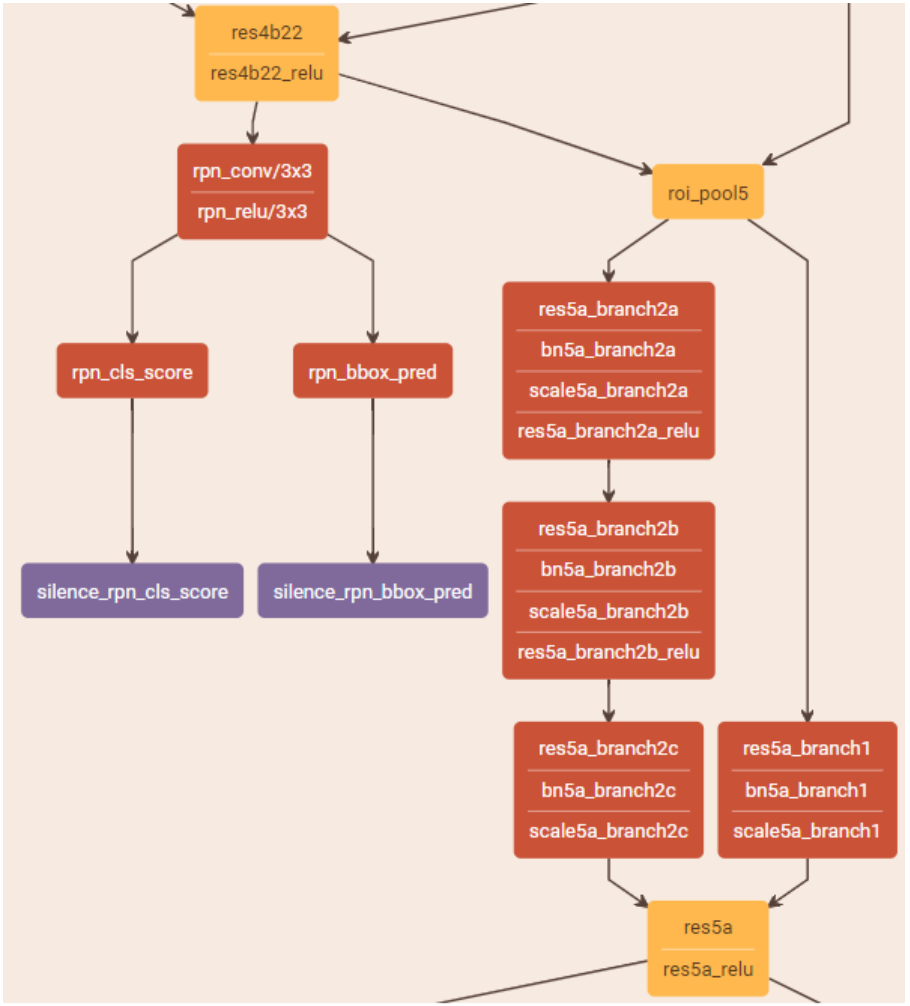


Faster R-CNN pipeline

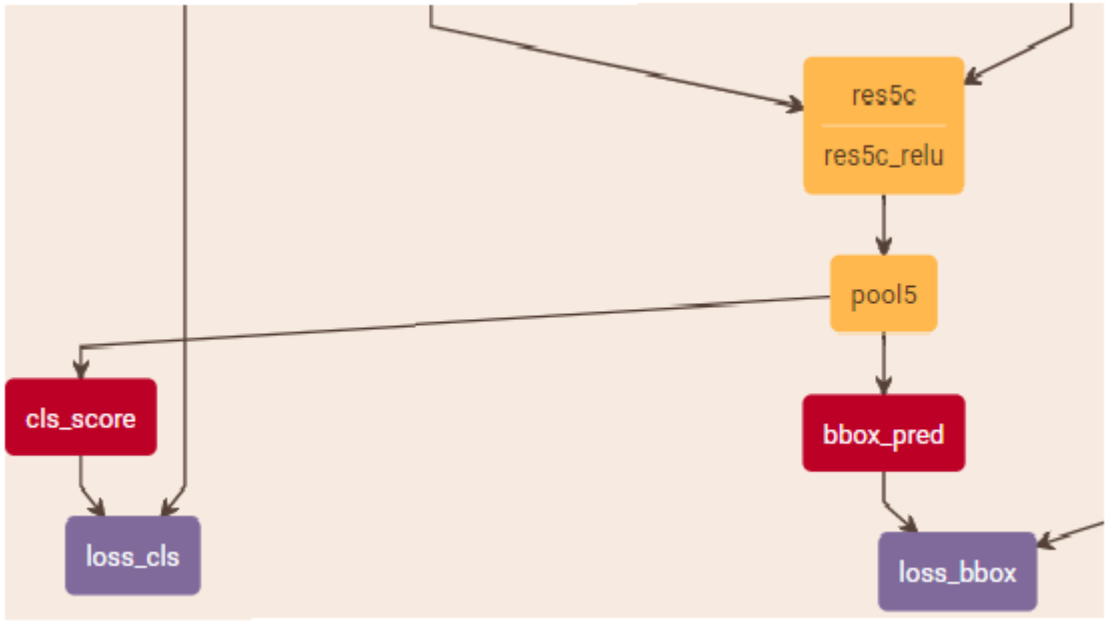


Region Proposals Network

Faster R-CNN With Resnet 101 Example



• • •



R-FCN

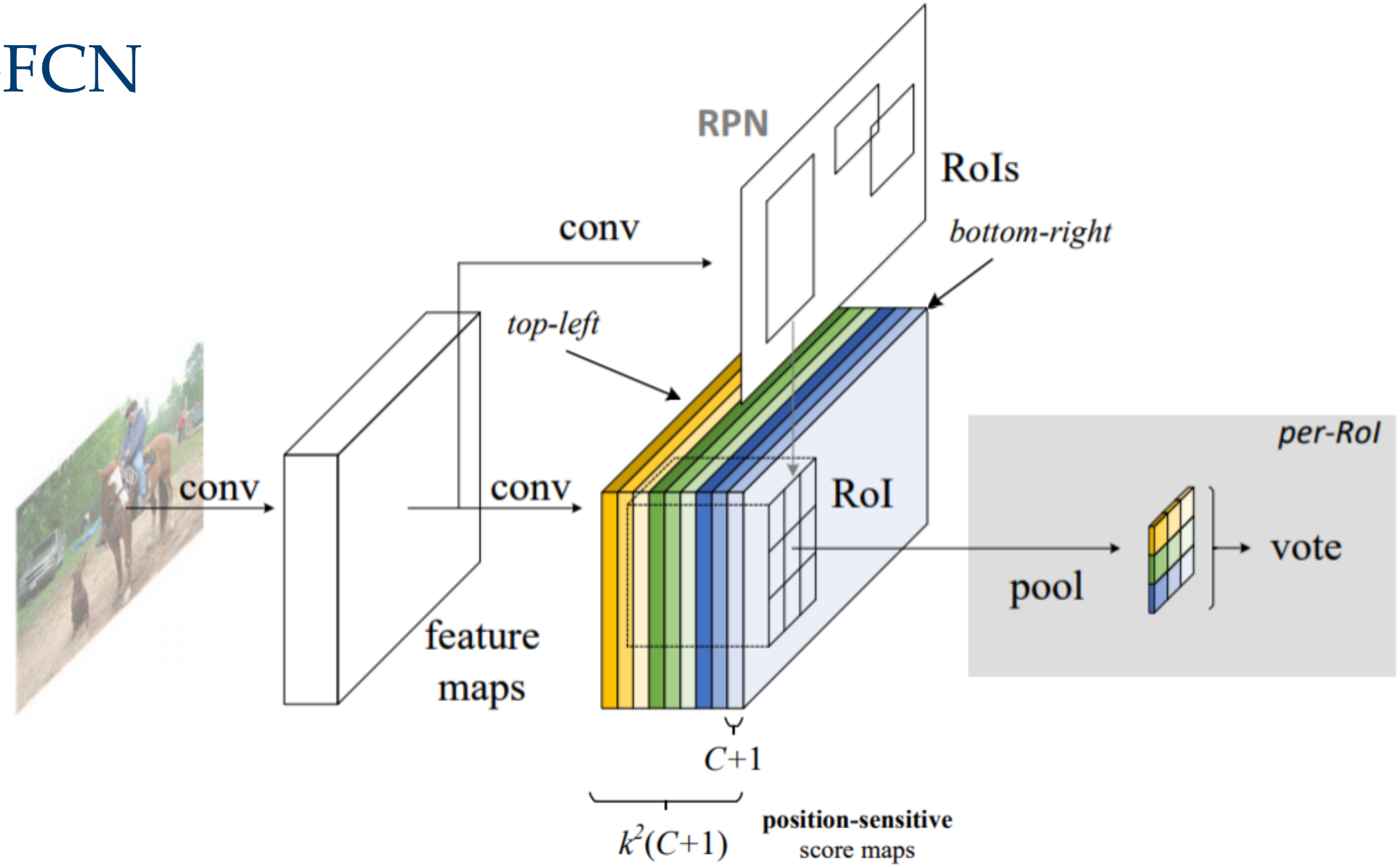
Motivation:

- Made object detection network with 100% shared computations
- Allow translation variance to localize object position

Methodologies of *region-based* detectors using **ResNet-101**

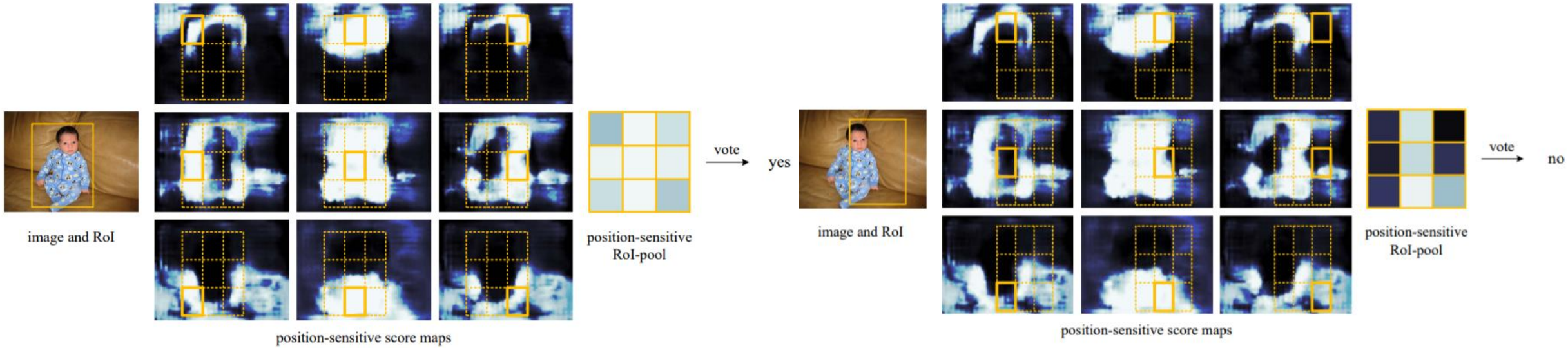
	R-CNN [7]	Faster R-CNN [19, 9]	R-FCN [ours]
depth of shared convolutional subnetwork	0	91	101
depth of RoI-wise subnetwork	101	10	**0**

R-FCN


RPN
RoIs
conv
bottom-right
top-left
conv
conv
RoI
per-RoI
pool
vote
feature
maps
C+1
k²(C+1)
position-sensitive
score maps

R-FCN

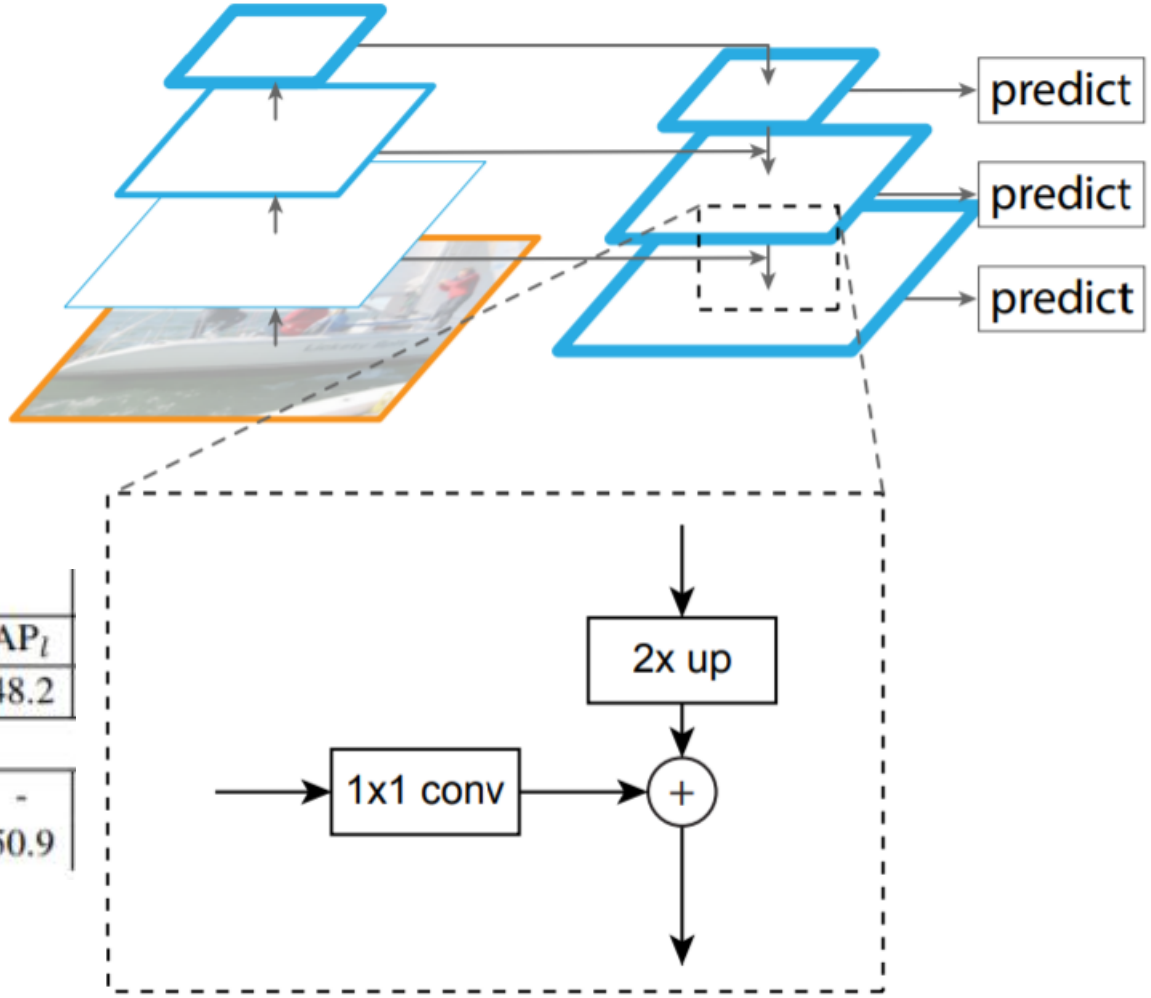
Position-sensitive score maps



Feature Pyramid Network (FPN)

Motivation:

- Leverage the pyramidal shape of a ConvNet's feature hierarchy
- Provide strong semantics at all scales

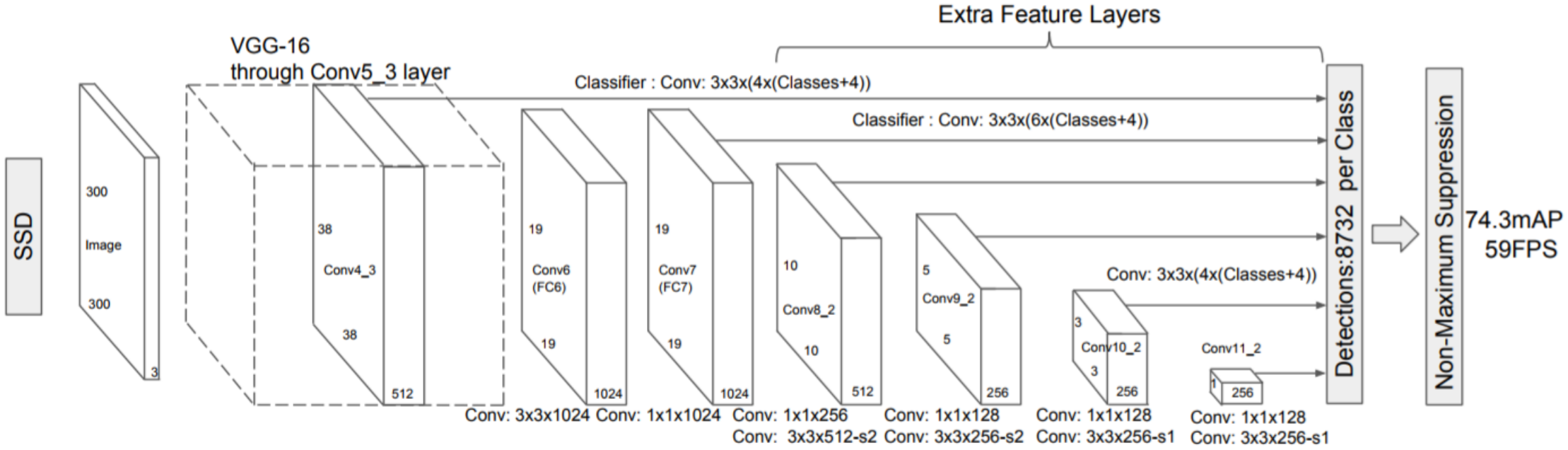


method	backbone	competition	test-dev AP	AP_s	AP_m	AP_l
ours, Faster R-CNN on **FPN**	ResNet-101	-	**36.2**	**18.2**	**39.0**	48.2
*Competition-winning **single-model** results follow:*						
G-RMI†	Inception-ResNet	2016	34.7	-	-	-
Faster R-CNN +++	ResNet-101	2015	34.9	15.6	38.7	50.9

Single Shot Multibox Detector

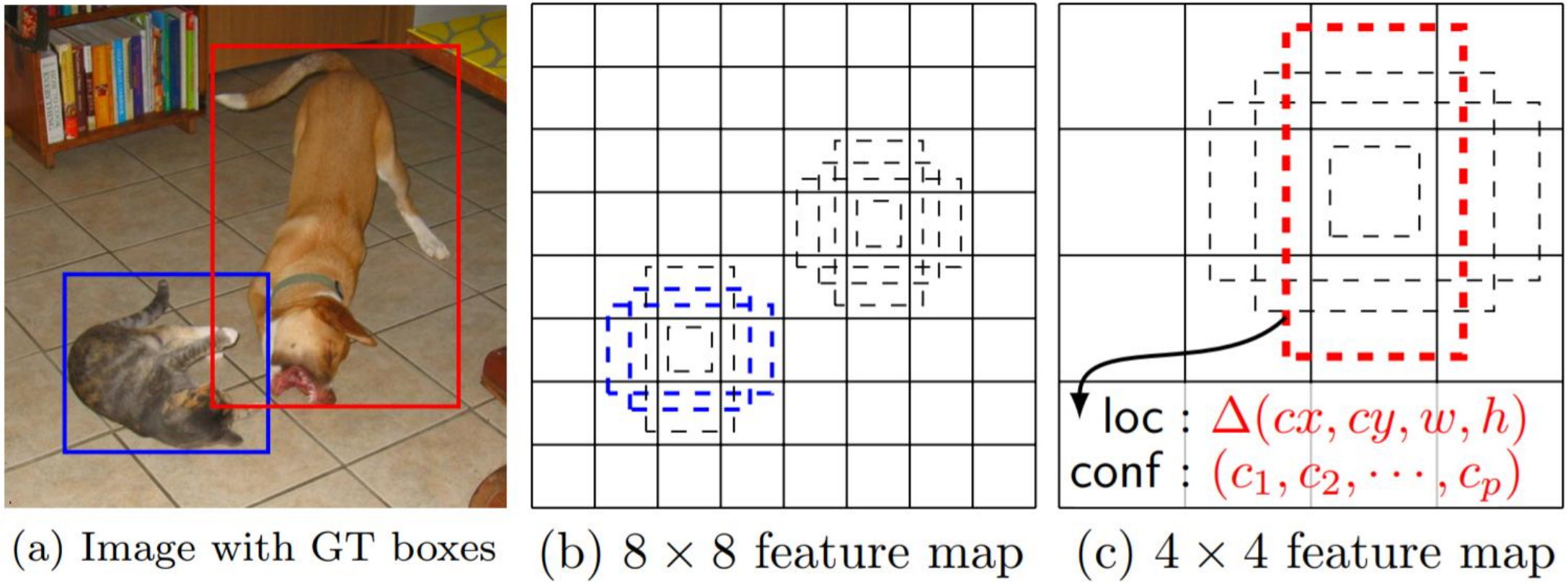
Motivation:

- Region proposal net can be eliminated at all
- Make multi-scale detection efficient



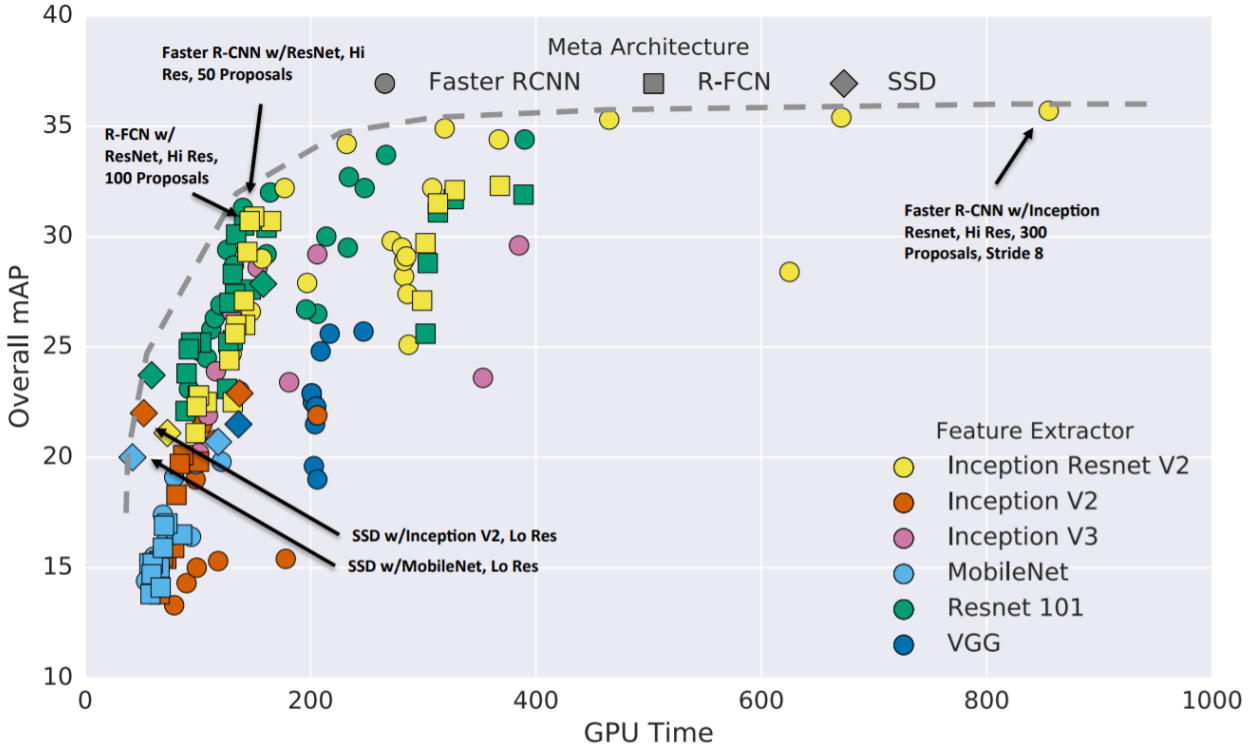
Single Shot Multibox Detector

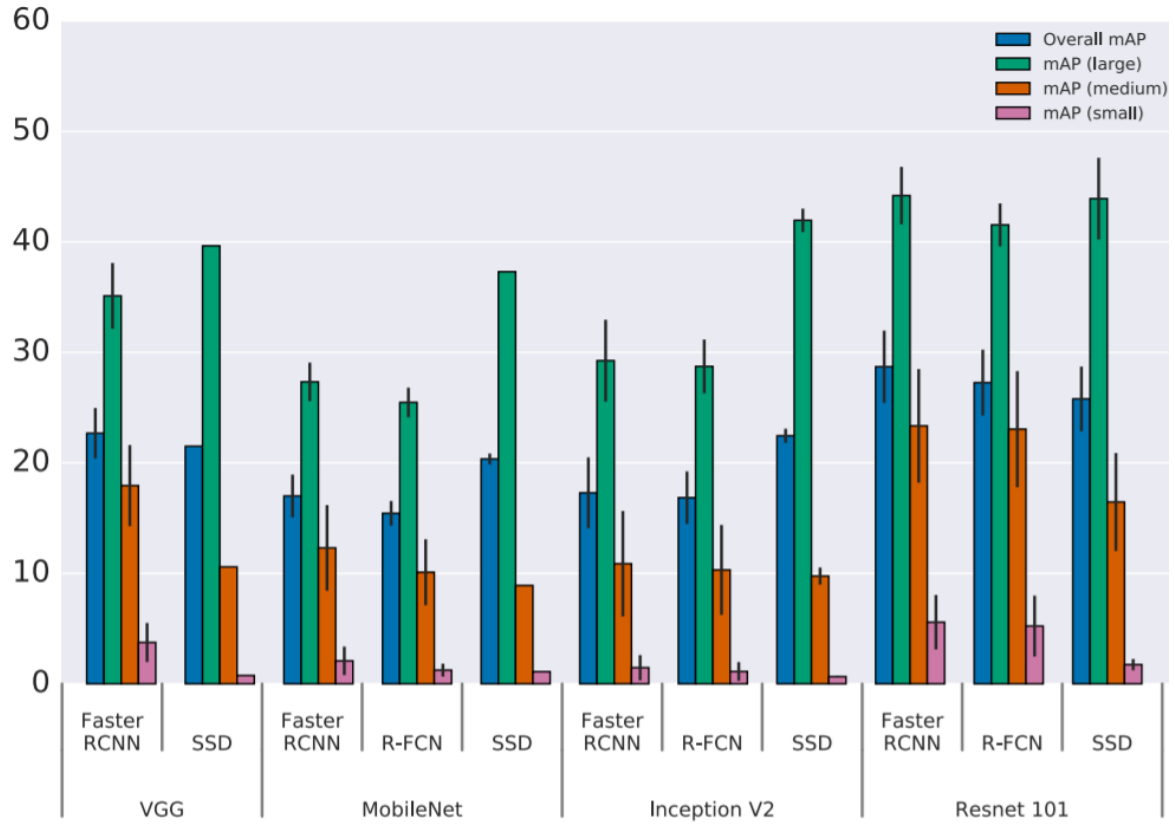
Default boxes



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

Detectors Speed-Accuracy trade-offs





Transfer Learning

Very few people train an entire Convolutional Network from scratch (with random initialization)

It is common to pre-train a ConvNet on a very large dataset (e.g. ImageNet), and then use the ConvNet for the task of interest, major scenarios:

- ConvNet as fixed feature extractor
  - Take a ConvNet pre-trained on ImageNet, remove the classification layer (which outputs the 1000 class scores), then treat the rest of the ConvNet as a fixed feature extractor for the new dataset
- Fine-tuning the ConvNet
  - Not only replace and retrain the classifier on top of the ConvNet on the new dataset, but to also fine-tune the weights of the pre-trained network
- Pre-trained models
  - Use one of the uploaded pre-trained models for own task

Recent state-of-the-art CNNs

- RetinaNet
- RefineDet
- Mask R-CNN (+instance segmentation)
- Cascade R-CNN
- Deformable Convolution Networks

Popular OD repositories

- Tensorflow Object Detection API: https://github.com/tensorflow/models/tree/master/research/object_detection
- Detectron (Caffe2): https://github.com/facebookresearch/Detectron
- mmdetection (PyTorch): https://github.com/open-mmlab/mmdetection

OpenVINO OD models

- OpenVINO: https://github.com/opencv/dldt
- Open Model Zoo: https://github.com/opencv/open_model_zoo
  - Face detection (2 variants)
  - Person detection (2 variants)
  - Vehicle detection (3 variants)
  - Public models: SSD, Faster-RCNN, R-FCN, etc.

Q&A