Advanced Methods in Object Detection

Kai Yi
Software Institute
Xi'an Jiaotong University
24th Aug, 2017

Outline

- Introduction to Object Detection
- Definition, Main Difficulties, Typical Traditional Methods.
- Mainstream Object Detection Methods
- Object Proposal Methods
- Others (mentioned latter)
- State-of-art Methods in 2D Object Detection
- R-CNNs
- (R-CNN, SPP-Net, Fast R-CNN, Faster R-CNN, YOLO(2), SSD)
- RRC
- SqueezeDet
- Object Detection From 2D to 3D
- Main Difference, Main Challenge, et al.
- State-of-art Methods in 3D Obejct Detection
- MVOD
- MV3D
- Others (mentioned latter)
- Create a real-time methods without decreasing the mAP
- Basic structure about my own method.

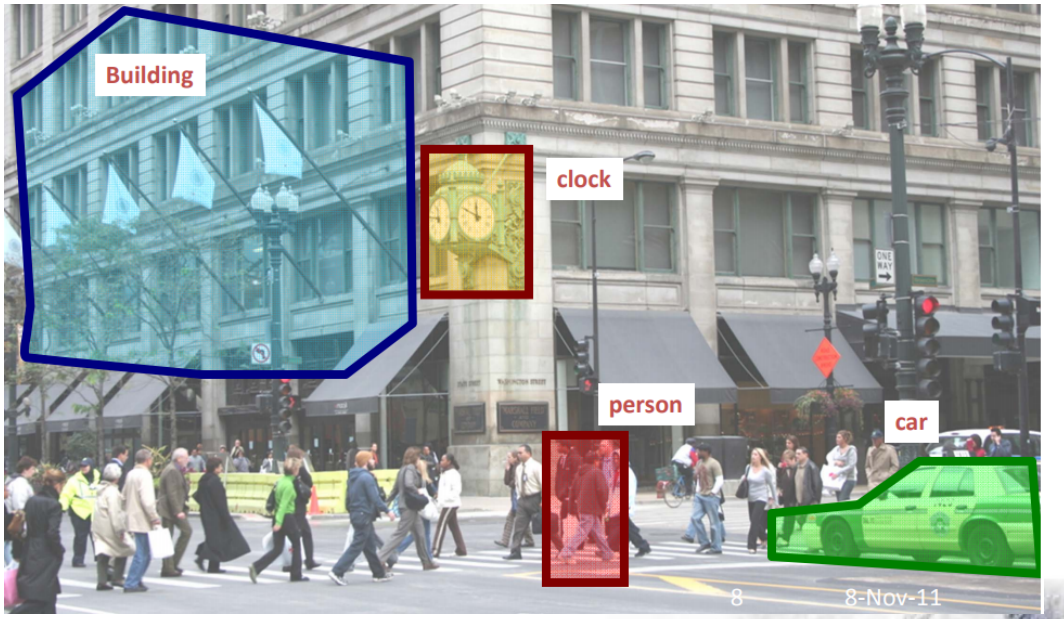
Part I:
Introduction to Object Detection

- Basic Concepts
- Main Difficulties
- Historical Methods

Basic Concepts

- Diff between classification and dete:
  - Main Diff: Accurate Localization
  - Classification: Tell us whether there is an particular object in an image
  - Detection: Whether there is A(B, C...) and where are they.

Fei-Fei Nov 2011. CS231A



Main Difficulties

Viewpoint variation

Scale variation

Deformation

Occlusion

Illumination conditions

Background clutter

Intra-class variation

Typical Historical Methods

Sliding Windows

1. Implementation

Encode a fixed size window and a fixed object. Then trying to compare the similarity between the slide and the model according to a probability model or something else.

2. Strength

Easy to implement and it can search all the object in an image in theory.

3. Weakness

High computational method.

Typical Historical Methods

Other Methods – Remains

HOG, Dense SIFT, Bag-of-Words, SPM et al.

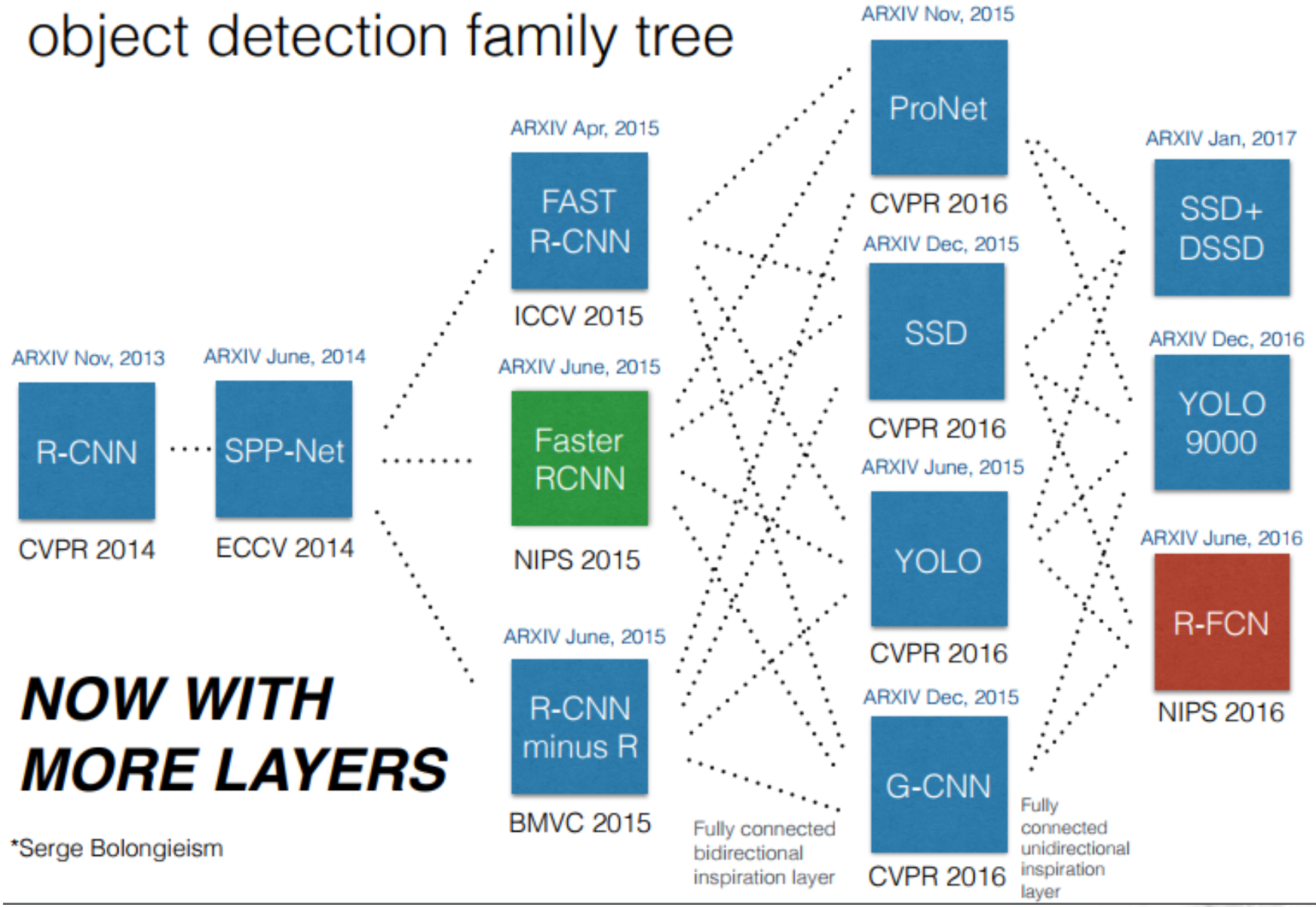
Part II:
Mainstream Object Detection Methods

- Region-based Methods(RCNNs)
- Single Path Methods(YOLO/SSD)
- Other Methods(DRL)

Part III:
State-of-art Methods in 2D Object Detection

- Faster R-CNN
- YOLO(2)
- SSD(DSSD)
- R-FCN
- RRC
- SqueezeDet

Some members of the *postdeepluvian** object detection family tree



NOW WITH MORE LAYERS

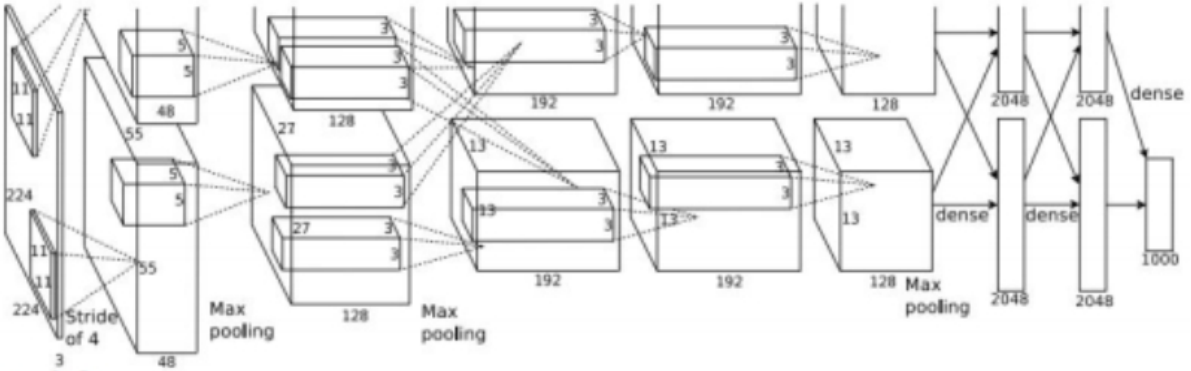
*Serge Bolongieism

CNN-Based Network Architectures

AlexNet
GoogleNet
VGG-Net
ResNet
SqueezeNet

Case Study: AlexNet

[Krizhevsky et al. 2012]



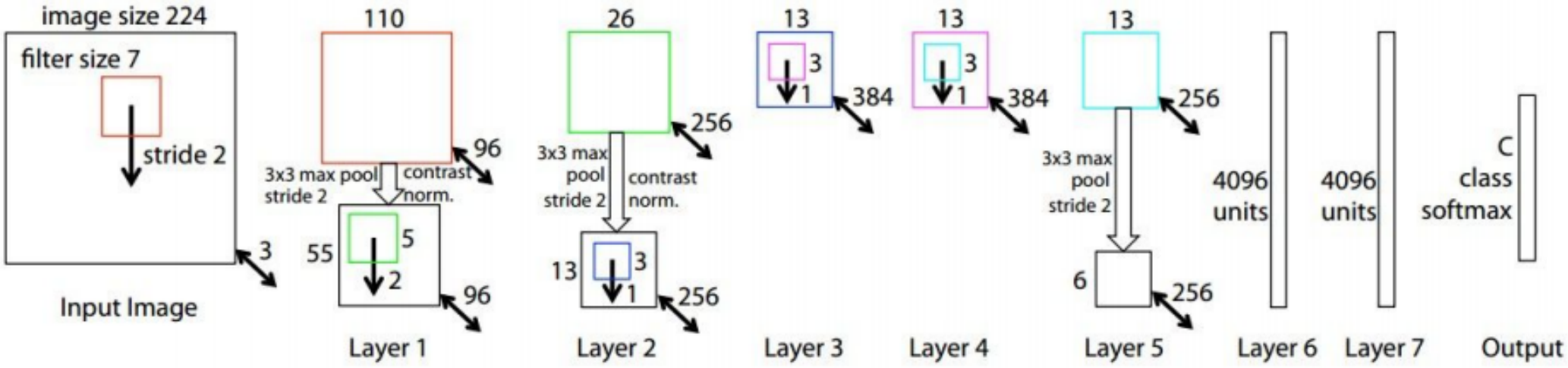
Full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

Details/Retrospectives:
- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

Fei-Fei et al. CS231N – Lecture 9

ZFNet

[Zeiler and Fergus, 2013]



TODO: remake figure

AlexNet but:
CONV1: change from (11x11 stride 4) to (7x7 stride 2)
CONV3,4,5: instead of 384, 384, 256 filters use 512, 1024, 512

ImageNet top 5 error: 16.4% -> 11.7%

Fei-Fei et al. CS231N – Lecture 9

Case Study: VGGNet

[Simonyan and Zisserman, 2014]

Small filters, Deeper networks

8 layers (AlexNet)
-> 16 - 19 layers (VGG16Net)

Only 3x3 CONV stride 1, pad 1
and 2x2 MAX POOL stride 2

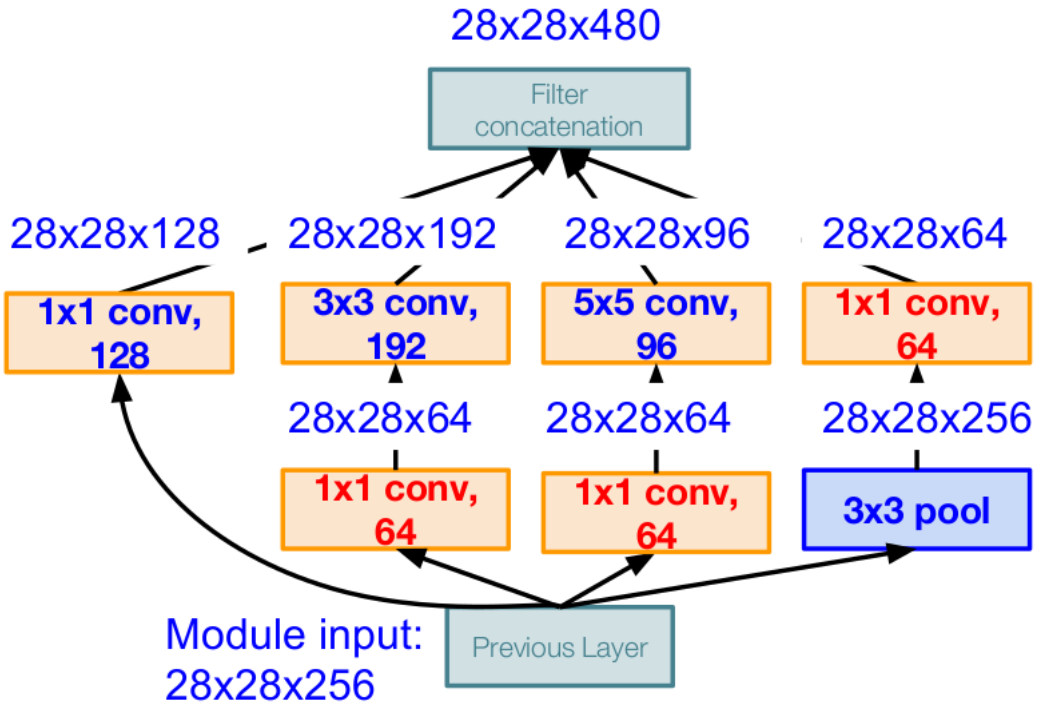
11.7% top 5 error in ILSVRC'13
(ZFNet)
-> 7.3% top 5 error in ILSVRC'14

------VGG-Net16

Fei-Fei et al. CS231N – Lecture 9

Case Study: GoogLeNet

[Szegedy et al., 2014]



Inception module with dimension reduction

Using same parallel layers as naive example, and adding "1x1 conv, 64 filter" bottlenecks:

Conv Ops:
[1x1 conv, 64] 28x28x64x1x1x256
[1x1 conv, 64] 28x28x64x1x1x256
[1x1 conv, 128] 28x28x128x1x1x256
[3x3 conv, 192] 28x28x192x3x3x64
[5x5 conv, 96] 28x28x96x5x5x64
[1x1 conv, 64] 28x28x64x1x1x256
Total: 358M ops

Compared to 854M ops for naive version
Bottleneck can also reduce depth after pooling layer

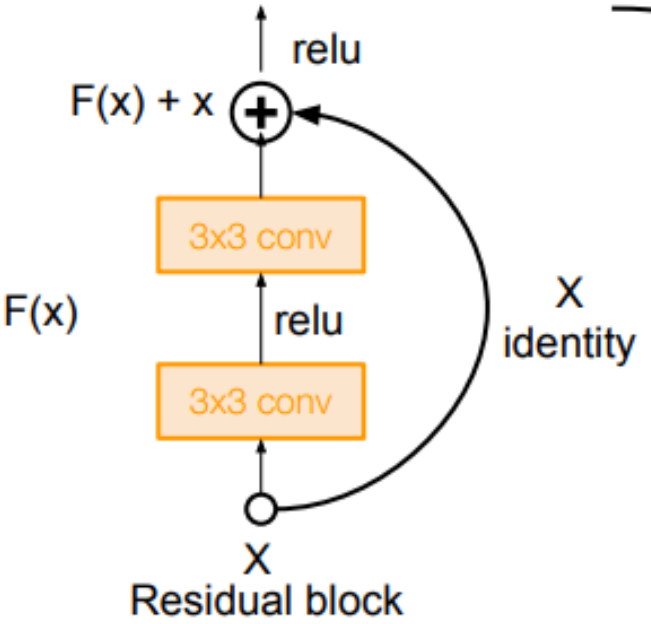
Fei-Fei et al. CS231N – Lecture 9

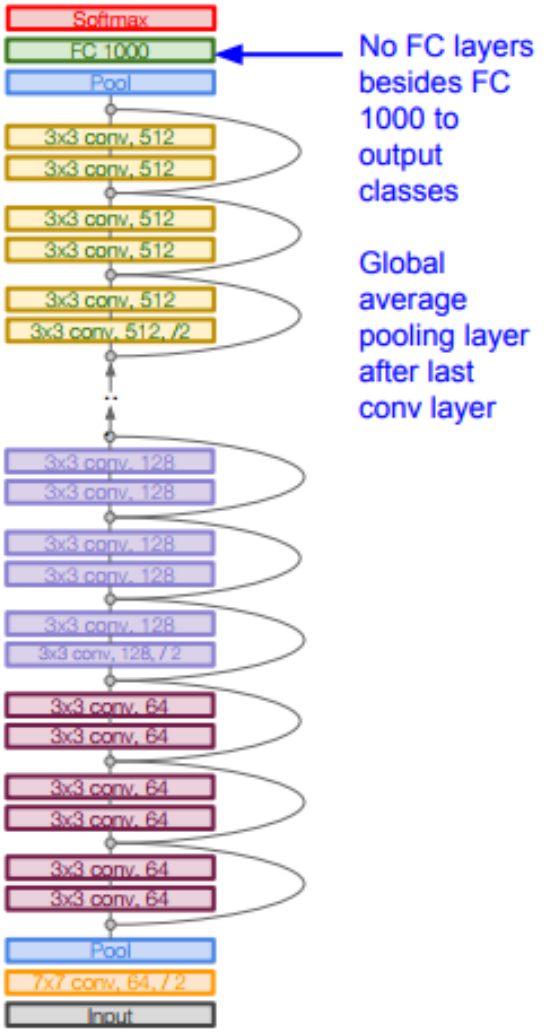
Case Study: ResNet

[He et al., 2015]

Full ResNet architecture:

- Stack residual blocks
- Every residual block has two 3x3 conv layers
- Periodically, double # of filters and downsample spatially using stride 2 (/2 in each dimension)
- Additional conv layer at the beginning
- No FC layers at the end (only FC 1000 to output classes)





Fei-Fei et al. CS231N – Lecture 9

SqueezeNet: AlexNet-level Accuracy With 50x Fewer Parameters and <0.5Mb Model Size

[Iandola et al. 2017]

- Fire modules consisting of a 'squeeze' layer with 1x1 filters feeding an 'expand' layer with 1x1 and 3x3 filters
- AlexNet level accuracy on ImageNet with 50x fewer parameters
- Can compress to 510x smaller than AlexNet (0.5Mb)

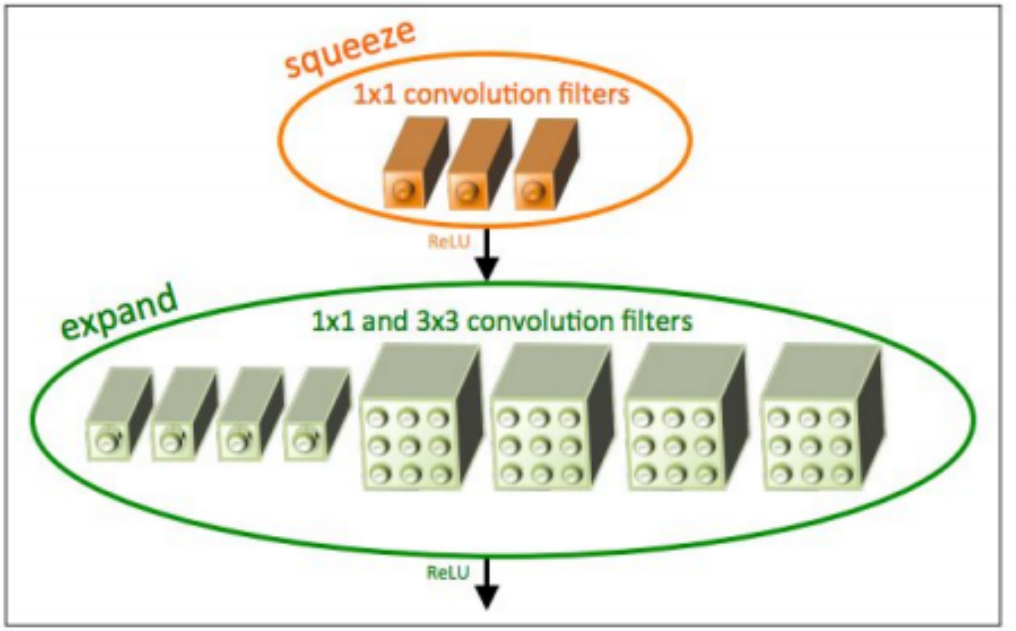
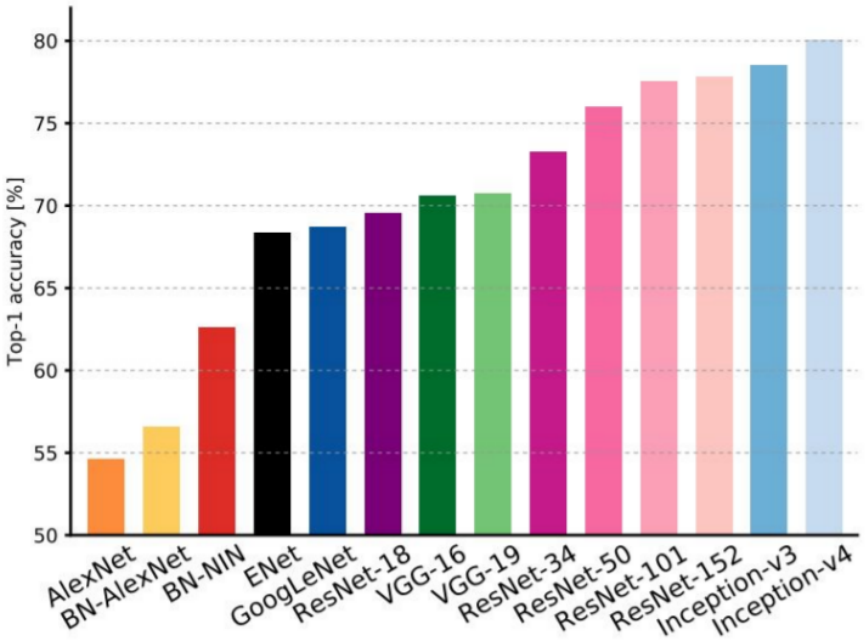
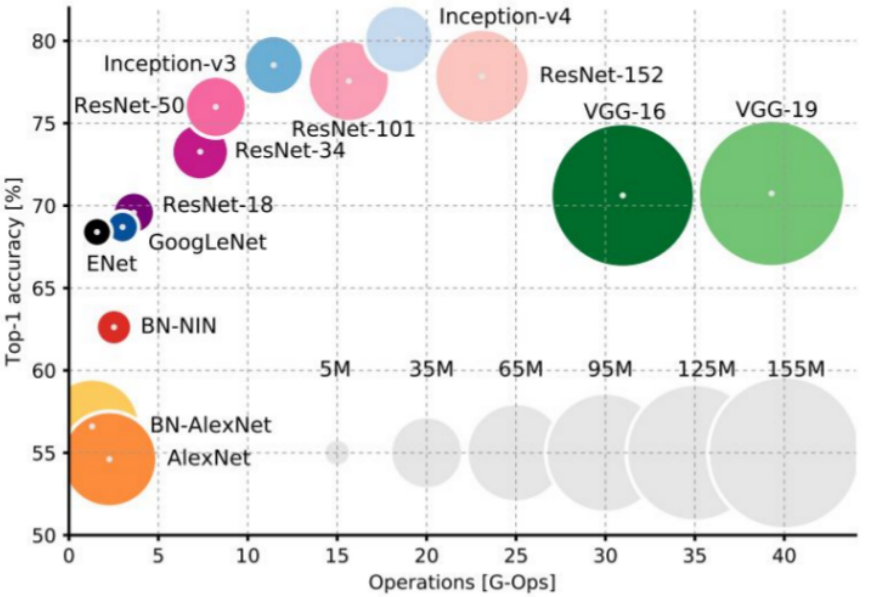


Figure copyright Iandola, Han, Moskewicz, Ashraf, Dally, Keutzer, 2017. Reproduced with permission.

Fei-Fei et al. CS231N – Lecture 9

Comparing Complexity



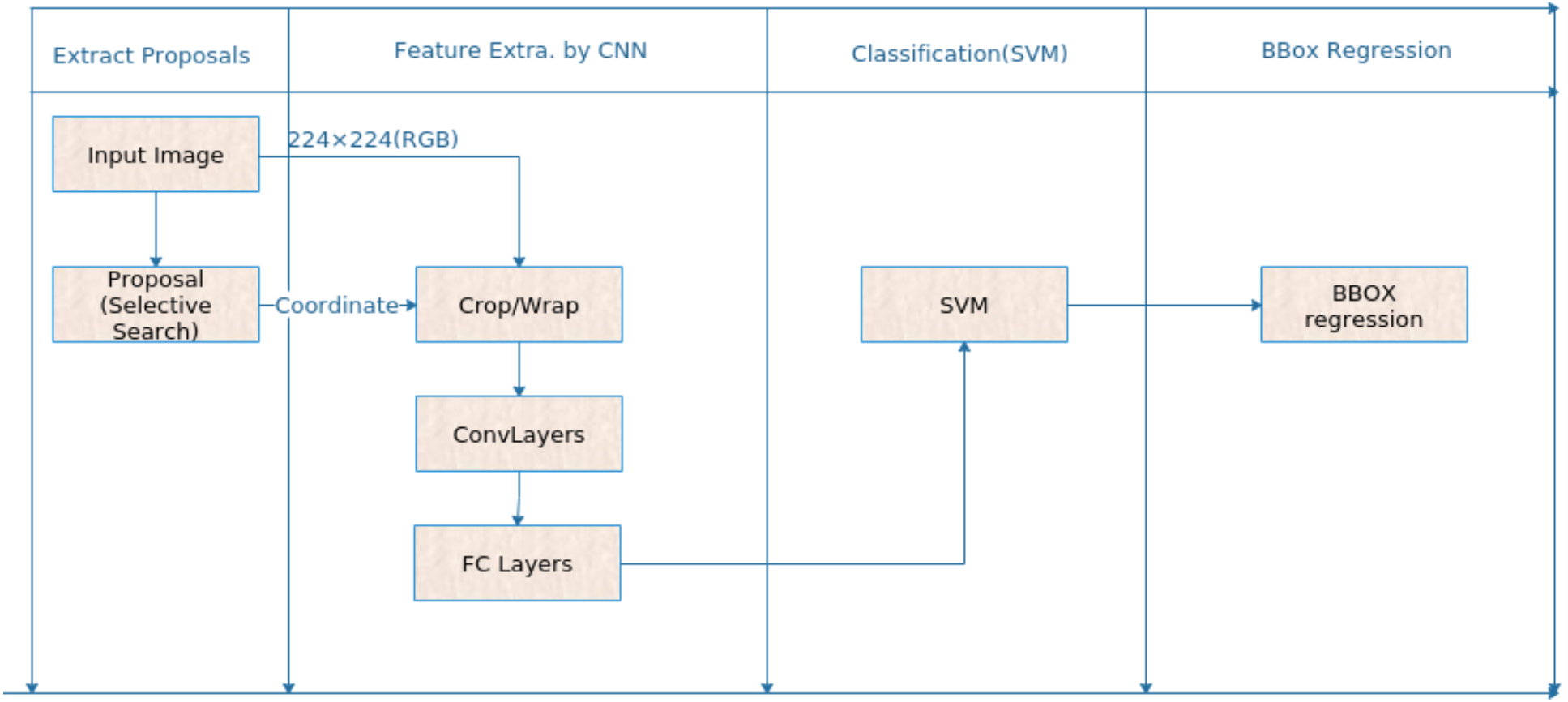


An Analysis of Deep Neural Network Models for Practical Applications, 2017.

R-CNN

- Introduced CNN into the field of object detection the first time.

Work-flow of R-CNN



Step1: Proposal Generation: Generate 1K~2K proposals on one input image by using selective search.
Step2: Feature Extraction: Extract features towards all proposals.
Step3: Classification: Feature to object (classification).
Step4: Tuning: Softmax regression to tune the position of bounding boxes.

Region Proposal Method of R-CNN

Normalize the size of object Proposal.

- Isotropic scaling and Anisotropic scaling.
- Best: Isotropic scaling with padding equals to 16.

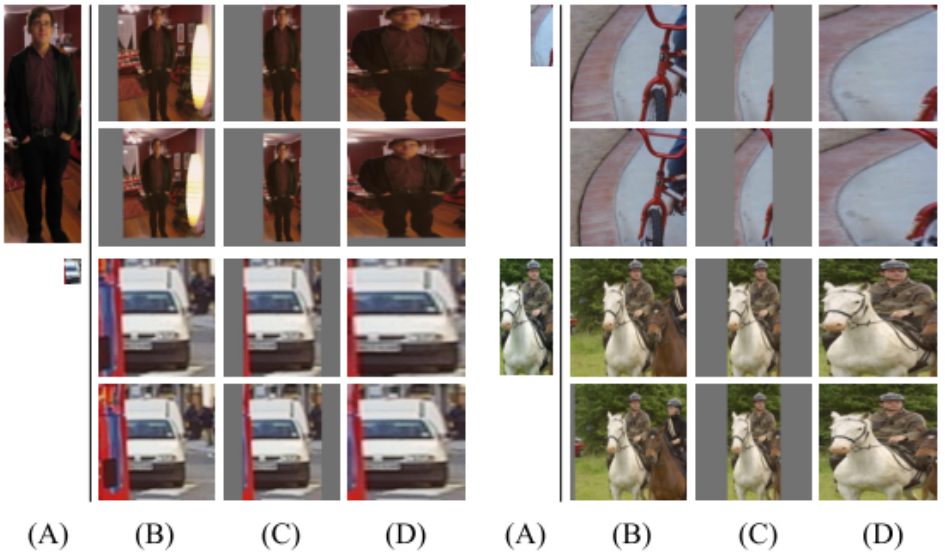
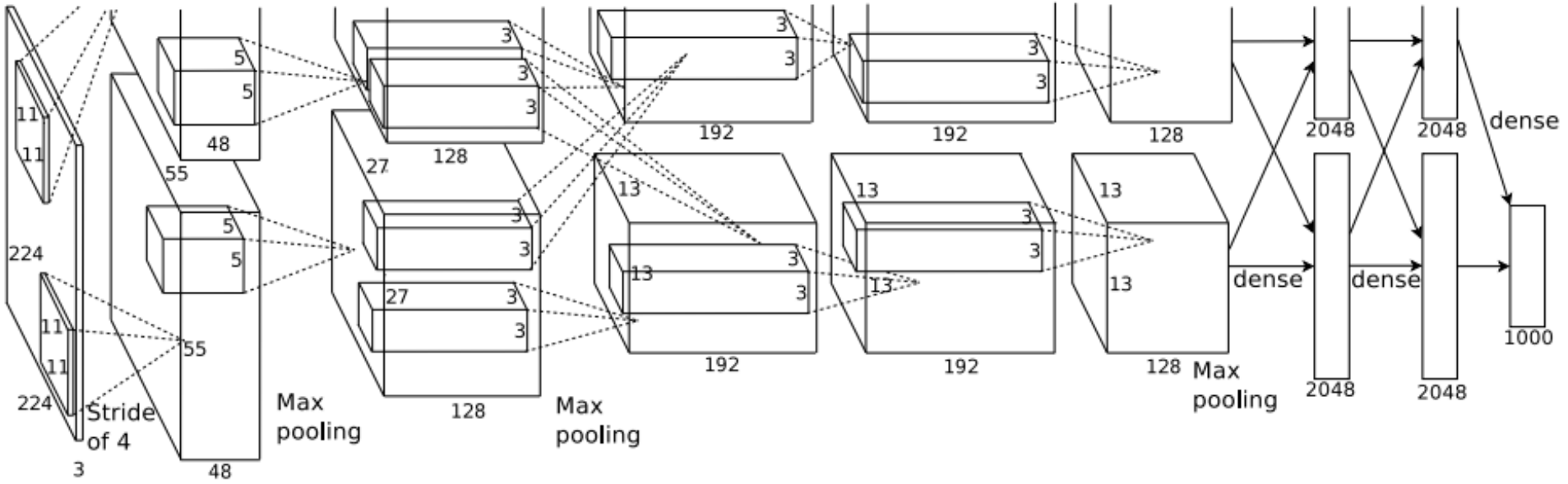


Figure 7: Different object proposal transformations. (A) the original object proposal at its actual scale relative to the transformed CNN inputs; (B) tightest square with context; (C) tightest square without context; (D) warp. Within each column and example proposal, the top row corresponds to $p = 0$ pixels of context padding while the bottom row has $p = 16$ pixels of context padding.

Feature Extraction

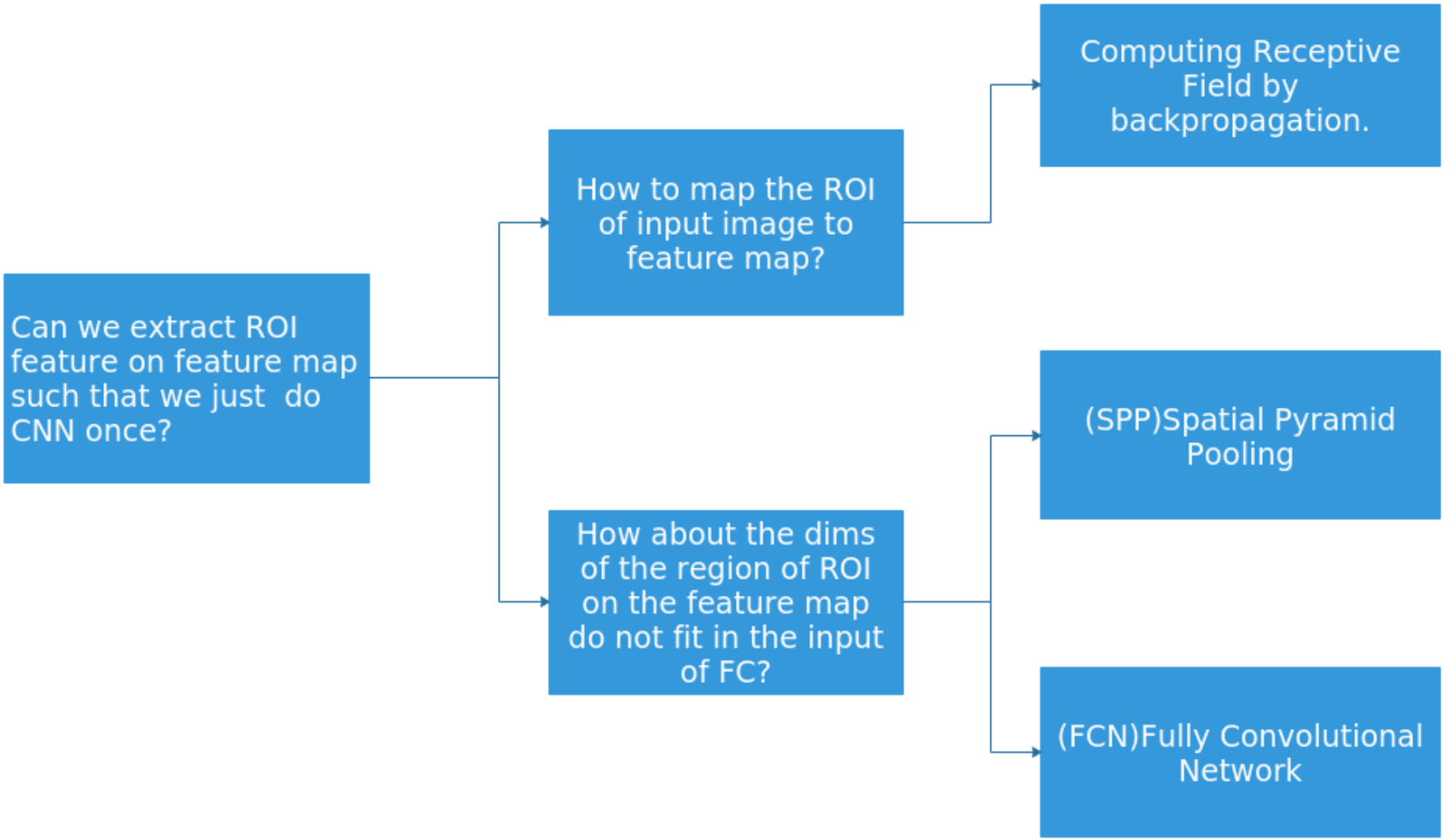


Alex K et al. NIPS12. ImageNet Classification with Deep Convolutional Neural Networks.

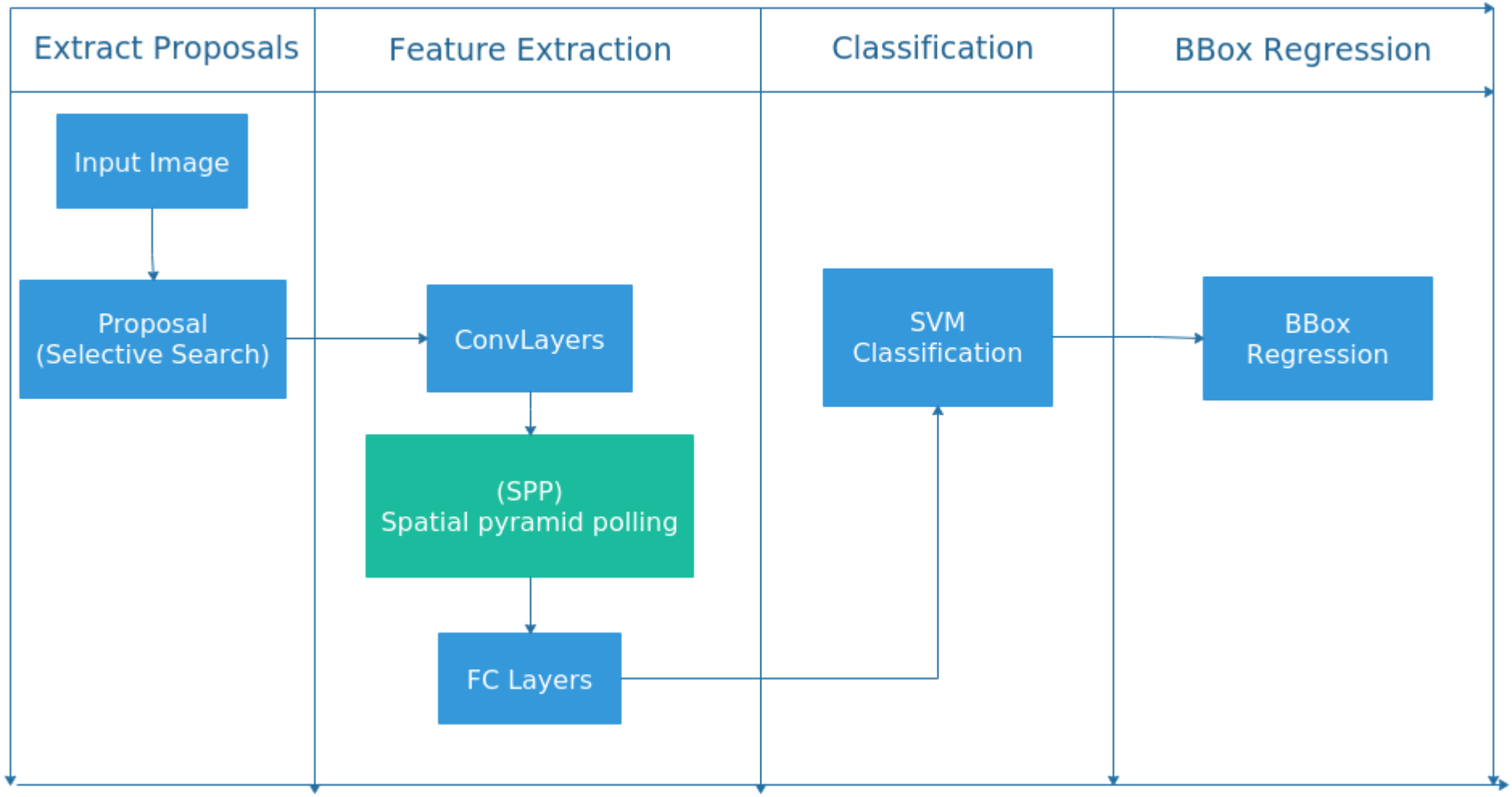
- Accuracy?
  - Alexnet: 58.5%;
  - VGG16: 66%.
- Efficiency?
  - VGG16 is 7x faster than AlexNet.

AlexNet!

Improvement



SPP-Net


Extract Proposals
Feature Extraction
Classification
BBox Regression
Input Image
Proposal
(Selective Search)
ConvLayers
(SPP)
Spatial pyramid polling
FC Layers
SVM
Classification
BBox
Regression

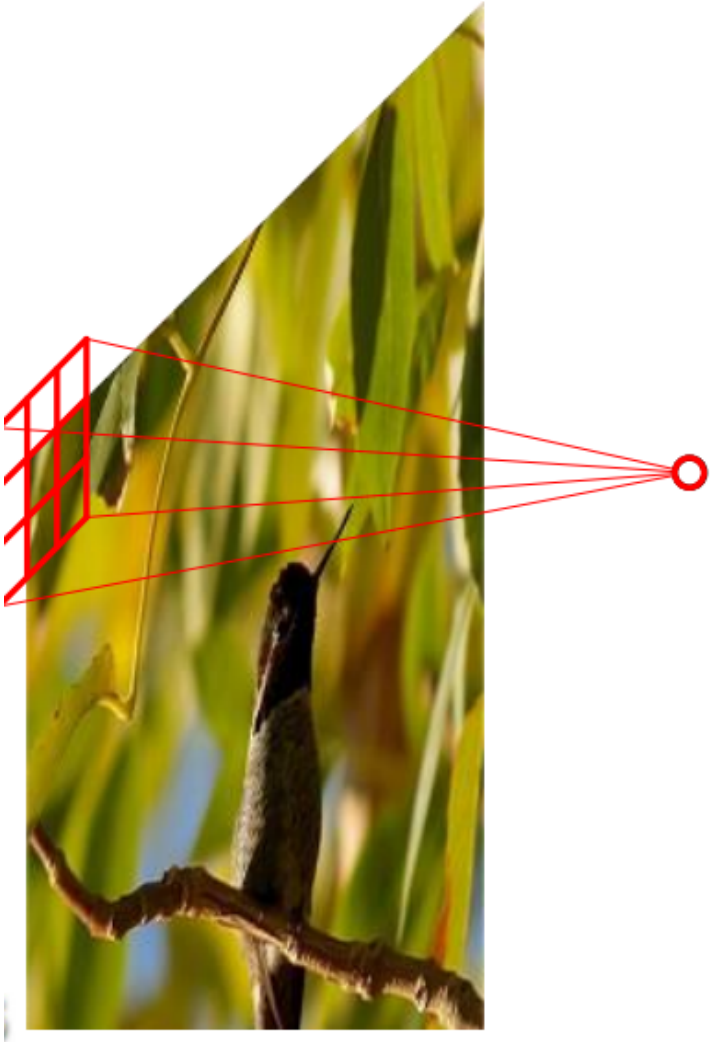
Computation of Receptive Field

- Map a feature map pixel to the center of their perceptive field on the image.

$$i_0 = g_L(i_L) = \alpha_L(i_L - 1) + \beta_L,$$

$$\alpha_L = \prod_{p=1}^{L} S_p,$$

$$\beta_L = 1 + \sum_{p=1}^{L} \left(\prod_{q=1}^{p-1} S_q \right) \left(\frac{F_p - 1}{2} - P_p \right)$$

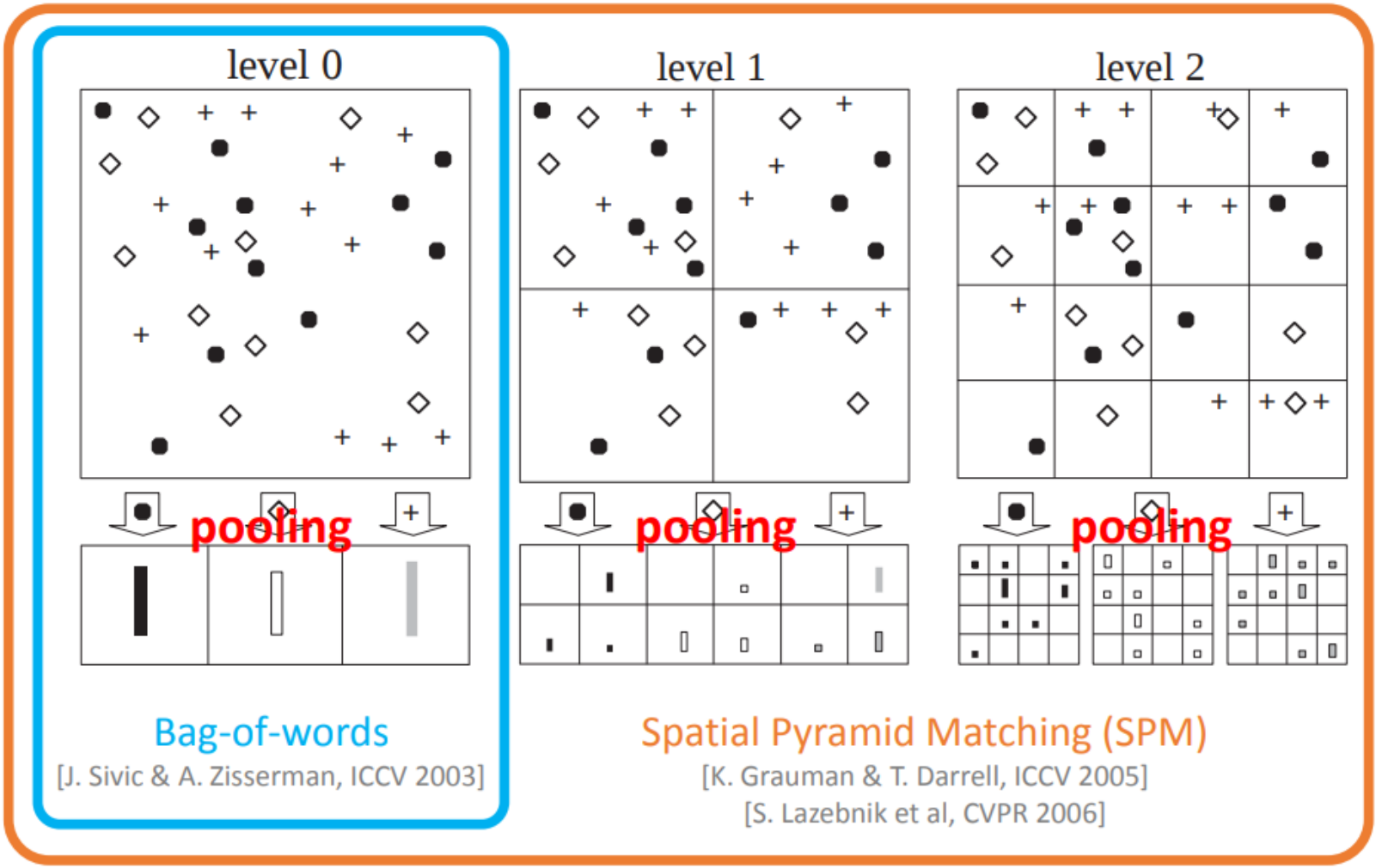


Kaiming He, Xiangyu Zhang, Shaoqing Ren,

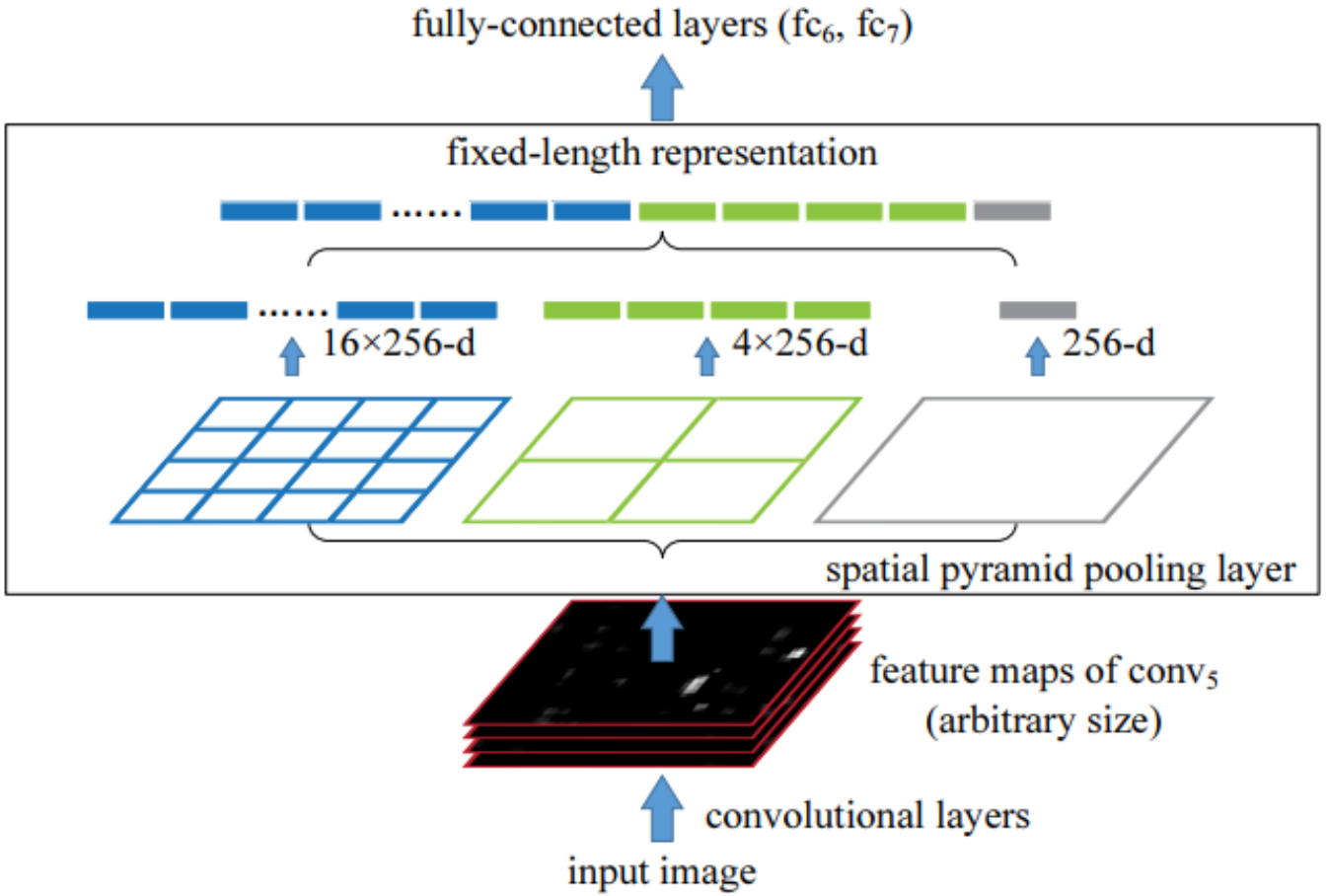
ROI pooling in SPP-Net

SIFT/HOG-based feature maps.

ROI pooling in SPP-Net was inspired by this.

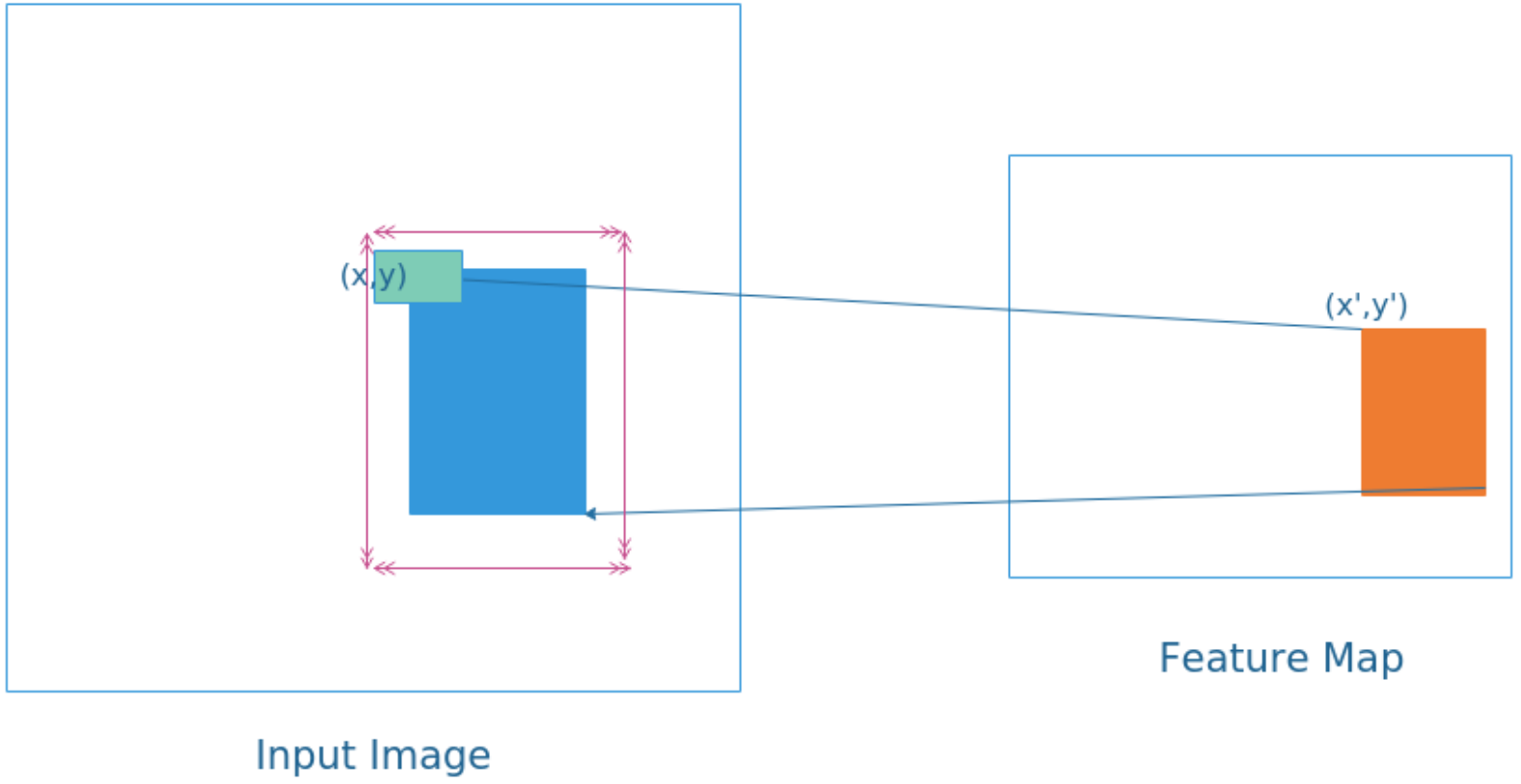


ROI pooling in SPP-Net

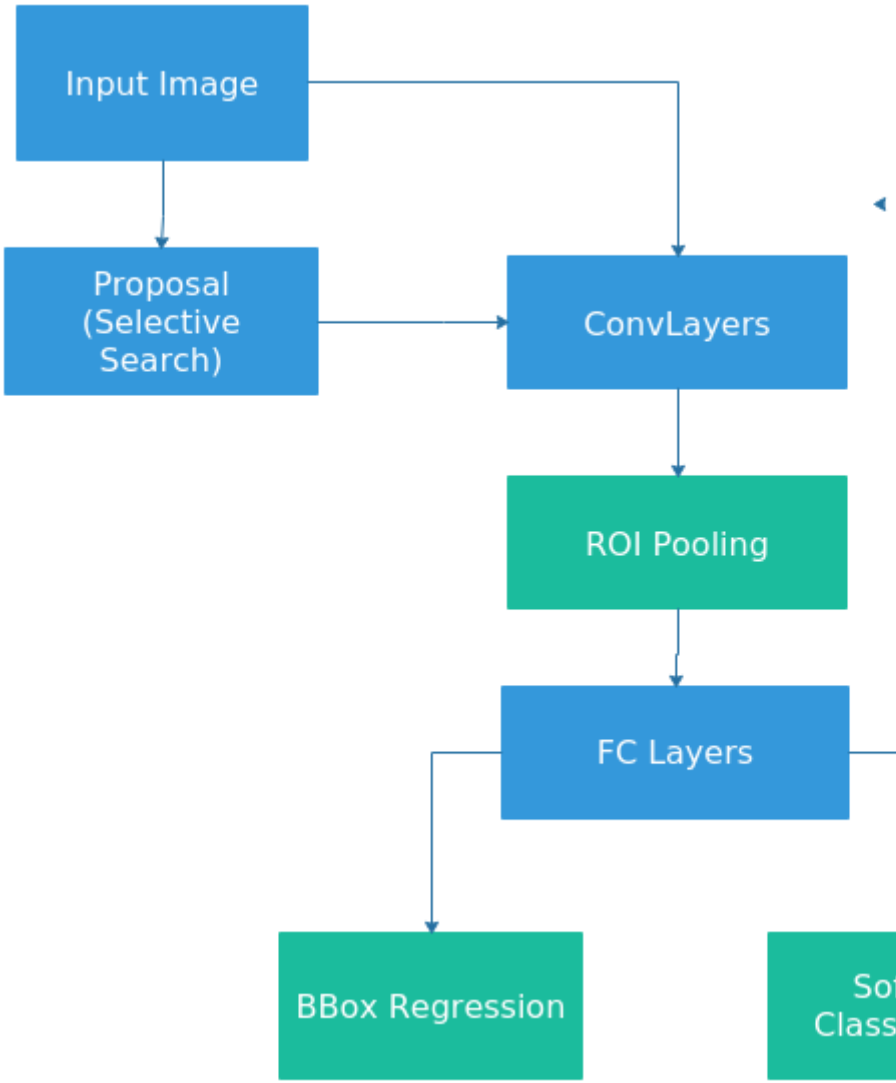


The dims of feature will always be (16+4+1)*256!!!

ROI pooling in SPP-Net



Fast R-CNN



ROI Pooling:
Single Scale Pooling

Joint Training:
Combine SVM and Bbox Regression

Tricks and other contributions

- Proposed a new training way: Training all the proposal of an image at once and then reflecting all the coordinates into conv5.
- Used more robust L1 as loss function instead of L2.
- Speed up by using the SVD at FC.

Bounding-box Regression

1. The definition of bounding-box regression:

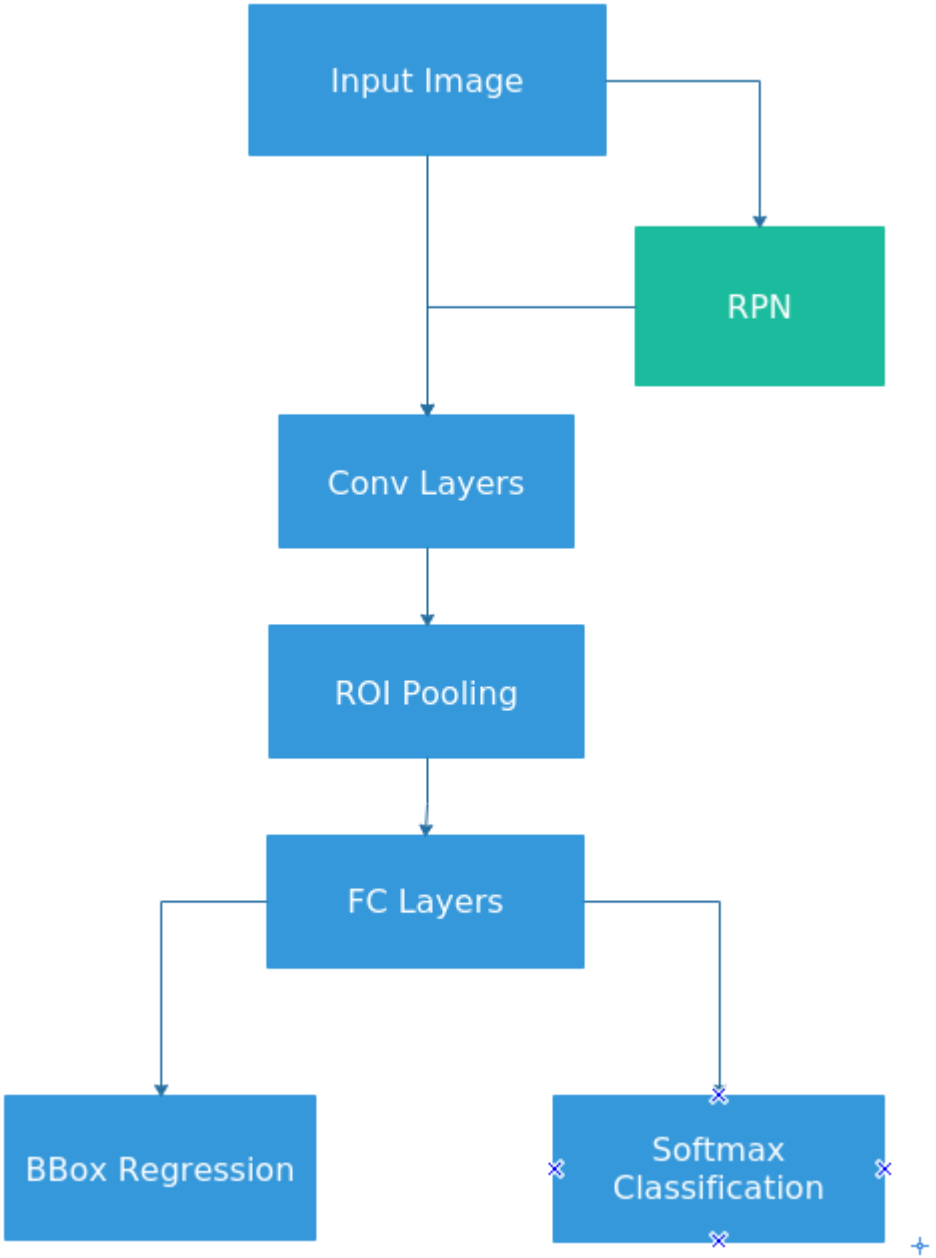
$$t^k = \left(t^k_{\mathrm{x}}, t^k_{\mathrm{y}}, t^k_{\mathrm{w}}, t^k_{\mathrm{h}}\right)$$

2. The definition of loss function:

$$L_{\mathrm{loc}}(t^u, v) = \sum_{i \in \{\mathrm{x,y,w,h}\}} \mathrm{smooth}_{L_1}(t^u_i - v_i),$$

in which

$$\mathrm{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

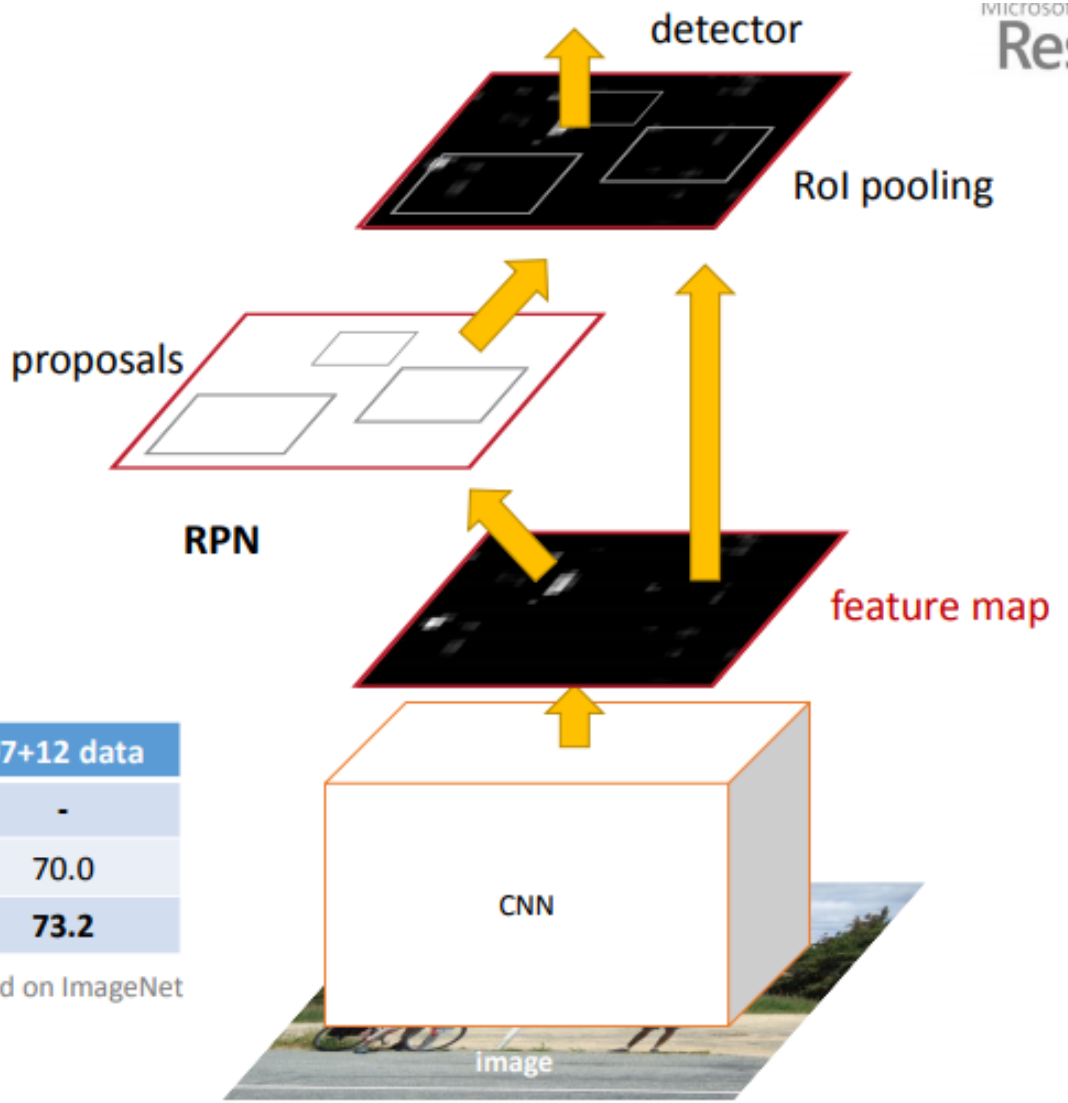


Faster R-CNN

- End-to-end detection method
- Real-Time
- Faster R-CNN ~= Fast RCNN + RPN

Microsoft Research

Faster R-CNN



system	time	07 data	07+12 data
R-CNN	~50s	66.0	-
Fast R-CNN	~2s	66.9	70.0
Faster R-CNN	198ms	**69.9**	**73.2**

detection mAP on PASCAL VOC 2007, with VGG-16 pre-trained on ImageNet

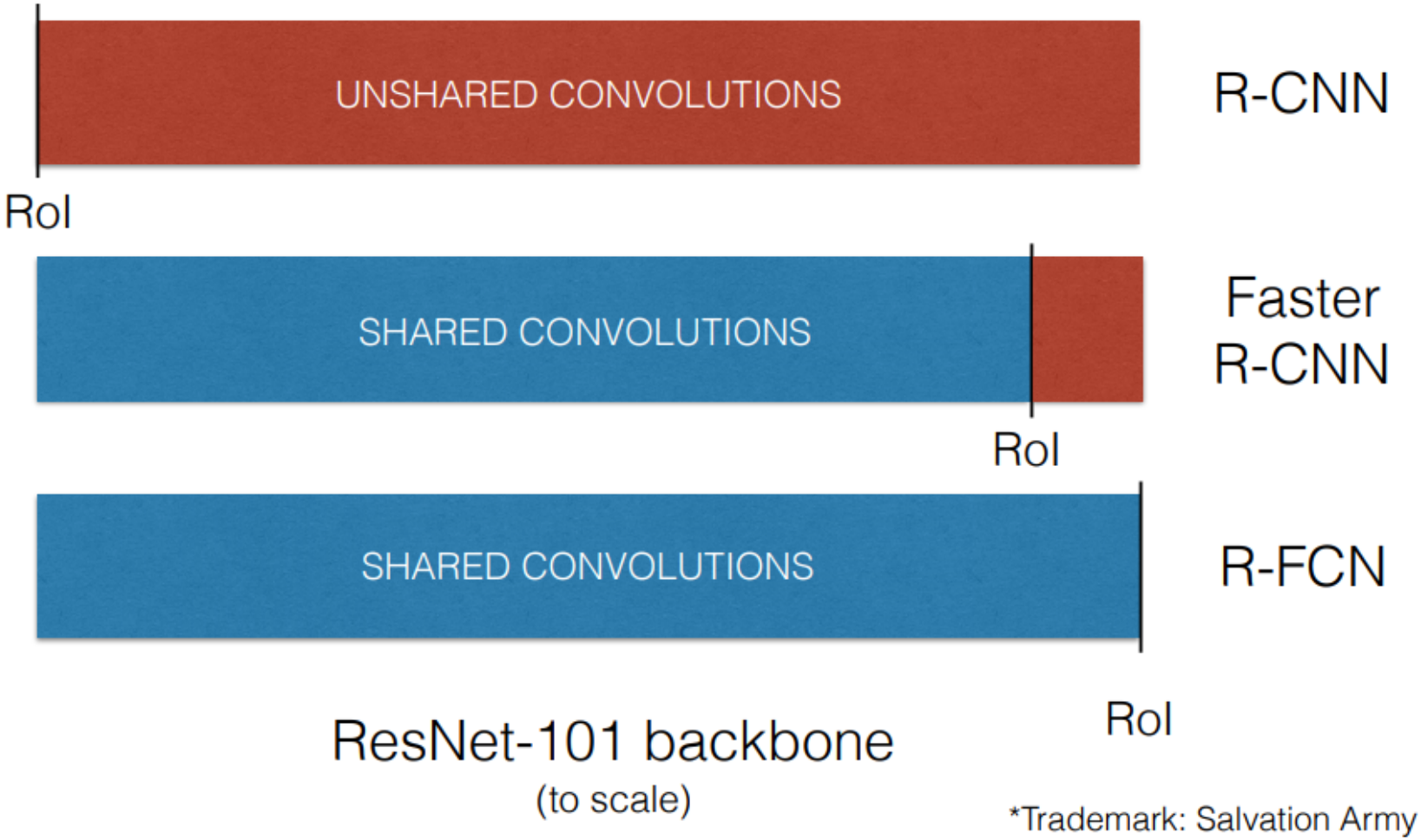
K He et al. ICCV15 Tutorial. Convolutional Feature Maps.

Keys to efficient CNN-based object detection

- Feature sharing
  - R-CNN => SPP-Net & Fast R-CNN: sharing features among proposal regions
  - Fast R-CNN => Faster R-CNN: sharing features between proposal and detection
  - All are done by shared convolutional feature maps
- Efficient multi-scale solutions
  - Single-scale convolutional feature maps are good trade-offs
  - Multi-scale anchors are fast and flexible.

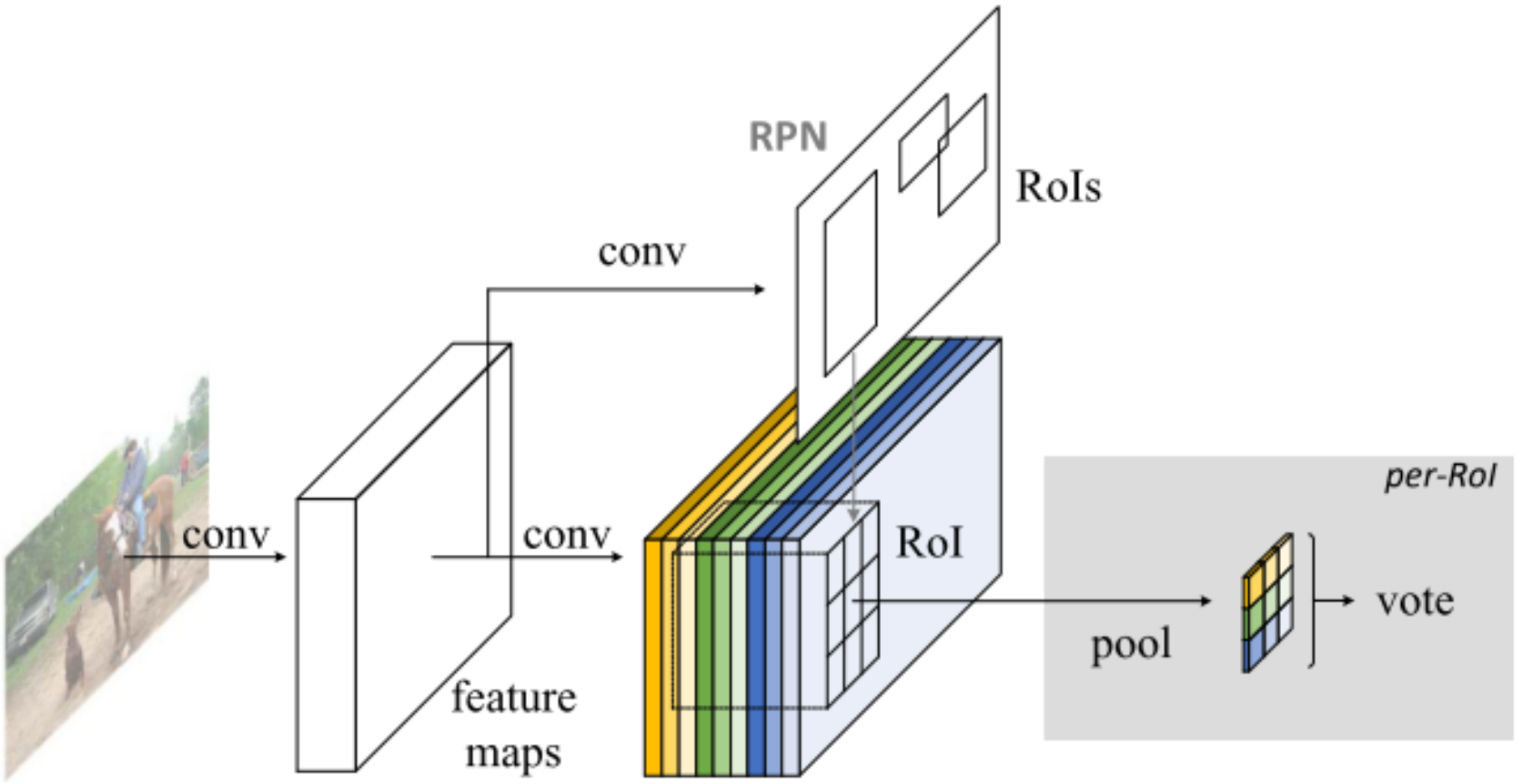
R-FCN

Motivation: Sharing is Caring!

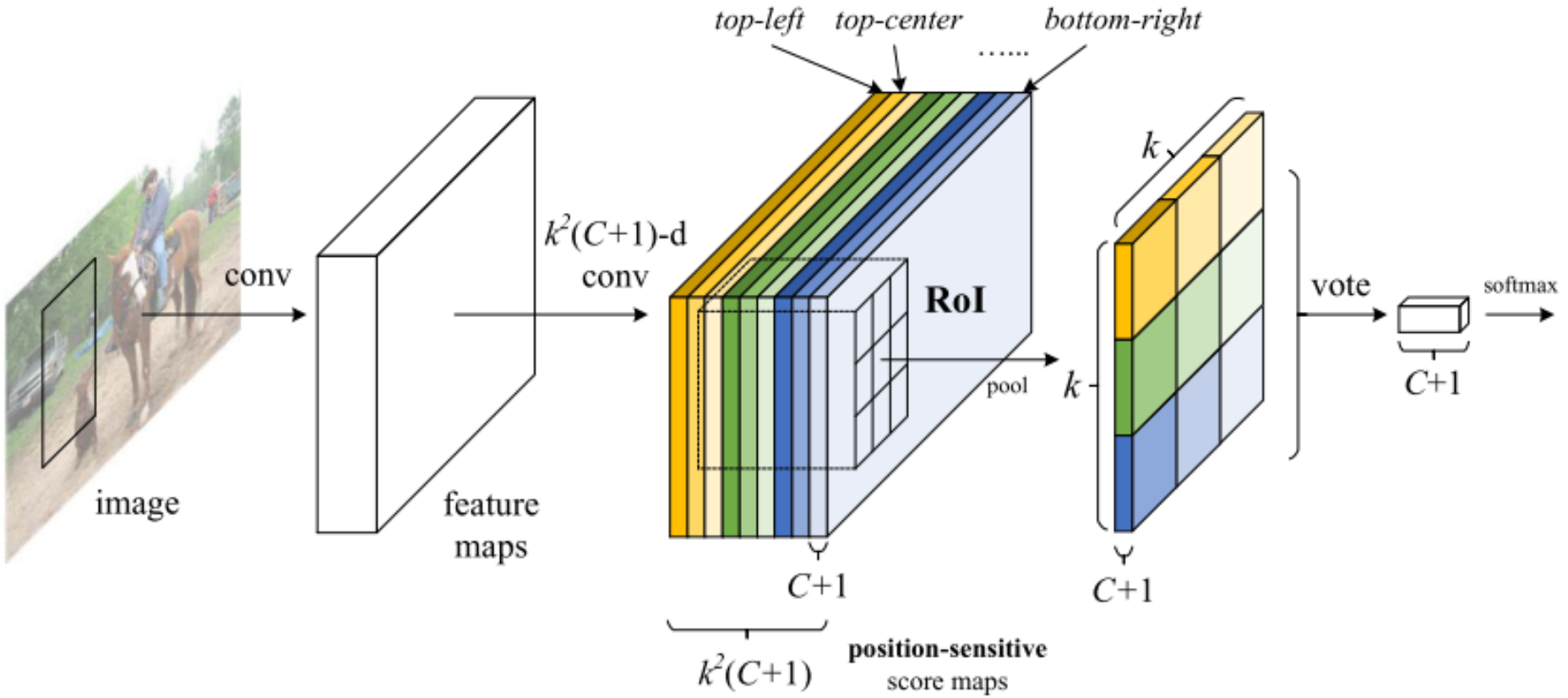


*Trademark: Salvation Army

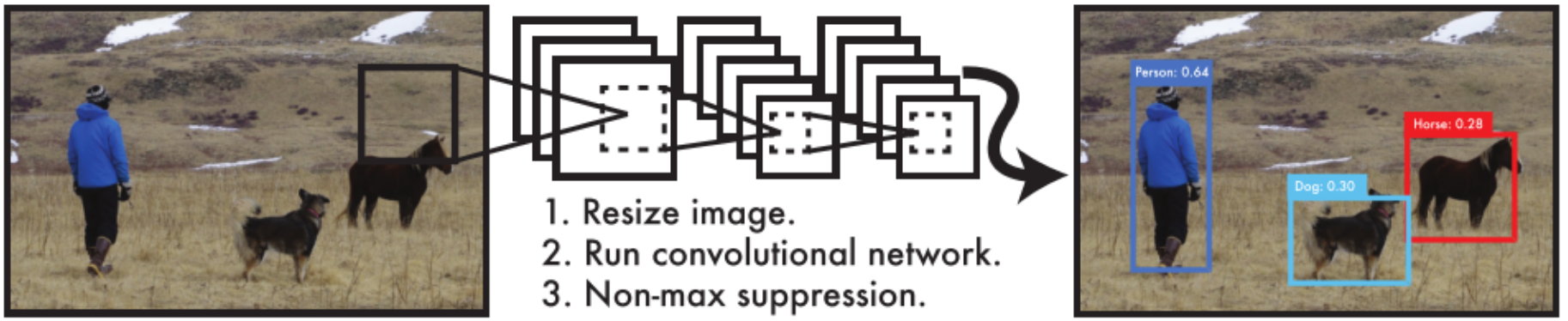
Architecture of R-FCN



Main Idea of R-FCN



YOLO



Main Contributions:

End-to-end object detection method without losing too much efficiency compared to Fast R-CNN. And it makes more localization errors but is less likely to predict false positives on background.

Workflow:

a). Resize the input image to 448*448;

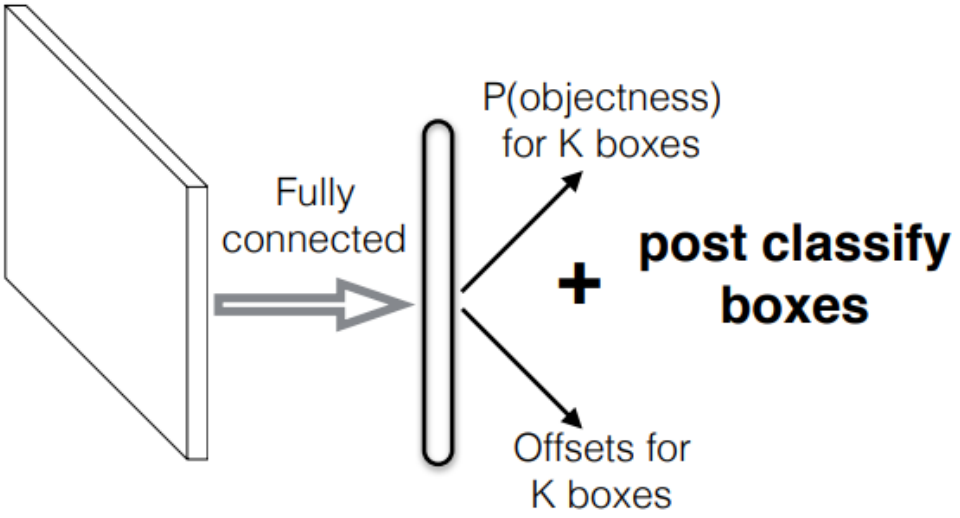
b). Run a GoogLeNet-like network to get the object confidence and coodinates;

c). NMS to select bounding box.

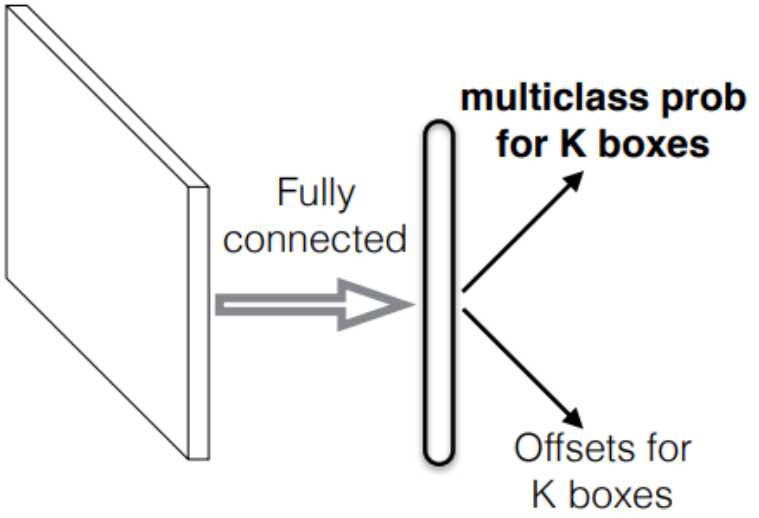
YOLO2

NOT THE HIGHLIGHT METHOD!

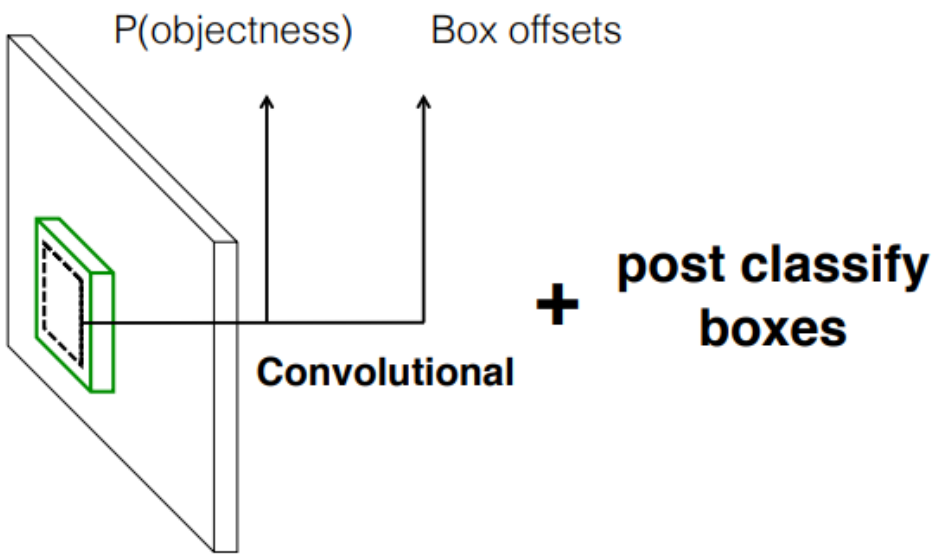
MultiBox [Erhan et al. CVPR14]



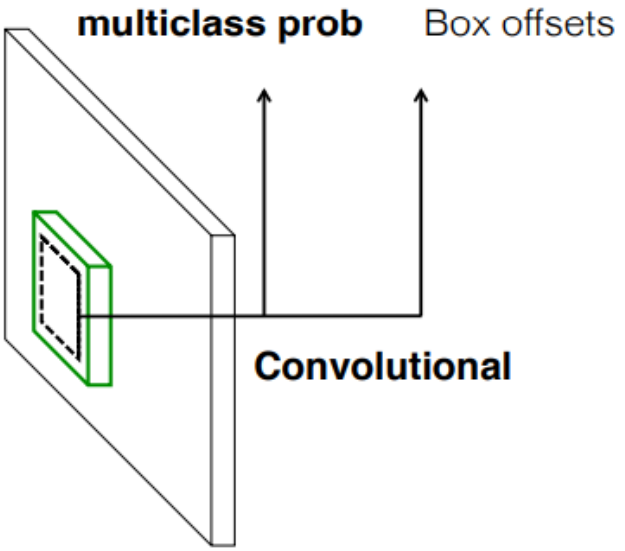
YOLO [Redmon et al. CVPR16]



Faster R-CNN [Ren et al. NIPS15]



SSD



Part IV:
Object Detection From 2D to 3D

- Main Challenges
- 3D Detection in KITTI
- Current Situations

Main Challenge From 2D to 3D

- Remains

3D Detection on KITTI

Current Situations

Part V:
State-of-art Methods in 3D Object Detection

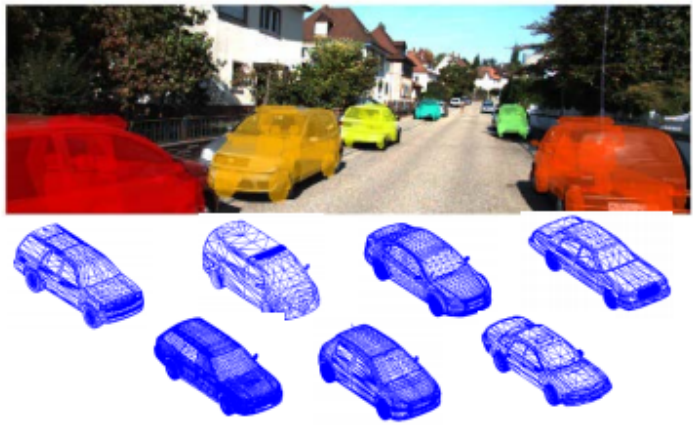
Mainstream 3D Detection Methods

- LIDAR-based Methods(VeloFCN)
- Mono-based Methods(Mono3D)
- Stereo-image-based Methods(3DOP)
- Image + LIDAR Methods(MV3D)

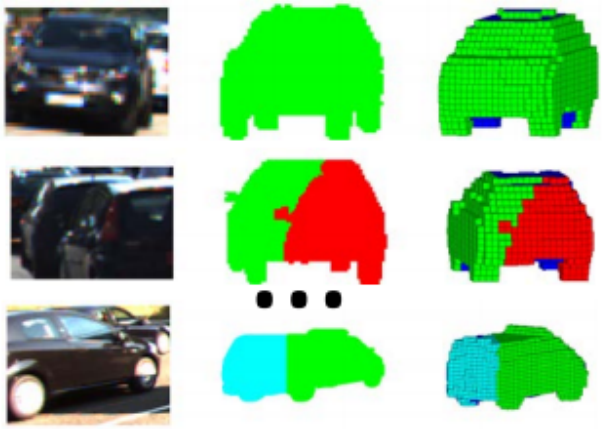
3DVP(CVPR15)

- Main Idea
  - Align the car model by CAD and the real car. Then do some fine-tuning.
- Strength of the approach
  - Estimate detailed properties of objects beyond 2D bounding boxes.
- Weakness of the approach
  - Time-consuming(40s)
- Future Direction
  - Be able to adapt to different problems using different CAD models(e.g., Cyclists, Pedestrians)
  - NO POSSIBLE TO BE REAL-TIME!

Training Pipeline

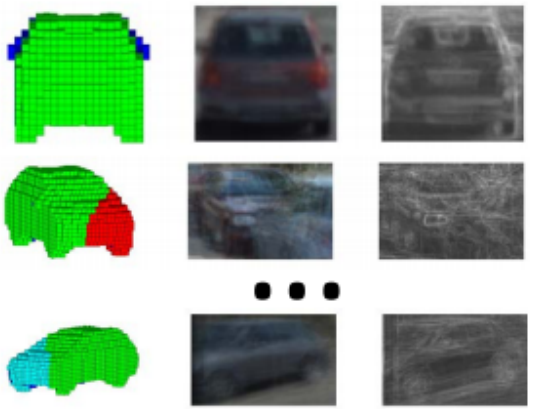


1. Align 2D images with 3D CAD models



2. 3D voxel exemplars

3. 3D voxel patterns



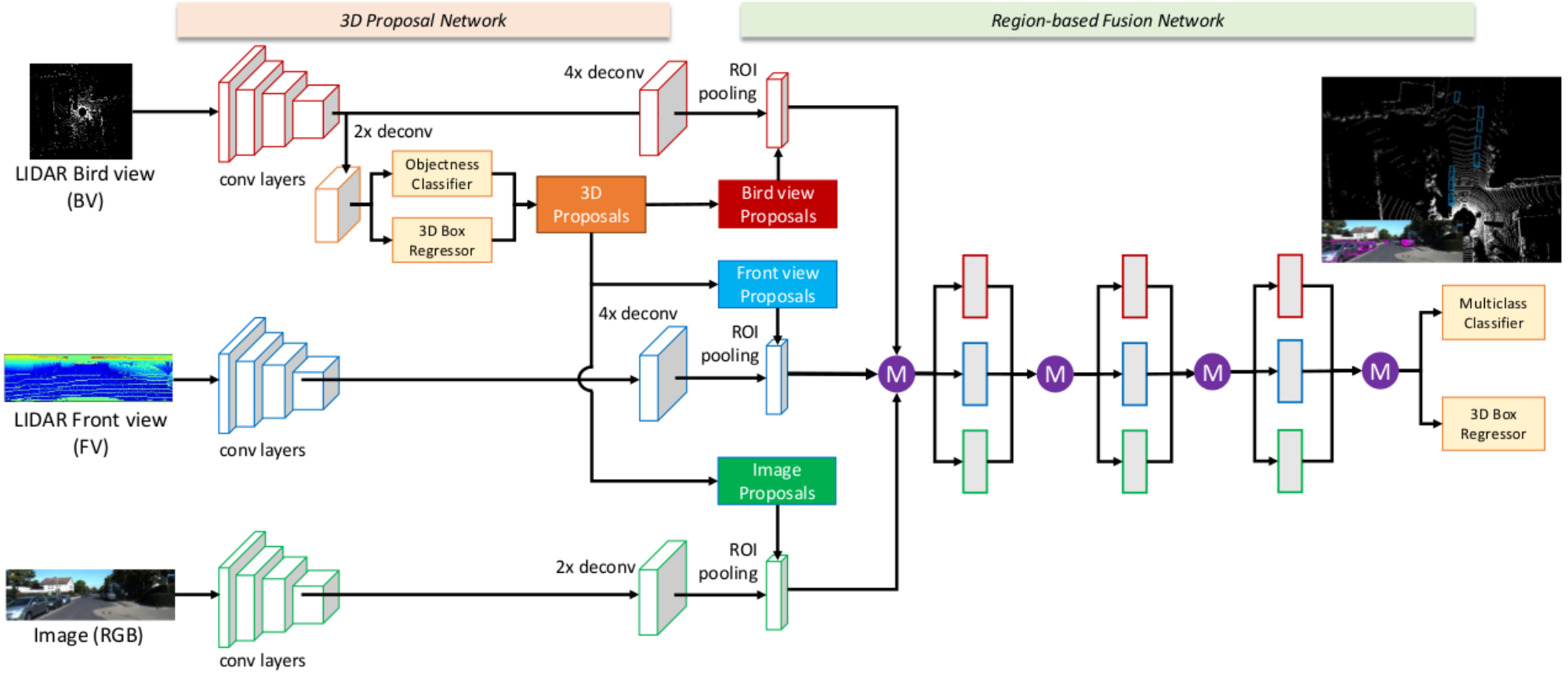
4. Training 3D voxel pattern detectors

3DOP

Mono3D

VeloFCN

MV3D


3D Proposal Network
Region-based Fusion Network
LIDAR Bird view (BV)
conv layers
2x deconv
4x deconv
ROI pooling
Objectness Classifier
3D Box Regressor
3D Proposals
Bird view Proposals
Front view Proposals
Image Proposals
LIDAR Front view (FV)
Image (RGB)
M
Multiclass Classifier
3D Box Regressor

Part VI:
Create an robust real-time methods both in 2D and 3D Object Detection with a comparable mAP

- Introduction
- Related Work
- Architecture Design
- Experiment Design
- Problems

Introduction

Related Work

Architecture Design

Experiment Design

Unsolved Problems

Reference

1) Fei-Fei et al. Spring 2017. CS231n: Convolutional Neural Networks for Visual Recognition – Lecture 2,3,9.
2) Fei-Fei et al. Nov 2011. CS231A: Computer Vision, From 3D Reconstruction to Recognition – Lecture 14.
3) Deep Learning for Objects and Scenes. CVPR17 Toturial.
4) R Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation.
5) R Girshick et al. Fast R-CNN.
6) S Ren et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.
7) K He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.
8) J Redmon et al. You Only Look Once: Unified, Real-Time Object Detection.
9) W Liu et al. SSD: Single Shot MultiBox Detector.
10) K He et al. Deep Residual Learning for Image Recognition.
11) FN Iandoal et al. SqueezeNet: AlexNet-Level Accuracy With 50x Fewer Parameters and <0.5MB Model Size.
12) B Wu et al. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving.
13) Alex K et al. NIPS12. ImageNet Classification with Deep Convolutional Neural Networks.
14) K He et al. ICCV2015 tutorial. Convolutional Feature Maps.
15) Ren et al. Accurate Single Sate Detector Using Reccurent Rolling Convolution.

Preference

1) S Lazebnik et al. CVPR06. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories.
2) Matthew D. Zeiler et al. ECCV14. Visualizing and Understanding Convolutional Networks.
3) A Canziani et al. ICLR17. An Analysis of Deep Neural Network Models for Practical Applications.
4) J Dai et al. NIPS16. R-FCN: Object Detection via Region-based Fully Convolutional Networks.
5) VGG Reading Group – Sam Albanie. NIP16. Robots.ox.ac.uk.
6) Fei-Fei et al. CS231N. Detection and Segmentation.
7) X Yu et al. CVPR15. Data-Driven 3D Voxel Patterns for Object Category Recognition.
8) X Wei et al. Context-aware Single-Shot Detector.
9) Chen Xiaozhi et al. CVPR16. Monocular 3D Object Detection for Autonomous Driving.
10) Chen Xiaozhi et al. NIPS15. 3D Object Proposals for Accurate Object Class Detection.
11) Chen Xiaozhi et al. CVPR17. Multi-View 3D Object Detection Network for Autonomous Driving.
12) H Gao et al. CVPR17. Densely Connected Convolutional Networks.
13) K Alex et al. NIPS12. ImageNet Classification with Deep Convolutional Neural Networks.
14) Gu Shixiang et al. Towards Deep Neural Network Architectures.
15) Zeiler Mathhew et al. ECCV14. Visuallizing and Understanding Convolutional Networks.

Reference

1) Bell et al. Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Network.

2) Lazebnik et al. CVPR06. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories.

3) Ashraf et al. Shallow Networks for High-Accuracy Road Object Detection.

4) Wu et al. CVPR17. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving.

5) Gidaris, Spyros, Komodakis, Nikos. ICCV15. Object detection via a multi-region & semantic segmentation-aware CNN model.

6) Dai JF et al. NIPS16. R-FCN: Object Detection via Region-based Fully Convolutional Networks.

Thanks!