Computer Vision – Lecture 12

Deep Learning III

17.06.2019

Bastian Leibe
Visual Computing Institute
RWTH Aachen University
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de

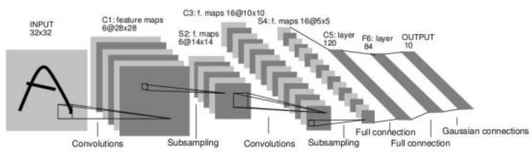
Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition & Categorization
  - Sliding Window based Object Detection
- Local Features & Matching
- Deep Learning
  - Convolutional Neural Networks (CNNs)
  - Deep Learning Background
  - CNNs for Object Detection
  - CNNs for Semantic Segmentation
  - CNNs for Matching
- 3D Reconstruction

Topics of This Lecture

- CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
  - ResNet
- CNNs for Object Detection
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
  - Mask R-CNN
  - YOLO / SSD

CNN Architectures: LeNet (1998)



- Early convolutional architecture
  - 2 Convolutional layers, 2 pooling layers
  - Fully-connected NN layers for classification
  - Successfully used for handwritten digit recognition (MNIST)

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11): 2278–2324, 1998.

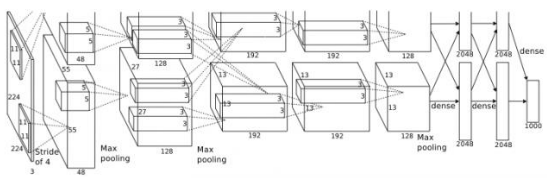
Slide credit: Svetlana Lazebnik

ImageNet Challenge 2012

[Deng et al., CVPR'09]

- ImageNet
  - ~14M labeled internet images
  - 20k classes
  - Human labels via Amazon Mechanical Turk
- Challenge (ILSVRC)
  - 1.2 million training images
  - 1000 classes
  - Goal: Predict ground-truth class within top-5 responses
  - Currently one of the top benchmarks in Computer Vision

CNN Architectures: AlexNet (2012)

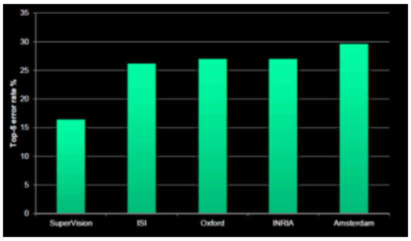


- Similar framework as LeNet, but
  - Bigger model (7 hidden layers, 650k units, 60M parameters)
  - More data (10^6 images instead of 10^3)
  - GPU implementation
  - Better regularization and up-to-date tricks for training (Dropout)

A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012.

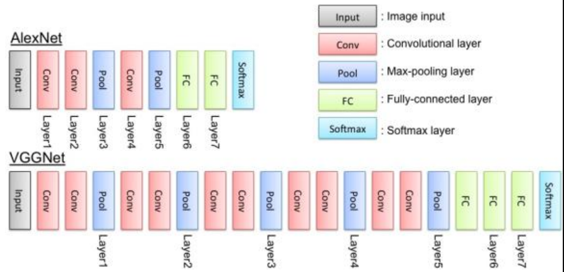
Image source: A. Krizhevsky, I. Sutskever and G.E. Hinton, NIPS 2012

ILSVRC 2012 Results



- AlexNet almost halved the error rate
  - 16.4% error (top-5) vs. 26.2% for the next best approach
  - ⇒ A revolution in Computer Vision
  - Acquired by Google in Jan '13, deployed in Google+ in May '13

CNN Architectures: VGGNet (2014/15)



K. Simonyan, A. Zisserman, [Very Deep Convolutional Networks for Large-Scale Image Recognition](), ICLR 2015

CNN Architectures: VGGNet (2014/15)

- Main ideas
  - Deeper network
  - Stacked convolutional layers with smaller filters (+ nonlinearity)
  - Detailed evaluation of all components
- Results
  - Improved ILSVRC top-5 error rate to 6.7%.
  - 138M parameters (VGG16), most of those in the FC layers (102M)

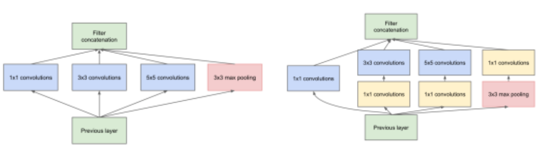
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Mainly used

Comparison: AlexNet vs. VGGNet

- Receptive fields in the first layer
  - AlexNet: 11×11, stride 4
  - Zeiler & Fergus: 7×7, stride 2
  - VGGNet: 3×3, stride 1
- Why that?
  - If you stack a 3×3 layer on top of another 3×3 layer, you effectively get a 5×5 receptive field.
  - With three 3×3 layers, the receptive field is already 7×7.
  - But much fewer parameters: $3 \cdot 3^2 = 27$ instead of $7^2 = 49$.
  - In addition, non-linearities in-between 3×3 layers for additional discriminativity.

CNN Architectures: GoogLeNet (2014)



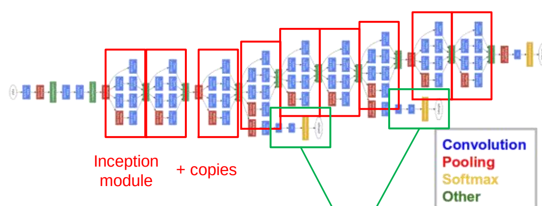
(a) Inception module, naïve version

(b) Inception module with dimension reductions

- Main ideas
  - "Inception" module as modular component
  - Learns filters at several scales within each module
  - 1x1 convolutions ("bottleneck layers") for dimensionality reduction

C. Szegedy, W. Liu, Y. Jia, et al, [Going Deeper with Convolutions](), arXiv:1409.4842, 2014.

GoogLeNet Visualization



- 22-layer network
  - No FC layers
  - Only 5M parameters
  - ILSVRC'14 winner with 6.7% top-5 error

Auxiliary classification outputs for training the lower layers (deprecated)

Results on ILSVRC

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

- VGGNet and GoogLeNet perform at similar level
  - Comparison: human performance ~5% [Karpathy]

http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/

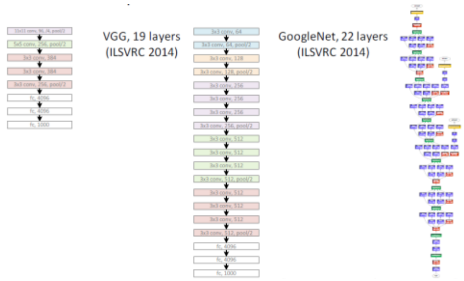
Image source: Simonyan & Zisserman

Residual Networks

AlexNet, 8 layers (ILSVRC 2012)

VGG, 19 layers (ILSVRC 2014)

GoogleNet, 22 layers (ILSVRC 2014)



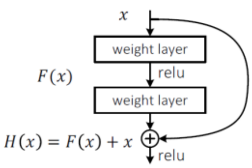
Residual Networks

AlexNet, 8 layers (ILSVRC 2012)

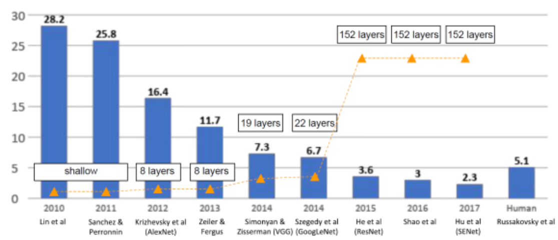
VGG, 19 layers (ILSVRC 2014)

ResNet, 152 layers (ILSVRC 2015)

- Core component
  - Skip connections bypassing each layer
  - Better propagation of gradients to the deeper layers

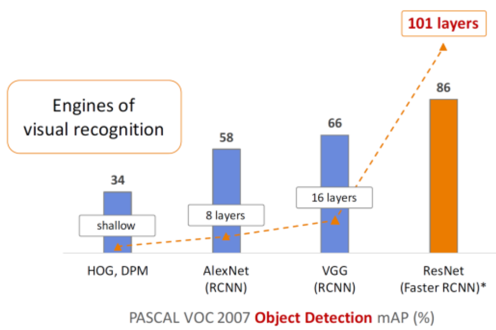


ILSRVC Winners



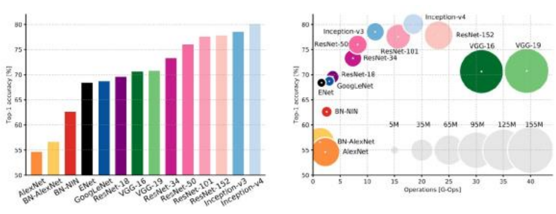
Slide credit: FeiFei Li

PASCAL VOC Object Detection Performance



Slide credit: Kaiming He

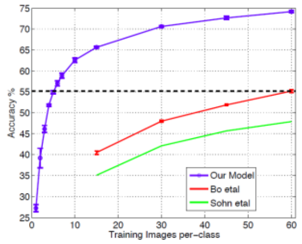
Comparing Complexity



A. Canziano, A. Paszke, E. Culurcello, An Analysis of Deep Neural Network Models for Practical Applications, arXiv 2017.

Figure credit: Alfredo Canziano, Adam Paszke, Eugenio Culurcello

The Learned Features are Generic



state of the art level (pre-CNN)

- Experiment: feature transfer
  - Train AlexNet-like network on ImageNet
  - Chop off last layer and train classification layer on CalTech256
  - ⇒ State of the art accuracy already with only 6 training images!

Image source: M. Zeiler, R. Fergus

Transfer Learning with CNNs

1. Train on ImageNet

2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

I.e., swap the Softmax layer at the end

Slide credit: Andrei Karpathy

Transfer Learning with CNNs

1. Train on ImageNet

3. If you have medium sized dataset, "finetune" instead: use the old weights as initialization, train the full network or only some of the higher layers.

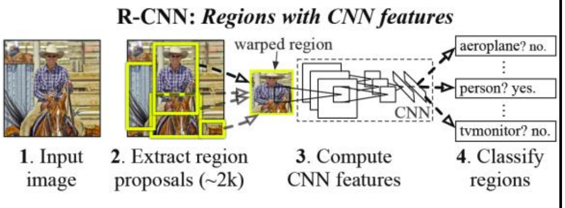
Retrain bigger portion of the network

Slide credit: Andrei Karpathy

Topics of This Lecture

- CNN Architectures
  - LeNet
  - AlexNet
  - VGGNet
  - GoogLeNet
  - ResNet
- CNNs for Object Detection
  - R-CNN
  - Fast R-CNN
  - Faster R-CNN
  - Mask R-CNN
  - YOLO / SSD

Object Detection: R-CNN



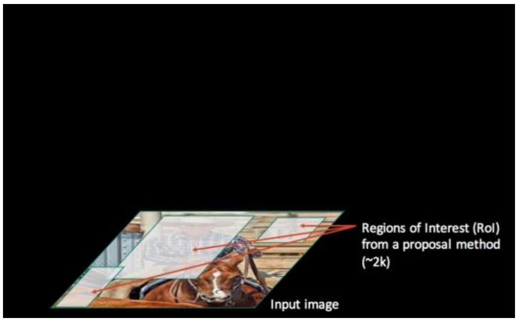
- Results on PASCAL VOC Detection benchmark
  - Pre-CNN state of the art: 35.1% mAP [Uijlings et al., 2013]
    33.4% mAP DPM
  - R-CNN: 53.7% mAP

R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014

R-CNN Pipeline

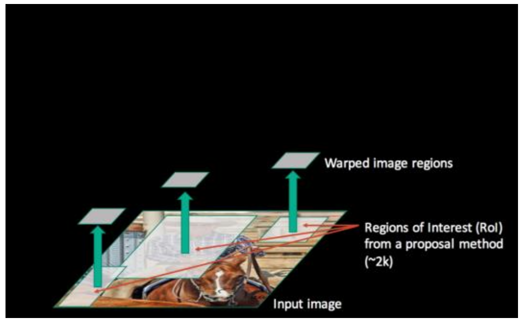
Slide credit: Ross Girshick

R-CNN Pipeline



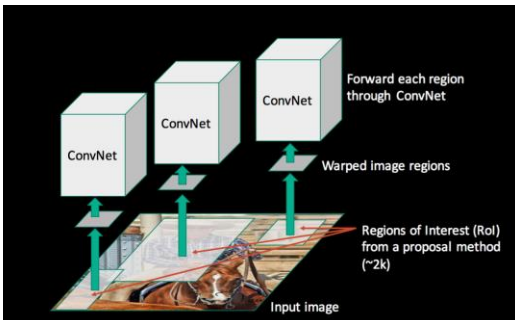
Computer Vision Summer'19

R-CNN Pipeline



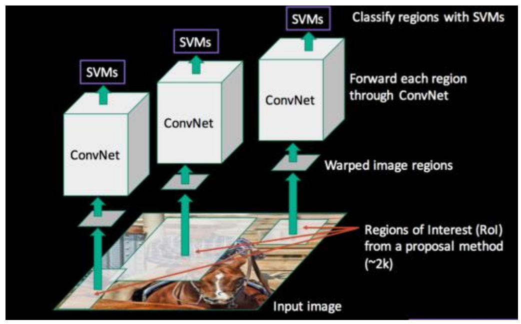
Computer Vision Summer'19

R-CNN Pipeline



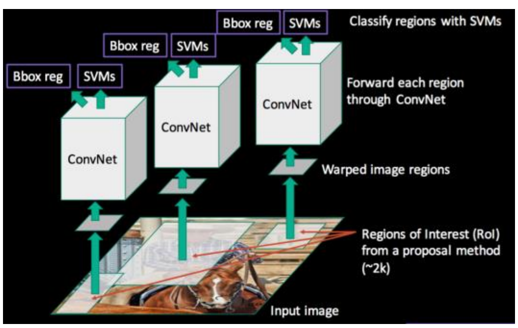
Computer Vision Summer'19

R-CNN Pipeline



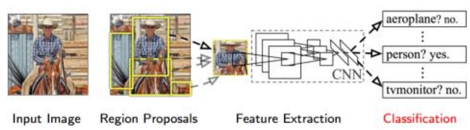
Computer Vision Summer'19

R-CNN Pipeline



Computer Vision Summer'19

Classification



- Linear model with class-dependent weights
  - Linear SVM

$$f_c(x_{fc7}) = w_c^T x_{fc7}$$

  - where
    - x_{fc7} = features from the network (fully-connected layer 7)
    - c = object class

Computer Vision Summer'19

Bounding Box Regressors

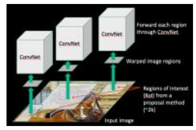
- Prediction of the 2D box
  - Necessary, since the proposal region might not fully coincide with the (annotated) object bounding box
  - Perform regression for location (x^*, y^*), width w^* and height h^*

$$\frac{x^* - x}{w} = w_{c,x}^T x_{pool5}$$
$$\frac{y^* - y}{h} = w_{c,y}^T x_{pool5}$$
$$\ln \frac{w^*}{w} = w_{c,w}^T x_{pool5}$$
$$\ln \frac{h^*}{h} = w_{c,w}^T x_{pool5}$$

  - Where x_{pool5} are the features from the pool5 layer of the network.

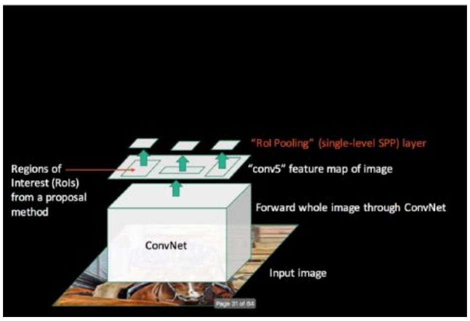
Problems with R-CNN

- Ad hoc training objectives
  - Fine tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressors (squared loss)
- Training (3 days) and testing (47s per image) is slow.
  - Many separate applications of region CNNs
- Takes a lot of disk space
  - Need to store all precomputed CNN features for training the classifiers
  - Easily 200GB of data



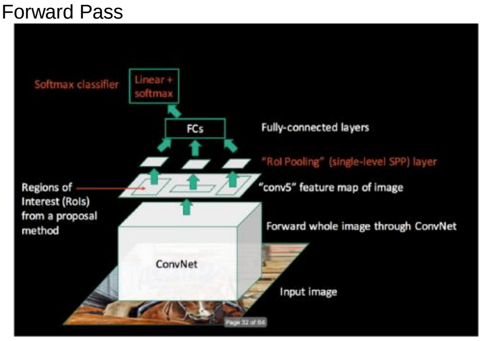
Fast R-CNN

- Forward Pass



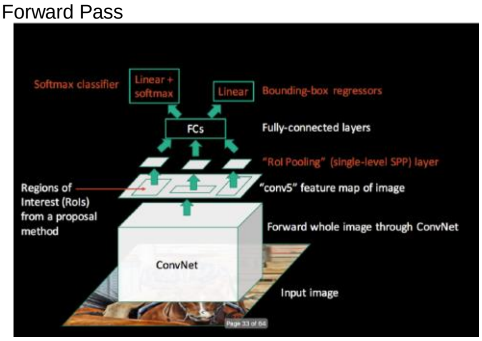
Fast R-CNN

- Forward Pass



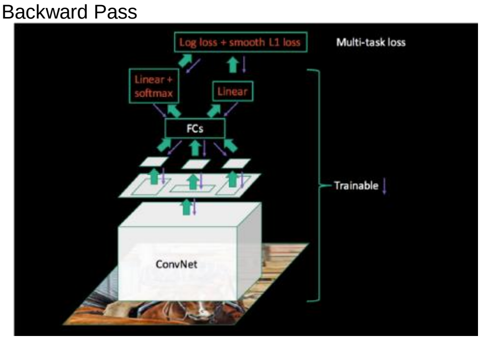
Fast R-CNN

- Forward Pass



Fast R-CNN Training

- Backward Pass



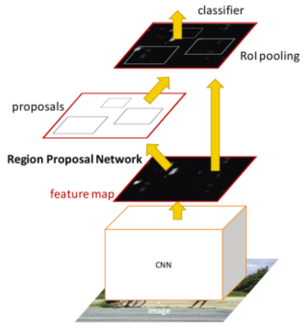
Region Proposal Networks (RPN)

- Idea
  - Remove dependence on external region proposal algorithm.
  - Instead, infer region proposals from same CNN.

  ⇒ Feature sharing

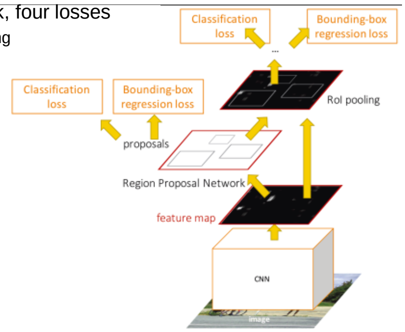
  ⇒ Object detection in a single pass becomes possible.

- Faster R-CNN = Fast R-CNN + RPN

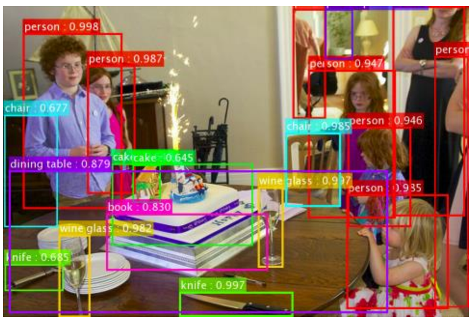


Faster R-CNN

- One network, four losses
  - Joint training



Faster R-CNN (based on ResNets)

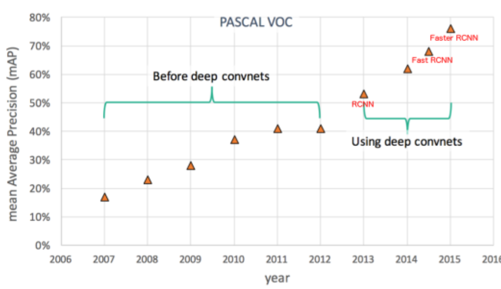


K. He, X. Zhang, S. Ren, J. Sun, [Deep Residual Learning for Image Recognition](), CVPR 2016.

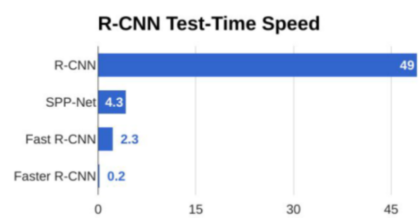
Faster R-CNN (based on ResNets)

K. He, X. Zhang, S. Ren, J. Sun, [Deep Residual Learning for Image Recognition](), CVPR 2016.

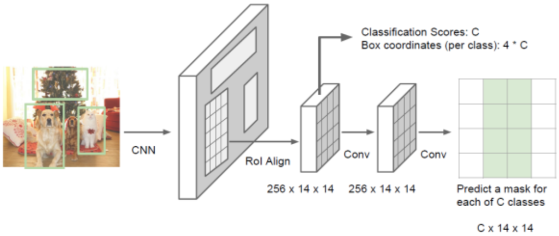
Object Detection Performance



Runtime Comparison



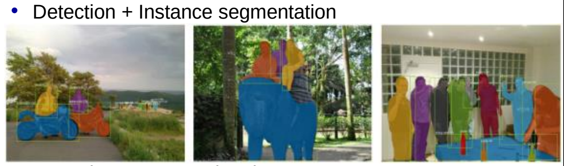
Most Recent Version: Mask R-CNN



K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, arXiv 1703.06870.

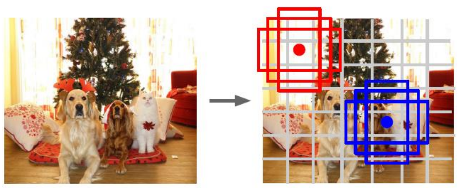
Mask R-CNN Results

- Detection + Instance segmentation



- Detection + Pose estimation

YOLO / SSD



- Idea: Directly go from image to detection scores
- Within each grid cell
  - Start from a set of anchor boxes
  - Regress from each of the B anchor boxes to a final box
  - Predict scores for each of C classes (including background)

YOLO-v3 Results

J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016.

Summary

- Object Detection
  - Find a variable number of objects by classifying image regions
  - Before CNNs: dense multiscale sliding window (HoG, DPM)
- Region proposal based detectors
  - Idea: Avoid dense sliding window with region proposals
  - R-CNN: Selective Search + CNN classification / regression
  - Fast R-CNN: Swap order of convolutions and region extraction
  - Faster R-CNN: Compute region proposals within the network
  - Mask R-CNN: Detection + instance segmentation + pose estimation
- Anchor box based detectors
  - Idea: Perform detection in a single step using grid of anchor boxes
  - YOLO, YOLO-v2, YOLO-v3
  - SSD

References and Further Reading

- LeNet
  - Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11): 2278–2324, 1998.
- AlexNet
  - A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012.
- VGGNet
  - K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015
- GoogLeNet
  - C. Szegedy, W. Liu, Y. Jia, et al, Going Deeper with Convolutions, arXiv:1409.4842, 2014.

References and Further Reading

- ResNet
  - K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, CVPR 2016.

References: Computer Vision Tasks

- Object Detection
  - R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014.
  - S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015.
  - K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, ICCV 2017.
  - J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, CVPR 2016
  - W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C-Y. Fu, A.C. Berg, SSD: Single Shot Multi Box Detector, ECCV 2016.