UNIVERSITAT DE BARCELONA

Object Detection and Segmentation

Meysam Madadi

So far: Image Classification

This image is CC0 public domain

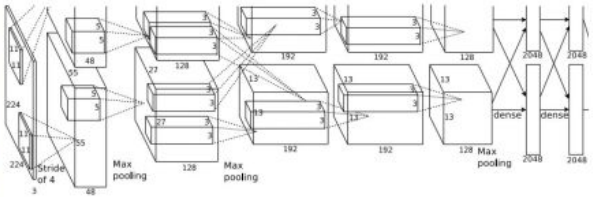


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Vector:
4096

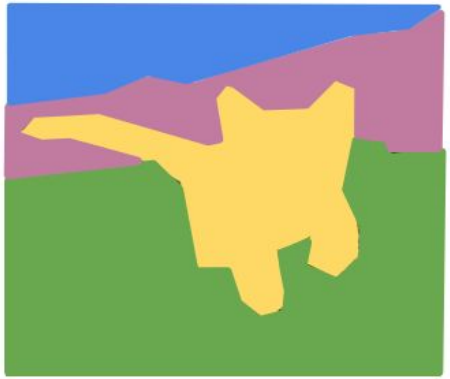
Fully-Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Other Computer Vision Tasks

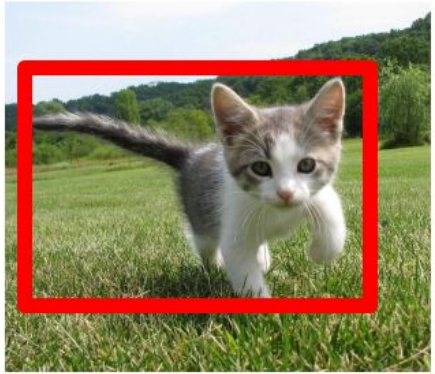
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

Classification + Localization



CAT

Single Object

Object Detection

DOG, DOG, CAT

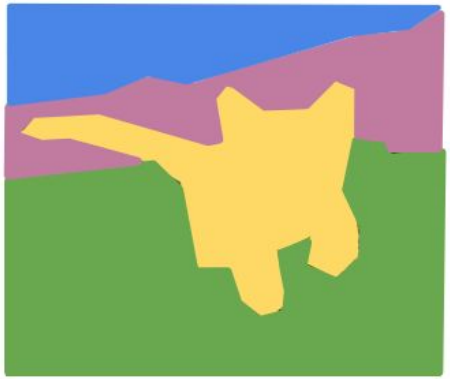
Instance Segmentation

DOG, DOG, CAT

Multiple Object

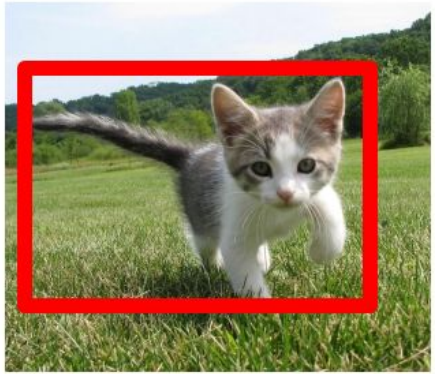
Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Semantic Segmentation



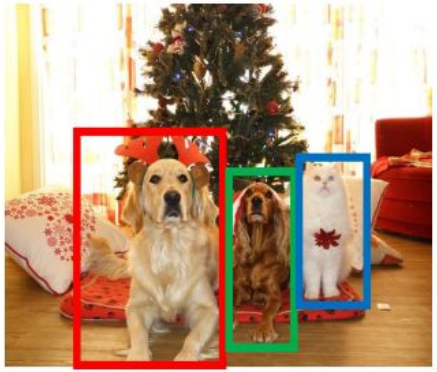
GRASS, CAT, TREE, SKY

No objects, just pixels



CAT

Single Object



DOG, DOG, CAT

DOG, DOG, CAT

Multiple Object

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Image segmentation

- Image segmentation is the process of **partitioning** an image into multiple **segments/super-pixels**, such that:
  - Every pixel is assigned a label such that pixels with the same label share certain characteristics like:
    - Color, intensity or texture,
  - Partitioning procedure does **not** attempt to understand what the segments represent.
- Possible solutions are:
  - Edge detection, color clustering/naming, graph partitioning, Markov Random Field, etc.

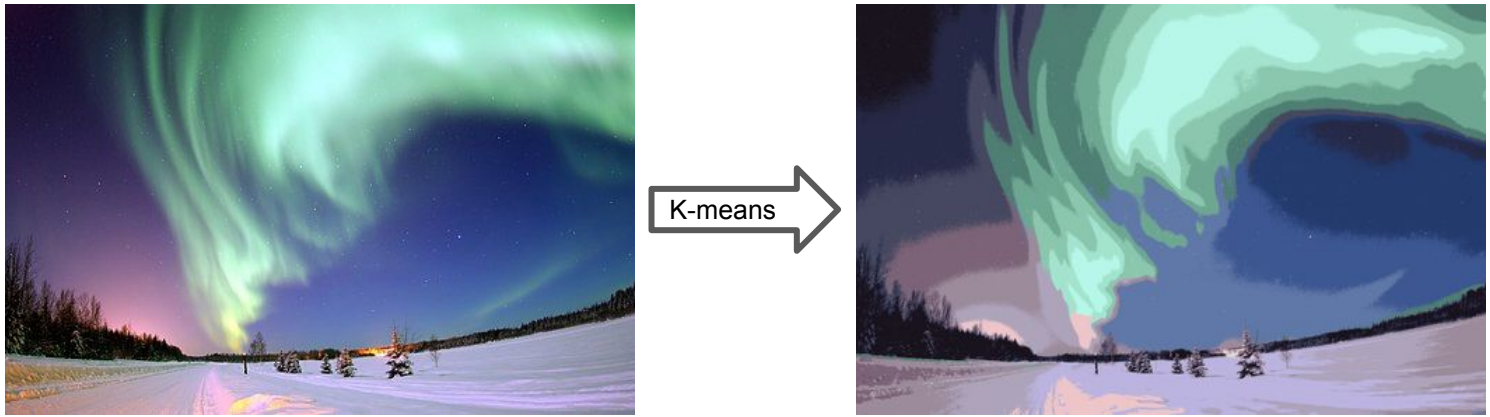


Image credit: https://en.wikipedia.org/wiki/Image_segmentation

Semantic segmentation

- Semantic segmentation is the process of partitioning an image into a set of semantically meaningful segments,
  - Can be seen as a classification task where each pixel is classified into several pre-defined classes.

0: Background/Unknown
1: Person
2: Purse
3: Plants/Grass
4: Sidewalk
5: Building/Structures

Image credit: https://www.jeremyjordan.me/semantic-segmentation/

Representing the task

0: Background/Unknown
1: Person
2: Purse
3: Plants/Grass
4: Sidewalk
5: Building/Structures

One-hot encoding

argmax

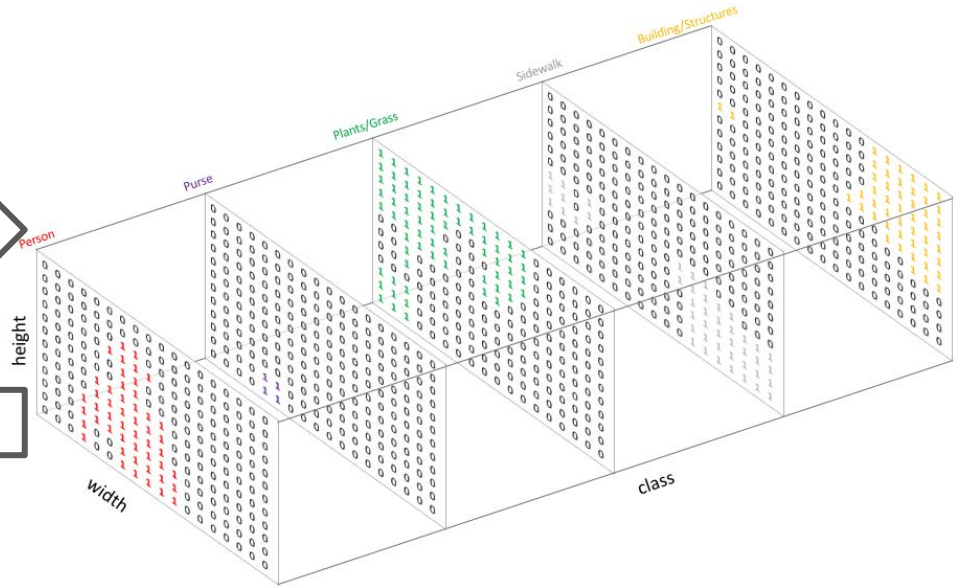
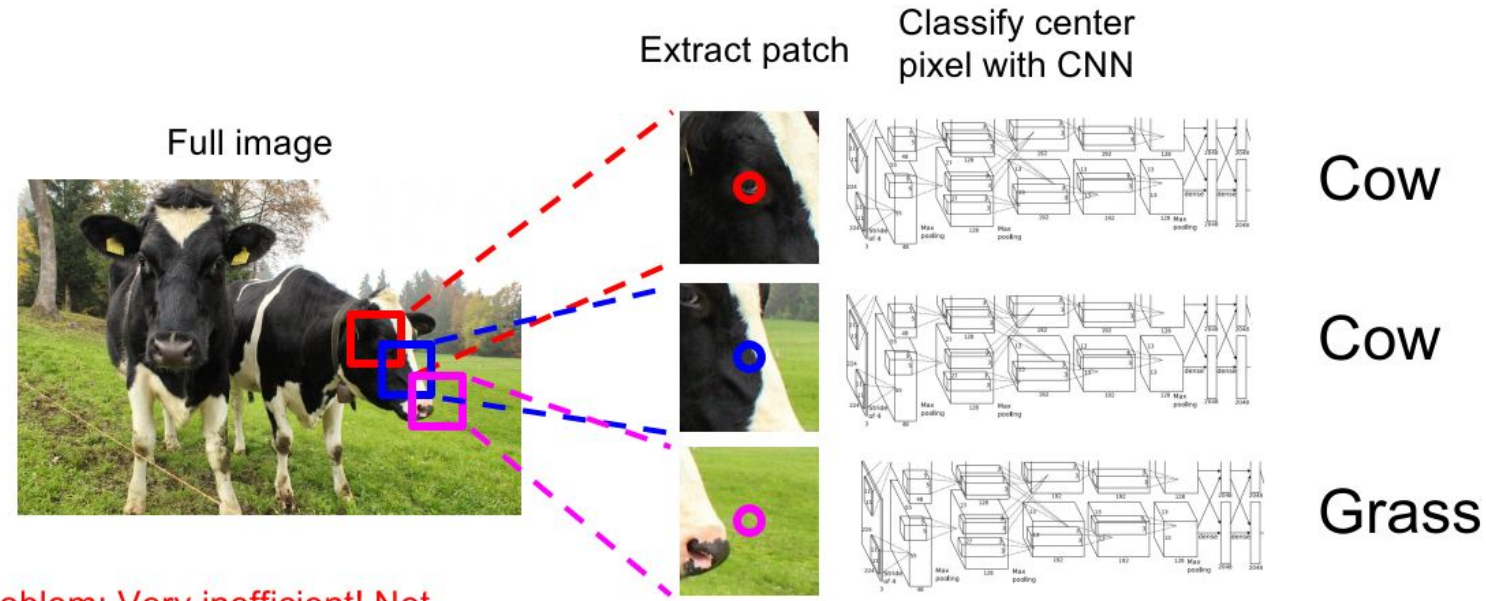


Image credit: https://www.jeremyjordan.me/semantic-segmentation/

Semantic Segmentation Idea: Sliding Window



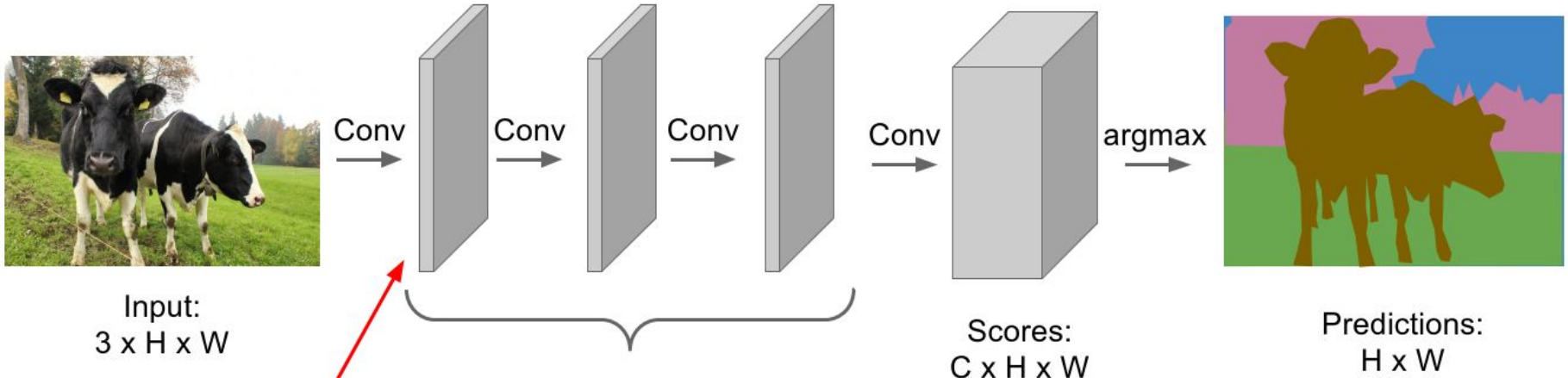
Problem: Very inefficient! Not reusing shared features between overlapping patches

Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Semantic Segmentation Idea: Fully Convolutional

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!



Problem: convolutions at original image resolution will be very expensive ...

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

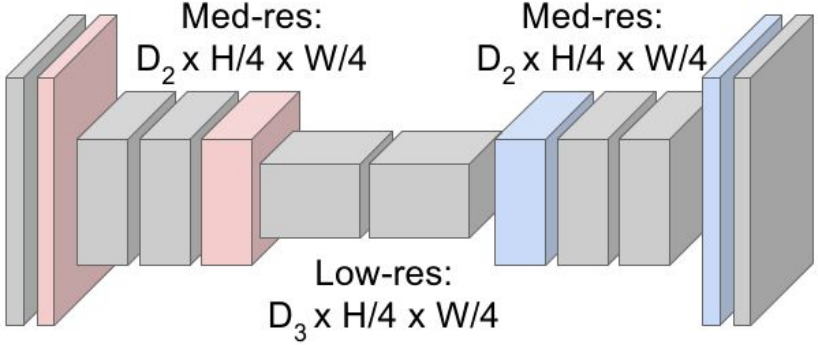
Semantic Segmentation Idea: Fully Convolutional

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
???

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

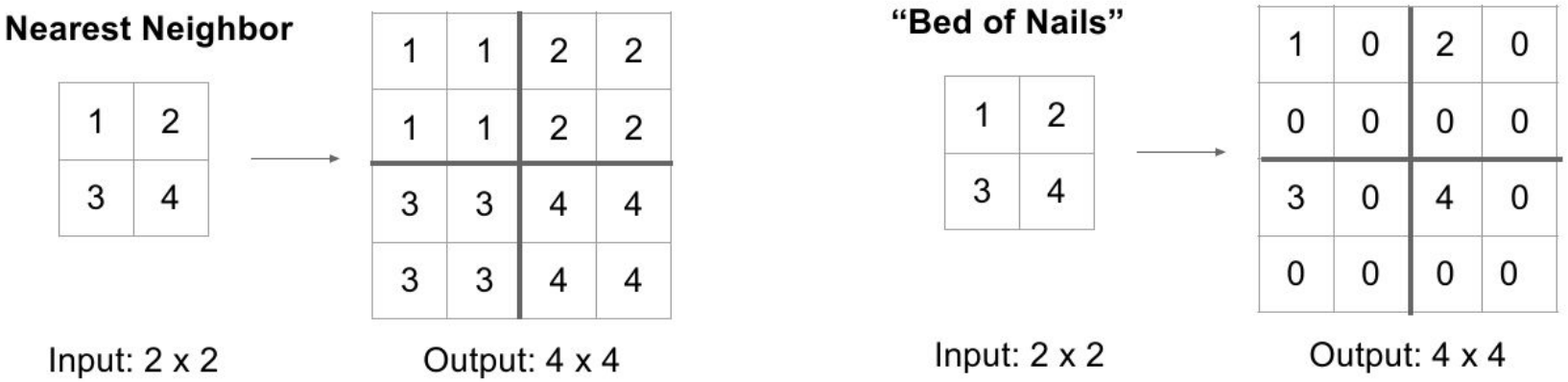
High-res:
D_1 x H/2 x W/2

Predictions:
H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

In-Network upsampling: “Unpooling”



Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

In-Network upsampling: “Max Unpooling”

Max Pooling
Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4

5	6
7	8

Output: 2 x 2

...

Rest of the network

Max Unpooling
Use positions from pooling layer

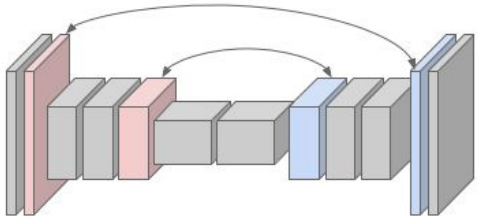
1	2
3	4

Input: 2 x 2

0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Output: 4 x 4

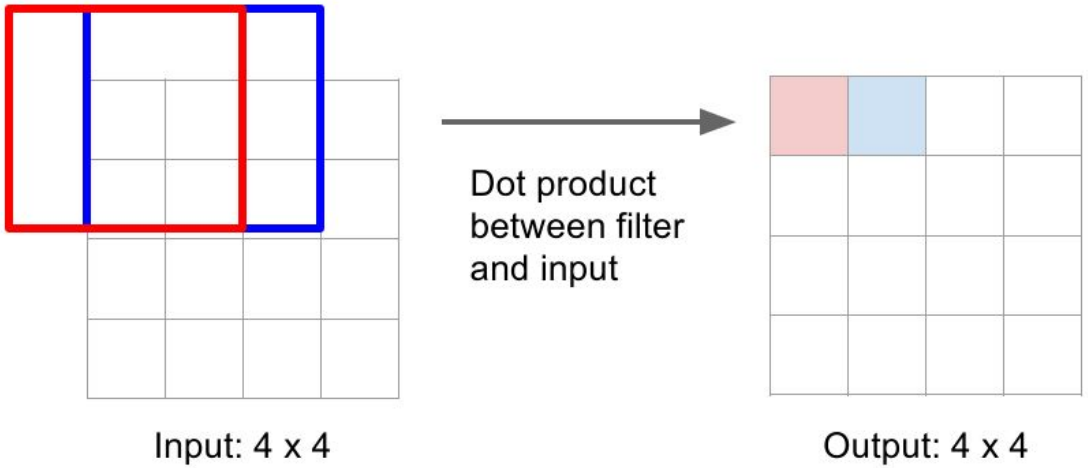
Corresponding pairs of downsampling and upsampling layers



Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

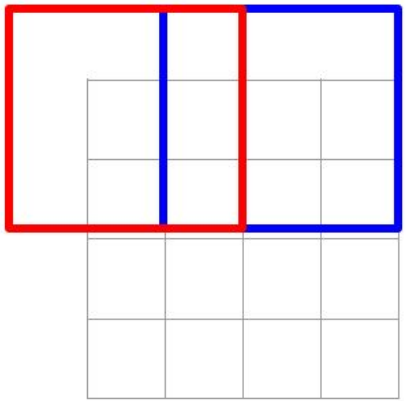
Recall: Normal 3 x 3 convolution, stride 1 pad 1



Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

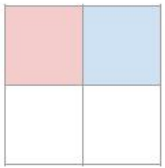
Learnable Upsampling: Transpose Convolution

Recall: Normal 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4

Dot product between filter and input



Output: 2 x 2

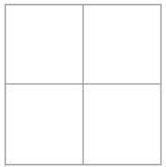
Filter moves 2 pixels in the input for every one pixel in the output

Stride gives ratio between movement in input and output

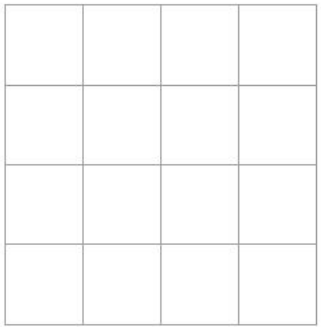
Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

3 x 3 **transpose** convolution, stride 2 pad 1



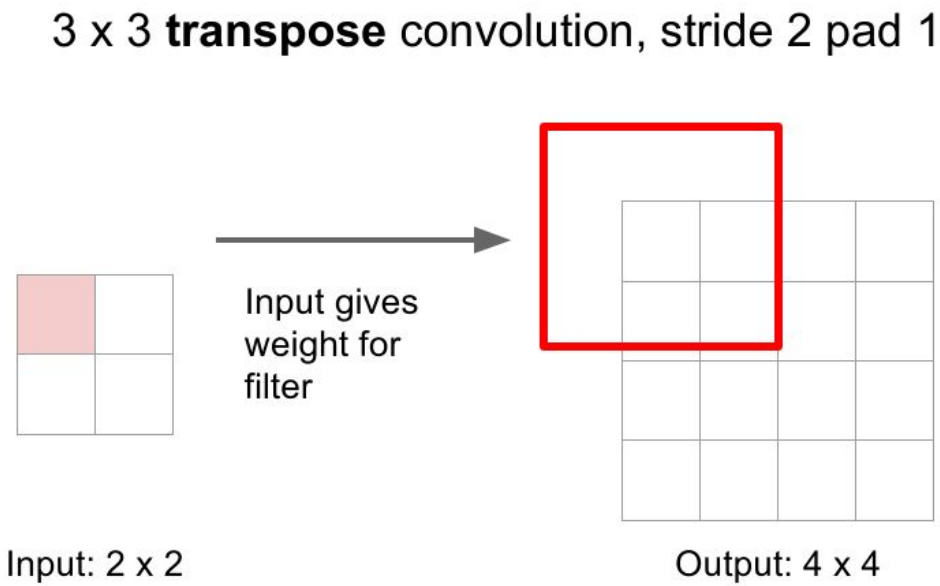
Input: 2 x 2



Output: 4 x 4

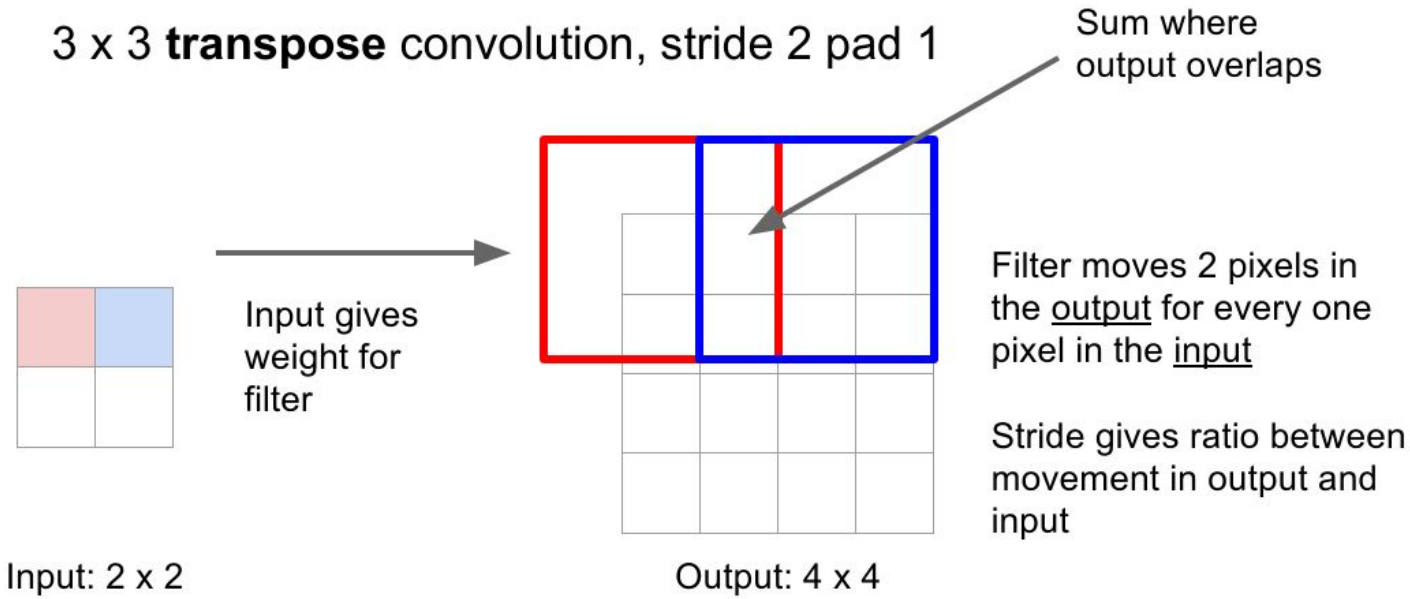
Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution



Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

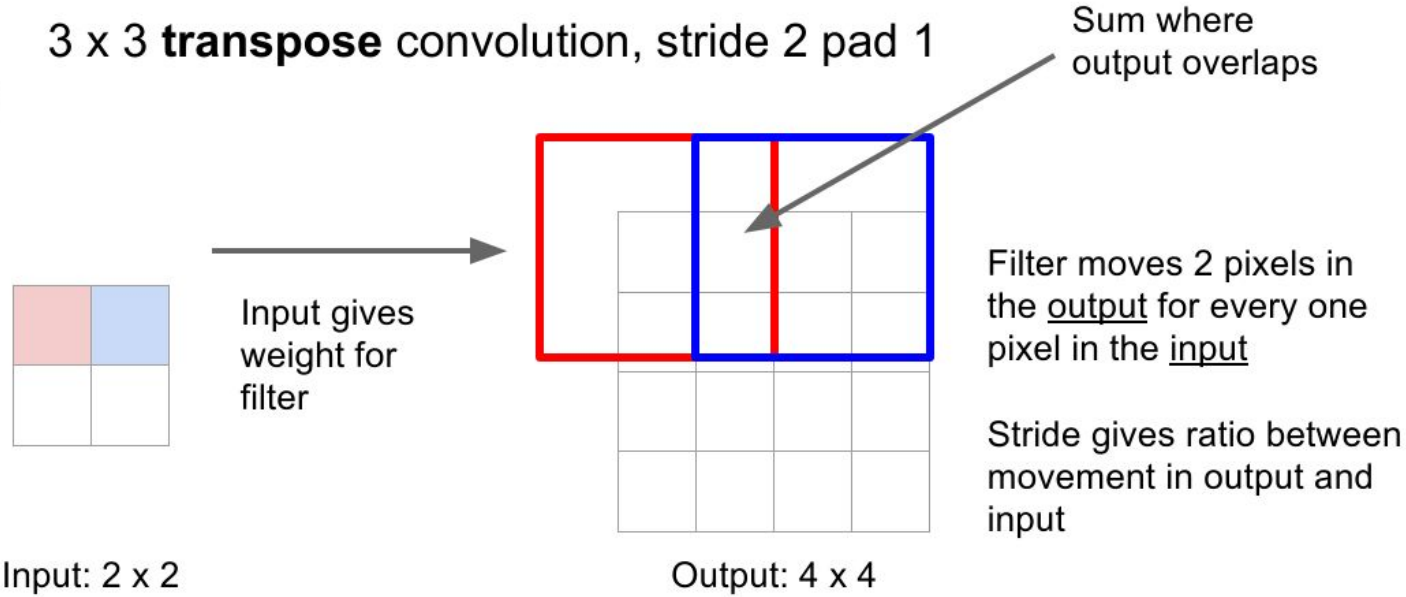


Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Learnable Upsampling: Transpose Convolution

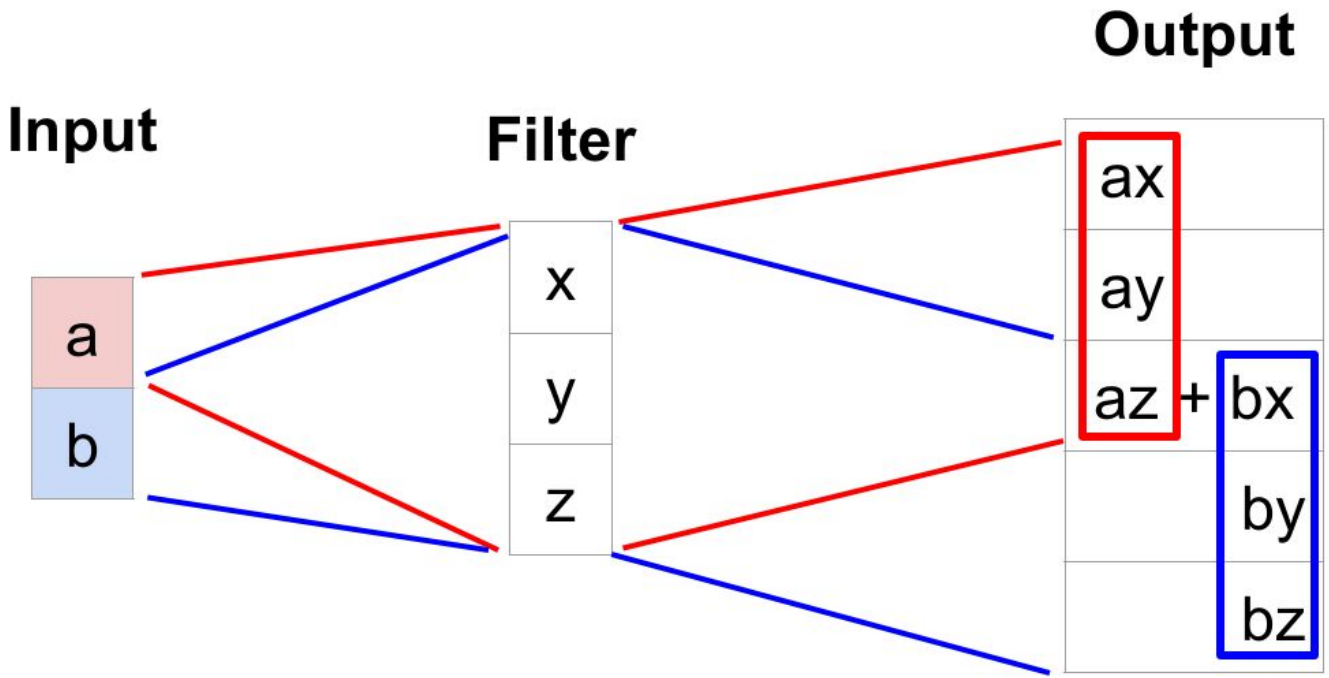
Other names:
-Deconvolution (bad)
-Upconvolution
-Fractionally strided convolution
-Backward strided convolution

3 x 3 **transpose** convolution, stride 2 pad 1



Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Transpose Convolution: 1D Example

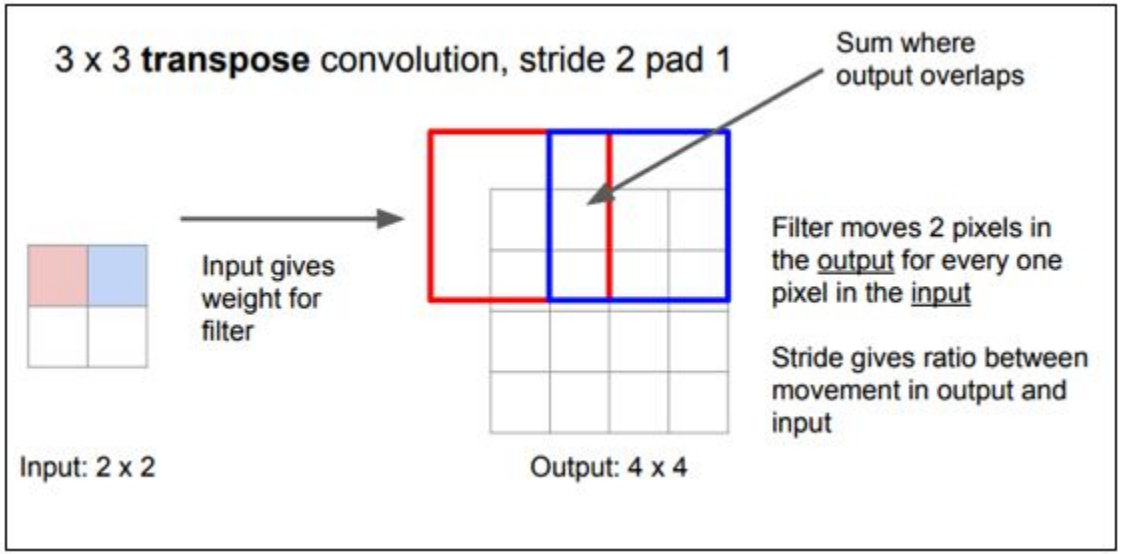


Output contains copies of the filter weighted by the input, summing at where at overlaps in the output

Need to crop one pixel from output to make output exactly 2x input

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Upsampling - transpose convolutions



Downside: Chekerboard artifact due to overlapped windows

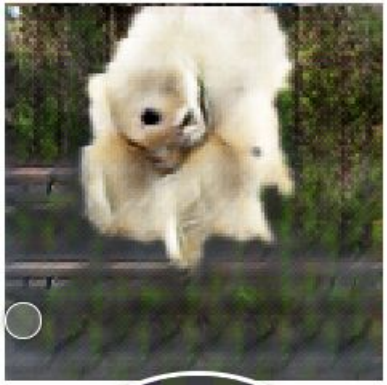
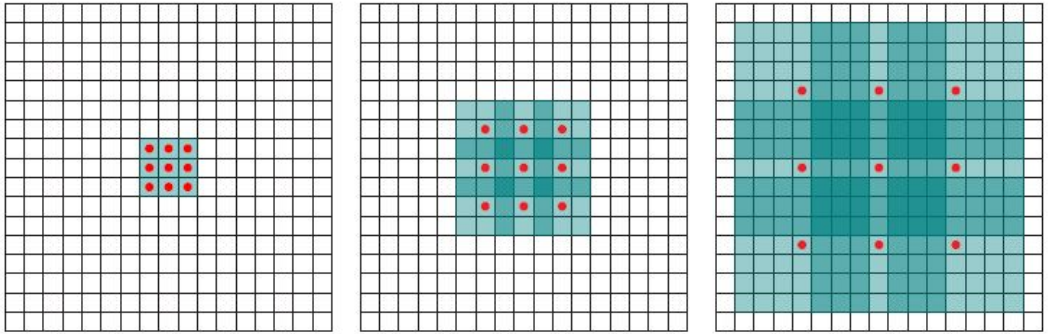


Image credit: https://distill.pub/2016/deconv-checkerboard/

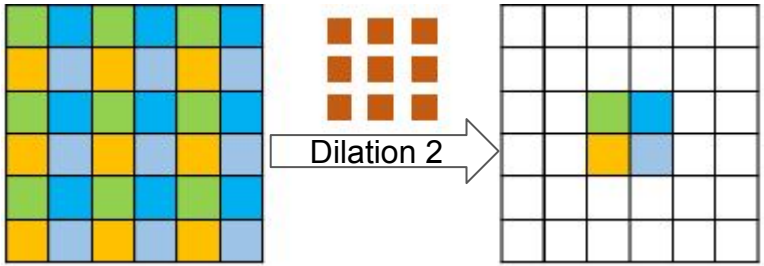
Dilated convolution

Red=Conv kernel, Green=receptive field

Normal Conv=Dilation 1 Dilation 2 Dilation 4



Upside: Supports exponential receptive field expansion without loss of resolution or coverage.



Downside: Has gridding artifacts for high frequency inputs.

Image credit: https://towardsdatascience.com/review-drn-dilated-residual-networks-image-classification-semantic-segmentation-d527e1a8fb5

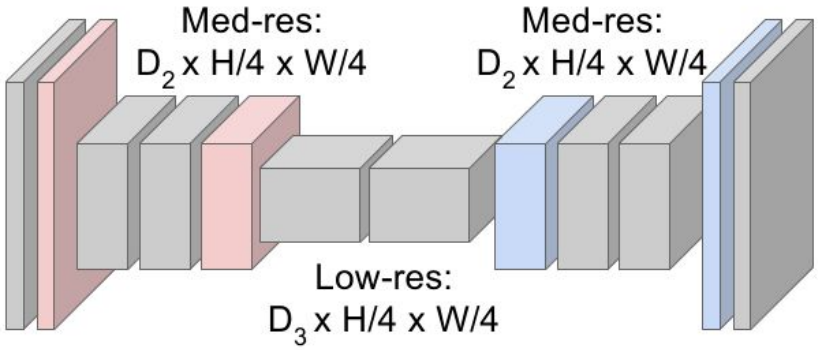
Semantic Segmentation Idea: Fully Convolutional

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
Unpooling or strided transpose convolution

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

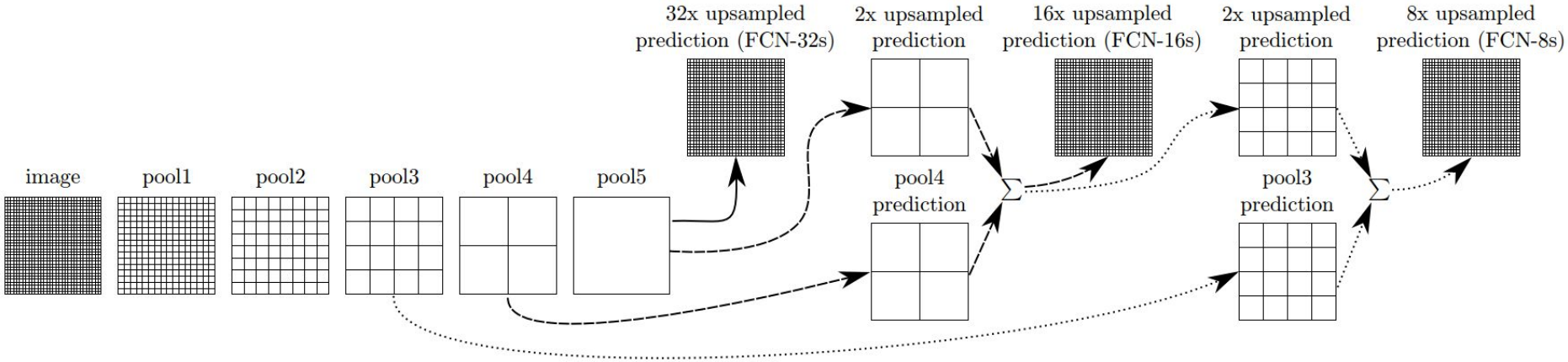
High-res:
D_1 x H/2 x W/2

Predictions:
H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Skip connections ===> more fine grain details



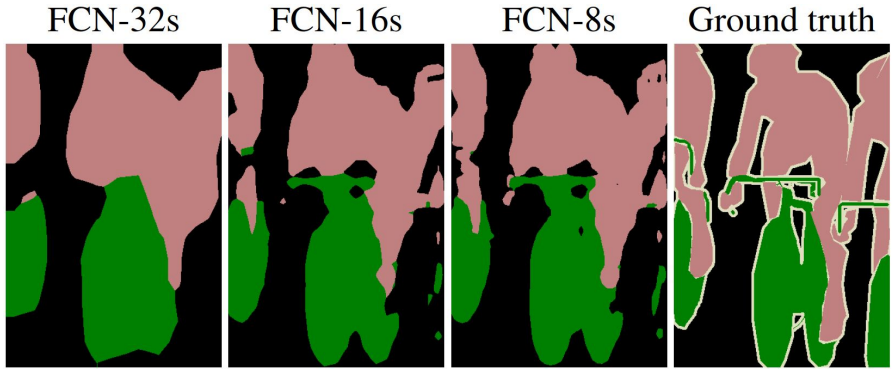
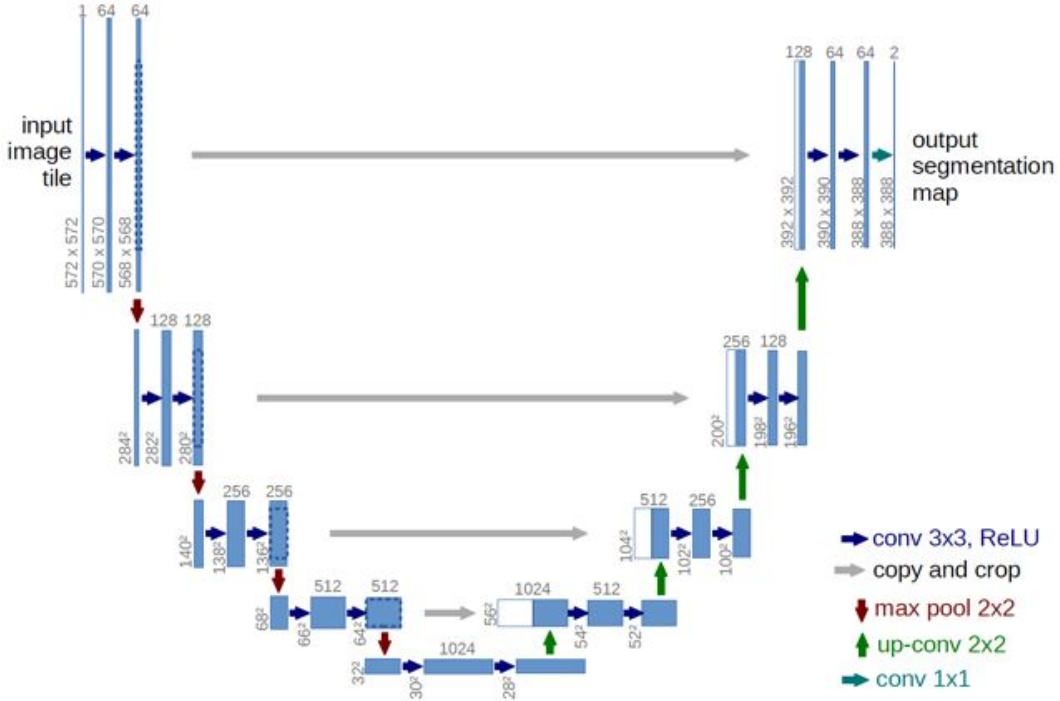


Image credit: https://arxiv.org/abs/1411.4038

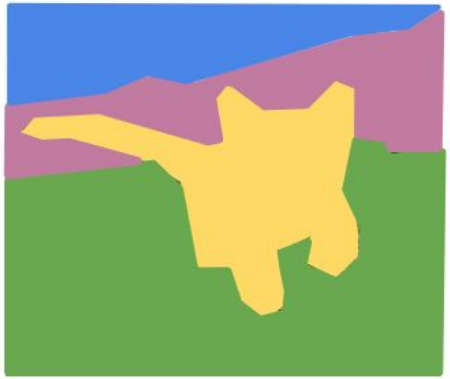
U-Net: deeper decoder with skip connection



- Simple to understand,
- Popular in many domains,
- Efficient on small dataset by the usage of data augmentation,
- Invariant to image scale due to fully convolutional layers,
- Decoder has more features than encoder.

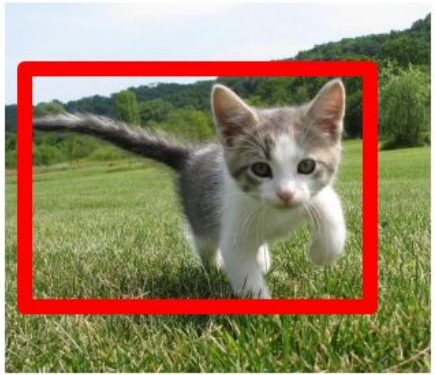
Image credit: https://arxiv.org/abs/1505.04597

Classification + Localization



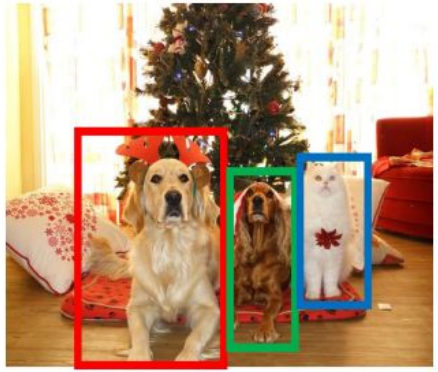
GRASS, CAT, TREE, SKY

No objects, just pixels



CAT

Single Object



DOG, DOG, CAT

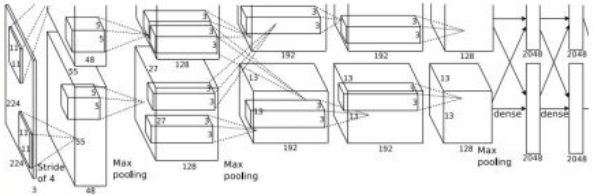
DOG, DOG, CAT

Multiple Object

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Classification + Localization

This image is CC0 public domain



Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Vector: 4096

Fully Connected: 4096 to 4

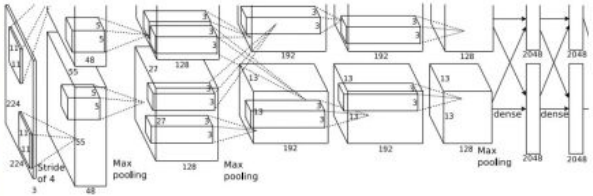
Box Coordinates
(x, y, w, h)

Treat localization as a regression problem!

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Classification + Localization

This image is CC0 public domain



Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax Loss

Correct label:
Cat

Vector: 4096

Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

L2 Loss

Correct box:
(x', y', w', h')

Multitask Loss

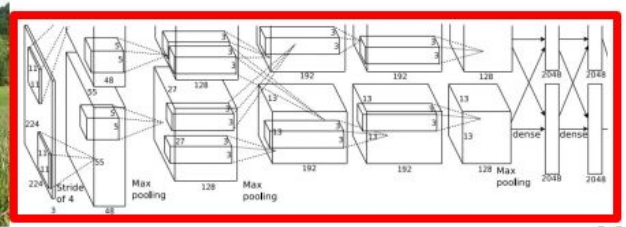
+ **Loss**

Treat localization as a regression problem!

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Classification + Localization

This image is CC0 public domain



Often pretrained on ImageNet
(Transfer learning)

Vector:
4096

Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label:
Cat

Softmax Loss

L2 Loss

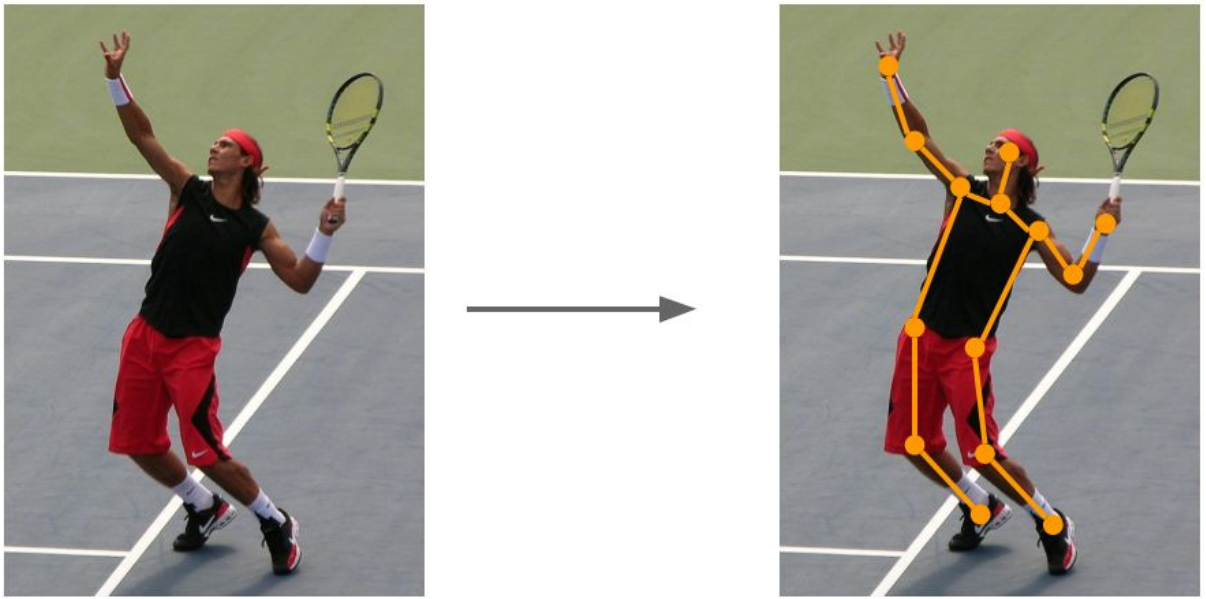
Loss

Correct box:
(x', y', w', h')

Treat localization as a regression problem!

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Aside: Human Pose Estimation



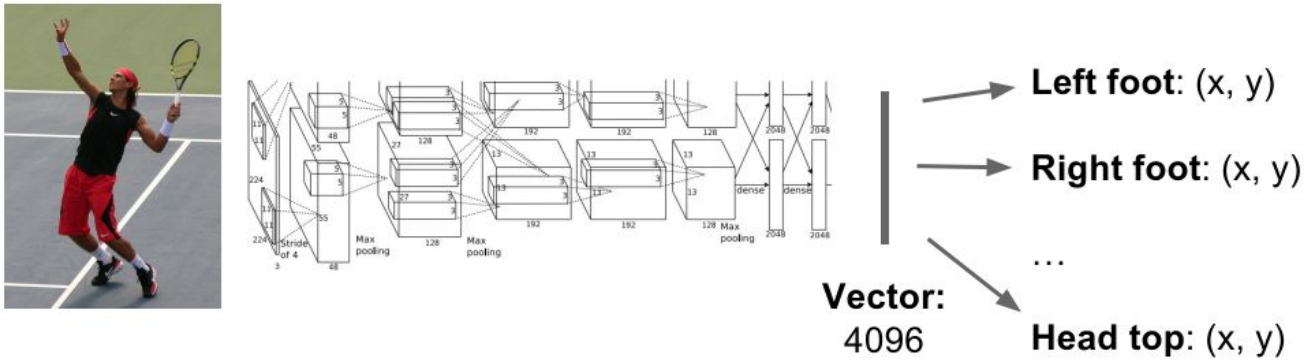
Represent pose as a set of 14 joint positions:

Left / right foot
Left / right knee
Left / right hip
Left / right shoulder
Left / right elbow
Left / right hand
Neck
Head top

Johnson and Everingham, "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation", BMVC 2010

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

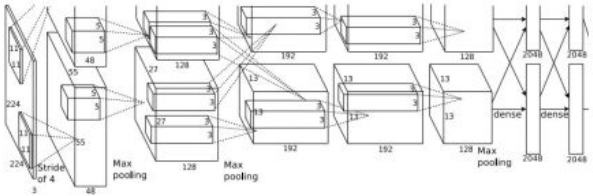
Aside: Human Pose Estimation

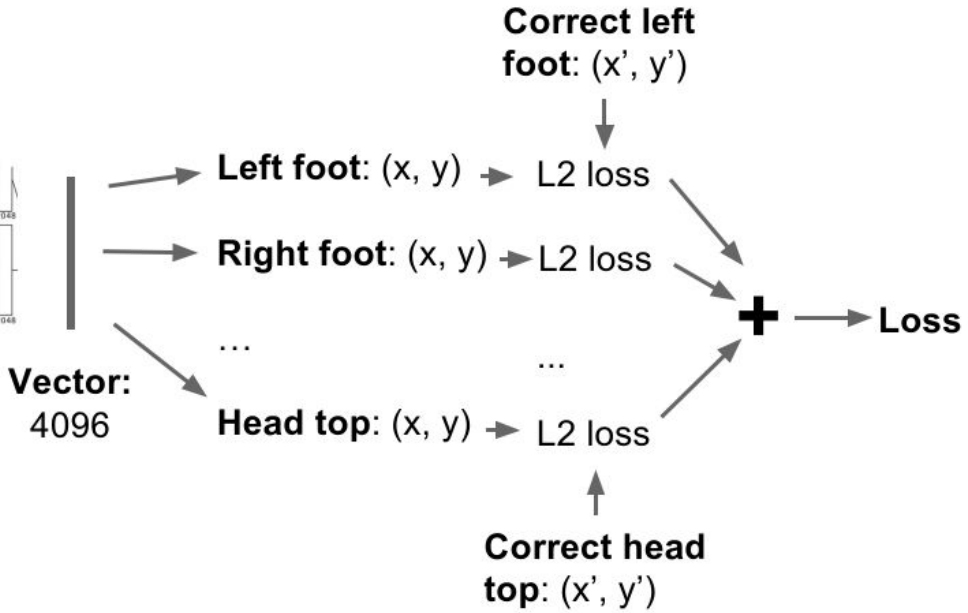


Toshev and Szegedy, “DeepPose: Human Pose Estimation via Deep Neural Networks”, CVPR 2014

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Aside: Human Pose Estimation

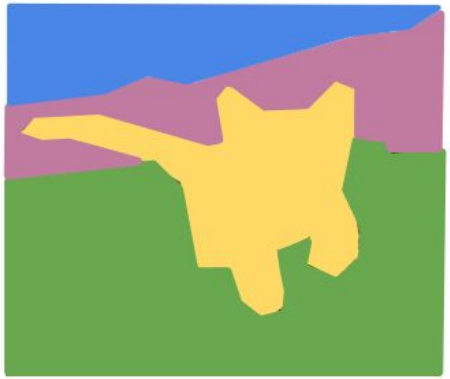




Toshev and Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks", CVPR 2014

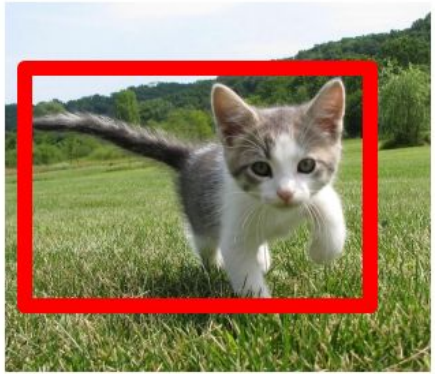
Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Object Detection



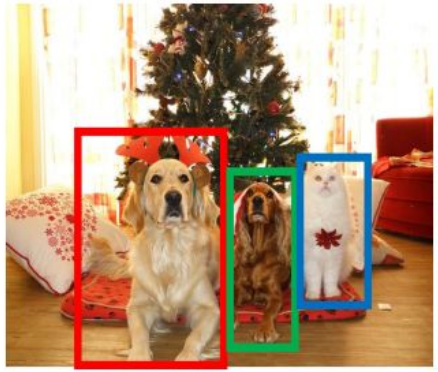
GRASS, CAT, TREE, SKY

No objects, just pixels



CAT

Single Object



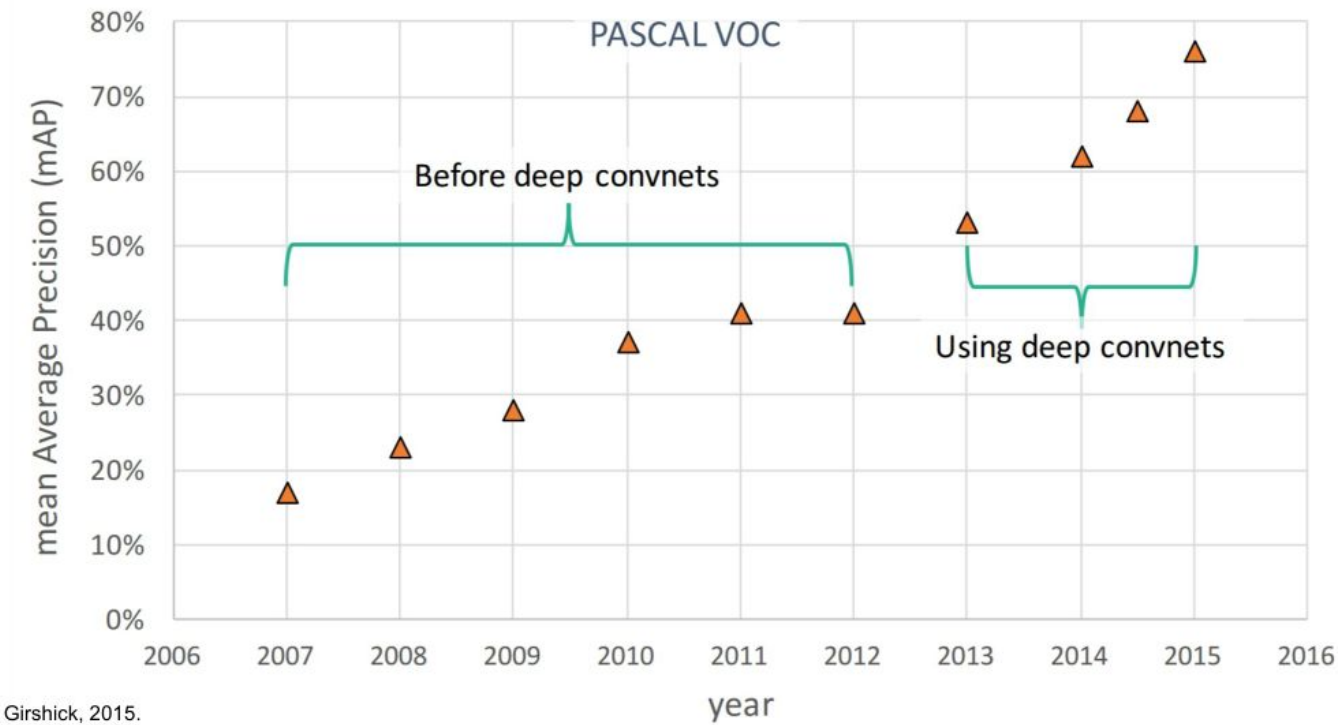
DOG, DOG, CAT

DOG, DOG, CAT

Multiple Object

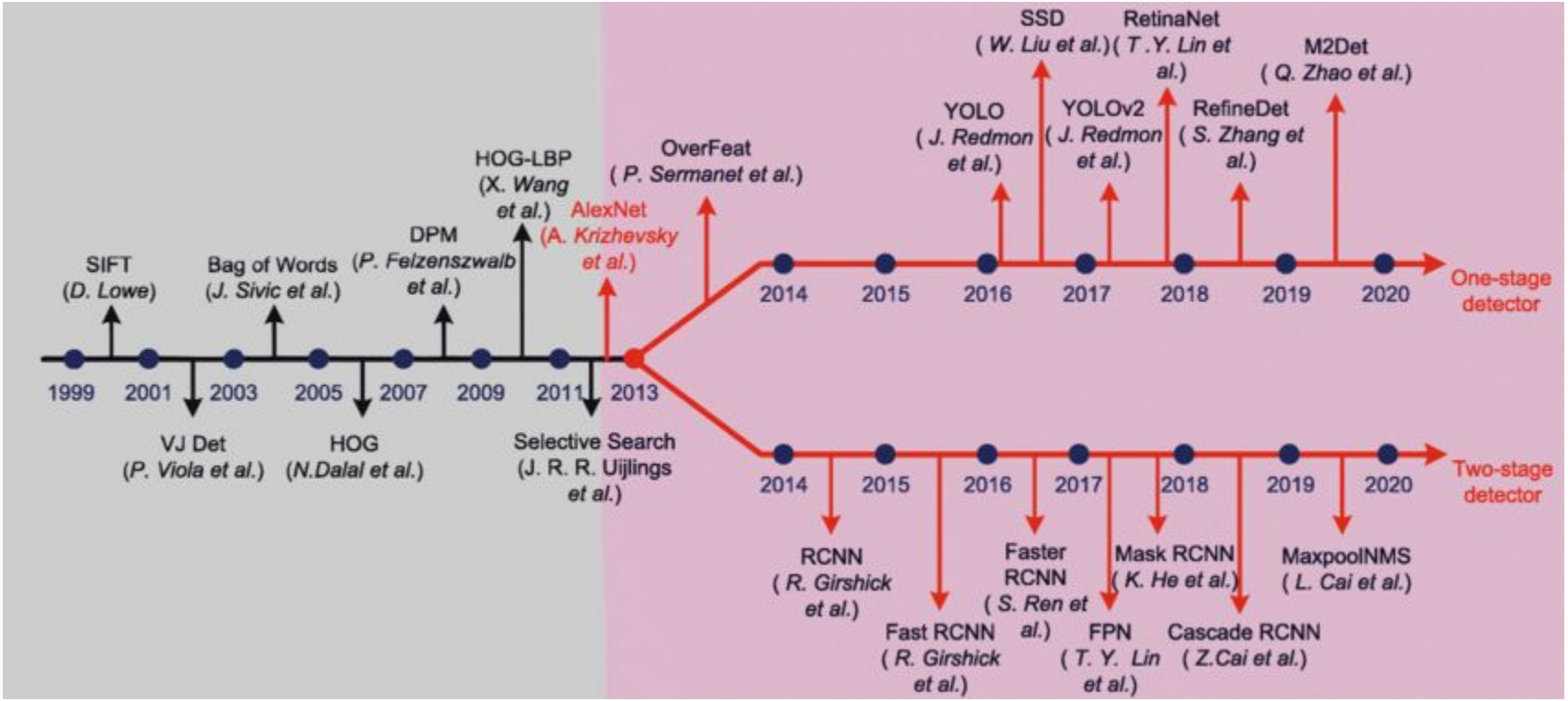
Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Object Detection: Impact of Deep Learning

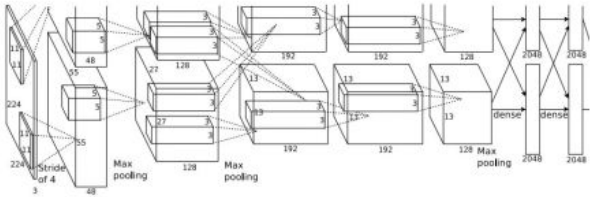


Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

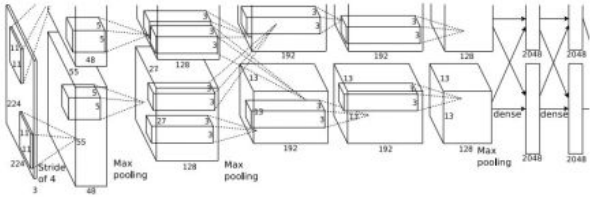
Evolution of object detection



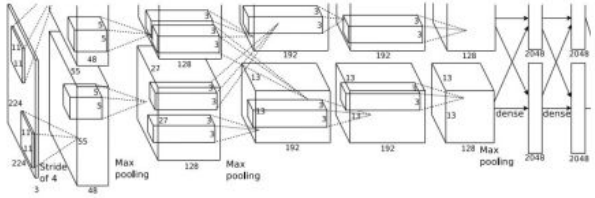
Object Detection as Regression?



CAT: (x, y, w, h)



DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

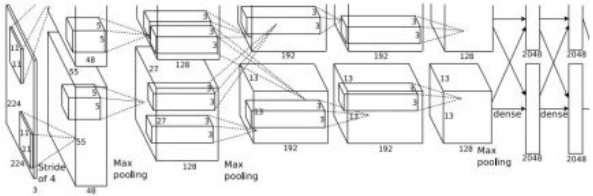


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

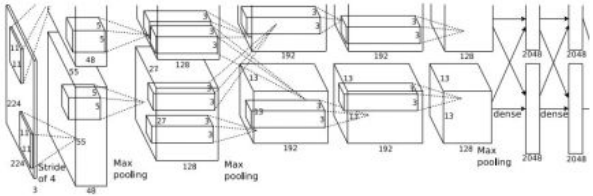
Object Detection as Regression?

Each image needs a different number of outputs!



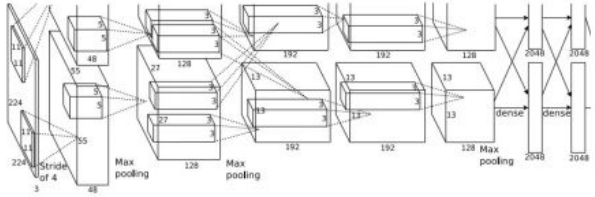
CAT: (x, y, w, h)

4 numbers



DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

16 numbers



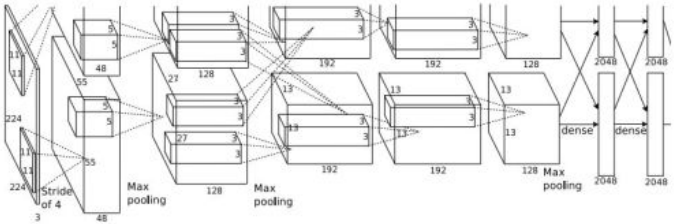
DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….

Many numbers!

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Object Detection as Classification: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

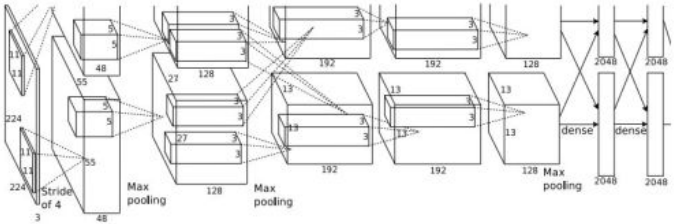


Dog? NO
Cat? NO
Background? YES

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Object Detection as Classification: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



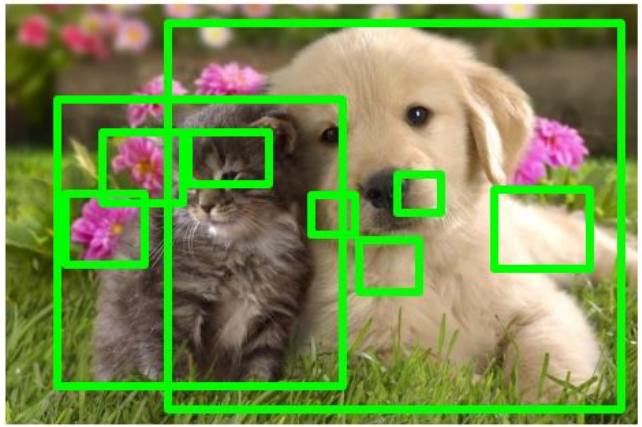
Dog? NO
Cat? YES
Background? NO

Problem: Need to apply CNN to huge number of locations and scales, very computationally expensive!

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Region Proposals

- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 1000 region proposals in a few seconds on CPU



Alexe et al, “Measuring the objectness of image windows”, TPAMI 2012
Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013
Cheng et al, “BING: Binarized normed gradients for objectness estimation at 300fps”, CVPR 2014
Zitnick and Dollar, “Edge boxes: Locating object proposals from edges”, ECCV 2014

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

R-CNN

Input image

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

R-CNN

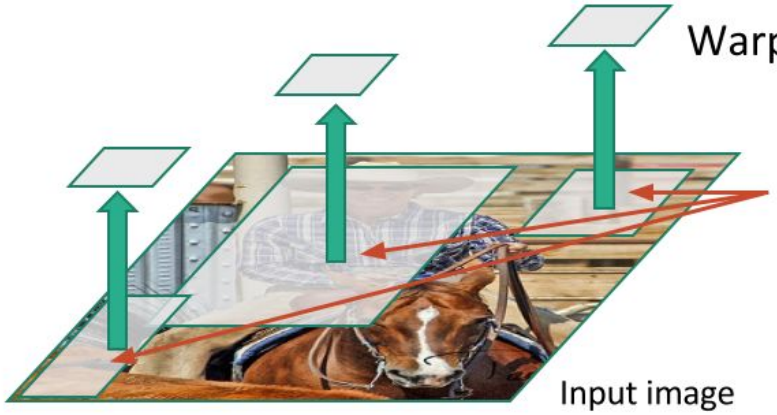
Input image

Regions of Interest (RoI) from a proposal method (~2k)

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

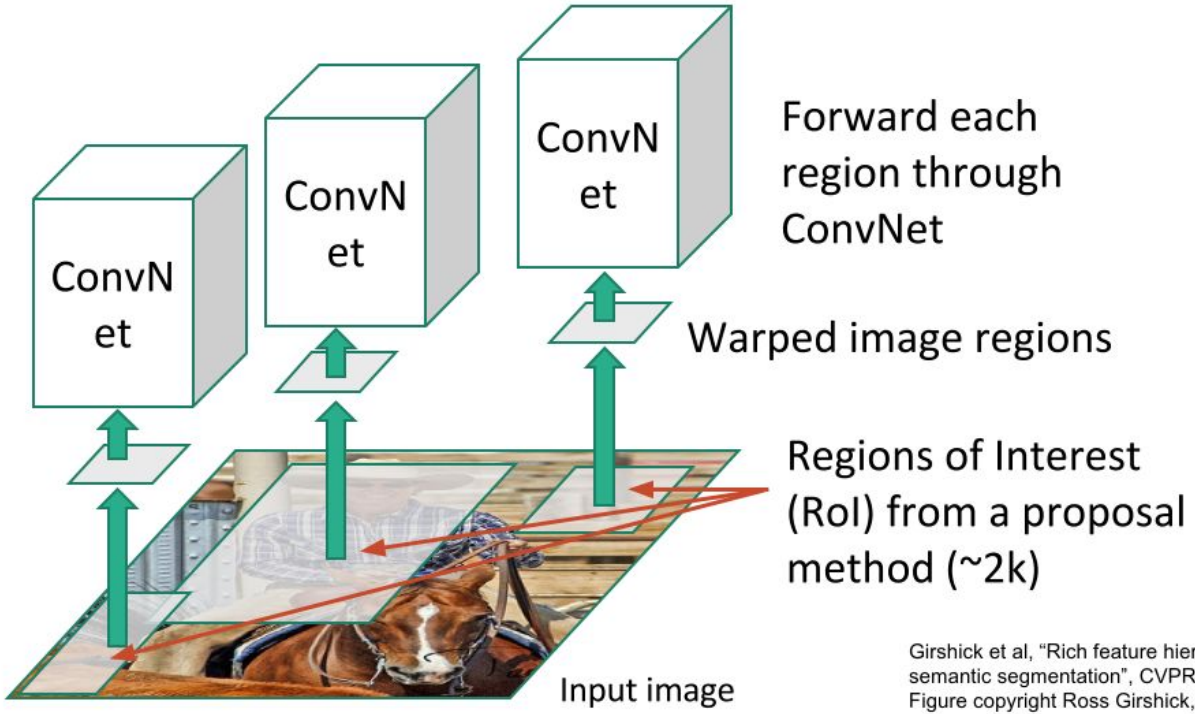
R-CNN



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

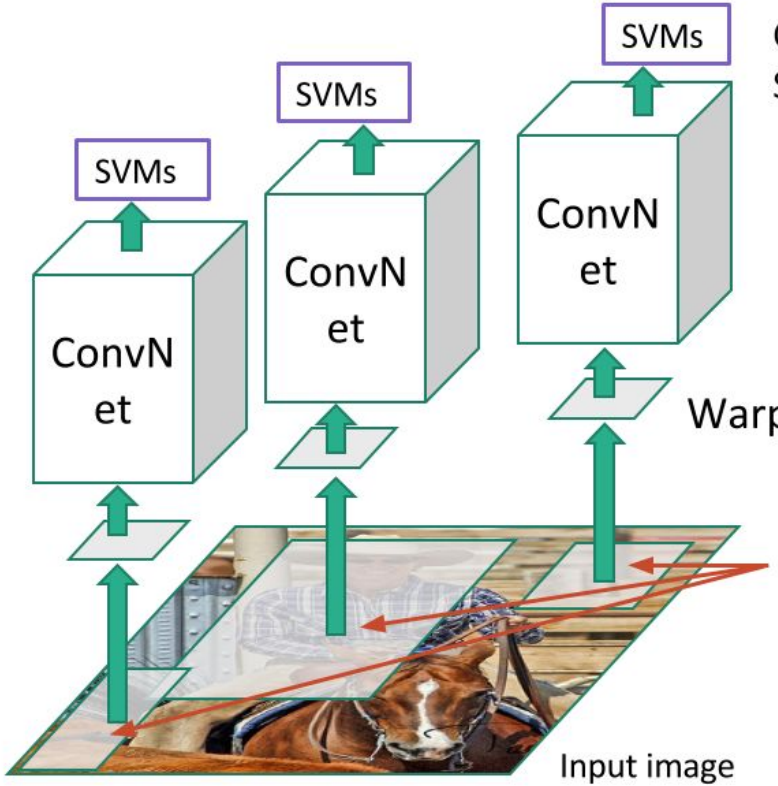
R-CNN



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

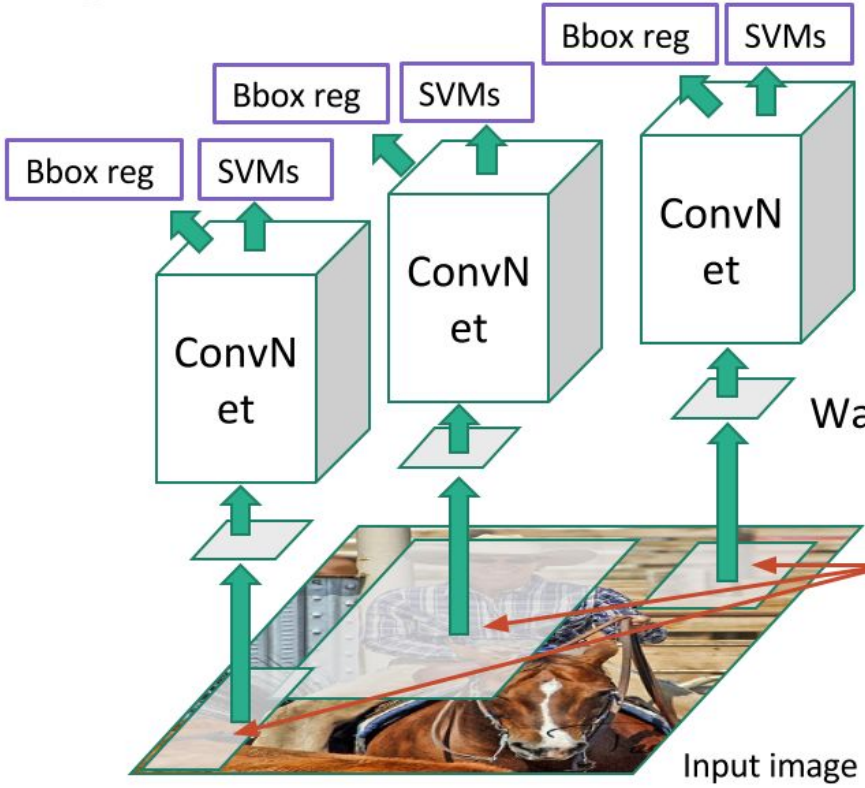
R-CNN



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

R-CNN



Linear Regression for bounding box offsets

Classify regions with SVMs

Forward each region through ConvNet

Warped image regions

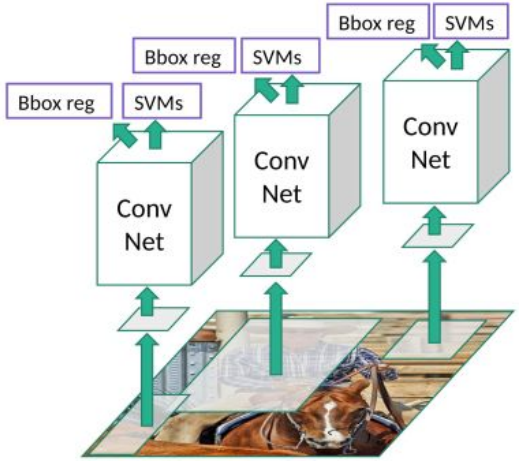
Regions of Interest (RoI) from a proposal method (~2k)

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

R-CNN: Problems

- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressions (least squares)
- Training is slow (84h), takes a lot of disk space
- Inference (detection) is slow
  - 47s / image with VGG16 [Simonyan & Zisserman. ICLR15]
  - Fixed by SPP-net [He et al. ECCV14]



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Slide copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

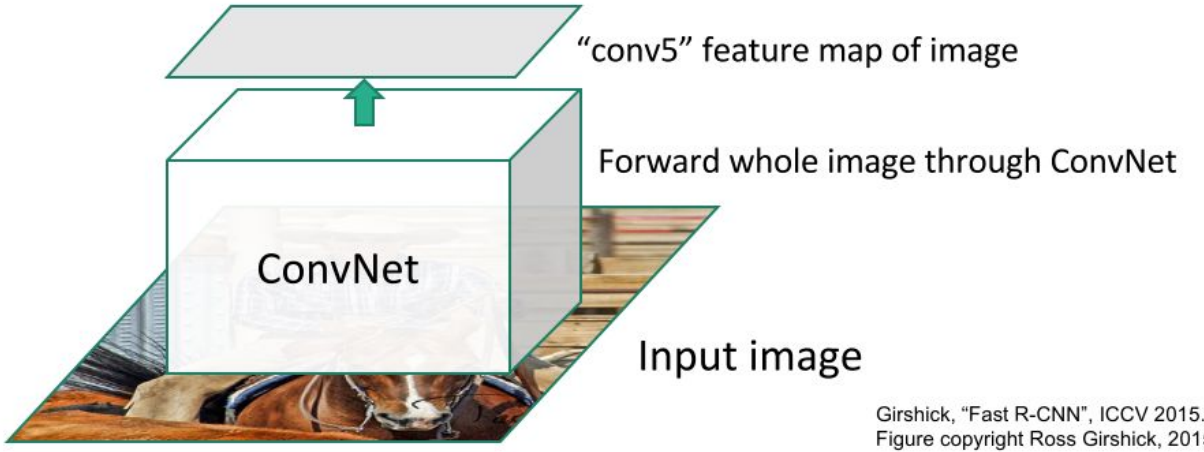
Fast R-CNN

Input image

Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

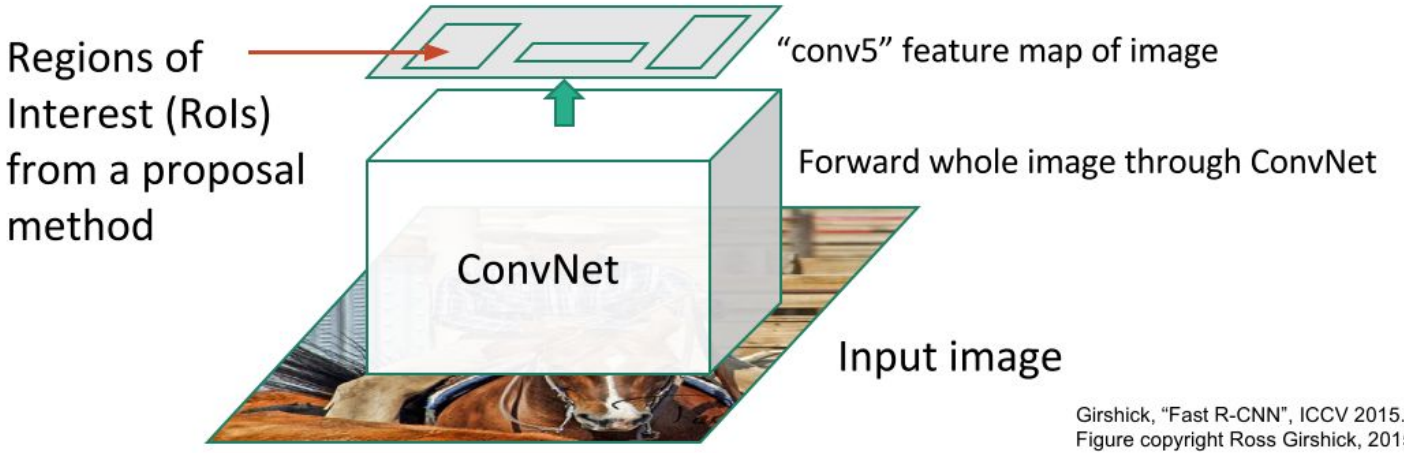
Fast R-CNN



Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

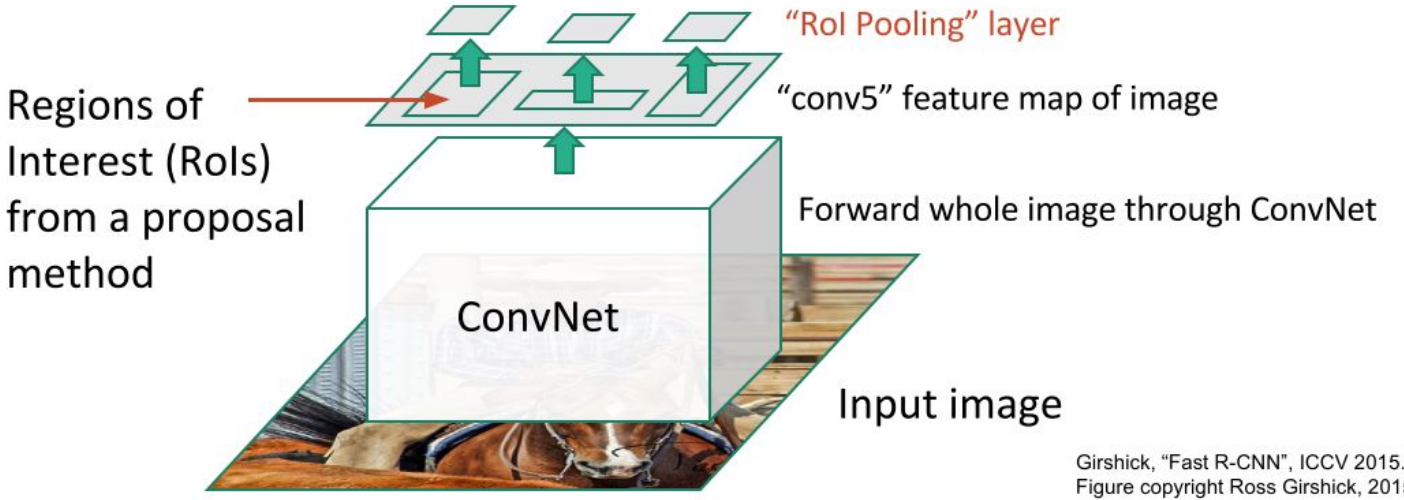
Fast R-CNN



Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

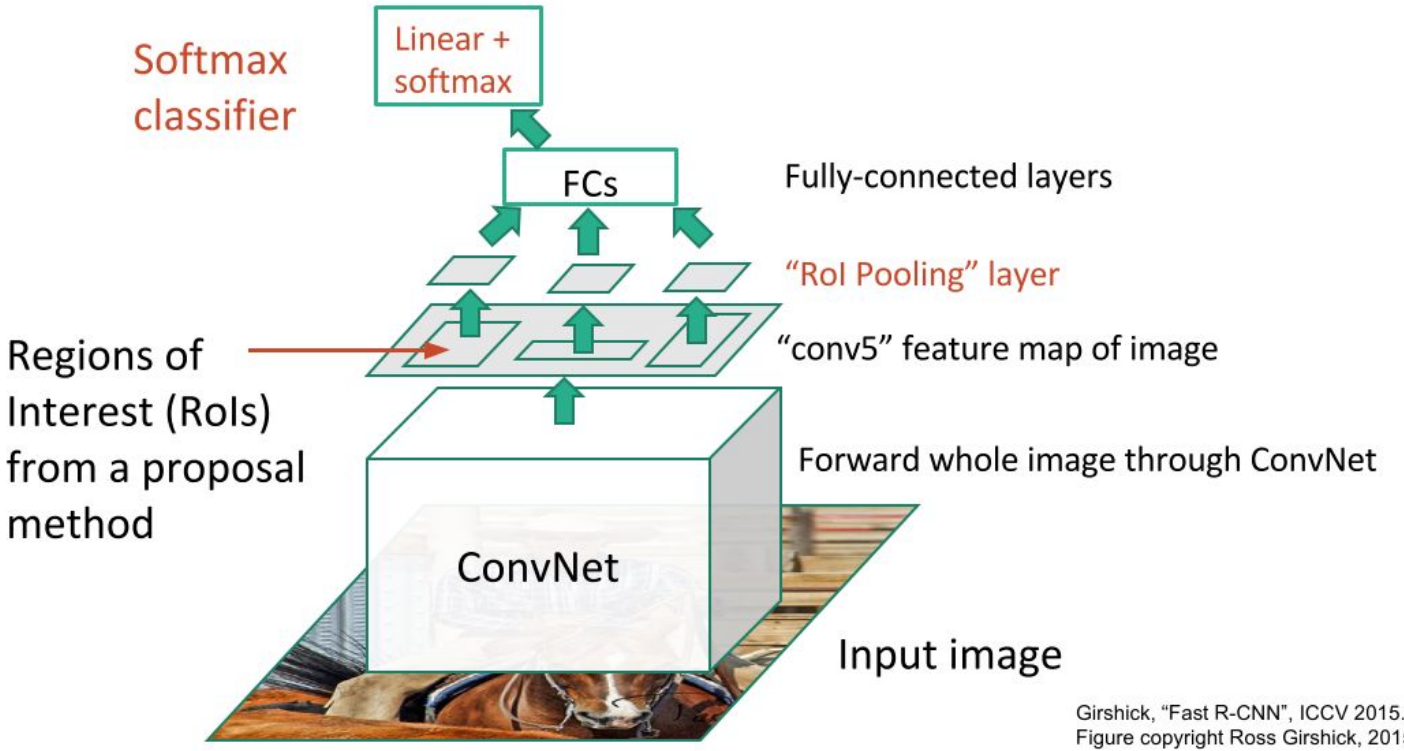
Fast R-CNN



Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

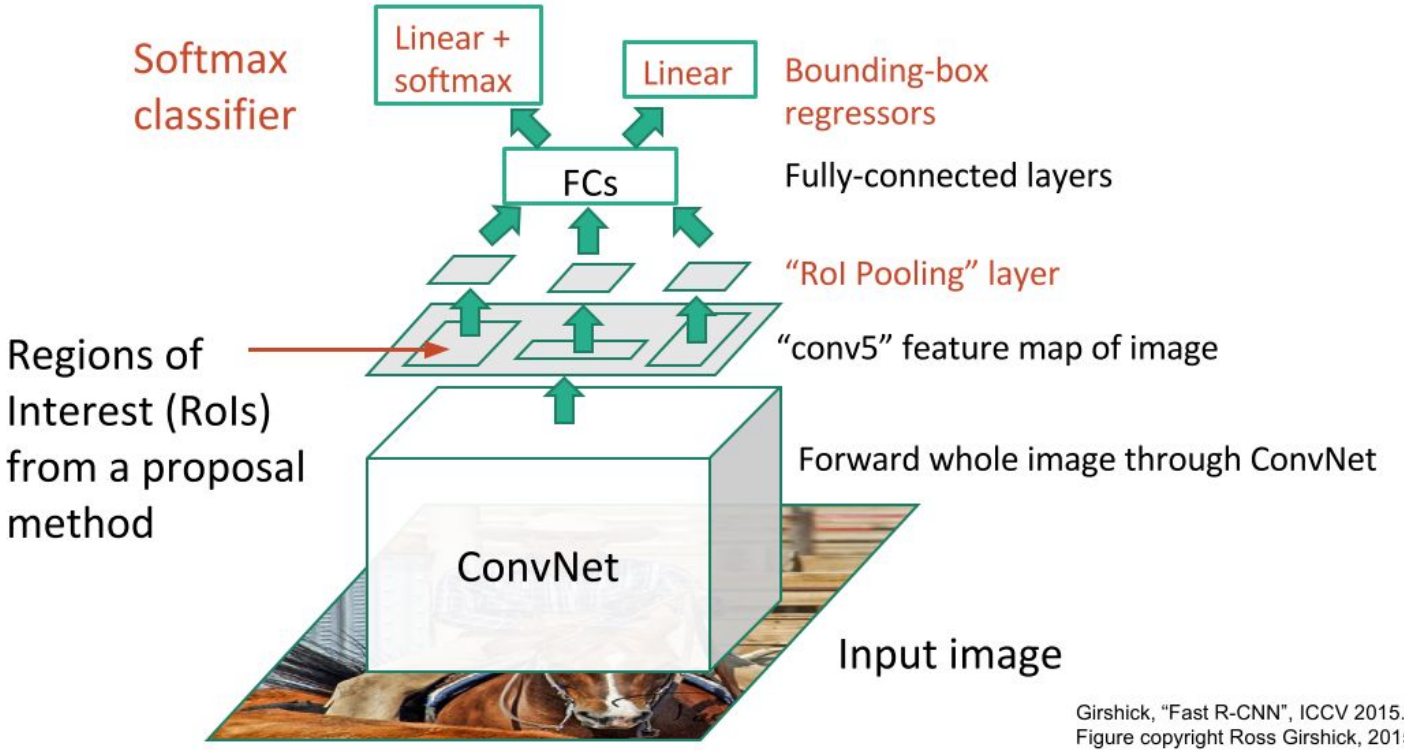
Fast R-CNN



Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

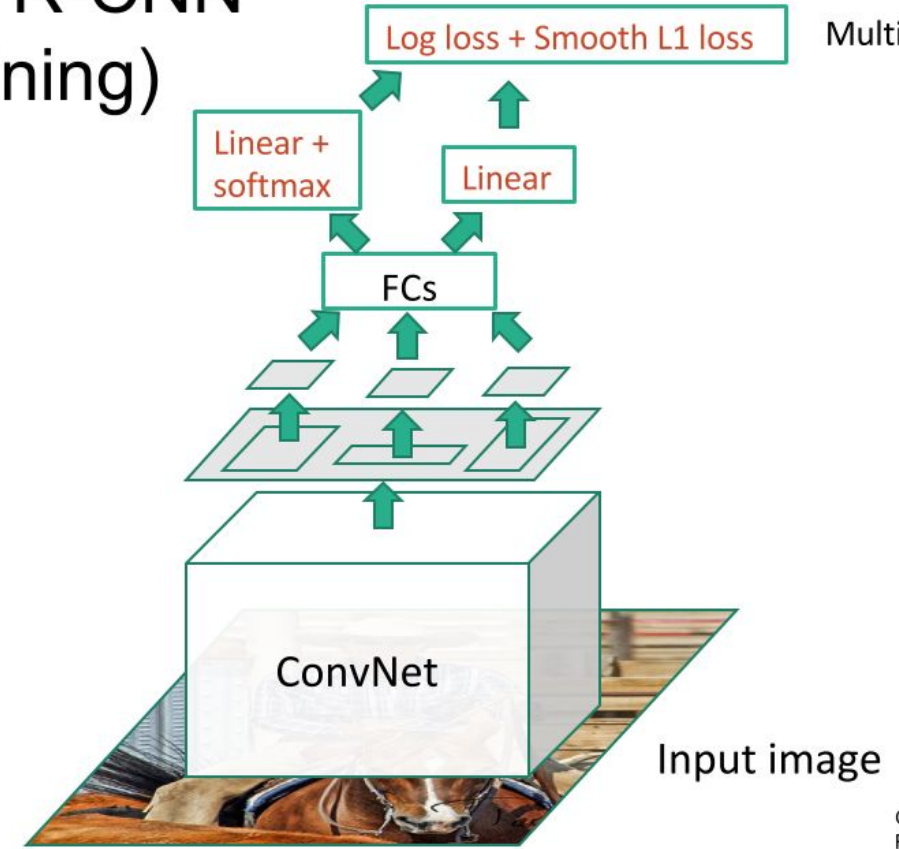
Fast R-CNN



Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Fast R-CNN (Training)

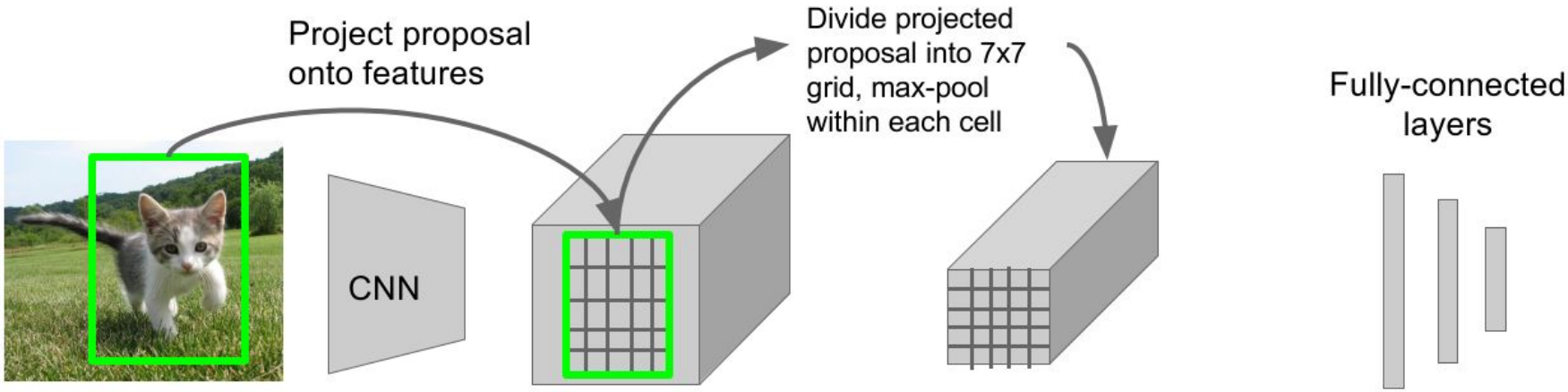


Multi-task loss

Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Fast R-CNN: RoI Pooling



Hi-res input image: 3 x 640 x 480 with region proposal

Hi-res conv features: 512 x 20 x 15;

Projected region proposal is e.g. 512 x 18 x 8 (varies per proposal)

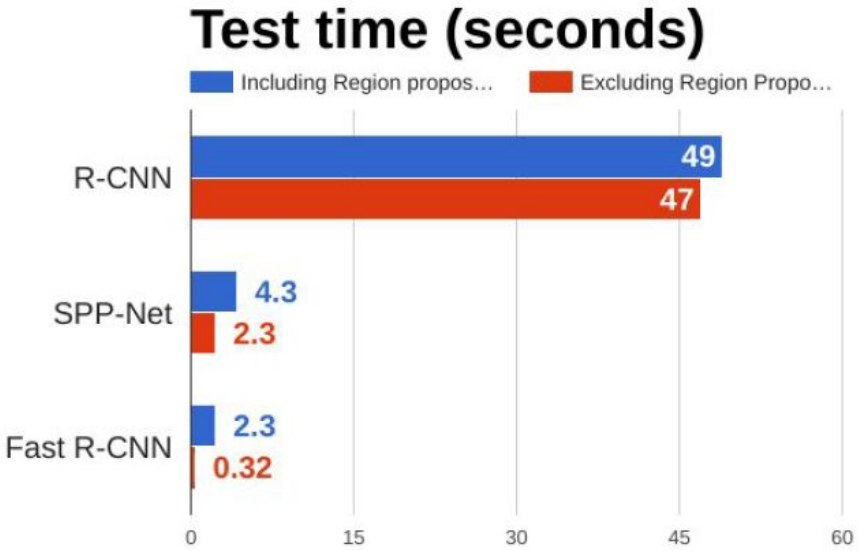
RoI conv features: 512 x 7 x 7 for region proposal

Fully-connected layers expect low-res conv features: 512 x 7 x 7

Girshick, "Fast R-CNN", ICCV 2015.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

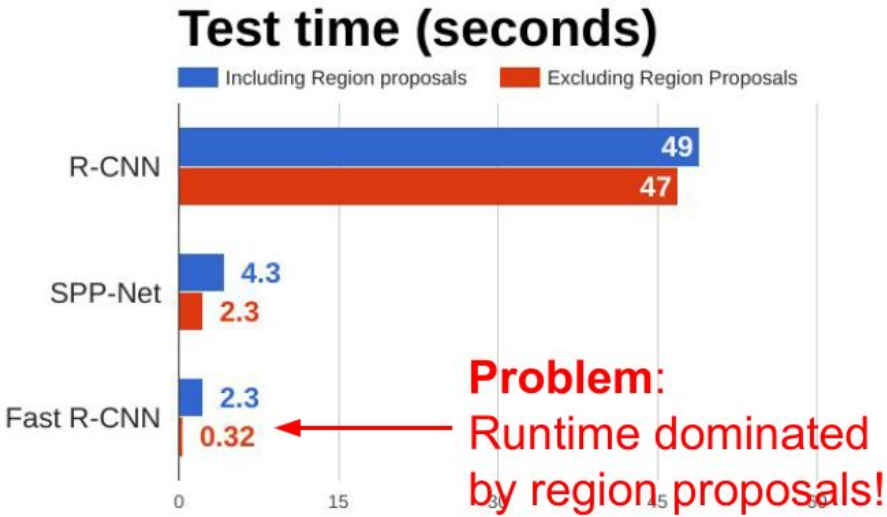
R-CNN vs SPP vs Fast R-CNN



Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
He et al, “Spatial pyramid pooling in deep convolutional networks for visual recognition”, ECCV 2014
Girshick, “Fast R-CNN”, ICCV 2015

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

R-CNN vs SPP vs Fast R-CNN



Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
He et al, “Spatial pyramid pooling in deep convolutional networks for visual recognition”, ECCV 2014
Girshick, “Fast R-CNN”, ICCV 2015

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

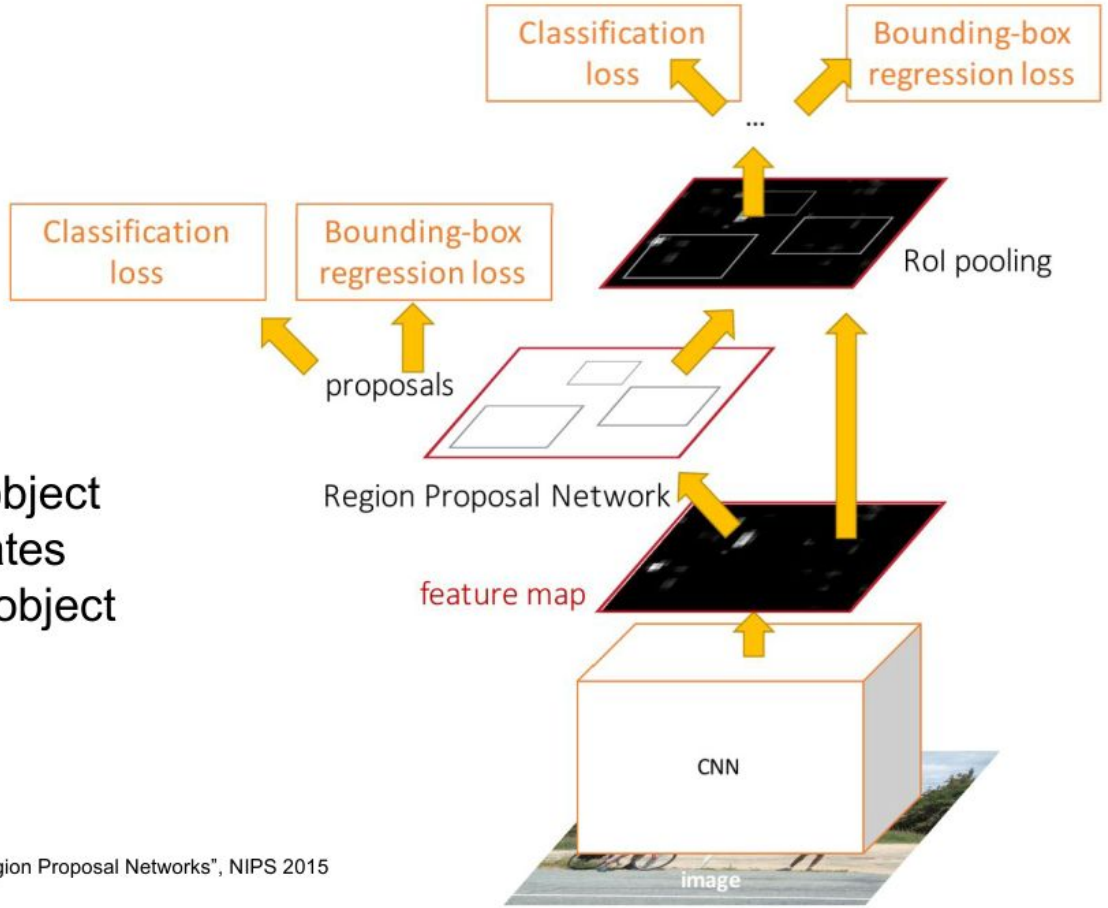
Faster R-CNN:

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

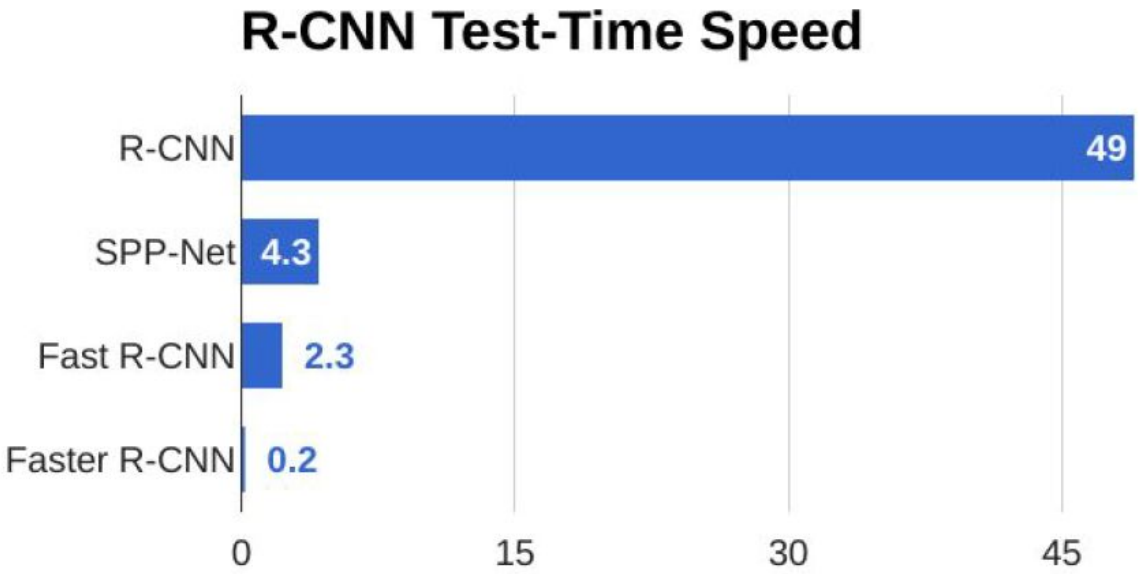


Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

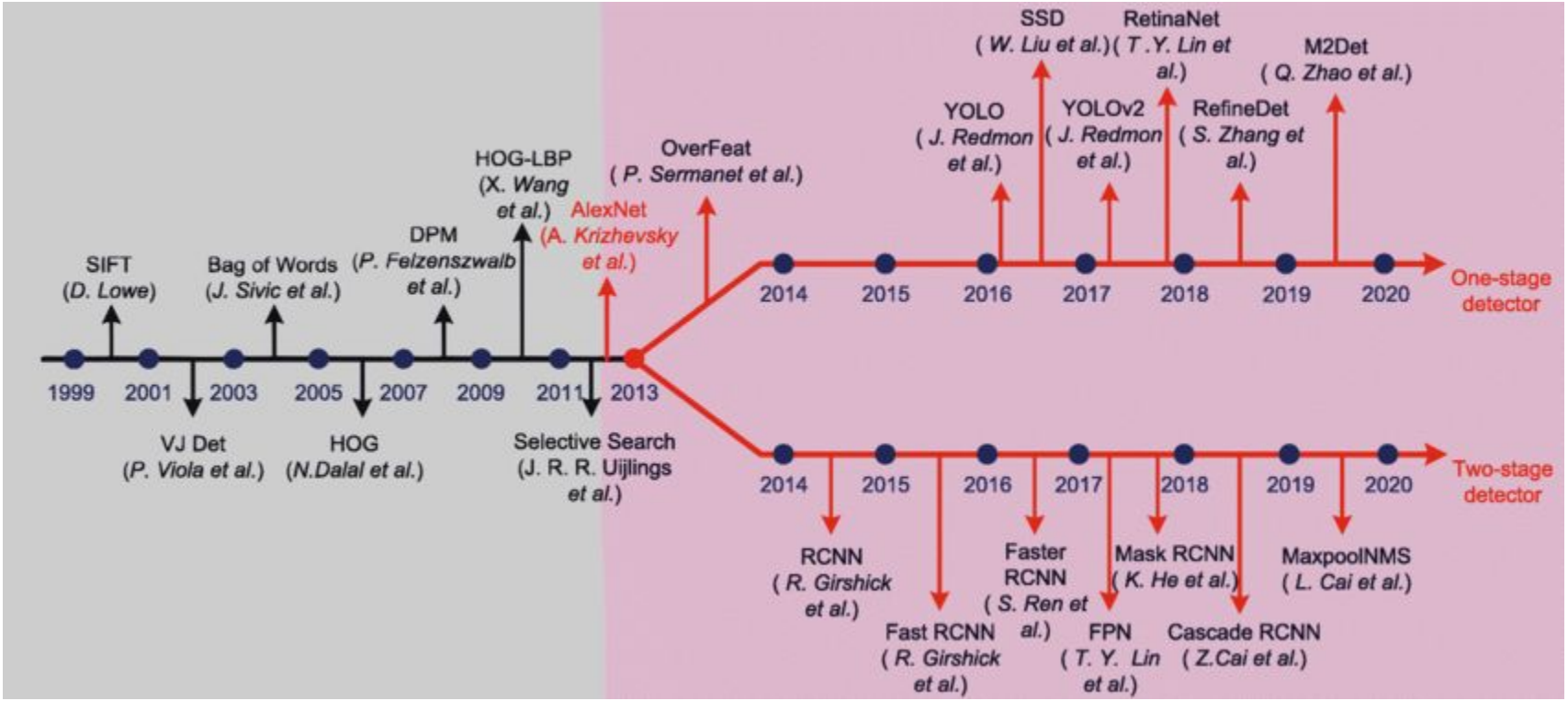
Faster R-CNN:

Make CNN do proposals!

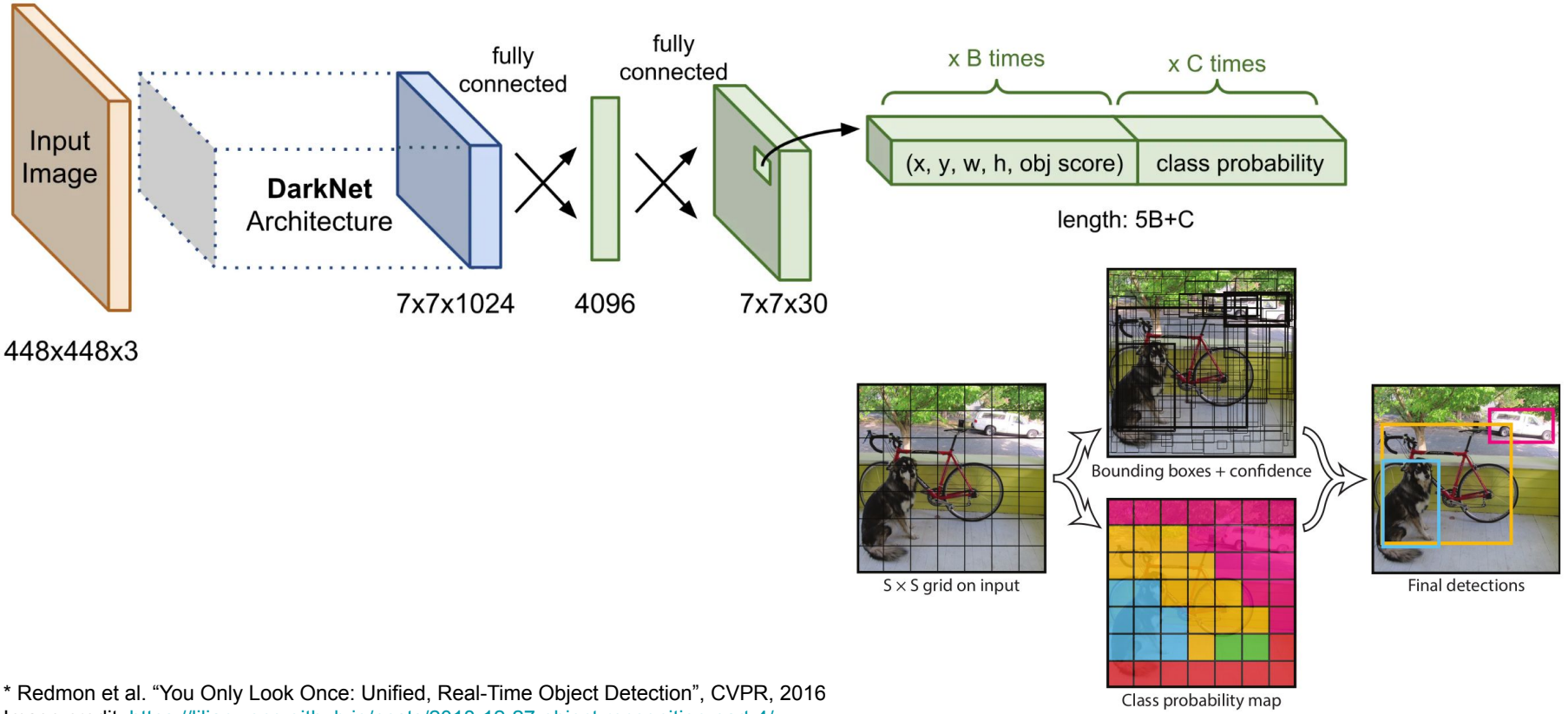


Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Evolution of object detection

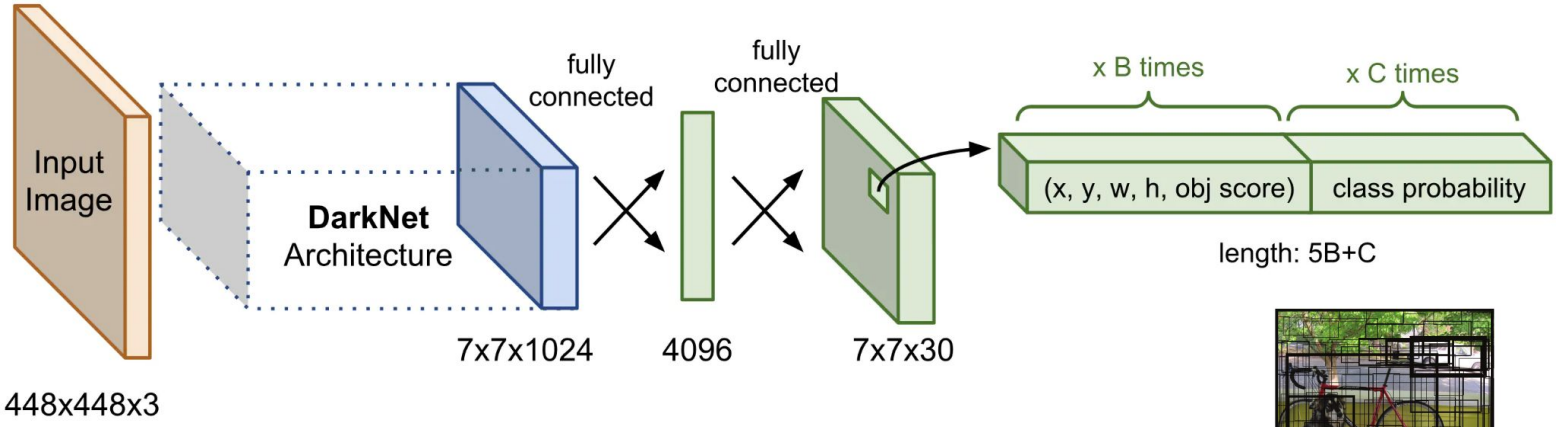


Single stage object detection - YOLO v1*

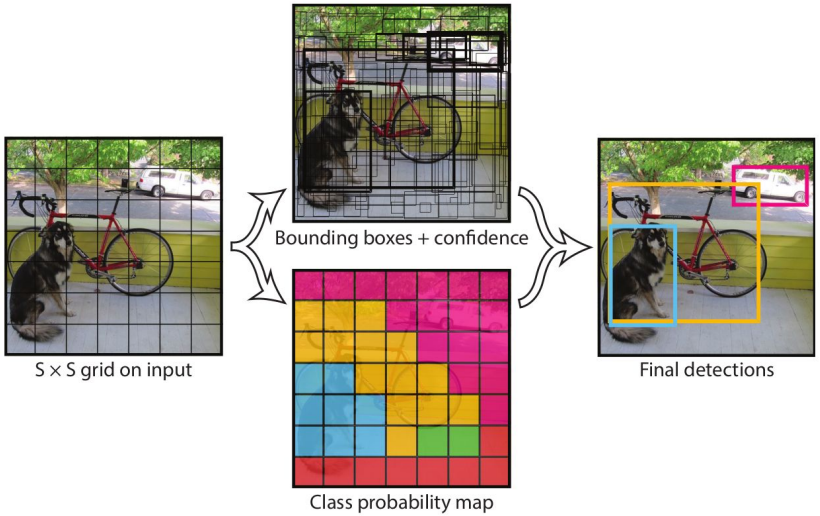


* Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”, CVPR, 2016
Image credit: https://lilianweng.github.io/posts/2018-12-27-object-recognition-part-4/

Single stage object detection - YOLO v1



- **Darknet (26 layers)**: inspired from Googlenet. Inception module replaced by 1x1 and 3x3 conv layers.
- **x, y**: relative to the cell coordinate (top-left).
- **w, h**: normalized by image width and height.
- During the training, **obj score** is computed by **IOU** between predicted box and groundtruth.
- **Class probabilities** are multiplied by obj score.



Single stage object detection - YOLO v1

- **Pros:**
  - It is fast,
  - Generalizes well to unseen domains.
- **Cons:**
  - Its accuracy is not great,
  - Does not work well on different image resolutions,
  - Only predicts one class per cell.

Single stage object detection - Non Maximum Suppression

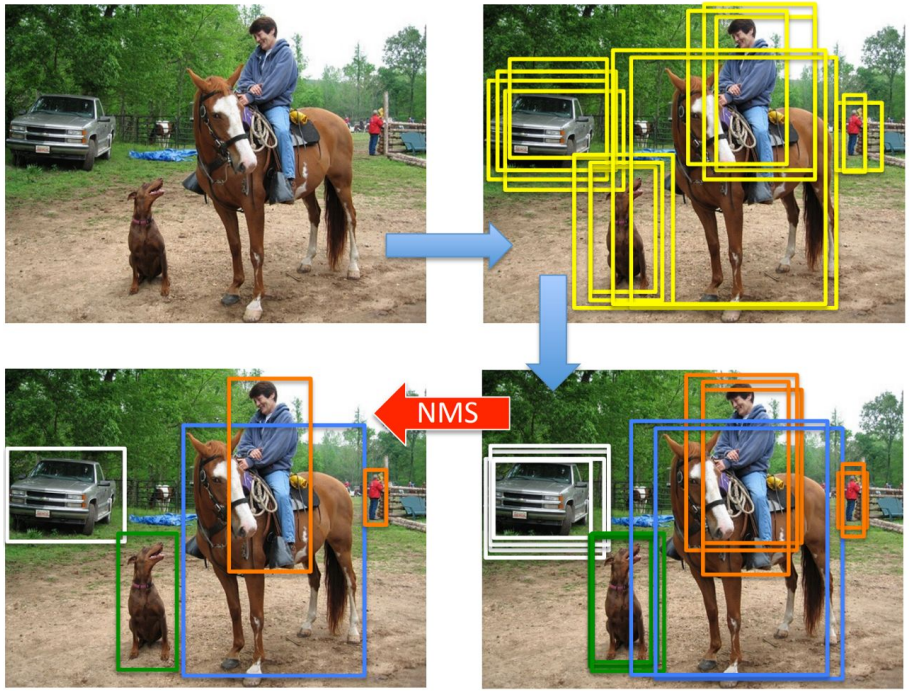


Image credit: https://arxiv.org/pdf/1704.04503.pdf

Input : $\mathcal{B} = \{b_1, .., b_N\}, \mathcal{S} = \{s_1, .., s_N\}, N_t$
$\mathcal{B}$ is the list of initial detection boxes
$\mathcal{S}$ contains corresponding detection scores
N_t is the NMS threshold

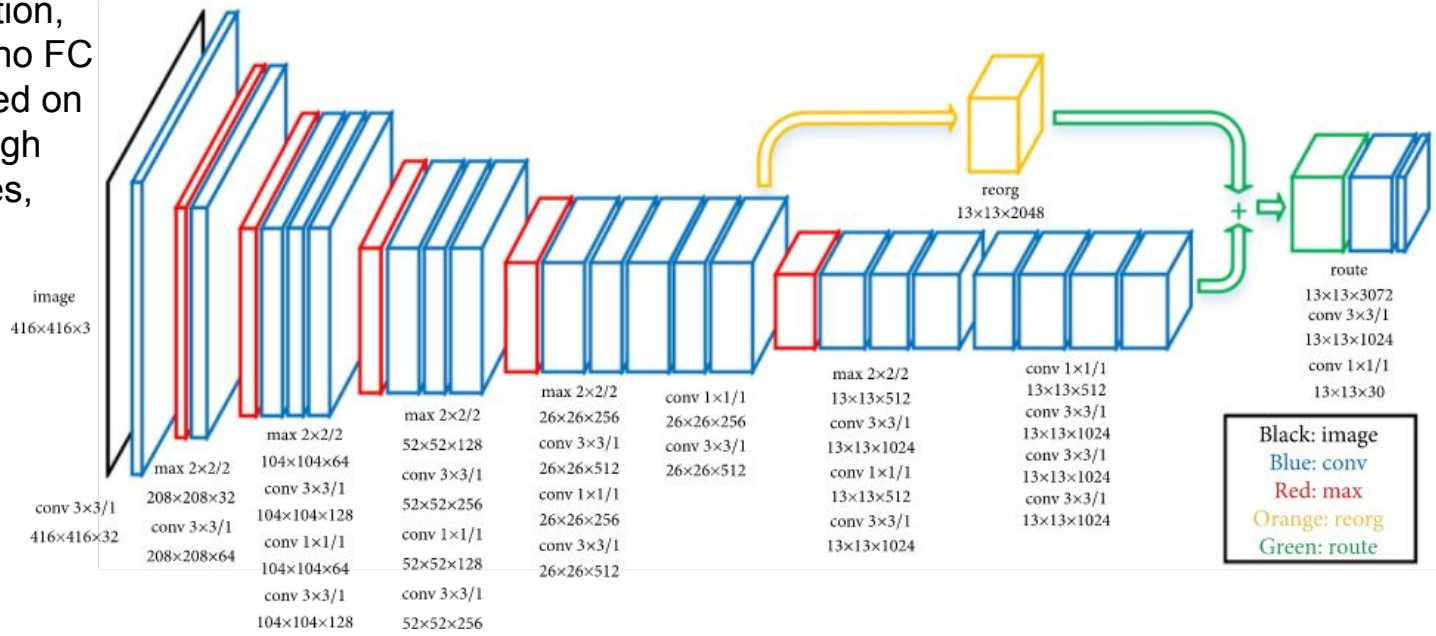
begin
  $\mathcal{D} \leftarrow \{\}$
  while $\mathcal{B} \neq$ *empty* **do**
    $m \leftarrow$ argmax $\mathcal{S}$
    $\mathcal{M} \leftarrow b_m$
    $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{M}; \mathcal{B} \leftarrow \mathcal{B} - \mathcal{M}$
    for b_i *in* $\mathcal{B}$ **do**
      if $iou(\mathcal{M}, b_i) \geq N_t$ **then**
        $\mathcal{B} \leftarrow \mathcal{B} - b_i; \mathcal{S} \leftarrow \mathcal{S} - s_i$
      end NMS
      $s_i \leftarrow s_i f(iou(\mathcal{M}, b_i))$ Soft-NMS
    end
  end
  return $\mathcal{D}, \mathcal{S}$

$$s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ s_i(1 - \text{iou}(\mathcal{M}, b_i)), & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases},$$

Single stage object detection - YOLO v2 *

Improvements over YOLOv1

- Darknet with 19 layers, batch normalization, reorg layer and no FC layers, pre-trained on imagenet with high resolution images,

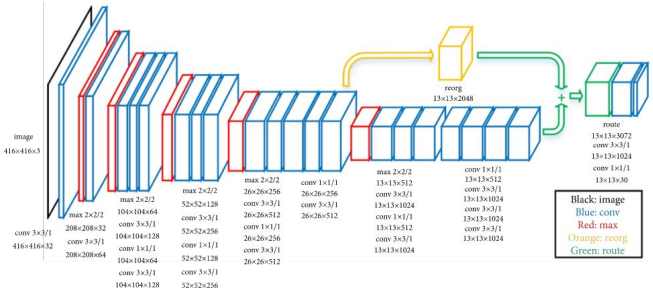


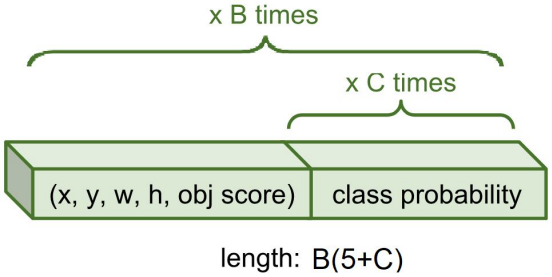
* https://arxiv.org/pdf/1612.08242.pdf
Image credit: https://www.hindawi.com/journals/mpe/2018/3518959/

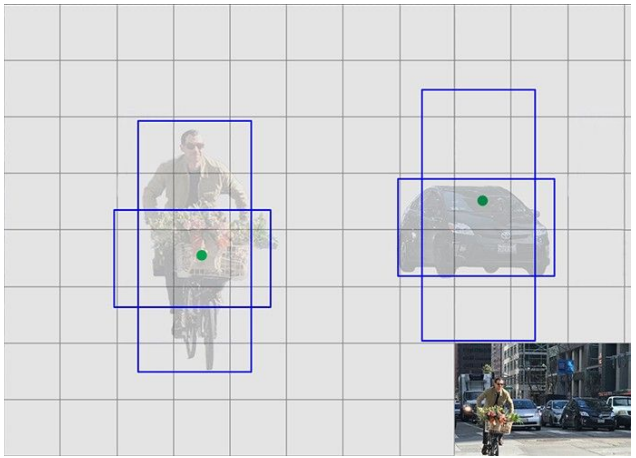
Single stage object detection - YOLO v2

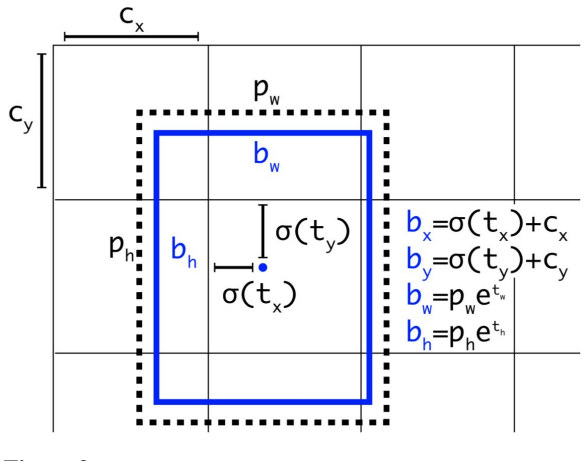
Improvements over YOLOv1

- Darknet with 19 layers, batch normalization, reorg layer and no FC layers, pre-trained on imagenet with high resolution images,
- Introduction of anchor boxes optimized by kmeans rather than predefined boxes,
- Object classes defined for each anchor in each cell,
- Multi-scale training every 10 epochs.





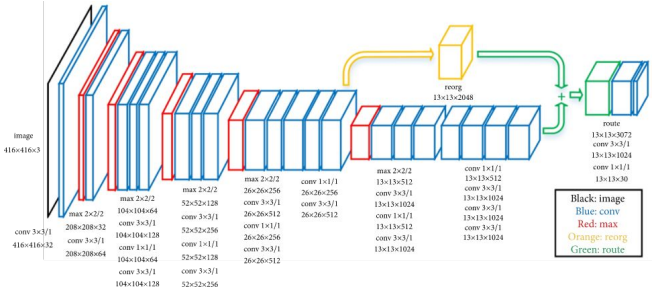


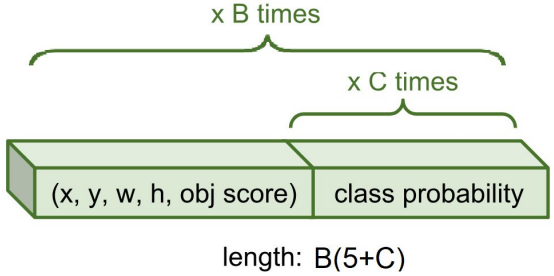


Single stage object detection - YOLO v2

Improvements over YOLOv1

- Darknet with 19 layers, batch normalization, reorg layer and no FC layers, pre-trained on imagenet with high resolution images,
- Introduction of anchor boxes optimized by kmeans rather than predefined boxes,
- Object classes defined for each anchor in each cell,
- Multi-scale training every 10 epochs.



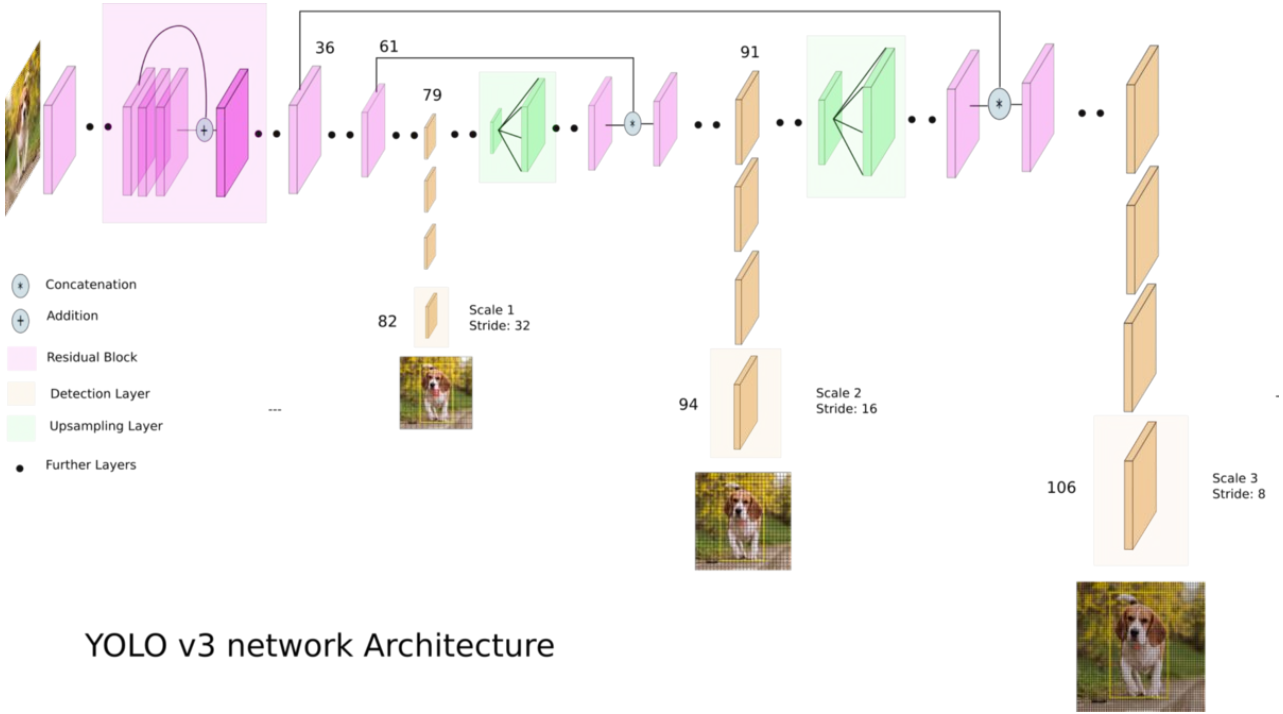


Localization improved but still low recall on small objects

Single stage object detection - YOLO v3

Improvements over YOLOv2

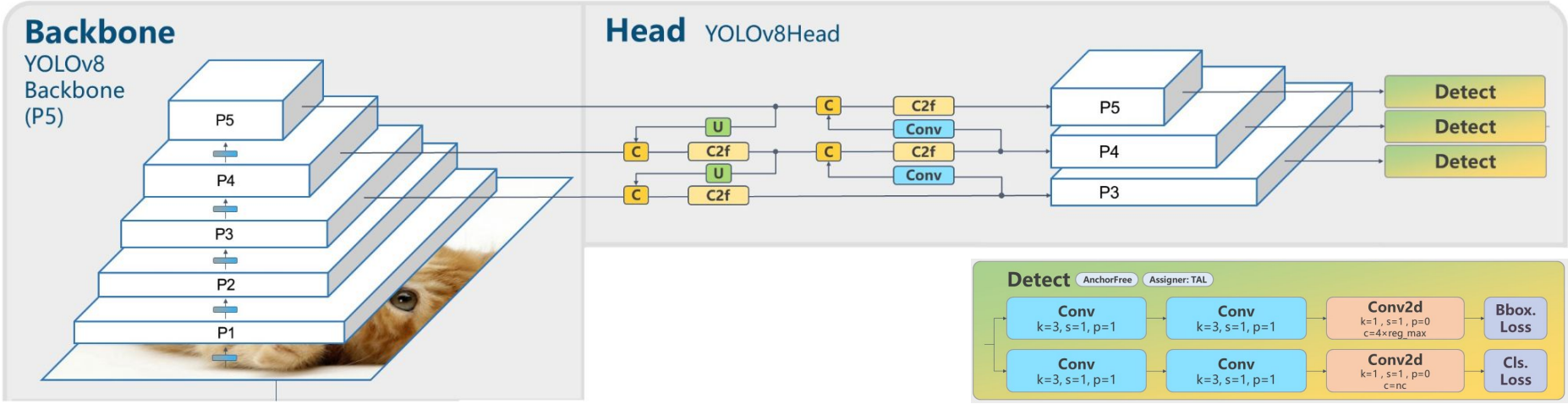
- Darknet with 53 layers, residual connections and feature pyramid layers at the end,
- Logistic regression for object score,
- Sigmoid activation and binary cross-entropy for class prediction.



YOLO v3 network Architecture

Image credit: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

Single stage object detection - YOLO v8



Features:

- New architecture,
- Fast detection with state-of-the-art accuracy,
- Anchor-free detections, which help to speed-up NMS,
- A well-structured python package with developer convenience features like semantic segmentation and object classification,

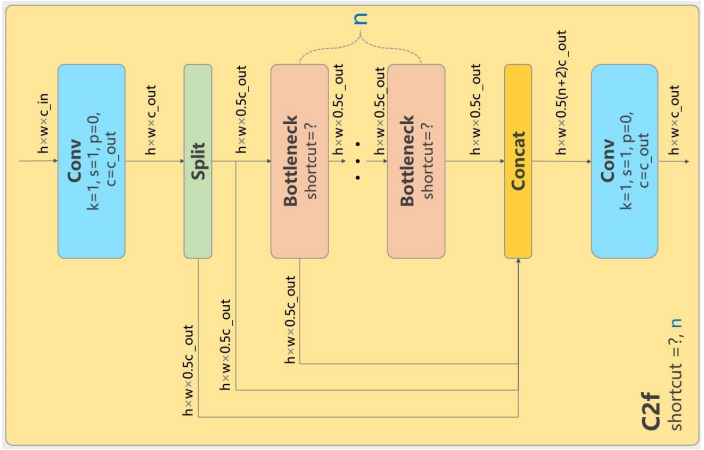
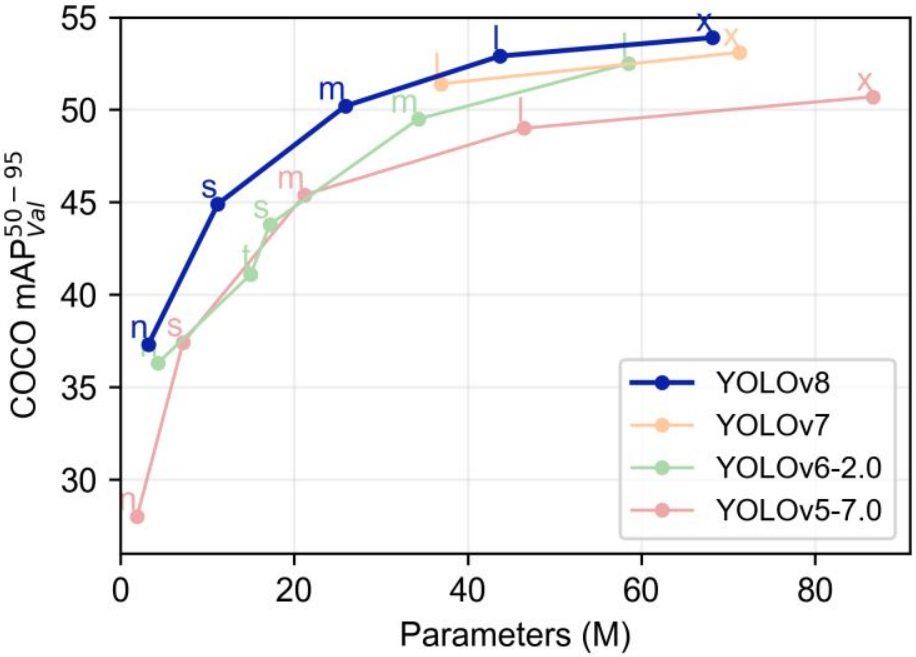
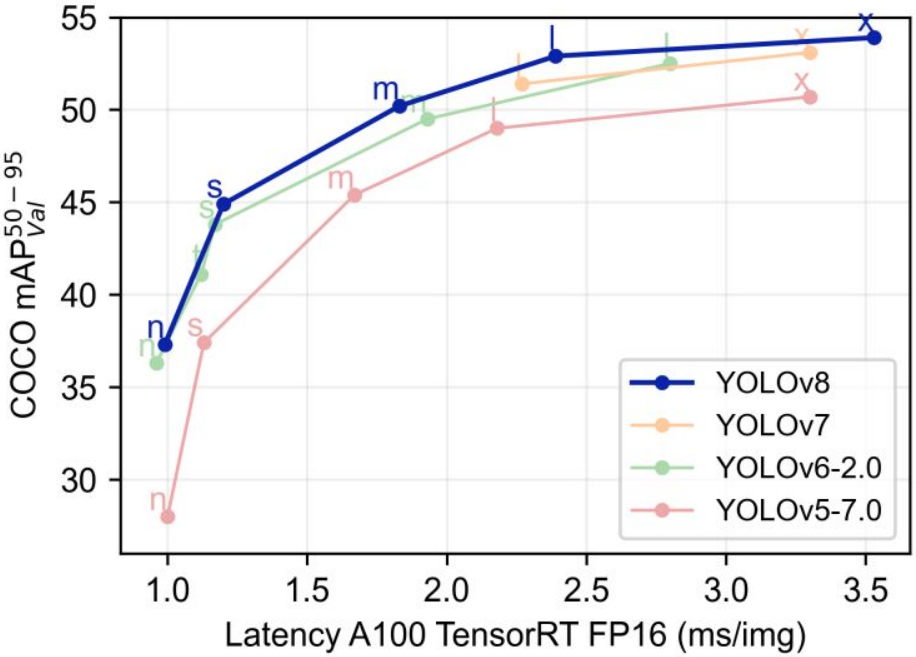
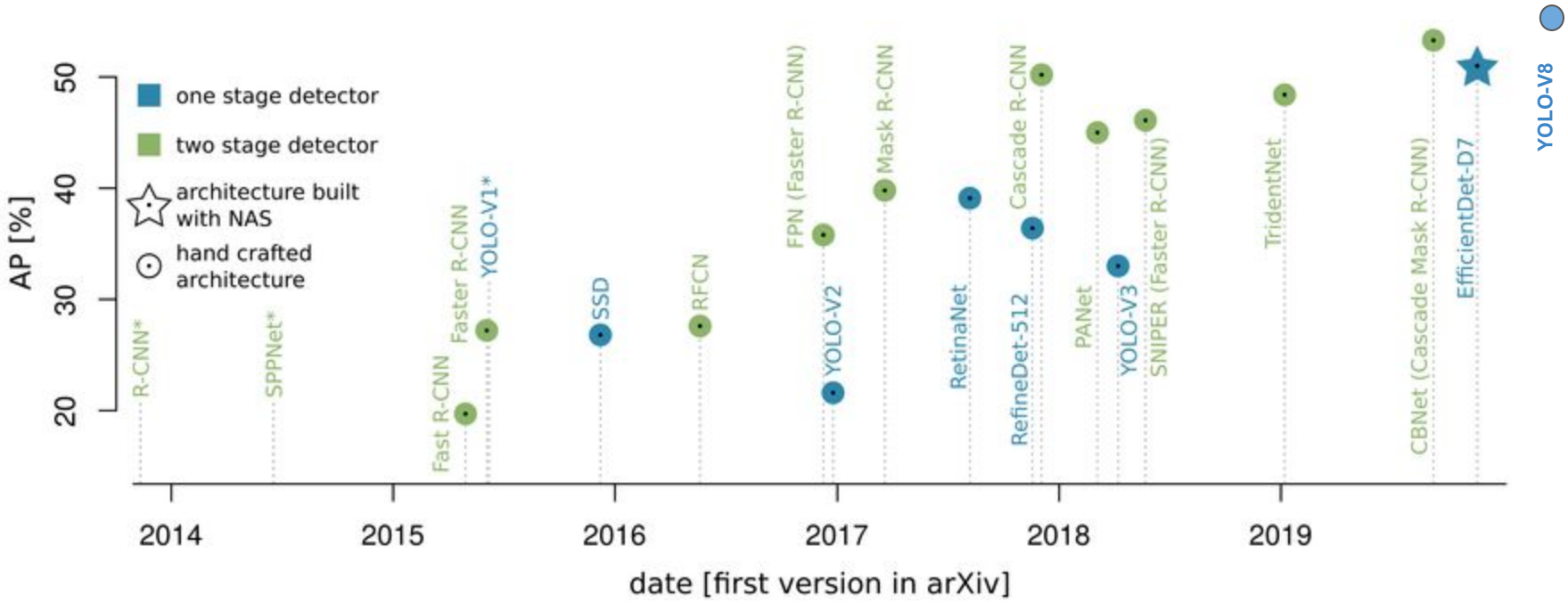


Image credit: RangeKing https://github.com/ultralytics/ultralytics/issues/189

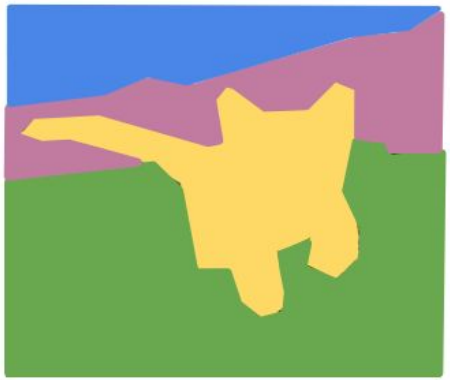
Single stage object detection - YOLO v8





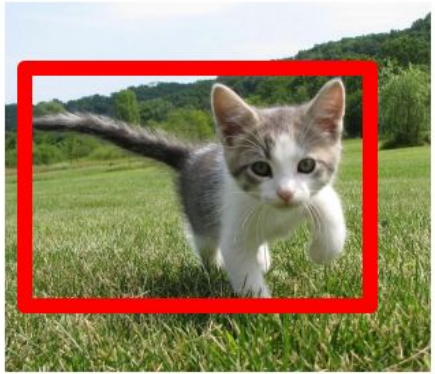

AP [%]
20
30
40
50
one stage detector
two stage detector
architecture built with NAS
hand crafted architecture
R-CNN*
SPPNet*
Fast R-CNN
Faster R-CNN
YOLO-V1*
SSD
RFCN
FPN (Faster R-CNN)
YOLO-V2
Mask R-CNN
RetinaNet
RefineDet-512
Cascade R-CNN
PANet
YOLO-V3
SNIPER (Faster R-CNN)
TridentNet
CBNet (Cascade Mask R-CNN)
EfficientDet-D7
YOLO-V8
2014
2015
2016
2017
2018
2019
date [first version in arXiv]

Instance Segmentation



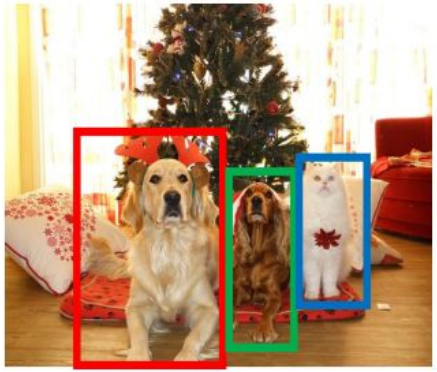
GRASS, CAT, TREE, SKY

No objects, just pixels



CAT

Single Object



DOG, DOG, CAT

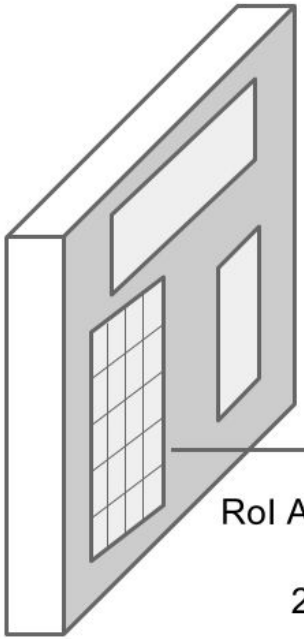
DOG, DOG, CAT

Multiple Object

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

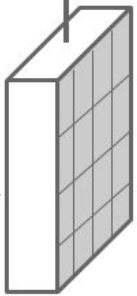
Mask R-CNN

CNN



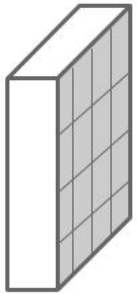
RoI Align

Classification Scores: C
Box coordinates (per class): 4 * C



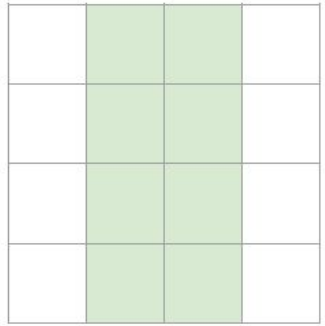
256 x 14 x 14

Conv



256 x 14 x 14

Conv



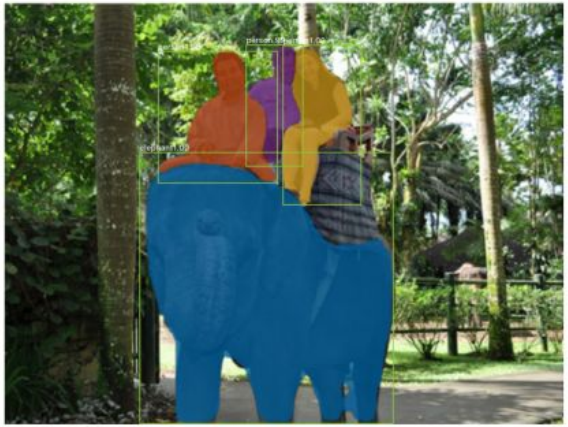
Predict a mask for each of C classes

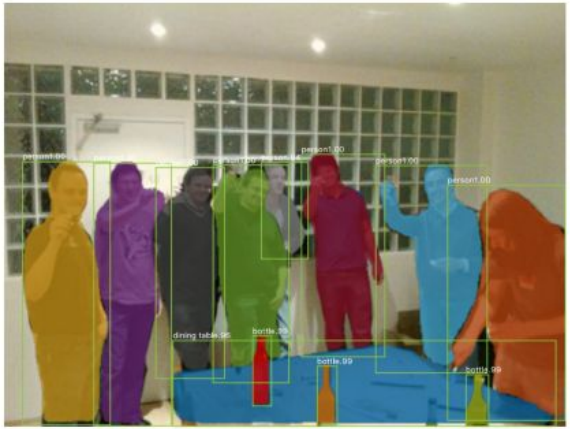
C x 14 x 14

He et al, “Mask R-CNN”, arXiv 2017

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Mask R-CNN: Very Good Results!

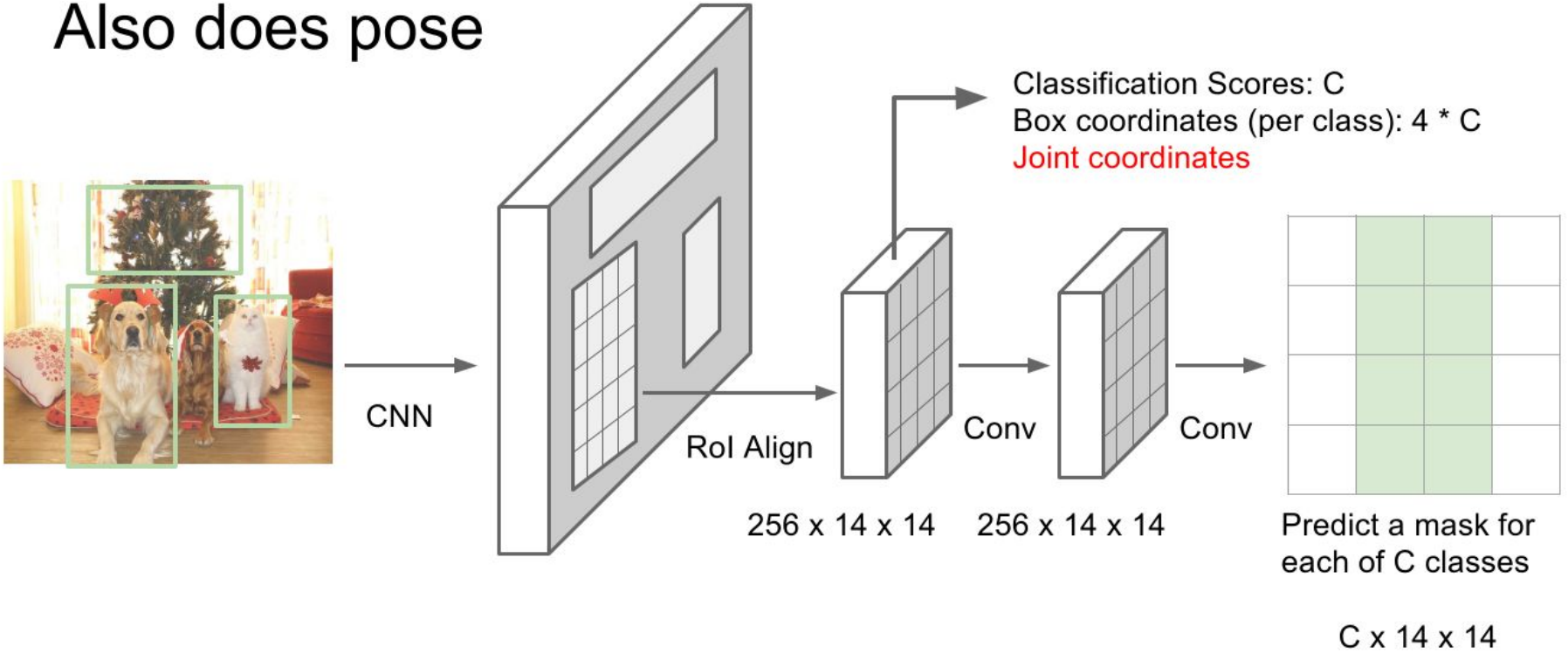




He et al, “Mask R-CNN”, arXiv 2017

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

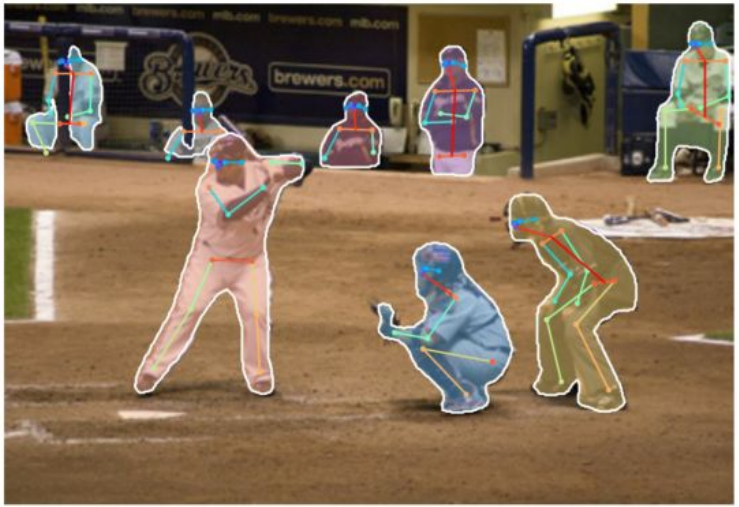
Mask R-CNN
Also does pose

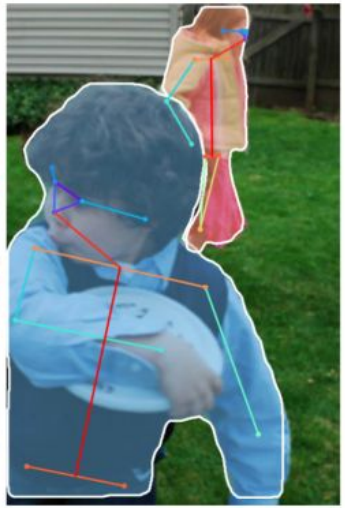


He et al, "Mask R-CNN", arXiv 2017

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung

Mask R-CNN
Also does pose





He et al, “Mask R-CNN”, arXiv 2017
Figures copyright Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, 2017.
Reproduced with permission.

Slide credit: Stanford University, Fei-Fei Li & Justin Johnson & Serena Yeung