CNN: Object Detection

ENEE 4584/5584 Neural Nets

Dr. Alsamman

Slide Credits:

- https://web.stanford.edu/class/biods220/lectures/lecture4.pdf
- http://www.cs.cornell.edu/courses/cs7670/2014sp/slides/VisionSeminar14.pdf
- http://vision.stanford.edu/teaching/cs231b_spring1415/papers/IJCV2004_FelzenszwalbHuttenlocher.pdf
- https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html
- https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/
- http://d2l.ai/chapter_computer-vision/rcnn.html
- https://medium.com/lsc-psd/easiest-rpn-explained-the-core-of-faster-r-cnn-3b0168c3e650
- https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11
- https://jonathan-hui.medium.com/understanding-region-based-fully-convolutional-networks-r-fcn-for-object-detection-828316f07c99

Computer Challenges

- Object detection dual priorities: classification & localization
- Real-time detection
- Multi-scale detection
- Overcome class imbalance

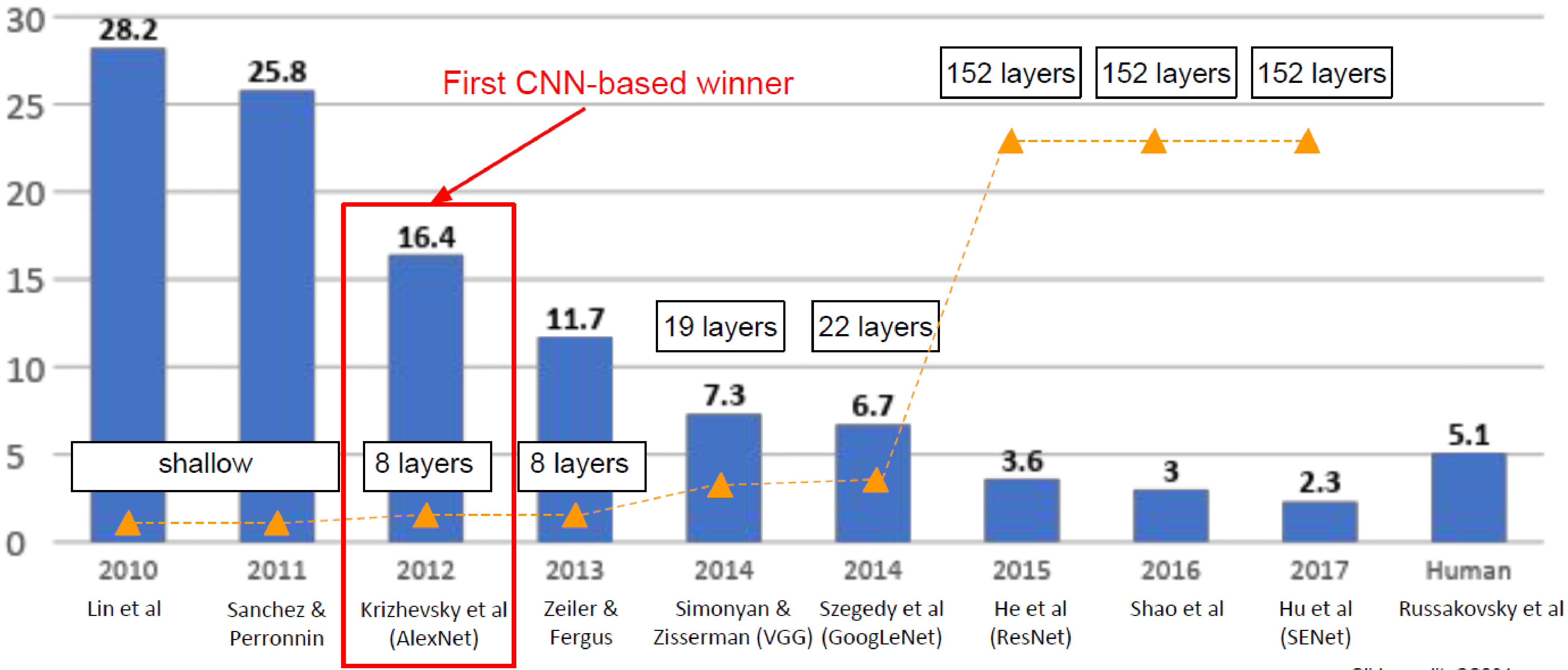
Techniques

- R-CNN (2013): https://arxiv.org/abs/1311.2524
  - **Ross Girshick**, Jeff Donahue, Trevor Darrell, Jitendra Malik
- Fast R-CNN (2015): https://arxiv.org/abs/1504.08083
  - **Ross Girshick**
- Faster R-CNN (2015): https://arxiv.org/abs/1506.01497
  - Shaoqing Ren, **Kaiming He**, **Ross Girshick**, Jian Sun
- Mask R-CNN (2017): https://arxiv.org/abs/1703.06870
  - **Kaiming He**, Georgia Gkioxari, Piotr Dollár, **Ross Girshick**
- YOLO[2016]: https://arxiv.org/abs/1506.02640
  - Joseph Redmon, Santosh Divvala, **Ross Girshick**, Ali Farhadi
- SSD[2016]: https://arxiv.org/abs/1512.02325
  - W Liu, D Anguelov, D Erhan, C Szegedy, S Reed, C-Y Fu, A C. Berg
- R-FCN [2016]: https://arxiv.org/pdf/1605.06409.pdf
  - J Dai, Y L Tsinghua, **Kaiming He**, J Sun

Image Detection

- UT's AlexNet was a game changer
  - competed in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2012)
  - achieved 15.3% error on top 5
  - -10.8% than runner up
- OX's VGGNet (2014) reduced the error to under 7%

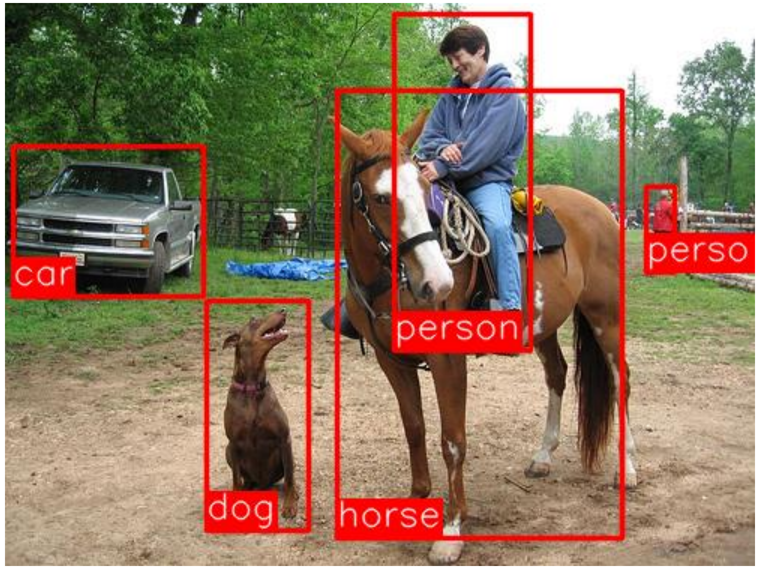
ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Slide credit: CS231n

Object Detection

- UCB team: Can AlexNet generalize to object detection?
  - Driven by a different challenge: PASCAL VOC
- Goal: Detect & classify objects **and** their boundary box
- Challenge: localization of objects
  - Object candidates ~ number of pixels
  - Varying in scale
  - Varying in color/texture
- Exhaustive search:
  - Pros: Captures all possible locations
  - Cons: Super slow, class dependent



Solution: Selective Search

- Data-driven (no class driven) segmentation
- Exploits image structures for proposals
  - Input: (color)image
  - Output: Set of object location hypotheses L
- Algorithm has 3 parts:
  1. Obtain initial regions
  2. Calculate similarity between regions & Merge
  3. Create a region hierarchy

Segmentation Algorithm

- Part 1: Derive regions R by graph segmentation:
  - Obtain Edges and their Vertices (E,V)
  - Sort E
  - Vs connected by E form a component C
  - Merge components based on a calculated weight measure
- Part 2: Calculate similarity between regions
  - Use 4 similarity measures: color, texture, size, fill
  - Find regions with highest similarity & merge them
- Part 3: region hierarchy
  - Repeat step 2 for newly merged regions

Figure 2: Two examples of our selective search showing the necessity of different scales. On the left we find many objects at different scales. On the right we necessarily find the objects at different scales as the girl is contained by the tv.

R-CNN Algorithm

- Regions with CNN features
- Based on a pre-trained CNN (AlexNet, VGGNet, GoogleNet, etc.)

1. Apply Selective Search
   - Produce 2K region proposals (aka RoI)
2. Use a pre-trained CNN (AlexNet) to classify regions
   - Resize (aka warp) each region to a standard CNN input (227x277x3)
   - CNN will produce a classification vector
3. SVM to detect each region
   - Binary SVM trained for each class independently
4. Tighten the region box
   - Apply a linear regression model on the box coordinates

Bounding Box Regression

- **Given**
  - a predicted bounding box coordinate $\boldsymbol{p} = (p_x, p_y, p_w, p_h)$
    - center coordinate, width, height
  - ground truth box coordinates $\boldsymbol{g} = (g_x, g_y, g_w, g_h)$
- **Learn**
  - scale-invariant transformation between two centers

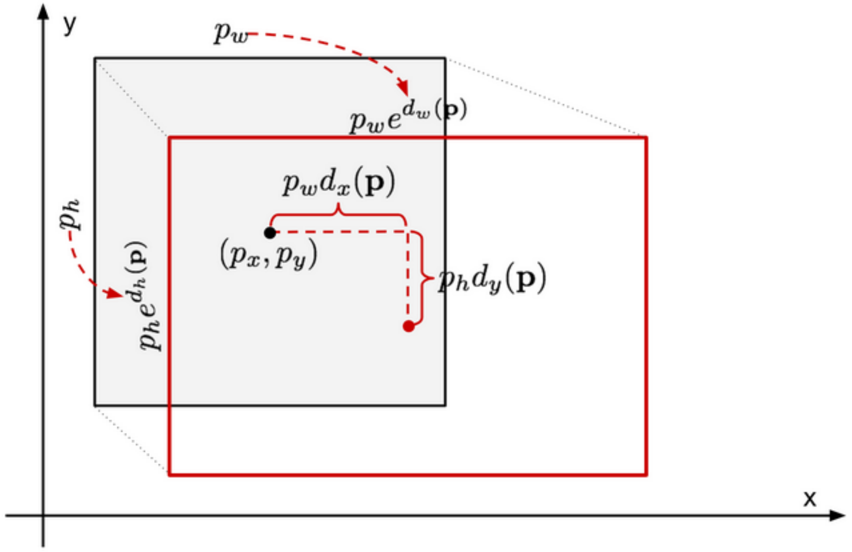
$$t_x = \frac{g_x - p_x}{p_w}, \qquad t_y = \frac{g_y - p_y}{p_h}$$

  - log-scale transformation between widths and heights.

$$t_w = \log\left(\frac{g_w}{p_w}\right), \qquad t_h = \log\left(\frac{g_h}{p_h}\right)$$

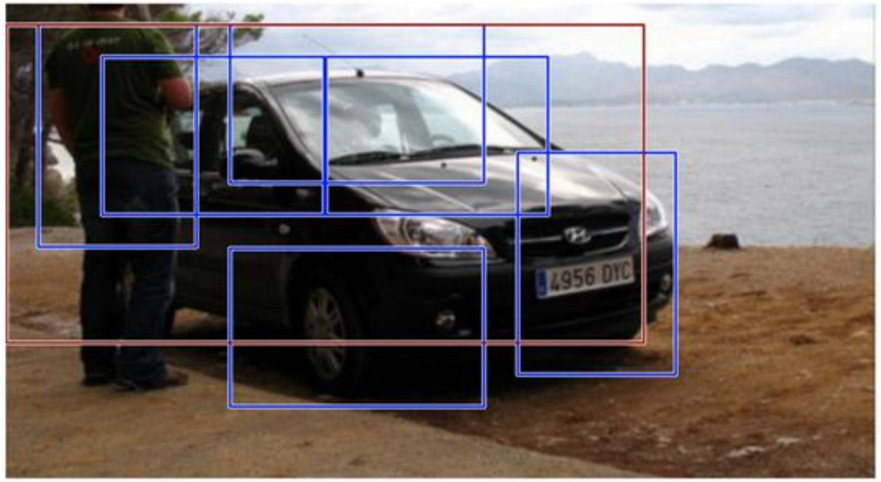
- **Regression Loss function:**

$$\mathcal{L}_{reg} = \sum_{i \in \{x,y,w,h\}} \left(t_i - d_i(P)\right)^2 + \lambda \|\boldsymbol{w}\|^2$$

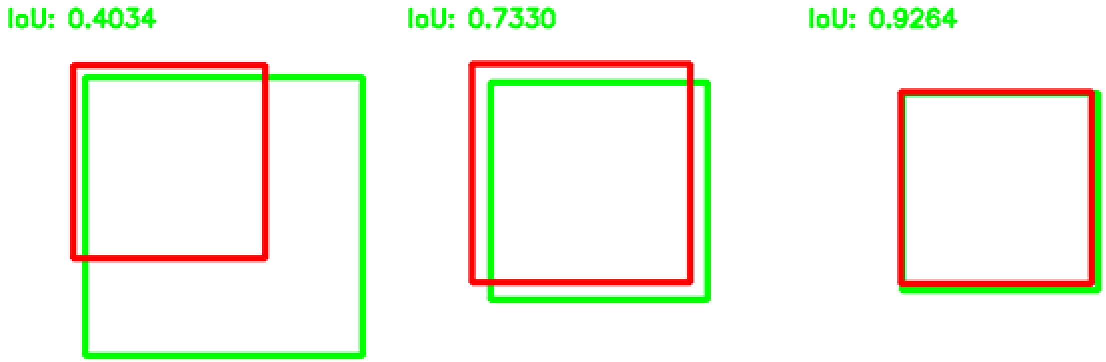


Non-maximal Suppression (NMS)

- multiple bounding boxes for the same object.
- Sort all the bounding boxes by confidence score.
- Discard boxes with low confidence scores.
- *While* there is any remaining bounding box,
  - Greedily select the one with the highest score.
  - Skip the remaining boxes with high IoU (i.e. > 0.5) with previously selected one.



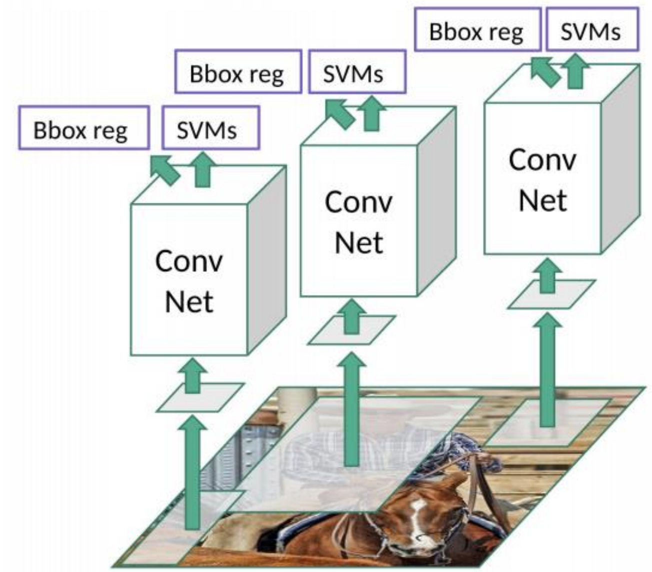
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Hard Negative Mining

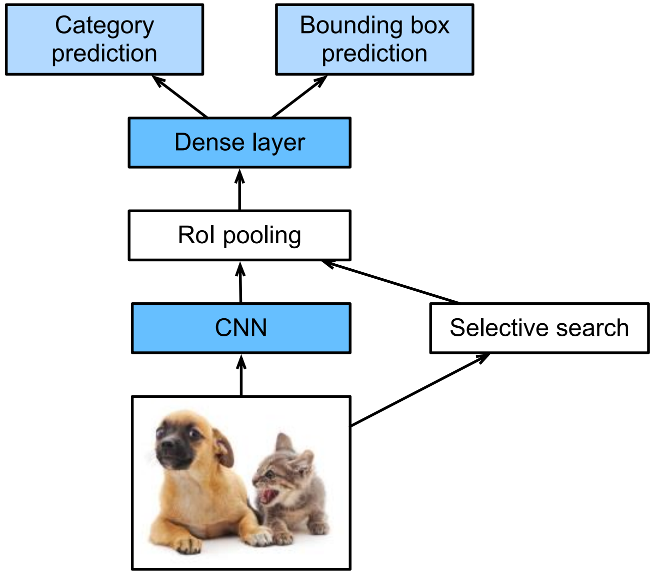
- bounding boxes without objects are considered negative examples
- Hard negative: noisy texture or partial object
- Hard negatives are typically misclassified
- Idea: find hard negatives and use them to augment training data

Fast R-CNN



- R-CNN: 3 disjoint models
  - 2000 Region proposals
  - CNN & SVM
  - Box regression

- Fast R-CNN:
  - Apply CNN first
  - Pool features that belong to the 2000 regions: RoI pooling



R-CNN Workflow

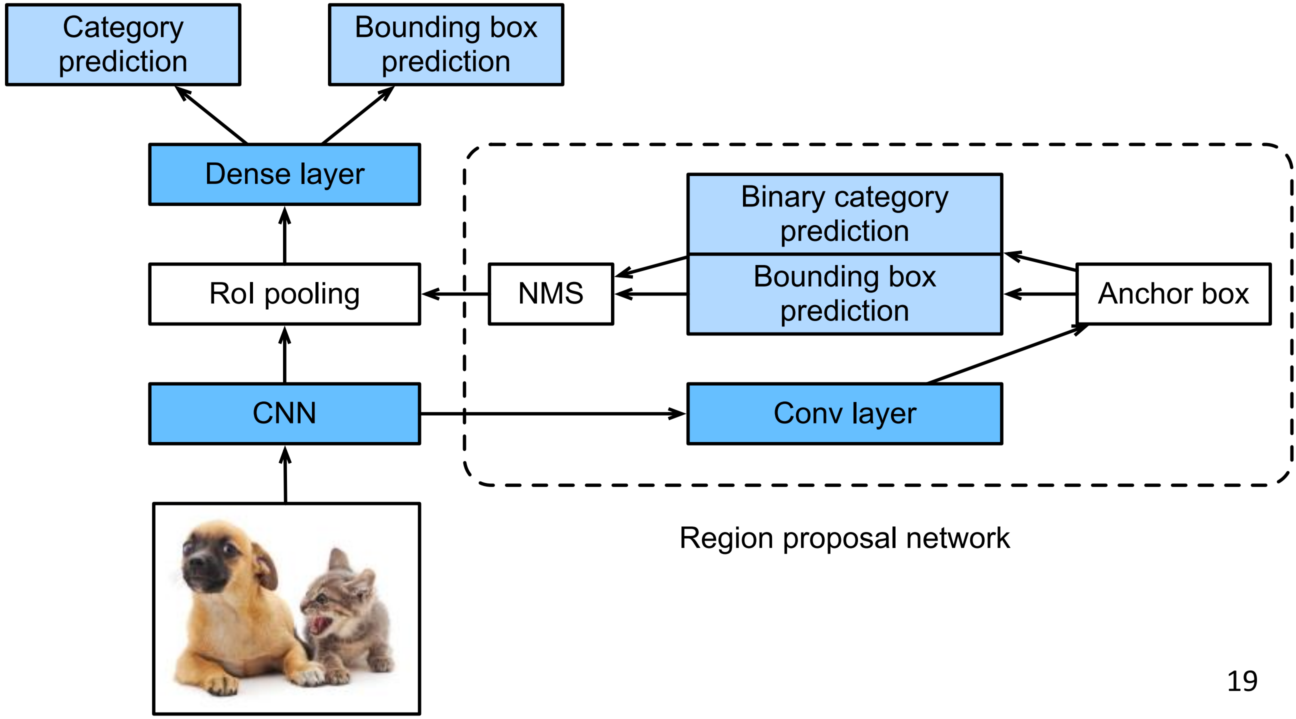
- Propose regions by selective search (~2k candidates per image).
- Alter the pre-trained CNN:
  - Replace the last max pooling layer of the pre-trained CNN with a RoI pooling layer.
  - The RoI pooling layer outputs fixed-length feature vectors of region proposals.
  - Replace the last fully connected layer and the last softmax layer (K classes) with a fully connected layer and softmax over K + 1 classes.
- Finally the model branches into two output layers:
  - A softmax estimator of K + 1 classes
    - same as in R-CNN,
    - +1 is the “background” class
    - outputs a discrete probability distribution per RoI.
  - A bounding-box regression model
    - predicts offsets relative to the original RoI for each of K classes.

Fast R-CNN

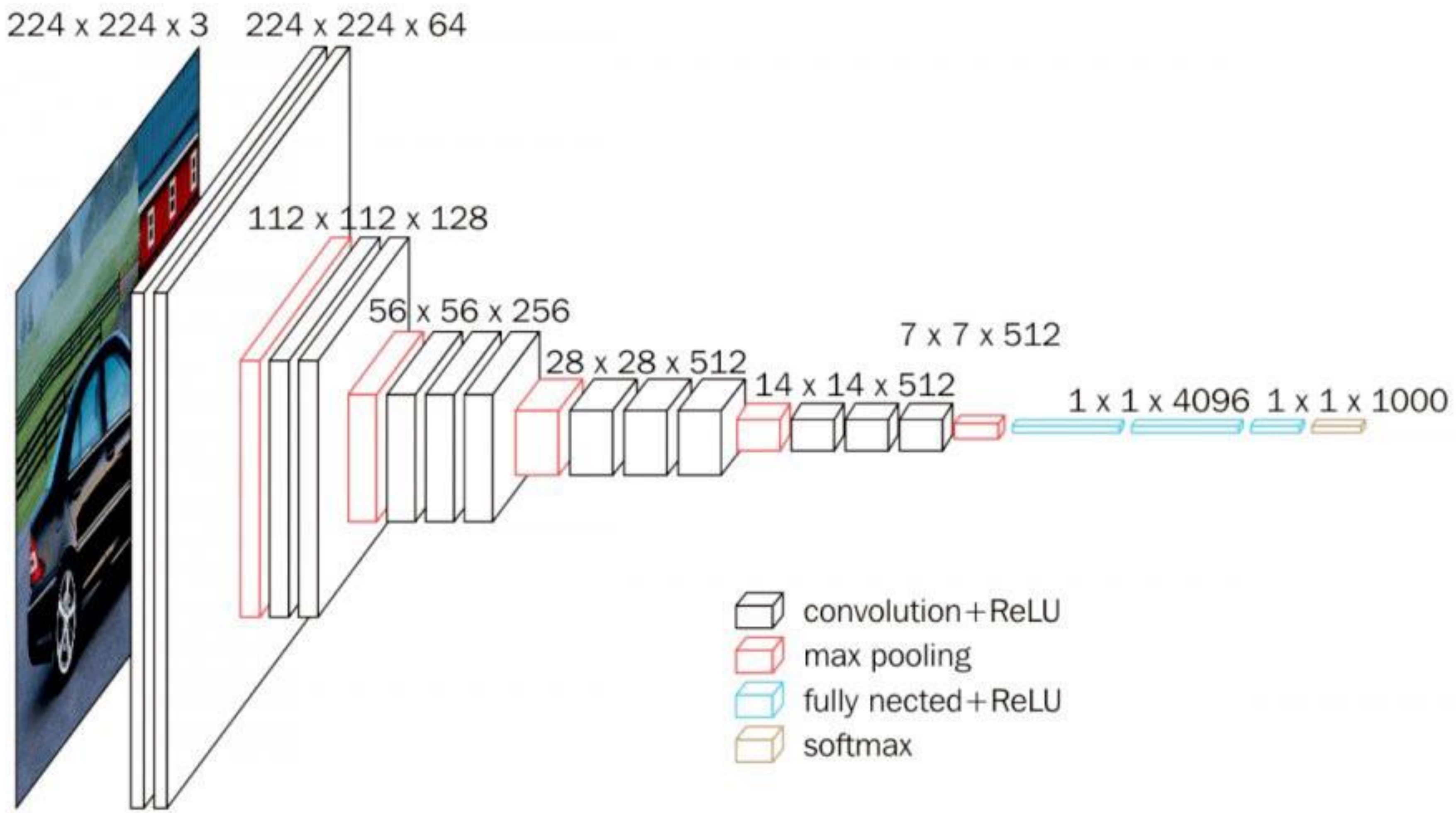
- Optimized for a loss combining two tasks: Classification + Localization
  - Classification: $\mathcal{L}_{cls}$
  - Localization: $\mathcal{L}_{box}$
- Improvements in speed
  - 20+x faster than R-CNN
- Bottleneck: 2000 region proposals
  - More than 2s for complete output

Faster R-CNN

- Integrates the region proposal algorithm into the CNN
- Starts with CNN
- Use CNN features to create a Region proposal Network (RPN)
- The remaining architecture is a Fast RCNN

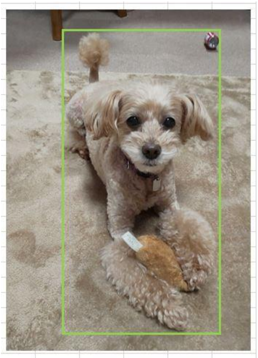


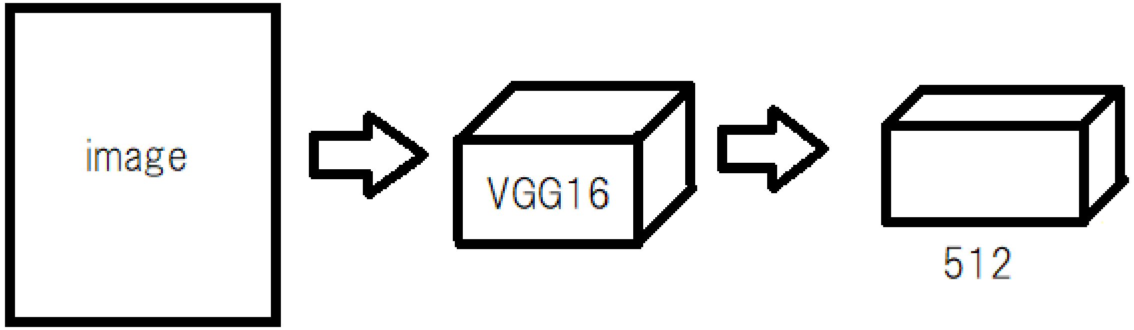
CNN: VGG16


224 x 224 x 3
224 x 224 x 64
112 x 112 x 128
56 x 56 x 256
28 x 28 x 512
14 x 14 x 512
7 x 7 x 512
1 x 1 x 4096
1 x 1 x 1000
convolution+ReLU
max pooling
fully nected+ReLU
softmax

Faster R-CNN Workflow

- Use a pre-trained CNN to generate features



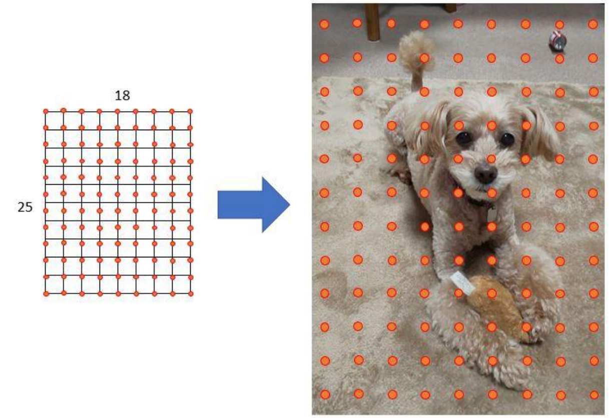


CNN Feature map output

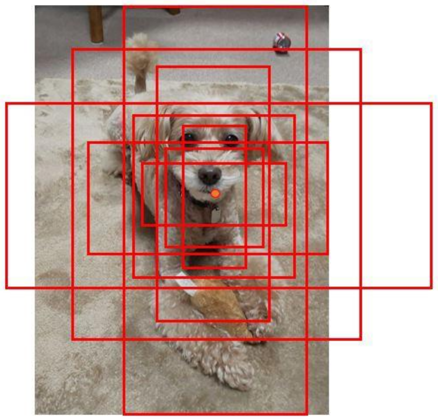
Input training image

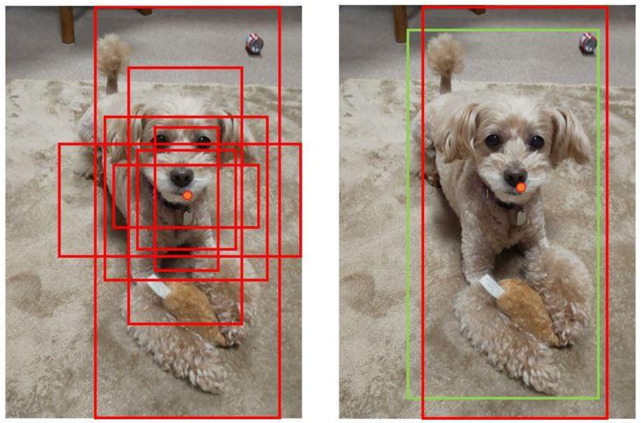
Ground truth bounding box

RPN



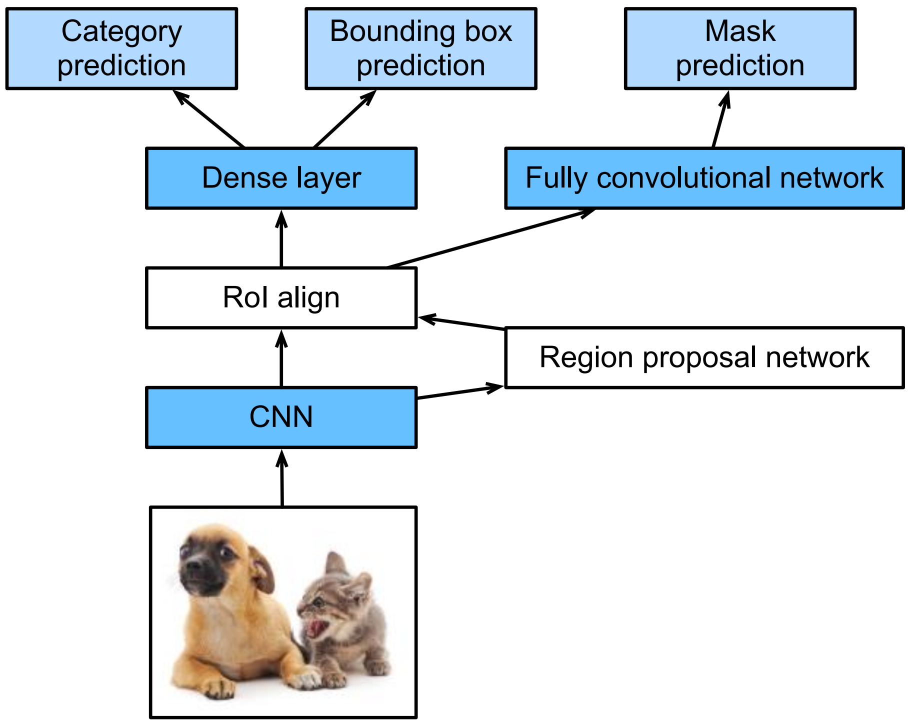
- Proposal regions: anchor boxes
- Each pixel in the feature map is an anchor center
- From each center create multiple (default 9) anchors based on :
  - 3 channels (RGB)
  - 3 scales: e.g. 64, 128, 256
  - 3 height-width ration: e.g. 1:1, 1:2, 2:1
- Notes: drop boxes that exceed the dimensions of the image
- Calculate IoU of anchor box & ground truth
  - IoU < 0.3, box labeled as background
  - IoU > 0.7, box labeled object
  - Ignore boxes 0.3<IoU<0.7





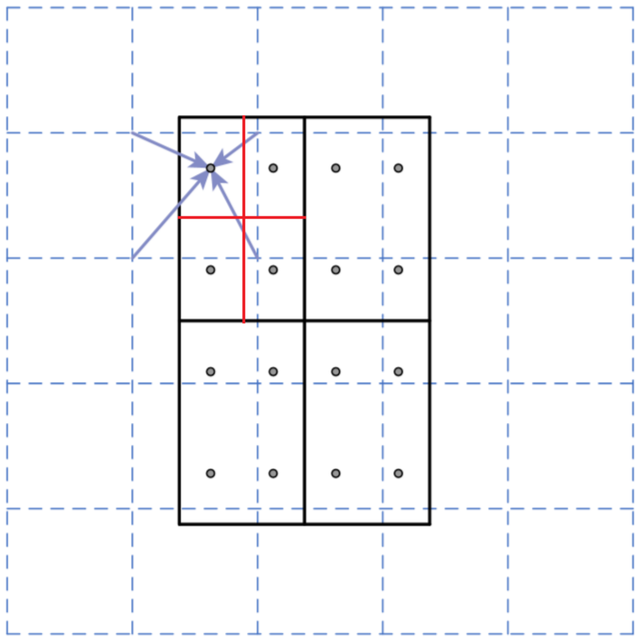
Mask R-CNN

- Faster R-CNN with extras:
  - Region of Interest aligning
    - Use bilinear interpolation
    - Replaces RoI pooling
    - More accurate
  - Mask prediction
    - Pixel level segmentation of best RoI



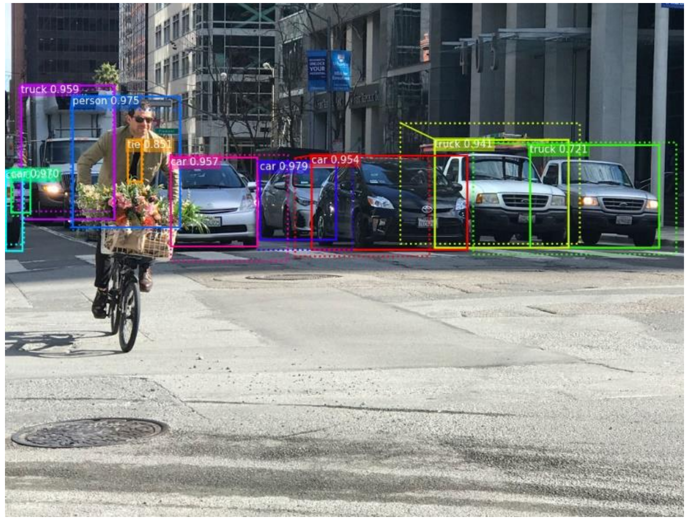
RoI Align

- Don’t round the center, height & width of each region.
  - Keep the floating point values
- Bilinear interpolation: used the quantized pixel values to interpolate the
- Max pooling is then applied to the properly aligned RoI

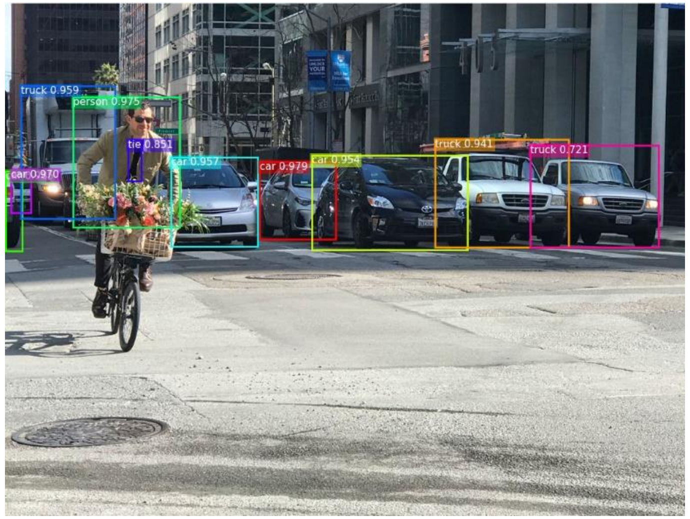


Mask Prediction

- Pixel level (instance) segmentation
- Applied to highest scoring 100 detection boxes



After nms.



Top boundary box predictions.

predictions from Mask