Trustworthy AI Systems

-- Image Segmentation

Instructor: Guangjing Wang

guangjingwang@usf.edu

Last Lecture

- Image classification
- Convolutional neural network
- Some practices for project

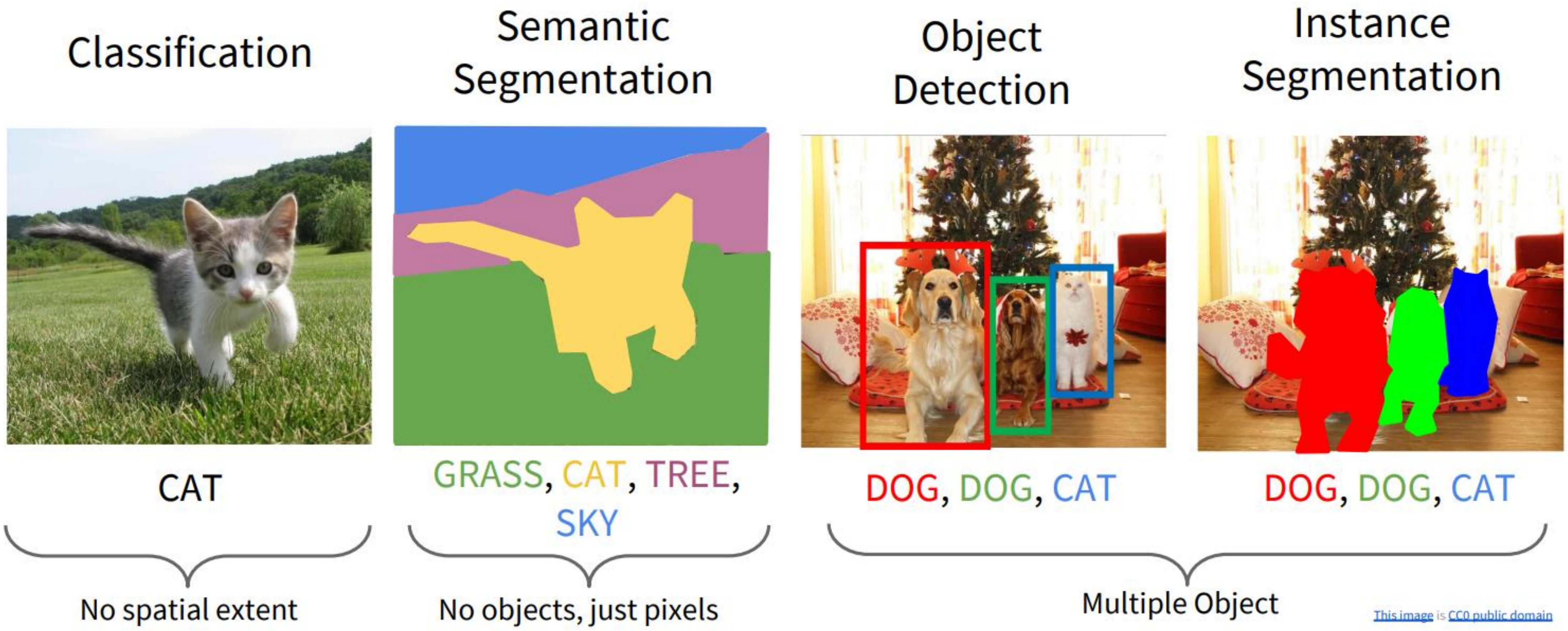
Homework 1: Paper Review

- Paper review is a basic task for a researcher
  - Paper Summary
  - Strengths
  - Weaknesses
  - Questions
  - Future Opportunities

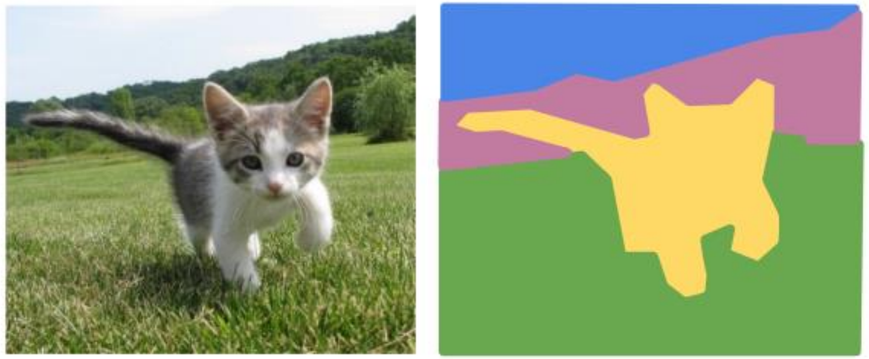
When you read a paper, thinking:

- What is the research problem and motivation?
- What are the challenges and technical contributions?
- How is the experimental evaluation?
- How is the related work, and overall presentation?

Computer Vision Tasks

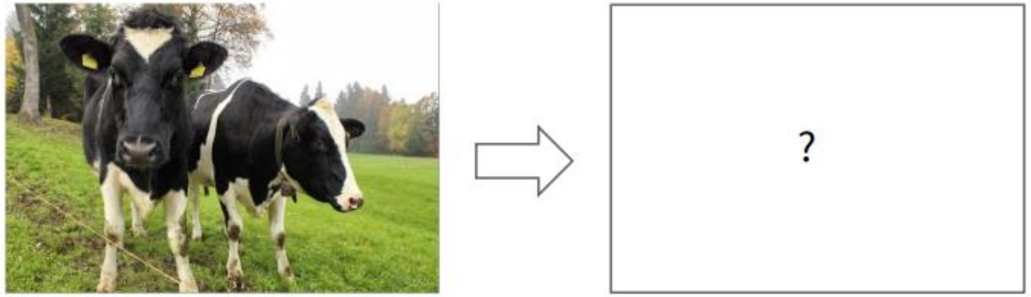


Semantic Segmentation: Problem



GRASS, CAT, TREE,
SKY, ...

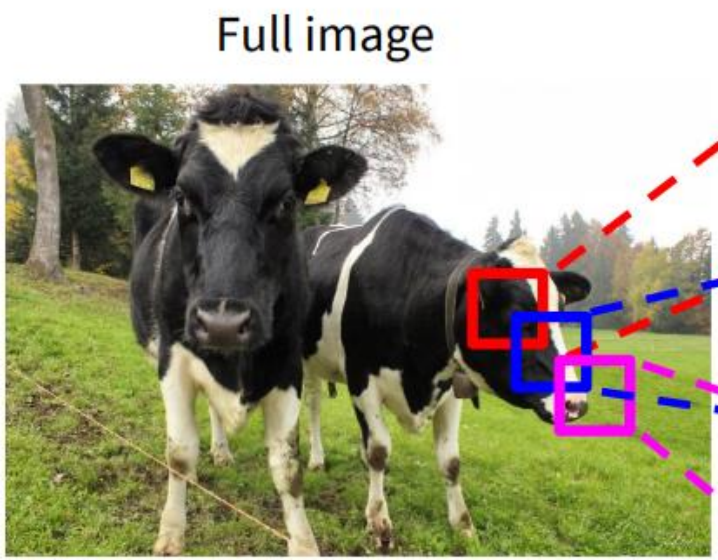
Paired training data: for each training image, each pixel is labeled with a semantic category.

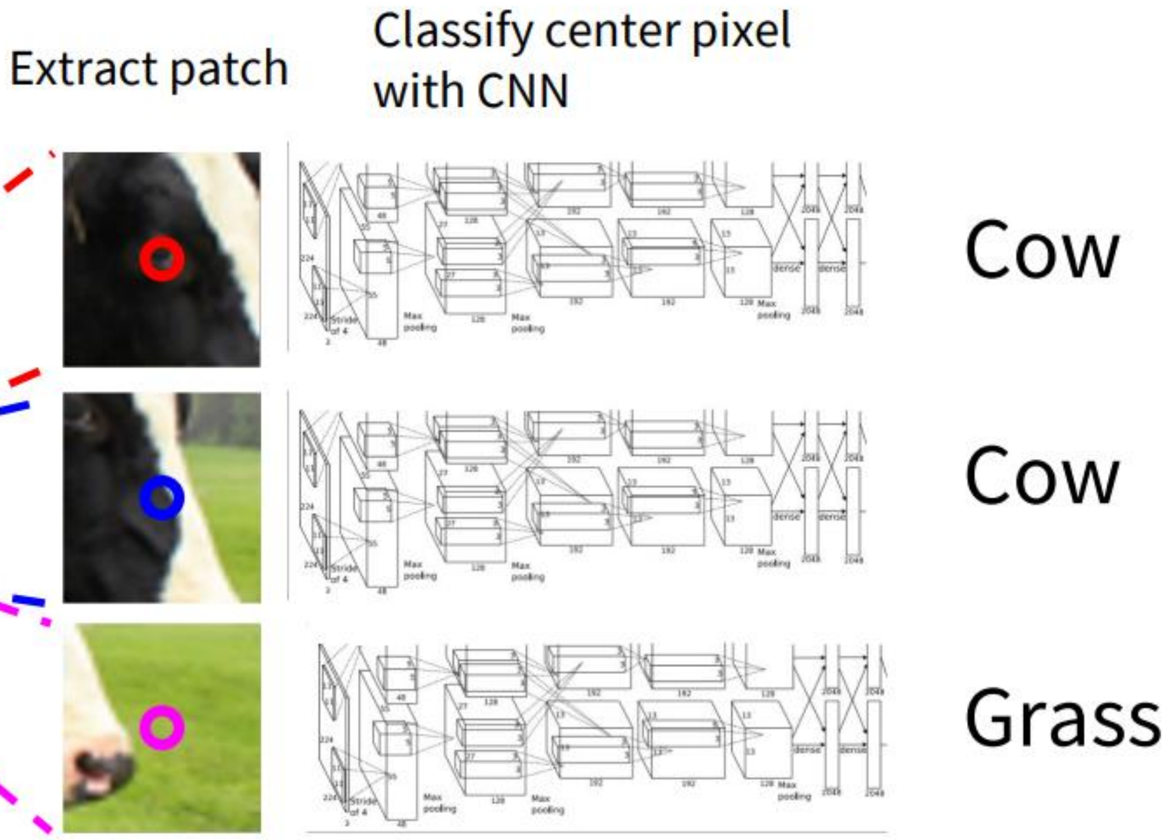


At test time, classify each pixel of a new image.

Label each pixel in the image with a category label.

Semantic Segmentation: Sliding Window

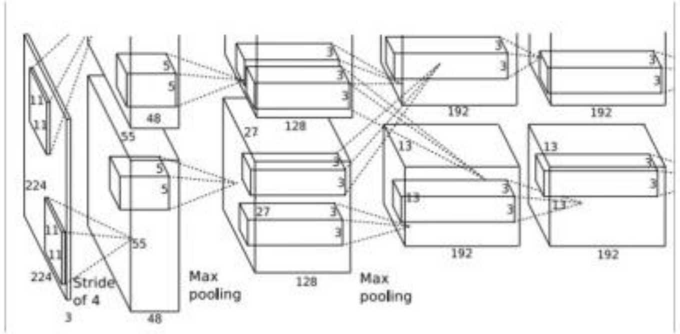




Problem: Very inefficient! Not reusing shared features between overlapping patches

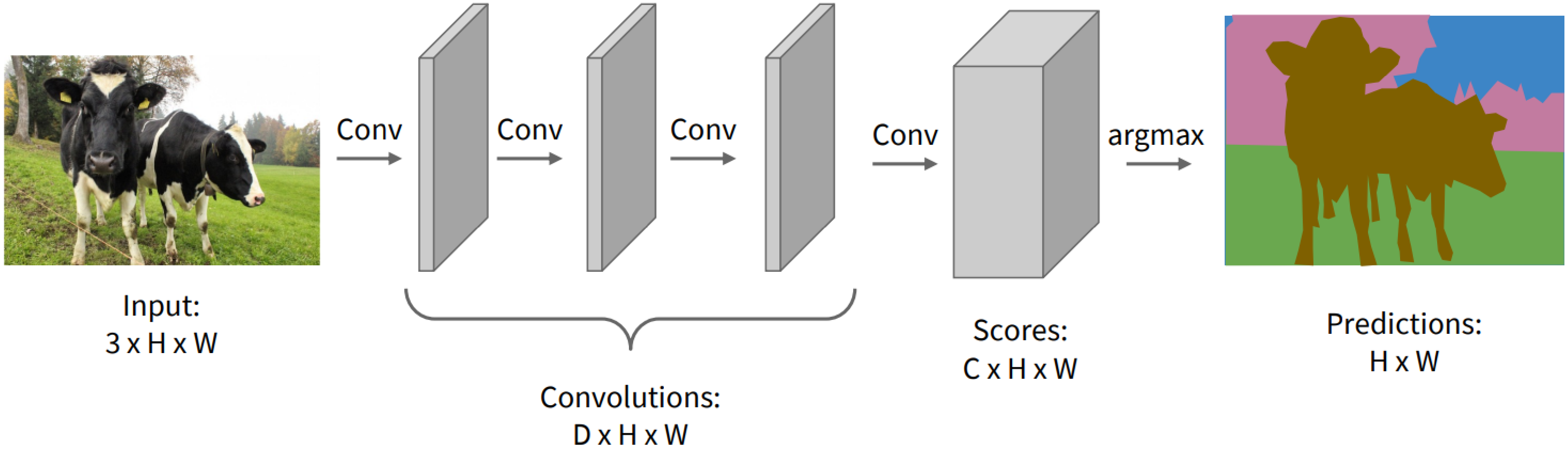
Semantic Segmentation: Convolution (1)

Full image



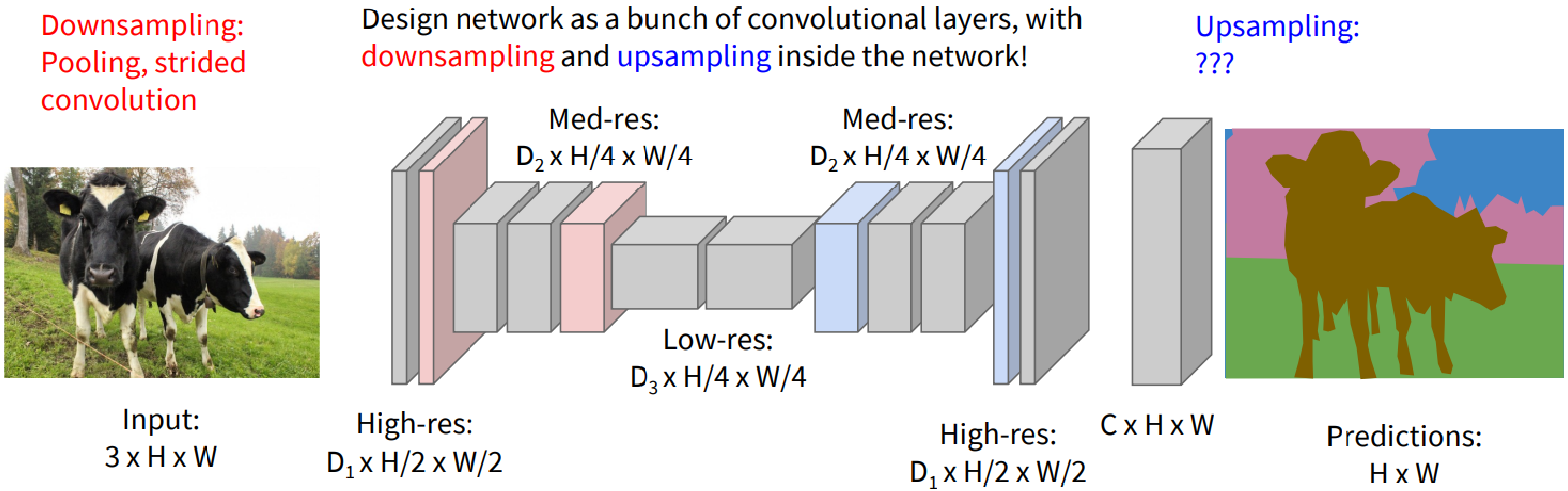
Encode the entire image with conv net, and do semantic segmentation on top

Semantic Segmentation: Convolution (2)



Potential problem?

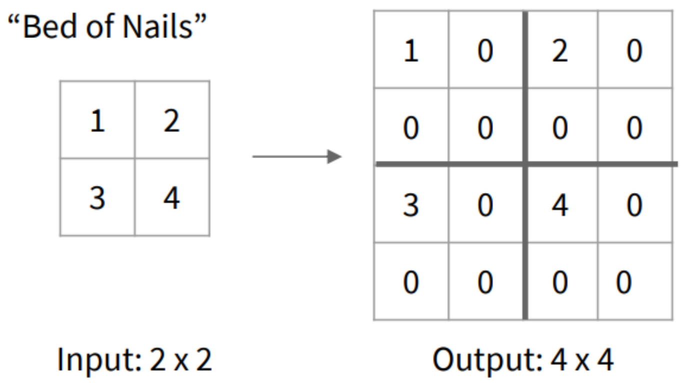
Semantic Segmentation: Convolution (3)



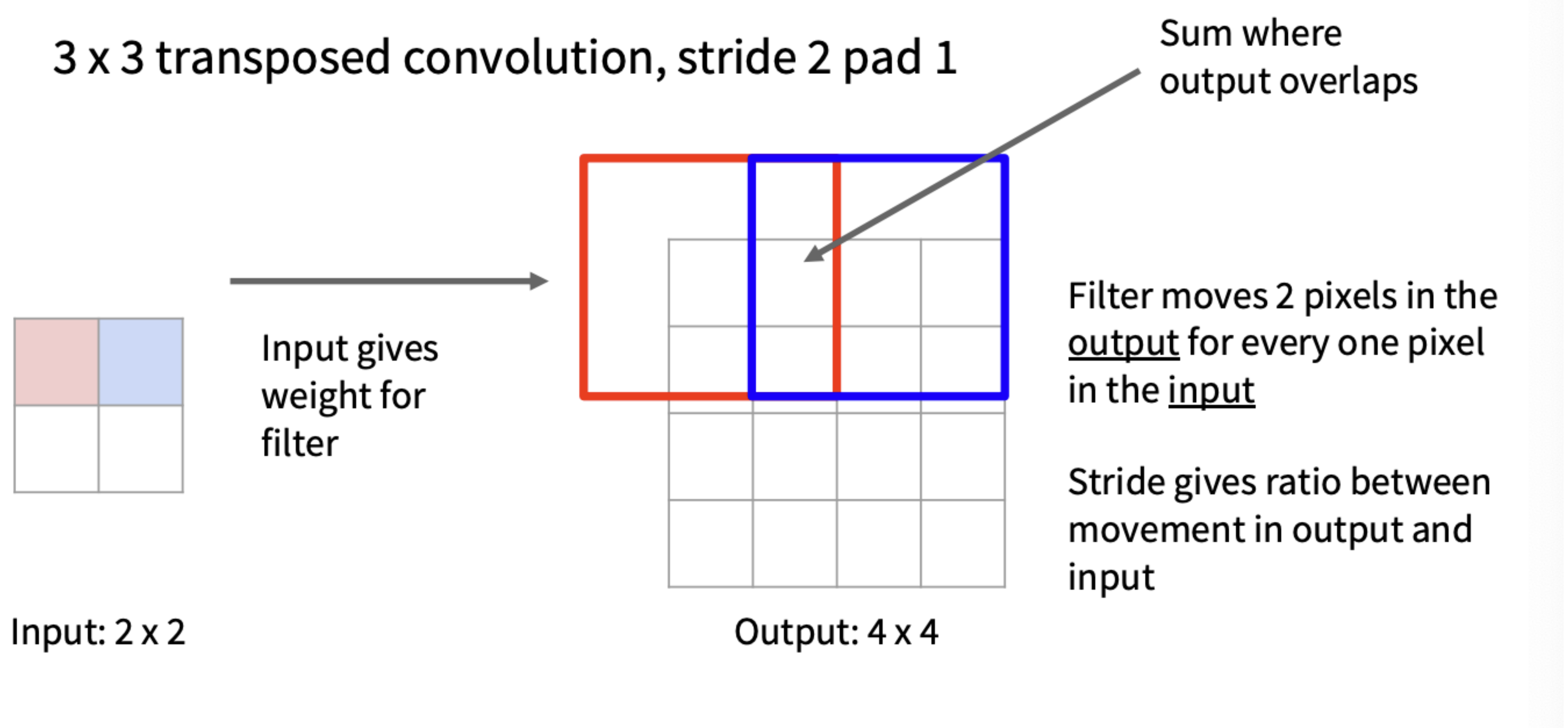
Upsampling

- Non-learnable upsampling
  - Fill the same
  - Fill zeros
  - Remember location then fill
  - You design it…

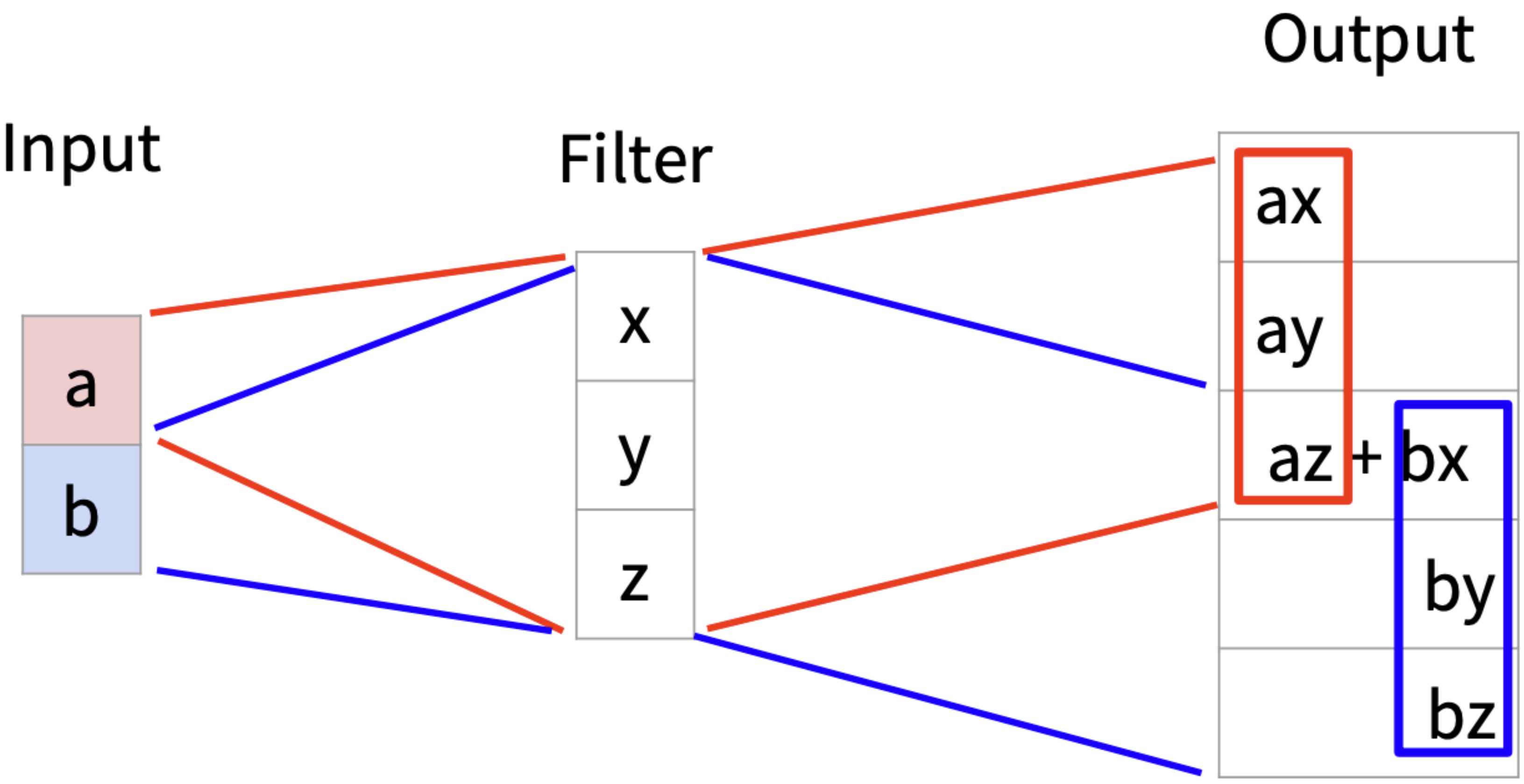
- Learnable upsampling
  - Transposed convolution



Upsampling: Transposed Convolution



Learnable Upsampling: 1D Example



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

Convolution as Matrix Multiplication

We can express convolution in terms of a matrix multiplication

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & z & 0 & 0 & 0 \\ 0 & 0 & x & y & z & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

Example: 1D conv, kernel size=3, <u>stride=2</u>, padding=1

Transposed convolution multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T\vec{a}$$

$$\begin{bmatrix} x & 0 \\ y & 0 \\ z & x \\ 0 & y \\ 0 & z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az + bx \\ by \\ bz \\ 0 \end{bmatrix}$$

Example: 1D transposed conv, kernel size=3, <u>stride=2</u>, padding=0

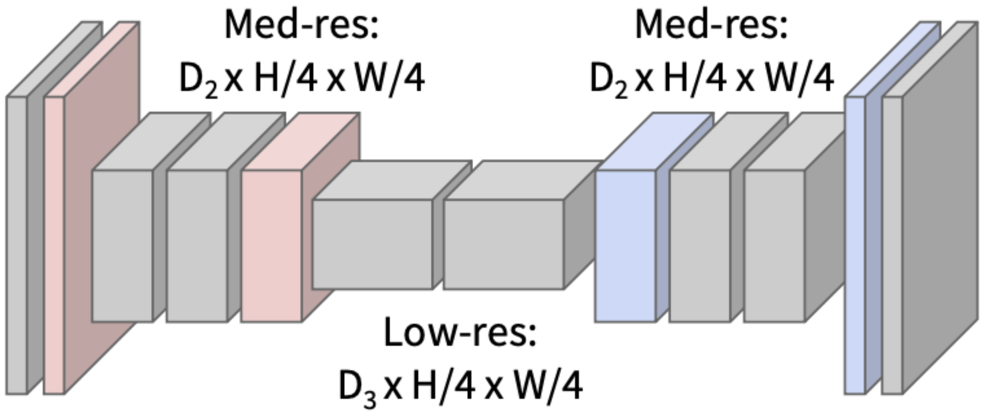
Semantic Segmentation: Fully Convolutional

Downsampling:
Pooling, strided
convolution

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

Upsampling:
Unpooling or strided
transposed convolution

Input:
3 x H x W



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

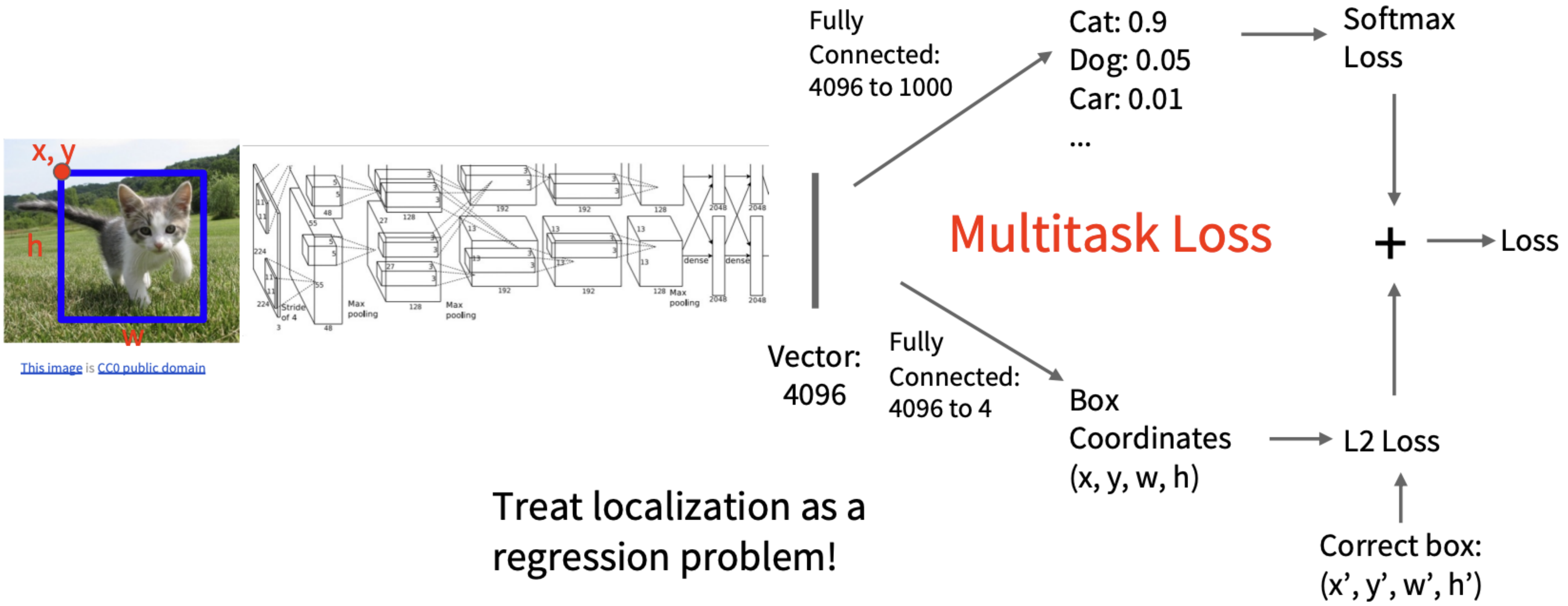
High-res:
D_1 x H/2 x W/2

Predictions:
H x W

Take a break

https://www.youtube.com/watch?v=JIPbilHxFbI

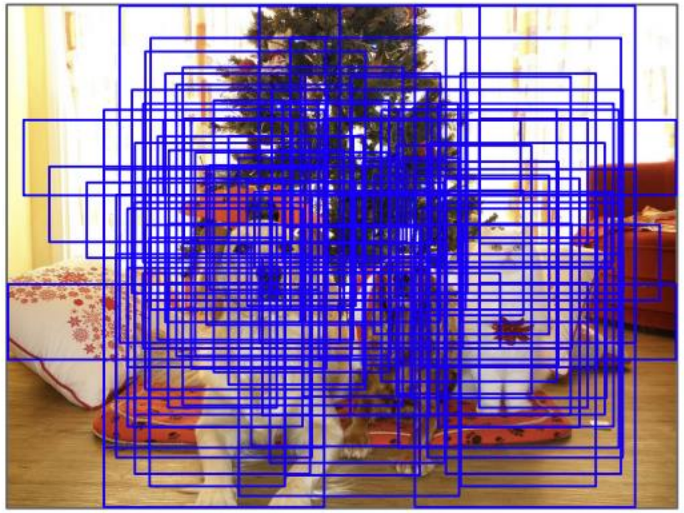
Object Detection: Classification + Localization



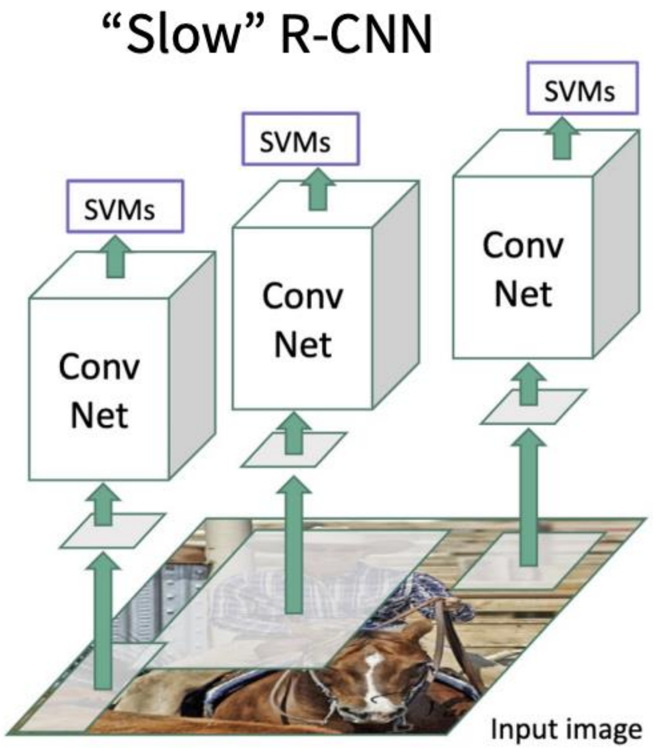
This image is CC0 public domain

Object Detection

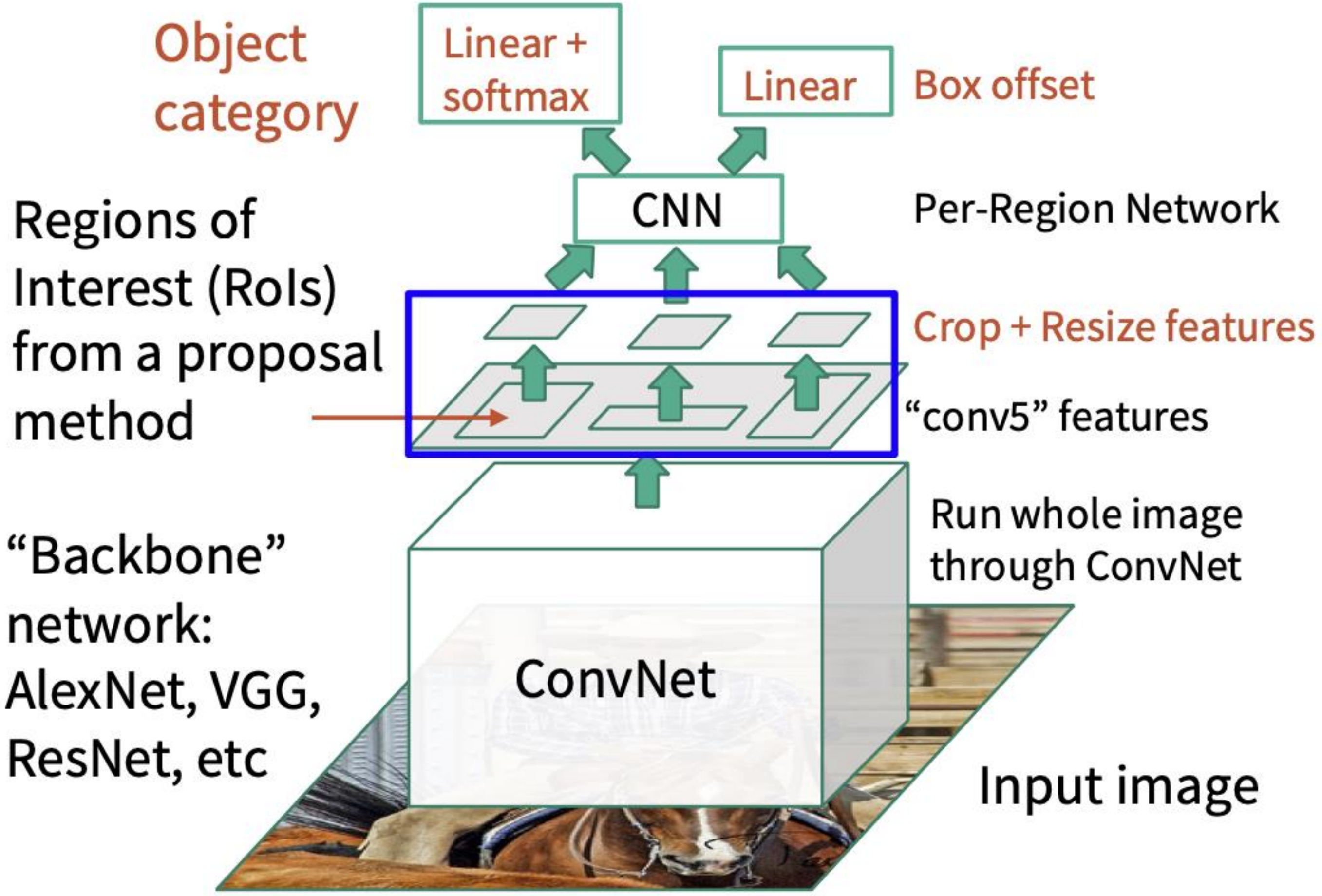
- What if there are multiple objects?
  - Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



R-CNN and Fast R-CNN

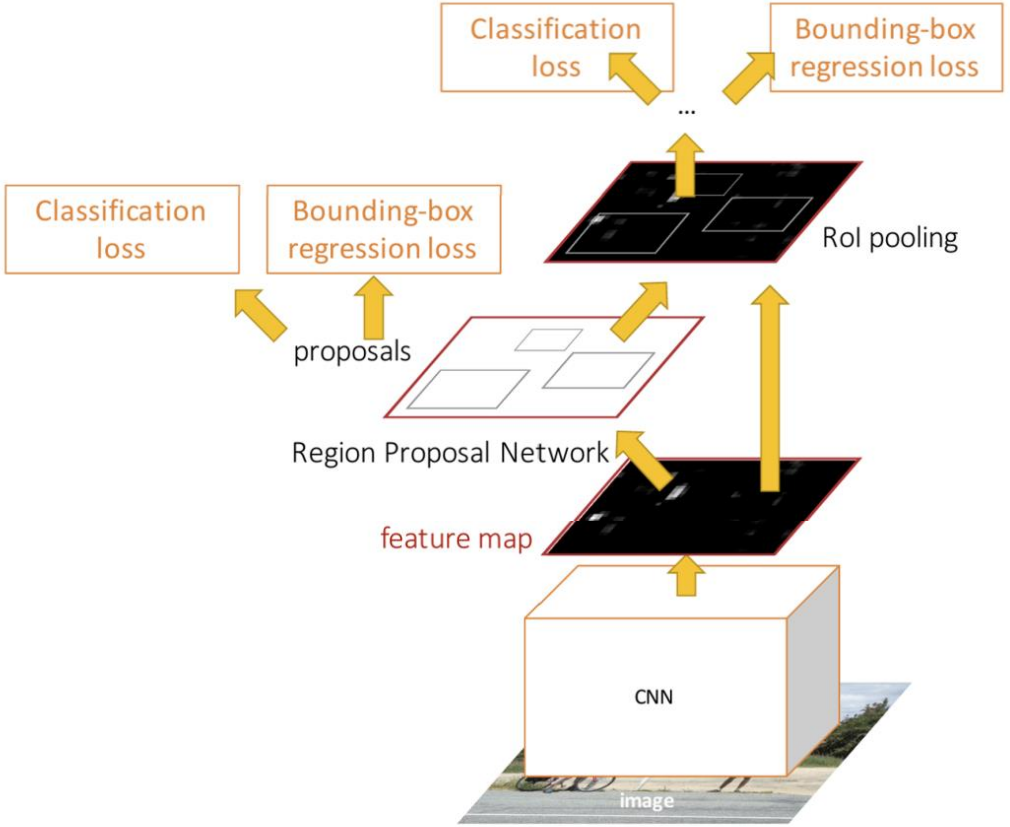


extracts around 2000 bottom-up region proposals,

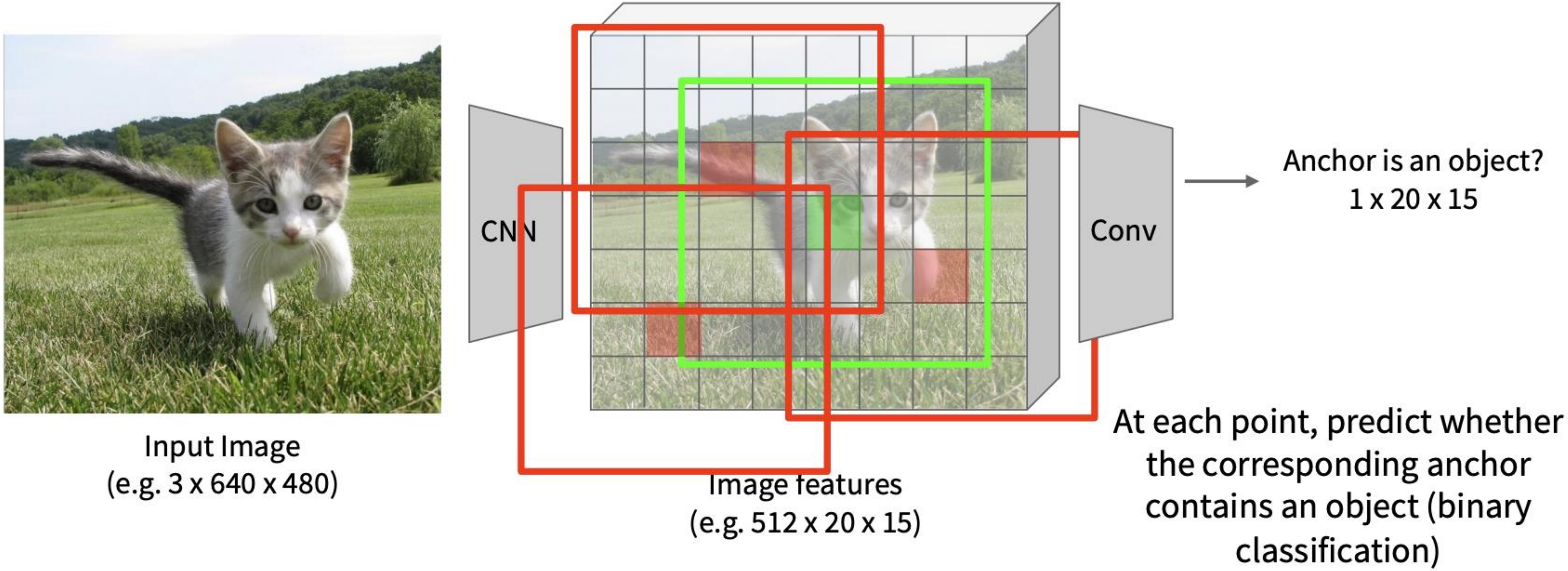


Faster R-CNN: Make CNN Do Proposals

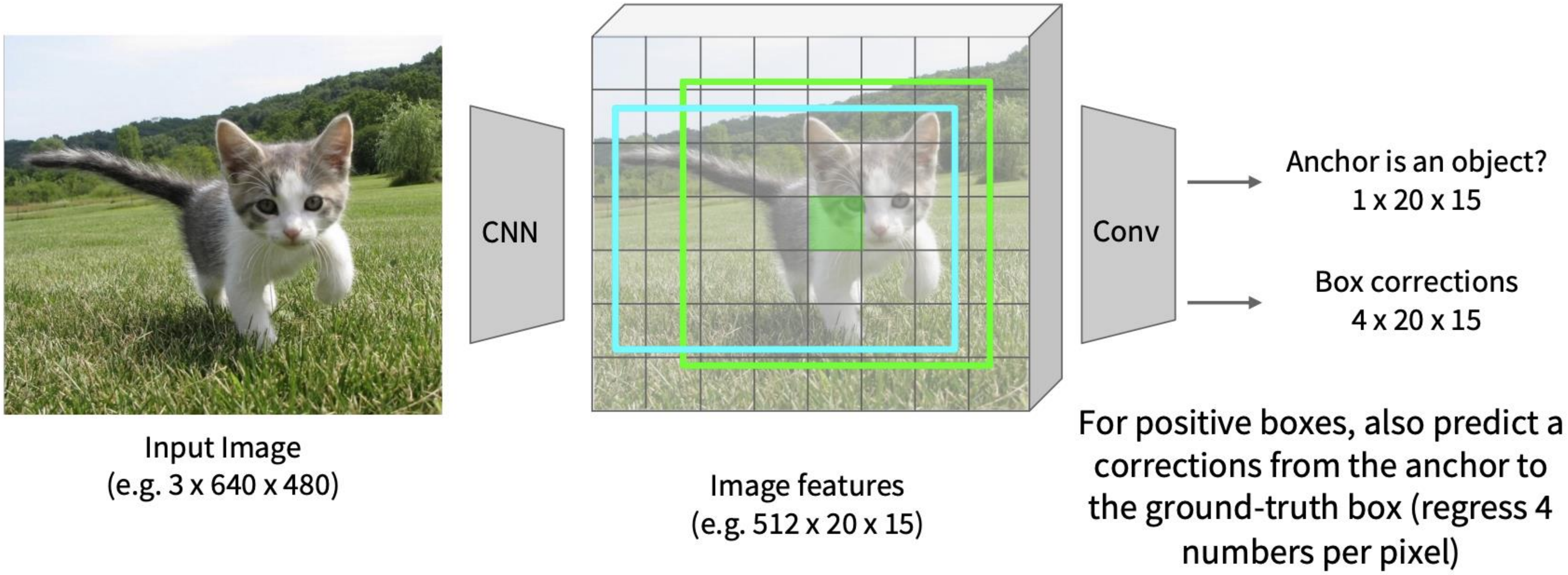
- Insert Region Proposal Network (RPN) to predict proposals from features



Region Proposal Network (1)



Region Proposal Network (2)



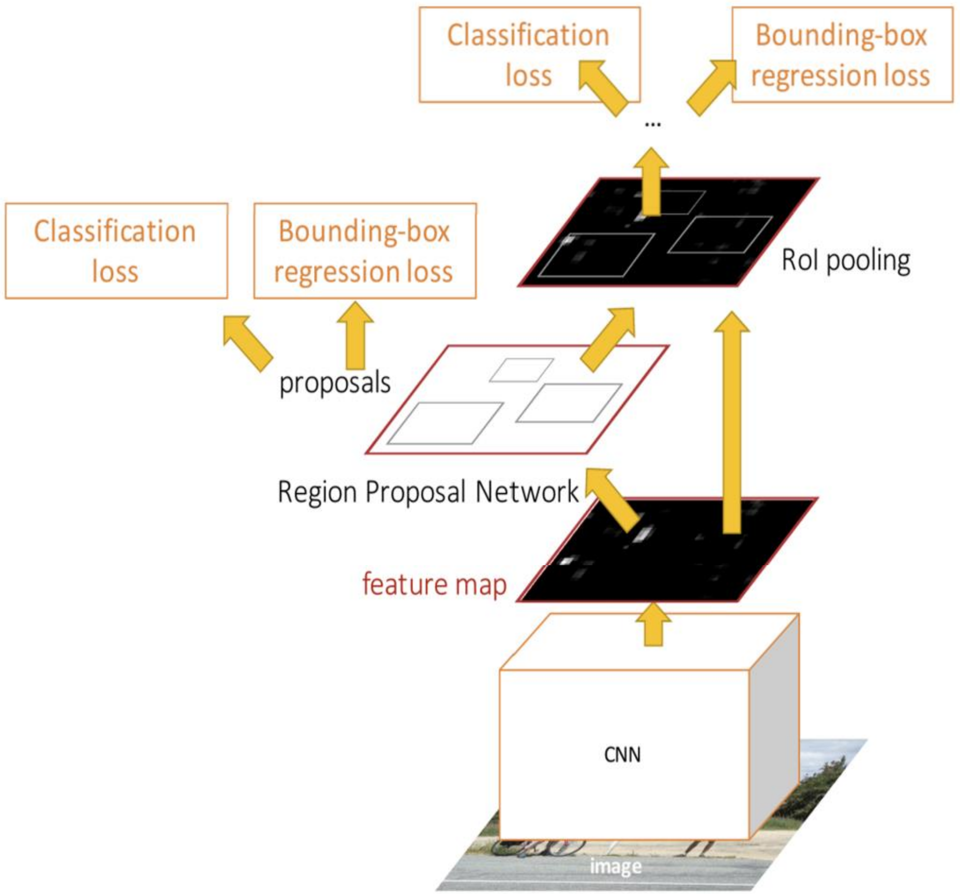
Faster R-CNN: Two Stages

Jointly train with 4 losses:

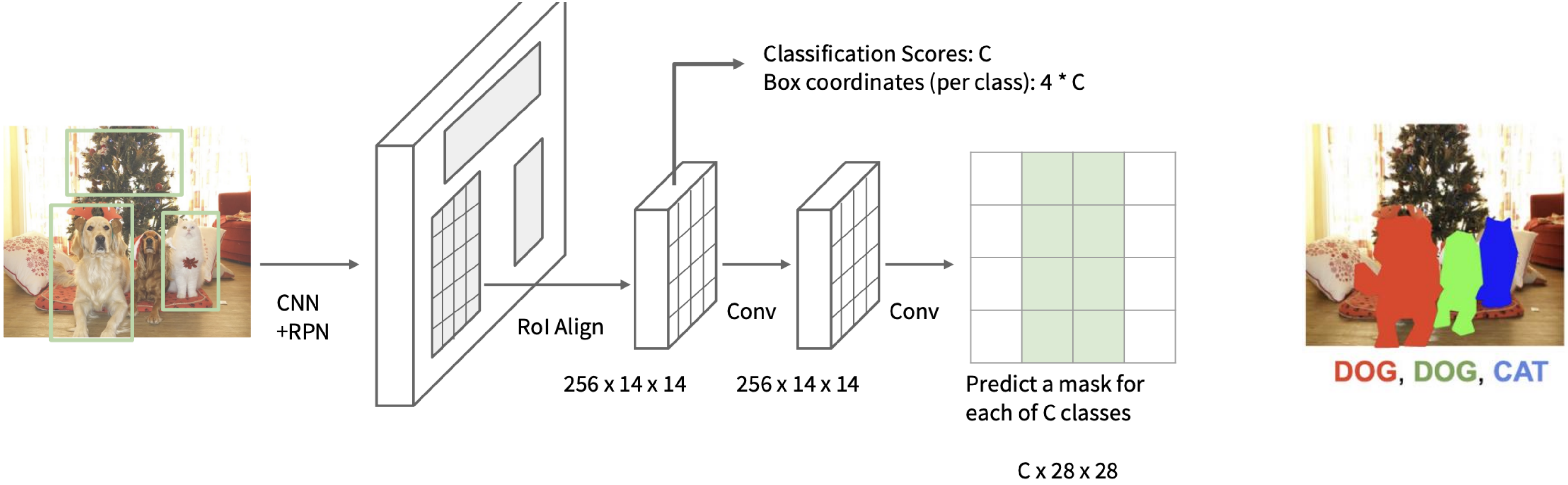
- RPN classify object / not object
- RPN regress box coordinates
- Final classification score (object classes)
- Final box coordinates

First stage: Run once per image
- Backbone network
- Region proposal network

Second stage: Run once per region
- Crop features: RoI pool / align
- Predict object class
- Prediction bbox offset

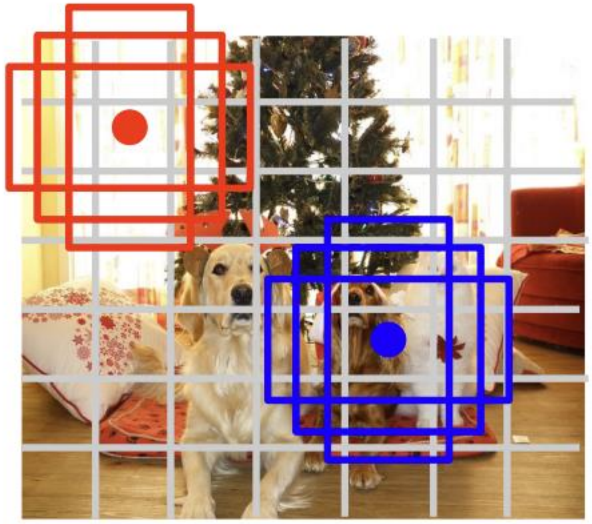


Instance Segmentation: Mask R-CNN



Yolo: Single Stage Object Detector

Input image
3 x H x W



Divide image into grid
7 x 7

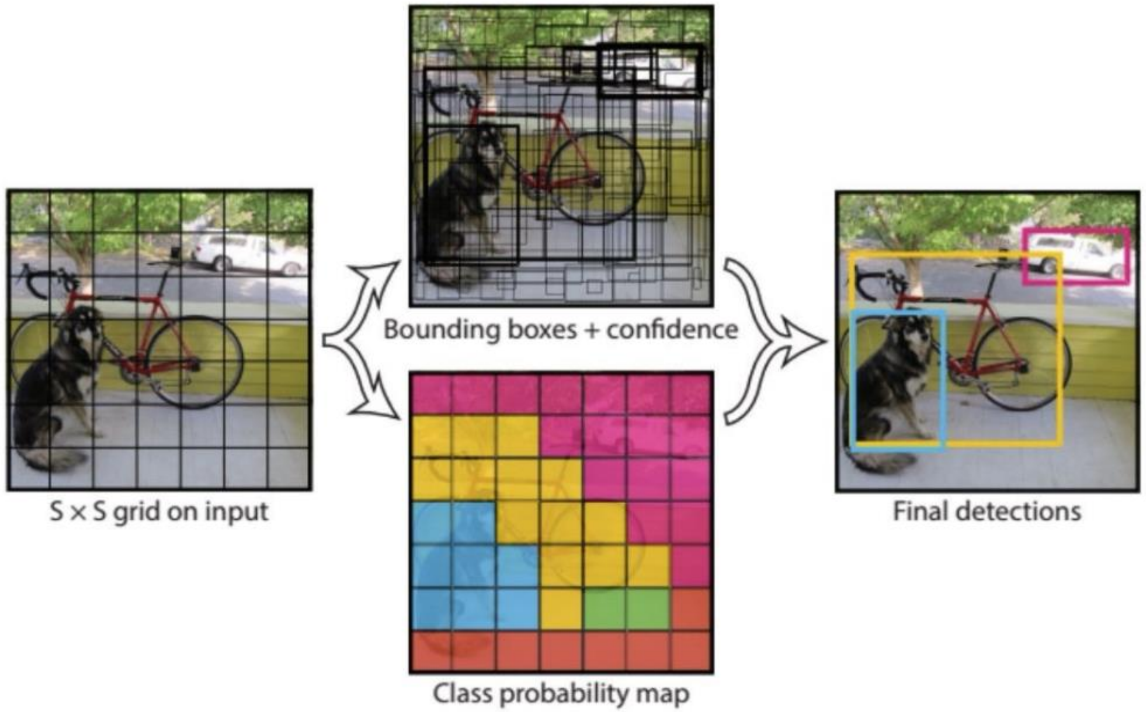
Image a set of base boxes centered at each grid cell Here B = 3

Within each grid cell:

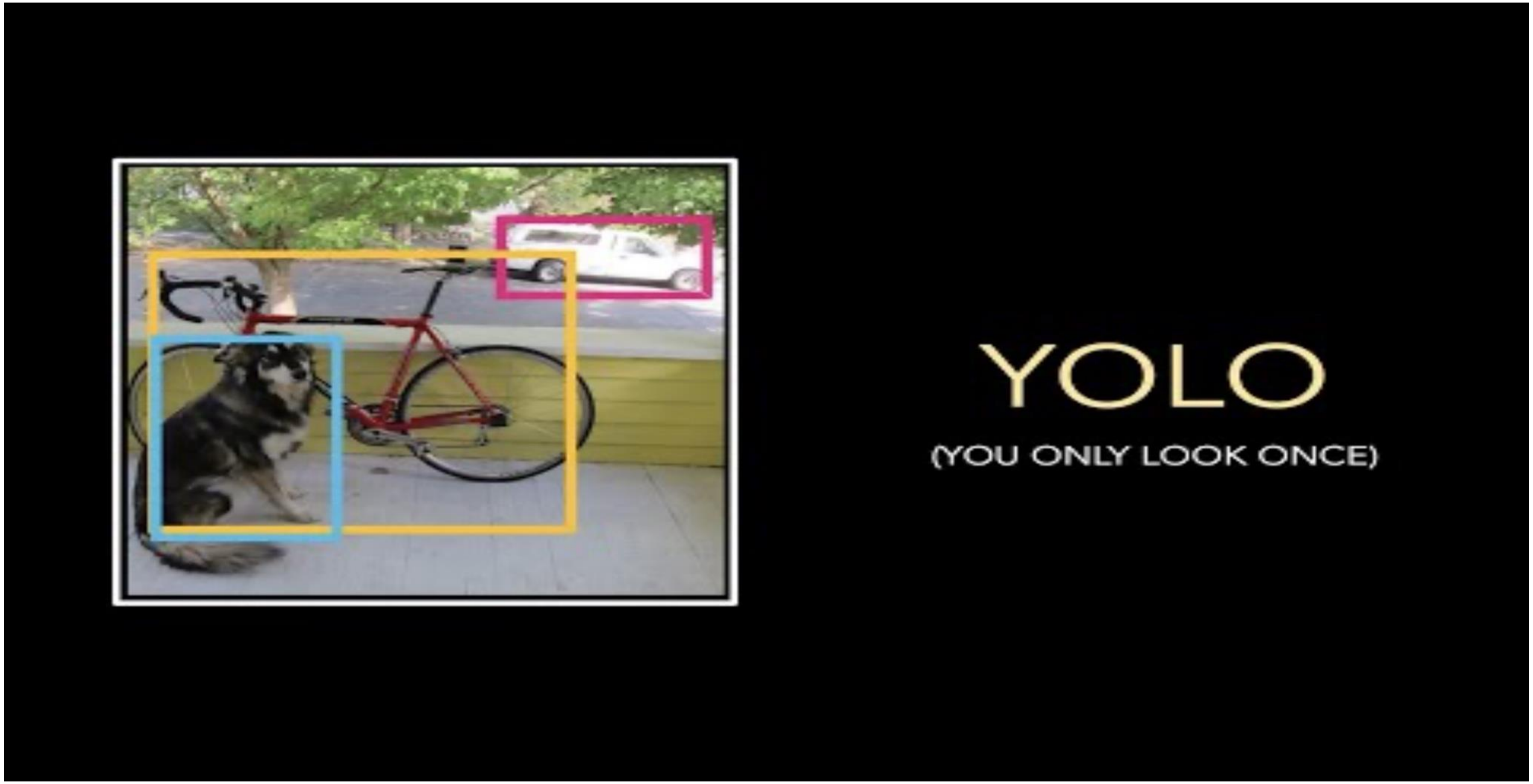
- Regress from each of the B base boxes to a final box with 5 numbers:
  (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)
- Looks a lot like RPN, but category-specific!
- Output: 7x7x(5*B+C)

Yolo: Non-Max Suppression

- If IoU(P1, P2) > Threshold: P = argmax(C(p1), C(p2))
  - Eliminating bounding boxes that have a high overlap with the box that has the highest confidence score

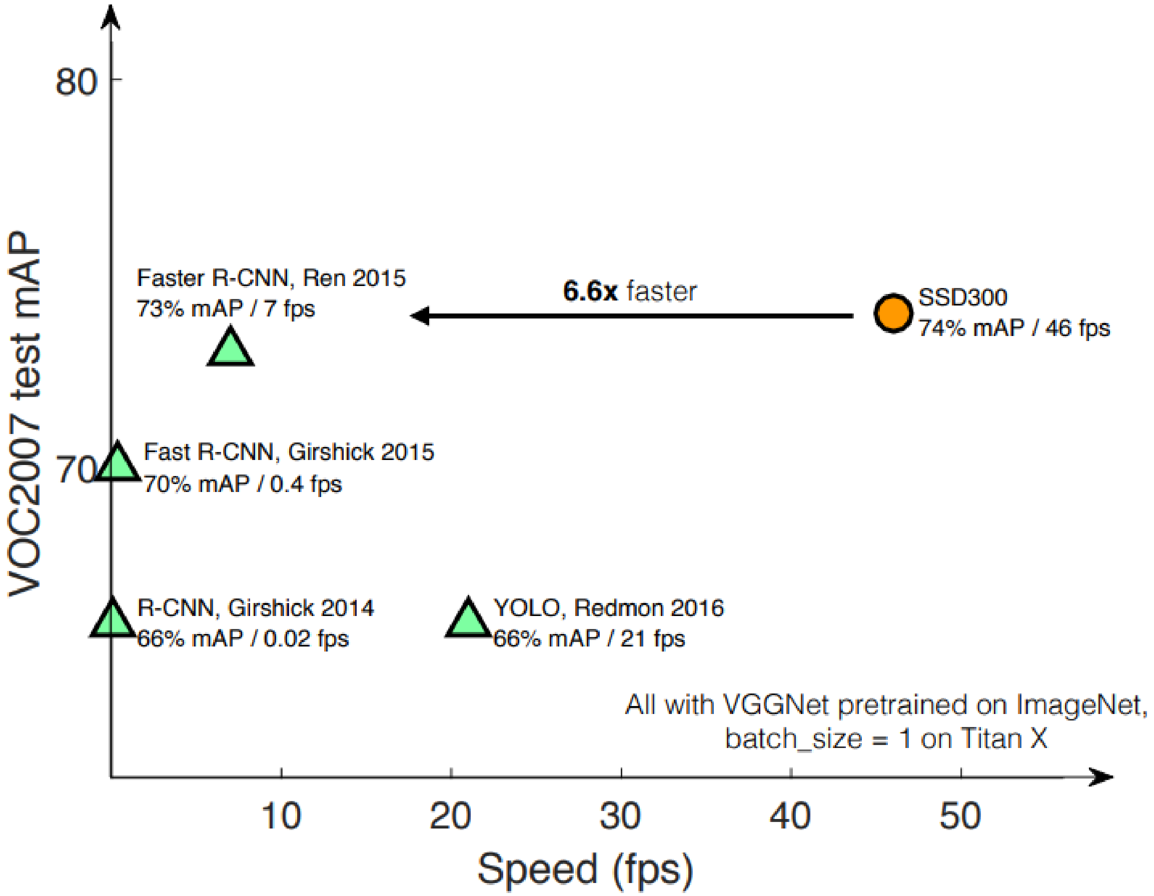


YOLO: Model as a Regression Problem



https://youtu.be/svn9-xV7wjk

Single-shot VS Two-shot Detector



https://www.cs.unc.edu/~wliu/papers/ssd_eccv2016_slide.pdf

Object Detection: Evaluation Metrics

- Intersection over Union (IoU)
  - Predicted bounding box (A) and ground truth bounding box (B)

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- Average Precision (AP)
  - The precision-recall curve that is created by varying the detection threshold.
  - mean Average Precision (mAP), which calculates AP for each class and then take the average

Midterm Project Group

- Please find your team member (1-3 members in a group)
- Sign your group in Canvas
- Random sign-up will be executed on Sep. 5th.

References

- https://cs231n.stanford.edu/slides/2024/lecture_9.pdf
- https://encord.com/blog/yolo-object-detection-guide/
- https://github.com/ultralytics/ultralytics
- https://github.com/facebookresearch/detectron2