Computer Vision – TP12
Object Detection Using Deep Learning

Miguel Coimbra, Hélder Oliveira

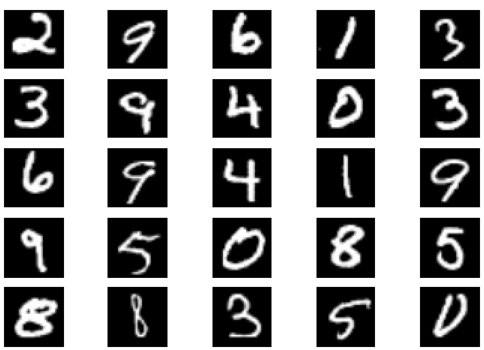
U.PORTO FC

Outline

- Object Detection Using Deep Learning
  - Object Detection
  - Location and Classification
  - Instance Segmentation

Image Classification

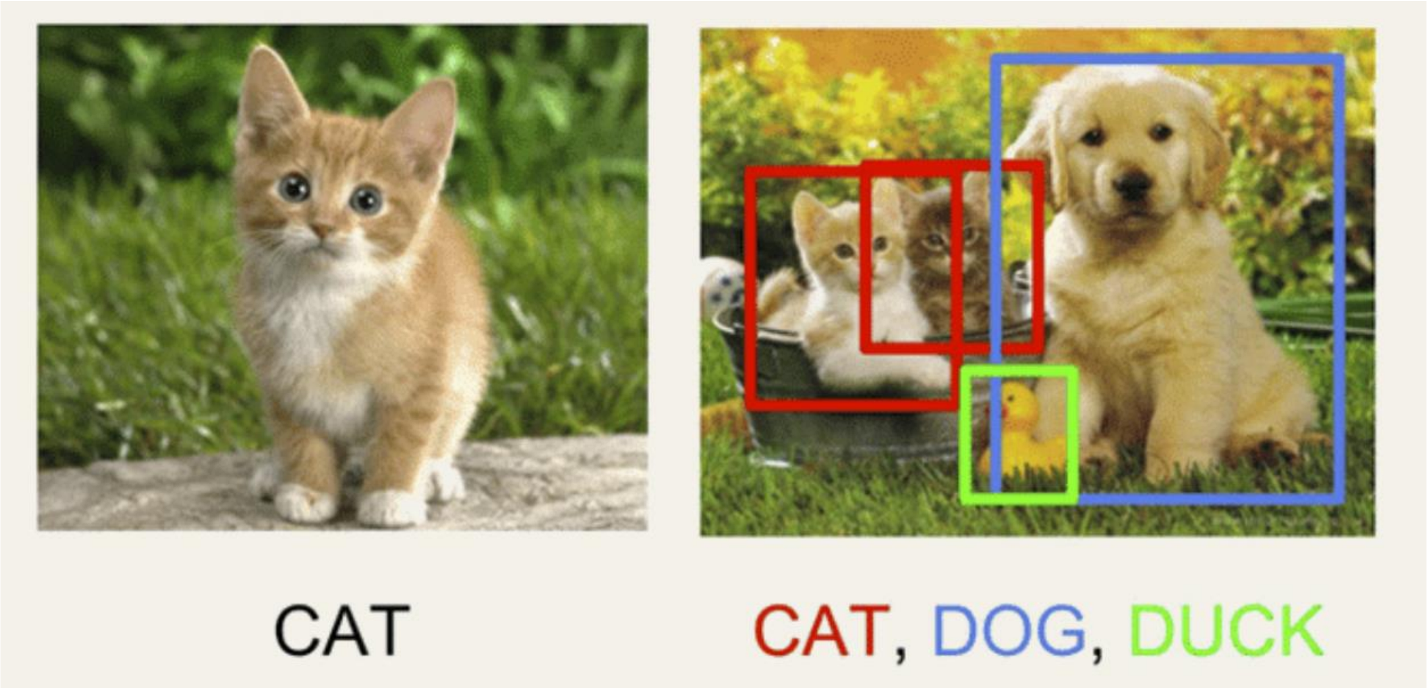
- K Classes
- Task: assign a class label to the image



Digit classification (MNIST)

Object recognition (Caltech-101)

Classification vs. Detection

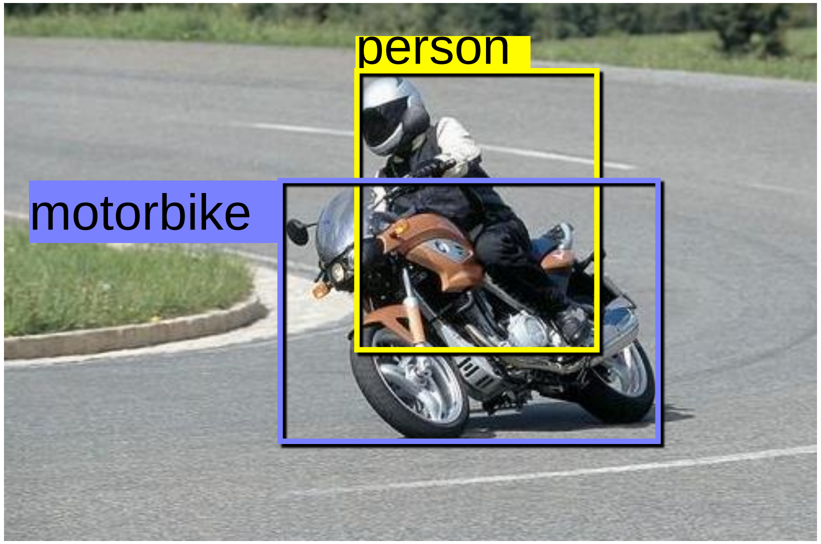


https://ambolt.io/en/image-classification-and-object-detection/

Problem formulation

{ airplane, bird, motorbike, person, sofa }

Input

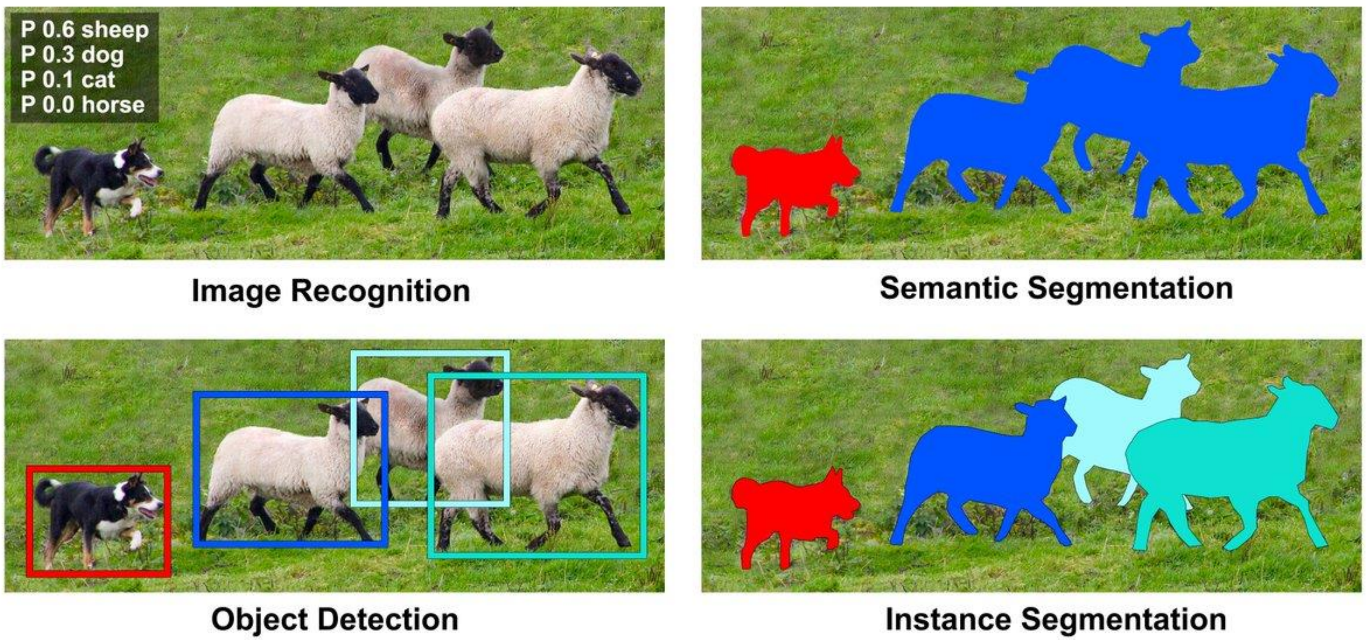


Desired output

Ross Girshick, Microsoft Research

Object Detection vs segmentation

- Object detection: given an input image, determine if there are objects of a given class (e.g. faces, people, cars, animals..) in the image and where they are located



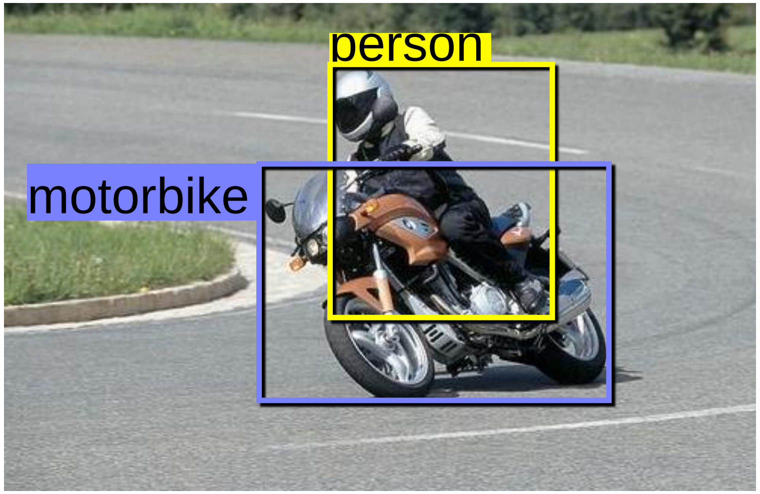
https://manipulation.csail.mit.edu/segmentation.html

Detection - Problems

1. Classifier must generalize over all exemplars of one class.
2. Negative class consists of everything else.
3. High accuracy (small FP rate) required for most applications.

Object Detection: Task Definition

- Input: Single RGB Image
- Output: A set of detected objects; For each object predict:
  - Category label (from fixed, known set of categories)
  - Bounding box (four numbers: x, y, width, height)

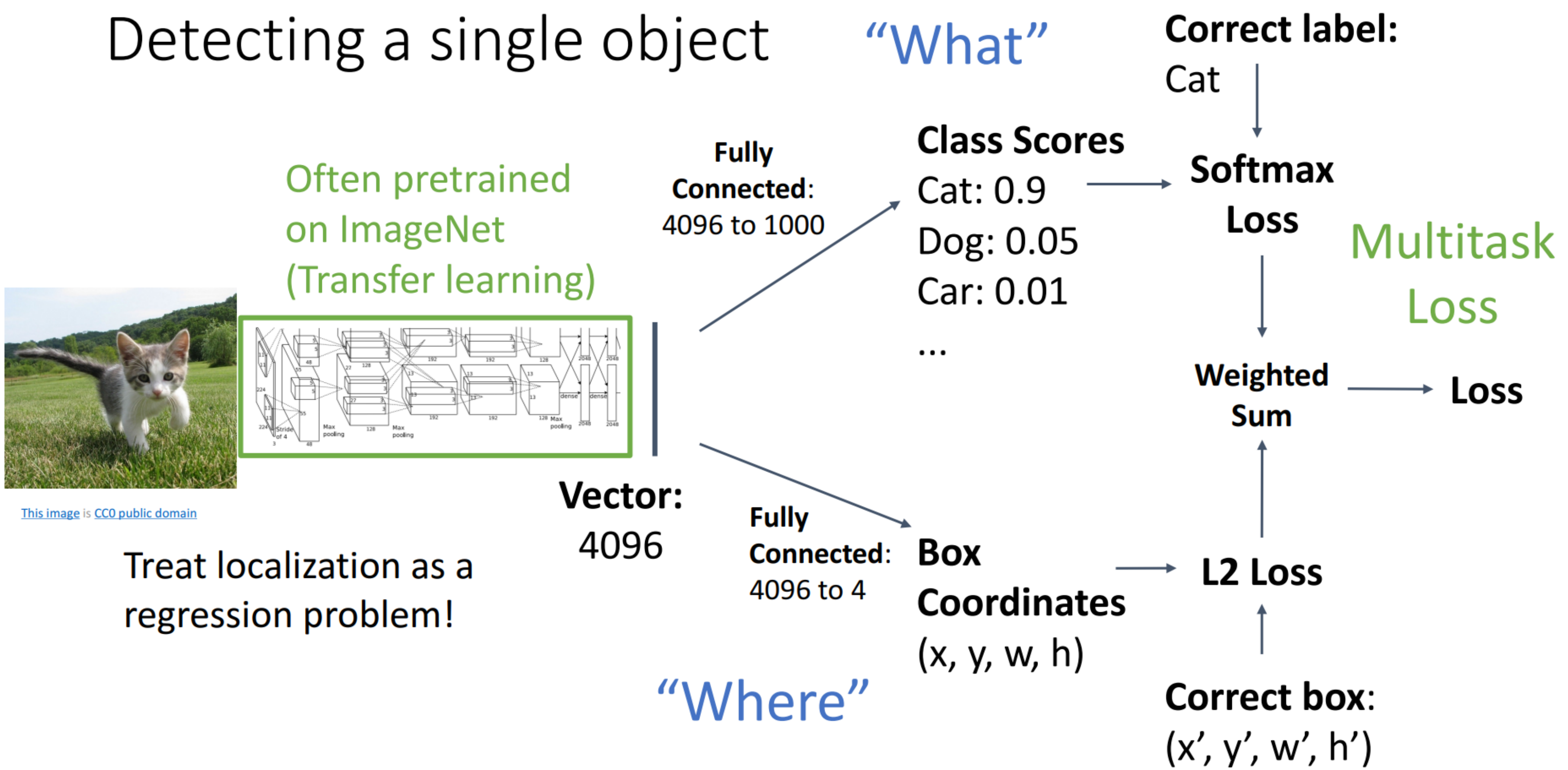


Ross Girshick, Microsoft Research

Object Detection: Challenges

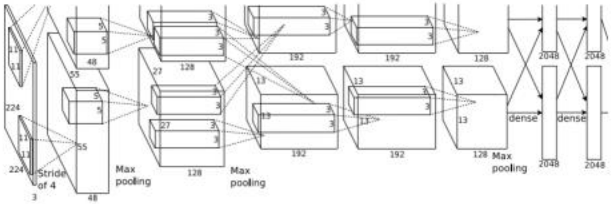
- Multiple outputs: Need to output variable numbers of objects per image
- Multiple types of output: Need to predict ”what” (category label) as well as “where” (bounding box)
- Large images: Classification works at 224x224; need higher resolution for detection, often ~800x600

Detection a single object

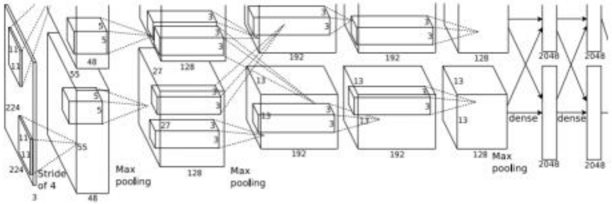


Michigan Online, Object Detection, Justin Johnson

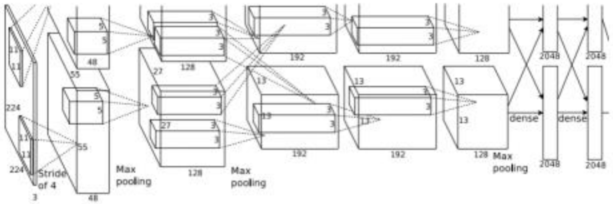
Detecting Multiple Objects

CAT: (x, y, w, h)

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

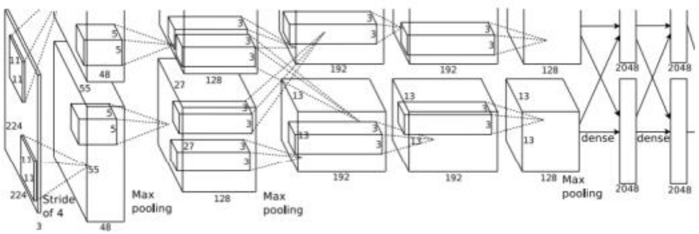



DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Michigan Online, Object Detection, Justin Johnson

Detecting Multiple Objects – Sliding Window

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

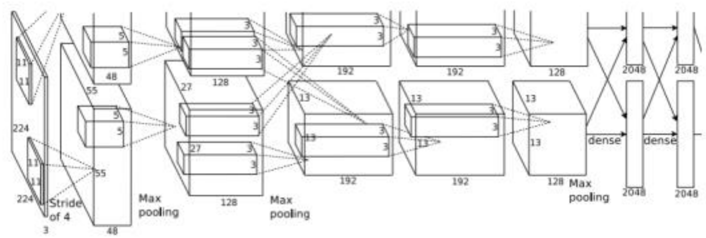


Dog? NO
Cat? NO
Background? YES

Michigan Online, Object Detection, Justin Johnson

Detecting Multiple Objects – Sliding Window

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

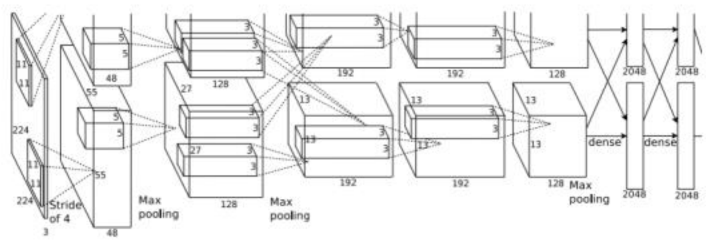


Dog? YES
Cat? NO
Background? NO

Michigan Online, Object Detection, Justin Johnson

Detecting Multiple Objects – 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

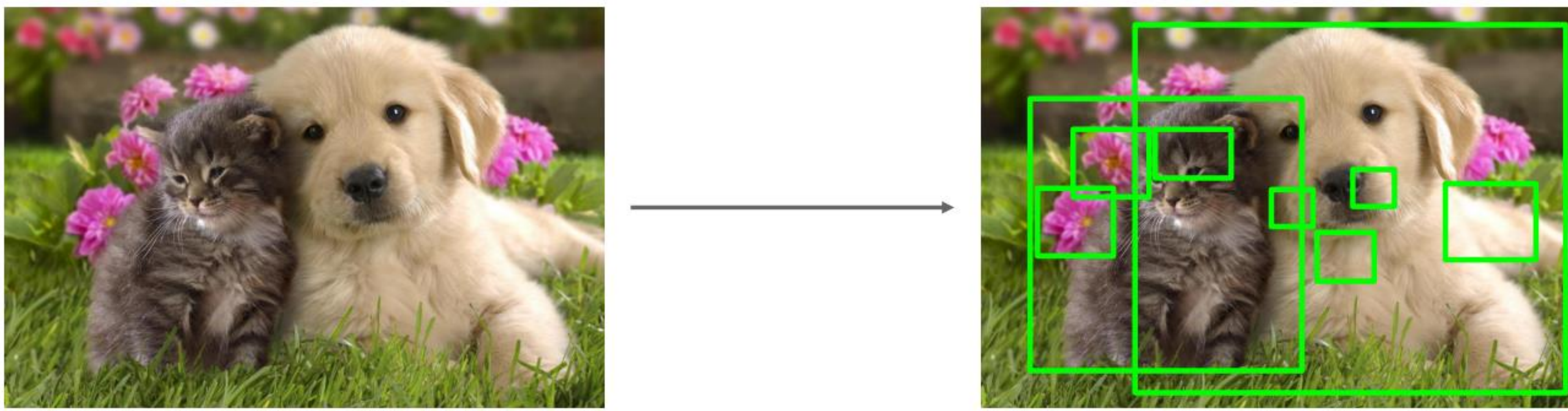
Issue: How many Windows can be tested???

Michigan Online, Object Detection, Justin Johnson

Detecting Multiple Objects – Region Proposals

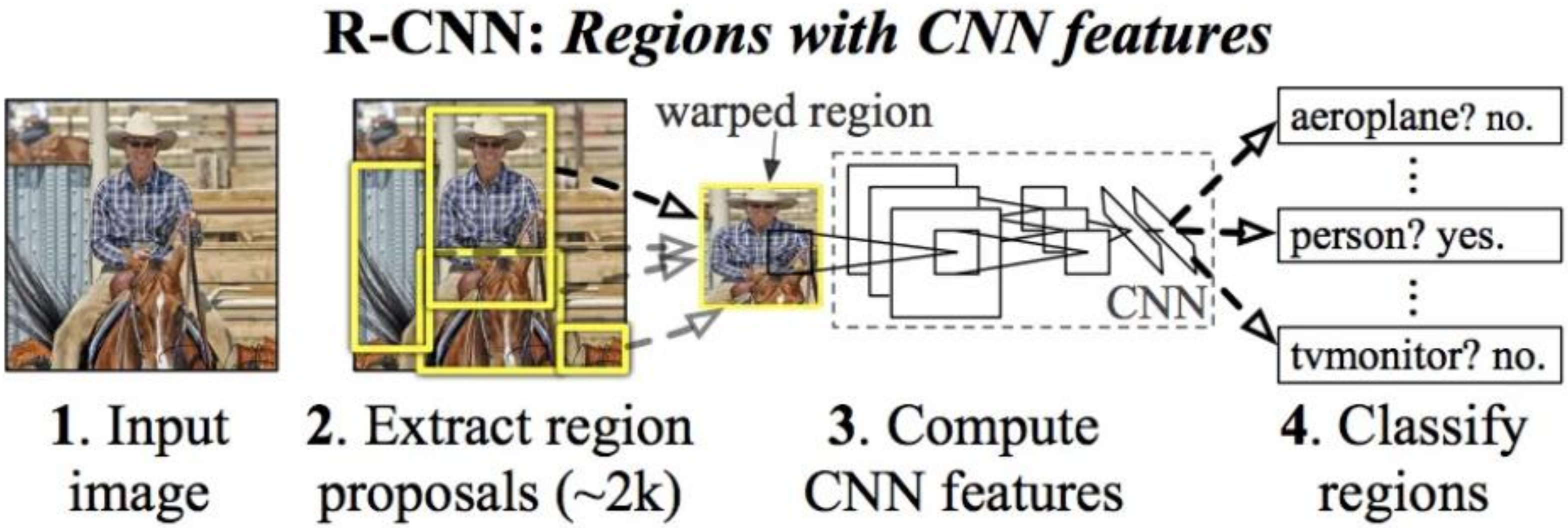
Region Proposals

- Find a small set of boxes that are likely to cover all objects
- Often based on heuristics: e.g. look for "blob-like" image regions
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



Michigan Online, Object Detection, Justin Johnson

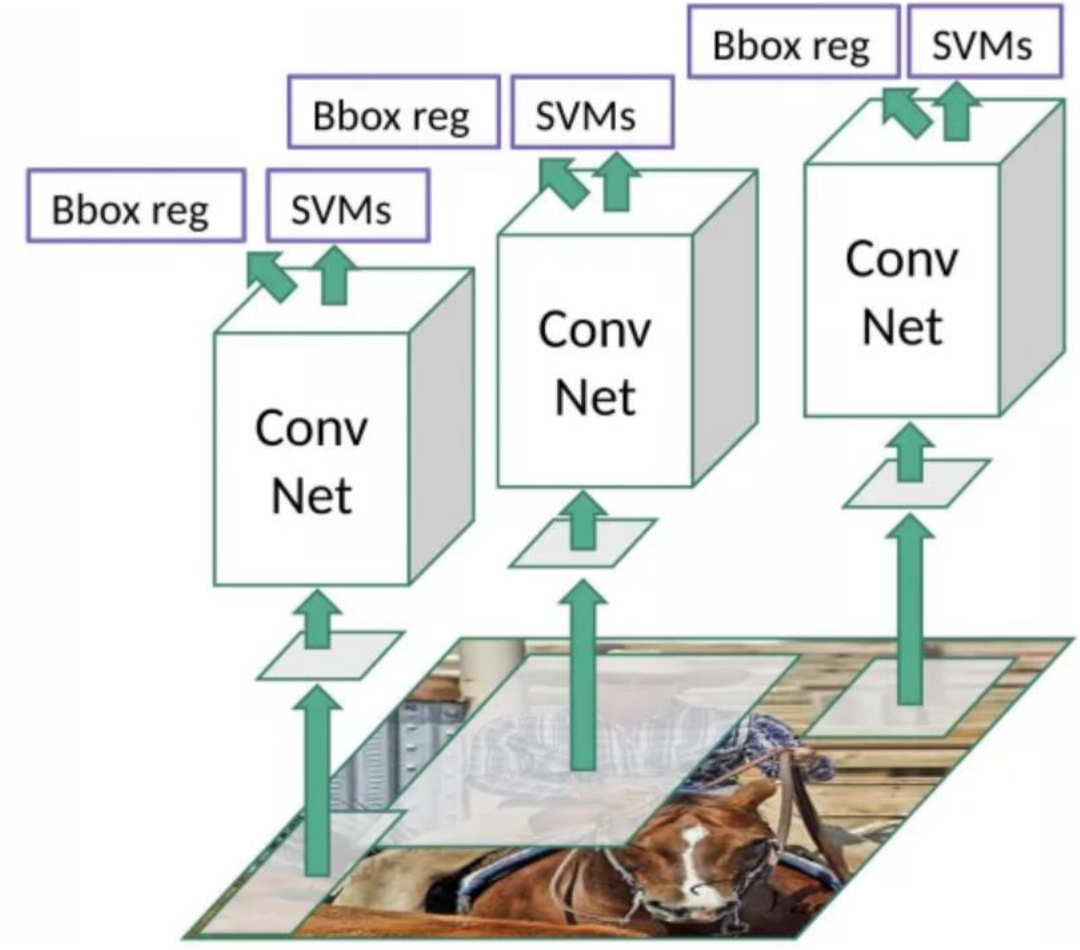
R-CNN



R-CNN

https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

R-CNN



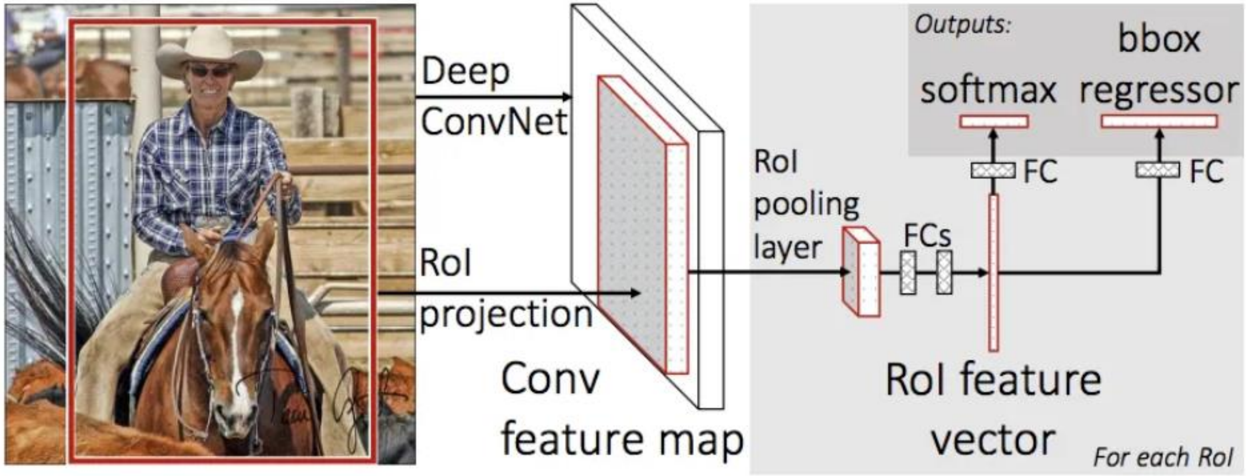
https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

R-CNN (Issues)

- It still takes a huge amount of time to train the network as you would have to classify 2000 region proposals per image.
- It cannot be implemented real time as it takes around 47 seconds for each test image.
- The selective search algorithm is a fixed algorithm. Therefore, no learning is happening at that stage. This could lead to the generation of bad candidate region proposals.

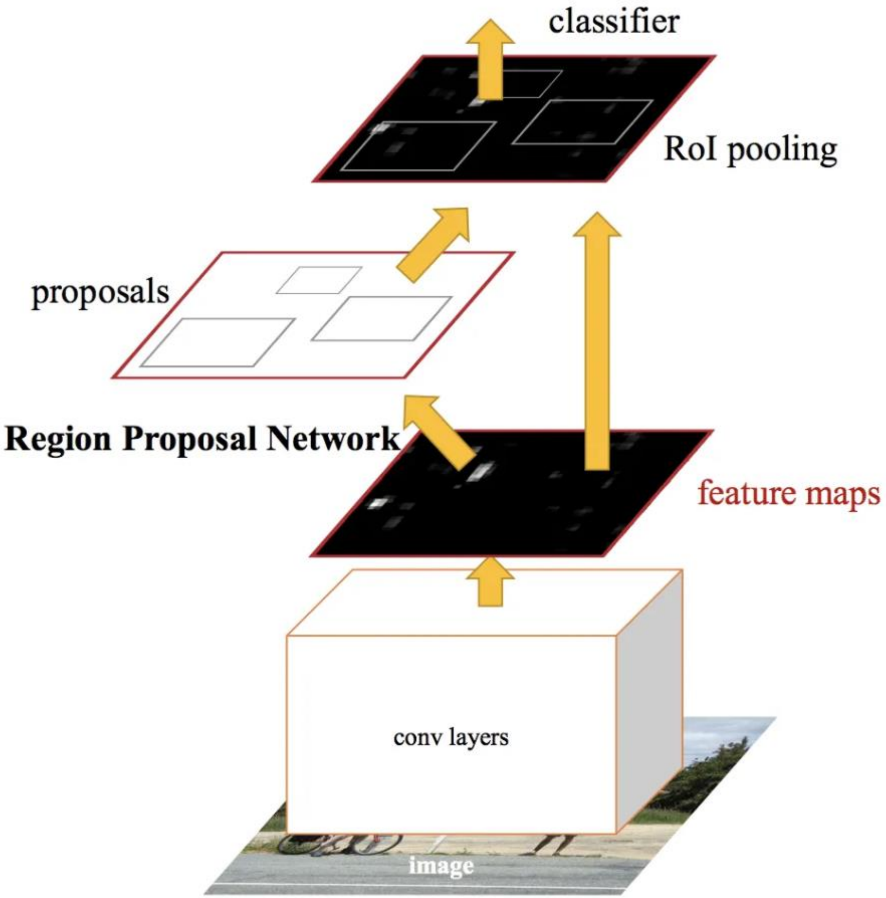
https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Fast R-CNN



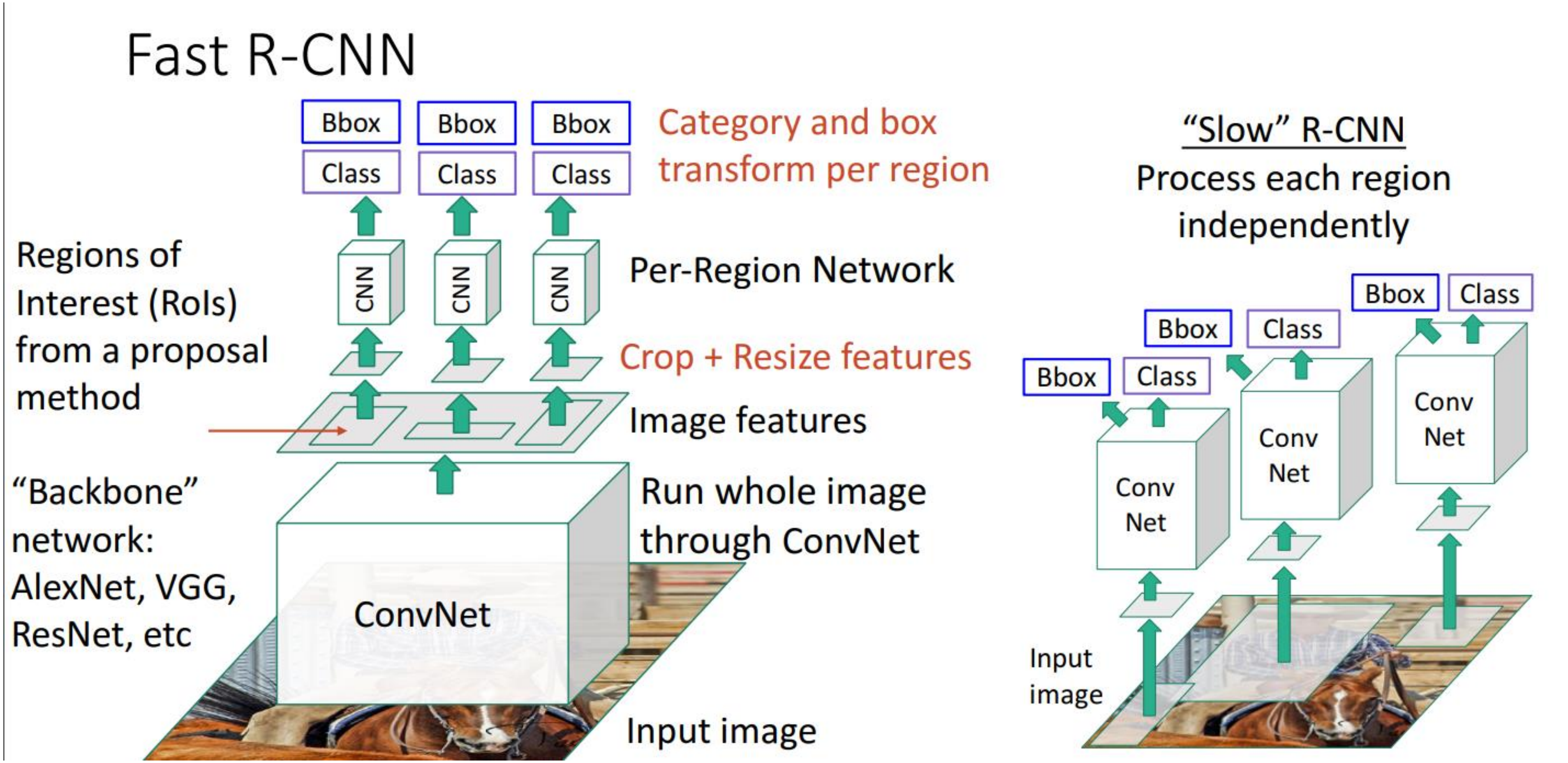
https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Fast R-CNN



https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Comparing R-CNN with Fast R-CNN

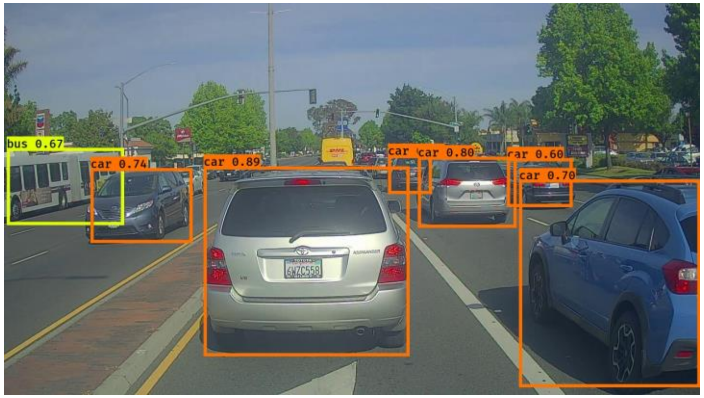


Michigan Online, Object Detection, Justin Johnson

YOLO Algorithm

- Formal Problem Setting
  - Given an image generate bounding boxes, one for each detectable object in image
  - For each bounding box, output 5 predictions: x, y, w, h, confidence. Also output class

  - x, y (coordinates for center of bounding box)
  - w,h (width and height)
  - confidence (probability bounding box has object)
  - class (classification of object in bounding box)

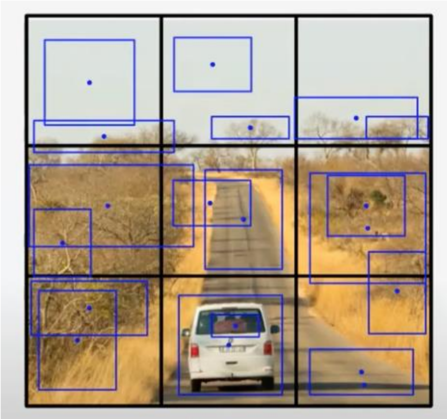


YOLO Versions

- YOLO, YOLO v2 (before 2018)
- YOLOv3 model, introduced by Redmon et al. in 2018
- YOLOv4 model, released by Bochkovskiy et al. in 2020
- YOLOv4-tiny model, research published in 2021
- YOLOR (You Only Learn One Representation) model, published in 2021
- YOLOX model, published in 2021 NanoDet-Plus model, published in 2021
- PP-YOLOE, an industrial object detector, published in 2022
- YOLOv5 model v6.1 published by Ultralytics in 2022
- YOLOv7, published in 2022
- YOLOv8. successor of YOLOv5 by the same company, 2023

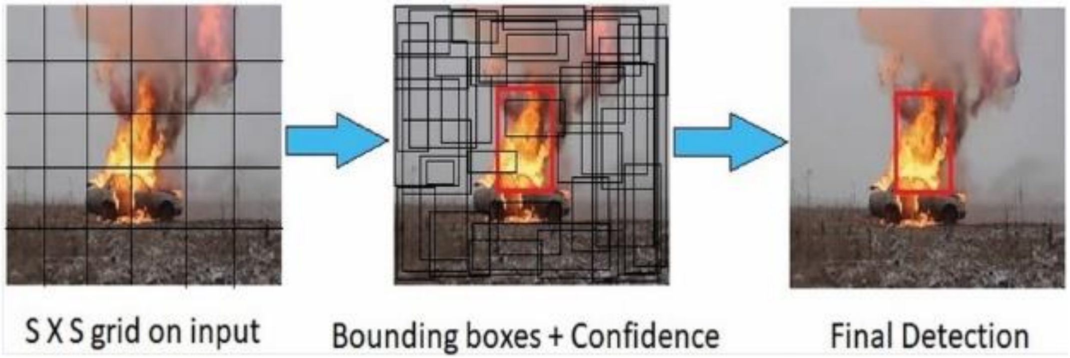
Yolo Overview

- First, image is split into a SxS grid
- For each grid square, generate B bounding boxes
- For each bounding box, there are 5 predictions: x, y, w, h, confidence



S = 3, B = 2

Shivang Sing, Robot Learning, Fall 2021



Saponara et al.

YOLO Class Probability

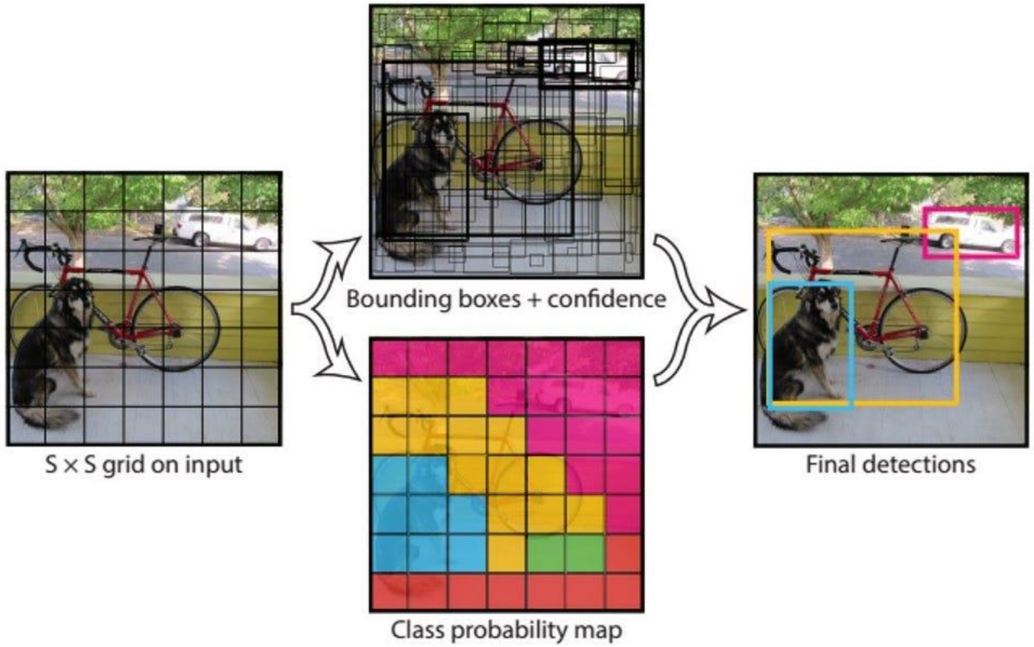
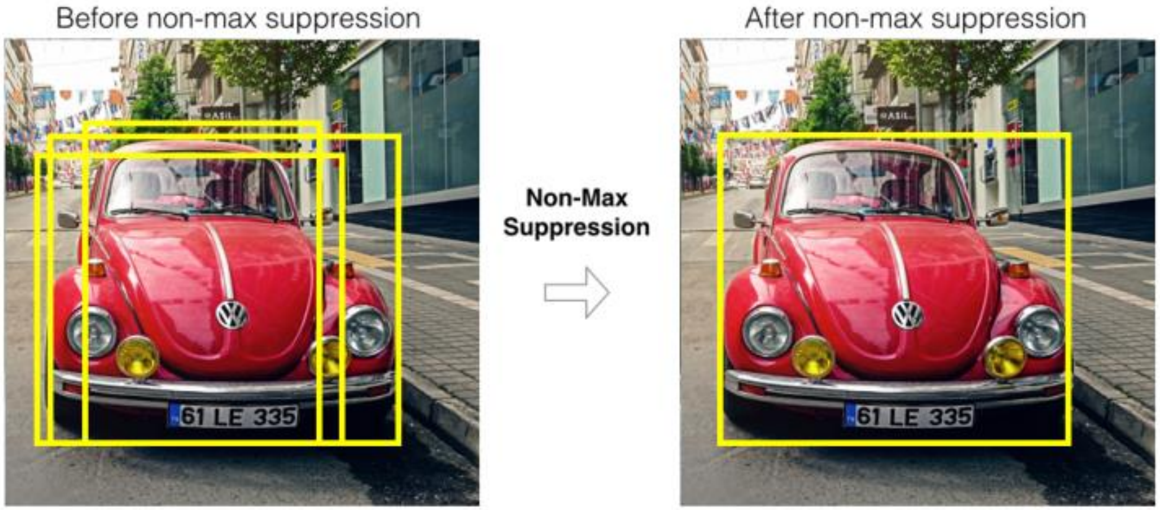


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

https://arxiv.org/abs/1506.02640

YOLO non-maximal suppression

- Most of the time objects fall in one grid, however it is still possible to get redundant boxes (rare case as object must be close to multiple grid cells for this to happen)
- Discard bounding box with high overlap (keeping the bounding box with highest confidence)



https://robocademy.com/2020/05/01/a-gentle-introduction-to-yolo-v4-for-object-detection-in-ubuntu-20-04/

YOLO Objective Function

- For YOLO, we need to minimize the following loss
- Sum squared error is used

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

Coordinate Loss: Minimize the difference between x,y,w,h pred and x,y,w,h ground truth. ONLY IF object exists in grid box and if bounding box is resp for pred

$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

Confidence Loss: Loss based on confidence ONLY IF there is object

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

No Object Loss based on confidence if there is no object

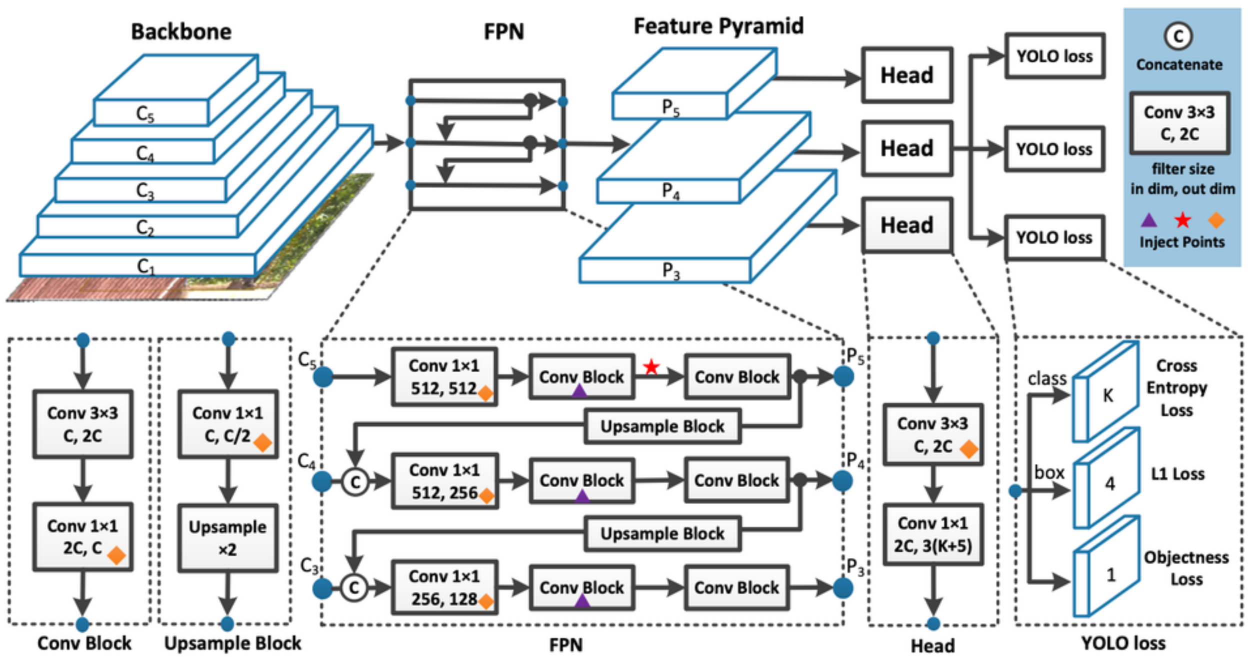
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Class loss, minimize loss between true class of object in grid box

Shivang Sing, Robot Learning, Fall 2021

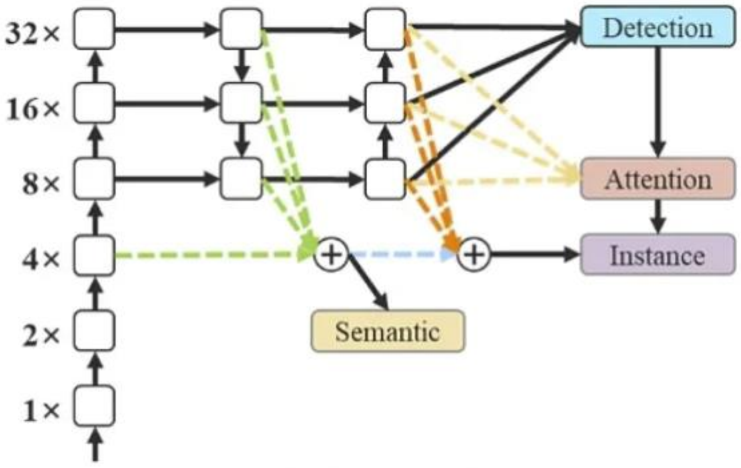
YOLO V7 architecture

The YOLO (You Only Look Once) v7 model is the latest in the family of YOLO models. YOLO models are single stage object detectors. In a YOLO model, image frames are featurized through a backbone. These features are combined and mixed in the neck, and then they are passed along to the head of the network YOLO predicts the locations and classes of objects around which bounding boxes should be drawn.

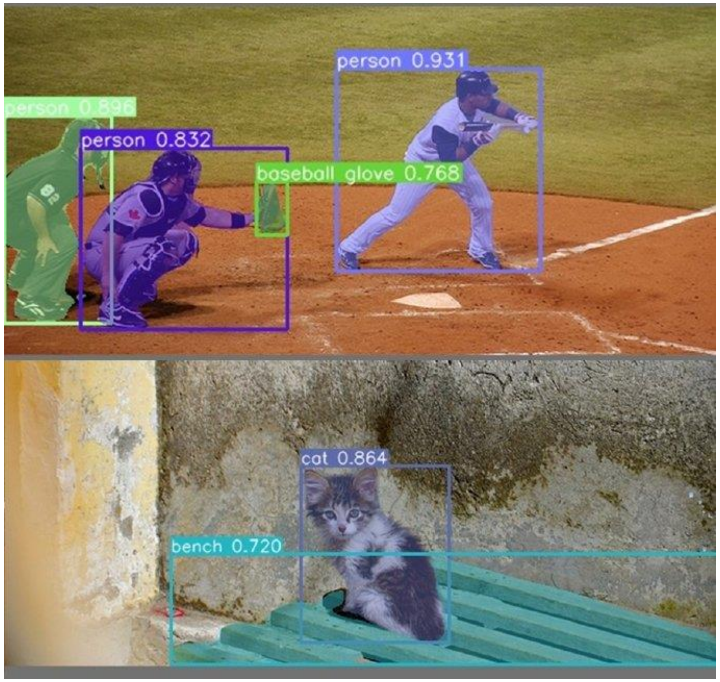


https://blog.roboflow.com/yolov7-breakdown/

YOLO v7 - mask

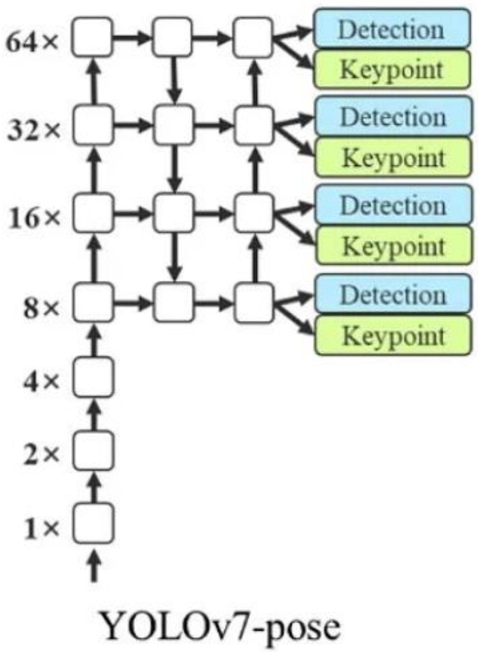


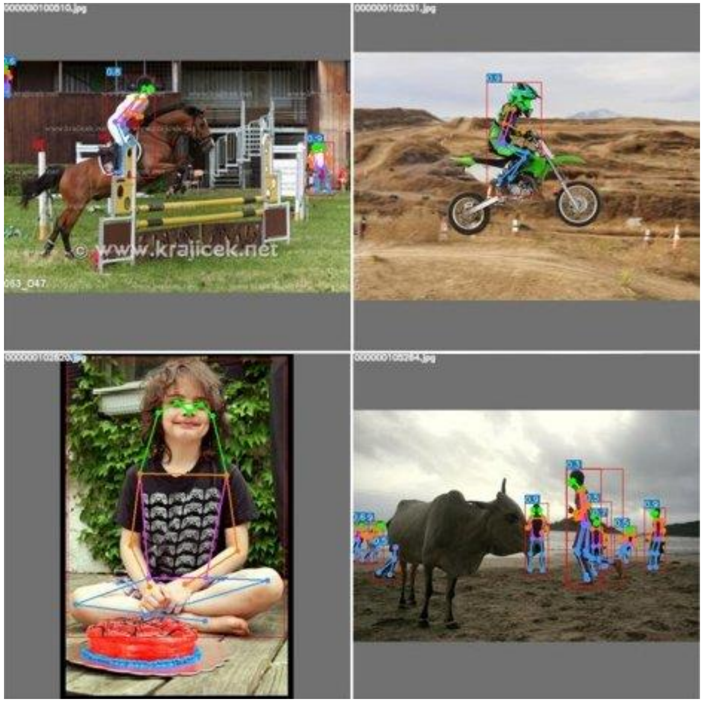
YOLOv7-mask



https://github.com/WongKinYiu/yolov7/blob/main/paper/yolov7.pdf

YOLO v7 - pose





https://github.com/WongKinYiu/yolov7/blob/main/paper/yolov7.pdf

YOLO Advantages

- The main difference between YOLO and other object detection systems is right there in its name: It only looks at an image once. When the algorithm was first introduced, it demonstrated the viability of a one-stage approach. Other methods use a two-stage process, first locating and then identifying objects.
- With its single stage, YOLO is blazingly fast and capable of processing up to 45 frames per second, depending on the hardware used. This means that videos recorded at that framerate or lower can be processed in real-time. There is also a version of YOLO capable of handling 155 frames per second – at the expense of accuracy.

https://www.linkedin.com/pulse/yolo-object-detection-its-applications-computer-vision-scanbotsdk/

YOLO Disadvantages

- The speed of the YOLO algorithm and similar one-stage models makes them especially suited for use cases like self-driving cars, where incoming objects must be processed as fast as possible.

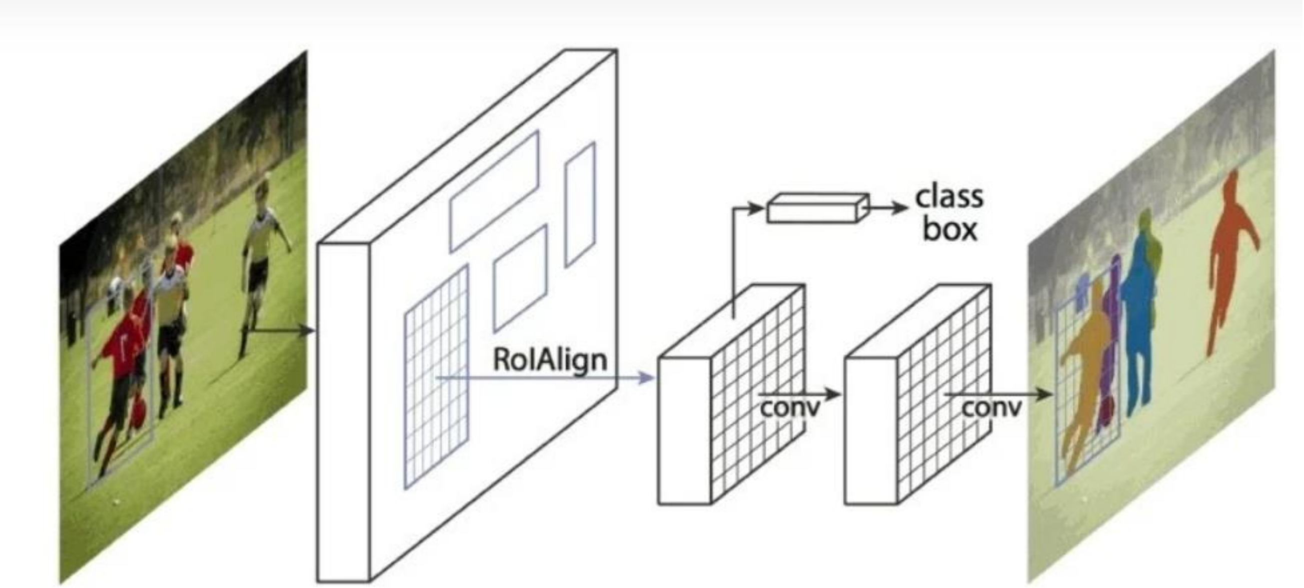
- YOLO struggled with detecting small objects and objects that are very close to each other. This is because the original YOLO algorithm could only recognize one object per grid cell, though newer versions can detect around five.

https://www.linkedin.com/pulse/yolo-object-detection-its-applications-computer-vision-scanbotsdk/

Other Examples

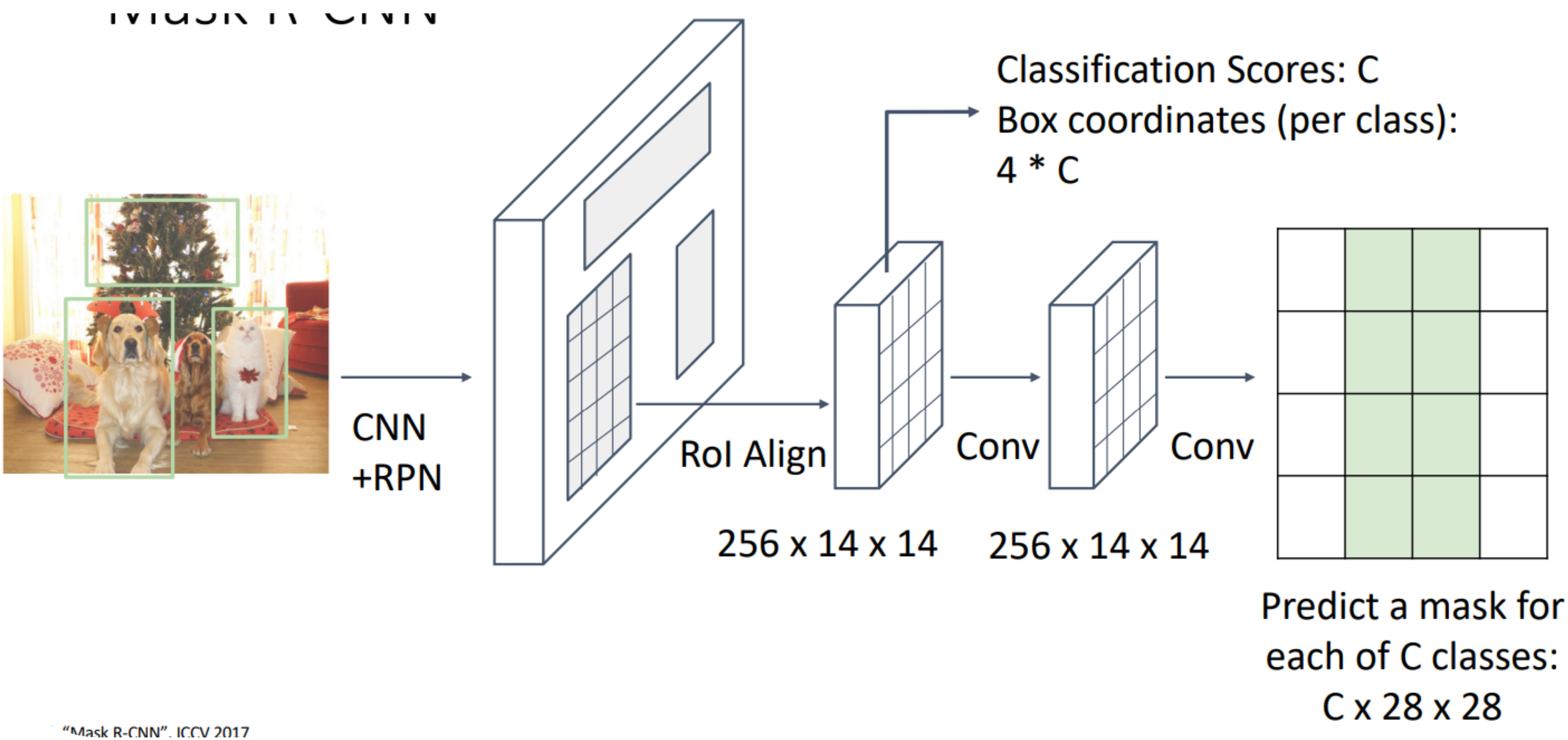
- Deep Multibox (Szegedy et. al 2014):
  - Train a CNN to find areas of interest
  - Drawbacks: Doesn’t address classification only localization
- MultiGrasp (Redmon et. al 2014)
  - Similar to YOLO
  - A much simpler task (only needs to predict object not multiple objects)

Mask R-CNN



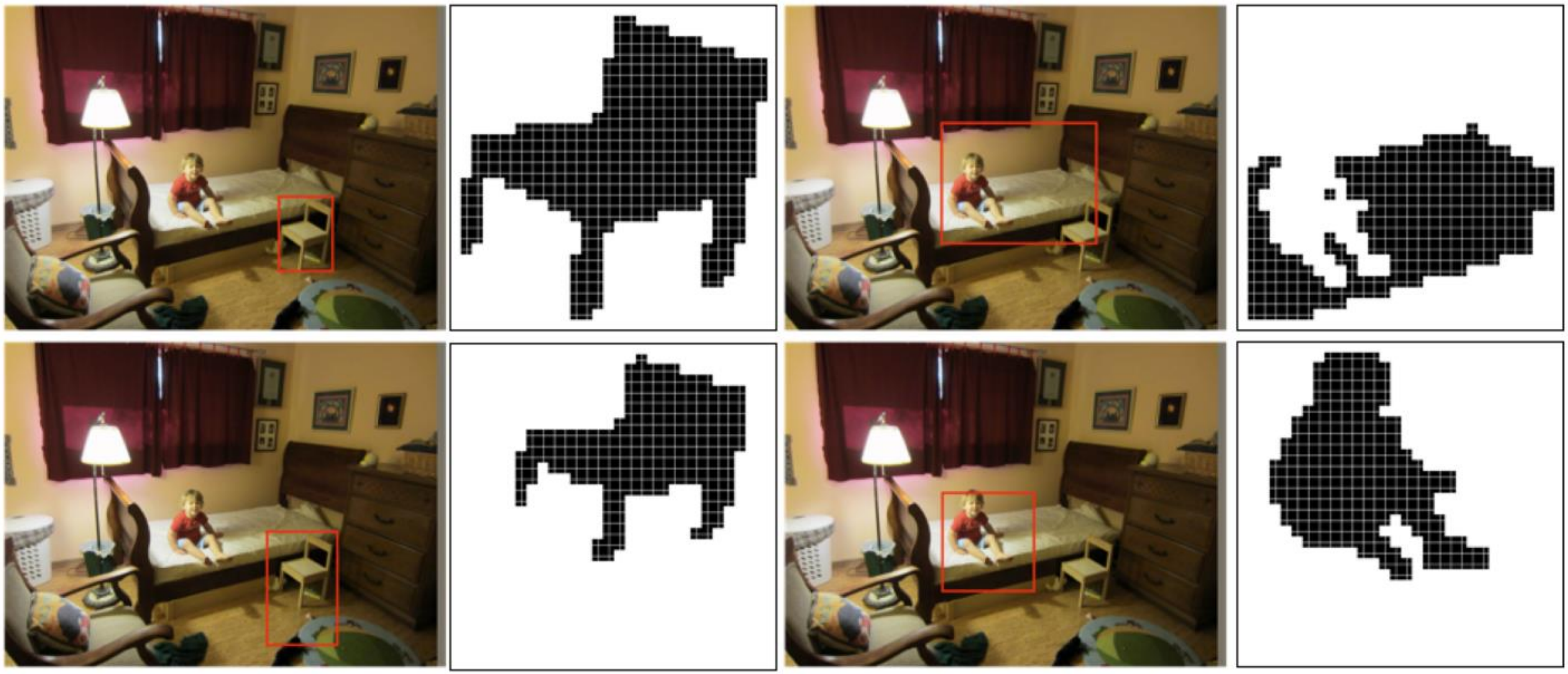
https://viso.ai/deep-learning/mask-r-cnn/

Mask R-CNN



Michigan Online, Detection and segmentation, Justin Johnson

Mask R-CNN (Trainning targets)



Michigan Online, Detection and segmentation, Justin Johnson

Some links

- Object Detection (Michigan Online)
  - https://www.youtube.com/watch?v=TB-fdISzpHQ&t=2144s
- Detection and Segmentation (Michigan Online)
  - https://www.youtube.com/watch?v=9AyMR4IhSWQ&list=PL5-TkQAfAZFbzxjBHtzdVCWE0Zbhomg7r&index=16&t=63s
- YOLO Algorithm (by Andrew Ng)
  - https://www.youtube.com/watch?v=9s_FpMpdYW8&list=PL_IHmaMAvkVxdDOBRg2CbcJBq9SY7ZUvs&index=8