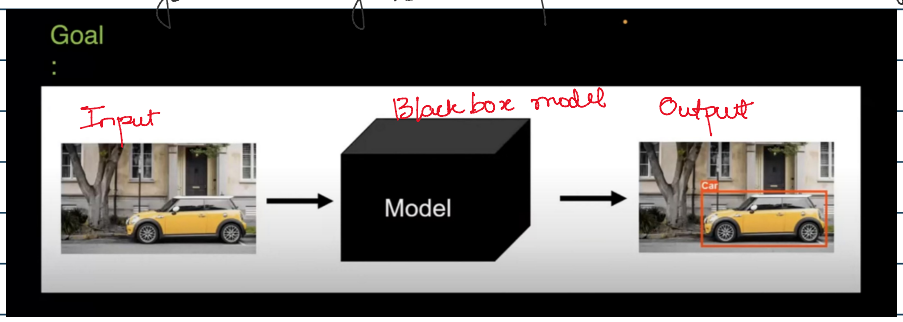
Object Detection Session-1

25 May 2024 10:55

Object Detection

Task: Given an image, I want to predicted what are the objects in the image and also found the bounding box around the object.

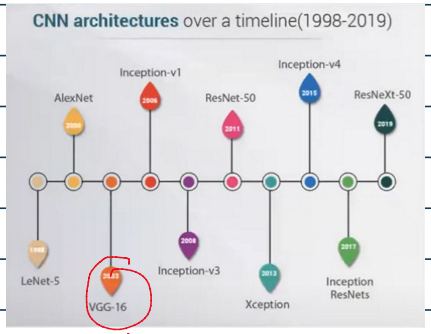


Agenda →

- Introduction
- Classification vs Localization vs Detection
- Object Detection using Sliding Window Approach
- Object detection using RCNN - Introduction
- Selective Search in RCNN for region proposals
- RCNN: Supervised Pre-training and Finetuning
- RCNN: SVM Training
- Why use SVM in R-CNN
- Bounding Box Regression Training in RCNN
- Non-Maximum Suppression | NMS in Object Detection
- RCNN Results

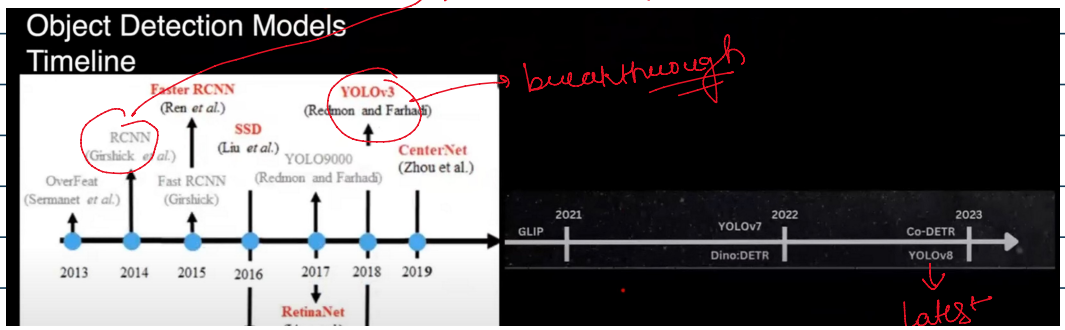
Image Classification models timeline

Image Classification architectures



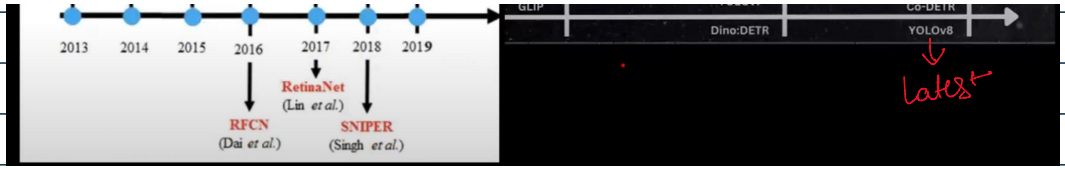
↳ breakthrough models

breakthroughs → best architecture for object detection

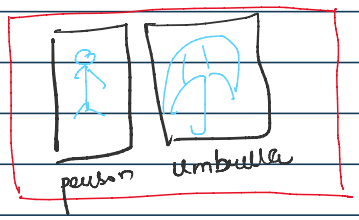


breakthrough

latest



Object detection is a part of computer vision where they not only to classify the image but also they to classify the objects inside the image and then also build a bounding box around it.



→ First let's understand the differences between classification, localization and object detection

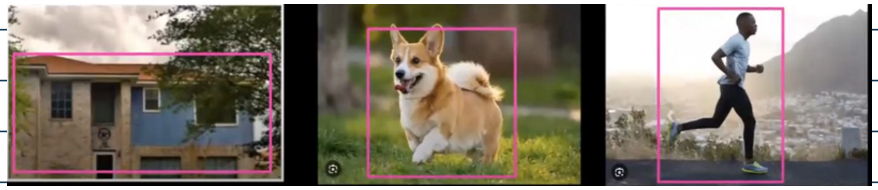
- **Classification**

  - Classify any image into a category

Home Dog Man

- **Localization / Object Localization:**

  - ~~Classification with Single label~~ + object Localization



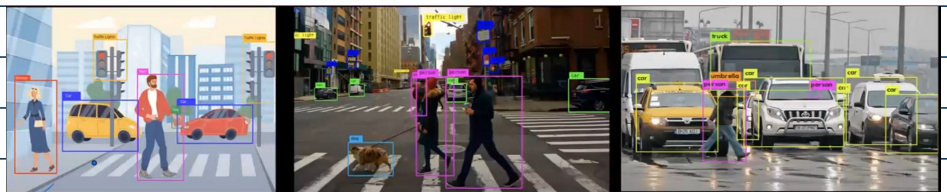
House (We can also have trees but we are concerned about houses)

Dog

Person

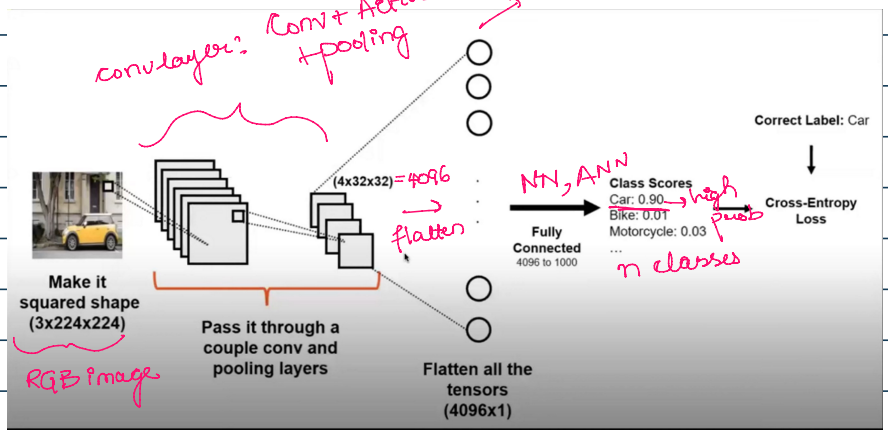
- **Object Detection**

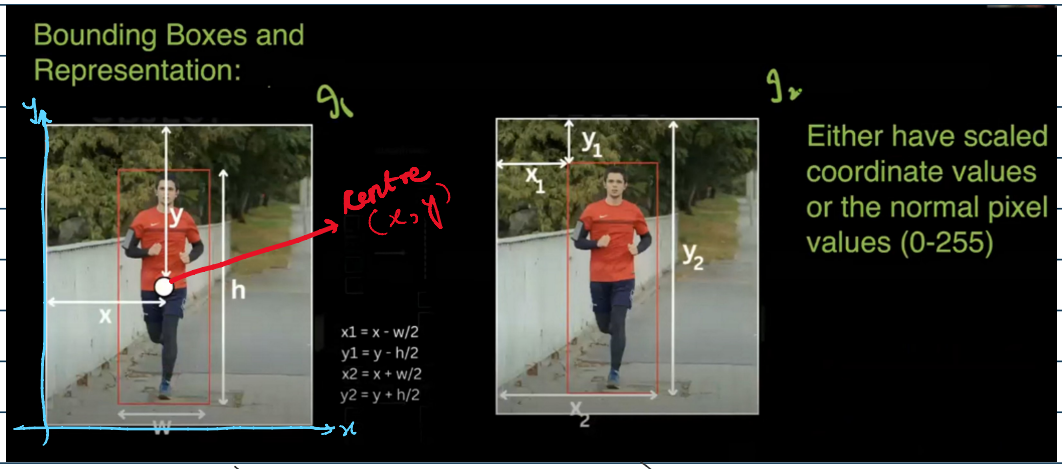
  - Requires us to detect all objects and all classes and all the bounding box in a given image irrespective of the shape and size of the object

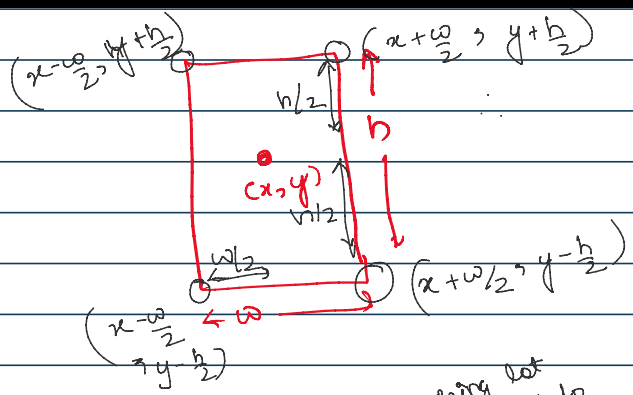


Before going deep into Object detection let's understand How classification works.

⇒ Working of Classification

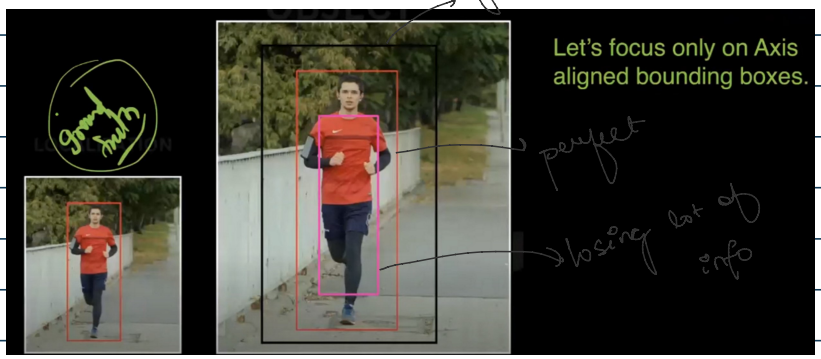




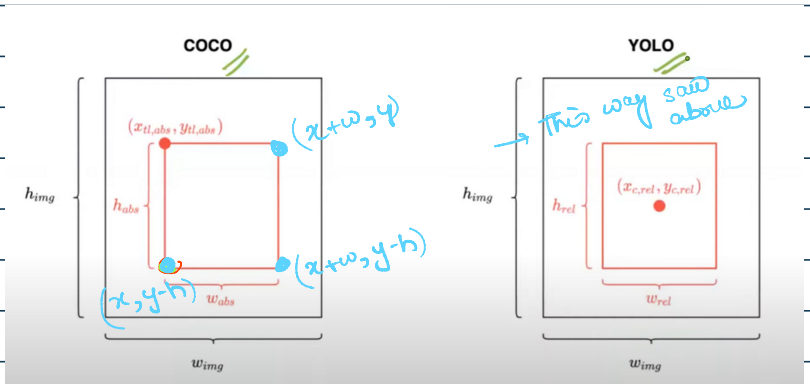


gaining lot info

For a given image we can have multiple bounding

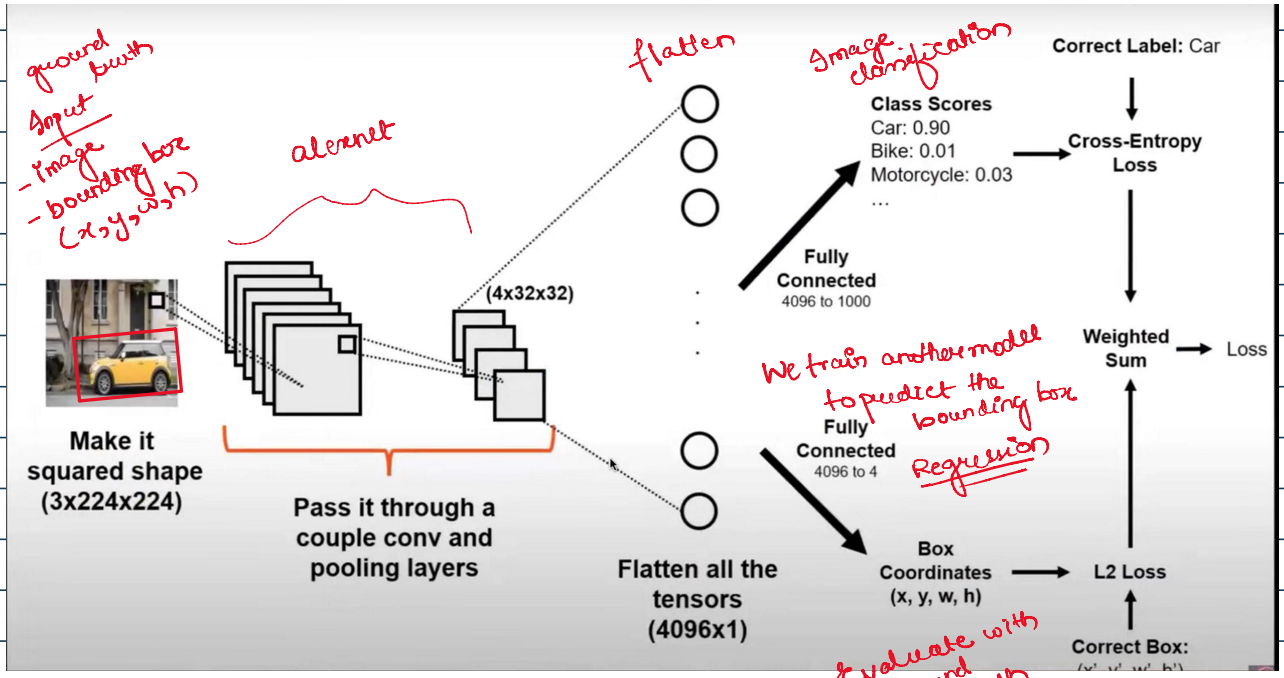


For a given image we can have multiple bounding box



- We can obtain all 4 coordinates if we have information
- COCO ⇒ one coordinate + height & width
- YOLO → middle point + height & width

Modified architecture

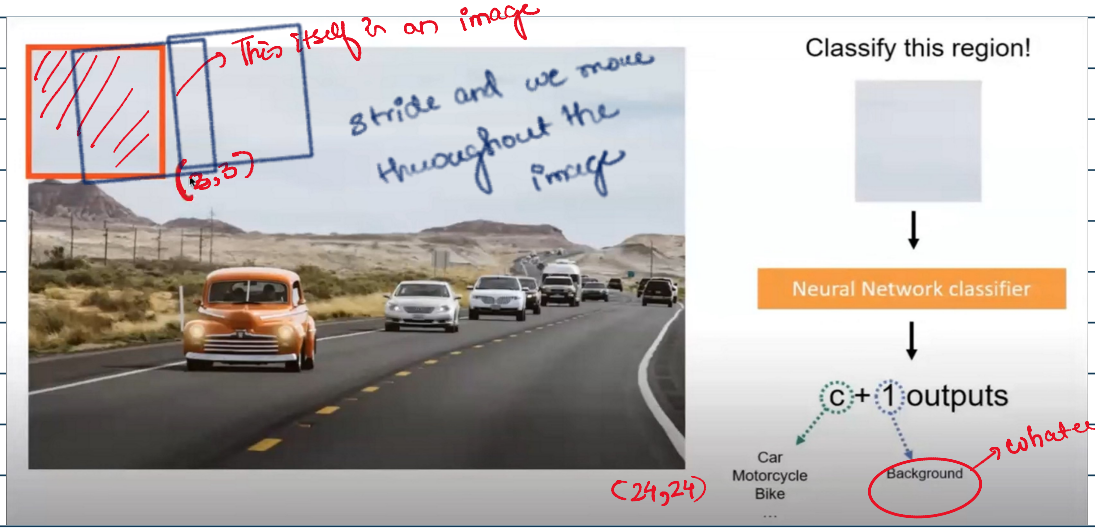


Disadvantage: This architecture will only work in case of one object in the image

In the image

If we have an image with many objects, it will be impossible to pass the BB values for each object.

Solution: SLIDING WINDOW



Problem

1. Each object is different
2. Shape is different
3. Box width and height varies
4. Possible combination of boxes are in millions
5. And we have multiple objects in an image

Computation cost

$$\sum_{h=1}^{H}\sum_{w=1}^{W}(W-w+1)(H-h+1)$$

$$= \frac{H(H+1)}{2} \cdot \frac{W(W+1)}{2}$$

= in millions

⇒ We can reduce the number of boxes by taking stride greater than 1. But even then, the computations will be large.

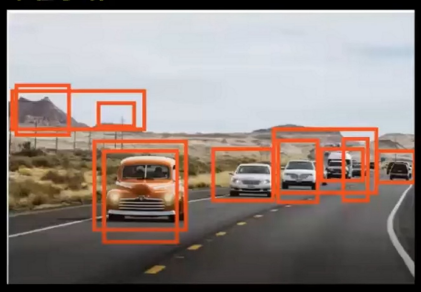
Solution: RCNN to decrease the number of bounding boxes and speed up training

R-CNN ⇒ Region CNNs.

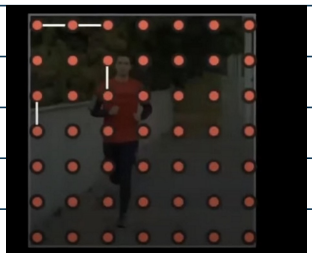
- Idea of RCNN is to decrease the region of proposals.
- Some of the cropped images are observed have no objects, then why pass them into model
- Then how can be segregate the crops with images and no images (background)
  - ↳ Selective Search (External Algorithm)

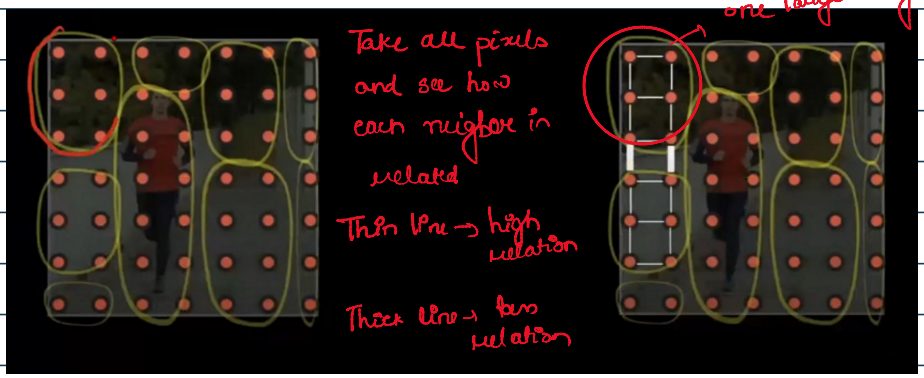
2. Object detection with R-CNN

Our object detection system consists of three modules. The first generates category-independent region proposals. These proposals define the set of candidate detections available to our detector. The second module is a large convolutional neural network that extracts a fixed-length feature vector from each region. The third module is a set of class-specific linear SVMs. In this section, we present our design decisions for each module, describe their test-time usage, detail how their parameters are learned, and show detection results on PASCAL VOC 2010-12 and on ILSVRC2013.

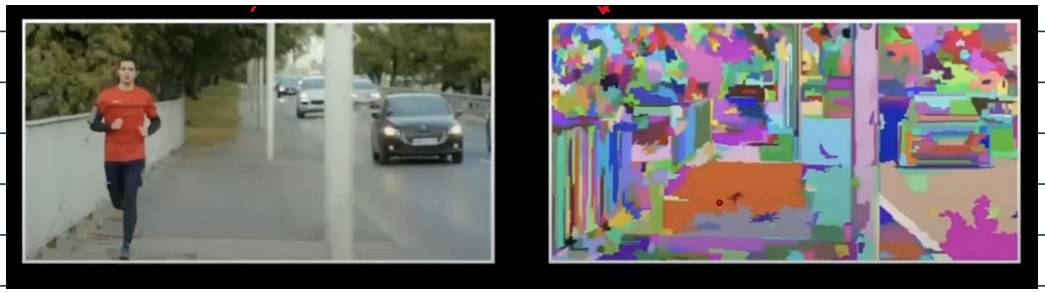


1. Section 1
   - Segments image into regions
   - Merge similar regions to create larger regions





In reality millions of pixels of the image are considered to figure out this

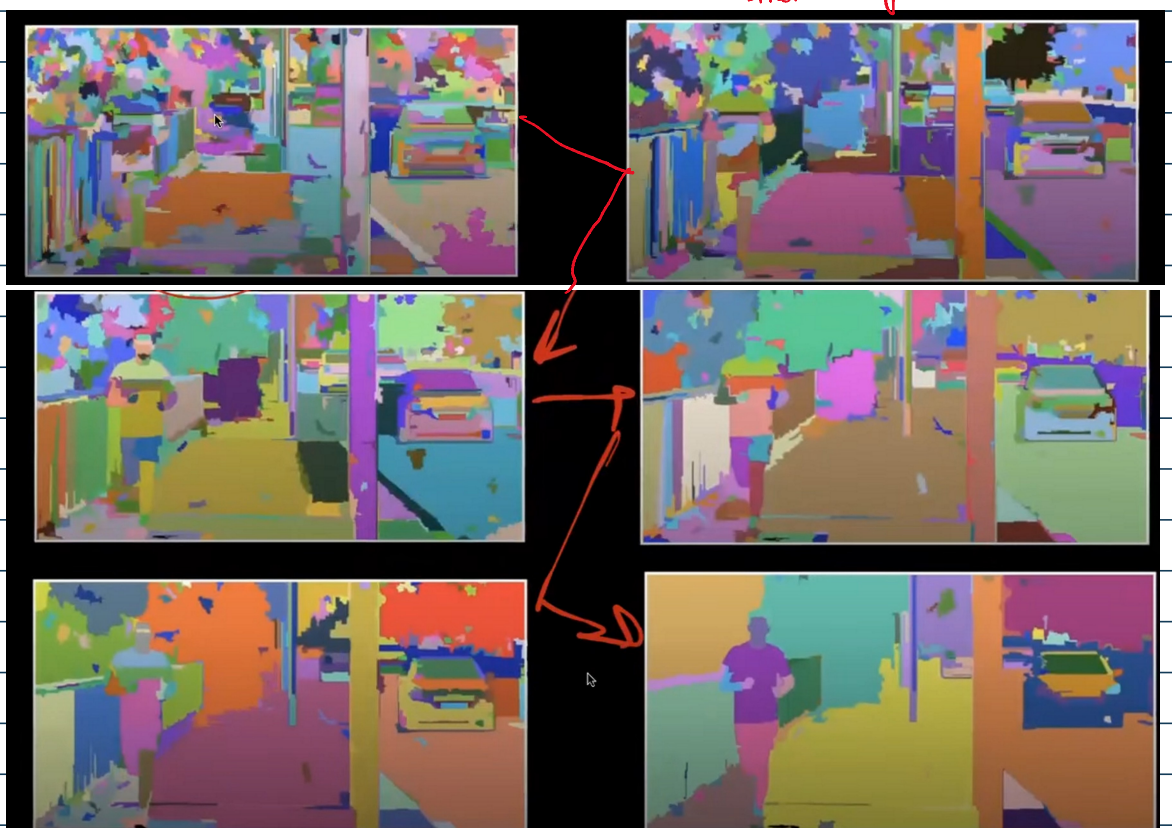


Next step is to merge similar regions to make larger regions

But how to merge?

1. Color Similarity
2. Texture "
3. Size "
4. Shape "
5. Linear combination of above measures

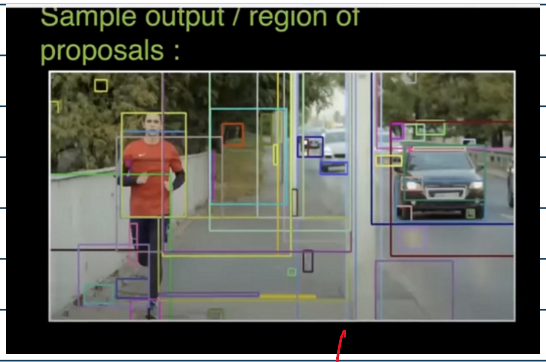
Initial segmentation



final output

* Selective Search generates regions not class label
  - ↳ It does not know what exactly are the objects
  - ↳ but it can understand the pixels and figure out the best regions for a probable object

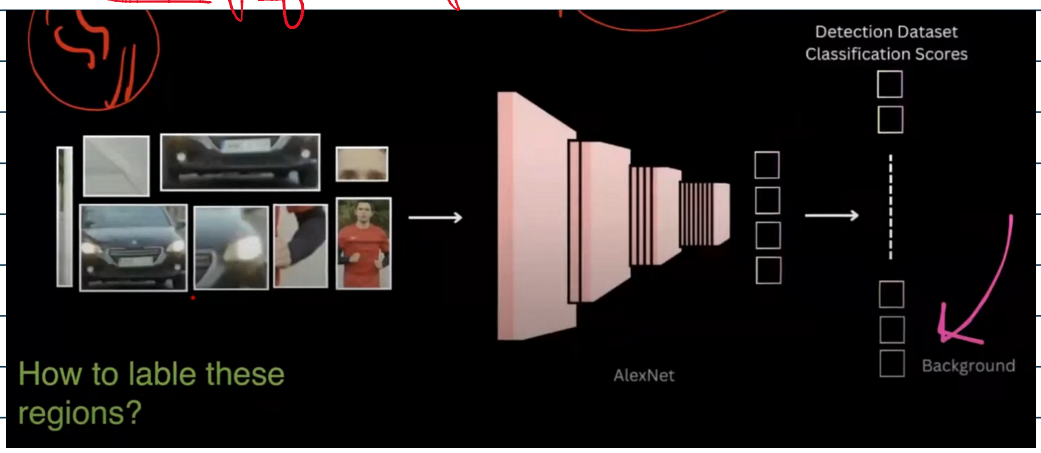
↳ but it can understand the pixels and figure out the best region for a probable object



↓ 2,000 boxes

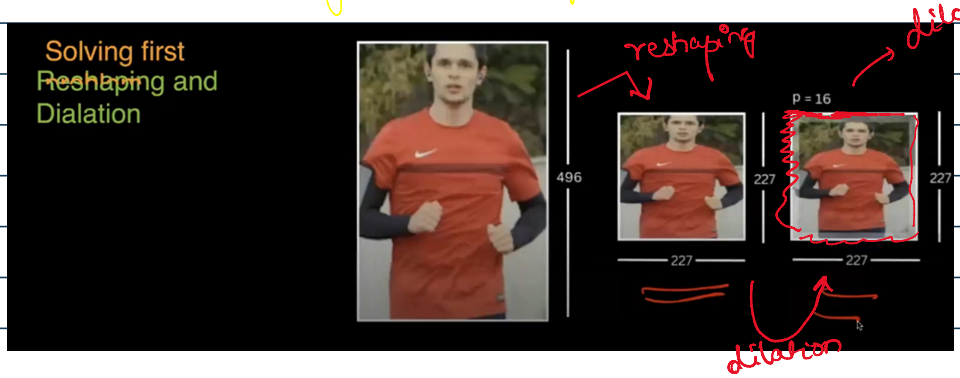
- ~~The final output after selective search~~
- It tells us these are the probable bounding boxes
- Usually for every image we generate 2000 region proposals.

Labelling of the Images



One problem maybe to pass images areas of varying height x width

That is also solved using Image reshaping and Dilation

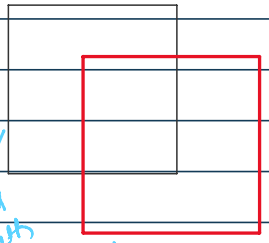


dilation mean padding extra information to maintain information

p=16 → hyperparam

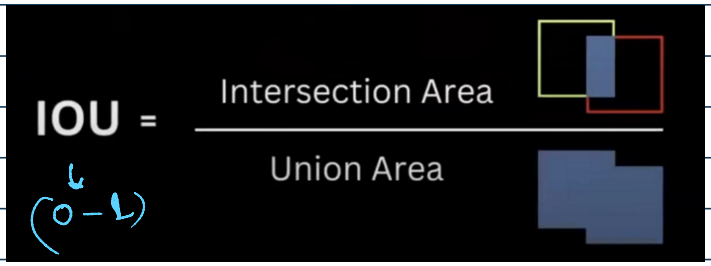
Now how to label the images that are reshaped?

IOU score



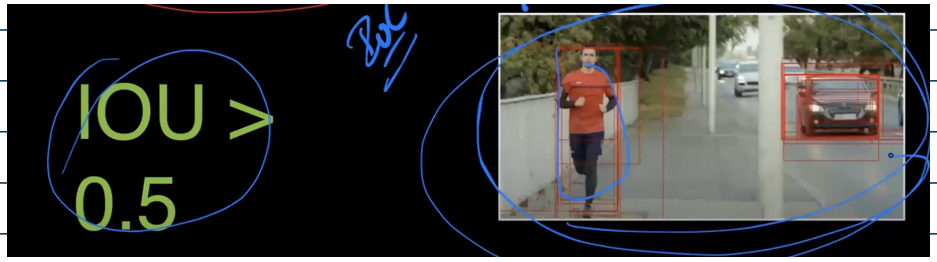
ground truth

Select search output



0 → no overlap

1 → both box are same



- Remaining boxes with IOU < 0.5 are considered as background class

Then for training

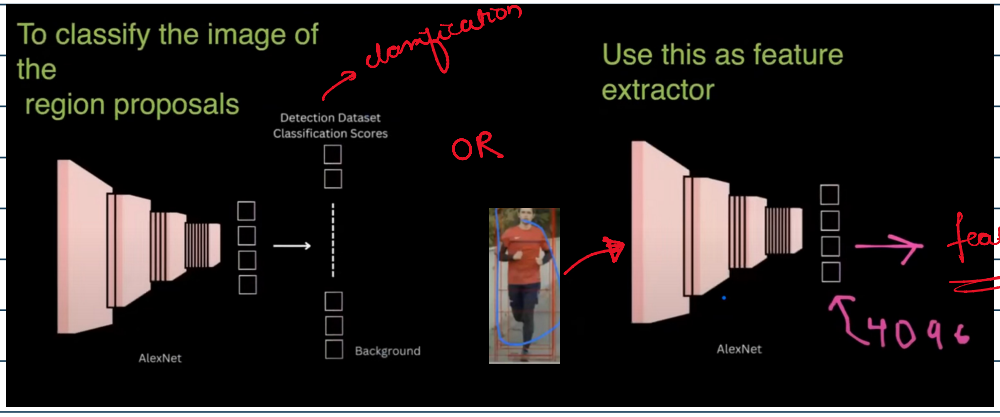
- Randomly sample some background proposals
- Get all region proposals
- Combine them as batch and pass to the model
- All will be resized to 227 * 227 images
- Fine-tune the model

What if a proposal box overlaps with 2 ground truth values?

After we have the bounding boxes, we can pass it to the alexnet for classifying the labels

or

use alexnet/CNN for feature extraction



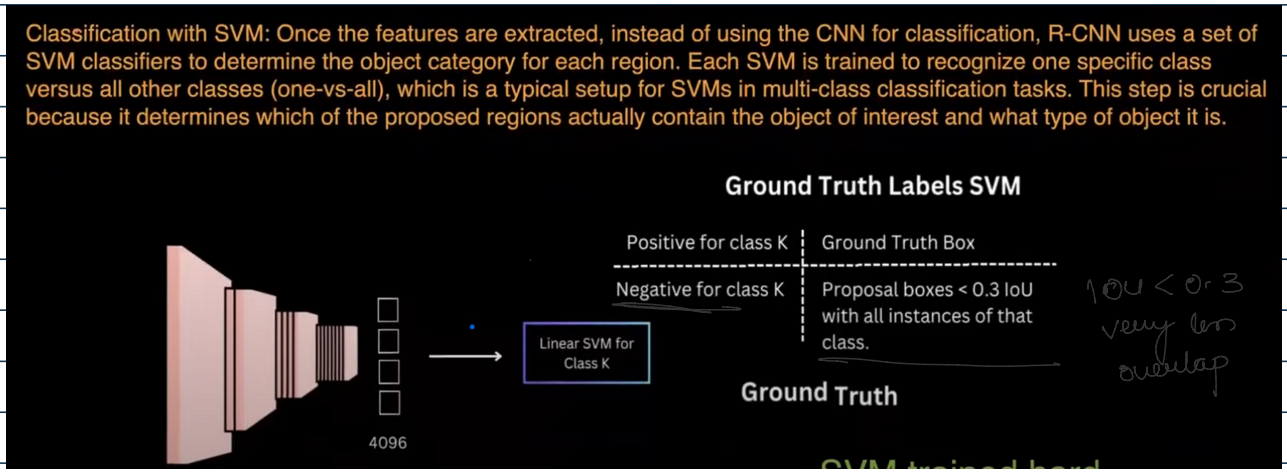
feature → SVM → person or not

So for each class label we have a SVM that tells whether it is that object or not.

Why SVM if we have CNN?

→ During experimentation they first used CNN → 40% accuracy

↳ Tried with SVM → 48% accuracy ↳ better learning



IOU < 0.3 very less overlap

↘ used to train SVM

↘ used for fine-tuning

positive samples

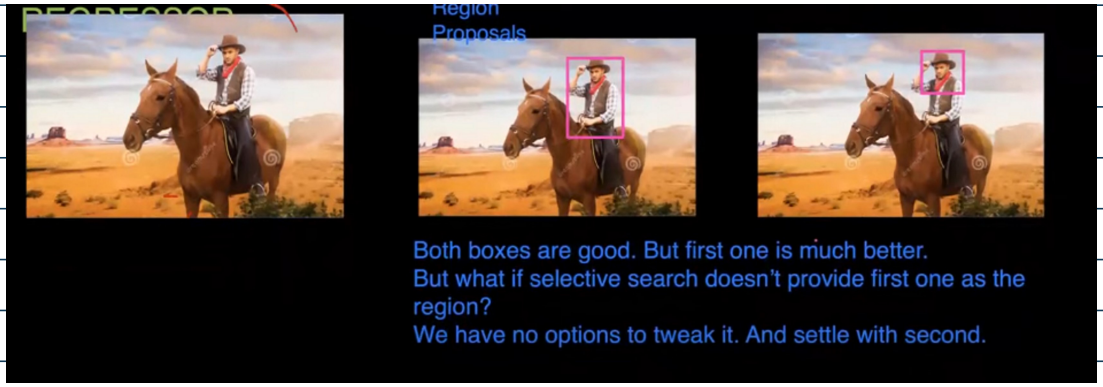
positive sample

-ve sample

STEPS :

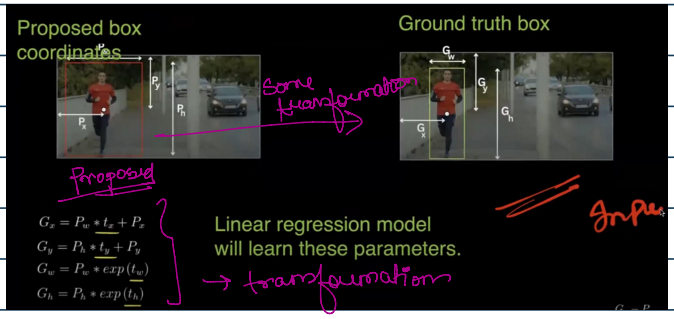
1. Train CNN on large classification dataset
2. Fine tune CNN with resized proposals on classes of detection dataset and background class
   - Proposals with label as Class K - Proposals with IOU ≥ 0.5 with GT boxes of that class
   - Proposals with label as Class Background - All Remaining Proposals
3. Train a binary classifier (SVM) for each class on the fc layer representation of proposals
   - Positive Labelled Proposals for Class K - GT Boxes for that class
   - Negative Labelled Proposals for Class K - Proposals with ≤0.3 IOU with ALL GT Boxes of Class K

Now suppose we have 2 bb and both are correct then how to choose



Or suppose the selective search algorithm gave a proposal region, but it is not close to ground truth.

To deal with it we perform some transformation to predicted box



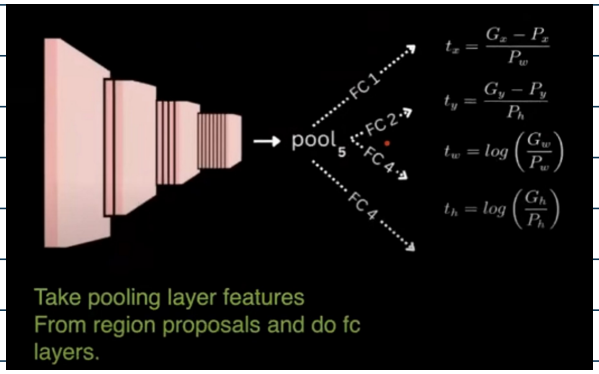
G_x, G_y, G_w, G_h → ground truth

P_x, P_y, P_w, P_h → proposals

t_x, t_y, t_w and t_h are parameters

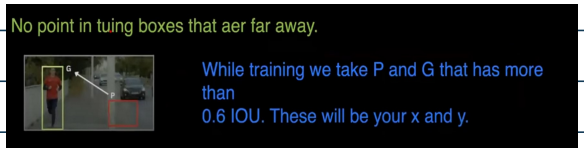
So researchers built a ML regression model to learn these parameters.

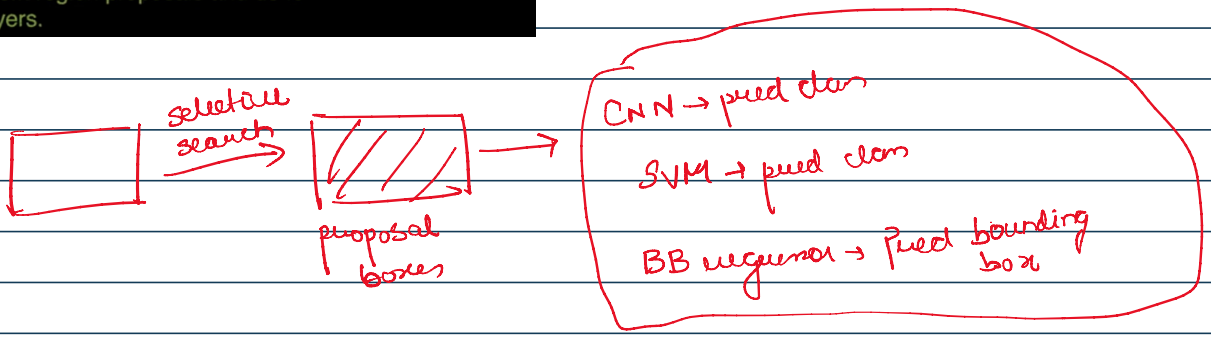
So researchers built a ML regression model to learn these parameters.



This is called

Bounding Box Regressor

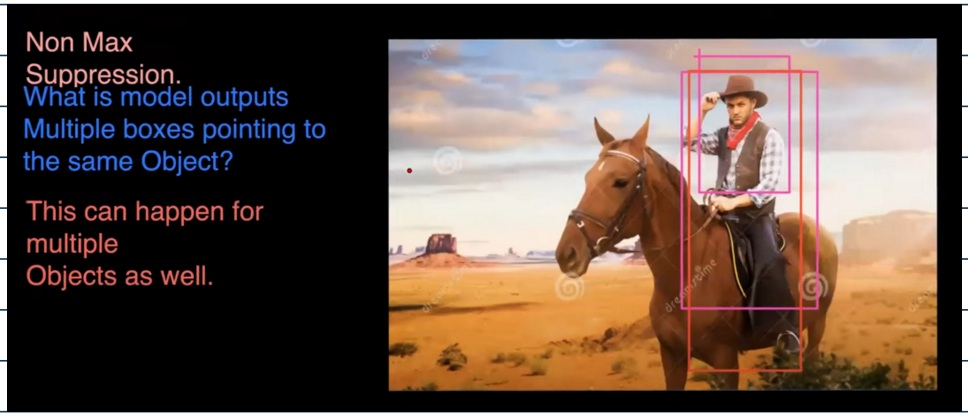




R-CNN

Training Done !!

Now we have the model trained.
We pass an image and get multiple boxes.

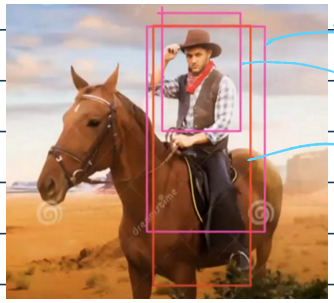


When to keep and which to discard?

Non-maximum Suppression

NMS is applied to each object sperately.
It utilises IOU score.
Frist get the all the predicated boxes and their score of objectiveness.
Sort them based on the object ness score.
Get the top one and then Calulctae IOU with others. And remove less IOU boxes.
NMS can be class agnostic, we shall see this in later architectures.

Image →



$B_3 = 0.8$
$B_2 = 0.7$
$B_1 = 0.9$
highest = 0.9

→ Calculate IOU with ground truth

↓

Keep box having high probability + high IOU

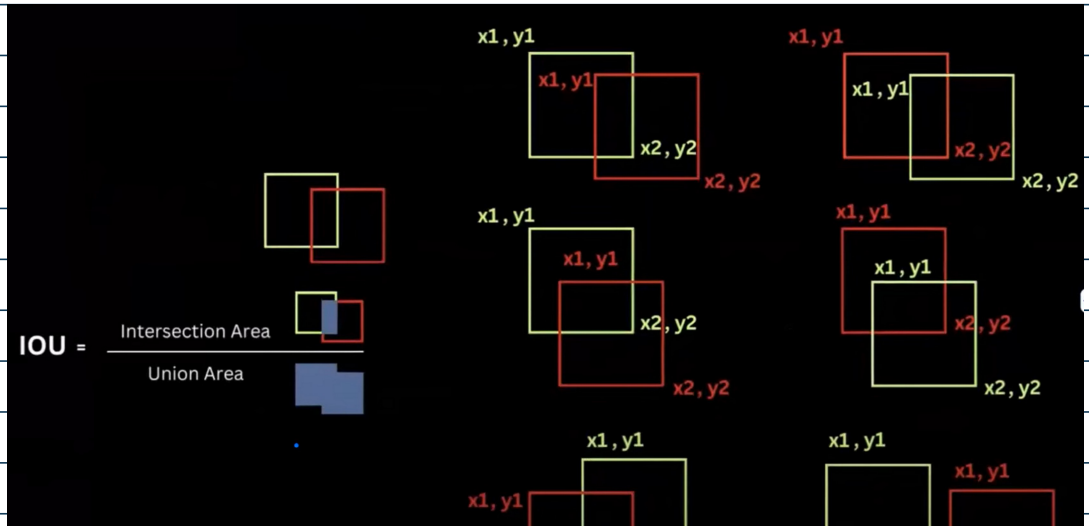
- Extract Region Proposals using Selective Search
- Train VGG on ImageNet classification dataset
- Fine tune CNN with resized proposals on classes of detection dataset and background class
- Train a binary classifier(SVM) for each class on the fc layer representation of proposals
- Train a class specific bounding box regressor on top of proposal features
- Filter predictions using NMS

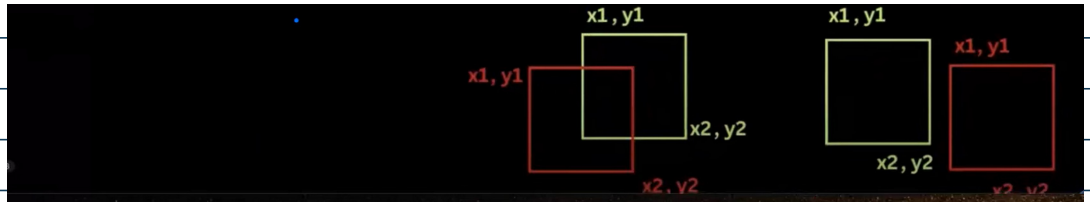
R-CNN flow

How to Evaluate?

- What metrics to use?

1.





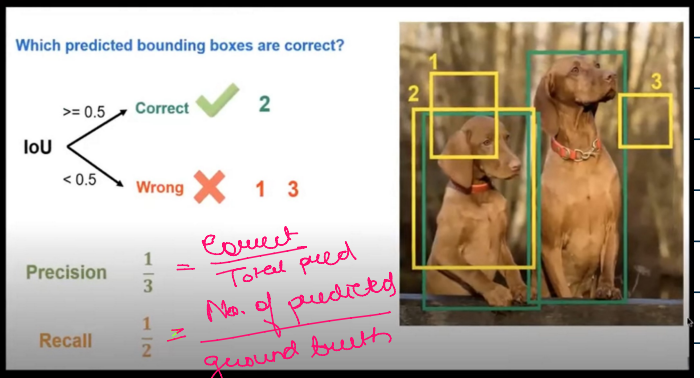
- MAP; Mean Average Precision

It is one of the benchmark metric for object detection

First let's understand precision and recall in case of bounding boxes.

1. Ground truth boxes
2. Precision boxes.

TP, FP, FN

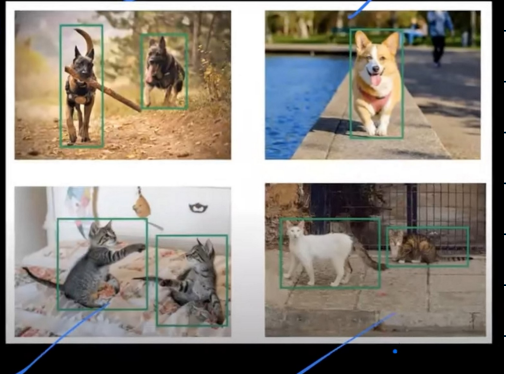


Precision = Correct / Total pred

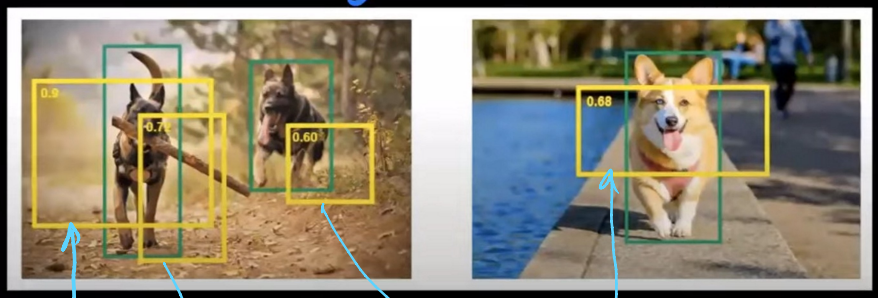
Recall = No. of predicted / ground truth

Calculating MAP in detail

Image only 2 objects cats and dogs. We have ground truth like show cased.

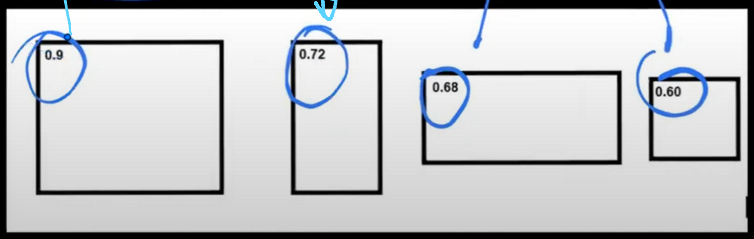


Map is calculated per class



Model makes predictions
Each give probability score of being dog.

1. sort bb boxes based on probability score.



Need to draw precision recall curve

Considering IOU > 0.5

Now for each predicted box calculate precision recall value

Predicted boxes = 4 Ground truth = 3

First take highest probability prediction box, 0.92 and find its IOU.

If greater than 0.5 : Correct prediction

∴ We checked 1 box and got correct : Precision = $\frac{1}{1}$ → 100%

We predicted 1 correctly out of 3 : Recall = $\frac{1}{3}$ = 0.33

∴ (P, R) = (1, 0.33)

- Now remove b1 from predicted and the ground truth

Move to next highest predicted box : 0.72

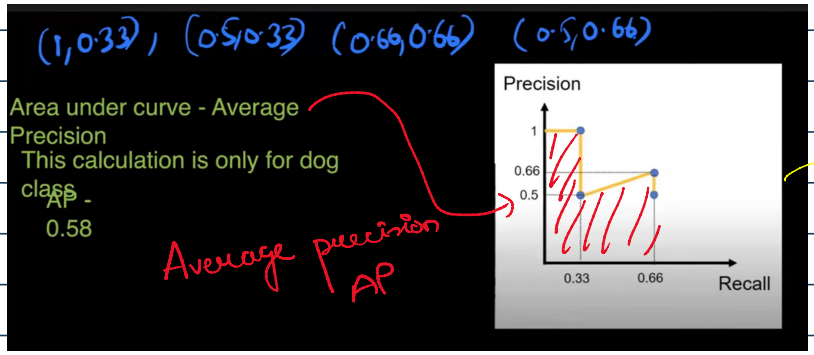
But for the previous ground truth, it is zero as we removed it

- Precision = 1/2 = 0.5 (out of 2 predicted box only 1 is correct)

Recall = 1/3 = 0.33

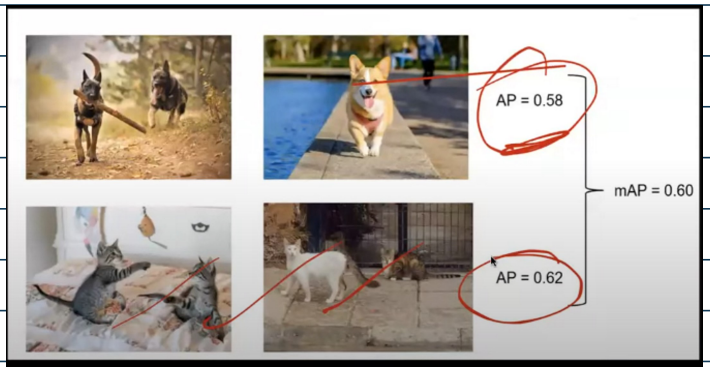
- (P, R) = (0.5, 0.33)

Similarly we repeat for all the predicted boxes and get their (Precision, Recall) values



→ Precision recall curve

This is only for dog class.

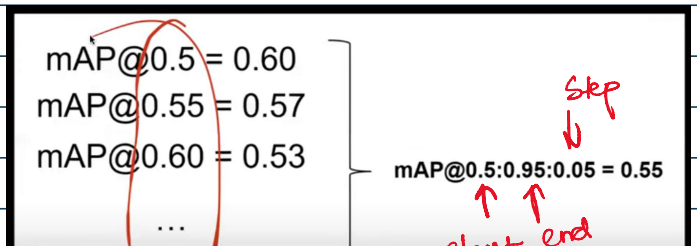


mAP@0.5 = 0.60

↑

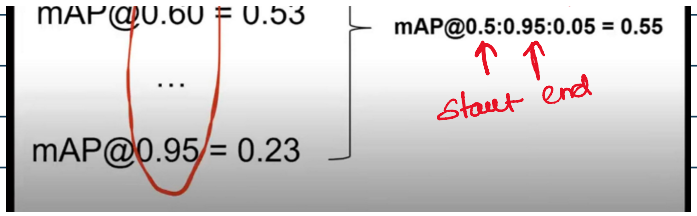
This is the IOU threshold for this task

We can try for different IOU values.



- Calculate mAP for different IOU's
- Take mean to get final mAP

higher than mAP



higher the mAP
→ better the model

Evaluated