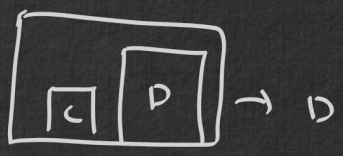
Object Detection

Classification → drawback → it will not tell the classes of all the objects present in an image.



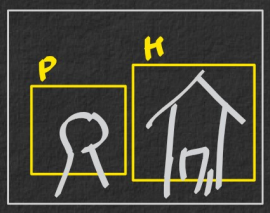
It will give preference to dominant object.

{ Experiment with Teachable Machine }

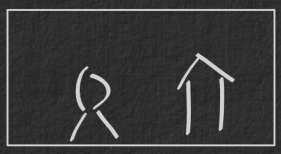
Localization → Find out position of objects in an image

Object detection: Find / localize different objects in an image & classify them individually.

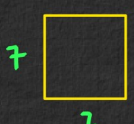
{ Depends on no. of classes it was trained upon }

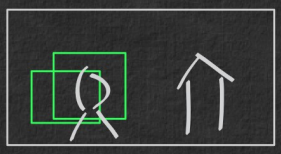


Naive approach of building an object detection model:-



→ Identify edges





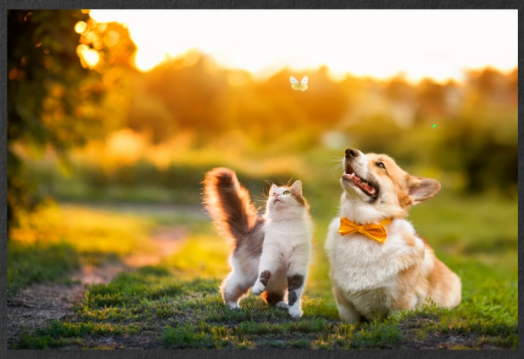
Sliding window approach

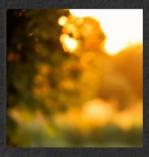
→ variable box size

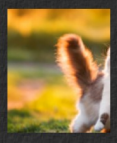
→ computation cost

→ Slowest approach

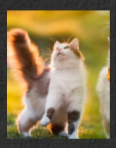
→ overlapping issue



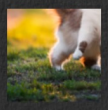


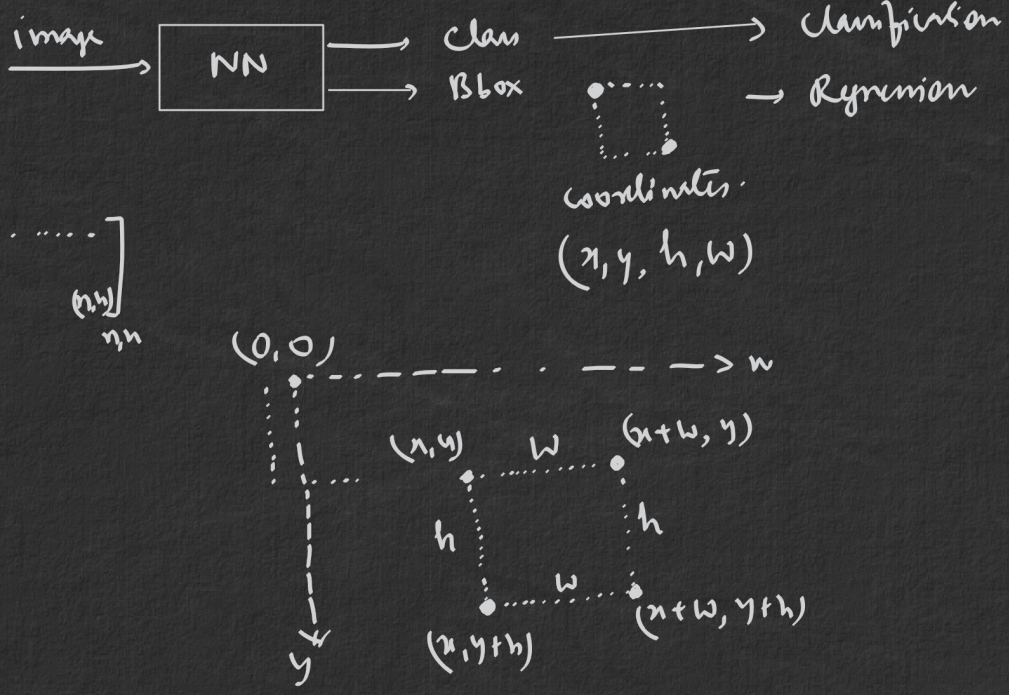


Cat
60%



Cat
90%





Object- detection models: -

- YOLO ✓ V1 ..... V7.
- SSD ✓
- RCNN

Detection

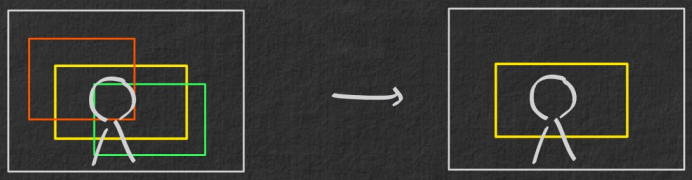
Common components available in almost all object-detection model/architecture:

i) Region proposal: DL model which will locate the region of interest (ROI)

ii) Feature extraction & Network prediction.

iii) Non-maximum suppression {NMS}

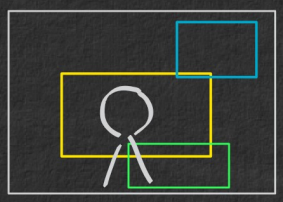
→ solves the issue of overlapping Bbox



i) Region-proposals:-

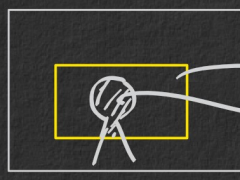
ROIs → Region the system believes have a high likelyhood of containing an object.

determined by object score



$ROI_1 >> ROI_2 >> ROI_3$

Object score → Object / background



ii) Feature extraction & network prediction.

Based on Region proposal.

A → Bounding box pred.

B → class prediction

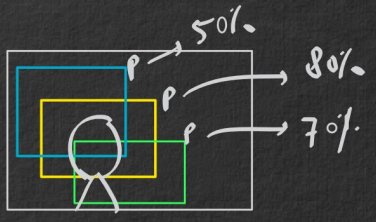
iii) NMS.

→ A technique that makes sure yours algo. detects each object only once.

It finds out box with maximum prediction probability and eliminates other boxes.

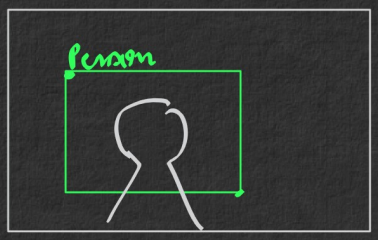
Algo :-

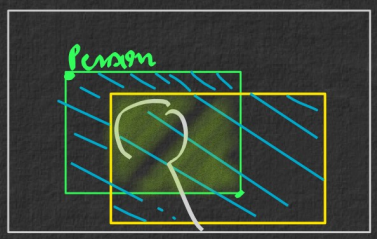
i) Discard all the Bboxes with threshold < confidence threshold



ii) Uses Intersection Over Union {IoU} to discard Bbox which has low IoU score.

Training

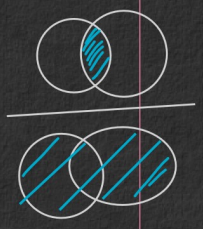




Ground Truth

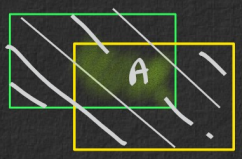
Person , {x, y, w, h}

$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$

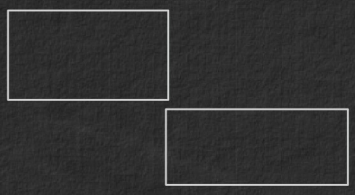


perfect Overlap

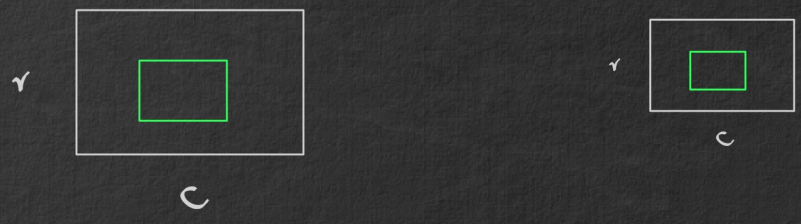
$$IoU \uparrow = \frac{A}{B} = 1$$



$$IoU = \frac{A}{B} \quad A < B \quad IoU < 1$$



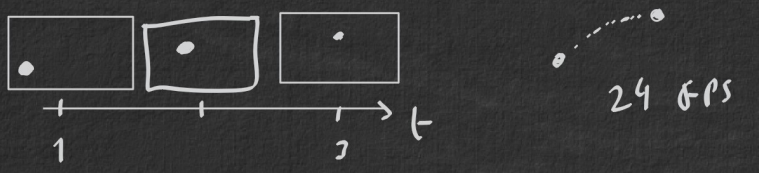
$$IoU = \frac{0}{B} = 0$$

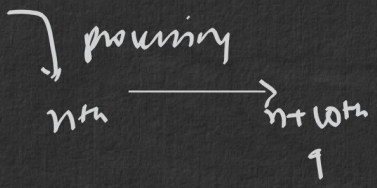


FPS → frame per second

How many frames in a video your algorithm is able to process every x second

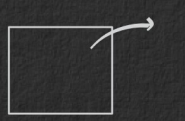
Faster RCNN → 7FPS
SSD → 59FPS
} Depends on Hardware

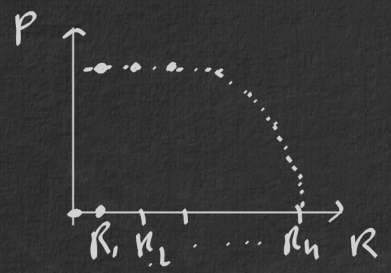




1> mAP → mean Average Precision

n no. of classes



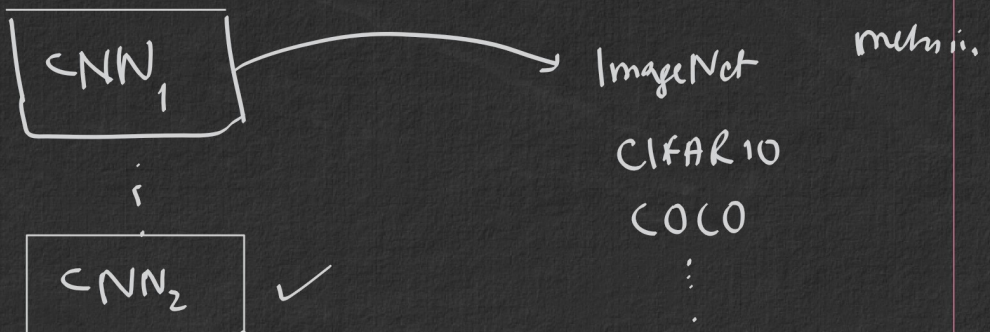


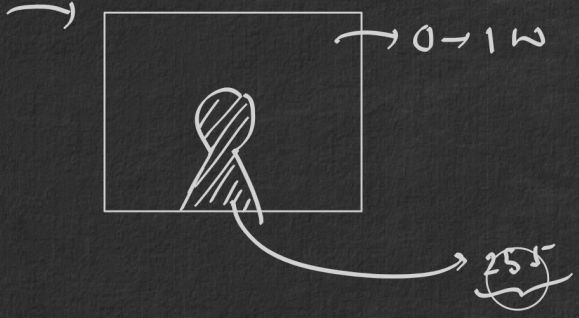
P_1 P_2 .... P_n on $R_1 \dots R_n$ class A ← present / absent

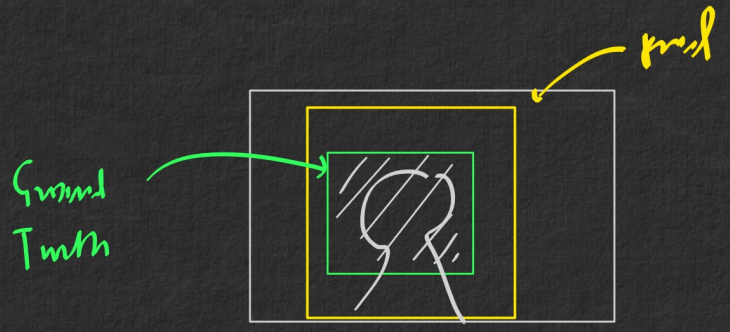
Average → Average Precision → one class class 1

	1	AP_1
	2	AP_2
	3	
	⋮	
class	n	AP_n

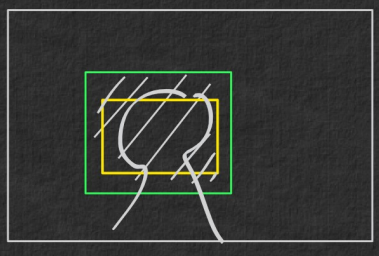
$$\text{mean}\ (AP_1 \cdots AP_n) \Rightarrow mAP$$



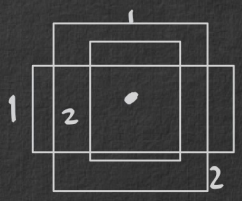




$$= \frac{A}{B} \qquad A \subset B \qquad IoU < 1$$

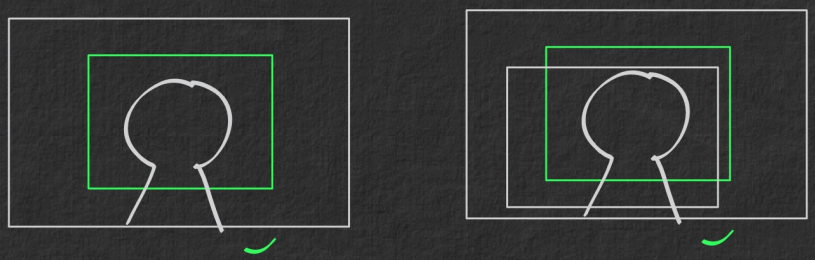


$$= \frac{A}{B} \qquad A > \square \qquad IoU > 1$$

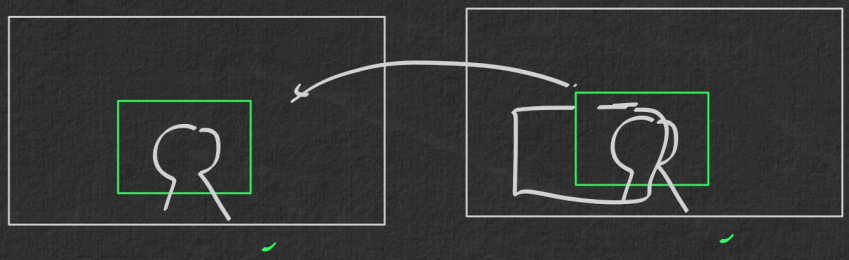


Tuning

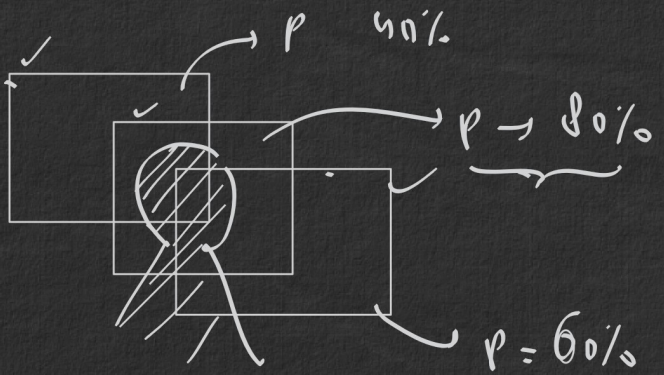
d 1

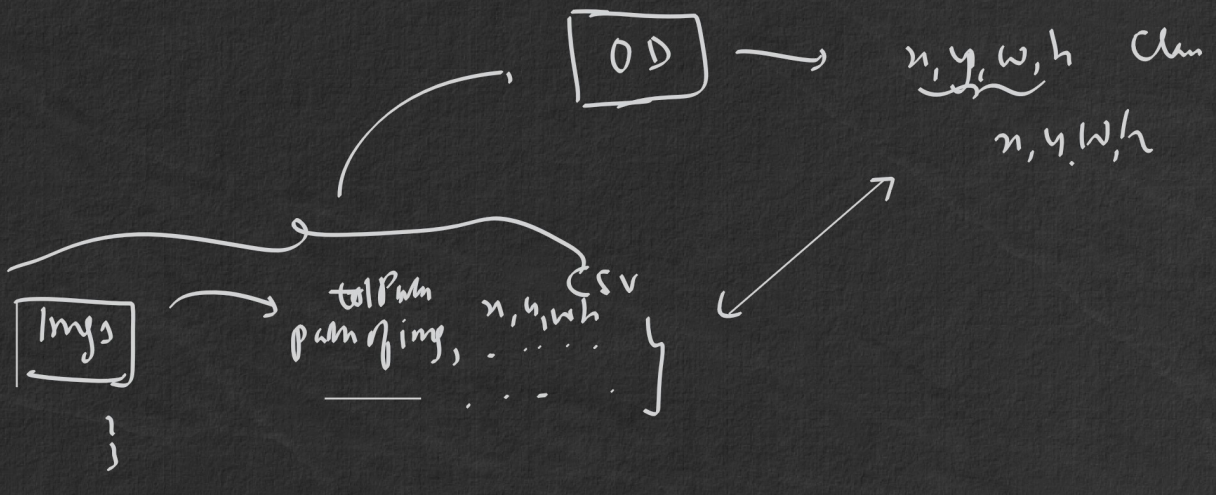


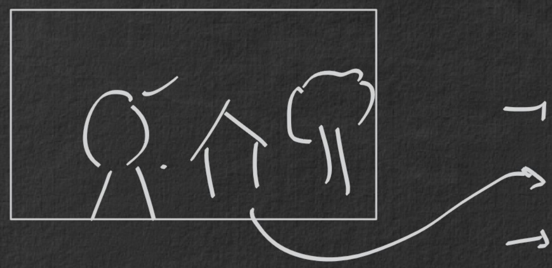
d_2



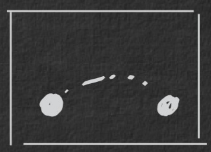
d_3



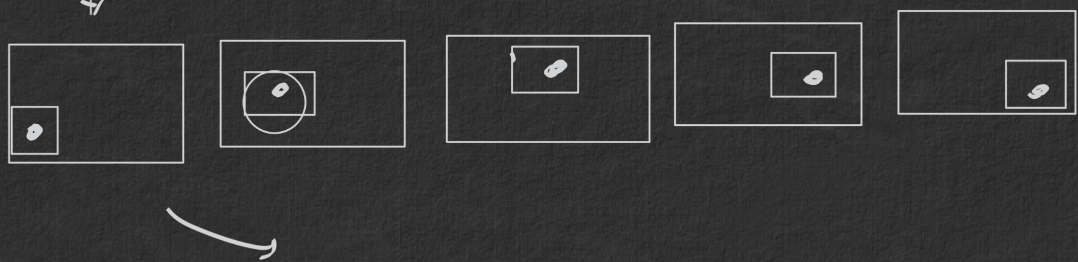




→ x,y,w,h



5 sec



video → OD → every frame

zoom

↓

meeting

↓

→ Names → time in   time out

100 classes x 3 hours

300 hours

Ajit   259 classes

pixel