Mask Up

Face mask detection through deep learning

Intro &
Problem Statement

Ongoing Pandemic

- Control of covid-19 pandemic
- Wearing mask is still the most effective way to inhibit virus transmission

Leverage tech to check for mask wearing compliance

- Use deep learning to allow computers to "see" the world around us
- Manpower could be deployed to help healthcare workers instead

How?

01.

Select Model

1. Single stage object detection model e.g. YOLO (you only look once)
2. Fast inference time, and high accuracy

02.

Collection of Dataset

1. Supervised problem
2. Sufficiently large dataset is required for training

03.

Training Model

1. Train the model so that it has good IoU score

04.

Deployment

1. Deploy trained model on the edge (IoT devices)

Data Collection and Cleaning

Data Collection

- Data available on Kaggle
– 848 images and annotation files

Duplicated images

- Using difPy, 5 duplicated images detected and removed (comparing MSE between 2 images)
- Important to remove duplicated imaged as they can cause data leakage

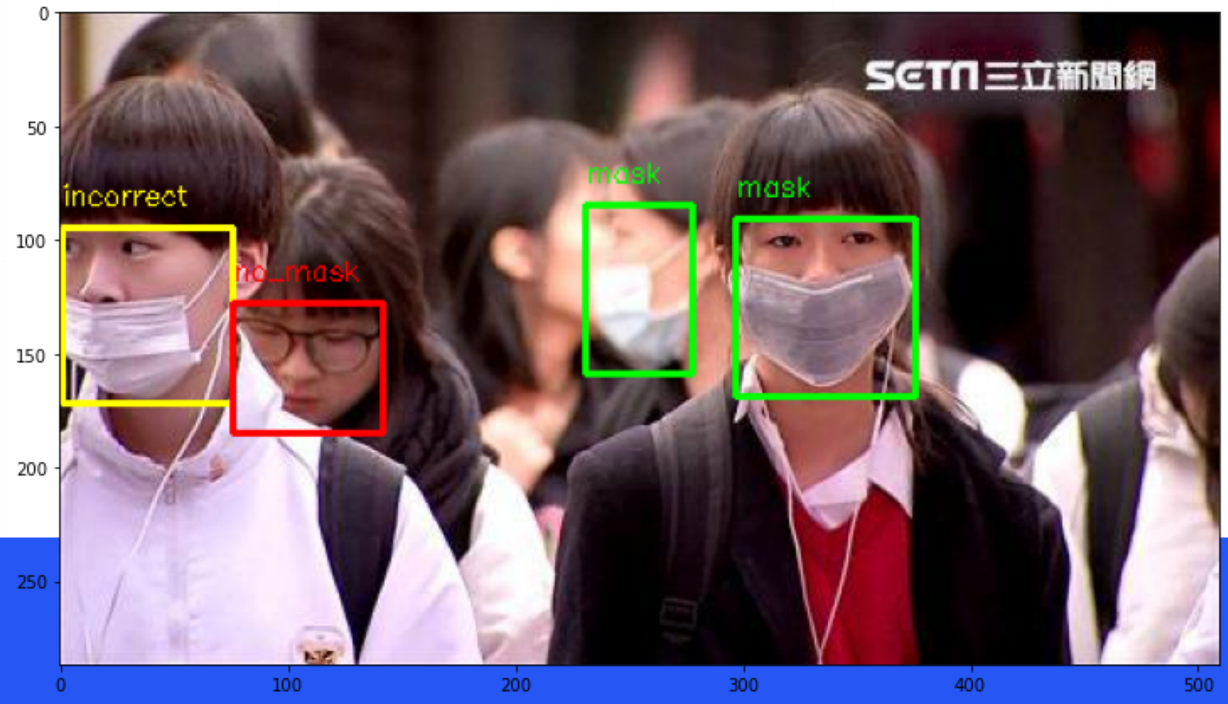
Change class label

- without_mask > no_mask
- with_mask > mask
- mask_weared_incorrect > incorrect

EDA

Check the quality of annotation

- Print out images and annotation information
- Check the bbox and class

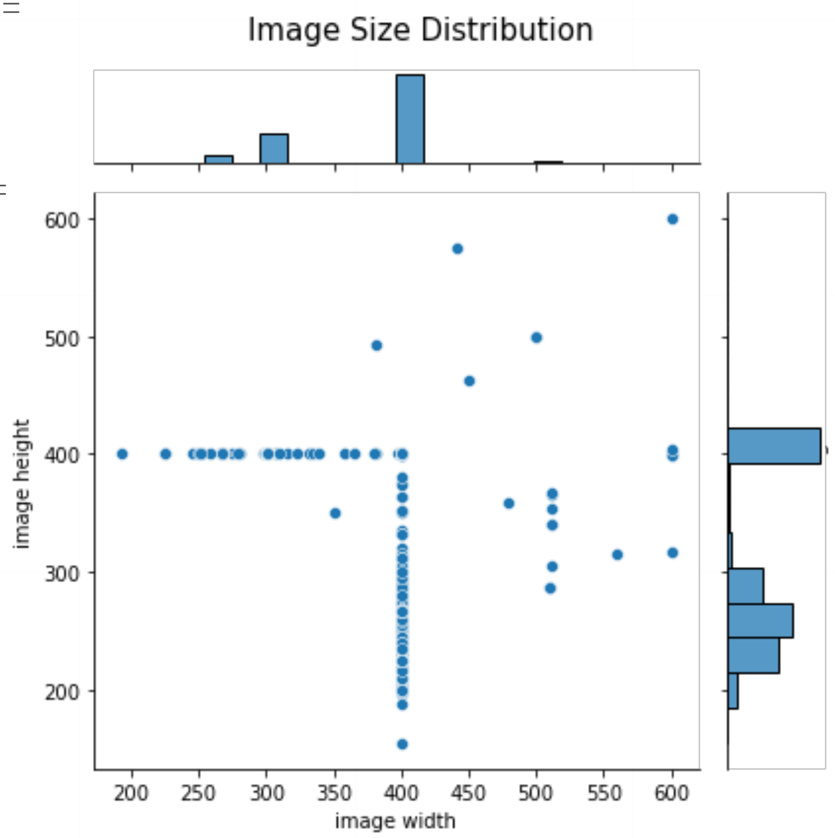


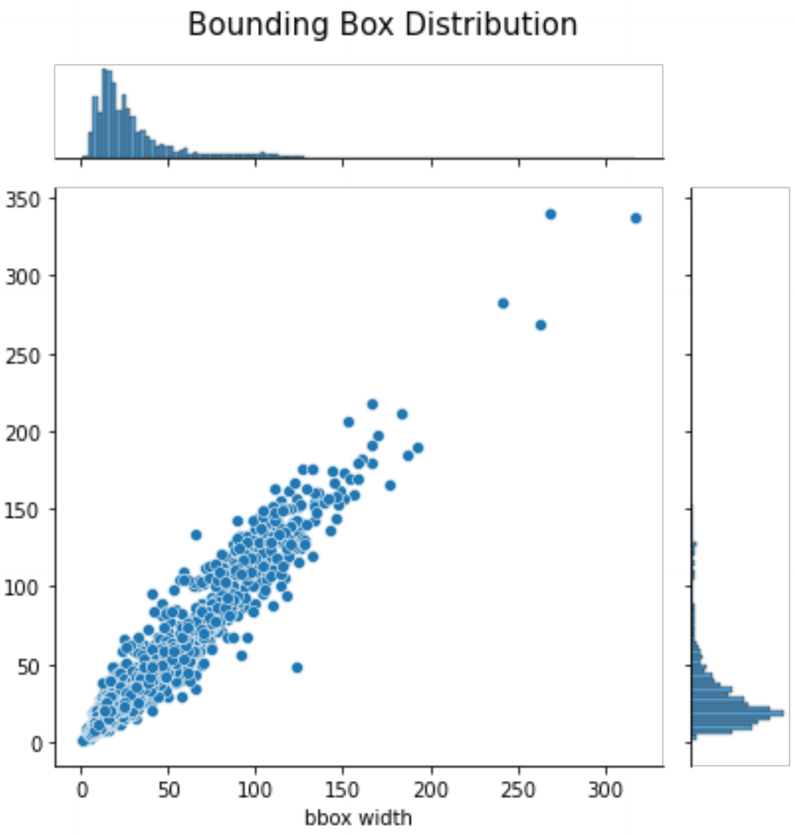
EDA

Check image size distribution

- ====== Bounding boxes statistics ======
- median width : 22.0
- median height: 24.0

- ============ Image statistics ========
- median width : 280.5
- median height: 280.5





Class distribution

- mask 0.79
- no_mask 0.18
- incorrect 0.03

Preprocessing - Roboflow image processing tool

Train, validation, split

- 70%, 20%, 10% split (593, 170, 85 images respectively)
- Class imbalance, hence split need to be stratify
- Roboflow does the split quite effortlessly ensuring that each class is represented in the correct proportion.

Image resizing

- Resized to square ratio, selected 640x640 and 320x320
- To maintained the aspect ration, padding is required. Black pixels used.

Preprocessing - Roboflow image processing tool

Image augmentation

- (1) horizontal flip, (2) +-10° rotation, and (3) applied mosaic

- Increase the training data from 593 -> 1,779 (~ 3 times)

- Note that validation and test set remained at 170 and 85 images. Ground truth images, they should not be augmented.

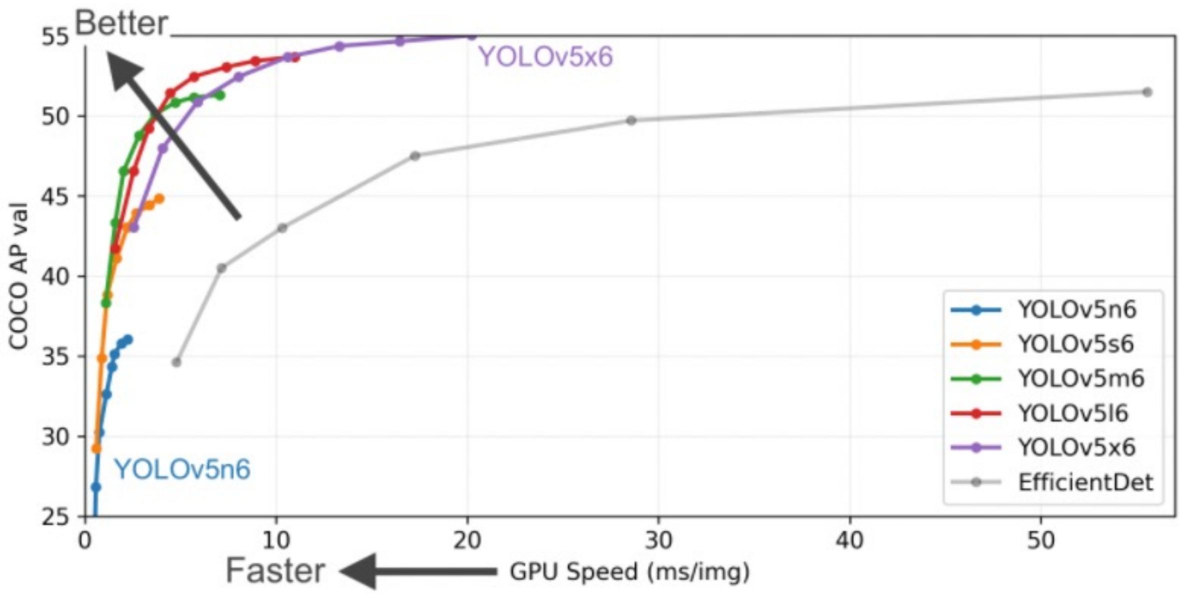
Modelling – YOLOv5

4 different runs

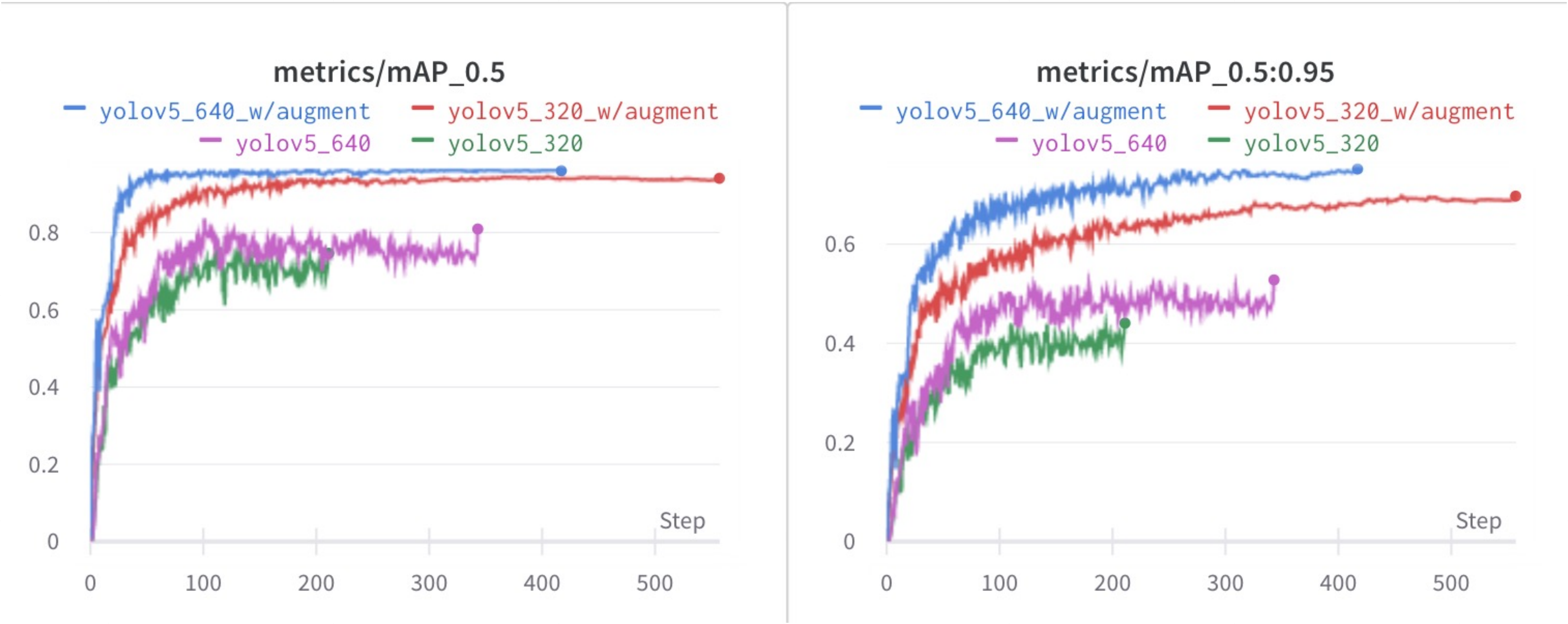
1. input image size 320, with no augmentations
2. input image size 320, with augmentations
3. input image size 640, with no augmentations
4. input image size 640, with augmentations

Transfer learning

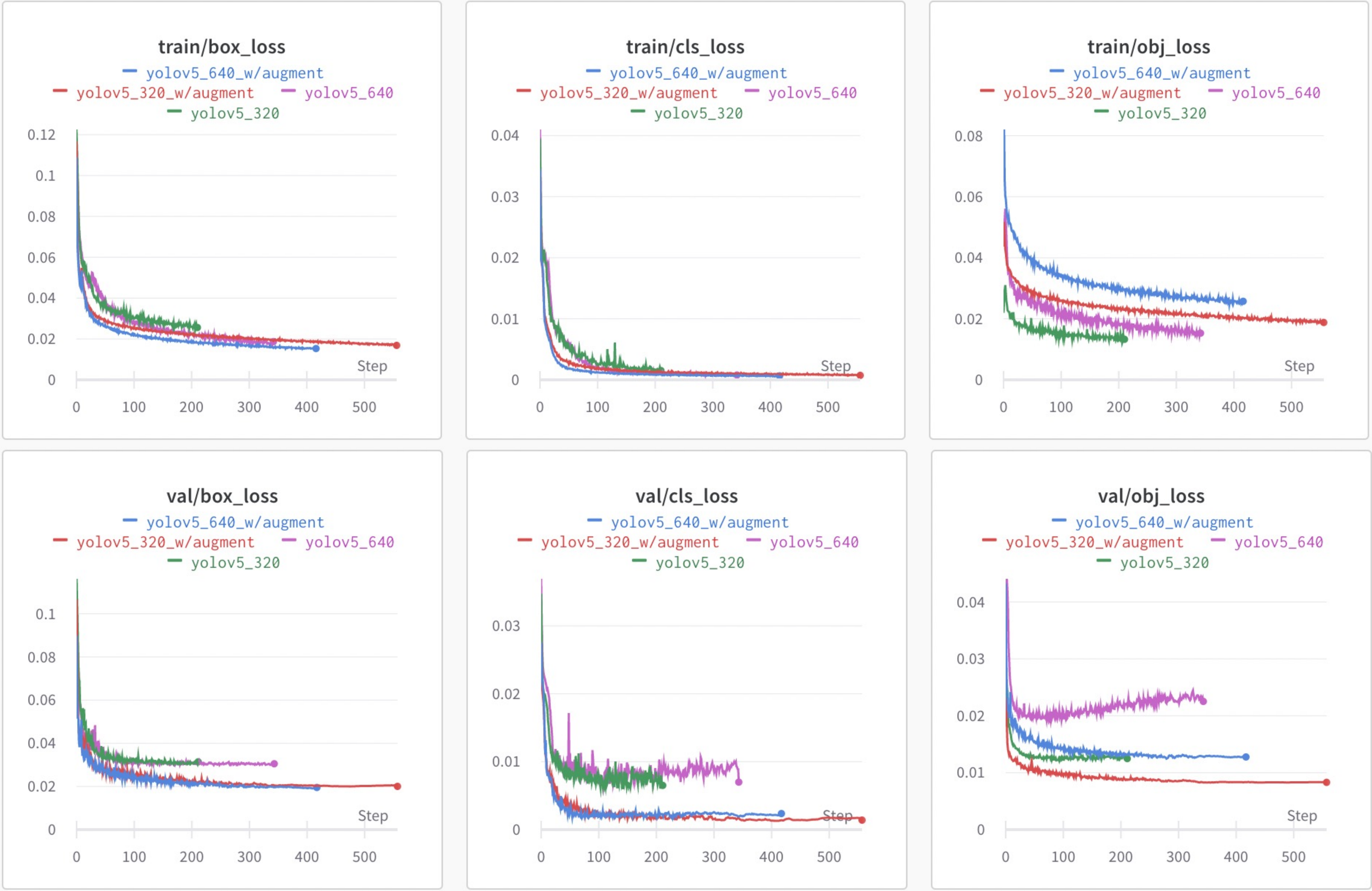
1. Pre-trained weights yolovs
2. “S” category is a lighter model, faster inference speed, but lesser accuracy



Results – Weights and Biases

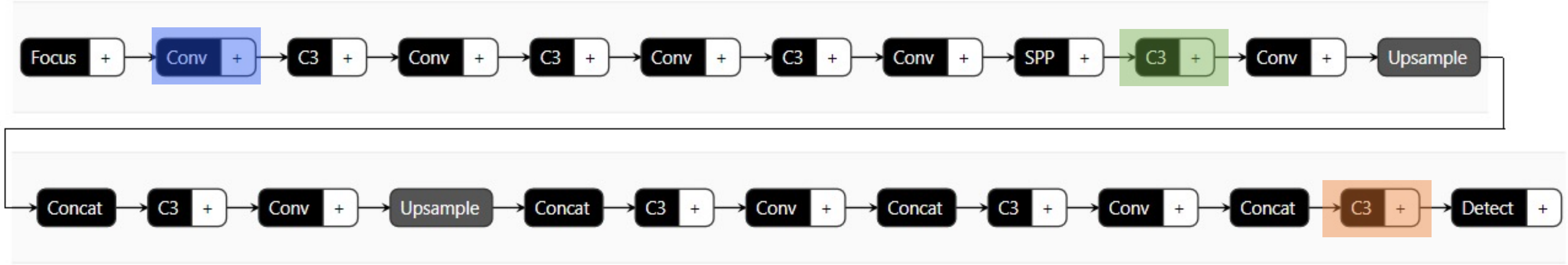


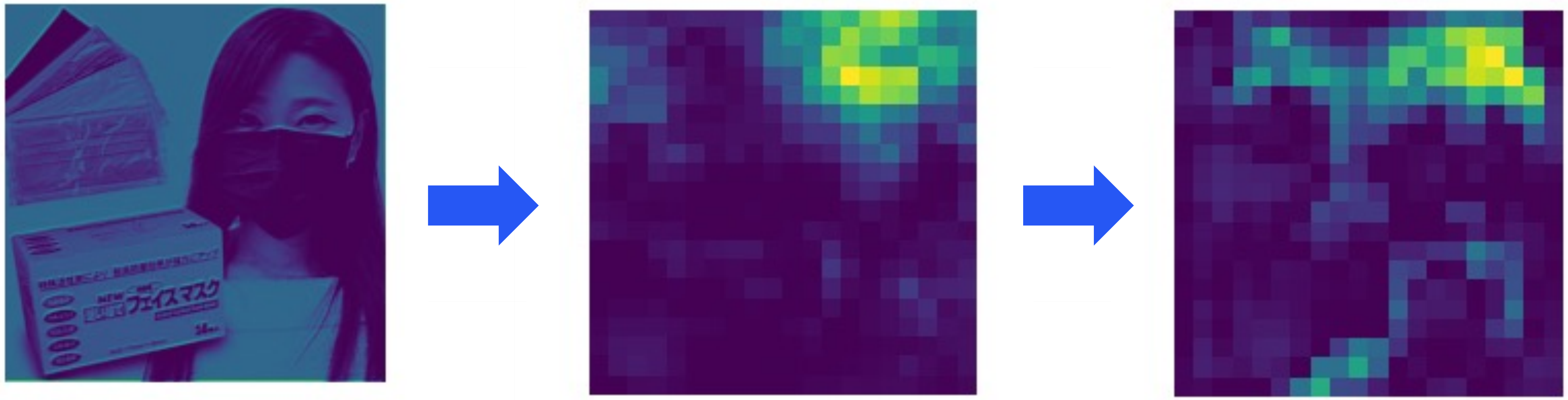
Results


train/box_loss
yolov5_640_w/augment
yolov5_320_w/augment
yolov5_640
yolov5_320
0.12
0.1
0.08
0.06
0.04
0.02
0
Step
0 100 200 300 400 500
train/cls_loss
yolov5_640_w/augment
yolov5_320_w/augment
yolov5_640
yolov5_320
0.04
0.03
0.02
0.01
0
Step
0 100 200 300 400 500
train/obj_loss
yolov5_640_w/augment
yolov5_320_w/augment
yolov5_640
yolov5_320
0.08
0.06
0.04
0.02
0
Step
0 100 200 300 400 500
val/box_loss
yolov5_640_w/augment
yolov5_320_w/augment
yolov5_640
yolov5_320
0.1
0.08
0.06
0.04
0.02
0
Step
0 100 200 300 400 500
val/cls_loss
yolov5_640_w/augment
yolov5_320_w/augment
yolov5_640
yolov5_320
0.03
0.02
0.01
0
Step
0 100 200 300 400 500
val/obj_loss
yolov5_640_w/augment
yolov5_320_w/augment
yolov5_640
yolov5_320
0.04
0.03
0.02
0.01
0
Step
0 100 200 300 400 500

Results - Summary

Visualizing feature maps





Stage 1 Stage 9 Stage 23

Discussion on inference result

GOOD, BAD, AND UGLY...


Ground Truth: maksssksksss253
Inference: maksssksksss253
0
100
200
300
400
500
600
no_mask
mask
incorrect
mask 0.95
no_mask 0.92
mask 0.93
no_mask 0.93
no_mask 0.92
mask 0.95
mask 0.94
incorrect 0.95
mask 0.96

GOOD


Ground Truth: maksssksksss179
Inference: maksssksksss179
mask
no_mask
mask 0.95
mask 0.81
no_mask 0.93
no_mask 0.91
mask 0.73
0
100
200
300
400
500
600

BAD


Ground Truth: maksssksksss633
Inference: maksssksksss633
mask
mask 0.92
0
100
200
300
400
500
600

UGLY


no_mask 0.91
mask 0.87
SOUTH KOREA
URGES CALM
cna
tine with COVID-19
UK tech tycoon Mike Lynch fails in bid to delay extradition ruling

Thanks !

Insight to computer vision

Contents

01

Overview

- Introduction to CV
- Object detection

02

Metrics

- Intersection over Union (IoU)
- Mean average precision (mAP)

03

Non-Max Suppression

- Common algorithm to tackle multiple bounding boxes

Overview – What is CV?

Classification

Object detection

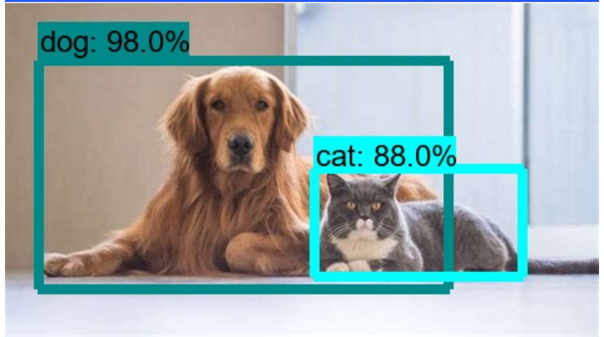


Image Segmentation (panoptic)

Keypoint detection

and many more...

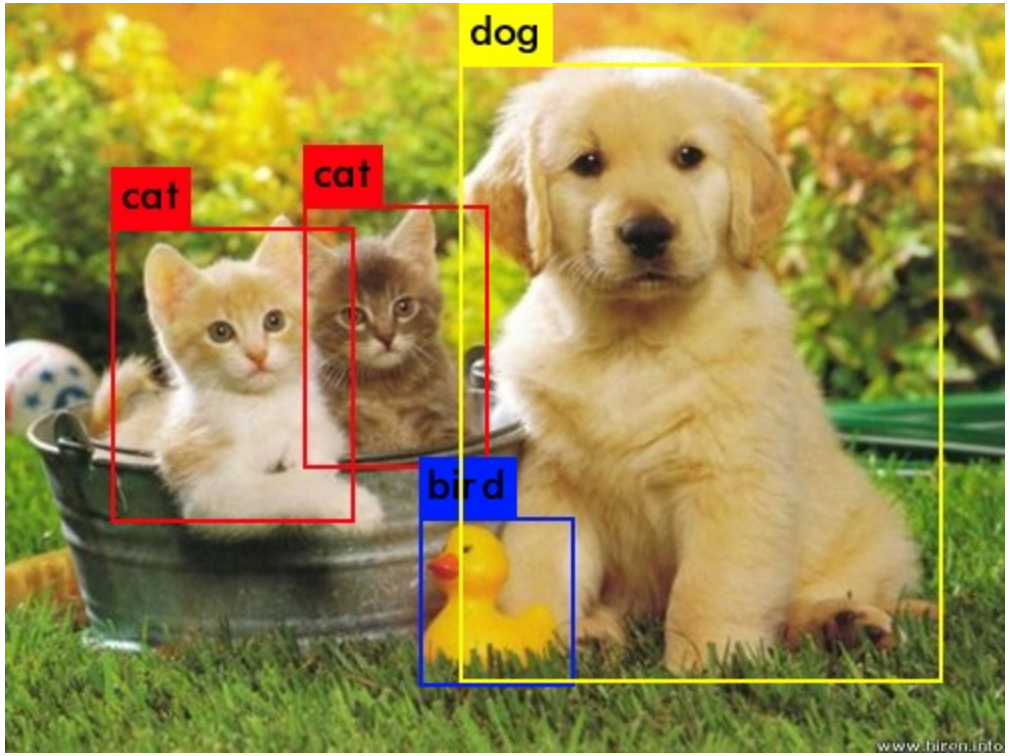
Object Detection Task

Localization (regression)

- Localized the subject in the image
- Done by drawing bounding boxes (bbox) around the subject
- Bbox described as (x1, y1, x2, y2) or (x, y, w, h)

Classification

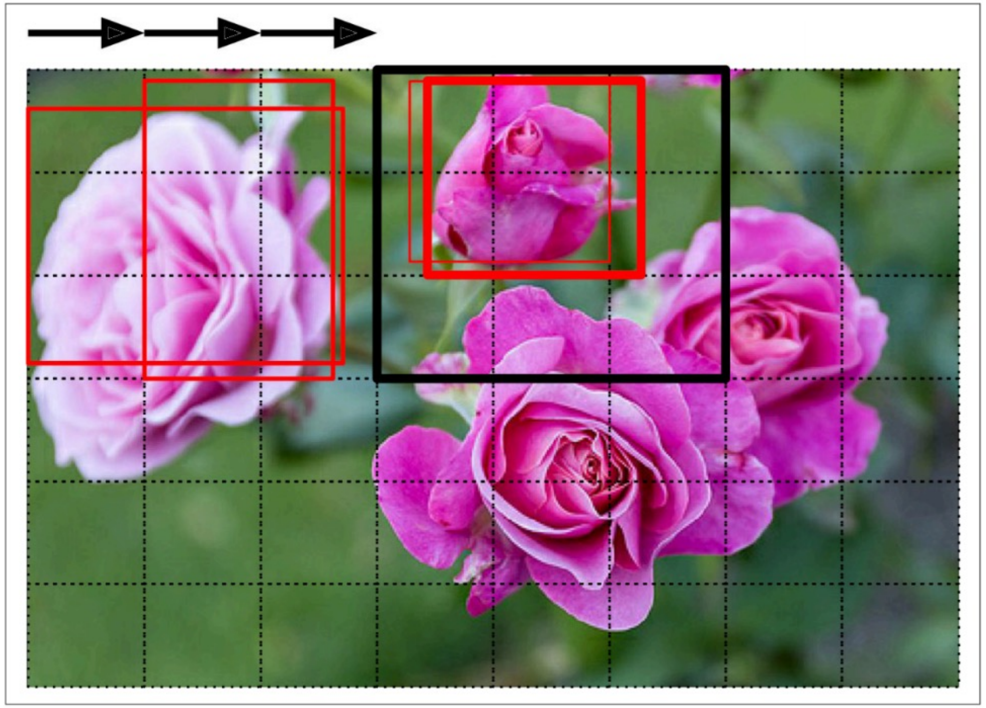
- Classify subject correctly (e.g. cat or dog)



Object detection models

Two-stage models

- Region-based CNN Family Models
  - R-CNN (2014), Fast R-CNN (2015), Faster R-CNN (2016), Mask R-CNN (2017)
- More accurate but are typically slower.



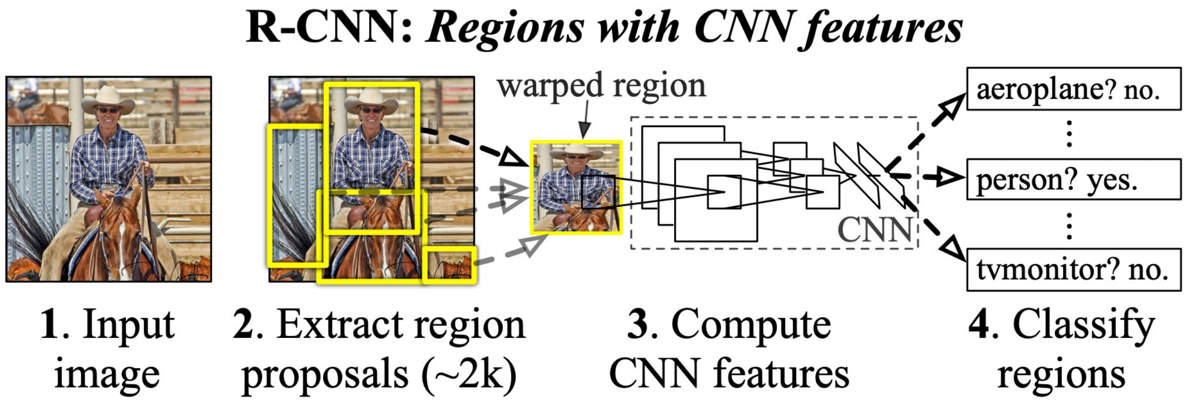


Figure 1: Object detection system overview. Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs.

Sliding window technique

Region proposal technique

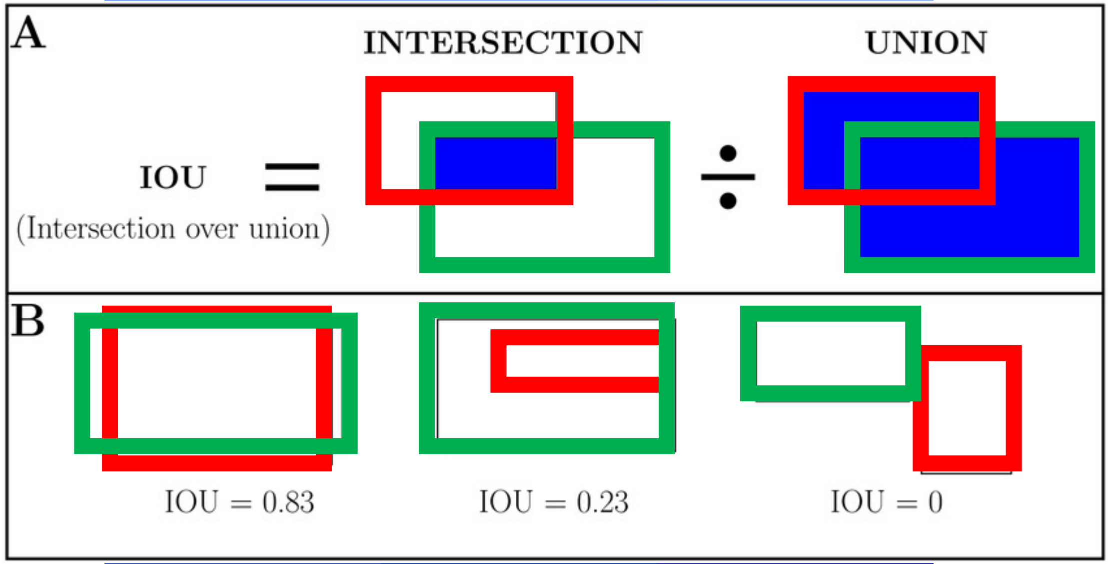
object detection models

One-stage models

- YOLO Family Models
  - YOLO (2015), YOLOv2 (2016), YOLOv3 (2018), YOLOv4 (2020), YOLOv5 (2021)
- SSD (2016)
- Fast inference speed, but not as good at recognizing a group of small objects

Evaluation metric: Intersection over Union (IoU)

- IoU quantify the amount of overlap area between predicted bbox overlapped with ground-truth bbox
- Range from 0 (prediction totally off) to 1 (prefect prediction)
- If prediction has IoU > 0.5, correct classification -> TP
- If prediction has IoU < 0.5, correct classification -> FP
- If no prediction when there is a target -> FN
  (0.5 is an arbitrary threshold)

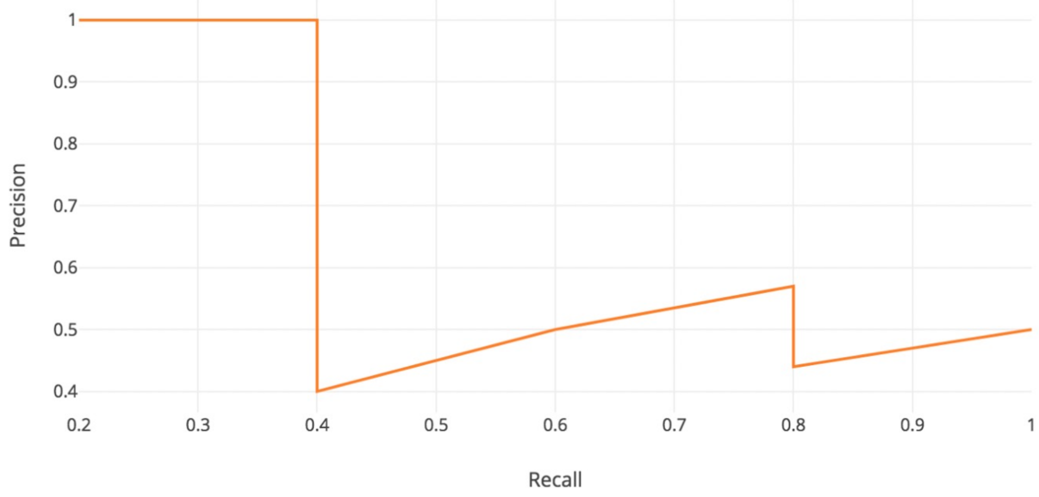


Evaluation metric: Mean Average Precision (mAP)

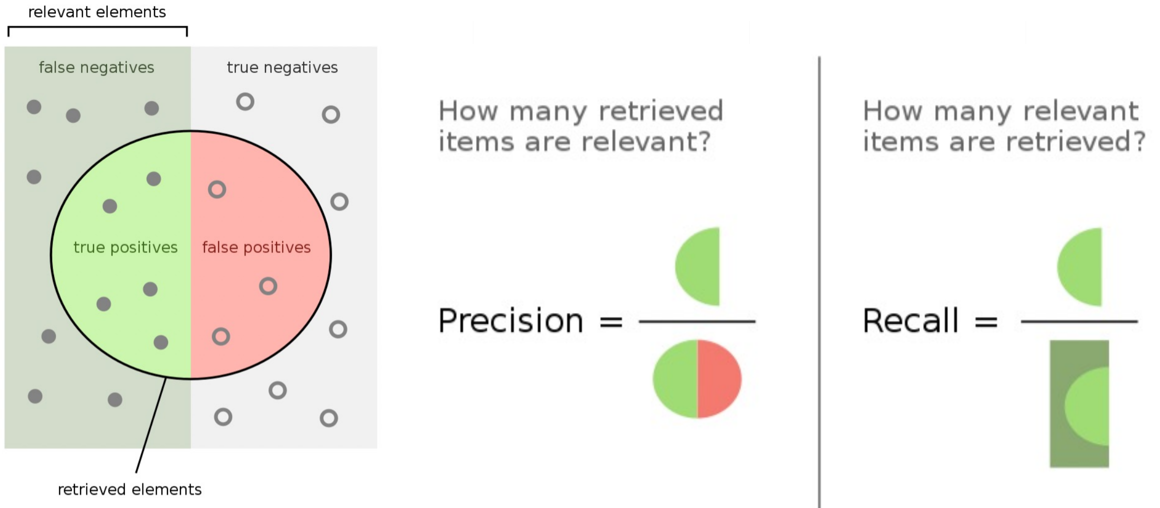
- Based on the concept of IoU, calculate TP, FP, FN to built a precision –recall curve
- STEPS :
  - Sort the predictions by confidence level
  - Compute the precision and recall for each row sequentially
  - Plot graph
- Calculate area under graph and you get AP!
- Repeat this for diff classes and average them e.g. AP_{CAT}, AP_{DOG}
- Hence mean AP

5 total cases in this e.g.

Rank	Correct?	Precision	Recall
1	True	1.0 1 / 1	0.2 1 / 5
2	True	1.0 2 / 2	0.4 2 / 5
3	False	0.67 2 / 3	0.4 2 / 5
4	False	0.5 2 / 4	0.4 2 / 5
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0



Precision-recall curve



Evaluation metric: Mean Average Precision (mAP)

COCO Object Detection Challenge

Average Precision (AP):

- AP % AP at IoU=.50:.05:.95 **(primary challenge metric)**
- $AP^{IoU=.50}$ % AP at IoU=.50 (PASCAL VOC metric)
- $AP^{IoU=.75}$ % AP at IoU=.75 (strict metric)

AP Across Scales:

- AP^{small} % AP for small objects: area < 32^2
- AP^{medium} % AP for medium objects: 32^2 < area < 96^2
- AP^{large} % AP for large objects: area > 96^2

Average Recall (AR):

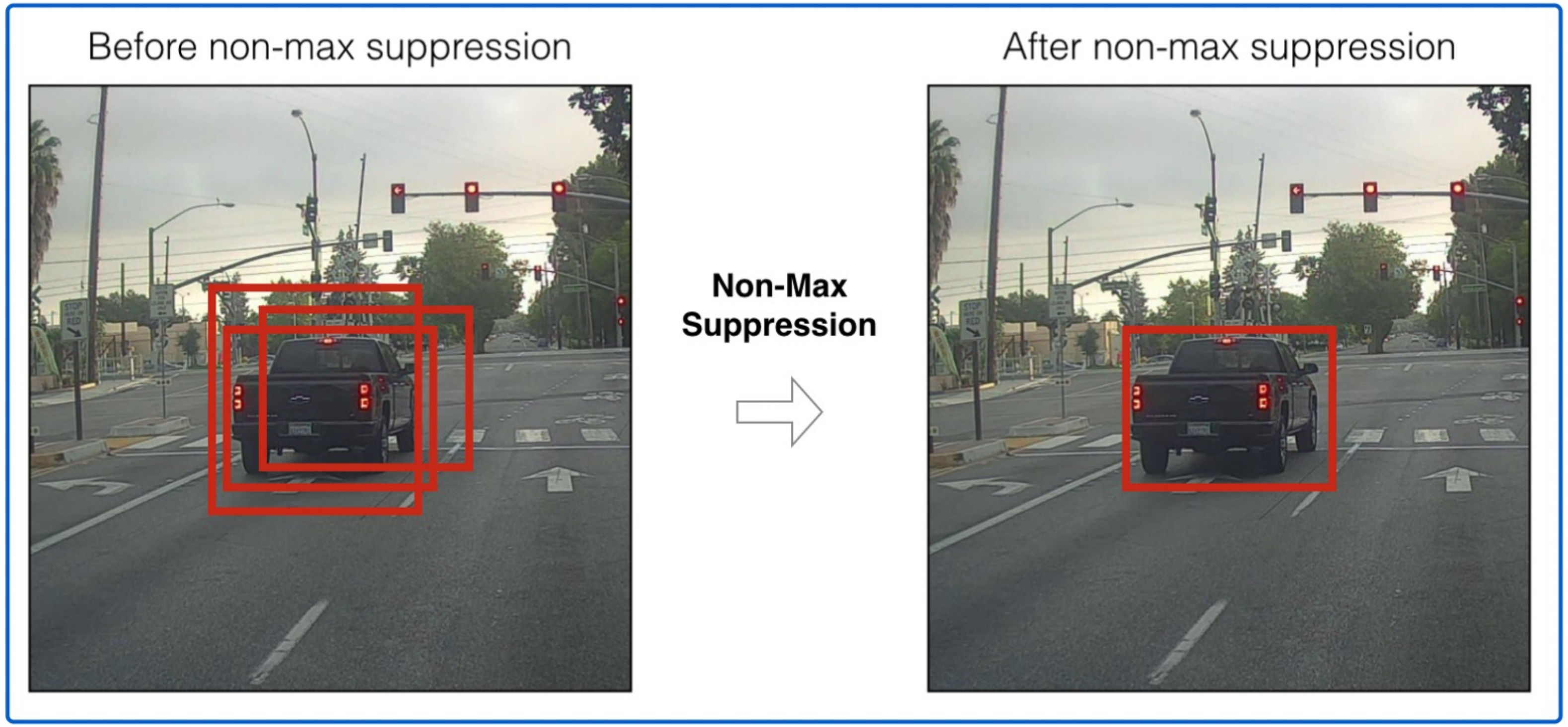
- $AR^{max=1}$ % AR given 1 detection per image
- $AR^{max=10}$ % AR given 10 detections per image
- $AR^{max=100}$ % AR given 100 detections per image

AR Across Scales:

- AR^{small} % AR for small objects: area < 32^2
- AR^{medium} % AR for medium objects: 32^2 < area < 96^2
- AR^{large} % AR for large objects: area > 96^2

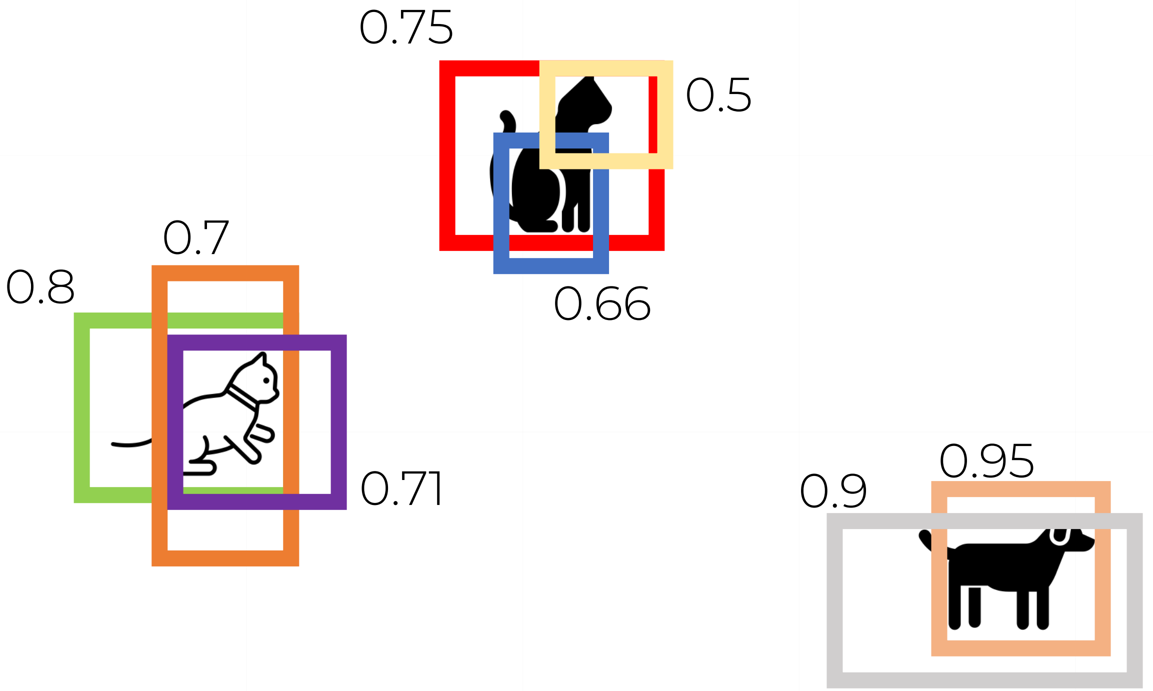
Non-max suppression (NMS)

- Most obj detection models proposed multiple bbox around the target.
- NMS is a technique to remove all except one bbox proposal

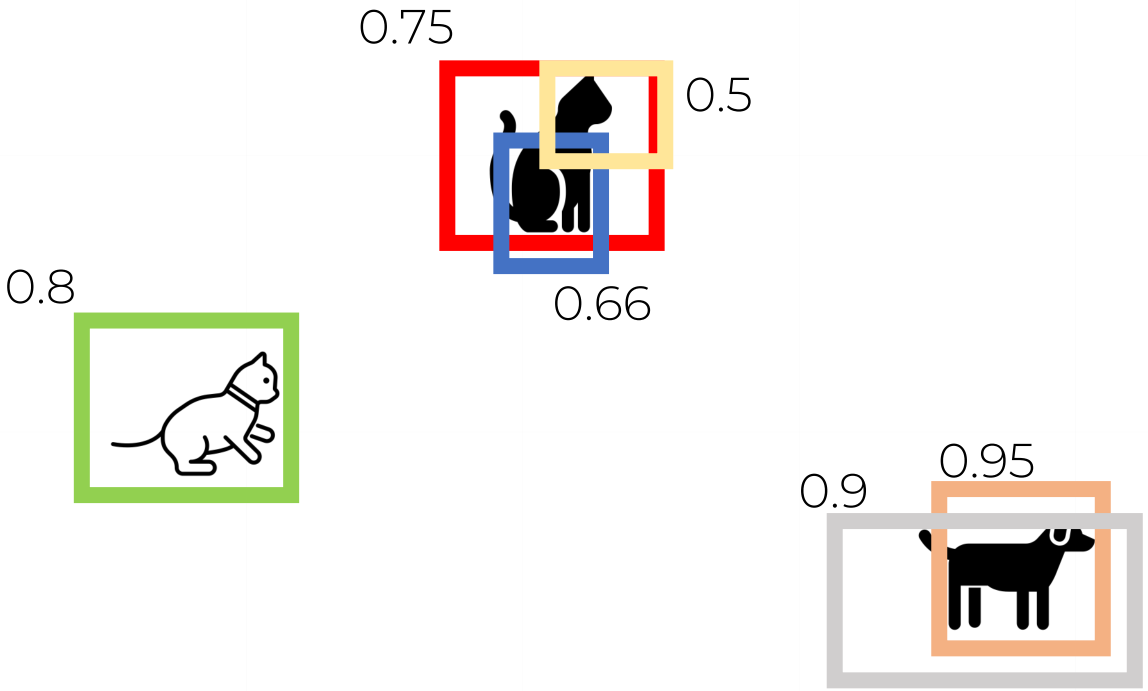


Steps:

1. Start from one class
2. Select the bbox with the highest conf idence score.
3. Check the IOU score of other bboxes.
4. Discard the bbox if IOU > 0.7 (arbitrar y threshold
5. Select the second highest confidence score and repeat step 3 and 4.
6. Repeat step 1 to 5 for other classes.

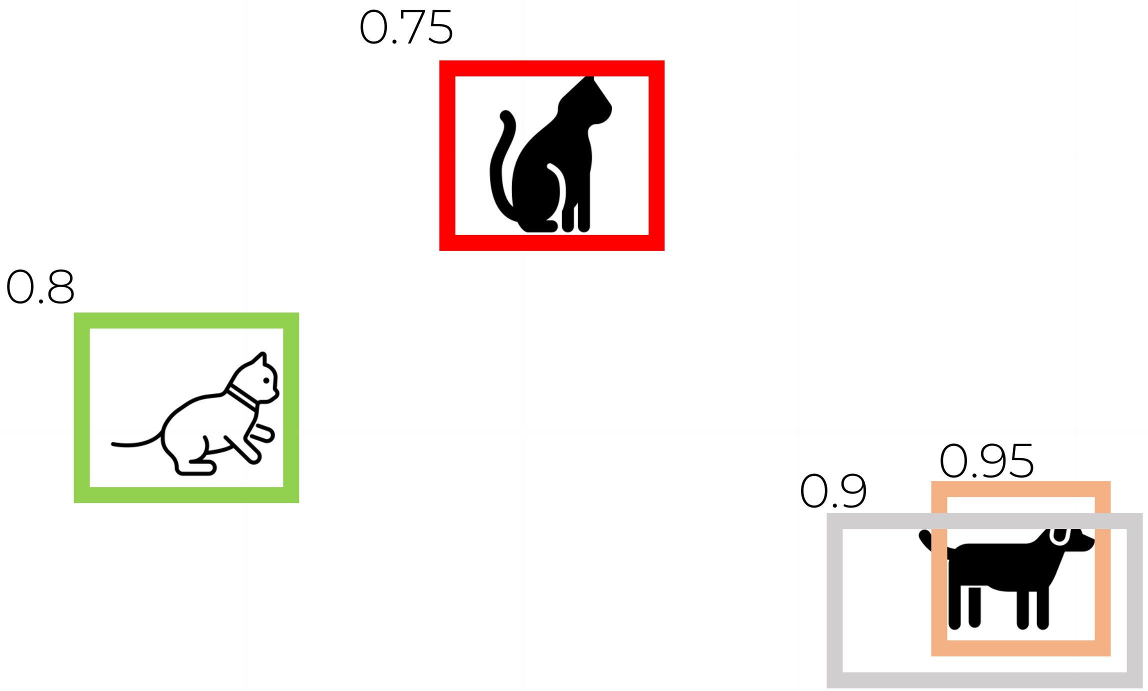


Steps:
1. Start from one class
2. Select the bbox with the highest conf idence score.
3. Check the IOU score of other bboxes.
4. Discard the bbox if IOU > 0.7 (arbitrar y threshold
5. Select the second highest confidence score and repeat step 3 and 4.
6. Repeat step 1 to 5 for other classes.



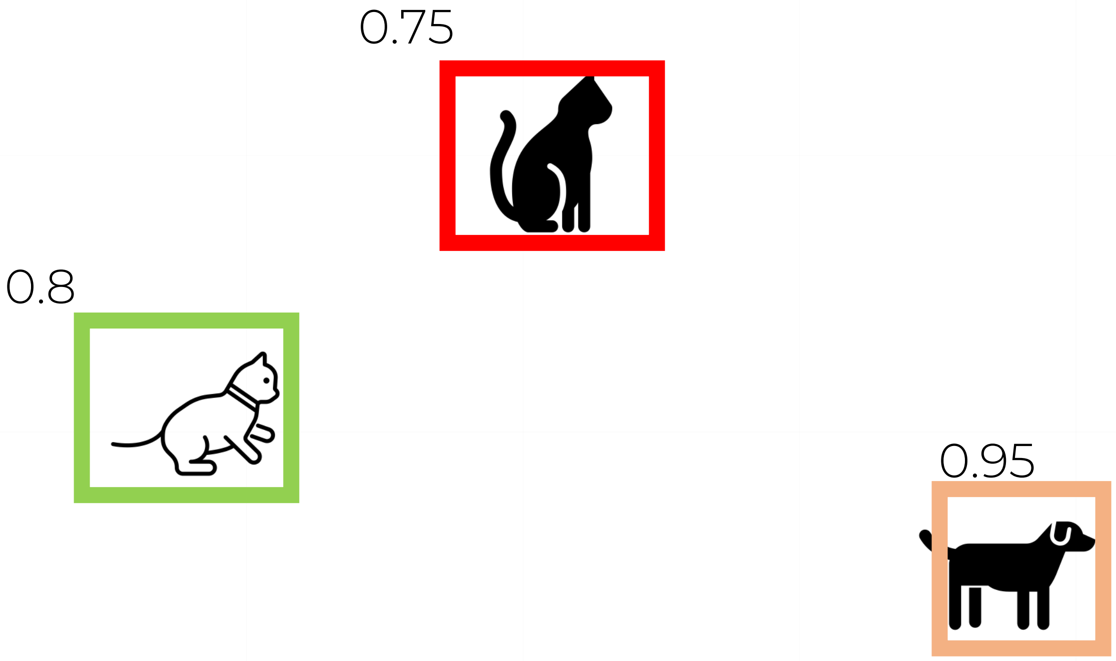
Steps:

1. Start from one class
2. Select the bbox with the highest conf idence score.
3. Check the IOU score of other bboxes.
4. Discard the bbox if IOU > 0.7 (arbitrar y threshold
5. Select the second highest confidence score and repeat step 3 and 4.
6. Repeat step 1 to 5 for other classes.



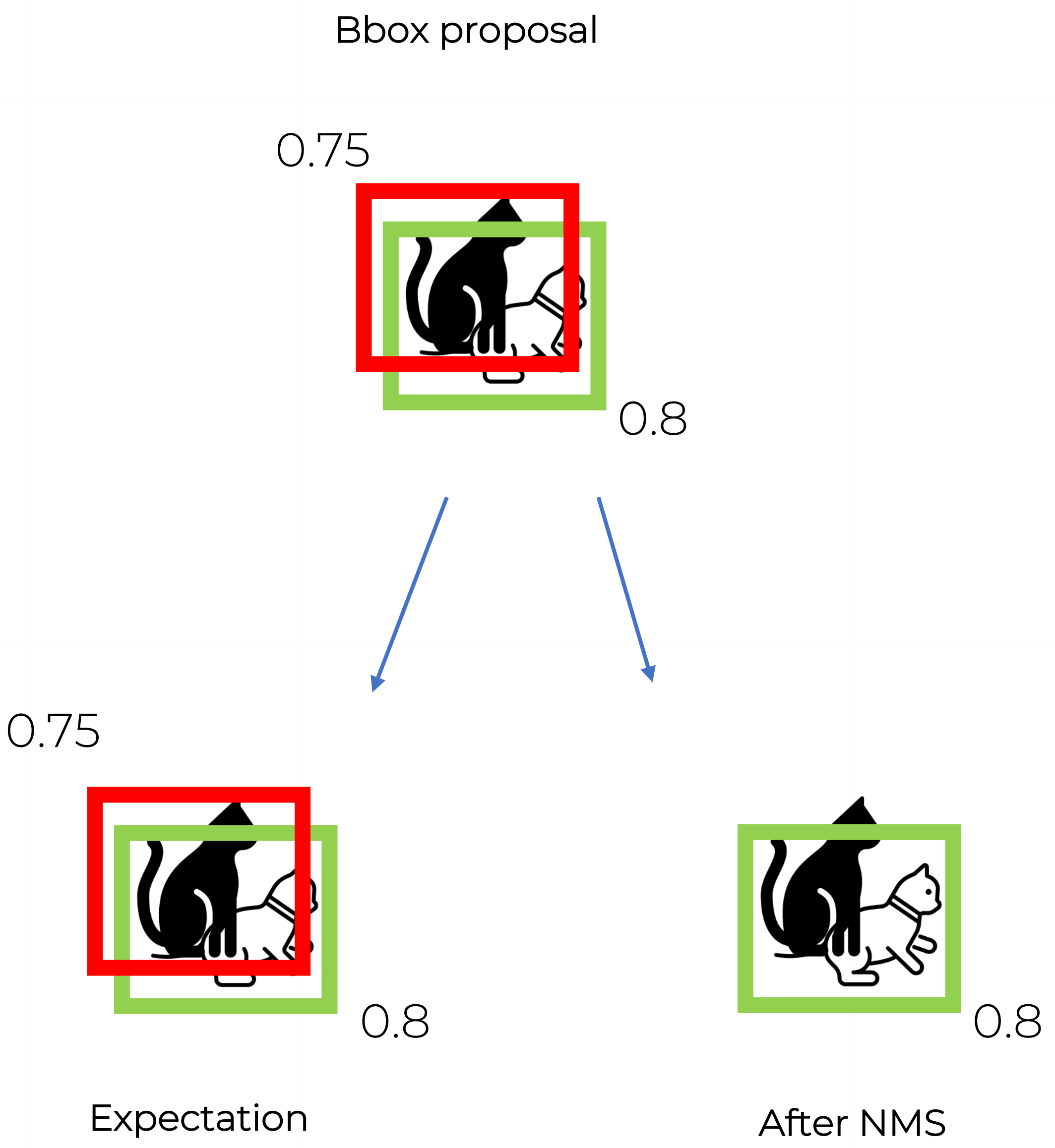
Steps:

1. Start from one class
2. Select the bbox with the highest conf idence score.
3. Check the IOU score of other bboxes.
4. Discard the bbox if IOU > 0.7 (arbitrar y threshold
5. Select the second highest confidence score and repeat step 3 and 4.
6. Repeat step 1 to 5 for other classes.



Steps:

1. Start from one class
2. Select the bbox with the highest confidence score.
3. Check the IOU score of other bboxes.
4. Discard the bbox if IOU > 0.7 (arbitrary threshold
5. Select the second highest confidence score and repeat step 3 and 4.
6. Repeat step 1 to 5 for other classes.



Problem?

NMS performed poorly if targets are close

Thanks !