FACE MASK DETECTION

KEZIAH SITINJAK

Contents

Part 1

Introduction

Part 2

Data Cleaning and EDA

Part 3

Overview of Computer Vision

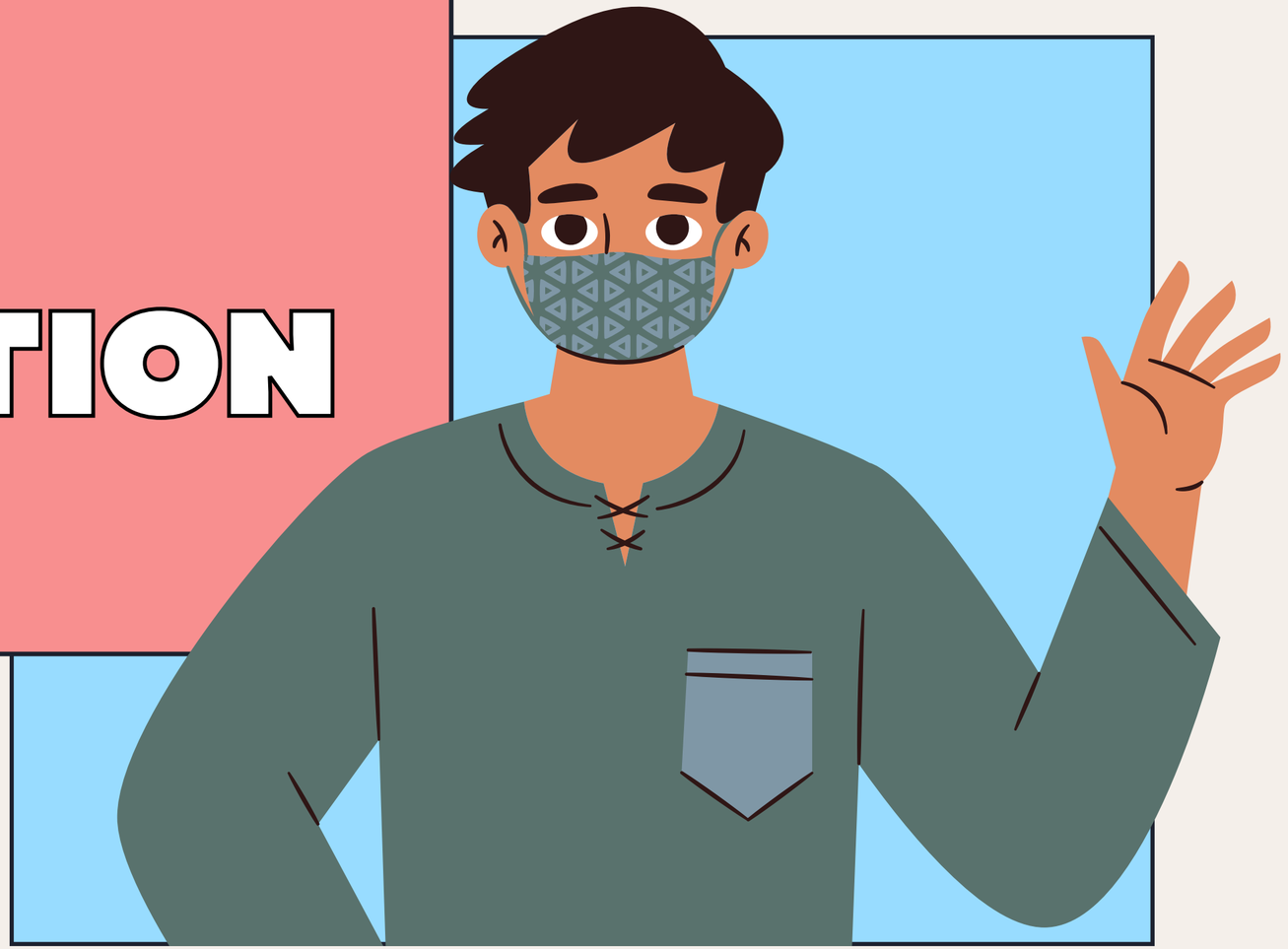
Part 4

Modeling and Evaluation

Part 5

Conclusion

INTRODUCTION



Background

- COVID-19 has posed a serious threat to global health.
- Wearing face masks in public areas is one of numerous precautions that may be taken to stop the spread of this infection.
- It is challenging to make sure that people are wearing masks in public transportation.

Objectives

Using computer vision and deep learning to detect whether a person is wearing mask, not wearing mask or improperly wearing mask.

DATA CLEANING & EDA

DATASET

832 IMAGES WITH ANNOTATIONS (11 DUPLICATE IMAGES REMOVED)

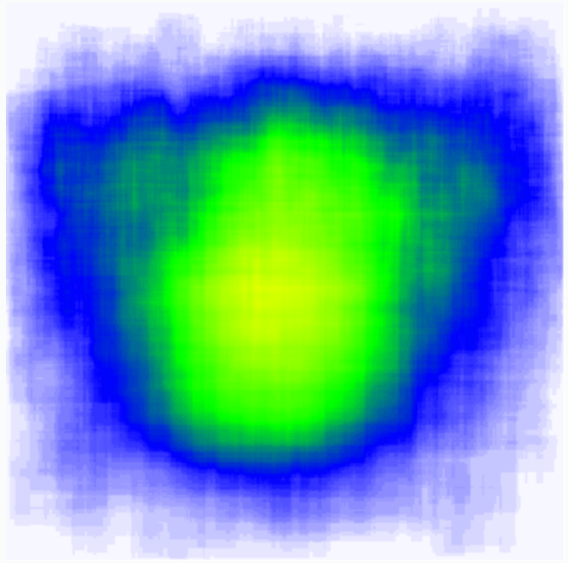
MEDIAN IMAGE RATIO

400 X 281

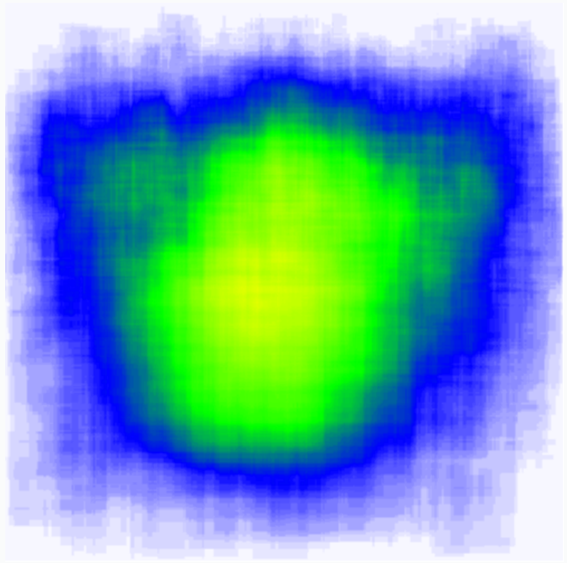
IMBALANCED CLASS

- with_mask : 79%
- without_mask : 18%
- mask_weared_incorrect : 3%

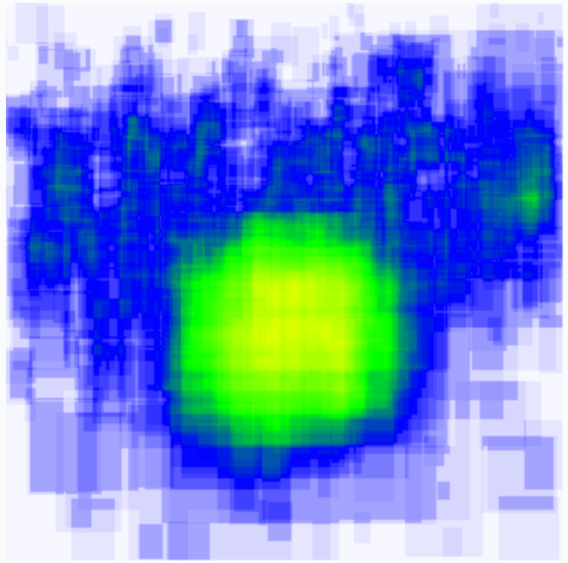
Annotation Heatmap



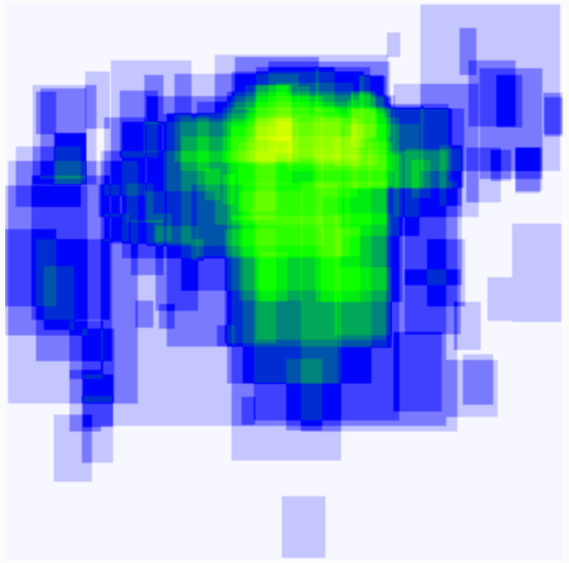
All
(4,021)



with_mask
(3,184)



without_mask
(716)



mask_weared_incorrect
(121)

Preprocessing

TRAIN-VALIDATION-TEST SPLIT

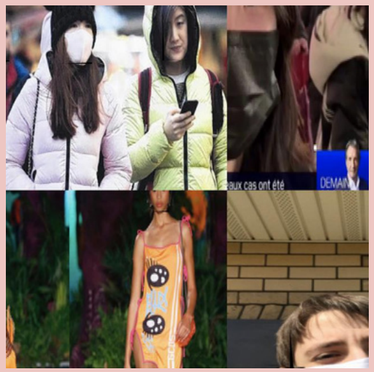
- 70%, 20%, 10% split (593, 170, 85 images respectively)
- Class imbalance, hence split need to be stratify (by Roboflow)

IMAGE RESIZE

- Our model require square image
- We resize the image to 320x320 or 640x640

IMAGE AUGMENTATION

- Applied mosaic, shear bounding box
- Only applied to training data



Applied mosaic, shear bounding box

- Increase the training data from 596 -> 1,788 (~ 3 times)
- Note that validation and test set remained at 168 and 84 images.

Ground truth images, they should not be augmented.

OVERVIEW OF COMPUTER VISION

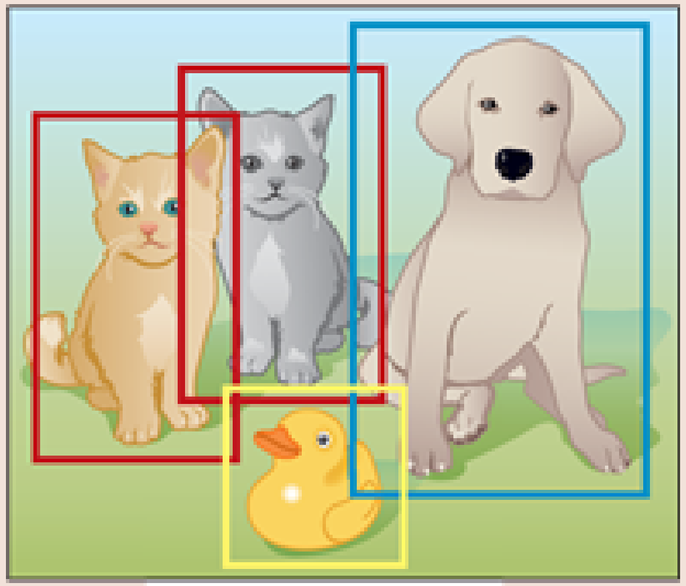
Classification vs. Object Detection

Image Classification

Cat

- One object and one label per image

Object Detection (classification and localization)

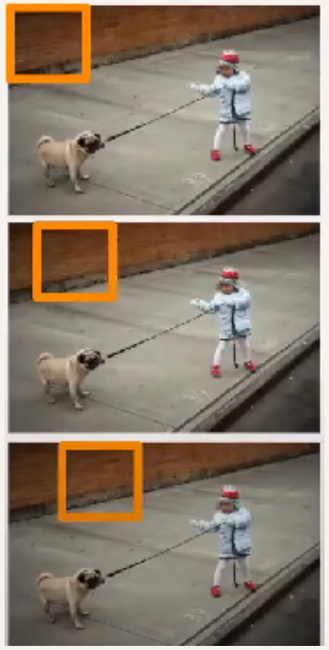


Cat, Cat, Duck, Dog

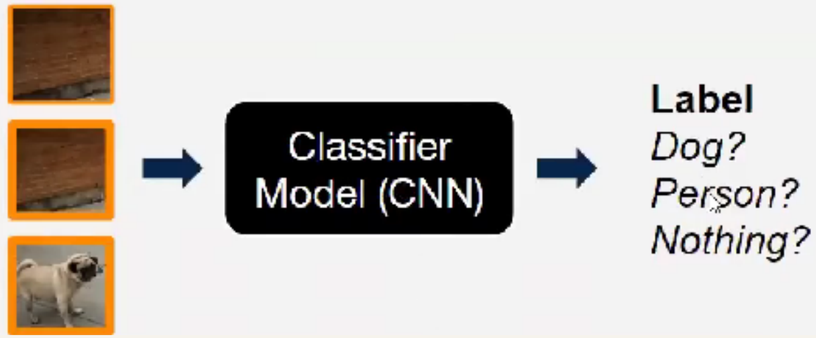
- Multiple objects per image
- Determine objects' location by drawing bounding boxes (bbox)
- Bbox described as (x, y, w, h)

Naive Approach

1. Scan the image with a sliding window



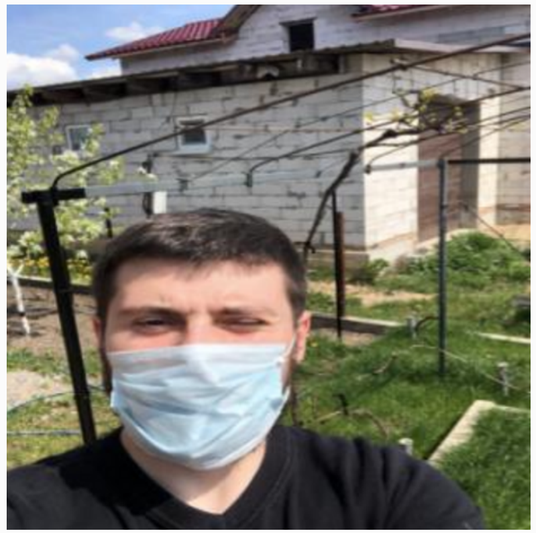
2. Feed the image to a classifier model to predict a label for that region



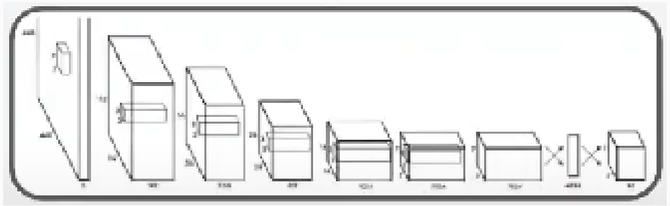
- Slow → Not good for real time uses
- Improved version: Region-based Convolutional Neural Net (**R-CNN**)
  - Strategically selects interesting regions to run through the classifier.

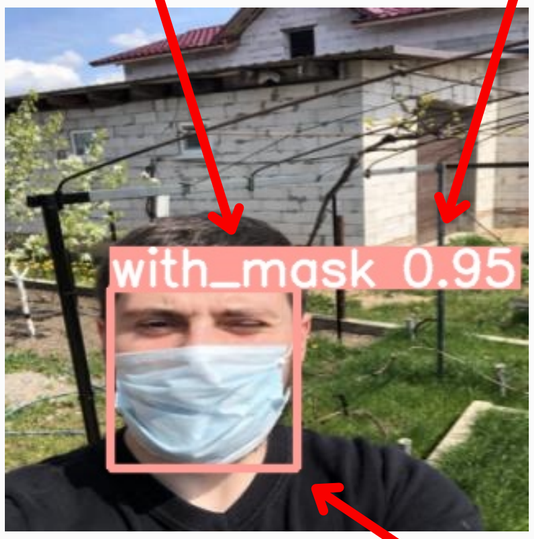
YOLO - You Only Look Once

- Instead of making predictions on many regions of an image, YOLO passes the entire image at once into a CNN (much faster!)
- The CNN that predicts the **labels, bounding boxes, and confidence probabilities** for objects in the image



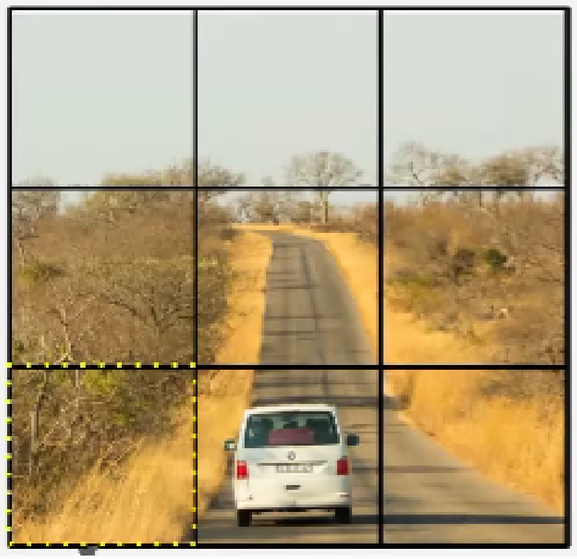
Convolutional Neural Network





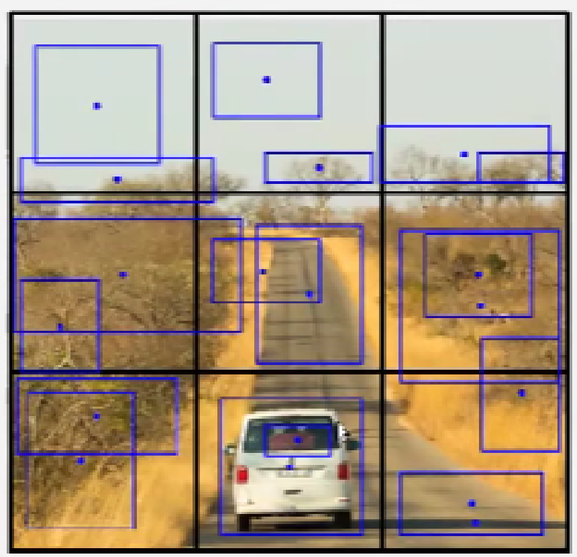
YOLO Steps

1. Divide the image into cells with an S x S grid.



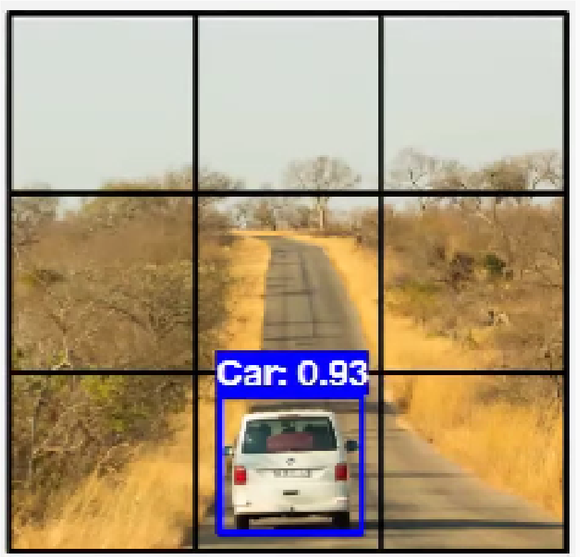
S=3

2. Each cell predicts B bounding boxes

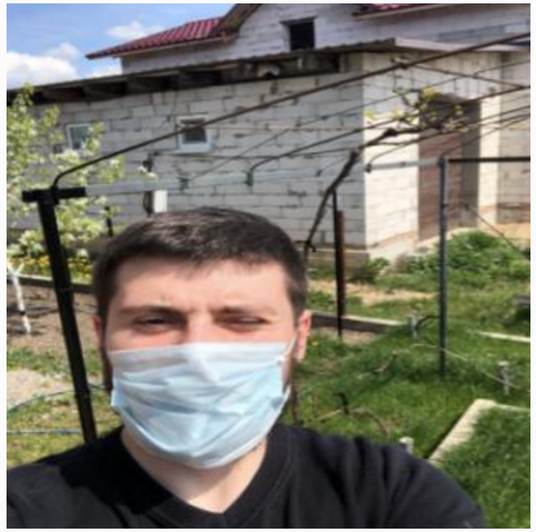


B=2

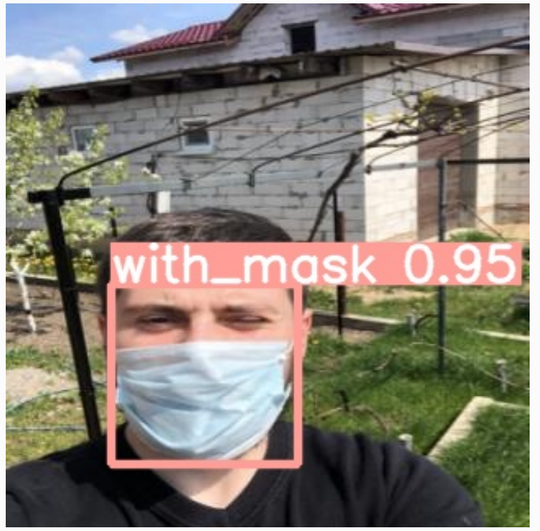
3. Return bounding boxes above confidence threshold

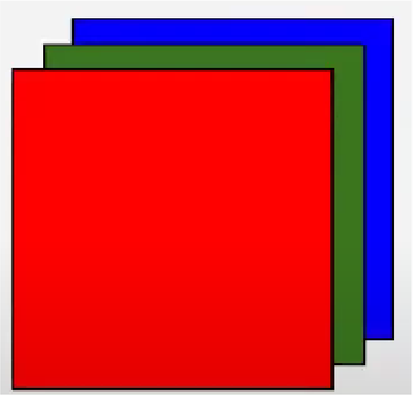


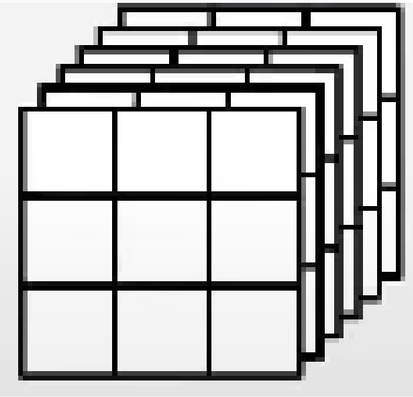
YOLO Overview



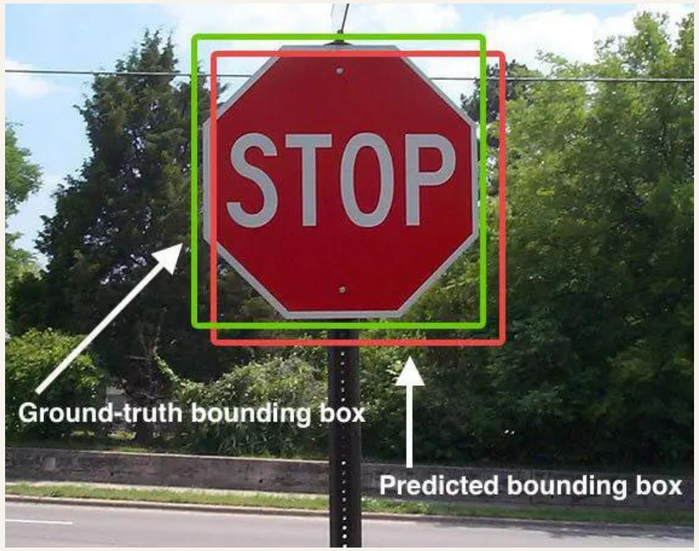
Convolutional
Neural Network



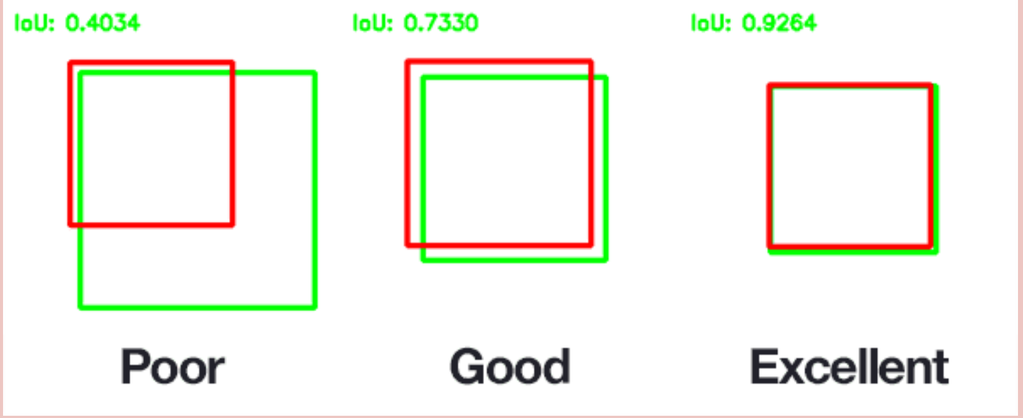




Evaluation Metric :
Intersection Over Union (IoU)



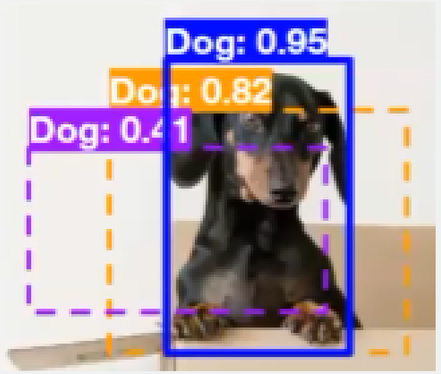
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Non-Max Suppresion

Double Counting Objects

NMS solves multiple counting by removing the box with lower confidence probability when the IoU between 2 boxes with the same label is above some threshold.



1. **Identify the box with the highest confidence**

IoU : 0.62

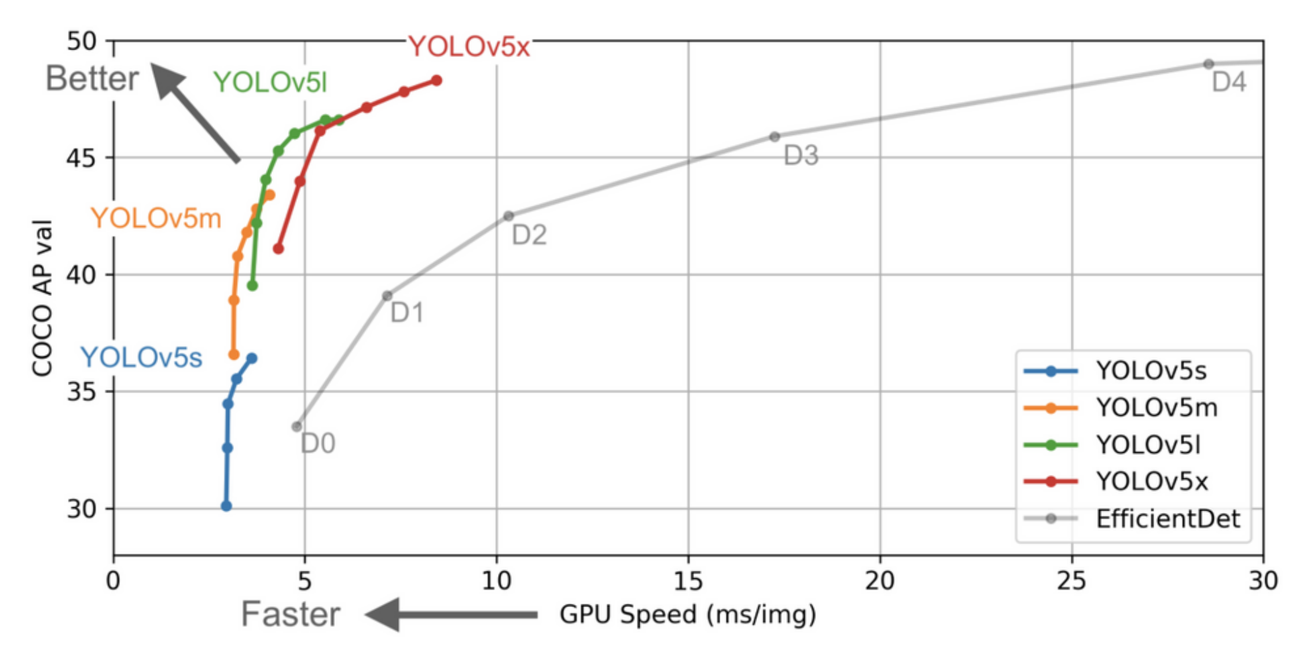
IoU : 0.47

2. Calculate the IoU between the highest confidence box and each of the other boxes

3. Suppress boxes with IoU above a selected threshold

IMPLEMENTING YOLO

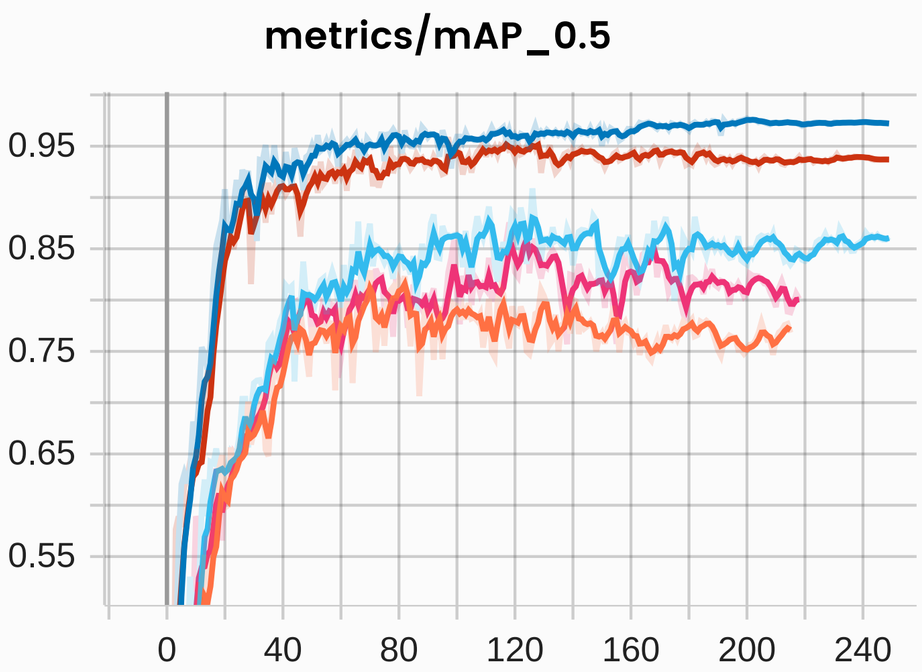
YOLO-V5 MODEL SELECTION

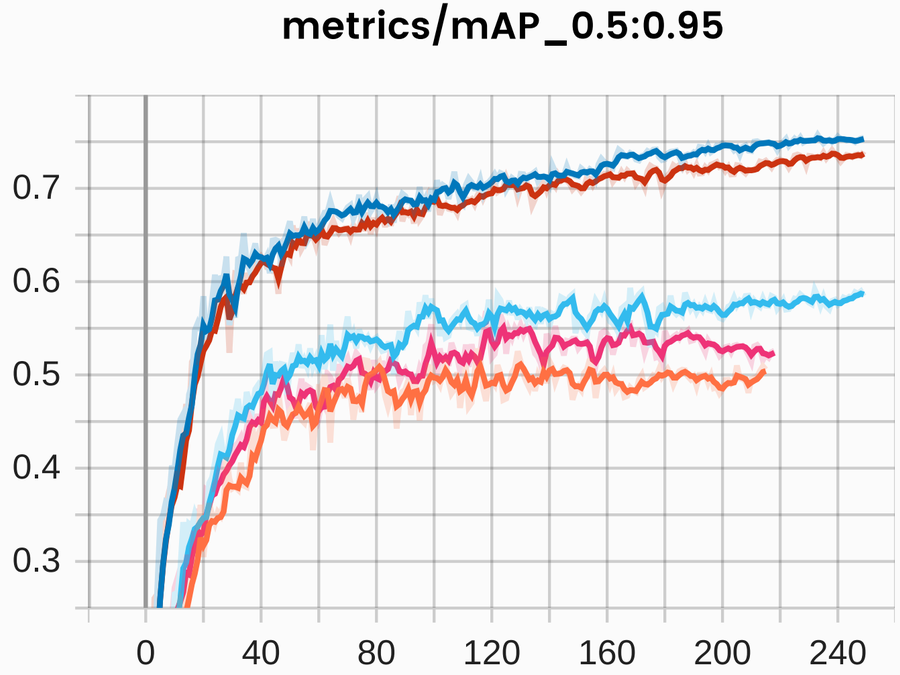


5 model runs:

1. Input image size 416 x416
2. Input image size 640 x 640 with augmentations
3. Input image size 320 x 320 with augmentations
4. Input image size 640 x 640
5. Input image size 320 x 320

EVALUATION METRICS

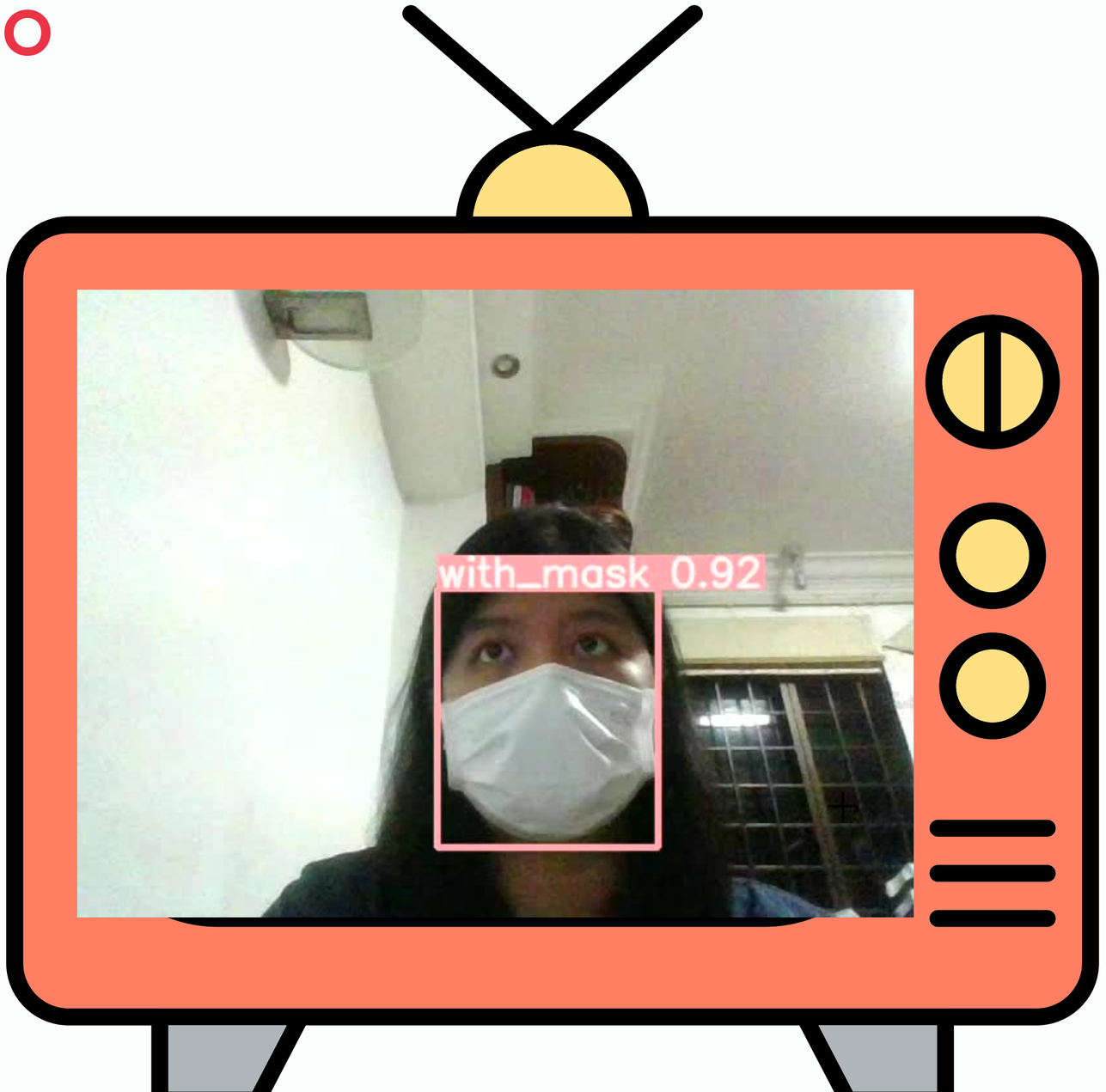




Models	mAP_0.5	m_AP_0.5:0.95
yolov5_416	0.82896	0.49641
yolov5_640_augment	0.97706	0.74817
yolov5_320_augment	0.95884	0.68944
yolov5_640	0.90891	0.57962
yolov5_320	0.87565	0.56838

CONCLUSION

VIDEO DEMO



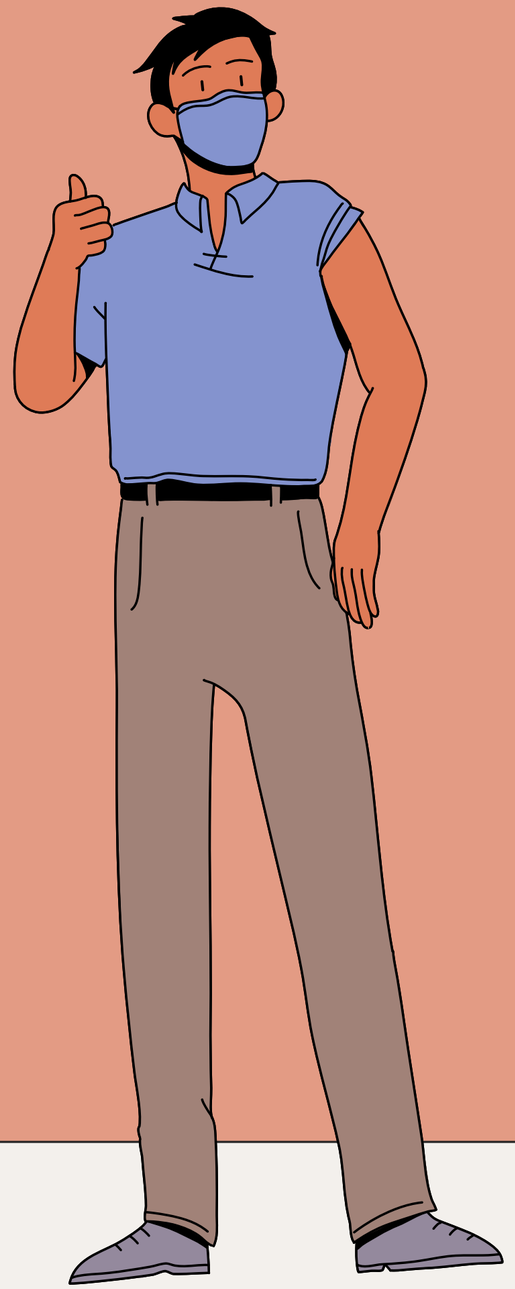
FUTURE WORKS

01 Collect more data

02 Explore other deep learning models such as Faster RCNN

03 Deploy the model

04 Fine tune the features



Thanks!

Any questions?