

Face Mask Detection Using Yolo

Gavin Jin (jj2915@nyu.edu)
Bernice Feng (bf1318@nyu.edu)

Introduction

Problem:

We would like to run an object detection algorithm to detect whether a person is wearing a mask or not. It is especially interesting to detect if a person is wearing mask correctly, since many people tend to wear incorrectly.

Importance:

As the COVID is still around us, mask is becoming a must in some situations. Such a detection algorithm may be helpful for some facilities to ensure a mask-requirement.



Dataset

Data link:

<https://www.kaggle.com/datasets/andrewmvd/face-mask-detection>

Consist of: 853 images with xml annotations belonging to 3 classes

- With mask
- Without mask
- Mask worn incorrectly

Train set: 767 images (~90%)

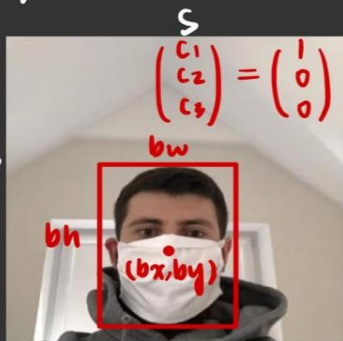
Test set: 86 images (~10%)



```
<annotation>
  <folder>images</folder>
  <filename>maksssskskss0.png</filename>
  <size>
    <width>512</width>
    <height>366</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>without_mask</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
      <xmin>79</xmin>
      <ymin>105</ymin>
      <xmax>109</xmax>
      <ymax>142</ymax>
    </bndbox>
  </object>
  <object>
    <name>with_mask</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
      <xmin>185</xmin>
      <ymin>100</ymin>
      <xmax>226</xmax>
      <ymax>144</ymax>
    </bndbox>
  </object>
  <object>
    <name>without_mask</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
      <xmin>325</xmin>
      <ymin>90</ymin>
      <xmax>360</xmax>
      <ymax>141</ymax>
    </bndbox>
  </object>
</annotation>
```

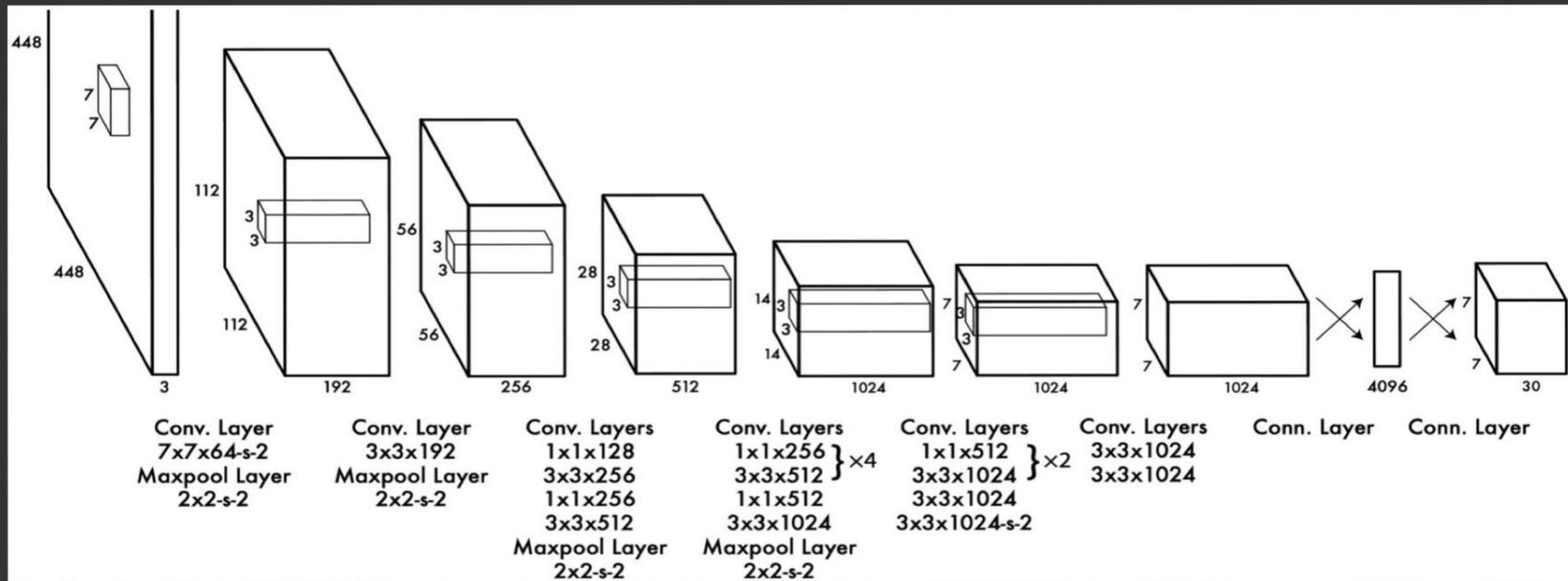
Network

input &
ground
truth =



output:

P_c	← confidence prediction represents IOU
b_x	← x-axis of center of box
b_y	← y-axis of center of box
b_w	← width of box
b_h	← height of box
c_1	← $P_r(\text{class with masks}) \in (0,1)$
c_2	← $P_r(\text{class without masks}) \in (0,1)$
c_3	← $P_r(\text{class wearing mask incorrectly}) \in (0,1)$



Loss Function

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{i,j}^{\text{obj}} (bx_i - \hat{bx}_i)^2 + (by_i - \hat{by}_i)^2 \quad \text{localization loss}$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{i,j}^{\text{obj}} (\sqrt{bw_i} - \sqrt{\hat{bw}_i})^2 + (\sqrt{bh_i} - \sqrt{\hat{bh}_i})^2 \quad \text{confidence loss}$$

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

$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{k \in \text{classes}} (P_i(k) - \hat{P}_i(k)) \quad \text{classification loss}$$

notation: $\mathbb{1}_i^{\text{obj}} = \begin{cases} 1, & \text{object appears in cell } i \\ 0, & \text{otherwise} \end{cases}$

$\mathbb{1}_{i,j}^{\text{obj}} = \begin{cases} 1, & j\text{th boundary box in cell } i \text{ for detection} \\ 0, & \text{otherwise} \end{cases}$

λ_{coord} : increase weight for loss in boundary box

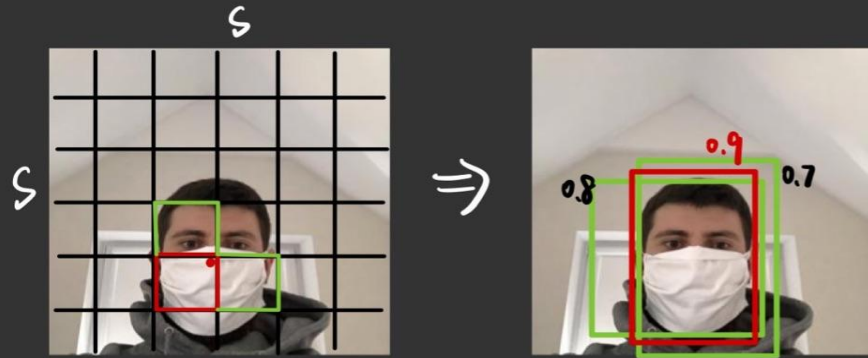
λ_{noobj} : weights down the loss when detecting b.g.

\hat{C}_i : box confidence score of box j in cell i

P_c	← confidence prediction represents IOU
bx	← x-axis of center of box
by	← y-axis of center of box
bw	← width of box
bh	← height of box
C_1	← $\Pr(\text{class with masks}) \in (0,1)$
C_2	← $\Pr(\text{class without masks}) \in (0,1)$
C_3	← $\Pr(\text{class wearing mask incorrectly}) \in (0,1)$

Non-max Suppression (NMS)

- make sure the algorithm detects only once

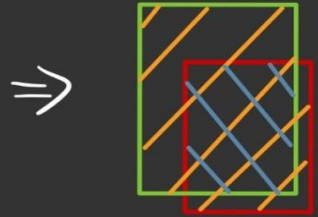


the one in the center

the other mis-identified ones

- ① select the one with highest confidence score
- ② get rid of the rest with high IOU
(if \geq threshold (usually 0.5))

confidence score
 $= \Pr(\text{class}_f) * \text{IOU}_{\text{pred}}^{\text{truth}}$
Intersection over Union (IOU)



$$\text{IOU} = \frac{\text{size of intersection}}{\text{size of union}}$$
$$= \frac{\text{size of } \text{blue square}}{\text{size of } \text{orange square}}$$

Anchor Box

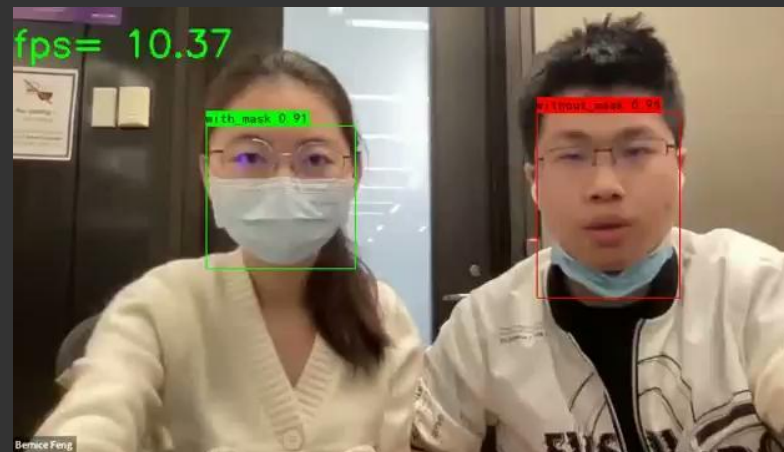
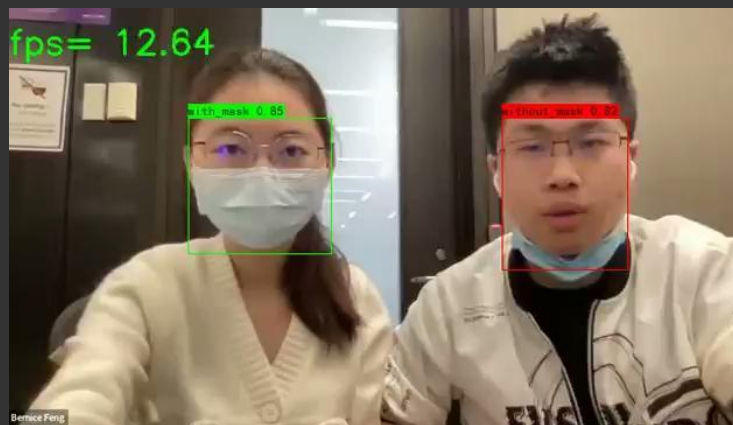
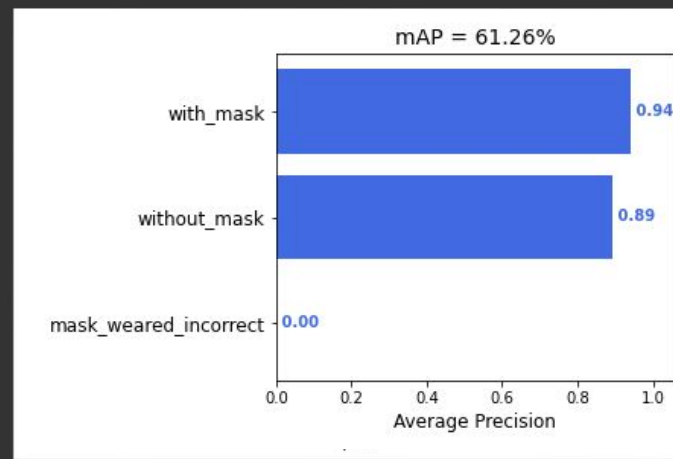
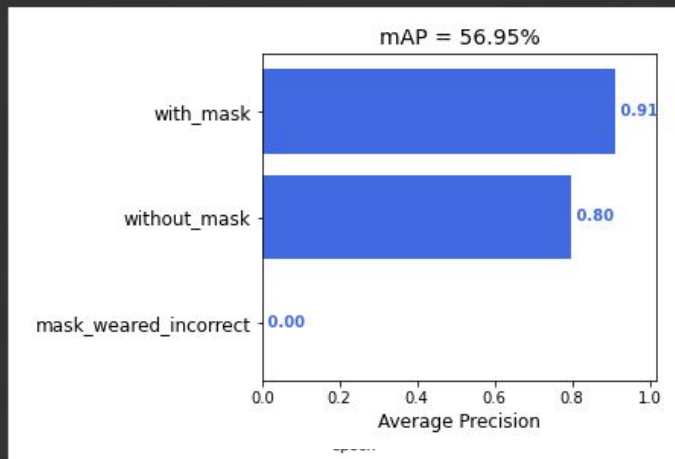
to solve one grid cell wants to detect multiple objects



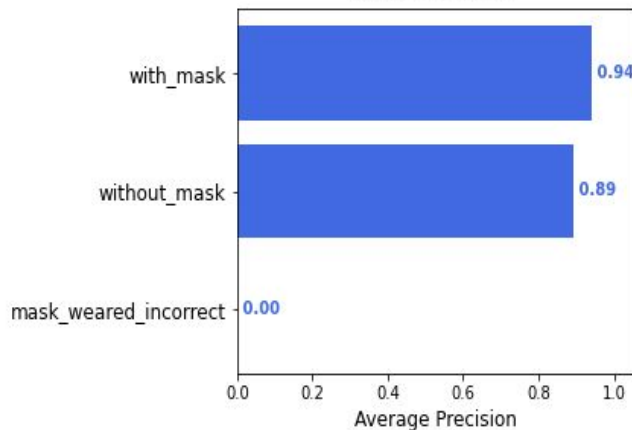
eg: for 2
anchor boxes

$y =$

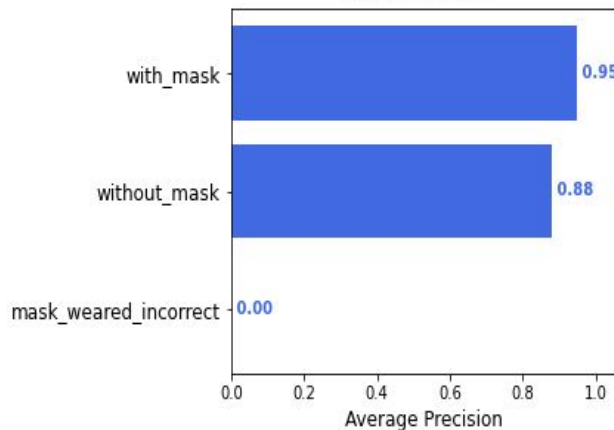
$$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ \vdots \\ c_3 \end{bmatrix}$$



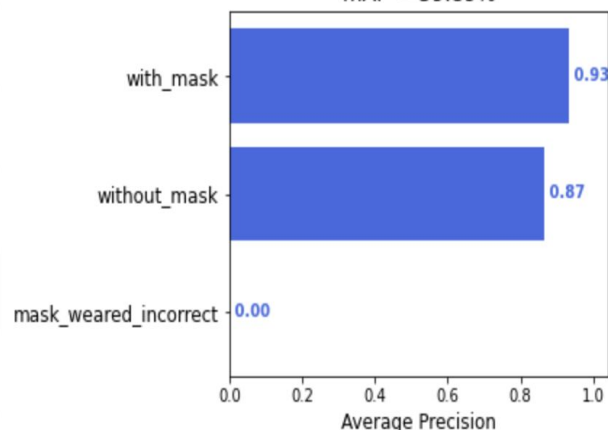
mAP = 61.26%



mAP = 60.86%



mAP = 59.89%



with freeze

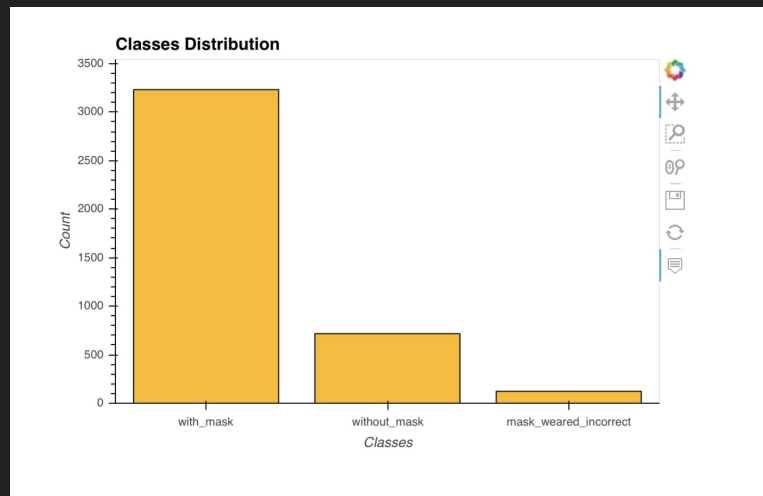
```
Epoch 2/60: 100%|██████████| 43/43 [00:34<00:00, 1.25it/s, loss=5.53, lr=0.0002]
Finish Train
Start Validation
Epoch 2/60: 100%|██████████| 4/4 [00:05<00:00, 1.30s/it, val_loss=4.77]
Finish Validation
Epoch:2/60
Total Loss: 5.531 || Val Loss: 4.766
Start Train
Epoch 3/60: 100%|██████████| 43/43 [00:34<00:00, 1.25it/s, loss=5.02, lr=0.0005]
Finish Train
Start Validation
Epoch 3/60: 100%|██████████| 4/4 [00:02<00:00, 1.34it/s, val_loss=4.69]
Finish Validation
Epoch:3/60
Total Loss: 5.024 || Val Loss: 4.693
Start Train
Epoch 4/60: 100%|██████████| 43/43 [00:34<00:00, 1.24it/s, loss=4.86, lr=0.001]
Finish Train
Start Validation
Epoch 4/60: 100%|██████████| 4/4 [00:02<00:00, 1.54it/s, val_loss=5.05]
Finish Validation
Epoch:4/60
Total Loss: 4.862 || Val Loss: 5.052
```

without freeze

```
Epoch 2/60: 100%|██████████| 86/86 [00:39<00:00, 2.15it/s, loss=3.63, lr=0.0002]
Finish Train
Start Validation
Epoch 2/60: 100%|██████████| 9/9 [00:02<00:00, 3.09it/s, val_loss=3.19]
Finish Validation
Epoch:2/60
Total Loss: 3.626 || Val Loss: 3.190
Start Train
Epoch 3/60: 100%|██████████| 86/86 [00:39<00:00, 2.15it/s, loss=4.01, lr=0.0005]
Finish Train
Start Validation
Epoch 3/60: 100%|██████████| 9/9 [00:02<00:00, 3.26it/s, val_loss=3.88]
Finish Validation
Epoch:3/60
Total Loss: 4.006 || Val Loss: 3.882
Start Train
Epoch 4/60: 100%|██████████| 86/86 [00:40<00:00, 2.14it/s, loss=4.5, lr=0.001]
Finish Train
Start Validation
Epoch 4/60: 100%|██████████| 9/9 [00:02<00:00, 3.26it/s, val_loss=6.31]
Finish Validation
Epoch:4/60
Total Loss: 4.503 || Val Loss: 6.311
```

Future Extensions

1. Our model does not perform well on our dataset, since it does not have a relative even distribution on each class. We could do data augmentation and better split the dataset to make our train/test set less biased.
2. Yolo has several upgraded version, each incorporating new techniques to the original model. We could try it out see how the add-ons increase the performance of our network,



Reference

1. <https://arxiv.org/pdf/1506.02640.pdf>
2. <https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088>