Final Report of Face Mask Recognition

Yize CHENG, Wenbin HU
Hong Kong University of Science of Technology
ychengbt@connect.ust.hk, whuak@connect.ust.hk

Abstract

Wearing face masks has been proven highly effective in preventing the spread of Covid-19. Hence, face mask detection has become an emerging object detection task since the outbreak of the pandemic. However, there are few works focusing on detecting the correctness of the way masks are worn and the suitability of the choice of masks' category. With the extensive usage of CCTV surveillance cameras, abundant image data of people wearing or not wearing masks of different categories in different manner can be collected in real time. In this project, we first collected images of people wearing different types of face masks to build our dataset, which contains 6500 images, containing of faces in 5 categories. We adopted deep-learning-based techniques to detect and classify faces in real time according to whether they are properly wearing face masks, and detailed to what type of mask is worn if one is wearing a mask in a proper manner. To be more specific, we experimented with the Faster-RCNN model with different baseline CNN structures and different ROI selection criterion, as well as the Yolov5 model , and eventually reached an mAP value of 0.975 with FPS of 109.8.

1. Introduction

Since the outbreak of Covid-19, millions of people worldwide has been infected by this highly contagious virus, which also exhibits a higher fatality rate comparing to normal influenza. Studies have proven that properly wearing face masks is highly effective in protecting individuals from either spreading or being infected by the virus [2]. It has also been illustrated that different types of masks can lead to differentiated protection effectiveness under various circumstances [6, 12]. For instance, figure 1 from Gurbaxni's research demonstrated a massive difference between the effectiveness of different masks.

Despite its effectiveness, only when chosen properly and worn correctly can a face mask's protectiveness be fulling made use of. However, many people display improper manners of wearing their face masks, such as not fully covering the nose, or even just hanging it over the neck and leaving the whole face exposed to the air. And due to the variance between different masks in terms of protection effectiveness, choosing the correct type of mask is also a significant factor when evaluating individuals' anti-epidemic behaviours.

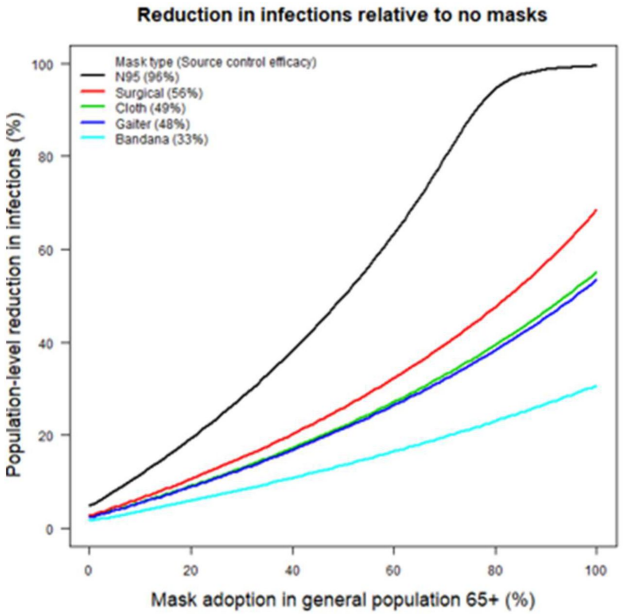

Figure 1. protectiveness of different types of mask [6].

In this project, we adopted deep-learning-based techniques to detect and classify faces according to whether they are properly wearing face masks, and detailed to what type of mask is worn if one is wearing a mask in a proper manner.

In the beginning, we started by choosing some face mask related datasets, and did manual labeling to obtain the bounding box coordinates and suit our own classification objectives. Initially we planed to categorize faces into 7 types – "no mask" and three categories of masks each for "proper manner" and "improper manner". But we soon

realized that masks improperly worn, regardless of exact type, show minimal protectiveness in reality. So it would be more practical to categorize "improper masks" as one class. Hence the total number of classes is reduced to 5. Then, we trained Faster-RCNN [16] with VGG16 [18], ResNet50 [7] and a modified ResNet50 structure as baseline CNN model to do object detection. We also trained the latest Yolov5 [8] on the same training set and compared their performance. Beyond which, we experimented with a new metric for measuring the similarity of two boxes to replace the traditional IOU score. Eventually we reached mAP of 0.997 and FPS of 1.302 in the best Faster-RCNN settings, and mAP of 0.975 and FPS reaching an astonishing 109.8 in Yolov5 settings.

2. Related Works

Face Detection + Face Classification. Face detection algorithms are quite well studied, hence using a face detector to locate faces in the image input and then passing the located area to a separate CNN classifier is a simple and intuitive way for face mask recognition. Goyal, Hiten, *et al* [5] used the Single Shot Detector (SSD) [11] to detect faces from the input image, and trained a separate CNN classifier to distinguish "faces with mask" to "faces without mask". Their approach is illustrated in figure 2. Despite

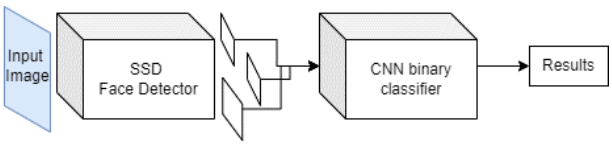

Figure 2. SSD Face Detector + CNN binary classifier

this was a vanilla approach, it already achieved a decent performance. Lin, Hong, *et al* [10] even went beyond this to build a segmentation model to identify some key points in the face region, which will be extracted from the original image to pass through a Face Mask Recognition Network(FMRN), which is a separate CNN classification network, to do binary classification. They referred to this process as "ROI extraction", which is essentially improving the quality of the region to be passed to the CNN classifier. The problem of these kind of approaches is that the network cannot be trained as a whole from end to end, since the task is divided into two stages, *i.e.* face extraction and classification. Also it cannot recognize improper masks effectively, since this scenario is sort of the intermediate stage between "face with mask" and "face without mask". In our project, we will train a network as a whole to simplify the task and the training process.

Real Time Detection with Yolo. Since the idea of "You Only Look Once"(Yolo) [14] came out in 2016, the Yolo family has almost become the standard approach towards real time recognition problems. Soon afterwards, different versions came out one after another. Susanto, Susanto, *et al* [20] built a face mask detection device at Politeknik Negeri Batam using a rather sophisticated model. They first use Yolov3 [15] to do dense prediction, and pass the extracted regions to a following Faster-RCNN to do sparse prediction. Their structure is shown in figure 3. G. Yang *et*

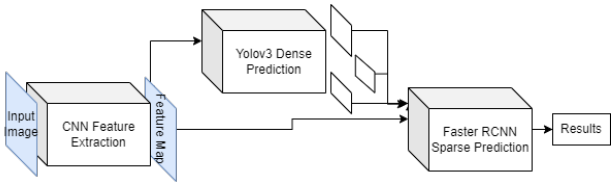

Figure 3. Yolov3 dense prediction + FasterRCNN sparse prediction

al [24] made use of the latest Yolov5 to build a guest pass system at the entrance of a shopping mall to ensure only those who are wearing face masks are granted access. They achieved real time detection with a reported accuracy of 97.9%. However, similar to other previous works, as mentioned in the "future work" session of their published paper, they struggle to well identify masks worn in an improper manner. In our project we will experiment with Yolov5 in a similar approach, but with our custom classification objectives to remedy the ambiguity for improper masks.

3. Dataset

3.1. Overview

To approach this task, abundant data is needed for the training and testing of the model. We first collected two public face mask datasets from Kaggle [4, 13] and one dataset proposed by an existing research [19]. These datasets were merged together to feed into our model. Some examples are shown in figure 4. To be more specific, we

Figure 4. examples of proper surgical, proper N95, proper cloth, improper worn mask, and faces without mask [13] [19]

in total collected 6583 images, with 5924 out of which for

training, 462 for validation, and 197 for testing. The distribution of data between all 5 classes can be found in figure 5.

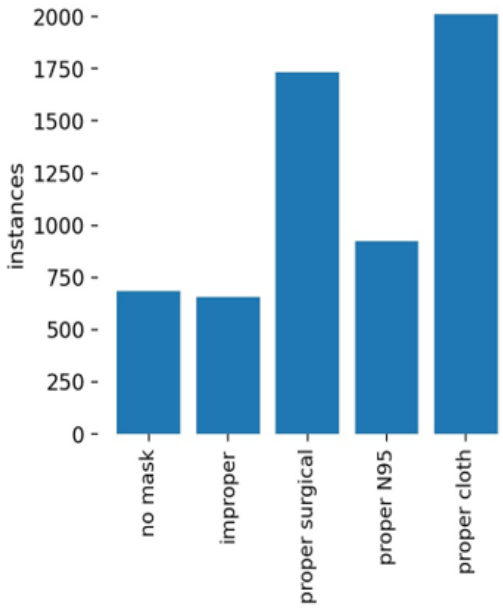

Figure 5. data distribution between 5 classes

The data is distributed among the 5 classes approximately in a $1:1:1:2.5:1:1.5$ ratio.

3.2. Manual Labeling

The original labels of the dataset we collected either met our classification objectives nor provide bounding box information. Therefore, manual labeling was inevitable. To help speed up the labeling efficiency, we took advantage of a labeling software *LabelImg* [22], whose usage is illustrated in figure 6, to help us obtain bounding box coordinates in

Figure 6. usage of the *LabelImg* software

order to have fully supervised training.

3.3. Preprocessing

Data Augmentation. Among the images that we collected, some of them has already been added with noise or blurring effect, or rotated by some degree to achieve certain augmentation. In the Faster-RCNN data loading process, random flips were further added to improve network robustness. And Yolov5, same as Yolov4 [3], adopts Mosaic data augmentation, which combines 4 different images into one according to certain ratios.

Warping. Both Faster-RCNN and Yolov5 have certain input image size requirement. But the warping size warping issue was handled by the code itself, hence no additional warping was performed on the data.

4. Methods

We approached this task using both the Faster-RCNN [16] and Yolov5 [8] model and compared their performance. The overall structure is shown in figure 7 and figure 11 respectively. Below explicitly describes some detailed information.

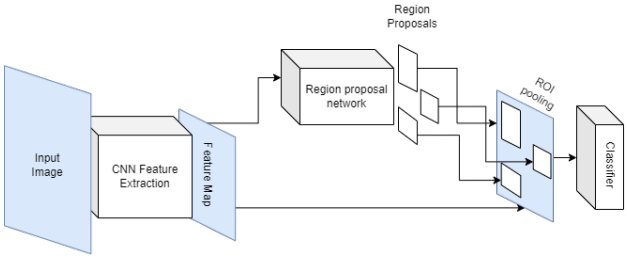

Figure 7. overall structure of Faster-RCNN

4.1. Faster-RCNN

The Faster-RCNN is the third generation of the RCNN object detection algorithm family. The key steps include feature extraction, region of interest proposal and dense layer classification. Among which, feature extraction from a CNN and region proposal by the region proposal network are the two crucial steps. We compared results of different CNN structures used for feature extraction and applied different region proposal selection criterion on the proposed regions.

4.1.1 CNN Models

We compared the results of using VGG16 [18], ResNet50 [7], and a modified ResNet50 for feature extraction.

VGG16. Having won the ImageNet [17] 2014 challenge, the idea of replacing conventional larger convolution kernels by smaller 3×3 kernels, proposed by Simonyan and Zisserman [18] in 2014 not only increased the deepness of the network for learning more complex features, but also cost less parameters to improve efficiency. We chose

a VGG16 model that was pretrained on ImageNet as the baseline feature extractor for the Faster-RCNN detector. Our new thinking regarding region proposal selection criterion was tested using the VGG16 as well.

ResNet50. As deep learning network become deeper and deeper, it also brings greater difficulties for the training process, such as degradation problems. He, Zhang *et al* [7] proposed residual connections in 2015 to reform the layers as learnable residual functions with reference to the inputs, *i.e.*

$$y(x) = f(x) + x$$

and compact into Identity blocks, which brought about great performance improvement. The performance of using ResNet50 as the feature extractor was chosen as one of the contrast experiment subjects as well.

Modified ResNet50. Inspired by how powerful residual connections bring, we modified the residual connections to further explore its effect. Instead of putting residual connections within each identity block separately, additional connections are added across different identity blocks, such that smaller residual connections are embedded in bigger ones. The idea is illustrated in figure 8. We hope this stronger skip connection structure can bring stronger feature re-usability. Originally we wanted to experiment this idea on ResNext [23], whose pre-trained weights, unfortunately, cannot be found. Such a CNN structure was the third

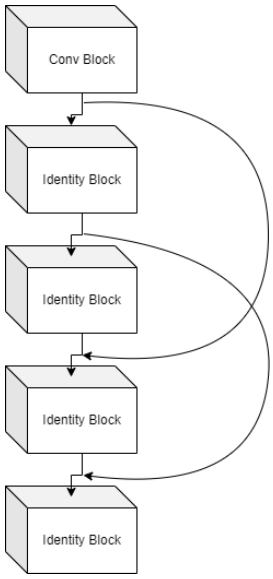

Figure 8. A modified version of ResNet

experiment subject for the feature extraction stage.

4.1.2 Region Proposal Selection Criterion

For models of the RCNN family, filtering the regions proposed from either a heuristic approach or the region proposal network(RPN) is a crucial step for accurate detection. The selection and filtering criterion is usually based on the intersection over union(IoU), *i.e.*

$$IOU = \frac{overlapping\ area}{union\ area}$$

However, since this is a pure area based metric, there can be scenarios where two overlapping situation leads to the same IoU score, while one of them is clearly a closer match. Therefore, we attempted with a distance based coincidence degree measurement

$$degree = e^{k(A-B-\epsilon)}$$

, where the definition of A, B and ϵ is illustrated in figure 9, and k is a hyper parameter that depends on the expected size of bounding boxes. It's quite obvious that the power is a value between $-\infty$ to 0, hence the overall value is between 0 and 1. The new $degree$ function takes the coordinate relationship into consideration so that we don't solely focus on the proportion of overlapping area, but actually measure the similarity in terms of distance. Also, this function allows us to make use of the characteristic of the exponential function $y = e^x$ that it only increases faster when it approaches 0 from $-\infty$, and grows rather slowly when it's far. This helps to eliminate many inferior regions, since the score only grows when the distance is close enough to 0, where the growing rate can be controlled by the parameter k.

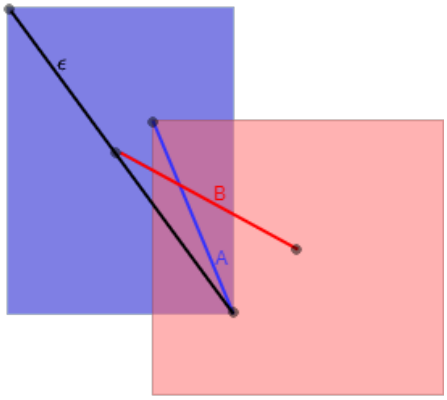

Figure 9. illustration of the "degree" metric: A = distance between two overlapping corners($-\infty$ if there is no overlap), B = distance between two box centers, ϵ is the diagonal length of the GT bounding box, k is a hyper parameter

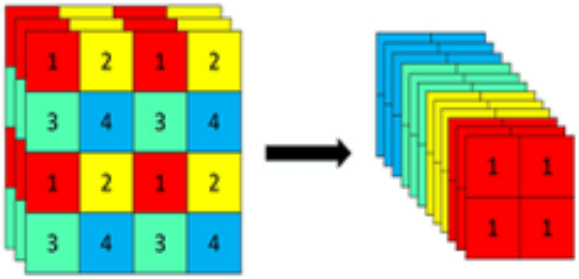

Figure 10. Illustration of Focus Layer [1]

4.2. Yolov5

We adopt a yolov5 model implemented by Ultralytics [8] and modify it for our facial mask recognition task. It uses FPN+PAN structure as its backbone and implements the model with basic components with skip connections inside. The additional feature helps detecting different scales of masks and makes the model more robust. The model and training process are illustrated in figure 11.

4.2.1 Feature pyramid

Feature pyramids architectures are commonly used in recognition and segmentation tasks nowadays. By aggregating signals from different scales, it can detect objects of different scales. Comparing to the original Yolo [14] model, which only adopts a simple GoogleNet [21], Yolov5 is more robust for our mask recognition task to detect different scales of masks. Yolov5 used PANet to construct a feature pyramid structure, which adds a bottom-up path aggregation strengthening performance.

4.2.2 Focus Layer

Focus layer is an innovative idea in yolov5. It samples every other one pixel, concatenates output feature maps for CNN process. It downsamples the feature map with losing information. The idea is demonstrated in figure 10.

4.2.3 Basic Components

Yolov5 is composed of a series of basic components. CBL is a standard CNN block. CSP, ResUnit utilize the idea of skip connection to aggregate information from both high levels and low levels and avoid gradient vanishing.

4.2.4 Loss Function

In yolov5, we use a compound loss composed with 3 parts: objectness loss, classification loss, bounding box regression loss. The regression loss is CIoU loss, classification loss(L_{CLA}) and objectness loss(L_{OBJ}) are binary cross entropy loss (BCE). Suppose the image is divided into an $S \times S$ grid, and the number of anchor boxes in each grid is B, and the confidence score of the j^{th} anchor box at grid position i is denoted as C_{ij}, the detailed expression for each loss function is shown in equation (1) to (6).

$$L_{OBJ} = \sum_{i \in S \times S} \sum_{j \in B} [C_{ij} \log C_{ij} + (1 - C_{ij}) \log (1 - C_{ij})] \tag{1}$$

$$L_{CLA} = \sum_{i \in S \times S} \sum_{j \in B} \sum_{c \in classes} [P_{ij}(c) \log P_{ij}(c) + (1 - P_{ij}(c)) \log (1 - P_{ij}(c))] \tag{2}$$

$$CIoULoss = 1 - CIoU \tag{3}$$

$$CIoU = IoU - \frac{\rho^2}{c^2} + \alpha v \tag{4}$$

$$\alpha = \frac{v}{1 - IoU + v} \tag{5}$$

$$v = \frac{\pi^2}{4} (\arctan(\frac{w^g}{h^g}) - \arctan(\frac{w^p}{h^p})) \tag{6}$$

where ρ is the Euclidean distance between the center points of the anchor box and ground truth, c is the diagonal distance of the overlapping area, and w and h are the width and height for the boxes, with superscript g and p representing ground truth and predicted boxes respectively.

$$Loss_{total} = CIouLoss + L_{OBJ} + L_{CLA}$$

5. Experiments

5.1. Contrast Experiment

We conducted contrast experiments on different choice of CNN models for Faster-RCNN. And we also compared the performance of the best Faster-RCNN result with that of Yolov5.

All Faster-RCNN models were trained for 40 epochs with a learning rate of 1×10^{-5}. It was found out that setting the k value of degree function as 0.1 was suitiable for our dataset. For easier transfer learning, our training were based on pre-trained model weights from ImageNet. Adam optimizer was chosen to facilitate faster and smoother optimization. We trained on Google Colab with NVIDA Tesla T4 GPU support which has a memory size up to 16GB. Excluding the data parsing time, the total training time for 40 epochs is approximately between 3.5 to 4 hours.

We adopt mean average precision(mAP) and frame per second(FPS) as dependent variables of the experiment, in

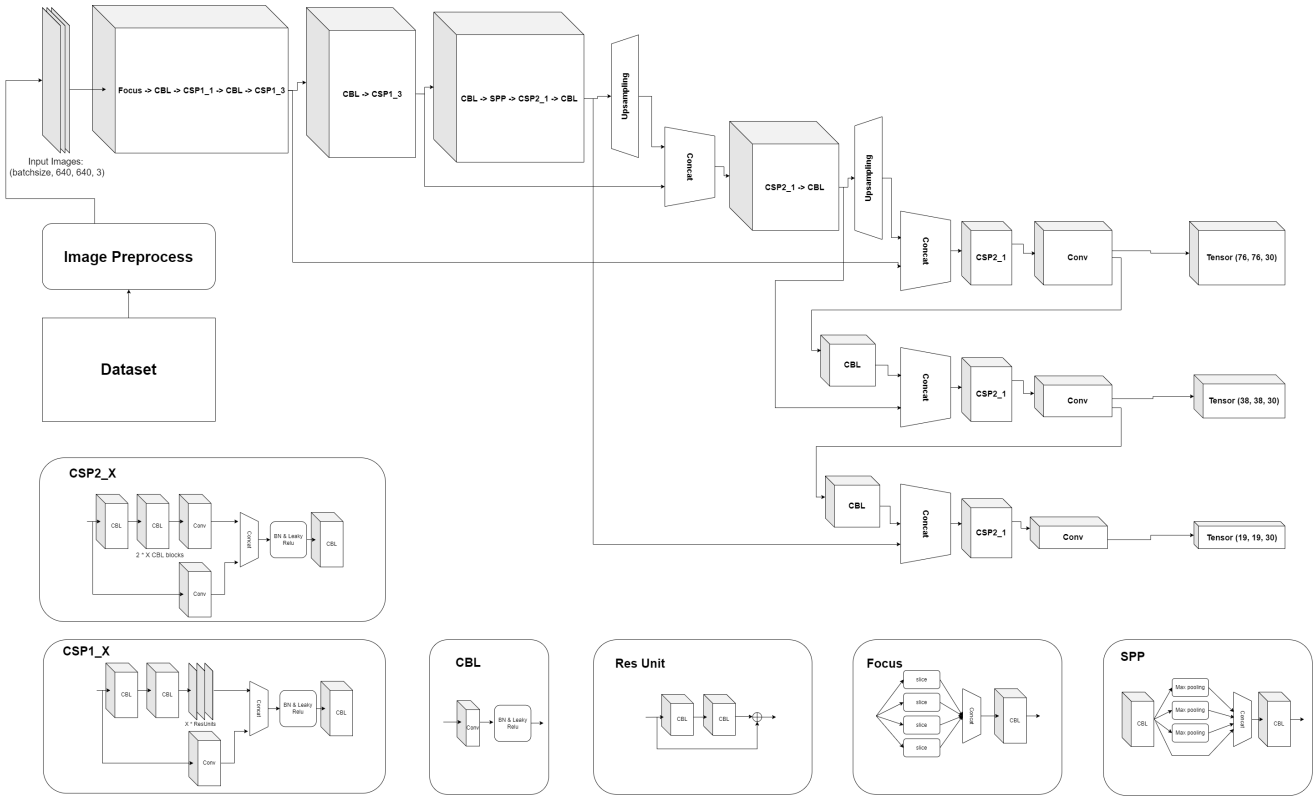

Figure 11. The Yolov5 model from ultralytics's team [8]

other words, our evaluation metric. Here, mAP measures the accuracy by

$$mAP = \frac{\sum_{i=1}^{C} AP_i}{C}$$

, where C is the number of classes, and given that r_i is the recall value at the i^th interpolation point,

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) P_{interpolated}(r_{i+1})$$

, which is the area under the interpolated Precision-Recall curve. And FPS measures the efficiency and speed of the prediction process. The detailed experimental results are shown in table 1. **Note that the IoU score is a part of the calculation of the mAP score, for the sake of fair comparison, we adopted the conventional IoU formula when calculating the mAP scores for VGG16 with** $degree$ **based box filtering.**

Here we summarize some notable observations. First, Yolov5 shows a significantly higher detection efficiency compared to any of the Faster-RCNN settings. It can be easily seen that the FPS of all Faster-RCNN models lie around 1 2.5, yet Yolov5 achieves an astonishing FPS value up to 109.8. Second, for all Faster-RCNN models, the FPS tends to drop as more and more epochs are trained. We believe a possible reason for this is that with further training, the region proposal network is able to propose more and more high quality anchor boxes, and hence fewer boxes were eliminated before ROI pooling and hence the classification layers have more ROIs to process. Third, our modified ResNet50 achieved almost identical performance as the original ResNet50, we think this is because the existing skip connections have already utilized the feature re-usability to a saturation level, that it's difficult for additional skip connections to improve. Fourth, since we adopted the conventional IoU formula when calculating the mAP scores for VGG16 with $degree$ based box filtering, it's not that meaningful to discuss the accuracy comparison between normal VGG and the one with $degree$ based box filtering. Since the normal VGG was trained to meet the conventional IoU metric while the other one was not. However, as mentioned, for all Faster-RCNN models, the FPS tends to drop as more and more epochs are trained. Yet this phenomenon is way less notable in the case of using $degree$ based box filtering. The FPS is still dropping, but in a way slower dropping rate. We speculate that this is due to the stricter selection criterion induced by the $degree$ function that there won't be a substantial decrease in the number of region proposals eliminated before passing into the classification layer, relieving

Table 1. Performance of Faster-RCNN under different settings on the test set, independent variables include choice of feature extractor and region proposal selection criterion

Model	VGG16		ResNet50		Modified ResNet50		VGG16 with *degree* based box filtering		Yolov5	
Epoch	mAP	FPS	mAP	FPS	mAP	FPS	mAP	FPS	mAP	FPS
5	0.859	2.441	0.974	2.033	0.866	1.978	0.864	1.901	0.566	-
10	0.839	2.475	0.968	2.061	0.933	1.602	0.913	1.908	0.845	-
15	0.925	2.412	0.992	1.842	0.962	1.446	0.919	1.805	0.877	-
20	0.954	2.525	0.984	1.968	0.812	1.329	0.867	1.715	0.912	-
25	0.933	2.421	0.972	1.633	0.983	1.383	0.953	1.828	0.957	-
30	0.964	2.387	0.988	1.415	0.942	1.251	0.920	1.923	0.952	-
35	0.965	2.409	0.984	1.435	0.968	1.261	0.927	1.890	0.972	-
40	0.978	1.984	0.997	1.302	0.985	1.303	0.953	1.801	0.975	109.8

the computation pressure of the classification layers.

The training curve of Yolov5 is shown in figure 12. Some sucessful detection results of Faster-RCNN and Yolov5 are presented in figure 13.

5.2. Failure Modes

Here we summarize some common failure modes.

Firstly, failing to identify faces in the distance. Sometimes when the face in the input image is far away, or in other words, appears to be small, the model sometimes can fail to detect its existence. We believe a major reason for such kind of failure lies in the limitation of our dataset. Many of our image data were originally designed for image classification tasks, and therefore, the faces often occupies a large proportion of the image. Therefore, the model may have only been trained strong enough to deal with "large faces". We believe this issue can be better solved if the training dataset consisted of more dense population with faces in the distance wearing masks.

Secondly, the confusion between "improper masks" and "no mask". We show a representative wrong detection result in figure 14. Just opposite to the problem that G. Yang *et al* [24] encountered in their research where they find it difficult to distinguish "improper masks" to "face with mask", we sometimes encounter a scenario where the model can detect a face as both "no mask" and "improper" at the same time, just like in figure 14, where the person hangs her mask on the neck, and the model boxes the face area without the hanging mask as "no mask" and the larger picture with a visible hanging mask as "improper". This remians a problem that requires further research to resolve.

6. Conclusion

In this project, we first collected and manually labeled some datasets for face mask recognition. Then we experimented with and compared multiple object recognition frameworks to approach this task. We also attempted to propose new structures and metric functions, some of which made a difference while some didn't. This process helped us to gain a deeper insight of different CNN architectures and further understand the effect of each component. As discussed in section 5.2, despite our overall detection accuracy was acceptable, there still lies unsolved issues. If there were future steps, we would want to first expand the training dataset to include faces that appear in the distance, we might also utilize some efficient neural networks to make our detection closer to real time. If we were able to reach a more stable detection accuracy and higher FPS, such a model is suitable in all kinds of guest-pass systems under this pandemic situation.

7. Acknowledgements

Two GitHub open source implementations were used as base for our project. They are Yolov5 from the Ultralytics [8] team and the keras implementation of Faster-RCNN by kbardool [9].

References

[1] Venkat Anil Adibhatla, Huan-Chuang Chih, Chi-Chang Hsu, Joseph Cheng, Maysam F. Abbod, and Jiann-Shing Shieh. Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once. *Mathematical Biosciences and Engineering*, 18(4):4411–4428, 2021. 5

[2] Kristin L Andrejko, Jake M Pry, Jennifer F Myers, Nozomi Fukui, Jennifer L DeGuzman, John Openshaw, James P Watt, Joseph A Lewnard, Seema Jain, California COVID, et al. Effectiveness of face mask or respirator use in indoor public settings for prevention of sars-cov-2 infection—california, february–december 2021. *Morbidity and Mortality Weekly Report*, 71(6):212, 2022. 1

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3

[4] Geekfx. Casia webmaskedface, Apr 2022. 2

[5] Hiten Goyal, Karanveer Sidana, Charanjeet Singh, Abhilasha Jain, and Swati Jindal. A real time face mask detec-

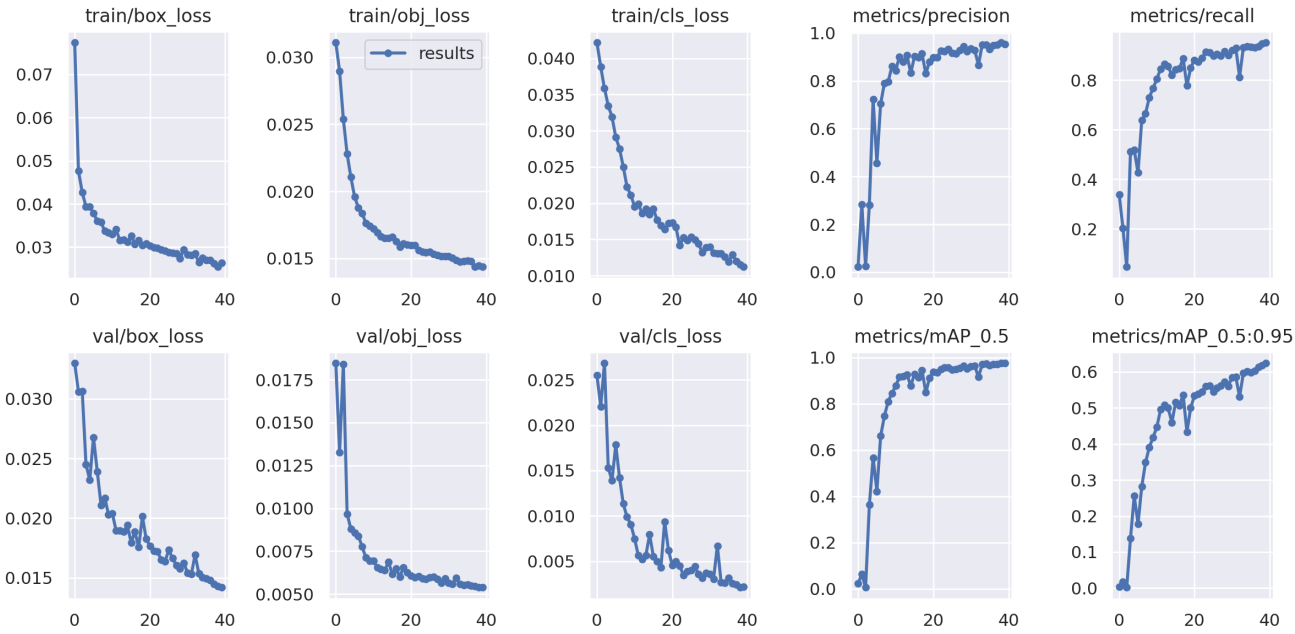

Figure 12. yolov5 training curve

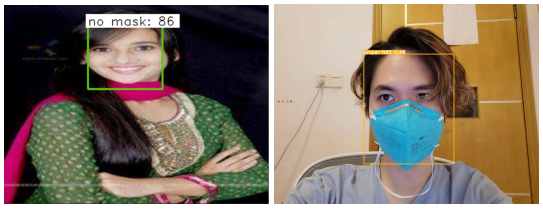

Figure 13. successful examples

Figure 14. A representative failure scenario

tion system using convolutional neural network. *Multimedia Tools and Applications*, pages 1–17, 2022. 2

[6] Brian M Gurbaxani, Andrew N Hill, Prabasaj Paul, Pragati V Prasad, and Rachel B Slayton. Evaluation of different types of face masks to limit the spread of sars-cov-2–a modeling study. 2021. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 3, 4

[8] Glenn Jocher. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements. https://github.com/ultralytics/yolov5, Oct. 2020. 2, 3, 5, 6, 7

[9] Kbardool. Kbardool/keras-frcnn: Keras implementation of faster r-cnn. 7

[10] Hong Lin, Rita Tse, Su-Kit Tang, Yanbing Chen, Wei Ke, and Giovanni Pau. Near-realtime face mask wearing recognition based on deep learning. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–7. IEEE, 2021. 2

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[12] C Raina MacIntyre, Holly Seale, Tham Chi Dung, Nguyen Tran Hien, Phan Thi Nga, Abrar Ahmad Chughtai, Bayzidur Rahman, Dominic E Dwyer, and Quanyi Wang. A cluster randomised trial of cloth masks compared with medical masks in healthcare workers. *BMJ open*, 5(4):e006577, 2015. 1

[13] PATNAIK. Face mask detector(mask ,not mask, incorrect mask), Apr 2022. 2

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2, 5

[15] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region

proposal networks. *Advances in neural information processing systems*, 28, 2015. 2, 3

[17] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 3

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3

[19] Xueping Su, Meng Gao, Jie Ren, Yunhong Li, Mian Dong, and Xi Liu. Face mask detection and classification via deep transfer learning. *Multimedia Tools and Applications*, pages 1–20, 2021. 2

[20] Susanto Susanto, Febri Alwan Putra, Riska Analia, and Ika Karlina Laila Nur Suciningtyas. The face mask detection for preventing the spread of covid-19 at politeknik negeri batam. In *2020 3rd International Conference on Applied Engineering (ICAE)*, pages 1–5. IEEE, 2020. 2

[21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 5

[22] Tzutalin. Tzutalin/labelimg: Labelimg is a graphical image annotation tool and label object bounding boxes in images. https://github.com/tzutalin/labelImg. 3

[23] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 4

[24] Guanhao Yang, Wei Feng, Jintao Jin, Qujiang Lei, Xiuhao Li, Guangchao Gui, and Weijun Wang. Face mask recognition system with yolov5 based on image recognition. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pages 1398–1404, Dec 2020. 2, 7