FAST REAL TIME FACE MASK CORRECTNESS DETECTION USING SINGLE SHOT DETECTION

HOE JIUN TIAN

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

FAST REAL TIME FACE MASK CORRECTNESS DETECTION USING SINGLE SHOT DETECTION

HOE JIUN TIAN

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE COURSE OF WIX3001 SOFT COMPUTING

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

UNIVERSITY OF MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Hoe Jiun Tian (I.C/Passport No: 990608-07-5269)

Matric No: 17137385/1

Name of Degree: Bachelor of Computer Science (Artificial Intelligent)

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Fast Real Time Face Mask Correctness Detection using Single Shot Detection

Field of Study: Deep Learning, Object Detection

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date: 13 JUN 2021

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

ABSTRACT

Wearing face masks is one of the important steps to restrict the spread of the pandemic virus. Thus, it is important to ensure that everyone wear their face mask and wear it in correct way. In this report, we use deep learning object detection algorithm to perform face masks detection. Our work is able to detect the face masks efficiently, with the mean average precision of 80% in average on three category, which are: wearing face mask correctly, wearing face mask incorrectly, and did not wear face mask. We manage to create a real time detection which can inference at 85 frame per seconds on just a single GPU. In this report, we also show the sample input and output using our own data.

Keywords: deep learning, object detection, covid-19.

ACKNOWLEDGEMENTS

We would like to express the gratitude to the course instructor, Dr Woo Chaw Seng, for his guidance and encouragement in finishing this assignment and also for teaching us in this course.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

The COVID-19 is spreading over the world, and has been declared as a pandemic. The world is taking various actions in order to stop the spreading of this virus, this include but not limited to social distancing, vaccination, lockdowns, and self-protections such as facial masks. Wearing face masks is one of the most effective ways to limit the spread of the virus, especially in public spaces.

During the COVID-19 pandemic, the use of facemasks by healthy individuals, which is a personal protective measure, has been attracting a lot of public attention. A recent review reported that wearing masks could possibly prevent the transmission of COVID-19 (Chu et al 2020). Certain researchers have pointed out that wearing face masks in public spaces, including by those who are infected but are asymptomatic and contagious, may be effective in preventing the transmission of the virus (Feng et al 2020, Leung et al 2020) . In many countries, facial masks are required to be worn at all times at all public places, such as restaurants, schools, et cetera.

However, there came another challenge, most of the people did not know how to wear the masks correctly. For example, some of them wear the mask without covering their nose under the masks. It is difficult for authorities to keep track of all the crowd in the public spaces, as the work will be too much out of their ability.

Moreover, there are also a lot of them in the crowd who did not wear face masks. It is difficult to detect all of them manually as the crowds are too much. There comes a strong need for a soft computing based artificial intelligent algorithm, to detect all these people who wear the face masks incorrectly, or even worse, do not wear the face masks at all. A machine based method can quickly identify all the incorrectly worn face masks in the crowdly public space, and the authority can react quickly to them, compared to manual identification which takes time and effort of the authorities.

1.1 Objective

This project is initiated with two objectives, which are not just to create an algorithm that can detect faces and classify whether they wear the masks correctly, but also evaluate the performance of the algorithm using the mean average precision, to ensure the algorithm is good in detecting and classifying whether the masks is wore correctly for detected faces.

1.1.1 To detect faces and classify whether they wearing face masks correctly

We want a detection algorithm that is fast enough (which can run real time inference with a video camera with at least 30fps) to detect faces in a crowd and classify whether the persons in the crowd wear the face masks correctly.

1.1.2 To evaluate the mean average precision of detection and classification for detected faces

In order to ensure the good performance of the model, we use mean average precision (mAP) as our metric for performance measure. We measure mAP@0.5 IoU as the performance metric, where IoU means for Intersection over Union. 0.5IoU is chosen because we want the detection to be able to correctly detect the faces, and we only count the mAP for the detection with over 0.5 IoU.

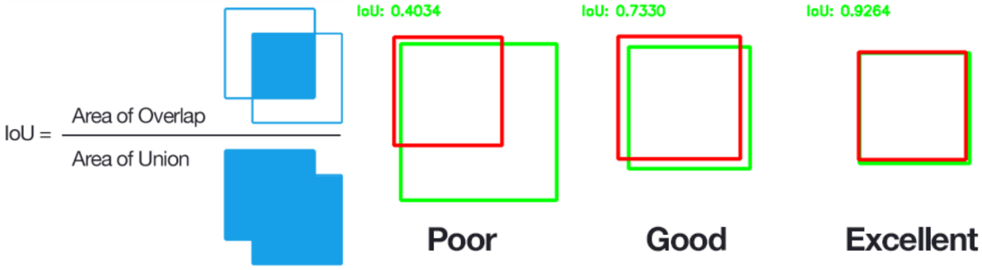

Figure 1.1 Intersection over Union (Rosebrock, 2016)

The formula of mean average precision is defined as follows. mAP is the mean of Average Precision over all classes.

$$\text{Precision} = \frac{\text{True Positive}}{True\ Positive + False\ Positive}$$

$$mAP = \frac{1}{|classes|} \sum_{c \in classes} \frac{\#TP(c)}{\#TP(c) + \#FP(c)}$$

1.2 Data Source

In order to train this model, we use the Mask Dataset provided by MakeML. The dataset is available in PascalVOC format and labelled with bounding box annotations. There are three classes in this dataset, which are: with mask, without mask, and incorrectly worn mask, which is suitable for our requirement to develop a model to detect incorrectly worn masks. The dataset contains 853 images, and the license is in Public Domain, so we can legally use this for any purpose. The data source can be obtained by using the official link provided by MakeML. https://makeml.app/datasets/mask.

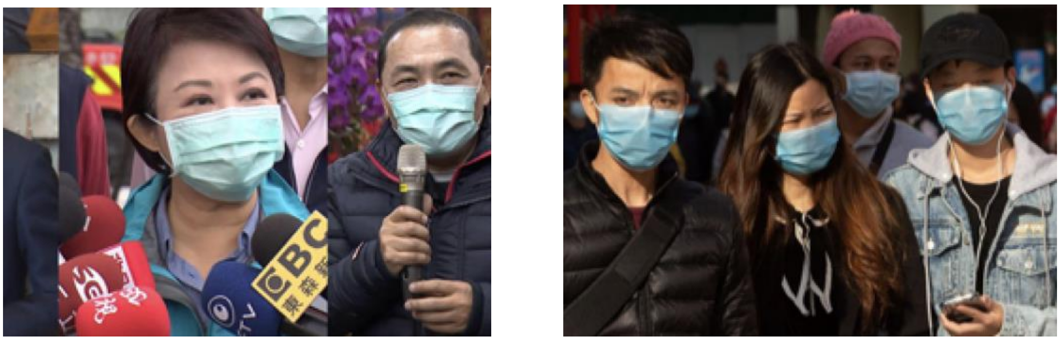

Figure 1.2 Samples from the dataset

CHAPTER 2: LITERATURE REVIEW

Face detection is one of the research fields in object detection. Object detection is a computer vision study which allows us to identify and locate the object in the image. With this kind of identification and localization, it can be used to determine the object and their precise location. Object detection typically uses Convolutional Neural Network as backbone to detect the object and classify them. There are two different types of modern object detection, which are region proposal based framework and regression based framework.

2.1 Region Based Framework

2.1.1 Region Convolutional Neural Network (R-CNN)

RCNN (Girshick et al, 2013) used selective search to extract just 2000 regions from the image and called them region proposals. Instead of trying to classify a huge number of regions, it only has to work with 2000 regions. These region are warped into a square and fed into a convolutional neural network that produces a 4096-dimensional feature vector as output. The CNN acts as a feature extractor and the output dense layer consists of the features extracted from the image and the extracted features are fed into an SVM to classify the presence of the object within that candidate region proposal.

This method has the 3 main disadvantage. The learning didn’t happen on the selective search stage, as the selective search algorithm is fixed. This method takes a long time to train as it needs to classify on 2000 region proposals per image. It cannot be used in real time application, as it takes ~47 seconds to infer an image.

2.1.2 Fast Region-based Convolutional Network (Fast R-CNN)

In Fast R-CNN (Girshick, 2015) solved the drawback of R-CNN which he proposed in 2013. R-CNN feeds the input image to the CNN to generate a convolutional feature map to identify the region of proposals and warp them into squares and by using a RoI pooling layer. It is then reshaped into a fixed size so that it can feed into a fully connected layer. Softmax layer is used to predict the class of the proposed region and also the offset values for the bounding box. In Fast R-CNN, there is no need to process 2000 region proposals, but only to generate a feature map once.

2.1.3 Faster R-CNN

R-CNN and Fast R-CNN use selective search which is slow and inefficient. Ren et al proposed Faster R-CNN (Ren et al 2016) with a region proposal network which eliminates the need of selective search and lets the network learn the region proposal instead. Instead of using the selective search algorithm on the feature map to identify the region proposals, a separate network is used to predict the region proposals. Faster R-CNN solved the drawback of R-CNN and Fast R-CNN, and it allows real time inference at 5 FPS on GPU.

2.2 Regression Based Framework

2.2.1 Single Shot Multibox Detection

The SSD (Liu et al, 2016) approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections.

2.2.2 You Only Look Once

In YOLO (Redmon et al, 2016), a single convolutional network predicts the bounding boxes and the class probabilities for these boxes. YOLO makes use of the whole topmost feature map to predict both confidences for multiple categories and bounding boxes. YOLO divides the input image into an S×S grid and each grid cell is responsible for predicting the object centered in that grid cell. Each grid cell predicts bounding boxes and their corresponding confidence scores.

CHAPTER 3: ANALYSIS AND DESIGN

3.1 Requirement Analysis

This project aims to provide real time detection of incorrectly worn face masks in public spaces. The requirements for this project include real time detection, high performance, light model, and is able to run on modern Deep Learning framework.

3.1.1 Realtime Inference

The developed system should at least be able to infer the input video stream at 30fps. Realtime inference is important because we want to detect all these incorrect worn face masks in real time, so that the authorities are able to take actions as quickly as possible.

3.1.2 High Performance

The developed system will be used by authorities in detecting whether people worn the face mask incorrectly, or did not wear a face mask at all. It is important to make sure that this system has a high precision. Besides, we want the system to have high recall, as we do not want to have false negatives.

3.1.3 Light Model

We want our system to have a light model as the lighter model requires less memory and computational power. Then it can compute faster with less memory. With these, we can use less cost computer devices, such as embedding systems, to run this system.

3.1.4 Software Tools Chosen

We use modern Deep Learning framework because they provide better modularity and are more customizable. We use the PyTorch framework, which is a modern framework that provides high modularity design and is very customizable together with other Python libraries like OpenCV, Numpy, Pillow and Pandas.

3.2 Methodology

3.2.1 System Design

In this system, we constructed YoloV5, which is the latest state of the art variant of YOLO (Redmon et al, 2016). The following diagram shows the architecture of YOLOv5.

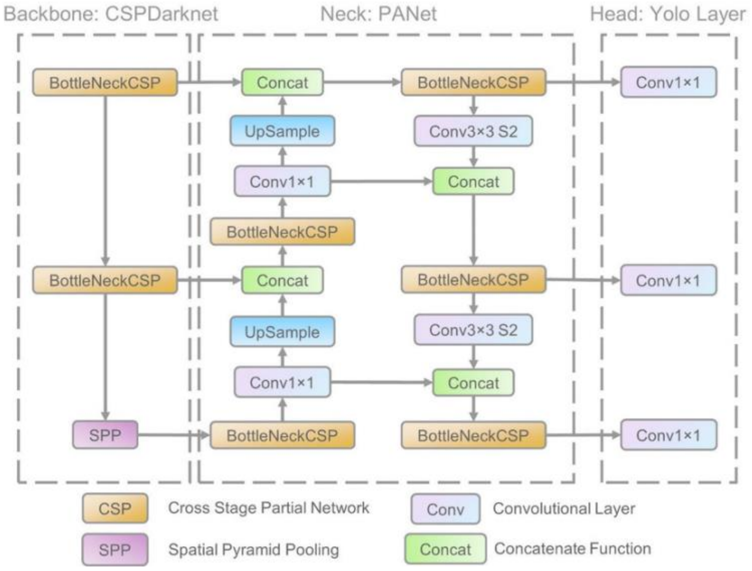

Figure 3.1 Architecture Diagram

We choose YOLOv5-S for face mask detection because it is light weight, and fast with state of the art performance, and thus we can do real-time inference with it.

3.2.2 Flowchart

We divide our work into training and inference stage. Training stage will only be run during training, while the inference stage will be run when system is being used by users.

3.2.3 Training Flowchart

In the training stage, we split the training images into batches with batch size of 16, and resize the images into a shape of 320x320 pixels. The images are then feed to the model to predict for boundary box and classification. Based on the outputs of forward pass, the loss is calculated by using the labels (ground truth) and used to run backward propagation to calculate the gradients. Finally, the model weight is updated with the gradient. This is repeated for 300 times which is when the model converges.

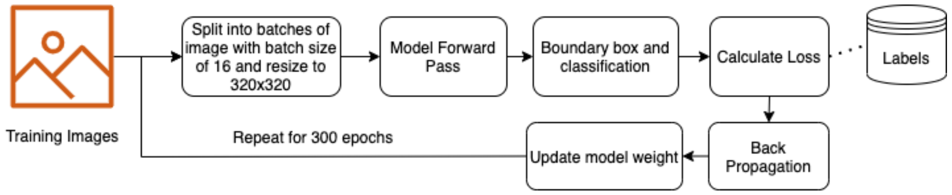

Figure 3.2 Training Flowchart

3.2.4 Inference Flowchart

In the inference stage, we take either real time video stream from camera or static images as source of image to be inferred. We then split them into a batch of images with batch size of 1, and resize to 320x320. The model will predict for boundary boxes which will indicate the location of faces in the images, and also classify whether they are wearing masks correctly. Finally, we render a new image frame, together with boundary box and class name so that the outputs can be easily understood by the user. In the case of a real time video stream, the inference process will keep running with real time input from the camera until it is terminated by user.

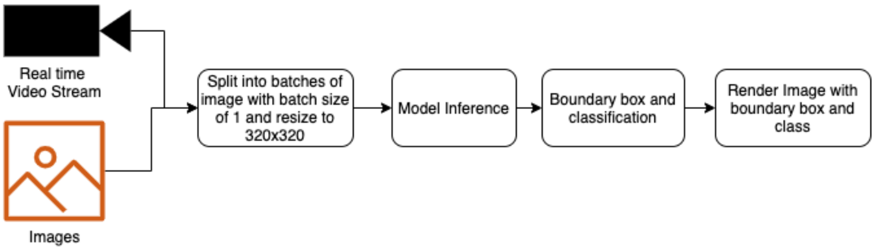
Real time Video Stream
Images
Split into batches of image with batch size of 1 and resize to 320x320
Model Inference
Boundary box and classification
Render Image with boundary box and class

Figure 3.3 Inference Flowchart

3.2.5 Processing steps

The dataset we are using is in PascalVOC format. The annotation includes boundary boxes and their class. The boundary box coordinates are in the form of xyxy, which is not compatible with YOLOv5, which requires the format to be xywh. Moreover, the class label in PascalVOC is in full text format, where the YOLOv5 prefers a 0-indexed class id as the label. Conversion code is written to convert the ground truth from PascalVOC format into YOLOv5 format, and split the data into training, testing and validation sets, which have 682, 85, and 86 images respectively. Mosaic Augmentation is also used to conquer the overfit problem and improve the model performance.

CHAPTER 4: EXPERIMENT RESULT

4.1 Experiment Setting

We trained on the Mask Face dataset, with batch size of 16, and image size of 320x320. We use the YOLOv5-s variant of the YOLO V5 model which has the lightest weight and fastest. We trained for 300 epochs in 3 hours on a single Nvidia Tesla P100 GPU.

4.2 Result Analysis

Our model converged at the end of training, with best mAP@.5 of 0.80. All three losses: Box, Objectness and Classification converge at the end of training.

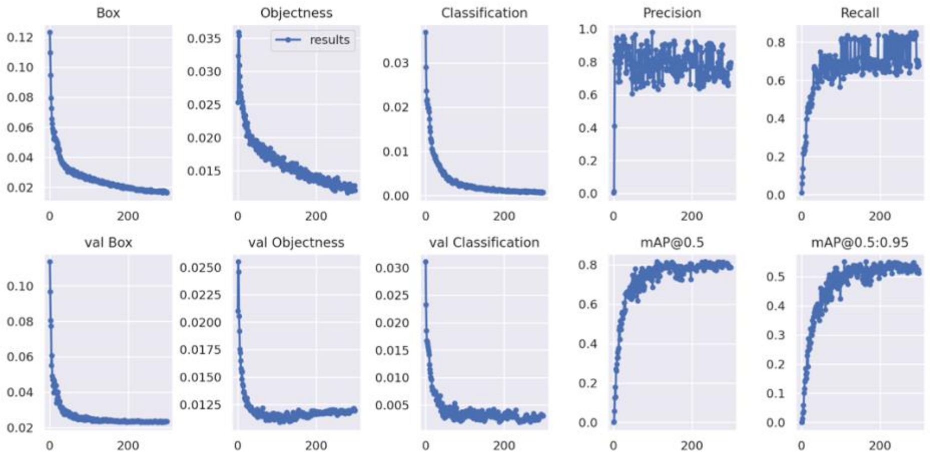

Figure 4.1 Training Metrics

We plotted the confusion matrix for the model, it shows we have good performance.

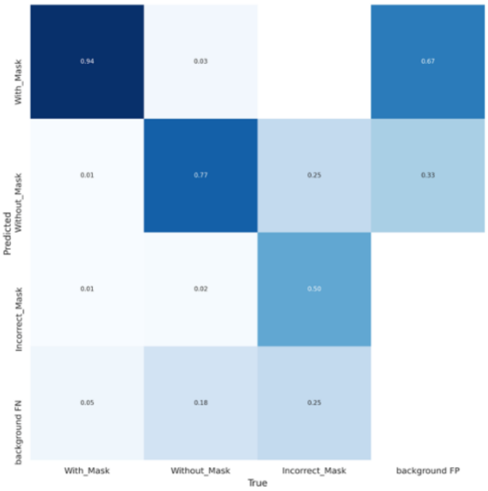

Figure 4.2 Confusion Matrix

We plot the graph for Recall vs Confidence and Precision vs Recall. This shows that our model has better recall than precision, which is what we intended for. This is sufficient to detect people who have worn masks incorrectly, or did not wear masks at all.

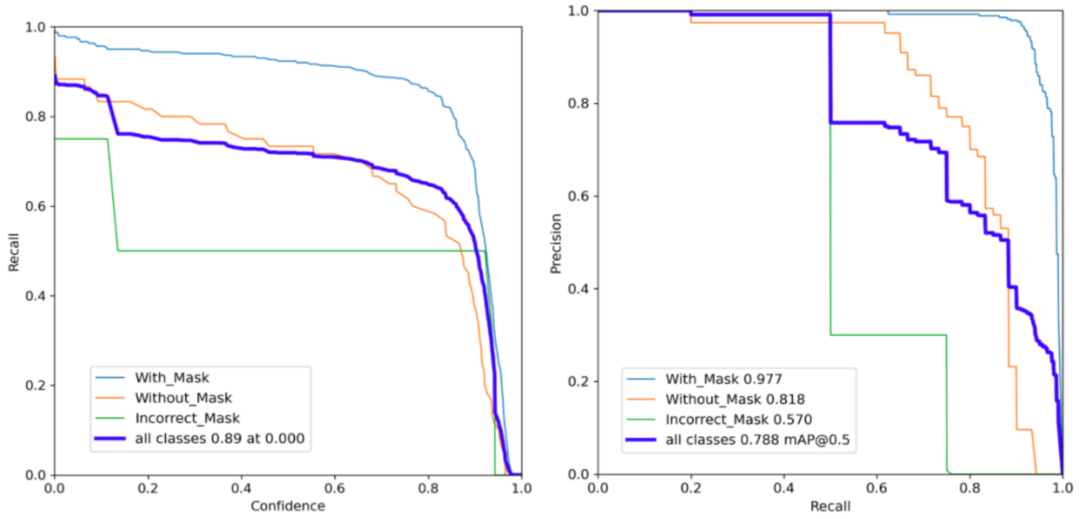

Figure 4.3 Recall-Confidence Curve and Precision-Recall Curve

We have tabulated the performance for different classes in the following table, which tell the performance for each class. We achieved an acceptable result in all the classes. However, we notice also the data imbalanced issue in the dataset, which lead to slightly poor performance for “Without Mask” and “Incorrect Mask” labels compared to “With Mask”, because these classes have comparably less samples than “With Mask”.

Table 4.1 Model Performance

Class	Images	Labels	Precision	Recall	mAP@.5
With Mask	85	301	0.982	0.886	0.977
Without Mask	85	60	0.949	0.65	0.818
Incorrectly worn	85	4	0.455	0.50	0.570

One of our system requirements is capable of real time inference. This required to predict the input at least 30 FPS. We record 11.76ms (85FPS) for the inference time on a single GPU. This is very sufficient and we can even opt for lower cost hardware.

4.2.1 Compare Result

We run the inference on a batch of test samples, and annotate the image with predicted boundary box and labels. We show the results in the following figure. We can see that all the test samples are labelled correctly. As a part of the requirements of the assignment, we're required to try the model on our own photo. All three classes correctly detected in my own photo using the model developed.

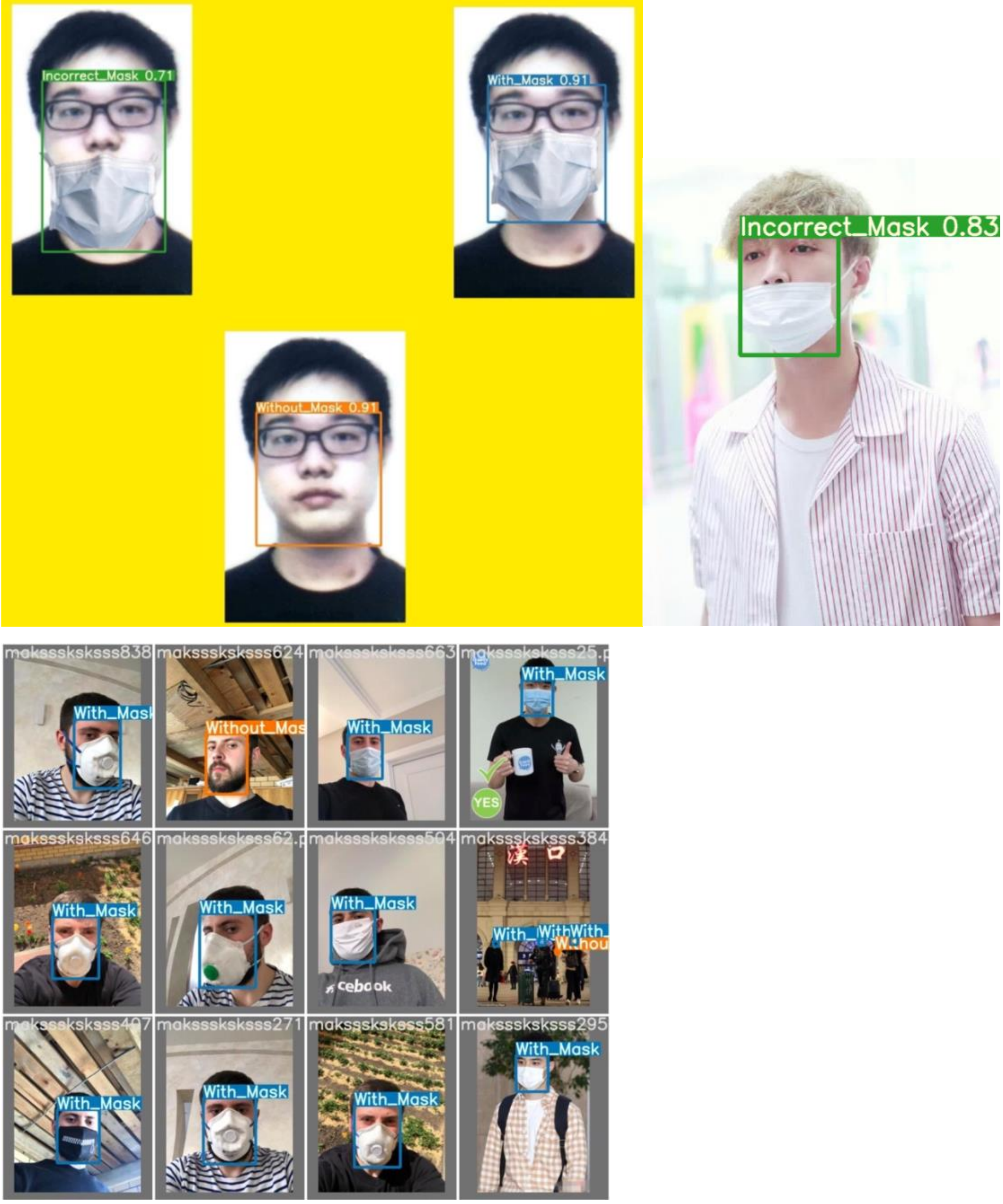

Figure 4.4 Sample Input and Output

CHAPTER 5: DISCUSSION

5.1 Strength

In this work, we developed a model that is able to detect incorrectly worn masks, and also detect people who did not wear masks. Our model has good performance and can predict correctly. Speed wise, we achieved 85 FPS using just a single GPU with this model, and this is far better than our objective, which is 30FPS. We are able to run the model with much lower hardware computational power at much lower cost. Besides, our model is a light model, which contains only 7059304 parameters, and this plays a significant role for its fast inference and low memory footprints. Lastly, we developed this model using YOLOv5 and PyTorch, which is the modern Deep Learning library that allows us for further customization. In all, we achieved all our objectives and system requirements.

5.2 Limitation

The challenges we face with this work is the data imbalance problem. The dataset is highly imbalanced, with most of the data being correct wear masks, and only a few are incorrect wearing masks and did not wear masks. With adding more data to the Without Mask and Incorrect Mask class, we should be able to improve the model performance even better.

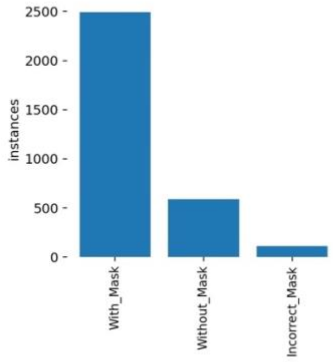

Figure 5.1 Data Imbalance

CHAPTER 6: CONCLUSION

In this work, we used knowledge in Object Detection, which is one of the topics in Image Processing. Object detection is a combination of Image Classification and Image Localization, but with ability to detect more than one object in the image. In object detection, the goal is to detect different types of objects that is appeared in the image.

In modern Object Detection algorithm, Convolutional Neural Network (CNN) is used as a building block of efficient object detection algorithm. There are two types of CNN-based algorithm, which are Region Proposal and Regression-based method. Region proposal method search on multiple region proposals for object, while regression based method usually do only a single shot search for all at once. As a result, region proposal take longer time but usually generate better results, while regression-based, is lot more faster with a slightly trade-off on performance.

In conclusion, we successfully trained an algorithm that is able to detect human faces, and classify whether they wear mask correctly, or not wearing at all. Our algorithm can detect the human faces that appear in the image and classify for their types with a good performance. We use regression-based algorithm so that our model is able to deploy with low-cost device with real-time inference.

REFERENCES

Chu, D. K., Akl, E. A., Duda, S., Solo, K., Yaacoub, S., Schünemann, H. J., ... & Reinap, M. (2020). Physical distancing, face masks, and eye protection to prevent person-to-person transmission of SARS-CoV-2 and COVID-19: a systematic review and meta-analysis. *The Lancet*, *395*(10242), 1973-1987.

Feng, S., Shen, C., Xia, N., Song, W., Fan, M., & Cowling, B. J. (2020). Rational use of face masks in the COVID-19 pandemic. *The Lancet Respiratory Medicine*, *8*(5), 434-436.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Girshick, R. B., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR abs/1311.2524 (2013). *arXiv preprint arXiv:1311.2524*.

Leung, C. C., Lam, T. H., & Cheng, K. K. (2020). Mass masking in the COVID-19 epidemic: people need guidance. *Lancet*, *395*(10228), 945.

Liu, Wei, 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*, pp. 21-37. Springer, Cham, 2016.

Machida, M., Nakamura, I., Saito, R., Nakaya, T., Hanibuchi, T., Takamiya, T., Odagiri, Y., Fukushima, N., Kikuchi, H., Amagasa, S., Kojima, T., Watanabe, H., & Inoue, S. (2020). Incorrect Use of Face Masks during the Current COVID-19 Pandemic among the General Public in Japan. *International journal of environmental research and public health*, *17*(18), 6484. https://doi.org/10.3390/ijerph17186484

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*.

Rosebrock, A. (2016). Intersection over Union (IoU) for object detection. *Online] http://www.pyimagesearch.com/2016/11/07/intersection-overunion-iou-for-objectdetection*.

APPENDIX A SOURCE CODE

The source code is available in the google drive folder in this link:

https://drive.google.com/drive/folders/1xTiBdwlPBdCM5oFAMDgD6bWCUzURmpBm?usp=sharing

I have also uploaded the dataset that I have converted at here:

https://www.kaggle.com/jiuntian/jkkkkl

.