MASK DETECTION USING DEEP LEARNING METHOD

TAN YU XUAN

UNIVERSITI TEKNOLOGI MALAYSIA

MASK DETECTION USING DEEP LEARNING METHOD

TAN YU XUAN

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Computer and Microelectronic System)

Faculty of Engineering
Universiti Teknologi Malaysia

JULY 2022

DEDICATION

This proposition is dedicated to my dad who taught me that knowledge best
learnt for himself is the greatest sort of knowledge. It is also dedicated to my mother,
who taught me that even one step at a time may fulfil the greatest goal.

ACKNOWLEDGEMENT

I communicated with a wide range of people, including researchers, academicians, and practitioners, to prepare this thesis. They have aided my comprehension and thoughts. I'd like to express my gratitude to everyone who helped make this proposal a reality. First and foremost, I would like to express our heartfelt appreciation to my project supervisor, Ts. Dr. Nor Aini Binti Zakaria, for her invaluable advice, guidance, and unwavering patience throughout the project's development. I learned a lot of new things, and this thesis would not have been the same without your continued support and interest.

In addition, I'd like to thank my beloved parents, friends, and colleagues for always being there for me and inspiring me. I would not have been able to finish this proposal without their support. Last but not least, I'd like to express my gratitude to everyone who assisted and motivated me to work on this proposal. Their perspectives and suggestions are extremely beneficial. Unfortunately, in just this limited space, it is impossible to list all of them. I don't have any bombastic words to express my gratitude, but my heart is still full of the kindnesses I've received from everyone.

ABSTRACT

Wearing face masks outdoors has been a new norm due to the COVID 19 pandemic as an initiative of controlling the spread of coronavirus. To reduce the risk of people being exposed to viruses, face masks were compulsory to be worn by Malaysians. However, there are people who refused to do so due to various reasons such as feeling lazy, uncomfortable, troublesome, and others, even the act of wearing a face mask is enforced by law. Therefore, it is essential to build a face mask detector to monitor automatically and ensure people are wearing masks correctly. The performance such as precision and response time of face mask detectors are important to support their application in the real-time working environment. The issue of performance enhancement in the form of adding more layers or implementing hybrid models such as spatial pyramid pooling (SPP) modules is increasing the complexity of the algorithm and making it bulky. The objective of this paper is to build a face mask detector by using the latest high-performance deep learning model, YOLOv4 and YOLOv5 together with MixUp technique which can contribute to high mean accuracy precision (mAP) and short inference time that suffice the requirements to be working in a real-time environment. This research conducted data sets collection and data annotation at the beginning stage of the algorithm, then MixUp technique was applied to the collected datasets to train the YOLOv4 and YOLOv5 using Google Colab. Next, the trained model was tested, and the performance was evaluated in terms of mAP using the average precision (AP) from the confusion matrix and inference time based on the time taken for prediction. The algorithm with the YOLOv5 model having slightly lower mAP than YOLOv4 but shorter training and inference time. However, both models able to detect and classify the input image to three classes included with-mask (1), without-mask (2), and incorrectly with-mask (3) with good performance.

ABSTRAK

Memakai topeng muka di luar rumah telah menjadi norma baharu berikutan pandemik COVID 19 sebagai inisiatif mengawal penyebaran coronavirus. Untuk mengurangkan risiko orang ramai terdedah kepada virus, topeng muka wajib untuk dipakai oleh rakyat Malaysia. Memakai topeng muka dikuatkuasakan oleh undang-undang, namun, terdapat pihak yang enggan membuat demikian atas pelbagai alasan seperti rasa malas, tidak selesa, menyusahkan dan lain-lain. Oleh itu, pembinaan pengesan topeng muka untuk memantau secara automatik amat penting untuk memastikan orang ramai memakai topeng dengan betul. Prestasi pengesan topeng muka seperti kepersisan dan masa pentaabiran adalah penting untuk menyokong aplikasinya dalam persekitaran kerja secara langsung. Isu peningkatan prestasi dengan penambahan lapisan atau modul pasca pemprosesan seperti modul Spatial Pyramid Pooling (SPP) meningkatkan kerumitan dan saiz algoritma. Objektif projek ini adalah membina pengesan topeng muka dengan menggunakan model pembelajaran mendalam yang terkini dan berprestasi tinggi, iaitu YOLOv4 dan YOLOv5 bersama-sama dengan teknik MixUp yang boleh menyumbang kepada min purata kepersisan yang tinggi dan masa pentaabiran yang pendek dan mencapai keperluan untuk bekerja dalam persekitaran kerja secara langsung. Penyelidikan ini menjalankan pengumpulan data dan anotasi data pada peringkat permulaan algoritma, kemudian teknik MixUp digunakan pada set data yang dikumpul untuk melatih YOLOv4 dan YOLOv5 dengan Google Colab. Seterusnya, model yang dilatih akan diuji, dan prestasi dinilai dari segi min purata kepersisan dan masa pentaabiran berdasarkan masa yang dieprlukan untuk mendapatkan keputusan ramalan. Pengesan topeng muka yang dibina dengan model YOLOv5 dalam projek ini mempunyai min purata kepersisan yang lebih rendah tetapi masa latih dan pentaabiran yang lebih pendek berbanding kepada model YOLOv4. Kedua-dua model boleh mengklasifikasikan imej kepada tiga kelas termasuk dengan topeng (1), tanpa topeng (2), dan dengan topeng tetapi cara memakai yang salah (3) dengan prestasi yang baik.

TABLE OF CONTENTS

DECLARATION **ii**

DEDICATION **iii**

ACKNOWLEDGEMENT **iv**

ABSTRACT **v**

ABSTRAK **vi**

TABLE OF CONTENTS **vii**

LIST OF TABLES **x**

LIST OF FIGURES **xi**

LIST OF ABBREVIATIONS **xiv**

CHAPTER 1 INTRODUCTION **1**

1.1 Background Study **1**

1.1.1 Facemask and COVID-19 1

1.1.2 Application of face mask detector 4

1.1.3 Object detection 5

1.1.4 Deep learning in detections 9

1.2 Motivation **18**

1.3 Problem Statements **19**

1.4 Research Aim and Objective **19**

1.5 Research Scope **20**

1.6 Chapter Summary **21**

CHAPTER 2 LITERATURE REVIEW **22**

2.1 Bags-of-Freebies (BoF) 22

2.2 Bags-of-Specials (BoS) 23

2.3 Datasets 24

2.3.1 Data labelling and data annotation 26

2.3.2 Data augmentation 28

2.4 Deep Learning Algorithm - YOLO 30

2.4.1 YOLOv4 30

2.4.2 YOLOv5 36

2.5 Summary of Related work 39

CHAPTER 3 RESEARCH METHODOLOGY 45

3.1 Introduction 45

3.2 Solution Problem Domain 45

3.3 Solution Details 47

3.3.1 Dataset Preparation Details 47

3.3.2 Classification Details 54

3.3.3 Performance Evaluation Details 57

3.4 Experimental Setup 61

3.4.1 Dataset : Kaggle 61

3.4.2 Data annotation tool : CVAT 62

3.4.3 Project platform : Roboflow 62

3.4.4 Deep learning executable document : Google Colab 63

3.5 Research Plan 64

CHAPTER 4 RESULTS AND DISCUSSION 67

CHAPTER 5 CONCLUSION 76

REFERENCES 78

LIST OF TABLES

Table 1.1 Object detector block and the respective model [12]. 9

Table 1.2 Error type of VOC 2007 analysis and reason [14]. 17

Table 2.1 Nueral networks parameters for image classification [12]. 32

Table 2.2 Summary of related work. 39

Table 3.1 Gantt chart for Project 1 64

Table 3.2 Gantt chart for Project 2. 65

Table 4.1 : Mean Average Precision (mAP) performance comparison between YOLOv4 and YOLOv5. 67

Table 4.2 : True Positive (TP), False Positive (FP) and Average Precision (AP) in every class of YOLOv4 trained by data set Without-MixUp data and With-MixUp data. 68

Table 4.3 : Inference result of images from different class and variation on YOLOv4 and YOLOv5. 69

Table 4.4 : Mean Average Precision (mAP) comparison between current paper and reference paper. 73

LIST OF FIGURES

Figure 1.1 Global statistics of COVID-19 confirmed cases and deaths by WHO [4] 1

Figure 1.2 Statistics of COVID-19 confirmed cases and deaths by COVIDNOW, Malaysia [7] 3

Figure 1.3 Application of face mask detector. 4

Figure 1.4 Real-world application of face mask detector (left) and temperature reader (right) 5

Figure 1.5 Three tasks of computer vision on face mask detection example, classification (left), object detection (middle) and segmentation (right) [2] 6

Figure 1.6 Graphics Illustration of how object detection happened 7

Figure 1.7 Model of object detector [12]. 8

Figure 1.8 Convolutional network architecture [9]. 10

Figure 1.9 Object detection using R-CNN model (two-stage object detection). 10

Figure 1.10 Overview of YOLO detection system [14]. 12

Figure 1.11 Object detection using YOLO mode regression problem (one-stage object detection) [14]. 12

Figure 1.12 Non-max suppression on cleaning the bounding boxes. 14

Figure 1.13 Breakdown result of VOC 2007 analysis on Fast R-CNN and YOLO [14]. 16

Figure 2.1 Using LabelImg to draw the ground truth on face mask image [2]. 27

Figure 2.2 Example of MixUp technique [2]. 29

Figure 2.3 Example of horizontal flip technique [3]. 30

Figure 2.4 Comparison of YOLOV4 to other state-of-the-art object detectors [12]. 31

Figure 2.5 Architecture selection for every parts in YOLOv4 [15]. 32

Figure 2.6 Original architecture of SPP proposed by He et al [19]. 33

Figure 2.7 Modified architecture of SPP proposed by Huang et al [19]. 34

Figure 2.8 Architecture of Path Aggregation Network (PAN) [21]. 35

Figure 2.9 Architecture of modified Path Aggregation Network (PAN) [12]. 36

Figure 3.1 General flow of project methodology. 46

Figure 3.2 Data annotation using PineTools. 47

Figure 3.3 Annotation using CVAT (new task). 48

Figure 3.4 Annotation using CVAT (start job). 49

Figure 3.5 Annotation using CVAT (manual labelling). 49

Figure 3.6 Dataset ratio allocation using Roboflow 50

Figure 3.7 Roboflow pre-processing. 50

Figure 3.8 Mismatching of EXIF orientation between image and annotation (Left) and correct EXIF orientation (Right) 51

Figure 3.9 Folder of dataset annotated in YOLO Darknet TXT format. 52

Figure 3.10 Example text (.txt) file of an image. 52

Figure 3.11 Label file of annotation. 52

Figure 3.12 Dataset information without MixUp Data. 53

Figure 3.13 Dataset information with MixUp Data. 54

Figure 3.14 : Example of YOLO Darknet TXT format. 55

Figure 3.15 Example of YOLOv5 PyTorch TXT format 56

Figure 3.16 GT bounding box and predicted bounding box. 58

Figure 3.17 Parameters of IoU equation, red box is representing the ground truth bounding box, yellow box s representing the predicted bounding box, blue area is representing the intersection area and green area is the union area. 58

Figure 3.18 Confusion Matrix 59

Figure 3.19 Roboflow dataset health check. 62

Figure 4.1 : Inference result of class Without-mask with variation of front face (i), left face (ii) and right face (iii) 72

Figure 4.2 : Inference result of class With-mask with variation of front face (i), right face (ii) and left face (iii) 73

Figure 4.3 : Inference result of class Incorrectly With-mask with variation of nose not covered (i), nose and mouth not covered (ii) and chin not covered (iii) 73

Figure 4.4 : Multiple object detection by using proposed face mask detector. 74

LIST OF ABBREVIATIONS

mAP	-	mean Average Precision
CCTV	-	Closed-circuit Television
YOLO	-	You Only Look Once
YOLOv1	-	You Only Look Once Version 1
YOLOv2	-	You Only Look Once Version 2
YOLOv3	-	You Only Look Once Version 3
YOLOv4	-	You Only Look Once Version 4
YOLOv5	-	You Only Look Once Version 5
AI	-	Artificial Intelligence
CV	-	Computer Vision
JSON	-	JavaScript Object Notation
CNN	-	Convolutional Neural Network
R-CNN	-	Regions with Convolutional Neural Network
SSD	-	Single shot Detector
FPN	-	Feature Pyramid Network
BiFPN	-	Bi-directional Feature Pyramid Network
NAS-FPN	-	Neural Architecture Search - Feature Pyramid Network
PAN	-	Path Aggregation Network
CPU	-	Central Processing Unit
GPU	-	Graphics Processing Unit
TPU	-	Tensor Processing Unit
VGG	-	Visual Geometry Group
SPP	-	Spatial Pyramid Pooling
ASPP	-	Atrous Spatial Pyramid Pooling
SPM	-	Spatial Pyramid Matching
RFB	-	Receptive Field Block
SAM	-	Spatial Attention Module
ASFF	-	Adaptively Spatial Feature Fusion

SFAM	-	Simplified Fuzzy Adaptive Resonance Theory Map
FCOS	-	Fully Convolutional One-Stage Object Detection
RPN	-	Region Proposal Network
SVM	-	Support Vector Machine
IoU	-	Intersection over Union
HOG	-	Histogram of Oriented Gradients
DPM	-	Dynamic Pipeline Mapping
VOC	-	Visual Object Classes
BoS	-	Bags-of-Specials
BoF	-	Bags-of-Freebies
CVAT	-	Computer Vision Annotation Tool
MSE	-	Mean Square Error
SE	-	Squeeze-and-Excitation
COCO	-	Common Objects In Context
KITTI	-	Karlsruhe Institute of Technology
MAFA	-	Masked Face Analysis
VIA	-	VGG Image Annotator
VoTT	-	Visual Object Tagging Tool
XML	-	Extensible Markup Language
GAN	-	Generative Adversarial Network
HSV	-	Hue, Saturation, Value
GIOU	-	Generalized Intersection over Union
Colab	-	Colaboratory
EXIF	-	Exchangeable Image File Format
PR	-	Precision-Recall
GT	-	Ground Truth
TP	-	True Positive
FP	-	False Positive
FN	-	False Negative
AP	-	Average Precision

CHAPTER 1

INTRODUCTION

1.1 Background Study

1.1.1 Facemask and COVID-19

First case of corona virus infection happened in December 2019 at Wuhan of China, then the disease spread around the world in a lightning speed [1][2][3]. According to the official statistics dashboard of World Health Organization (WHO) which can be accessed through https://covid19.who.int/, there have been 270,031,622 confirmed cases of COVID-19 including 5,310,502 deaths globally, reported to WHO as of 14 December 2021 [4]. Figure 1.1 is showing the global statistics of corona virus cases, the trend chart is indicating the number of corona virus cases were remained high even mass numbers of vaccine have been distributed.

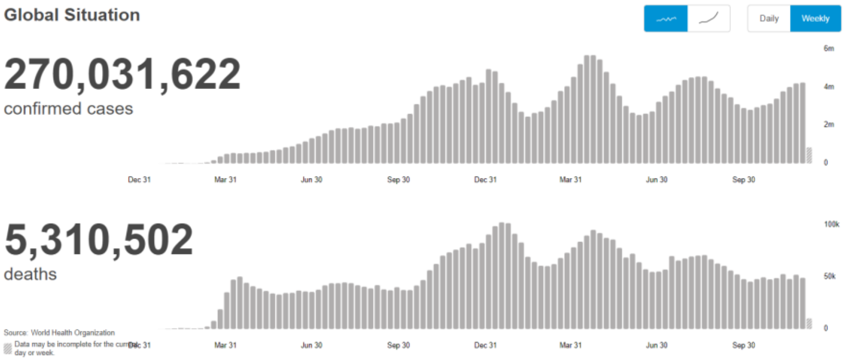


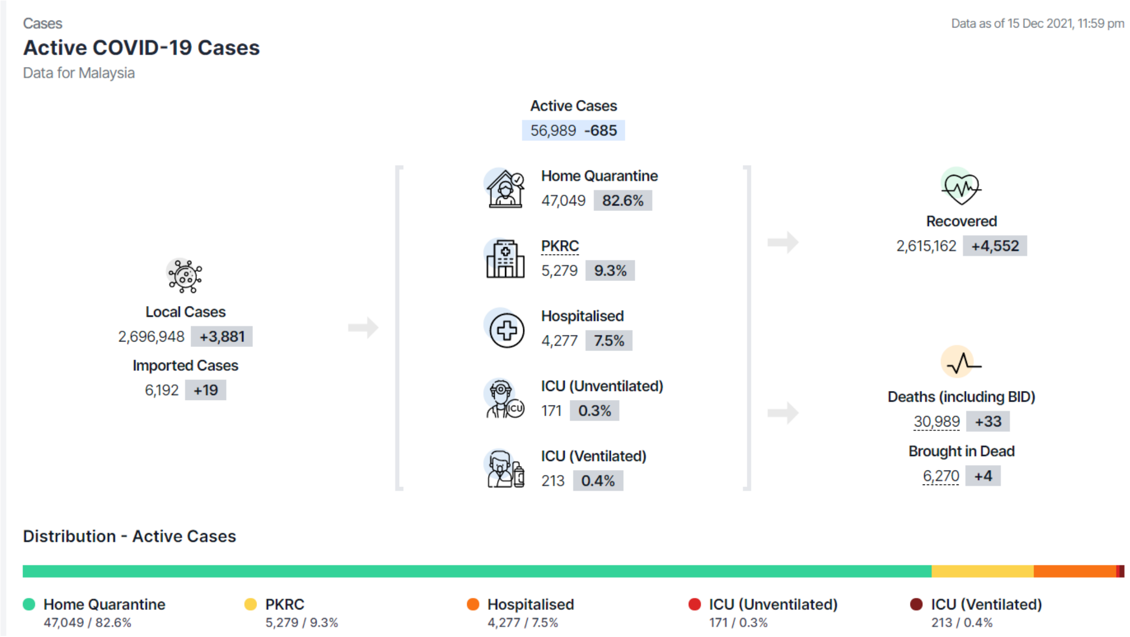
Figure 1.1 Global statistics of COVID-19 confirmed cases and deaths by WHO [4]

According to WHO, corona virus spread via two main routes, respiratory droplets, and any form of physical contact [3]. The droplets were generated by the infected patient when they are sneezing or coughing, these droplets can be inhaled easily by others which is in distance of approximately 4 feet to patient. Besides, these infection-causing droplets can also stick on several surfaces where the virus can stay active and alive for days. This is how an infected patient became the source of infection and cause big group of infection cases in a short duration [5].

At current stage, there are various COVID-19 vaccines such as Pfizer, Sinovac, AstraZeneca, CanSino and others have been distributed to fight against corona virus, but these vaccines do not guarantee zero infection even with individual was vaccinated completely. Vaccine is playing a big role in boosting human immune system against corona virus and hence reduce the risk of death. Therefore, measures such as practicing social distancing, wearing face masks correctly and sanitizing hands regularly are important efforts for individuals to reduce their risk from exposing to corona virus.

Utilization of face masks for outdoor activities have been suggested by the WHO as control measure of limiting corona virus spread, face masks can protect healthy individual from the external virus and prevent the spreading of virus from corona patient as well. There are individuals who unknowingly spread the corona virus to others which cause a chain reaction of infection. Some infected patient does not have symptoms such as fever, joint pain, tasteless and others at the initial state which cause them unaware themselves as the carrier of corona virus. Wearing mask can greatly reduce the risks of people exposed to corona virus at public area because the virus from the patient sneeze and cough were blocked by mask [5]. Besides, the corona virus also can be encountered from the virus-contaminated surface such as stair handrail, lift buttons, door handler and others. People can be easily infected when they touched their mouth, nose or eyes with hands that contacted the virus-contaminated surface [5].

Moreover, Dzisi et al. suggested public transports passengers to wear masks because it is a high-risk closed area [6]. According to Junaidi et al., there are studies which suggested the general public to put on mask during the COVID-19 pandemic, depending on local conditions [1]. Therefore, there is no doubt on the importance of wearing face masks properly as an initiative to control virus during the COVID-19 pandemic.



Cases

Active COVID-19 Cases

Data for Malaysia

Data as of 15 Dec 2021, 11:59 pm

Active Cases
56,989 -685

Local Cases
2,696,948 +3,881

Imported Cases
6,192 +19

Home Quarantine
47,049 82.6%

PKRC
5,279 9.3%

Hospitalised
4,277 7.5%

ICU (Unventilated)
171 0.3%

ICU (Ventilated)
213 0.4%

Recovered
2,615,162 +4,552

Deaths (including BID)
30,989 +33

Brought in Dead
6,270 +4

Distribution - Active Cases

● Home Quarantine 47,049 / 82.6%

● PKRC 5,279 / 9.3%

● Hospitalised 4,277 / 7.5%

● ICU (Unventilated) 171 / 0.3%

● ICU (Ventilated) 213 / 0.4%

Figure 1.2 Statistics of COVID-19 confirmed cases and deaths by COVIDNOW, Malaysia [7]

Malaysia is not exempted from the COVID-19 pandemic, Figure 1.2 shows that Malaysia have approximately 2.7 million of confirmed cases and 31 thousand of death as of 15 December 2021 according to COVIDNOW, official statistic dashboard from Ministry of Health Malaysia. To reduce the risk of people being exposed to viruses, face masks were compulsory to be wore by Malaysians. However, there are people who refuse to do so due to various reasons such as feeling lazy, uncomfortable, troublesome and others, even the act of wearing a face mask is enforced by law. Ensuring everyone is wearing face masks is one of the huge obstacles in the prevention work on the COVID-19 pandemic. Therefore, it is essential to build a face mask detector to monitor automatically and ensure people are wearing masks correctly [2]. In this project, the face mask detector aimed to be built with the ability of classifying the input images into three classes, three classes which are with-mask (1), without-mask (2), and incorrectly with-mask (3).

1.1.2 Application of face mask detector

Figure 1.3 is illustrating the system flowchart of face mask detector application, similar system design is proposed by Kumar et al. [8]. The face mask detection system can be divided into three main segments, camera and the captured images or videos as the vision system, deep learning-based face mask detector and result-driven devices such as alarm and monitor which indicated by the green, blue, and orange block in Figure 1.3. Camera is used to capture and provide the datasets to the face mask detector in form of images or videos automatically. Then, the trained face mask detector will detect face masks and generate detection result such as with-mask, without-mask, and with-mask but not wearing correctly. Lastly, the detection result is used to control the behaviour of output devices. For example, alarm is triggered if an individual was detected as without-mask, monitor system is used to display the detection result in form of graphics [8].

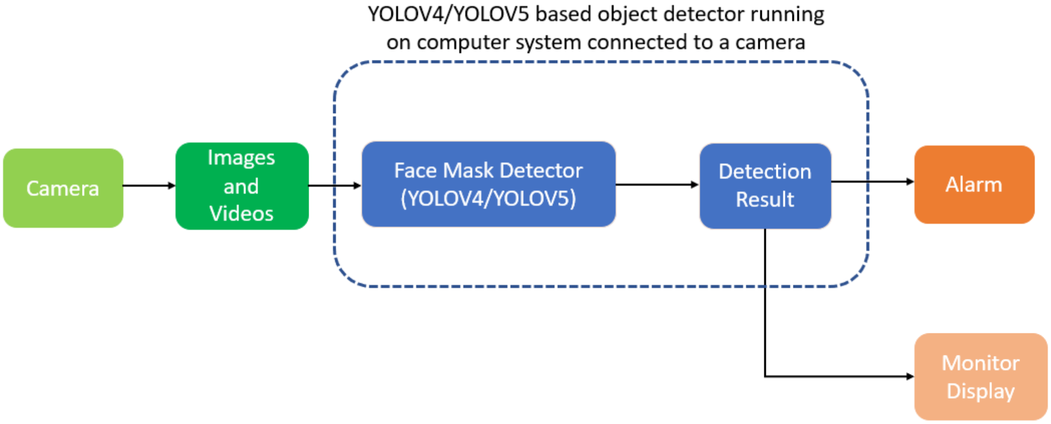

YOLOV4/YOLOV5 based object detector running on computer system connected to a camera
Camera
Images and Videos
Face Mask Detector (YOLOV4/YOLOV5)
Detection Result
Alarm
Monitor Display

Figure 1.3 Application of face mask detector.

Figure 1.4 is a real-world application of face mask detector and temperature sensor which located at the entrance of a shopping mall in Malaysia after visitor register their attendance. It will automatically detect the temperature and face mask wearing status of an individual in front of it. Besides of monitoring the visitors at the entrance, the face mask detector also can be applied by having a CCTV camera distributed evenly in the shopping mall and capture image or video in real-time, passing the image to a computer system executing the proposed YOLOv4 or YOLOv5 based pre-trained face mask detector. Then the face mask detector will draw a bounding box and detect if the individual is wearing mask or not. The system will trigger the alarm sound when visitor is detected not wearing mask and display the detected image on monitor located in the mall.

This paper is focusing on building the face mask detector using YOLOv4 and YOLOv5, which is the core component of the application. A face mask detector with good performance in terms of accuracy and response time is required to enable it to be applied in real-time working environment and assisted the monitor process in occasion such as shopping mall, market, and others to ensure people is wearing the mask properly. Location such as shopping mall with this face detection system can ensure people wearing mask in a more effective and efficient way, hence giving visitors more confidence on the corona virus risk control and more willing to pay a visit.

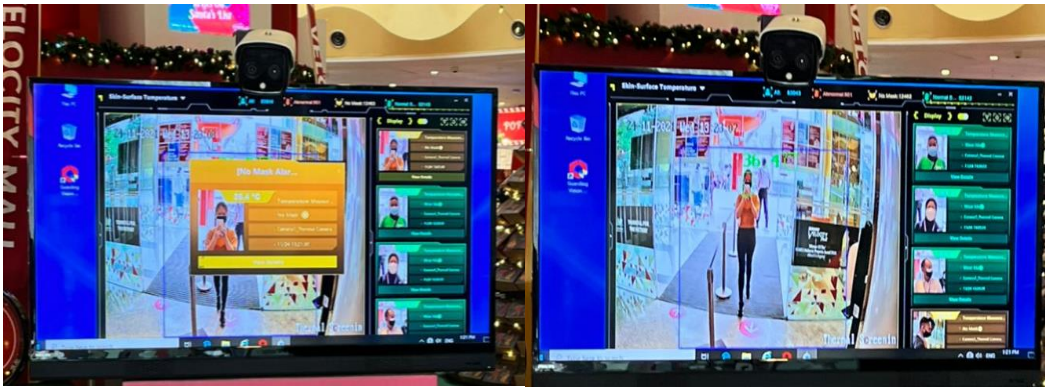


Figure 1.4 Real-world application of face mask detector (left) and temperature reader (right)

1.1.3 Object detection

Analysis on image and video have gotten a lot of attentions over the past years, artificial intelligence (AI) is the dominating technologies in this area which invented based on human biology. Computer vision (CV) is one of the branches of artificial intelligence which enable the computers to learn, detect and locate the objects by using cameras, images or videos and deep learning models, making computer able to "see" and process information based on what it captured [3][9]. Application of computer vision is wide range of industries such as transportations, security, banking, agriculture, manufacturing, and others [10][11]. Currently, it is also used in the biomedical field like medical imaging, disease detection at early stage, detection of tumor and others which supported doctors in the diagnosis and treatment of patient’s illness [3].

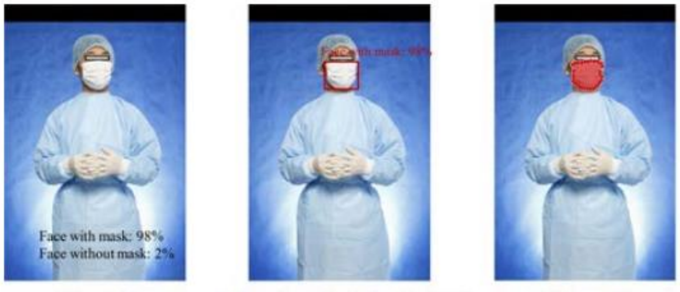


Figure 1.5 Three tasks of computer vision on face mask detection example, classification (left), object detection (middle) and segmentation (right) [2]

As illustrated in Figure 1.5, tasks of computer vision can be divided into classification of image (left figure), object detection (middle figure), and segmentation of image (right figure). Image classification and object detection both will classify the object to its category based on the input image, but object detection need to draw bounding box to locate the position of object [2]. In other words, object detection is involving two stages, object localization and object classification [3]. Next, segmentation can be categorized into semantic segmentation and instance segmentation, the prior is clustering pixels according to similarities while the latter is applied on multiple objects [9].

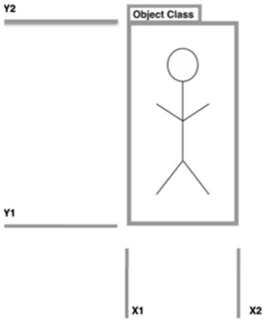


Figure 1.6 Graphics Illustration of how object detection happened

The definition of object detection is a process which detect and recognize the instances of objects in a image or a video into several classes such as animals, humans and so on [5]. As referring to Figure 1.6, Object detection model is functioning by predicting the coordinates (annotated as X1,X2,Y1 and Y2 in the Figure 1.6) and the label of object class. The behavior of object detection is detecting foreground object in each image or frame, foreground object is defined as an object different from the static background. It may be related to its appearance, or some local movement and it tends to change from frame to frame. Next, background object is referring to the stationary objects in a frame that belongs to the background [11]. Applying object detection is providing image as input of object detection model and receive predicted coordinates and label of class in form of JSON data.

There is two different approach of object detection algorithm, traditional method, and the deep learning-based method. Firstly, the traditional method in object is known as sliding window technique which first draw sub-windows or bounding boxes with varying scales and positions on an image. The sub-window will be resized and detected by CNN model to tell if it has any objects, hence classify the object. The next sub-window or bounding box will be obtained by sliding the current one with a particular step size such as one pixel. The first drawback of this approach is required a lot of computation resources. Secondly, it is not 'smart' enough as it created the needs to design for different manual features and algorithms for different type of objects, which caused it contributes to poor generalization performance [2].

Deep learning-based object detection algorithms have made significant advances in the general and specific objects' categories detection in recent years. The popular algorithms which have been proposed are R-CNN, YOLO series and Single shot Detector (SSD) and they are performing well based on the public datasets testing result. As compared to the traditional method, deep learning-based object detection algorithm improve the dependence on manual extraction of feature and generalization performance, for instance, features learnt by the deep learning algorithm on the ImageNet dataset is performing for other tasks as well. This is because deep learning approach is a multiple layer composite mapping, while traditional approach is a simple shallow mapping only [2].

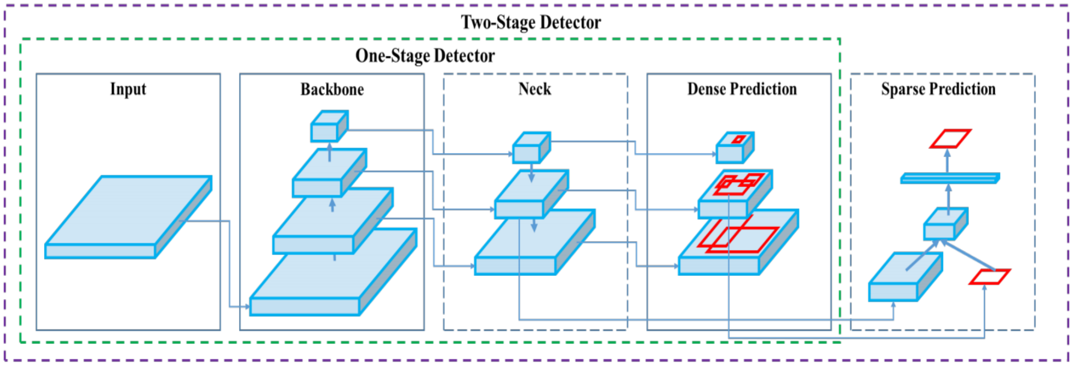


Figure 1.7 Model of object detector [12].

Figure 1.7 is showing the building blocks such as input, backbone, neck, and prediction block that form the object detector model. The building blocks can be divided to 2 parts, backbone part which is pre-trained on ImageNet to extract features and head that responsible for class prediction and objects bounding box drawing. Furthermore, there is two types of architecture for head part, one-stage object detector and two-stage object detector. Recently, the object detectors were improved by having additional layers added in between backbone and head which known as neck, the neck layers were used to collect the feature maps from different stage. From Table 1.1 which showing the respective model to each of the blocks in object detector, the popular networks in neck are Feature Pyramid Network (FPN), BiFPN and Path Aggregation Network (PAN). Besides, the most popular models of one-stage architecture in head is YOLO series and SSD, while the two-stage architecture in head is represented by the R-CNN series and its enhanced model such as fast R-CNN and faster R-CNN [12]. SqueezeNet , MobileNet and ShufflfleNet are the example of object detectors backbone running on central processing unit (CPU) while backbone on graphics processing unit (GPU) platform could be DenseNet, ResNet, ResNeXt and VGG. It is observable that there are head with type of anchor based and anchor free, anchor based means the head network is applied to each anchor box while anchor box is the ground truth bounding box [12].

Table 1.1 Object detector block and the respective model [12].

Blocks in object detector	Models
Input	Image, Patches, Image Pyramid
Backbone	ResNet-50, SpineNet, EffificientNet-B0/B7, ResNeXt-50, ResNeXt-101, CSPResNeXt50, CSPDarknet53, VGG16
Neck : Additional blocks	SPP, ASPP, RFB , SAM
Neck : Path-aggregation blocks	FPN, PAN, NAS-FPN, Fully-connected FPN, BiFPN, ASFF, SFAM
Head : Dense Prediction (one-stage, anchor based)	RPN, SSD, YOLO, RetinaNets
Head : Dense Prediction (one-stage, anchor free)	CornerNet, CenterNet, MatrixNet, FCOS
Head : Sparse Prediction (two-stage, anchor based)	R-FCN, Mask R-CNN, Faster R-CNN
Head : Sparse Prediction (two-stage, anchor free)	RepPoints

1.1.4 Deep learning in detections

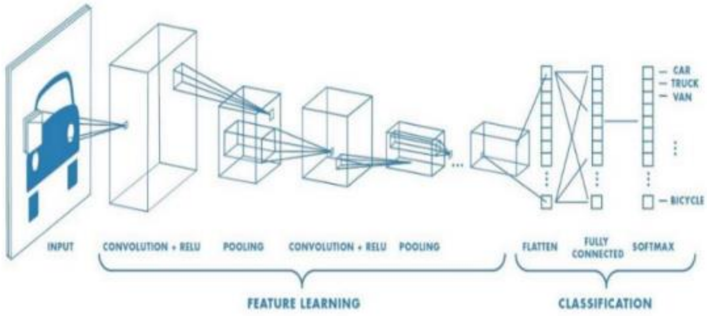


Figure 1.8 Convolutional network architecture [9].

There are three main components in convolutional neural network (CNN) which is known as convolutional layer, pooling layer, and fully connected layer as shown in Figure 1.8. Convolutional layer contains filters and feature maps. Filters are the processors of a specific later and they are unique from each other, filter usually receive pixel value as input and output to the feature map. The movement of filter is traversing along the image and moving by a pixel at a time. Next, pooling layer is implemented convolutional layer for the purpose of dimensionality reduction by generalizing features learnt from the previous feature maps. This layer is important to reduce the over fitting issue in training phase. Finally, the fully connected layer is utilized to assign the feature to class probability after the features extraction and consolidation from convolutional layer and pooling layer respectively [9].

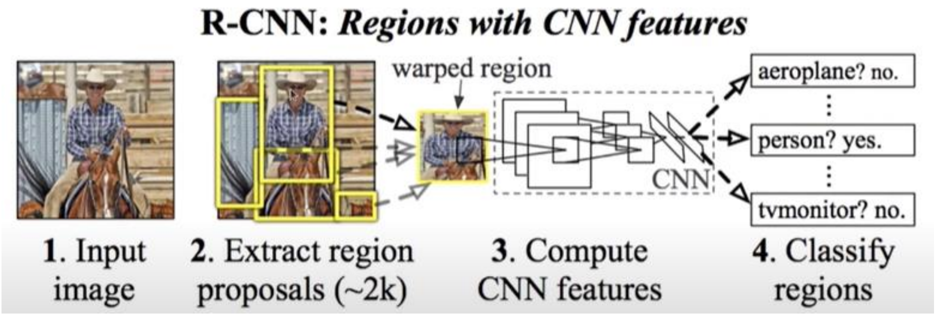


Figure 1.9 Object detection using R-CNN model (two-stage object detection).

The old object detection system like Deformable Parts Models (DPM) is utilizing sliding window method in which the classifier is running at evenly spaced location over the whole image. In the year of 2015, the pioneer of deep learning-based object detection is region-based convolutional neural networks (R-CNN) which is using the Region Proposal Network (RPN). Figure 1.9 is showing what RPN does, deterministic algorithm such as Selective Search (SS) is implemented to extract region proposals (potential bounding boxes) for the image. Usually, approximately 2000 regions will be proposed. The proposed regions were then resized to 224 x 224 and feed into CNN. Then, prediction is applied to the crop or proposed regions to determine the class for those proposed regions [3]. Therefore, R-CNN is known as two-stage object detection as well, because it needs two stage to perform object detection, first stage is performing region proposal using RPN and the second stage is feed the proposed regions into CNN for classification. Next, post-processing module is applied for bounding boxes refining to eliminate duplicate detections and rescore the box based on the other objects in the scene. As enhanced model, fast R-CNN is proposed by Girshick which replaced the SVM classifier with softmax regression to reduce the size of algorithm and time cost. To further improve the fast R-CNN, Girshick proposed faster R-CNN [13]. But this architecture is involving complicated pipelines that is slow and difficult to be optimized because every individual component need to be trained separately [14].

YOLO is a new approach in object detection, instead of repurposing classifiers to perform detection like R-CNN, YOLO reframe object detection as a single regression problem which separate the bounding boxes and associated class probabilities spatially [3]. In other words, YOLO is a single-stage neural network that direct predict coordinates of bounding boxes and probabilities of class from full images pixels in one evaluation only. In summary, using this system, you only look once (YOLO) at an image to predict the presenting objects and their location. In YOLO, whole detection pipeline is a single network which can have performance further enhanced from end to end directly [14].

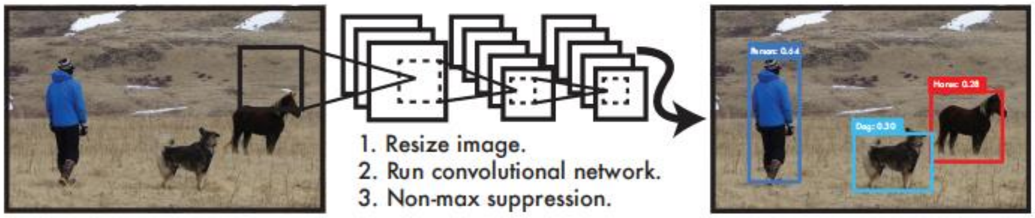


Figure 1.10 Overview of YOLO detection system [14].

Figure 1.10 is the overview illustration of YOLO detection system which is direct and simple to understand. Firstly, system resizes the image to the size of 416 x 416. Then, runs a single convolutional network on the image to obtain the class label for the detected object and finally perform non-max suppression to thresholds the detection results and remove the unwanted or bounding boxes predicted incorrectly based on the model's confidence [14].

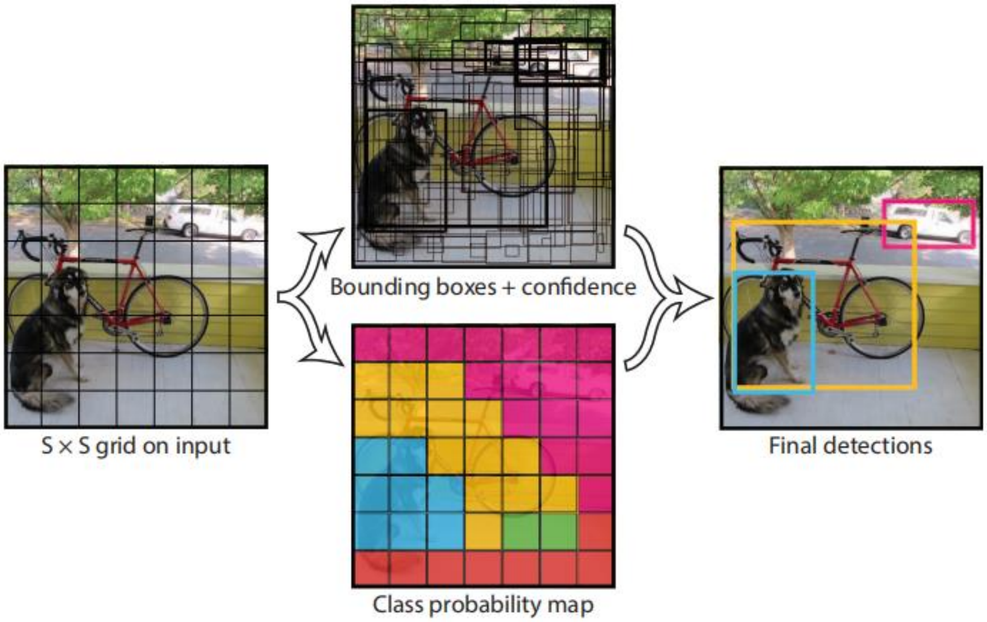


Figure 1.11 Object detection using YOLO mode regression problem (one-stage object detection) [14].

According to Figure 1.11 which shows how YOLO treat object detection as regression problem, the YOLO system began by dividing the input image into a grid with size S x S while it is 7 x 7 grid in this example. The grid cell is responsible for predicting both if there is a bounding box in that cell and the class probability of that cell. Firstly, every grid cell predicts C conditional class probabilities which denoted as $\Pr(Class_i/Object)$, which means that the probabilities are conditioned on the grid cell that contain an object [14]. Every grid cell will have only one class of probabilities regardless the number of B bounding boxes, the class probability of each cell is predicted and generate class probability map which has the blue cell corresponds to dog class, yellow cell corresponds to bicycle class and pink cell corresponds to the car class [14].

Secondly grid cell is responsible to predict the B bounding boxes and the confidence scores for the bounding boxes if it is the center point of the object. However, it is difficult for the grid cells to know if they are being the center of the objects, there might be multiple cells think that they are the center of the same object in an image. Therefore, in the image of bounding boxes + confidence in Figure 1.11, many different bounding boxes were generated. Confidence score reflects the model's confident on the box if the box contains objects and how accurate is the predicted box. The confidence score is defined as $\Pr(object) * IoU_{pred}^{truth}$, while $\Pr(object)$ is referring to the probability of an object exist. The confidence score is zero when there is no object available in that grid cell, otherwise the confidence score is required to be equal to the Intersection over Union (IoU) between the ground truth and predicted bounding box [14]. The details of the IoU are described at Section 3.3.3 of this paper. Every bounding box is consisting of five predictions denoted as x, y, w, h, and confidence. The x and y are representing the x-axis and y-axis coordinates of the bounding box center relative to the grid cell bounds. Next, w and h are representing the width and height of the bounding boxes relative to the entire image.

$$\mathbf{Pr}\left(\frac{\boldsymbol{Class_i}}{\boldsymbol{Object}}\right) * \ \mathbf{Pr}(\boldsymbol{object}) * \ \boldsymbol{IoU_{pred}^{truth}} = \mathbf{Pr}(\boldsymbol{Class_i}) * \ \boldsymbol{IoU_{pred}^{truth}} \qquad \mathbf{(1.1)}$$

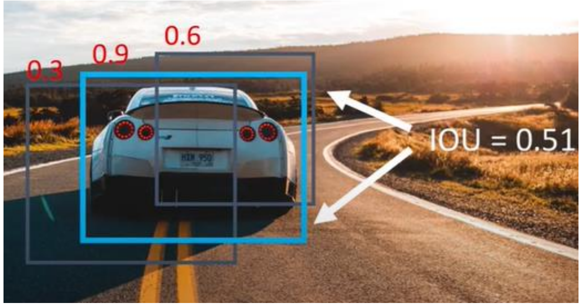


Figure 1.12 Non-max suppression on cleaning the bounding boxes.

During testing, the conditional class probabilities is multiplied with the confidence score of individual bounding box to calculate the class-specific confidence score for each bounding boxes as shown in Equation (1.1) [14]. The class-specific confidence score is telling the probability of that class appear in the bounding box and how good is the predicted bounding box matching the ground truth bounding box. Next, non-max suppression algorithm is applied, to obtain the only unique bounding box which perfectly fits the object left. Firstly, non-max suppression algorithm will discard the predicted bounding boxes that have IoU score lower that the IoU threshold because it means the predicted bounding box is not matching the ground truth box. Next, if there are more than one bounding boxes for one object, we will find the IoU between the highest probability scoring box to another box which is 0.6 scoring box in Figure 1.12. If the IoU is greater than a threshold which is 0.50 in this example, the lower scoring box is removed.

1.1.4.1 Strength and weakness of YOLO

The most representing strength of YOLO is its fast speed. Reframing object detection into a regression problem eliminates complicated pipeline because neural network was simply run on a new image at test time for detections. According to Redmon et al., the YOLO base network is having speed of 45 frames per second (fps) when executing on a Titan X GPU platform without batch processing. The performance can be further improved to 150 fps, means that live video in real-time can be processed with latency shorter than 25 milliseconds. In terms of mean average precision (mAP), YOLO can perform twice better as compared to other real-time systems. Secondly, YOLO sees the full image during training and testing phase which enable it to encode the contextual information of classes implicitly as well as the appearance. Therefore, YOLO reduce the background errors as compared to Fast R-CNN which always wrongly classify background patches in an image due to its architecture which cannot see larger context. Lastly, YOLO can learn the generalizable representations of objects and not likely to deconstruct when receive unexpected inputs. The experiment of training various models on natural images and being tasted on artworks shows that YOLO beats other top object detector like DPM and R-CNN [14].

Even though YOLO has short inference time, it is having difficulty in locating small objects precisely and its accuracy is low as compared to the state-of-the-art detectors [9]. Secondly, behaviour of YOLO learning on bounding boxes prediction from data cause it to be weak in generalizing to object in unseen or unusual configurations. The last limitation is the loss function of YOLO treats error in small and big bounding boxes similarly. Tiny error in a large box is safe to be neglected but it brings huge impact to Intersection over Union (IOU) in small box. Therefore, the main error source in YOLO is localizations [14].

1.1.4.2 Comparison of YOLO to other deep learning model

DPM is utilizing sliding window method for object detection. It is using a disjoint pipeline for static features extraction, regions classification, bounding boxes prediction for high scoring regions. But these disparate parts were replaced by a single CNN in YOLO, where the CNN can extract features, predict bounding box, perform non-maximal suppression and reasoning contextual parallelly. Furthermore, CNN trains the features in-line and perform features optimization for the detection task while DPM is using static features. In summary, the unified architecture of YOLO is better than DPM in terms of speed and accuracy [14].

Region proposals is implemented in R-CNN instead of sliding window for object detection. Potential bounding boxes generation, features extraction, bounding boxes scoring, bounding boxes adjustment and duplicate detections were handled by Selective Search, convolutional network, SVM, linear model and non-max suppression respectively. Every stage of this complicated pipeline requires independent tuning precisely and causing slow system which cause 40 seconds time cost per image at test time. There are some similarities between YOLO and R-CNN, potential bounding boxes are proposed by every grid cell and these boxes were scored using convolutional features. In terms of bounding boxes proposal, YOLO is using spatial constraints on the grid cell to mitigate multiple detections of the same object. YOLO also proposed lesser number of bounding boxes, which is 98 boxes per image as compared to approximately 2000 boxes per image from Selective Search. Lastly, the individual components are combined into a jointly optimized model [14].

As compared to the fast detectors like Fast R-CNN and Faster R-CNN which is focusing on improving the R-CNN speed by replacing Selective Search with CNN to propose regions and share computation. Even both fast detectors have better speed and accuracy than R-CNN, but they are still performing weak in real time due to its poor performance on small objects identification and localization and long inference time [13]. Besides, initiatives such as HOG computation, cascading and running computation on GPUs have been used to speed up the DPM pipeline, but only 30Hz DPM can be used in real-time environment. YOLO is using a single pipeline and it is fast by design, therefore it does not need to optimize the individual component of a large pipeline as Fast and Faster R-CNN [14].

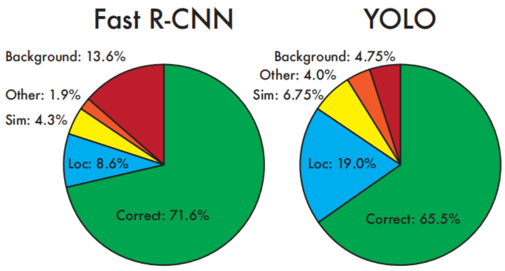


Figure 1.13 Breakdown result of VOC 2007 analysis on Fast R-CNN and YOLO [14].

Table 1.2 Error type of VOC 2007 analysis and reason [14].

Error type	Reason
Correct	correct class and IoU > 0.5
Localization	correct class, 0.1 < IoU < 0.5
Similar	class is similar, IoU > 0.1
Other	class is wrong, IoU > 0.1
Background	IoU < 0.1 for any object

Redmon et al. have conducted VOC 2007 error analysis to study the differences between YOLO and Fast R-CNN as the state-of-the-art detectors because Fast R-CNN is one of the best detectors on PASCAL. The VOC 2007 analysis error type breakdown across 20 classes is represented in pie chart in Figure 1.13. Each prediction can be classified to results shown in Table 1.2 according to its correctness and IoU. For example, when the class is classified correctly, it is classified as Correct when IoU > 0.5 and classified as Localization (denoted as Loc in Figure 1.13) when 0.1 < IoU < 0.5. It is observable where YOLO is bad in object localization as localization error is the major error in YOLO as compared to similar error, background error and other error. In Fast R-CNN, 13.6% of background error is the major error, means that it gives false positive to image that does not have any objects. Fast R-CNN is having lesser localization error but causing approximate tripe more background error than YOLO. In summary, YOLO is creating more localization errors, but it is having low false positive prediction on background [14].

1.2 Motivation

In this research, we are focusing on building a deep-learning based face mask detector as wearing face mask have been a new norm globally due to the COVID-19 pandemic. Wearing face mask is prove as an effective measure to control the spread of corona virus, however not everyone is willing to wear face mask due to various reason such as feeling troublesome, lazy, or uncomfortable. Therefore, it has always been a challenge to ensure people wearing face mask in public area. As a solution to this issue, autonomous face mask detector shall be implemented to monitor the mask wearing status. The two-stage object detection such as R-CNN before YOLO introduced is having issue of large computation cost and need to be divided into several stages and hence increase the difficulty of system speed optimization [15]. There is various deep learning algorithm which can be used as the head of object detector, due to the consideration of performance, we are building our face mask detector using YOLO series of one-stage object detection, which are YOLOv4 and YOLOv5 that proposed in year 2020 and evaluate the mean average precision (mAP) and inference time of model.

The reference papers are reviewed and further analyzed based on their strength and weakness. According to the results and analysis conducted by each author, it is observable that several algorithms to build face mask detector have been built and performance was evaluated. However, it is observable where Bags-of-Special (BoS) enhancement have been applied in several proposed algorithm for the purpose of performance improvement. The plugin modules and post-processing layers are increasing the complexity and size of the algorithm. Therefore, we are proposing an algorithm which use Bags-of-Freebies such as MixUp technique to boost the model performance.

In summary, the following list is the motivations of this project:

1. Face mask detector performance in terms of mean average precision (mAP) and inference time can be improved by using YOLOv4 and YOLOv5 of latest one-

stage object detection which prove to perform better than two-stage object detector.

2. Performance enhancement through adding plugin modules and post-processing layers which increase the complexity and size of algorithm can be avoided by using Bags-of-Freebies technique such as MixUp technique to boost the model performance.

1.3 Problem Statements

The problem statements of this project are listed as below:

1. Old and unsuitable architecture of deep learning model cause the lower performance of face mask detector built in terms of mean average precision (mAP) and inference time which is not sufficing the standard of real-time working application.

2. The inference time of face mask detector model is not evaluated.

3. Performance enhancement in form of Bags-of-Special (BoS) such as adding more plugin modules and post-processing approach is increasing the complexity and size of the algorithm.

1.4 Research Aim and Objective

The main aim of this project is to propose a high-performance YOLOv4 and YOLOv5 face mask detector using data Mixup of Bags-of-Freebies (BoF) technique which can contribute to high mean accuracy precision (mAP) and short inference time.

The objectives of the project include:

1. To build a face mask detector by using state-of-the-art high-performance deep learning model, YOLOv4 and YOLOv5 which can contribute to high mean accuracy precision (mAP) and short inference time that suffice the requirements to be working in a real-time environment.

2. To evaluate the inference time of face mask detector built.

3. To propose an algorithm that implements the data MixUp technique to enhance the data and hence boost the performance of the algorithm without adding much weight to the complexity and size of the algorithm.

1.5 Research Scope

The project scopes are listed as follows:

a) This project utilized datasets from Kaggle where the differences in datasets chosen might affect the results.

b) The YOLOv4 Darknet and YOLOv5 PyTorch deep learning model is selected to be evaluated in this project.

c) This project is performing annotation manually using CVAT which might cause differences in results based on individual annotation skill.

d) Roboflow and Google Colab is used as the tool for the process from dataset upload to online training and testing in this project.

1.6 Chapter Summary

This project is arranged as 5 chapters organizations, Chapter 1 is introducing the research background study such as the relationship between COVID-19 and face mask, application of face mask detector, basic information of object detection and comparison between R-CNN as two-stage object detection and YOLO series as one-stage object detection. Moreover, this chapter also stated the motivation, problem statements, research aim, research objectives and research scopes of this project.

Then, Chapter 2 is providing a literature of background information for more advanced knowledge exposure to assist readers on understandings this project context. This chapter is describing the datasets used in reference research, data annotation, data augmentation, architecture of YOLOv4 Darknet and YOLOv5 PyTorch from YOLO series algorithm.

Furthermore, Chapter 3 is illustrating the methodology of this research. Firstly, the solution in the problem domain have shown the general methodology of this project in form of flowchart. Secondly, the solution details are describing the details of the procedure to conduct the experiment. Thirdly, the experiment setup details such as the Bags-of-Freebies (BoF), Bags-of-Specials (BoS), dataset, tools to perform data MixUp, annotation, training, and testing were stated. Lastly, the research planning is represented by using Gantt Chart which shows the research activities and its respective timely systematically.

Next, results and discussion of this project is shown in the Chapter 4. Since this is Project 1 and the experiment is not completed, the interim result is discussed. Lastly, the Chapter 4 is the conclusion which discuss expected result of this paper.

REFERENCES

[1] A. Junaidi and J. Lasama, *Object Detection for Using Mask in COVID-19 Pandemic with Faster R_CNN Inception V2 Algorithm*, vol. 746 LNEE. Springer Singapore, 2021.

[2] C. Li, J. Cao, and X. Zhang, “Robust deep learning method to detect face masks,” *PervasiveHealth Pervasive Comput. Technol. Healthc.*, no. 1, pp. 74–77, 2020, doi: 10.1145/3421766.3421768.

[3] R. R. Mahurkar and N. G. Gadge, “Real-time Covid-19 Face Mask Detection with YOLOv4,” *Proc. 2nd Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2021*, pp. 1250–1255, 2021, doi: 10.1109/ICESC51422.2021.9533008.

[4] World Health Organization, “WHO Coronavirus (COVID-19) Dashboard | WHO Coronavirus (COVID-19) Dashboard With Vaccination Data.” https://covid19.who.int/ (accessed Dec. 15, 2021).

[5] Vinay Sharma, “Face Mask Detection using YOLOv5 for COVID-19,” pp. 10–14, 2018, [Online]. Available: https://scholarworks.calstate.edu/downloads/wp988p69r?locale=en.

[6] E. K. J. Dzisi and O. A. Dei, “Adherence to social distancing and wearing of masks within public transportation during the COVID 19 pandemic,” *Transp. Res. Interdiscip. Perspect.*, vol. 7, p. 100191, 2020, doi: 10.1016/j.trip.2020.100191.

[7] COVIDNOW, “COVID-19 Cases in Malaysia - COVIDNOW.” https://covidnow.moh.gov.my/cases (accessed Dec. 15, 2021).

[8] A. Kumar, A. Kalia, A. Sharma, and M. Kaushal, “A hybrid tiny YOLO v4-SPP module based improved face mask detection vision system,” *J. Ambient Intell. Humaniz. Comput.*, no. 0123456789, 2021, doi: 10.1007/s12652-021-03541-x.

[9] Mohana and H. V. Ravish Aradhya, “Object detection and tracking using deep learning and artificial intelligence for video surveillance applications,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 12, pp. 517–530, 2019.

[10] S. Mane and S. Mangale, “Moving Object Detection and Tracking Using Convolutional Neural Networks,” *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. ICICCS 2018*, no. Iciccs, pp. 1809–1813, 2019, doi: 10.1109/ICCONS.2018.8662921.

[11] A. Raghunandan, Mohana, P. Raghav, and H. V. R. Aradhya, “Object Detection Algorithms for Video Surveillance Applications,” *Proc. 2018 IEEE Int. Conf. Commun. Signal Process. ICCSP 2018*, no. April, pp. 563–568, 2018, doi: 10.1109/ICCSP.2018.8524461.

[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” 2020, [Online]. Available: http://arxiv.org/abs/2004.10934.

[13] Q. Zhang, C. Wan, W. Han, and S. Bian, “Towards a fast and accurate road object detection algorithm based on convolutional neural networks,” *J. Electron. Imaging*, vol. 27, no. 05, p. 1, 2018, doi: 10.1117/1.jei.27.5.053005.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[15] D. Thuan, “Evolution of Yolo Algorithm and Yolov5: the State-of-the-Art Object Detection Algorithm,” p. 61, 2021.

[16] Atta-ur-Rahman and V. U. Ahmad, “Retinane,” *13C-NMR Nat. Prod.*, pp. 30–33, 1992, doi: 10.1007/978-1-4615-3288-0_5.

[17] W. Jian and L. Lang, “Face mask detection based on Transfer learning and PP-YOLO,” *2021 IEEE 2nd Int. Conf. Big Data, Artif. Intell. Internet Things Eng. ICBAIE 2021*, no. Icbaie, pp. 106–109, 2021, doi: 10.1109/ICBAIE52039.2021.9389953.

[18] T. Araújo, G. Aresta, A. Galdran, P. Costa, A. M. Mendonça, and A. Campilho, “Uolo - Automatic object detection and segmentation in biomedical images,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11045 LNCS, pp. 165–173, 2018, doi: 10.1007/978-3-030-00889-5_19.

[19] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, and R. Wang, “DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection,” *Inf. Sci. (Ny).*, vol. 522, pp. 241–258, 2020, doi: 10.1016/j.ins.2020.02.067.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8691 LNCS, no. PART 3, pp. 346–361, 2014, doi: 10.1007/978-3-319-10578-9_23.

[21] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path Aggregation Network for Instance Segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8759–8768, 2018, doi: 10.1109/CVPR.2018.00913.

[22] R. Gupta, D. Kumar, K. Jaiswal, and N. Vishwakarma, “Garbage Detection based on Deep Learning,” *SSRN Electron. J.*, 2021, doi: 10.2139/ssrn.3884955.

[23] A. Aggar, A. A. Rahem, and M. Zaiter, “Iraqi Traffic Signs Detection Based On Yolov5,” pp. 5–9, 2021, doi: 10.1109/aca52198.2021.9626821.

[24] M. Kilinc and U. Uludag, “Gender identification from face images,” no. Icoei, pp. 1–4, 2012, doi: 10.1109/siu.2012.6204517.

[25] P. Henderson and V. Ferrari, “End-to-end training of object class detectors for mean average precision,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10115 LNCS, pp. 198–213, 2017, doi: 10.1007/978-3-319-54193-8_13.

[26] K. Kishida, “Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments,” *NII Tech. Reports*, vol. 2005, no. 14, pp. 1–19, 2005.