Face Mask Protocol Detection Using YOLOv8

Befekadu Eshetu Worku

A Thesis Submitted to the Department of Computer Science and Engineering,

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master’s in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

February 2024

Adama, Ethiopia

Face Mask Protocol Detection Using YOLOv8

Befekadu Eshetu Worku

Advisor: Mesfin Abebe (Ph.D.)

A Thesis Submitted to the Department of Computer Science and Engineering,

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

February 2024

Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled "**Face Mask Protocol Detection Using YOLOv8**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation

Befekadu Eshetu	____________________	____________________
Name of student	Signature	Date

RECOMMENDATION

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled "**Face Mask Protocol Detection Using YOLOv8**" prepared under my/our guidance by **Befekadu Eshetu** submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science and Engineering.

Therefore, I/we recommend the submission of the revised version of the thesis to the department following the applicable procedures.

Mesfin Abebe (Ph.D.)	______________________	___________________
Major Advisor	Signature	Date
_______________________	______________________	___________________
Co-advisor	Signature	Date

APPROVAL PAGE

I/we, the advisors of the thesis entitled "**Face Mask Protocol Detection Using YOLOv8**" and developed by **Befekadu Eshetu** hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Mesfin Abebe (Ph.D.) ______________________ ___________________

Major Advisor Signature Date

We, the undersigned, members of the Board of Examiners of the thesis by **Befekadu Eshetu** have read and evaluated the thesis entitled "**Face Mask Protocol Detection Using YOLOv8**" " and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

_______________________ ______________________ ___________________

Chairperson Signature Date

_______________________ ______________________ ___________________

Internal Examiner Signature Date

_______________________ ______________________ ___________________

External Examiner Signature Date

Finally, approval and acceptance of the thesis proposal are contingent upon the submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_______________________ __________________ ___________________

Department Head Signature Date

_______________________ ___________________ ___________________

School Dean Signature Date

_______________________ ___________________ ___________________

Office of Postgraduate Studies, Dean Signature Date

ACKNOWLEDGMENT

First and foremost, I would want to convey my sincere thankfulness to Almighty God for his unwavering support throughout this research, from its inception to its triumphant completion in such a lovely way.

I extend my deepest gratitude to my advisor Dr. Mesfin Abebe for his invaluable guidance and inspiration throughout this research journey. His unwavering support and insightful advice fueled my passion for this project, ultimately leading to its successful completion.

I am deeply grateful to the students and staff of ASTU for their invaluable support during the data collection process. Their dedication, guidance, and willingness to share knowledge were instrumental in the success of this research. I would also like to express my sincere appreciation to my family members for their unwavering encouragement and support throughout the journey of completing this thesis. Finally, my heartfelt thanks go to all my friends who offered their unwavering support throughout my studies.

TABLE OF CONTENTS

DECLARATION iii
RECOMMENDATION iv
APPROVAL PAGE v
ACKNOWLEDGMENT vi
LIST OF TABLES v
LIST OF FIGURES vi
LIST OF ABBREVIATIONS AND ACRONYMS vii
ABSTRACT ix
CHAPTER ONE 1
1. INTRODUCTION 1
1.1. Background of the study 1
1.2 Motivation of the study 3
1.3. Statement of the problem 4
1.4 Research questions 5
1.5. Objectives of the study 5
1.5.1 General objective 5
1.5.2. Specific objectives 5
1.6 Significance of the study 6
1.7. Delimitation Scope 6
1.7.1. Scope of the study 6
1.7.2. Limitation of the study 6
1.8 Organization of the thesis 7
CHAPTER TWO 8
2. LITERATURE REVIEW AND RELATED WORK 8
2.1 Introduction 8
2.2 Literature Review Model 13
2.3 Overview of detection 14

2.3.1. Face Detection 14
2.3.2 Face Mask Detection 15
2.4 Data pre-processing tools 15
2.5 Model Selection 17
2.6 Related Work 20
CHAPTER THREE 24
3. RESEARCH METHODOLOGY 24
3.1. Research Design 24
3.2. Data Collection Methods 25
3.3. Data Pre-processing Technique 27
3.5. Evaluation Method 27
3.5.1 Classification Accuracy 28
3.5.2. Confusion Matrix 28
3.5.3. F1 Score 28
3.6. Design and Development Tools 29
3.6.1. Hardware tools 29
3.6.2. Software tools 29
CHAPTER FOUR 31
4. PROPOSED SOLUTION FOR FACE MASK PROTOCOL DETECTION 31
4.1 Chapter Overview 31
4.2 Proposed Model Architecture 31
4.3 Understanding of The Data 32
4.3.1 Data Assessment 32
4.3.2 Data Exploration 33
4.4 Face Mask Image Preprocessing 33
4.4.1 Balancing Face Mask Image 34
4.4.2 Processing Method 34
4.4.3 Select Model 35
4.4.4 Bounding Box and Removing Background 36
4.4.5 Assign Classes 36
4.4.6 Save Labeled Dataset 37

4.5 Detection Model Building ... 38
4.6 Model Evaluation and Prototype ... 40
4.7 Prototype Development ... 40
CHAPTER FIVE ... 42
5. IMPLEMENTATION AND EXPERIMENTATION ... 42
5.1 Chapter Overview ... 42
5.2 Working Environment ... 42
5.2.1 Computers with Hardware Functions ... 42
5.2.2 Computers with Software Configuration ... 42
5.3. Setup Environments ... 43
5.4. Training Procedure ... 43
5.5. Dataset Processing ... 44
5.6. Preprocessing Implementations ... 45
5.7. Detection Models Implementation ... 45
5.7.1 Model Training ... 46
5.8. Implementation of Model Evaluation ... 47
CHAPTER SIX ... 48
6. RESULTS, EVALUATION, AND DISCUSSIONS ... 48
6.1 Chapter Overview ... 48
6.2 Face Mask Protocol Detection Image Preprocessing Result ... 48
6.2.1 Collected Datasets Image Result ... 48
6.2.2 Public datasets Image Result ... 49
6.3 Experimental Setup ... 49
6.4 Model Learning Curves ... 50
6.4.1 Precision-Confidence Curve ... 50
6.4.2 Recall-Confidence Curve ... 51
6.4.3 Precision-Recall Curve ... 51
6.4.4 F1_curve ... 52
6.5 Models Performance and Evaluation Result ... 53
6.5.1 Face Mask Protocol Detection Model Result ... 53
6.5.2 Face Mask Protocol Detection Confusion_Matrix ... 55

6.5.3 Face mask protocol detection Performance metrics reports 56
6.6 Tuning Hyperparameters Result 57
6.7 Discussion 58
CHAPTER SEVEN 60
7. CONCLUSION AND FUTURE WORK 60
7.1 Conclusion 60
7.2 Future Work 61
REFERENCES 62
APPENDIX 70
Appendix A: Sample Code 70
Appendix B: Training Sample 71
Appendix C: The Best Performance Image Result in the Collected Dataset 72
Appendix D: The Best Performance Image Result in the Public Dataset 73
Appendix E: On Research Question one Experiment result 74
Appendix F: Face Landmark Experiment Result 74
Appendix G: Text based detection results 75

LIST OF TABLES

Table 2.1 Summary of related work 21
Table 3.1 Hardware tool used for this study implementation... 29
Table 5.1 Types of the YOLO model 47
Table 6.1 In Collected Dataset Performance metrics reports 57
Table 6.2 In Public Dataset Performance metrics reports... 57
Table 6.3 In the collected dataset best Performance... 58
Table 6.4 In the Public dataset best Performance... 58

LIST OF FIGURES

Figure 2.1 The architecture of the transfer learning model 11
Figure 2.2 Sample of IoT (Bell, 2021)........ 12
Figure 2.3 YOLO evolution timeline........ 19
Figure 3.1 Flowchart Diagram of the Research........ 25
Figure 3.2 Data set sample Face mask wearing........ 25
Figure 3.3 Data set sample Incorrect form wearing a face mask........ 26
Figure 3.4 Data set sample Incorrect form wearing a face mask........ 26
Figure 3.5 Data set sample Not wearing a face mask........ 26
Figure 4.1 Face mask detection model architecture........ 32
Figure 4.2 Dataset Preprocessing method........ 33
Figure 4.3 Checking balancing Face mask image........ 34
Figure 4.4 LabelIng Multi-modal data annotation running methods........ 35
Figure 4.5 Bounding box in labeling multi-modal data annotation........ 36
Figure 4.6 Assign class not wearing a face mask........ 37
Figure4.7 Assign class in correct form wearing a facemask........ 37
Figure 4.8 Assign class face mask-wearing........ 37
Figure 4.9 Saved labeled folder........ 38
Figure 4.10 Face Mask Detection Model Building........ 39
Figure 4.11 Prototype Development Architecture........ 41
Figure 6.1 In 20epoch YOLOv8s Model........ 48
Figure 6.2 In 20epoch YOLOv8m Model........ 48
Figure 6.3 In 20epoch YOLOv8x Model........ 48
Figure 6.4 In 50epoch YOLOv8s Model........ 49
Figure 6.5 In 50epoch YOLOv8m Model........ 49
Figure 6.6 In 50epoch YOLOv8x Model........ 49
Figure 6.7 In Collected dataset 20 epoch YOLOv8s Model........ 50
Figure 6.8 In Public Dataset 50 epoch YOLOv8m Model........ 50
Figure 6.9 In Collected Dataset 20 epoch YOLOv8s Model........ 51
Figure 6.10 In Public Dataset 50 epoch YOLOv8m Model........ 51
Figure 6.11 In Collected Dataset 20 epoch YOLOv8s Model........ 52
Figure 6.12 In Public Dataset 50 epoch YOLOv8m Model........ 52
Figure 6.13 In Collected Dataset20 epoch YOLOv8s Model........ 53
Figure 6.14 In Public Dataset 20 epoch YOLOv8m Model........ 53
Figure 6.15 In Collected Dataset 20 epoch YOLOv8s Model........ 54
Figure 6.16 In Public Dataset 50 epoch YOLOv8m Model........ 54
Figure 6.17 In Collected Dataset 20 epoch YOLOv8s Model........ 55
Figure 6.18 In Public Dataset 50 epoch YOLOv8m Model........ 56

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neuron Network
CCTV	Closed Circuit Television
CLS_LOSS	Class Loss
CNN	Convolutional Neural Network
COVID-19	Coronavirus Disease
CPU	Central Processing Unit
CSS	Cascading Style Sheets
CV	Computer Vision
DFL_LOSS	Distribution Focal Loss
DL	Deep Learning
FLD	Forming Limit Diagram
FPS	Frames per Second
GB	gigabytes
GHZ	gigahertz
GIOU	Generalized Intersection over Union
GPS	Global Positioning System
GPU	Graphical Processing Units
H1N1	Swine flu
HD	high-definition
HPAI	Highly Pathogenic Avian Influenza
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transfer Protocol
IOT	Internet of Things
IP	Image Processing
MAP	Mean Average Precision

MB	megabyte
ML	Machine Learning
RAM	Random Access Memory
R-CNN	Region-Based Convolutional Neural Network
RFID	Radio Frequency Identification
RNN	Recurrent Neural Networks
ROC CURVES	receiver operating characteristic curve
SARS	Severe Acute Respiratory Syndrome
SARS-COV-2	Severe acute respiratory syndrome coronavirus 2
SSD	Single Shot Detector
TB	tuberculosis
TL	Transfer Learning
VGG	Visual Geometry Group
VOC	Volatile Organic Compounds
WFH	Working From Home
Wi-Fi	Wireless Fidelity
WSN	Wireless sensor networks
XML	extensible markup language
YOLO	You Only Look Once

ABSTRACT

A facemask is crucial for choosing the right type and ensuring its effectiveness in its intended use. By prioritizing proper coverage, filtration, fit, comfort, and durability, we can maximize the benefit of facemasks in protecting individuals and communities from the spread of airborne illnesses and some other dust. Facemask protocol detection has emerged as a significant technology in response to the global pandemic, aiming to curb the spread of infectious diseases. This technology has made advancements in recent years, offering promising solutions for public health and safety. This study uses machine-learning approaches to build a facemask protocol detection, and this research was to develop, test, and evaluate machine-learning methods for predicting the facemask wearing, incorrect form wearing, and not wearing a facemask. To carry out the study successfully, it was essential to detect the facemask protocol. Bounding box, passing class balance the dataset, model training using YOLOv8s, YOLOv8m, and YOLOv8x model testing, and finally comparing models using evaluation metrics are also methods used to implement and design for facemask protocol detection. We used collected datasets and downloaded them from Kaggle public datasets for this research. Each balanced severed dataset class is divided into three assigning classes: face mask-wearing, incorrect form-wearing, and not wearing facemasks. The experiment was conducted on three dataset types, obtaining three models for each dataset. When compared to YOLOv8s, YOLOv8m, YOLOv8x models. The YOLOv8s model produced greater test accuracy on the collected dataset, and in a public datasets, YOLOv8m will be more good accuracy. In the collected dataset, Precision-Recall mAP is 0.995, F1 Confidence 0.99, and Precision Confidence 0.960, in 20 epochs. In the public datasets, Precision-Recalll mAP is 0.994, F1 Confidence 0.98, and Precision Confidence 0.936, epoch.

Keywords: - *Face mask detection, facemask protocol, Face landmarks, YOLO, Multi-modal data annotation.*

CHAPTER ONE

1. INTRODUCTION

In this chapter, we explain the introductory part of this study including the background of the study, a statement of the problems, research questions, motivation of the study, objective of the study, scope, limitations, and organization of the study.

1.1. Background of the study

A face mask is a device that is plastic metal or cloth that covers part of the face, The first mask was aimed at stopping a bad smell in the 1700s (Goh et al., 2020). Facemask uses a fund of services, for medical it covers the mouth and nose and is special to reduce saliva and for the general anesthetic spread of infectious agents such as viruses or bacteria, and it is used for medical surgery, For maximum protection against airborne particles, opt for an N95 mask. While the surgical mask offered the best defense among the three tested, its efficacy remains limited. Consider alternatives like N95s for situations demanding higher levels of safety. (Bowen, 2010).

The word 'mask' has a surprisingly spooky past. It comes from the Medieval Latin word 'masca,' which means 'specter or nightmare.' This connection hints at the way masks have often been used to disguise or hide our true selves, both for fun and for more sinister purposes(Ike et al., 2020). Though commonly associated with preventing infections, the surgical mask was first introduced in the 1890s for a different purpose: shielding surgeons and assistants from splashes of blood and bodily fluids during operations. While it provided crucial protection from this biological hazard, it wasn't initially intended to protect medical personnel from airborne pathogens(Goh et al., 2020). The image of surgeons masked and gloved is ubiquitous, but these protective measures weren't always standard practice. It was only in the 1920s, in operating rooms across Germany and the USA that surgical masks emerged as a vital weapon against surgical site contamination. Initially focused on safeguarding procedures, their impact soon extended beyond the sterile walls, playing a key role in curbing the spread of droplet and airborne viral respiratory diseases, especially in endoscopic and smaller surgeries. (Matuschek et al., 2020).

Face masks used for sports cover all the face parts it will be playing hockey goalkeeper and ball catcher, Face masks in the work environment prevent you from breathing bad air or from spreading germs, thick substances, or dust which you spread on your face to protect the face when you are in a dangerous situation.

Any sickness produced by a bacterium that spreads through the air is considered an airborne ailment. Airborne infections caused by bacteria, viruses, and fungi are numerous and have therapeutic significance. Sneezing, coughing, spraying liquids, spreading dust, or any other action that produces aerosolized particles can all transmit these pathogens. It's critical to understand that airborne illnesses(Ather & Edemekong, 2019)

A cloth face mask has been made from basic textiles, most often cotton or cloth. A mask is an essential piece of personal protective equipment for front-line healthcare workers, and it is probably the most potent psychological emblem in the public sector. (Goh et al., 2020). Since its emergence in 2019, COVID-19, caused by the SARS-CoV-2 virus, has evolved into a global pandemic, affecting billions of lives and reshaping healthcare systems worldwide(Wiersinga & Prescott, 2020). The common symptoms include muscle aches, fatigue, nausea, vomiting, and diarrhea. People may also have a loss or change in their sense of taste and smell (Wiersinga & Prescott, 2020).

Deep learning, a subset of AI, mimics human learning by training computers on vast amounts of data. These models can then categorize and understand information in various formats, from images and text to live streams and audio. Models are trained to utilize a huge quantity of labeled data and multi-layer neural networks Deep learning has produced excellent results on a variety of problems in recent years, including face detection, visual recognition, speech recognition, classification, and natural language processing (Tian, 2020).

Detection is the noticing or discovering something or the action of accessing information without specific cooperation from the sender, Handcrafted features and shallow trainable structures are the foundation of traditional object identification techniques. (Zhao et al., 2019). While humans excel at recognizing familiar faces, their ability to identify individuals in diverse contexts pales compared to the precision of machines. This is where automatic face detection systems step in, playing a crucial role in applications like face recognition, expression analysis, and human-computer interaction.(Kumar et al., 2019). Face detection gives you the data you need to do things like enhance selfies and portraits, or create characters from a person's photo. Face detection is the stepping stone to all facial analysis algorithms, including face alignment, face modeling, face relighting, face recognition, face verification/authentication, head pose tracking, facial expression tracking/recognition, gender/age recognition, and many more(Kumar et al., 2019).

The Internet of Things (IoT) is a network of interconnected computer devices or digital machines or smart devices that a data stored in the cloud. The term "Internet of Things" describes a kind of network that uses information-sensing devices to link anything to the Internet by predetermined protocols. This allows for information exchange and communications to be conducted to accomplish intelligent identification, location, tracking, monitoring, and administration.(Patel et al., 2016).In our nation or university IoT can be used in so many services one of them is a CCTV camera as a security mechanism. That CCTV camera connects to the internet and is stored in a data center it will be managed few persons. The CCTV camera is in many places and connects with wi-fi or wired network and in one office live video stream can be tracked. Our university or Adama Hospital can use four types of CCTV cameras they are dome, bullet, outdoor, and ptz Those CCTV cameras assign an IP address and it is maintained or operated by an IP address.

To detect a face mask, the machine first needs to understand face detection because if it comes the accurate face mask detection. The recent approaches to face mask detection and recognition are based on detecting facial annotation to facemask it is accomplished by training multiple Convolutional Neural Networks (CNN) for detecting different facial landmarks and face mask detection.

1.2 Motivation of the study

The use of facemasks is advised for illnesses spread through droplets, while respirators are recommended for respiratory aerosols. However, there is inconsistency in recommendations and terminology across guidelines. The conventional understanding of droplet and airborne transmission in clinical practice has evolved, revealing a complexity beyond previous assumptions. (Liang et al., 2020). The general public has access to a variety of face masks designed for protection against inhaling dust, pollutants, allergens, and pathogenic organisms. Recent news reports highlight the extensive utilization of face masks as a preventive measure against various health threats, such as the Swine flu (H1N1), Severe Acute Respiratory Distress Syndrome (SARS), and Highly Pathogenic Avian Influenza (HPAI) virus outbreaks in Asia. Additionally, face masks have been employed to safeguard individuals from inhaling dust generated by the collapse of the World Trade Center(Bowen, 2010). Wearing a face mask plays a bubbly role in controlling the spread of the virus and bacteria. But most people are not wearing face masks in public places, libraries, and hospitals which increases the spread of viruses and bacteria. it is required to identify everyone wearing a face mask in public places, such as libraries and hospitals, as well as anyone who is not wearing one or is not wearing it correctly.

Based on the identified problems the researcher was motivated to work on facemask detection using machine-learning techniques that can analyze and classify the face mask protocol properly based on the collected dataset and public dataset. For domain experts, it helps to identify whether wearing a face mask and it covering incorrectly or in the protocol, make fast decision-making if it is not wearing a face mask, and help to give proper treatment, which reduces the transmission of infections and diseases transmitted through droplets of vires and some airborne diseases.

1.3.Statement of the problem

A face mask can play a big part in preventing the virus from spreading. However, the majority of people do not utilize face masks in social places, which contributes to the transmission of infections. Many people are required to wear face masks during hospitalization or library coronavirus protocols. Object detection algorithms are used to identify people with and without face masks in public areas, including libraries, hospitals, and other areas where people congregate. If employers out the customer will remove the face mask or it is not taken incorrectly that means, it covered only the chin and the clasp on-ear, it covered the only chin, mouth it is not covered nose than the clasp on-ear, the correct form of the taken mask will be it covers the chin, mouth, and nose than the clasp on-ear.

To detect a face mask, the machine first needs to understand face detection because the dataset's image or video stream includes all body parts or above the waist but the face mask will be worn only on the face to maximize the accuracy of face mask detection first detect the face part. Face masks are not recorded in most face mask detection systems, but the algorithm is run on each frame from a video, image, or streaming camera. The more recent approaches to face mask detection are based on detecting facial annotation i.e., nose, mouth, right eye, and left eye which means first we detect the face then it will detect the face mask i.e., nose, mouth, the chin will be covered by mask it is accomplished by training multiple Convolutional Neural Networks (CNN) for detecting different facial annotation and face mask detection.

1.4 Research questions

Based on these facemask detection the following research questions will be attempted to be answered.

Research question 1: How the impact of dataset type and diversity on the performance of facemask detection models?

Research question 2: Which are the necessary features to prepare an effective facemask protocol dataset in facial annotation?

Research question 3: Which deep learning algorithm is more accurate for the prediction of Facemask protocol detection?

1.5. Objectives of the study

1.5.1 General objective

The general objective of this study is to develop Facemask protocol detection using yolov8.

1.5.2. Specific objectives

The overall goal of this research will be the following specific objectives.

- To review related research have a better understanding of the subject and other people's work.
- To collecting and preparing datasets for face and facemask detection.
- To evaluating the detection algorithms and performance.
- To design and develop an end-to-end deep learning model for facemask detection.
- To explore methods for detecting improper mask usage, such as masks not covering the nose or mouth adequately.

- To develop annotation guidelines and standards for labeling key facial landmarks, such as eyes, nose, mouth, and facial contours.

1.6 Significance of the study

Large-scale data is freely available and being rapidly produced in the CNN era, therefore models should not be limited to small-scale applications. That is, models should be used in a variety of contexts and as a step between other applications. This study develops a model that can be scaled to big datasets and trained on them. Face mask detection is a critical problem with several applications in image and streaming camera applications. Some of these programs are useful for modifying software recommendations, virtual assistants, image indexing, and visually challenged persons.

Face mask detection is for when you go to the hospital, people with pneumonia and other infectious diseases are required to have a face mask if the employees are sicker will remove the face mask and control them. the other is to have face mask detection a lot of people in a crowded place remove their masks, but to control what they do.

1.7. Delimitation Scope

1.7.1. Scope of the study

The Scope of the study focuses on developing face mask detection in a CNN algorithm. It will find the way to face at a different position and detect the face, if taken mask or not taken face mask the datasets that were selected for this study were those that have ground truth labels, this study covers only supervised training mechanisms for detection trained or test with the standard datasets.

1.7.2. Limitation of the study

The study will focus on detecting face mask protocol. It will be found that high-performance GPU process computers and it is more expensive, our university has a limited number of GPU computers that can only be used for a certain number of days, and there are some limitations to using Google Colab.

1.8 Organization of the thesis

It is divided into six chapters, each of which focuses on a different component of the research and provides a reason.

Chapter One: - describes the background of the study, statement of the problem and the research question, methodology to achieve the objective of the study, the scope and limitations, and the expected outcomes of this study.

Chapter Two: - This chapter explains the background literature and related works for face detection and face mask detection. Theoretical framework of machine learning, Deep Learning, detection, and CNN. Current papers that focus on

Chapter Three: - In this chapter, the research methodology is explained in detail, along with the many approaches and strategies used to design the solution and choose the best one. From the dataset's collection through the model's performance testing using pertinent performance.

Chapter Four: - will cover points about the proposed model in detail Face detection, face mask detection data understanding, the model selected, and evaluation methods used.

Chapter Five: - Describe the suggested model implementation, including how to build up a functioning configuration, implementation of the data set description model, and evaluation. and how to visualize the model summary using a graph.

Chapter Six: - Present the estimated key results of the proposed model, compare the new result with the most recent baseline models, and explain the importance of the result.

Chapter Seven: - The summary of the entire effort, a synopsis of the findings, and recommendations for potential future work are explained in this section.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORK

This chapter reviews recent surveys of academic articles, journals, conference papers, and other sources that are linked to face detection, face mask detection, and CNN and what will be the gap, result, and method of those document

2.1 Introduction

A large amount of data is produced by both machines and humans. The majority of data collected today is unstructured and comes from a variety of sources, including sensors used to gather climate data, digital photos, and videos published on various websites such as Facebook, and tweets from Twitter. In recent years, the significant advancement in GPU computing power has led to a flourishing era for deep learning technology. The enhanced capabilities of GPUs have played a pivotal role in driving the rapid growth and effectiveness of deep learning applications across various domains (Jiang et al., 2021). The implemented system comprises three fundamental modules: first, the training of a Convolutional Neural Network (CNN) for classification; second, the utilization of a facial annotation algorithm for face detection; and third, the integration of the input video stream to execute the face mask detection task. This multi-module approach ensures a comprehensive system capable of classifying images, detecting faces, and addressing the specific task of identifying the presence or absence of face masks in real-time video streams(Computing & Technology, 2021).

Artificial Intelligence (AI): - is a fascinating field that explores the creation of intelligent agents, which are systems that can reason, learn, and act autonomously. In simpler terms, it's about building machines that can think and solve problems like we humans do. A computational concept designed to enable machines to think and tackle intricate problems in a manner akin to human intelligence. Similar to human cognition, AI systems engage in tasks, encounter challenges, learn from errors, and strive for self-improvement. This concept has its roots as far back as 1950 when Alan Turing introduced the Turing test. Notably, the first chatbot computer program, ELIZA, emerged in the 1960s. Advancements continued, leading to significant milestones such as the creation of IBM Deep Blue in 1977. This chess computer triumphed over a world chess champion in two out of six games, with one victory for the champion and three draws (Haton, 2006).

Image Processing (IP): - Image processing is a method used to improve the raw images that are captured by cameras or sensors on satellites, spacecraft, and other aircraft or images captured during daily life for a variety of uses. A technique for performing operations on images uses the usage of a digital computer to process digital images using an algorithm to produce a better image or extract important information. In the last four to five decades, the field of image processing has witnessed the development of diverse techniques. Many of these techniques were initially designed to enhance images captured by space probes, unmanned spacecraft, and military surveillance missions. The proliferation of potent personal computers, high-capacity memory devices, and sophisticated graphics software has contributed to the increasing popularity of image-processing systems. The applications of image processing are extensive and diverse. It finds utility in Remote Sensing, Medical Imaging, Non-destructive Evaluation, Forensic Studies, Textiles, Material Science, Military operations, the Film industry, Document processing, Graphic arts, and the Printing Industry. As technology continues to advance, the scope and impact of image processing in various domains are expected to expand even further (Chithra & Santhanam, 2018).

Machine Learning (ML): - A formal definition of machine learning states that a program learns from experience E concerning task class T and performance measure P if its performance on T, as measured by P, demonstrably improves with repeated exposure to E. (Jain, 2015). ML is a kind of artificial intelligence (AI) that enables software applications to learn from their experiences and increase their accuracy in predicting results without being explicitly programmed to do so. To forecast new output values, machine-learning algorithms use historical data as input. ML aims to build algorithms that can learn from data and improve their performance over time. While often used interchangeably with data mining, ML stands apart by focusing on the development of adaptive models for prediction, decision-making, and automation. Data mining, on the other hand, delves into extracting meaningful patterns and insights from vast datasets without necessarily building predictive models (Ławrynowicz & Tresp, 2014). Some applications of ML Spam filtering are Credit card fraud detection, Recommendation systems, Scene classification, Search engines, Detecting faces in images, Digit recognition on checks zip codes, MRI image analysis, and Handwriting recognition.

Computer Vision (CV): - In the 1960s - started as a student summer project at MIT, is a discipline focused on devising methodologies to enable computers to perceive and comprehend the contents of digital images and videos. Essentially, it constitutes a scientific domain that empowers computers to capture, interpret, comprehend, and process visually perceivable objects. Pattern learning and object recognition stand out as inherent tasks within the realm of computer vision. (Sethi et al., 2020). Computer vision uses operations on the data received from video cameras to create insightful and practical descriptions of visual situations and sequences as well as the objects that fill them. Traditional computer vision encompasses various tasks, including image processing, image classification, object detection, and image recognition. Among these, object detection stands out as a crucial capability that involves identifying instances of visual objects belonging to specific classes within images.(Jiang et al., 2021). Computer vision faces several challenges, with one of the core issues being Object Recognition. Currently, recognizing objects that are not only hard but also at an appropriate scale remains a significant open problem. Various obstacles contribute to the complexity of object recognition, including deformation, appearance variation, and scale variation(Hays, 2022).

Deep Learning (DL): - Deep learning, a subset of machine learning and artificial intelligence (AI), mirrors the way humans acquire knowledge. This technique involves computational models with multiple processing layers that learn representations of data at various levels of abstraction. (Lecun et al., 2015). Data science, a field that incorporates statistics and predictive modeling, has embraced deep learning as a fundamental element of its toolkit. In the realm of modern data science, advanced deep learning algorithms are seamlessly integrated with geometric techniques to construct robust models with diverse applications. These models go beyond traditional statistical approaches, covering crucial aspects such as detection, tracking, and validation. (Suresh, Palangappa, et al., 2021). This approach has significantly advanced various fields, including speech recognition, visual object recognition, object detection, and domains such as drug discovery and genomics. The power of deep learning lies in its capacity to uncover complex structures within extensive datasets. The backpropagation algorithm is employed to guide the machine in adjusting its internal parameters, ensuring that each layer accurately computes representations based on the information from the preceding layer. This iterative process enables the model to progressively learn and refine its understanding, leading to state-of-the-art performance in a wide range of applications. (Mathew et al., 2021).

Transfer Learning (TL): - Transfer learning stands out as one of the most frequently employed approaches for computer vision tasks, including classification and segmentation (Venkateswarlu, 2021). Training time is decreased by transfer learning when the application areas are closely connected. There are two popular methods for carrying out transfer learning, which are as follows.

1) Using a pre-trained model: Utilizing a pre-trained model involves employing a model that has been previously trained on a sizable benchmark dataset. Numerous pre-trained models, such as ResNet, MobileNet, and GoogleNet, have undergone training on extensive ImageNet datasets. These models are designed to accept color images as input and excel in performing 1000 class classifications. When the objective is to classify any class defined in the ImageNet dataset, these pre-trained models often yield optimal accuracy due to the comprehensive and diverse nature of their training data.(Venkateswarlu, 2021).

2) Defining a custom output layer: Defining a custom output layer involves treating the pre-trained model as a feature extractor by excluding its original output layer. The features extracted by the model can then be utilized for a specific classification task by appending a custom output layer. Subsequently, the custom model is trained to enhance its classification performance based on the specific requirements of the new task(Venkateswarlu, 2021).

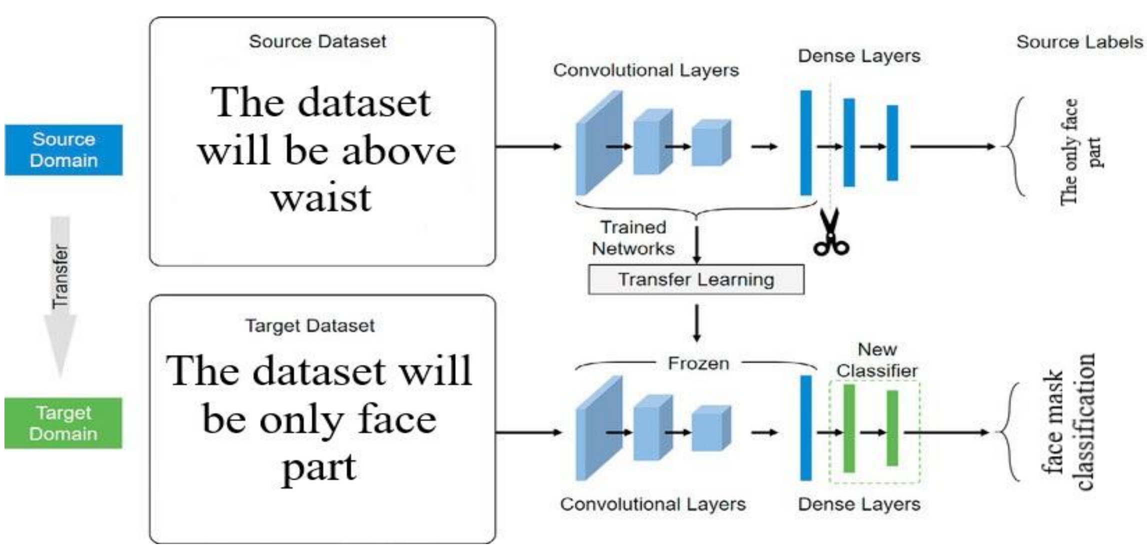


Figure 2.1 The architecture of the transfer learning model

Internet of Things (IoT): - The story of our interconnected world can be traced back to 1912 when Chicago witnessed the deployment of the very first telemetry system. This groundbreaking innovation laid the foundation for today's Internet of Things (IoT), a vast network that seamlessly connects everyday objects like smartphones, TVs, laptops, computers, sensors, and actuators. Through this intricate web, devices converse with each other and humans, fostering new forms of communication and interaction(Mask & Using, 2021). IoT denotes a network infrastructure designed to connect a diverse range of objects to the Internet. This connectivity is established through predefined protocols, utilizing information-sensing devices to exchange information and facilitate communication. The primary objective of the IoT is to enable intelligent functionalities such as recognition, positioning, tracking, monitoring, and management of connected devices. Through this interconnected system, the Internet of Things aims to enhance efficiency and provide smarter, more responsive solutions in various domains.(Patel et al., 2016). IoT presents a transformative opportunity for various sectors, particularly in agriculture. In the agricultural domain, IoT technologies find applications in diverse areas, including precision agriculture or smart farming, irrigation monitoring, environmental monitoring, and cattle animal tracking. These technologies play a crucial role in optimizing various processes within agriculture. Different IoT technologies, such as Radio Frequency Identification (RFID), Wireless Sensor Networks (WSN), sensors, and Global Positioning System (GPS), have been extensively integrated into different agricultural practices.(Tegegne et al., 2019).

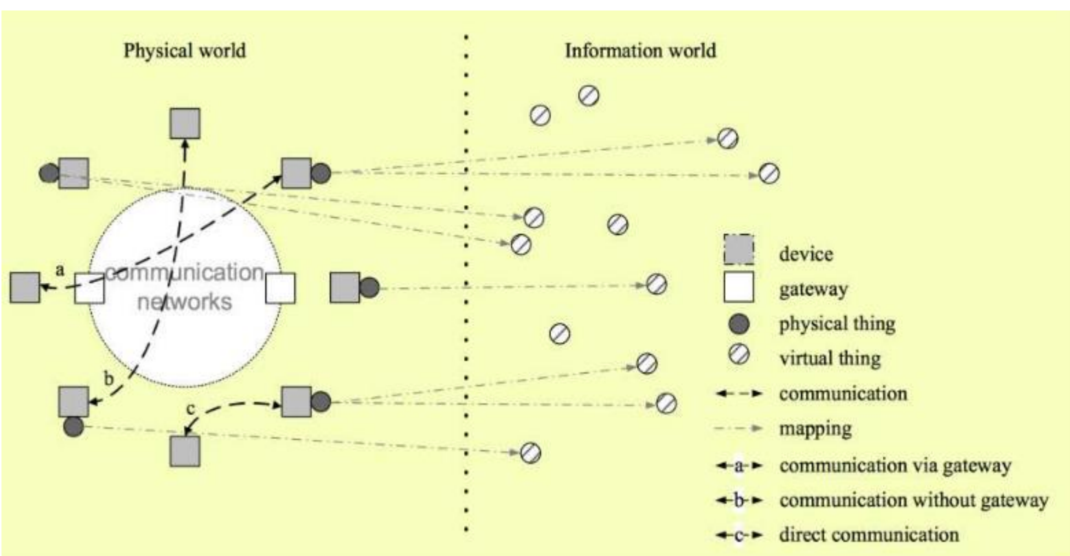


Figure 2.2 Sample of IoT (Bell, 2021)

2.2 Literature Review Model

Convolutional Neural Networks CNN: - Artificial Intelligence algorithms known as Convolutional Neural Networks (CNNs) are grounded in multi-layer neural networks. These networks are adept at learning pertinent features from images, enabling them to undertake various tasks such as object classification, detection, and segmentation(Todt & Krinski, 2019)**.** Deep convolution neural networks, or CNNs, are now widely used for computer vision applications including image classification(Venkateswarlu, 2021). Keras is utilized to implement each layer in the CNN model(Das et al., 2020). A CNN is made up of several building pieces, including the convolutional, pooling, and fully connected layers. Using a backpropagation algorithm, it can automatically and adaptively learn the spatial hierarchies of data(Jiang et al., 2021). Convolutional Neural Networks (CNNs) represent a category of deep neural networks inspired by biological processes. The concept of convolution was initially introduced by Kunihiko Fukushima in 1980 under the name Neocognitron. The contemporary understanding and development of Convolutional Neural Networks can be attributed to the work of Yann LeCun, whose groundbreaking work was published in 1998. LeCun introduced a specific CNN called LeNet, designed for handwritten recognition tasks.(Todt & Krinski, 2019).

Recurrent neural network RNN: - is an artificial neural network designed for handling sequential or time series data. Unlike traditional feedforward neural networks, RNNs have the unique capability of incorporating previous outputs as inputs while preserving hidden states. This characteristic makes them particularly effective for tasks involving sequential information. Neural networks emulate the functions of the human brain, enabling computer systems to recognize patterns and effectively address a variety of complex problems.

Artificial neural network ANN: - ANN can be conceptualized as computational models inspired by the structure and functioning of biological neural networks(Yegnanarayana, 1994). ANNs are a testament to the human brain's incredible capabilities and offer a glimpse into the future of intelligent machines. By understanding their inner workings and potential, we can harness their power to solve complex problems and shape a better tomorrow.

2.3 Overview of detection

Develop a novel object detection method that seamlessly integrates the strengths of both one-stage and two-stage detectors, while incorporating real-time capabilities and leveraging transfer learning at the backend(Sethi et al., 2020). The initial research on face detection, conducted in 2001, focused on employing handcrafted features and traditional machine-learning algorithms to train classifiers for effective detection and recognition. This approach involved designing features manually and utilizing established machine-learning techniques and low detection accuracy(Sethi et al., 2020). Detection is a fundamental task in computer vision and finds widespread applications in various domains. This includes tasks such as face detection, skeleton detection, pedestrian detection, and sign detection. (Jiang et al., 2021).

2.3.1. Face Detection

In recent years, there has been a notable surge in the development of face-detection methods leveraging deep Convolutional Neural Networks(Sethi et al., 2020). face detection serves as a crucial preliminary step in any face recognition or verification process. The primary goal of a face detection algorithm is to identify and locate faces within a given input image, typically providing the output in the form of bounding boxes that delineate the detected faces. Face detection poses inherent challenges due to the variability in facial characteristics such as size, shape, and color. Faces are not static entities; they can change expressions, poses, and lighting conditions, making the task of accurately detecting them challenging. Additionally, obstacles such as occlusions, where a part of the face is obscured, or varied camera angles further complicate the detection process(Das et al., 2020). Face Detection is an application software designed to analyze and process human faces. This software includes features that allow users to input images, enabling the detection of various facial components such as eyes, nose, mouth, and the entire face. The primary function of face detection is to identify one or more human faces within images or videos. This technology holds significance in numerous applications, particularly in biometric systems, security implementations, and surveillance systems. Additionally, it plays a vital role in image and video indexing systems. Face detection can be considered a specific instance of object-class detection. In the broader context of object-class detection, the objective is to locate and determine the sizes of all objects within an image that belong to a specified class, with faces being a notable category within this classification(Loy, 2021).

2.3.2 Face Mask Detection

The facemask detection model has evolved into a crucial and indispensable tool during the COVID-19 pandemic. Manual monitoring of individuals wearing face masks in public and crowded areas is both challenging and labor-intensive(Suresh, Mb, et al., 2021). The task's key issue is successfully detecting the face from the image and then determining wearing or not it has a mask on it. the proposed approach should be able to detect a face and a mask in motion to execute surveillance activities. Aims to detect facemasks on human faces on live streaming video and also with human face images. The face detector model is built using deep learning and a single-shot detector (SSD)(Suresh, Palangappa, et al., 2021). Face masks are required by law to be worn in public. The primary goal of the surveillance is to identify anyone who is not wearing a face mask. Using computer vision, a real-time automated model connected with public surveillance cameras can determine whether people are wearing face masks and keeping their distance from others in public places, and it can then notify the appropriate authorities.

2.4 Data pre-processing tools

If we are using a collected dataset of images captured above the waist, the accuracy of the face mask detection model will decrease if do not pre-process the images to keep only the facial parts. This is because the model will be distracted by the other objects in the image, such as the background, clothing, and hands. To increase the accuracy of the face mask detection model, we should pre-process the images to keep only the facial parts. This can be done using a variety of tools, such as:

Facial landmark detection: Facial landmark detection is a computer vision task wherein a model is tasked with predicting key points that signify specific regions or landmarks on a human face, such as the eyes, nose, and lips. Serving as a foundational task, facial landmark detection lays the groundwork for various other computer vision applications, including head pose estimation, gaze direction identification, facial gesture detection, and face swapping. Typically, standard facial datasets offer annotations consisting of 68 x and y coordinates that pinpoint the critical points on an individual's face. Dlib, a widely employed open-source library, is adept at recognizing these landmarks in images, making it a commonly used tool for facial landmark detection tasks.[1]

LabelIng Multi-modal data annotation: The well-known image annotation tool, initially developed by Tzutalin with contributions from numerous individuals, is now no longer in active development and has seamlessly transitioned into the Label Studio community. Explore Label Studio, the highly adaptable open-source data labeling tool designed for images, text, hypertext, audio, video, and time-series data. LabelImg, a graphical image annotation tool, written in Python and utilizing Qt for its graphical interface, has played a significant role in this community.[2]

LabelBox: the Label Box platform is the process of labeling data for machine learning models. LabelBox is a data-centric AI platform designed to streamline and empower the entire AI development process, from data labeling to model deployment and beyond. Here's a breakdown of what LabelBox offers. LabelBox is your one-stop shop for building and deploying powerful AI solutions. It's not just a labeling tool, but a comprehensive platform that empowers your entire team to turn raw data into intelligent applications.[3]

VGG Image Annotator: VGG Image Annotator is a straightforward and standalone manual annotation tool designed for images, audio, and video. VIA operates directly within a web browser, eliminating the need for any installation or setup. The entire VIA software is encapsulated within a single self-contained HTML page, totaling less than 400 Kilobytes in size. This enables it to function as an offline application in the majority of modern web browsers. VIA stands as an open-source project crafted entirely with HTML, Javascript, and CSS, devoid of any dependencies on external libraries. Developed within the Visual Geometry Group (VGG), VIA is released under the BSD-2 clause license, offering versatility for deployment in both academic projects and commercial applications.

---

[1] https://datagen.tech/guides/face-recognition/facial-landmarks/
[2] https://github.com/HumanSignal/labelImg
[3] https://docs.databricks.com/en/partners/ml/labelbox.html

2.5 Model Selection

MobileNet: - The pre-trained model employed in our implementation is MobileNetV2. This model leverages depth-wise separable convolutions, a width multiplier, linear bottlenecks, and shortcut connections, having undergone training on the ImageNet dataset. The use of MobileNetV2 contributes to faster processing and enhances the accuracy compared to a conventional CNN model. MobileNets are designed as low-latency, low-power models specifically parameterized to align with the resource constraints of diverse use cases. (Computing & Technology, 2021).

ResNet: - Numerous successful deep Convolutional Neural Network (CNN) models are available, including five proposed variants of ResNet: ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. These models demonstrate that both performance and computational time are directly proportional to the number of layers. Specifically, the architecture of ResNet-50 closely resembles that of ResNet-34(Venkateswarlu, 2021). ResNet-50, like other convolutional neural networks, necessitates input images with dimensions of 224-by-224-by-3. However, in our dataset, the images have varying sizes. To address this, we employed an augmented image dataset, which automatically resized the training images to meet the required dimensions for effective utilization in the ResNet-50 model(S. Hussain et al., 2021).

Inception: - The core building block of the Inception model is the inception module. This module aims to improve upon traditional CNNs by combining filters of different sizes (1x1, 3x3, 5x5) in parallel within a single layer. This allows the model to capture information from various spatial scales without significantly increasing the number of parameters, leading to increased efficiency and performance. (Venkateswarlu, 2021).

Viola-Jones: - The Viola-Jones object detection framework is a machine-learning method for object detection proposed in 2001 by Paul Viola and Michael Jones. It was primarily motivated by the problem of face detection, although it can be adapted to the detection of other object classes. (Yegnanarayana, 1994).

YOLO: - YOLO (You Only Look Once) gained popularity for pioneering the single-stage approach, showcasing real-time predictions with impressive detection speed. However, it faced challenges, particularly in terms of lower localization accuracy when compared to two-stage detectors(Sethi et al., 2020). The YOLO network partitions an image into a grid of size GxG, where each grid cell generates N predictions for bounding boxes. Notably, each bounding box is constrained to predict only one class during the prediction process. (Sethi et al., 2020).

Yolov1 (Jun 2015): First introduced to the computer vision community through a paper released in 2015 by Joseph Redmon and his colleagues. This paper presented the YOLO algorithm, which revolutionized object detection by enabling unified and real-time detection capabilities. (M. Hussain, 2023).

Yolov2 (Dec 2016): Joseph Redmon introduced YOLO-v2/9000, in 2016. The objective behind this version was to address and alleviate the inefficiencies observed in the original YOLO while retaining its remarkable speed factor. This iteration aimed to improve both accuracy and efficiency in object detection. (M. Hussain, 2023).

Yolov3 (Apr 2018): recognized as an incremental improvement in architectures, departed from the notion followed by some architectures like VGG, which emphasized that deeper networks with more internal layers result in higher accuracy. YOLO-v2, the predecessor, also incorporated a higher number of convolutional layers compared to its original version, contributing to its improved performance(M. Hussain, 2023).

Yolov4 (Apr 2020): The first variant of the YOLO family after the original author discontinued further development. Introduced to the computer vision community in April 2020 by Alexey Bochkovsky and collaborators, YOLO-v4 represented the distillation of a comprehensive array of object detection techniques. It underwent rigorous testing and enhancements to provide a real-time, lightweight object detector with improved performance(M. Hussain, 2023).

Yolov5 (May 2020): As of August 1st, 2021, there is no official paper available for YOLOv4 in the traditional sense; however, development updates and information are often communicated through GitHub. One noteworthy development reported on GitHub is the implementation of the first version of YOLO in PyTorch rather than Darknet(M. Hussain, 2023).

Yolov6 (June 2022): the Meituan Technical Team in China developed The YOLOv6. The authors strategically designed this industry-oriented object detector with a focus on introducing three significant improvements over the previous YOLOv5. YOLOv6 demonstrates exceptional results in terms of both accuracy and speed on the COCO dataset, surpassing the performance of its predecessors.[4]

Yolov7: A month after the release of YOLOv6, YOLOv7 was introduced. While there were other variants released in between, such as YOLO-X and YOLO-R, these variants primarily emphasized GPU speed enhancements in terms of inferencing. In contrast, YOLOv7 proposed several architectural reforms to improve accuracy while maintaining high detection speeds.

Yolov8 (the latest release January 2023): YOLO-v8, confirmed by Ultralytics, who also released YOLO-v5. While a formal paper release is forthcoming and additional features are expected to be added to the YOLO-v8 repository, initial comparisons suggest its superiority as the new state-of-the-art YOLO model over its predecessors. The continuous evolution of YOLO architectures underscores the commitment to advancing the state-of-the-art in real-time object detection.

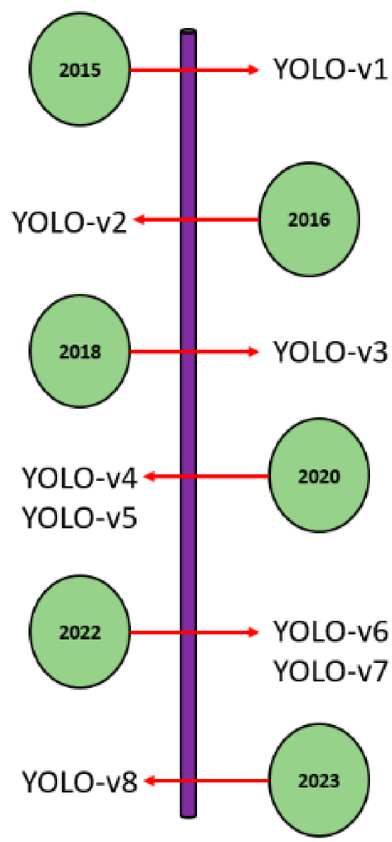


Figure 2.3 YOLO evolution timeline

[4] https://www.datacamp.com/blog/yolo-object-detection-explained

EfficientNet: - is a convolutional neural network design and scaling method that introduces a compound coefficient to uniformly scale all dimensions, including depth, width, and resolution. This scaling approach allows EfficientNet models to surpass the performance of previous convolutional neural networks (CNNs) in terms of both accuracy and efficiency. Additionally, EfficientNet achieves these improvements while simultaneously reducing the overall parameter size, making it a highly efficient and effective architecture for various computer vision tasks.

2.6 Related Work

We developed a rapid and precise mask detector featuring a channel attention mechanism, which significantly enhanced the feature extraction capabilities of the underlying network. Additionally, to bolster the accuracy and robustness of the model, we incorporated Generalized Intersection over Union (GIoU) and focal loss. Moreover, we applied tailored data augmentation techniques corresponding to these improvements(Jiang et al., 2021). We demonstrate the performance of SE-YOLOv3 on the Properly-Wearing-Masked Face Detection Dataset. The implementation and experiments were conducted using the TensorFlow deep learning framework. The network underwent training and testing on a local machine equipped with an RTX 2070 GPU boasting 8 GB of memory. To optimize the model during the training phase, the Adam optimizer was employed. Furthermore, a dynamic adjustment of the learning rate was implemented through a warm-up and attenuation strategy(Jiang et al., 2021). This highlights the importance of monitoring whether individuals are wearing masks in public spaces. Automating this task can significantly contribute to controlling the spread of the virus. Designed to accept live video input from a camera, the model actively monitors faces, discerning whether individuals are wearing masks or not (Computing & Technology, 2021).

Table 2.1 Summary of related work

Authors and paper	Title	Methods	Gap
Das, ArjyaAnsari, Mohammad WasifBasak, Rohini(Das et al., 2021)	Covid-19 Face Mask Detection Using TensorFlow, Keras, and OpenCV	The proposed method consists of a cascade classifier and a pre-trained CNN, which contains 2D convolution layers, connected to layers of dense neurons. The Convolution layer learns from 200 filters. Kernel size is set to 3 x 3 which specifies the height and width of the 2D convolution window.	● It can be extended to detect if a person is wearing the mask properly or not. ● The model can be further improved to detect if the mask is virus-prone or not ● The type of the mask is surgical or not.
Saisupriya, N. Priyanka Rashmi, S.Parthasarathy, G.(Saisupriya & Rashmi, S.Parthasarathy, 2021)	Face Mask Detection Using CNN	We are using the transfer learning concept where utilizing a pre-trained model has MobileNetV2, The CNN for classification, face detection using the Viola-Jones Algorithm, and obtaining the input video stream to carry out the facemask detection.	● It can be extended to detect if a person is wearing the mask properly or not.
Jiang, Xinbei,Gao, Tianhan,Zhu, Zichen Zhao, Yukang (Jiang et al., 2021)	Real-Time Face Mask Detection Method Based on YOLOv3	It improved the accuracy of YOLOv3, Firstly, it introduced an attention mechanism to the backbone network, which can help the network allocate more resources to important features.	● It can be extended to detect if a person is wearing the mask properly or not. ● It's inaccurate Face Detection Dataset for Correctly-Wearing Masked Faces
Pandey, Vivek Kumar Gupta, Vikas Kumar Kumar, Sushant(Pandey et al., 2022)	Detection of Face Mask Using Convolutional Neural Network	The method is using MobileNetV2 called Transfer Learning. Transfer learning is using some pre-trained models to train your present model and get the prediction, which saves time and makes training the different models easy.	● For multiple faces also it gives quite a good accuracy. ● It can be enhanced to determine whether someone is correctly donning the mask.

Arora, Rajat(Arora, 2020)	COVID-19 Face Mask Detection	It decided to build a very simple and basic Convolutional Neural Network (CNN) model using TensorFlow with Keras library and OpenCV. The developed architecture is trained for 10 epochs	● It does not prioritize facemask detection over several faces. people always try to break it. ● It can be extended to detect if a person is wearing the mask properly or not.
IsunuriB Venkateswarlu, Jagadeesh Kakarla, Shree Prakash(Venkateswarlu, 2021)	Face mask detection using MobileNet and Global Pooling Block	We have proposed a pre-trained MobileNet with a global pooling block for facemask detection. We also have selected three popular pre-trained deep networks namely RestNet50, MobileNet, and GoogleNet for the proposed model.	● It does not focus on facemask detection over multi-face images. ● The datasets have masks and don't have masks; a protocol is not accurate.
Suresh, KMb, Palangappa Bhuvan, S(Suresh, Mb, et al., 2021)	Face Mask Detection by using Optimistic Convolutional Neural Network	The proposed system uses Computer Vision and MobileNet to help the public ensure that they are wearing face masks and to keep away from the spread of the COVID-19 virus.	●Identify a person if he is doing any crime by wearing a face mask. ●Identify what type of mask is the person wearing. ● Coughing and Sneezing Detection. ● Temperature Screening

Summary of related work

The above tables illustrate previous works on face detection and face mask detection. this study comes up with a solution for already identified problems, to overcome face mask detection, the papers selected for this study are from 2021-2022.

The review research will focus on social distance and facemask detection, but there will be a gap in the above-reviewed research it will be the accuracy and not covering the protocol that wearing a face mask, It can be extended to detect if a person is wearing the mask or not.

The other revised paper will focus on whether the type of mask is surgical or not, and its gap will be inaccurate Face Detection Dataset for Correctly-Wearing Masked Faces.

Multiple faces also give quite a good accuracy, it gap will be extended to detect if a person is wearing the mask properly or not, It only focuses on face mask wearing or not but it does not focus on the protocol.

The other paper will not focus on facemask detection over multi-face, and the datasets have masks and don't have masks.

CHAPTER THREE

3. RESEARCH METHODOLOGY

In this chapter, the overall methodology and the procedure of developing face mask detection to build these detection machine learning models different methods, techniques, and procedures have to be used in each phase. The entire research process has been explained, rationalized, and elaborated. Starting from the selection of publicly available and collecting datasets to the necessary preprocessing steps taken to ease the training, the preparation of data for training the models has been included in the data collection and preparation section. Model selection and change detection scheme are briefly provided. Finally, the software and hardware used in the training process, and the evaluation metrics commonly used for face mask detection.

3.1. Research Design

The research design refers to the overall strategy or plan used to conduct a research study. Research design can be categorized into quantitative, qualitative, or hybrid research designs. The quantitative research design involves the collection and analysis of numerical data through structured methods such as surveys, experiments, and statistical analysis, it is often used to study phenomena that can be measured and quantified and aims to provide precise and objective results. Qualitative research design, on the other hand, involves the collection and analysis of non-numerical data such as interviews, observations, and documents. Hybrid research design combines elements of both quantitative and qualitative research design to provide a more comprehensive understanding of a phenomenon, it can involve collecting both numerical and non-numerical data and analyzing them using both statistical and interpretive methods.

This study used an experimental research design approach because the quantitative research design approach, which entails identifying research objectives and creating machine learning models to demonstrate the concept, finds the relationship between two variables, i.e., independent variable and dependent variables, within the data samples (Kamiri & Mariga, 2021).

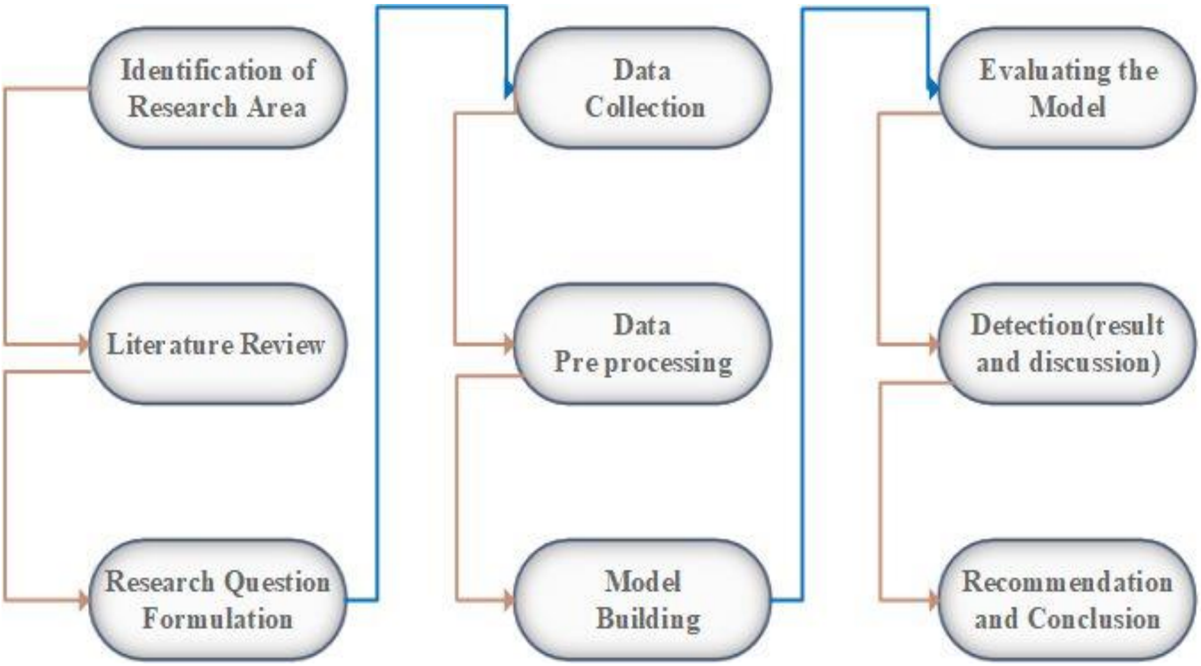


Figure 3.1 Flowchart Diagram of the Research

3.2. Data Collection Methods

We gather the data using a digital camera to capture images. Each individual is photo graphed to obtain the necessary data. Face masks wearing voluntary person that photo capture. The data set includes both males and females, as well as images with various age variations. In a one person will be captured four image

The first image will be in the correct form wearing the face mask that miens it covers the chin, mouth, and nose

Figure 3.2 Data set sample Face mask wearing

The second image will be is not properly wearing a face mask; it just covers the chin and mouth and does not cover the nose,

Figure 3.3 Data set sample Incorrect form wearing a face mask

Third will be it is not properly donning a face mask because it barely covers its chin and leaves its mouth and nostrils uncovered,

Figure 3.4 Data set sample Incorrect form wearing a face mask

The last will be it does not have a face mask.

Figure 3.5 Data set sample Not wearing a face mask

3.3. Data Pre-processing Technique

The most critical aspect of implementing machine learning or deep learning is the dataset. for the sake of our research, we used four types of datasets First In the correct form wearing the face mask, second it is not wearing a facemask in the correct form it covers the chin and mouth but does not cover the nose, it is not wearing a facemask in the correct form it covers the only chin it is not covering the nose and mouth and some will be it is not covering chin, forth it is not wearing a face mask.

In dataset processing time there are three class

1. **Face mask wearing** will be classified if it is the correct form of wearing the face mask.
2. **Incorrect form Wearing a face mask** in this class is not wearing a facemask in the correct form it covers the chin and mouth but it is not covering the nose, and it is not wearing a facemask in the correct form it covers the only chin it is not covering the nose and mouth and some will be it is not covering chin.
3. **Not wearing a face mask** third will be classified it is not wearing a face mask.

We select two models the first for face detection or dataset preprocessing and the second will be for facemask detection. LabelIng Multi-modal data annotation is one of the selected models for face detection or dataset preprocessing the points belong to the nose, the eyes, the mouth, and the edge of a face. YOLO possesses a distinct advantage in speed, coupled with superior Intersection over Union in bounding boxes and enhanced prediction accuracy compared to real-time object detectors. Its innovative approach involves employing an end-to-end neural network that simultaneously predicts class probabilities and bounding boxes, streamlining the detection process. Operating at up to 45 frames per second (FPS), YOLO significantly outpaces its competitors, making it a remarkably fast algorithm. Moreover, it excels in categorizing whether a facemask is worn correctly, incorrectly, or not at all, showcasing its versatility and effectiveness in various applications.

3.5. Evaluation Method

The next step after system implementation is system evaluation to determine whether the system has achieved the desired goal or not. We used two evaluation measures following execution Classification Accuracy, Confusion Matrix, and F1 Score.

3.5.1 Classification Accuracy

The ratio of the number of correct predictions to the total number of input samples is known as the accuracy.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ prediction\ made} \qquad 3.1$$

3.5.2. Confusion Matrix

The matrix you're referring to is likely the confusion matrix, which provides a comprehensive overview of a model's performance. It consists of four important terms:

True Positives: The number of instances correctly predicted as positive by the model.

True Negatives: The number of instances correctly predicted as negative by the model.

False Positives: The number of instances incorrectly predicted as positive by the model.

False Negatives: The number of instances incorrectly predicted as negative by the model.

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Sample} \qquad 3.2$$

3.5.3. F1 Score

The F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \qquad 3.3$$

Precision: It is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \qquad 3.4$$

Recall: It is the number of correct positive results divided by the number of ***all*** relevant samples (all samples that should have been identified as positive).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad 3.5$$

3.6. Design and Development Tools

That is, they are the things that are needed to make or distribute this work or research, and they are divided into two parts hardware tools and software tools.

3.6.1. Hardware tools

To properly implement this investigation, we shall require hardware tools. In the table below, you'll find a list of the hardware tools required for this investigation.

Table 3.1 Hardware tool used for this study implementation

No.	Device name	Used in the study
1.	RAM	Because most systems have limited RAM, to speed up the training process.
2.	Hard Disk	Keeping the large number of datasets that were used to train, test the model, and store its progress.
3.	GPU	Because training big datasets on the CPU takes time, we utilized the GPU to speed up the training and computing.

3.6.2. Software tools

Different software and coding tools will be utilized to implement the research through coding, and writing, and these are given below with their descriptions.

Microsoft Office packages: - It is used to prepare the study document and create slides for presentation, and Visio is used to create the proposed system's diagrams and flowcharts.

Mendeley Desktop: - It is well known for its reference manager, which is used to organize and exchange research papers, as well as create bibliographies for scholarly publications and references to related works.

Anaconda: - is a Python and R programming language distribution for scientific computing. It is a program that installs Python and all of its modules, as well as the R programming language, depending on the Python edition. it includes a navigator program for viewing various settings such as vs-code, Jupiter notebook, spider, and installed modules.

TensorFlow: - A software application for implementing machine learning methods, notably neural networks, is publicly available. it's a free, end-to-end open-source machine learning platform. it can also be used on desktop PCs or in the cloud.

Python: - Python programming is utilized to implement and show the primary programming language used in this study to write the programs and packages.

Jupyter Notebook: - Among AI and deep learning researchers, build open-source tools, open standards, and services for interactive computing using Python, as well as develop Python code.

Keras: - Keras is a Python interface to a machine learning framework that is freely available Keras serves as a user interface for TensorFlow. It was a user interface till version 2.3.

CHAPTER FOUR

4. PROPOSED SOLUTION FOR FACE MASK PROTOCOL DETECTION

4.1 Chapter Overview

This chapter describes the model, used in the algorithms utilized in the face mask protocol detection implementation. Furthermore, the outcomes of every algorithm within the model were provided beneath every component. Using local data sets, the model will be trained. Together with diagrams and descriptions, Data sets will be preprocessing techniques, and face mask image preprocessing techniques proposed are also presented along with diagrams and descriptions.

4.2 Proposed Model Architecture

This research aimed to build a prediction model for face mask protocol detection based on the Internet of Things. It predicts which face mask samples are with face mask-wearing, not wearing a face mask, incorrect form wearing face mask based on face mask utilization collected dataset. This study's proposed architecture specifies that initially, the collected data was raw. Since it is raw data it is prone to inconsistency, noise, variety, duplication and the captured image will be above the flank and some unwonted background, scaling, classing so on. Because of this, we need to undergo the data preprocessing stage such as data cleaning, handling missing values, excluding the only face parts, and handling categorical values to transform raw data into meaningful information and make it suitable for the deep learning algorithm. Following the preprocessing of the dataset, Facial landmark detection (FLD), LabelIng Multi-modal data annotation, LabelBox, and VGG Image Annotator (VGGIA) are some dataset preprocessing models bet we selected LabelIng Multi-modal data annotation used to build the preprocessing techniques.

After balancing the dataset and preprocessing the yolo is used to build the predictive model. Their predictive performance is assessed by using various performance metrics to determine the best-performing model. Also, other machine learning algorithms such as resnet, mobile net, and inception algorithm were used to train the dataset for comparative analysis. And based on the experimental result yolo shows the best performance for both facemask detection.

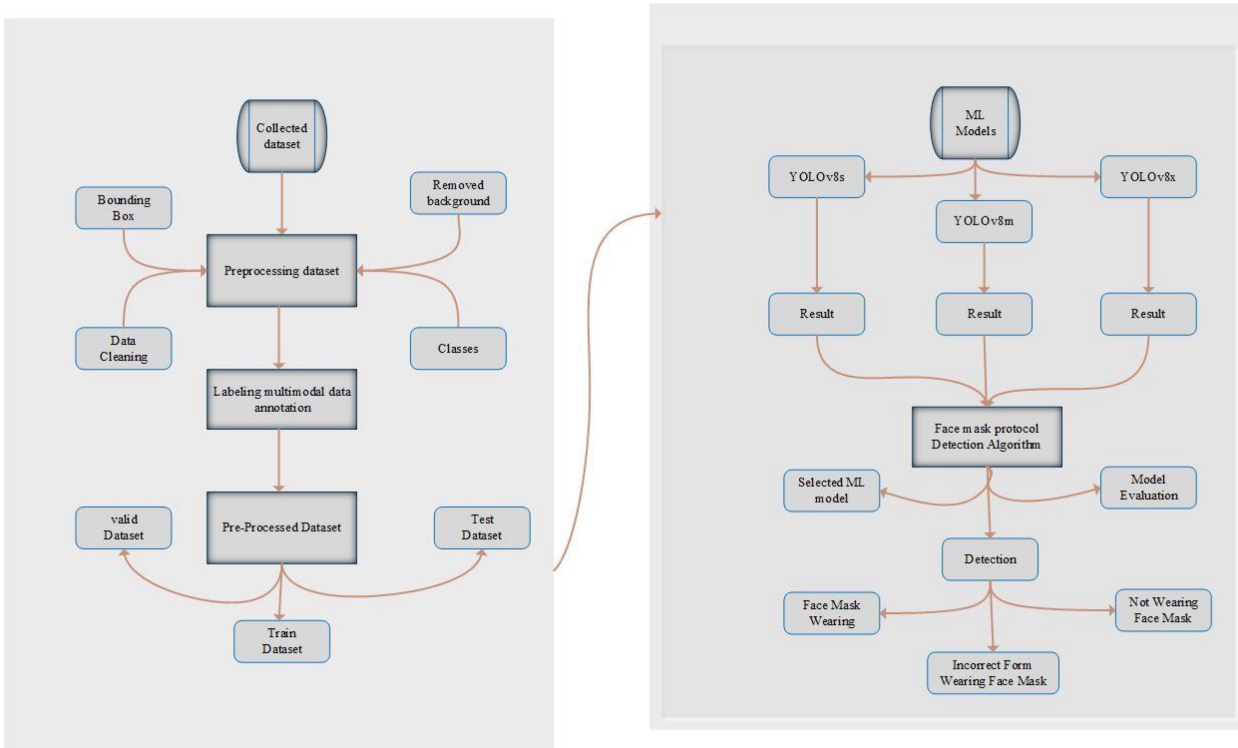


Figure 4.1 Face mask detection model architecture

4.3 Understanding of The Data

Data Understanding are two key steps.

4.3.1 Data Assessment

This should be undertaken before the kick-off of a project as it is an important step to validate its feasibility. It should answer the following questions:

What data is available? How much data is available?

What format will the data be in? Where does it reside?

How can the data be accessed? Which fields are most important?

The key Considerations of data assessment are Collecting Ground Truth Data, Quantity of Data, and Ethics.

4.3.2 Data Exploration

This is a phase for creating meaningful summaries of your data, this is also the time you should test your assumptions. The key Considerations of data exploration are Missing Values, Outliers, and Un-balanced Data[5].

Machine learning performance is highly dependent on the type and quantity of available datasets. The face mask images collected are capture images with 4 types per person. The face mask protocol detection preprocessing component cleans the dataset by removing unnecessary background, bounding box, data cleaning, and classing techniques are applied to collected images. There is no data in Adama and Welenchiti Hospital on face masks.

4.4 Face Mask Image Preprocessing

Preprocessing images is an important step in training a successful model. It involves preparing the images in a way that makes them easier for the model to learn from. This can include balancing, cropping, assigning, and normalizing the images.

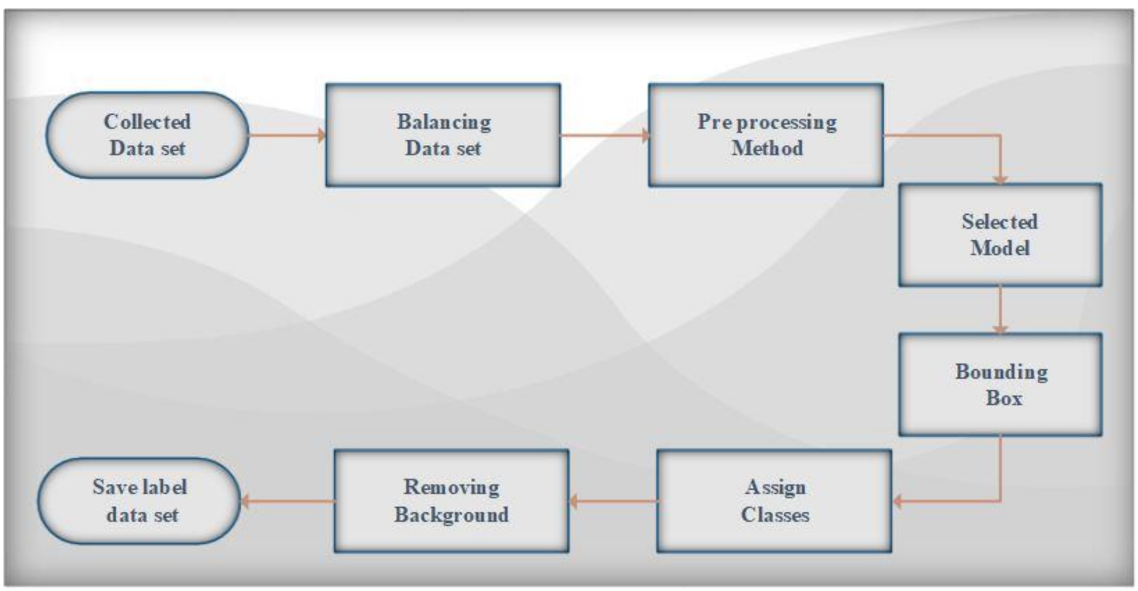


Figure 4.2 Dataset Preprocessing method

We opted for LabelImg as our image annotation tool. Created by Tzutalin with contributions from numerous collaborators, LabelImg has transitioned into the Label Studio community, as it is no longer under active development. LabelImg is a graphical image annotation tool developed using Python and Qt for its graphical interface. Annotations made with LabelImg are saved as XML files in the PASCAL VOC format, commonly used in ImageNet.

[5] https://ibm-cloud-architecture.github.io/refarch-data-ai-analytics/preparation/data-understanding/

Additionally, LabelImg supports YOLO and CreateML formats, providing versatility in annotation options[6].

4.4.1 Balancing Face Mask Image

In this step ensure that every person is captured in four images and that the images are balanced. it will be cheek by name, title, or image.

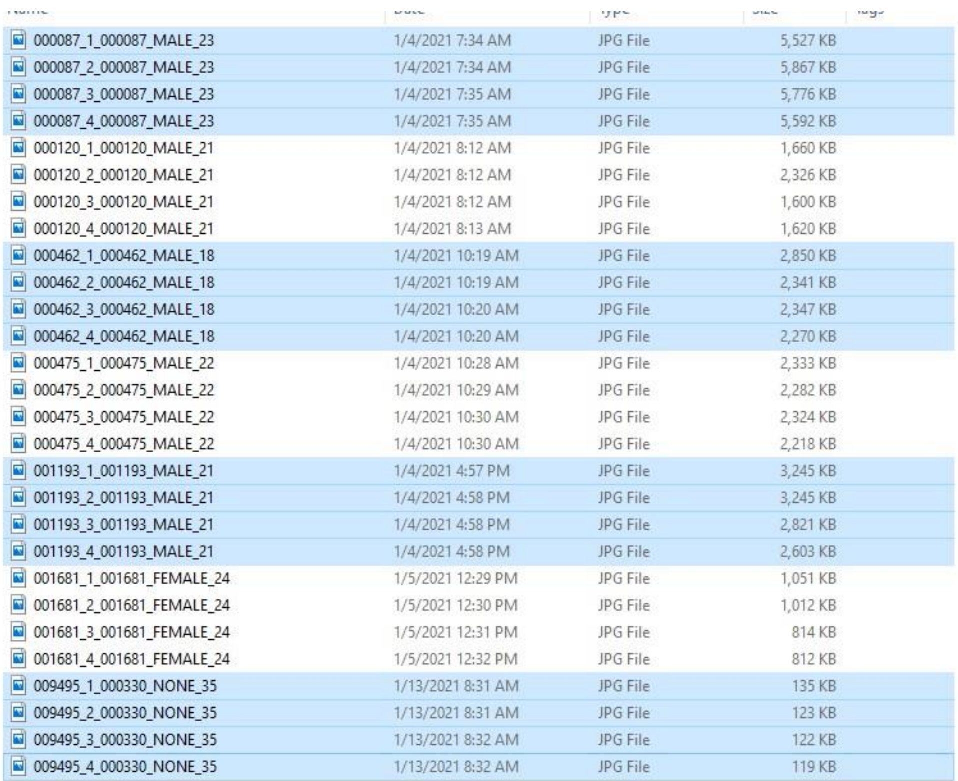


Figure 4.3 Checking balancing Face mask image

4.4.2 Processing Method

At processing method how to use labeling Multi-modal data annotation .it will work first install Get from PyPI but only python3.0 or above, and run in command prompt some commands then it opens user graphic box it shows in figure 4.4.

---

[6] https://github.com/HumanSignal/labelImg

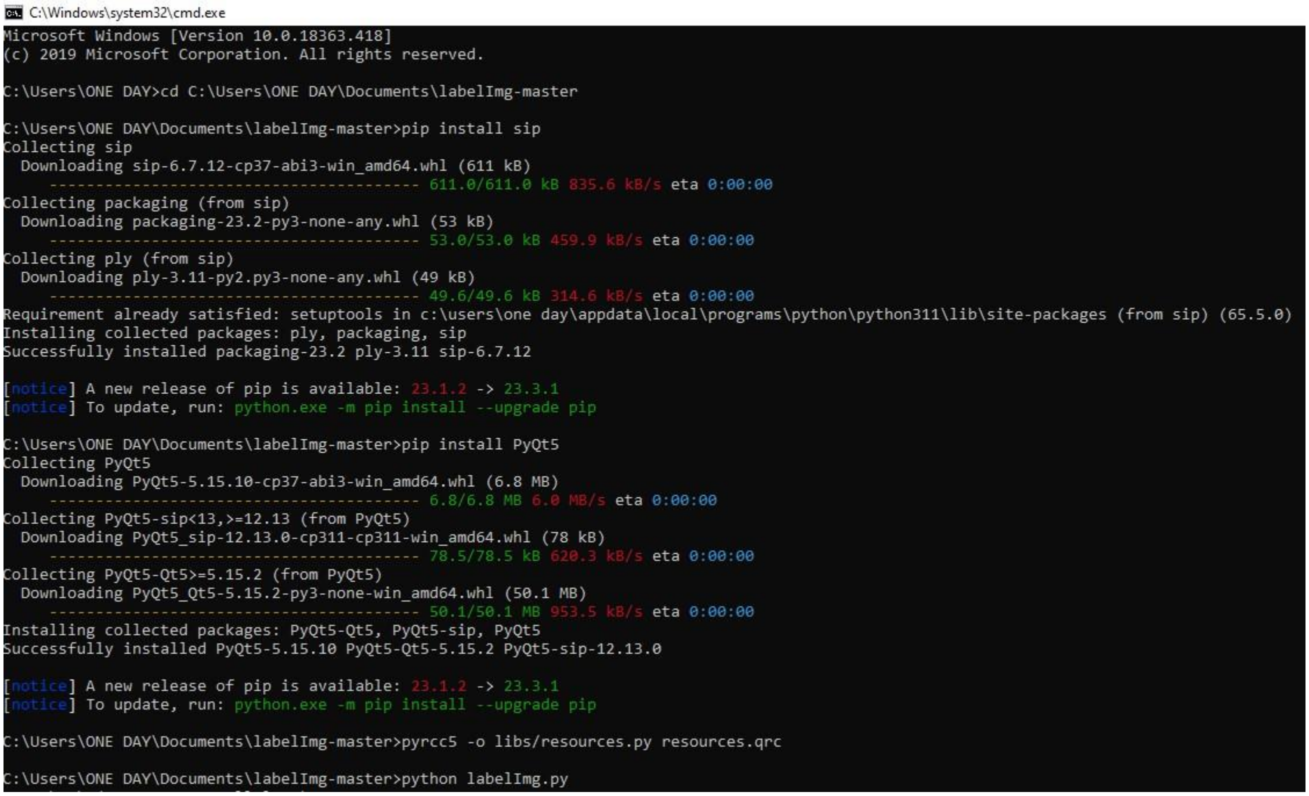


Figure 4.4 LabelIng Multi-modal data annotation running methods

4.4.3 Select Model

At select model labeling, Multi-modal data annotation works with some other models like Pascal Voc and Yolo.

Pascal VOC (Visual Object Classes) is a format to store annotations for localizer or object detection datasets. In Pascal voc format, for each image, there is an XML annotation file containing image details, bounding box details, classes, rotation, and other data[7].

YOLO (You Only Look once) is one of the selected algorithms for face mask protocol detection and the data set will be preprocessed, it will be saved file in supports YOLO and Create ML formats. We selected Yolo and it will create in accurate format.

[7]https://mlhive.com/2022/02/read-and-write-pascal-voc-xml-annotations-in-ython#:~:text=In%20PASCAL%20VOC%20format%2C%20for,2%20labels%20for%20a%20class.

4.4.4 Bounding Box and Removing Background

Bounding boxes are typically defined by four coordinates: the top-left x-coordinate, the top-left y-coordinate, the bottom-right x-coordinate, and the bottom-right y-coordinate. These coordinates can be normalized, meaning that they are represented as a percentage of the image width and height, or absolute, meaning that they are represented in pixels. Bounding boxes are a powerful tool for object detection. They can be used to represent objects of any shape or size, and they can be used to train machine learning models to accurately detect and localize objects in images.

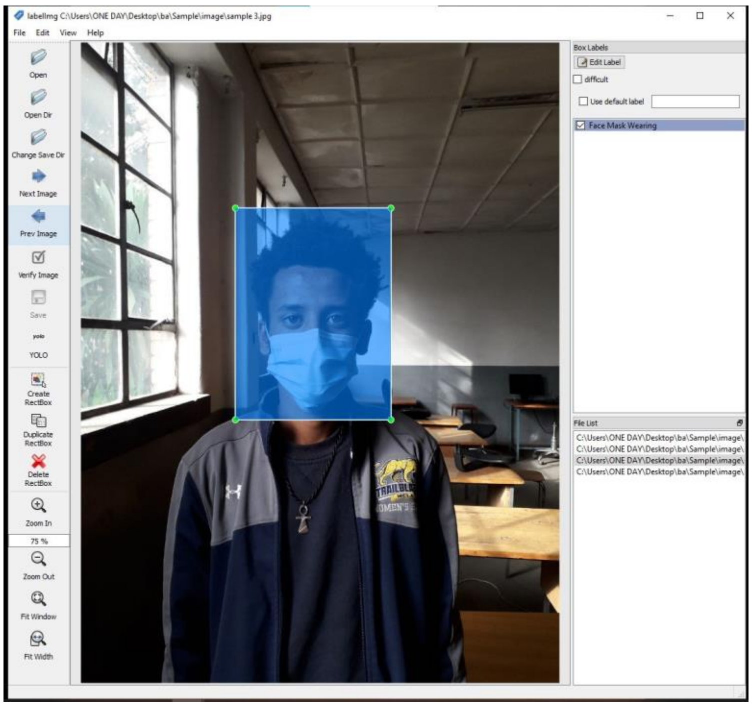


Figure 4.5 Bounding box in labeling multi-modal data annotation

4.4.5 Assign Classes

In dataset processing time there are three classes and in chapter three briefly describes how to assign and the resin to assign, the assign or classify class will be face mask-wearing, Incorrect form wearing a face mask, and not wearing a face mask.

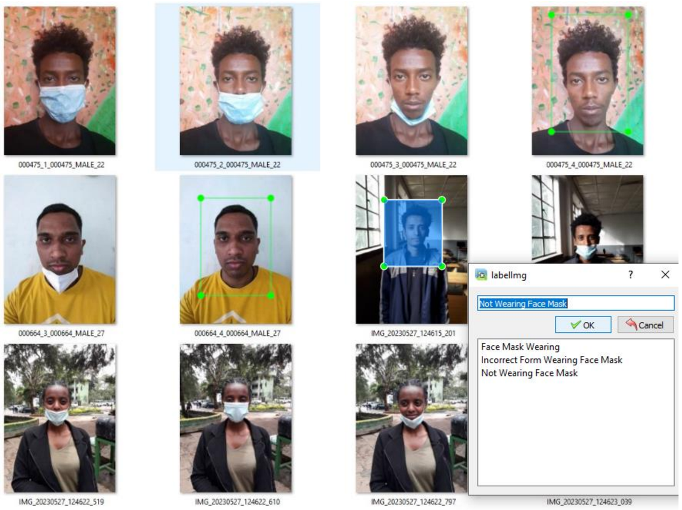


Figure 4.6 Assign class not wearing a face mask

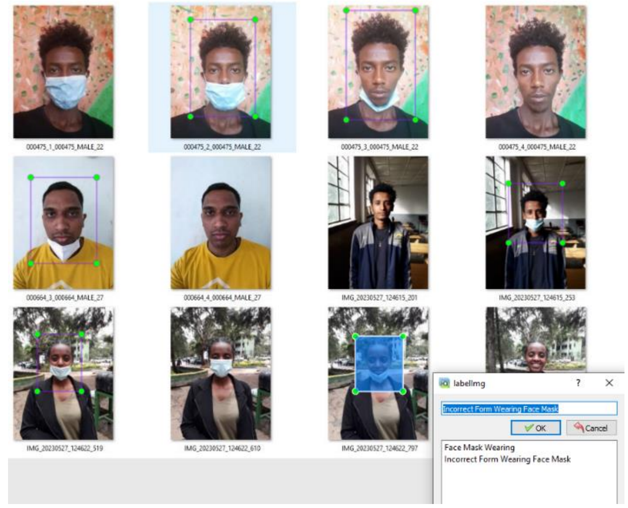


Figure4.7 Assign class in correct form wearing a facemask

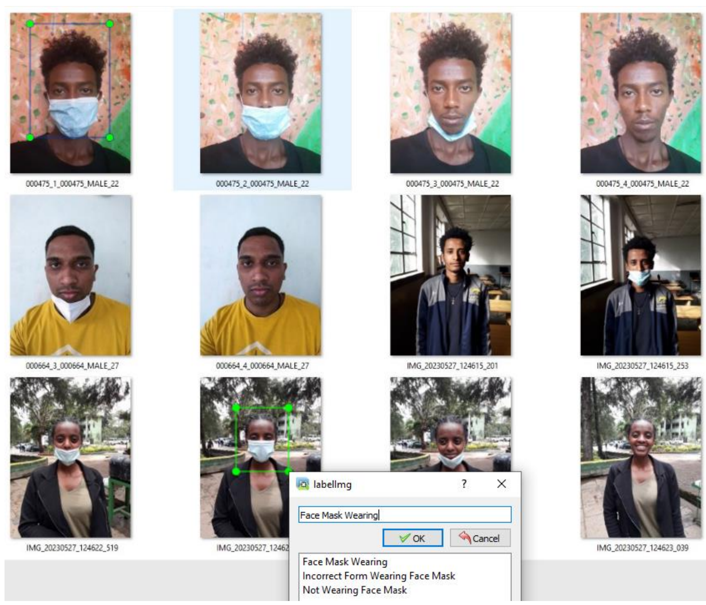


Figure 4.8 Assign class face mask-wearing

4.4.6 **Save Labeled Dataset**

Labeled data is raw data that has been assigned one or more labels to add context or meaning. These labels often serve as a target for the model to predict. Labeled data is fundamental because it forms the basis for supervised learning, a popular approach to training more accurate and effective machine learning models[8].

If labeled data is saved in a folder with some other files, assign classes and bounding boxes, and create a separate text file for each image in a new folder. The bounding box

[8] https://www.datacamp.com/blog/what-is-labeled-data

information from each labeled image file. It will then assign a class label to each bounding box and create a new text file in the output folder with the bounding box information for that image.

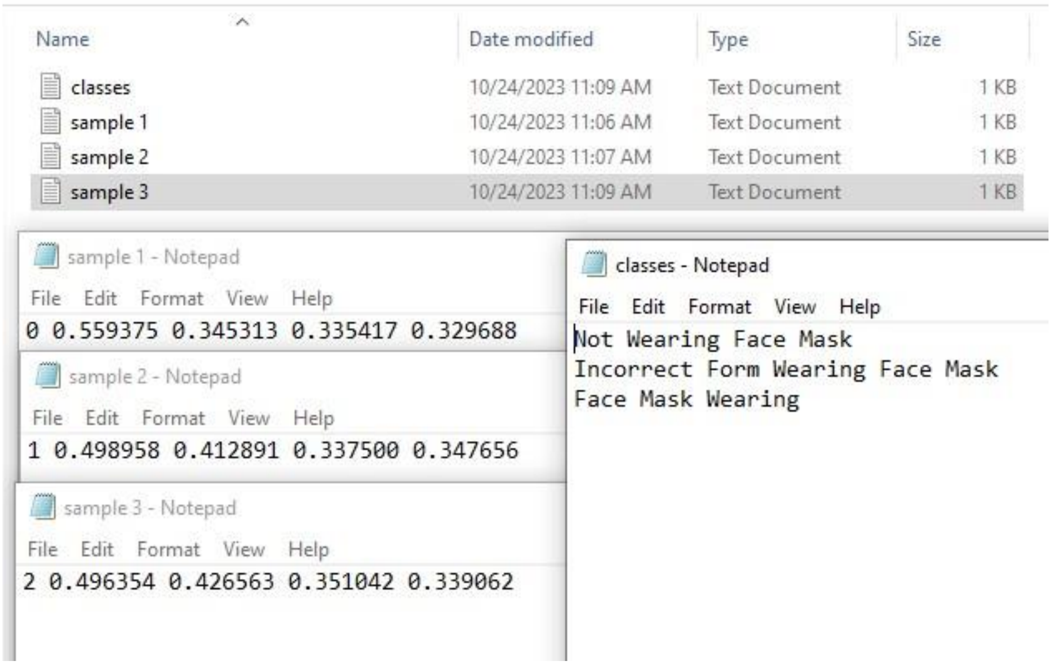


Figure 4.9 Saved labeled folder

4.5 Detection Model Building

This part of the proposed framework for facemask protocol detection uses a machine learning classifier on the dataset's facemask wearing, in correct wearing a facemask and not wearing a facemask. In this study, machine learning methods such as yolo v5, yolo v7, and yolo v8 create a detection model from which one will be chosen to detect face mask protocol detection. To evaluate the model, the dataset is separated into a training set and a test set. Using the learning algorithm, the modeling process detects patterns in the training set of the dataset and maps cases with their metrics (features) to the target class. Using the trained detection model.

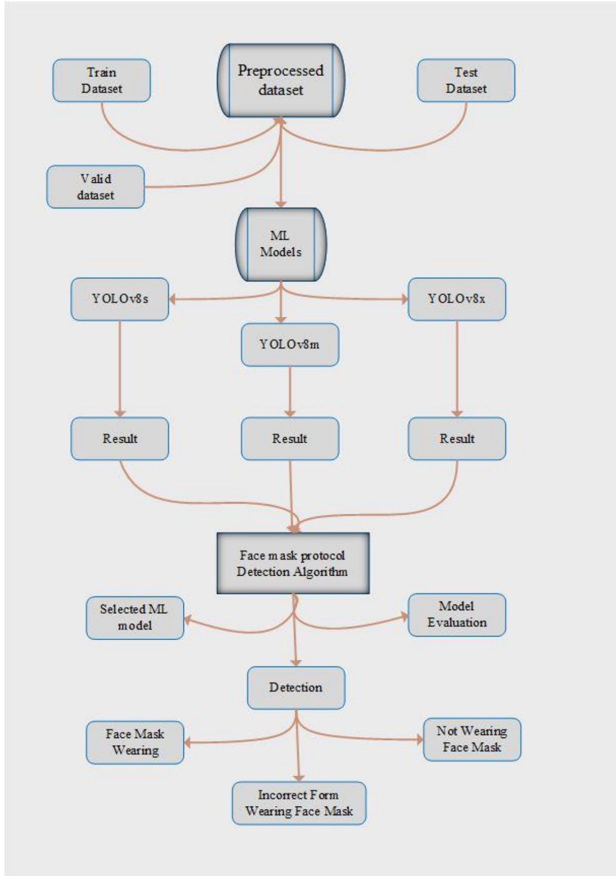


Figure 4.10 Face Mask Detection Model Building

YOLO-V5: - YOLOv5, introduced in 2020 by the team behind the original YOLO algorithm and maintained by Ultralytics, represents an open-source project. This iteration of YOLO builds upon the accomplishments of its predecessors while introducing numerous new features and enhancements. YOLOv5 adopts a more intricate architecture known as EfficientDet, which is based on the EfficientNet network architecture. Leveraging this sophisticated architecture enables YOLOv5 to attain higher accuracy and improved generalization across a broader spectrum of object categories.Yolov5 is commonly used for detecting objects. Yolov5 comes in four main versions: small (s), medium (m), large (l), and extra-large (x),[9].

YOLO-V7: - Yolov7 has demonstrated superior performance over both transformer-based and convolutional-based object detectors. Notably, it outperformed various object detectors such as YOLOR, YOLOX, Deformable DETR, DINO-5scale-R50, and ViT-Adapter-B. Its remarkable speed and accuracy have led to a 50% reduction in the cost of running the model on the same

[9] https://blog.roboflow.com/yolov5-improvements-and-evaluation/

dataset. Additionally, the parameters in the hidden layers of the neural networks have been reduced by up to 40%. Model scaling has been significantly simplified, as Yolov7 is capable of maintaining the original model design and structure while implementing compound scaling[10].

YOLO-V8: - YOLOv8 YOLOv8 is built upon a deep convolutional neural network (CNN) architecture similar to its predecessors, but it introduces several new components to enhance efficiency and accuracy. CSPNet backbone architecture, which improves efficiency and accuracy compared to previous backbones. FPN+PAN neck architecture is designed to better aggregate features from different levels of the backbone. PANet head architecture, which enhances robustness to occlusion and scale variations. A novel training procedure that combines supervised and unsupervised learning. YOLOv8 involves dividing the input image into a grid of cells. For each cell, YOLOv8 predicts a set of bounding boxes and class probabilities. It then utilizes a non-maxima suppression (NMS) algorithm to filter out overlapping bounding boxes and select the most likely bounding boxes for each object in the image[11].

4.6 Model Evaluation and Prototype

Model evaluation is used to estimate the generalization accuracy of the predictive model on the out-of-sample dataset. The face mask protocol detection model is trained with the entire dataset and its performance. Metrics that are appropriate to the model such as confusion matrix, accuracy, precision, recall, and f1score were discussed in detail in chapter three.

4.7 Prototype Development

How to understand how the model works in the real world, and how to work in the real world face mask protocol detection. it uses some security cameras if it is implemented in ASTU and Adama Hospital those are dome, bullet, and ptz cameras and it is controlled by IP address security passwords. The best-performing detection is deployed to the flask server using an HTML web page. The employer or camera controller gets by configuring IP address and correct user name and password, then if a person enters, sit, sleeping the model bounding box on the face and write face mask wearing or incorrect form wearing a face mask or not wearing a face mask in controlled screen.

---

[10] https://pythonistaplanet.com/yolov7/

[11]https://medium.com/@muhammadshabrozshahab/yolo-v8-104f1375242c#:~:text=Working%20Principle%3A%20YOLOv8%20is%20a%20state%2Dof%2Dthe%2D,for%20their%20speed%20and%20accuracy.

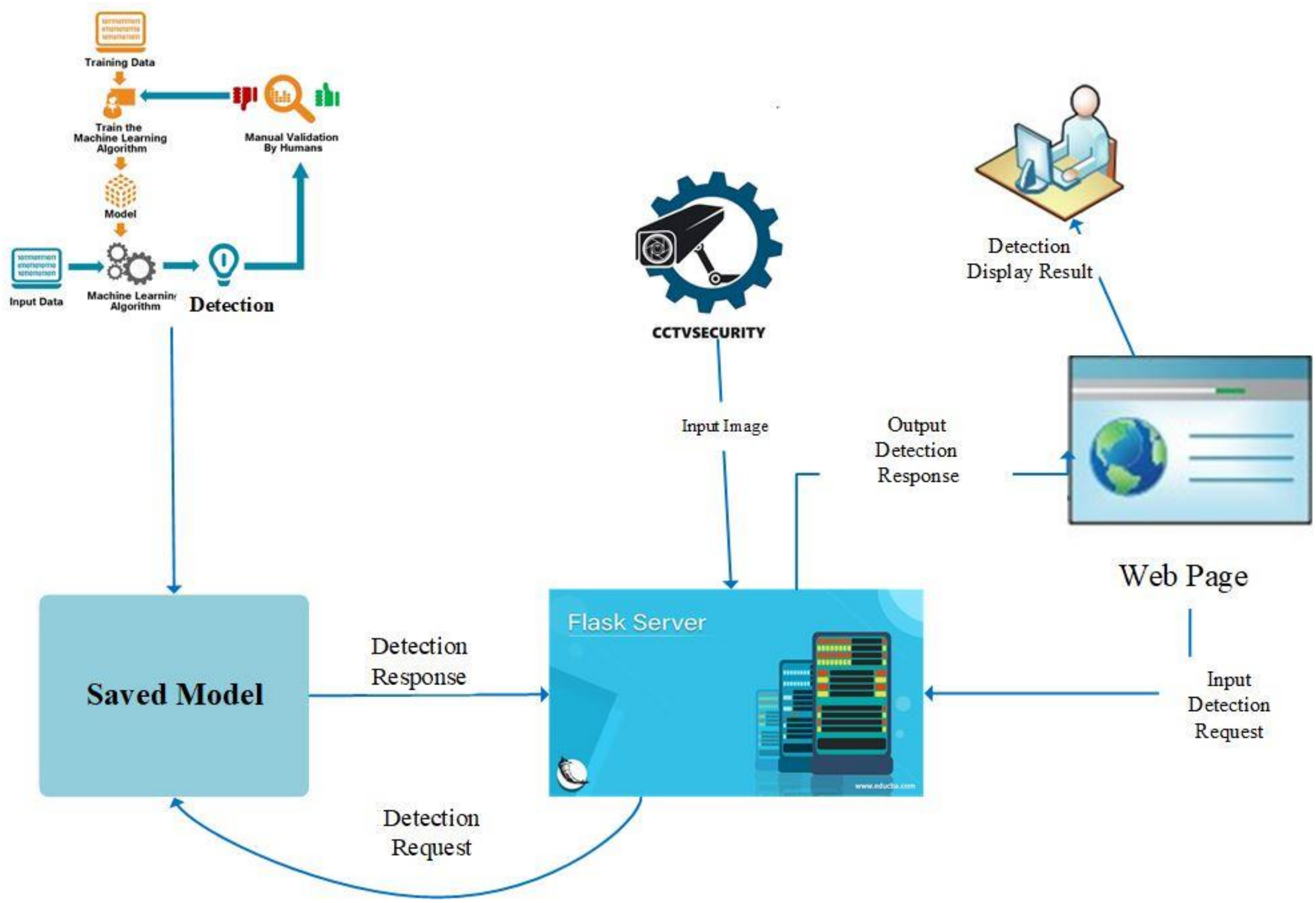


Figure 4.11 Prototype Development Architecture

CHAPTER FIVE

5. IMPLEMENTATION AND EXPERIMENTATION

5.1 Chapter Overview

This chapter describes the instruments required for experimentation to achieve the suggested framework for face mask protocol detection, as well as the methods used to implement the models to be evaluated. The environment was configured as outlined in the third chapter to put the suggested strategy into practice. Python programming, the Kearas framework backend as Tensor Flow, Pandas, and other components are included in the package. While significant detail codes were illustrated in the appendix section, significant code segments were presented in this section.

5.2 Working Environment

The suggested method has been put into practice using several kinds of development tools and packages. Every suggested solution is implemented and tested using the Python programming language, from preprocessing through model construction. To choose the best model, it is also used to assess the models that have been put into practice. We go over the instruments that were utilized to create and apply the suggested model.

5.2.1 Computers with Hardware Functions

- Hard Disk: 1TB
- Processor: Intel® Core™i5, 7th Gen 2.3GHz
- RAM: 4GB
- Graphics: Intel HD Graphics 620

5.2.2 Computers with Software Configuration

- Operating system: Windows 10 Pro
- System Type: x64-based processor
- For documentation purposes: Microsoft Word 2016
- For designing purposes: Microsoft Visio 2016

Google Collaboratory (Google Collab) is a free Jupyter Notebook environment for a maximum of 12 hours from the Google cloud server, is used for developing or training a proposed model.

5.3.Setup Environments

Setup environments are the process of creating a computing environment to develop and run thesis research. This includes installing the necessary software and hardware and configuring them to work together.

5.3.1 Application Software

Web browsers: Web browsers are used to access the internet and view web pages. We will be using web browsing and to working with Google Colab, Chrome, and Firefox.

5.3.2 Integrated Development Environment and Editors

Jupyter Notebook: is a web-based application that contributes a front-end to many different languages and interactive shells such as Python. The Jupyter Notebook with a 6.4.13 version is used to execute the proposed model.

Google Colab: is a cloud-based Jupyter Notebook environment that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. We can train our model for 12 hours using Colab GPUs.

5.3.3 Programming Language and Module Libraries

Python: Python is a general-purpose programming language that is used for a wide variety of tasks, including web development, data science, machine learning, and artificial intelligence. It is a popular language for beginners and experienced programmers alike, due to its simplicity and readability.

5.4.Training Procedure

The training procedure in this proposed research

- Collect massive dataset in wearing a face mask, incorrect form wearing a face mask, not wearing a face mask.
- Pre-processing dataset
- Scaling and transforming the data by training and test split
- Building the model
- Train the model

5.5. Dataset Processing

Dataset processing for YOLOv8 involves several key steps to prepare our data for effective object detection training. Here's a summary, incorporating the best aspects of previous responses and addressing potential issues:

Data Format: Images ensure consistent image format and resolution across your dataset. Consider using a consistent aspect ratio. Annotations choose an annotation format compatible with YOLOv8, such as txt, COCO (.json), or Pascal VOC (.xml). Ensure bounding boxes accurately represent object locations and appropriate class labels. Class definitions a unique label for each object class you want the model to detect and maintain consistent label names throughout the dataset.

Data Cleaning and Preprocessing: Missing values address missing bounding boxes or image files appropriately be mindful of maintaining class balance if deleting. Normalization consider normalizing pixel values in each image for better training stability. Data augmentation apply techniques like random cropping, flipping, blurring, and color jittering to artificially increase data size and improve model generalizability. Resize YOLOv8 supports various input resolutions, choose a consistent size based on your dataset and hardware limitations. Consider using the input resolution recommended by your chosen YOLOv8 variant.

Dataset Splitting: Train-validation-test split: Divide your dataset into separate sets for training (majority), validation (tuning hyperparameters), and testing (final performance evaluation). A common split is 70% training, 15% validation, 15% testing.Stratification: If your dataset has imbalanced classes, ensure each set maintains the class proportions to avoid biased model learning.

Implementation Methods: Custom Python scripts write your own scripts using libraries like OpenCV or imgaug to implement preprocessing steps tailored to your dataset and format. Preprocessing libraries utilize libraries like albumentations or imgaug, which offer various built-in preprocessing functions and data augmentation techniques.YOLOv8 preprocessing tools YOLOv8 supports built-in preprocessing tools like dataset.py and images.py that can handle resizing, normalization, and some augmentation techniques.

5.6.Preprocessing Implementations

When working with YOLOv8, specific preprocessing steps are crucial for optimal training and accurate object detection. Here's a breakdown of key implementations to consider:

Image Preprocessing: Resize Choose a consistent image resolution based on your dataset and hardware limitations. YOLOv8 supports various input sizes, but maintaining consistency is important. Normalize pixel values (e.g., 0-1 or mean-standard deviation) across each image for improved training stability. Apply random color jittering (brightness, contrast, saturation, hue) for data augmentation and improving model robustness.

Bounding Box Preprocessing: Format standardization: Ensure consistent bounding box format YOLOv8, COCO, Pascal VOC based on our annotation tool and YOLOv8. Coordinate normalization Normalize bounding box coordinates based on image resolution. Clip bounding boxes to stay within image boundaries to avoid invalid values during training.

Label Handling: Verify consistent and unambiguous object class labels across annotations. Use one-hot encoding for multi-class classification, representing each class as a binary vector. Decide if you want a "background" class for non-object regions and handle it accordingly.

Data Augmentation: Randomly crop images and corresponding bounding boxes to increase dataset size and introduce variations. Flip images horizontally and adjust bounding boxes accordingly for improved model generalizability. Other techniques: Explore additional augmentations like random scaling, rotation, blurring, and mosaic augmentation based on your dataset and needs.

Implementation Methods: Write our scripts using libraries like OpenCV or imaging to implement preprocessing steps tailored to your dataset and format. Utilize libraries like augmentations or imaging, which offer various built-in preprocessing functions and data augmentation techniques. YOLOv8 supports built-in preprocessing tools like dataset.py and images.py that can handle resizing, normalization, and some augmentation techniques.

5.7.Detection Models Implementation

Object detection models are a type of artificial intelligence (AI) that can be used to identify and locate objects in images and videos. They are widely used in a variety of applications, including security, surveillance, robotics, and others. Object detection models are typically trained on a large dataset of images and videos that are labeled with the location of objects. The model is then able to use this training data to identify and locate objects in new images and videos. There are many different types of object detection models, but they can be broadly classified into two categories:

Two-stage detectors: These detectors first generate a set of object proposals, and then classify each proposal as an object or not. Examples of two-stage detectors include R-CNN, Fast R-CNN, and Faster R-CNN.

One-stage detectors: These detectors directly predict the class and location of objects in an image or video. Examples of one-stage detectors include YOLO, SSD, and Retina Net.

In this, we proposed a machine learning model stacking ensemble that is compared with other models YOLO are good model that is implemented to face mask protocol detection. For the proposed YOLO model we have compared three models which are YOLOv8x, YOLOv8m, and YOLOv8s, and selected the accurse model. In the training time, there are 16 batches and 224 images.

5.7.1 Model Training

Model training is the process of teaching a machine learning model to perform a specific task. This is done by feeding the model some amount of dataset that is labeled with the desired output. The model then learns to identify patterns in the data and use those patterns to make detection. YOLO are there are 5 models those are. We selected YOLOv8s and The training procedure cycled through the dataset, known as epochs, a predetermined number of times and we have assigned 20 epochs. images refers to the input image size of the model. It is a hyperparameter that determines the size of the images that the model is trained on and used to make detection and we assignee 224. "data=data.yaml" refers to the configuration file that specifies the dataset to be used for training or inference. The "data. yaml" file contains information about the location of the dataset, the classes of objects to be detected, and other parameters related to the dataset. "plots=True" is an optional flag that can be used to enable the generation of plots during training and inference. When this flag is set to True, the model will generate plots that visualize the detected objects and their bounding boxes. This can be useful for debugging and evaluating the performance of the model. "%cd /content/drive/MyDrive/Kidu" is used to navigate to the directory where the YOLO model files are located. This is necessary because the YOLO model needs to access the configuration files, training data, and other files that are located in this directory.

Table 5.1 Types of the YOLO model[12]

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

5.8.Implementation of Model Evaluation

The implementation of model evaluation is one of the most important steps in the deep learning process. The process of assessing the performance of a trained model on a dataset that is not used for training. As it helps to ensure that, the model can generalize to new data and that it is not simply overfitting the training data. We evaluated our model's performance using evaluation metrics such as the confusion matrix, average loss of the model's bounding box, precision, recall, Mean Average Precision (mAP).

[12] https://github.com/ultralytics/ultralytics

CHAPTER SIX

6. RESULTS, EVALUATION, AND DISCUSSIONS

6.1 Chapter Overview

This chapter delves into the experimental results obtained from this study, providing a comprehensive overview of model performance, evaluation metrics, and image data preprocessing techniques. Additionally, it presents the performance of selected models and a newly developed stack ensemble compared to three selected models. Finally, the best-performing model is deployed using a Flask server. A thorough comparison with previous studies is also included to demonstrate how this study addresses gaps identified in the literature review, answers the research question, and implements the specific research objectives outlined.

6.2 Face Mask Protocol Detection Image Preprocessing Result

To evaluate the impact of image preprocessing on face mask detection accuracy, experiments were conducted using various preprocessing techniques and a face mask detection model. The results demonstrated that image preprocessing significantly improved detection accuracy compared to using raw images. We show the result in local data sets and global datasets in different epochs and models.

6.2.1 Collected Datasets Image Result

The local dataset image results include the original image, the predicted bounding boxes for the detected facemask, and the corresponding class labels. These results are used to train and test the model; all experiments are conducted on local images. We display the results in several epochs and models.

Figure 6.1 In 20epoch YOLOv8s Model

Figure 6.2 In 20epoch YOLOv8m Model

Figure 6.3 In 20epoch YOLOv8x Model

6.2.2 Public datasets Image Result

The global dataset image results include the original image that will be downloaded from Kaggle, the predicted bounding boxes for the detected facemask, and the corresponding class labels. These results are used to train and test the model; all experiments are conducted on global images. We display the results in several epochs and models.

Figure 6.4 In 50epoch YOLOv8s Model *Figure 6.5 In 50epoch YOLOv8m Model* *Figure 6.6 In 50epoch YOLOv8x Model*

6.3 Experimental Setup

This experimental setup provides a framework for evaluating and comparing various models for face mask protocol detection. By carefully collecting data, preprocessing it effectively, and choosing appropriate models and metrics, researchers can develop accurate and reliable face mask detection systems for various applications.

The experiments were conducted in the Jupyter Notebook environment using Python and various related packages for implementation and simulation. Preprocessed datasets were utilized for training and testing various models for face mask detection and prediction. To evaluate the model's performance for the classification task, various metrics were applied, including accuracy, precision, recall, F1 score, and a confusion matrix. A detailed analysis and comparison of the models' results are presented in the following sections.

6.4 Model Learning Curves

A learning curve is just a plot showing the progress over the experience of a specific metric related to learning during the training of a machine learning model. Learning curves are a widely used diagnostic tool in machine learning for algorithms that learn from a training dataset incrementally. The model can be evaluated on the training dataset and a hold-out validation dataset after each update during training and plots of the measured performance can created to show learning curves. Reviewing the learning curves of models during training can be used to diagnose problems with learning, such as an underfit or overfit model, as well as whether the training and validation datasets are suitably representative[13].

6.4.1 Precision-Confidence Curve

The Precision-Confidence Curve (PC curve) is a vital tool in object detection and classification tasks, offering valuable insights into your model's confidence and decision-making process. Unlike the single-point metric of MAP (Mean Average Precision), the PC curve provides a more nuanced understanding of how model performance changes as its confidence threshold varies. The PC curve is a graph that plots the precision (positive predictive value) of your model's predictions on the x-axis, and the confidence score of those predictions on the y-axis. Each point on the curve represents the precision achieved at a specific confidence threshold[14].

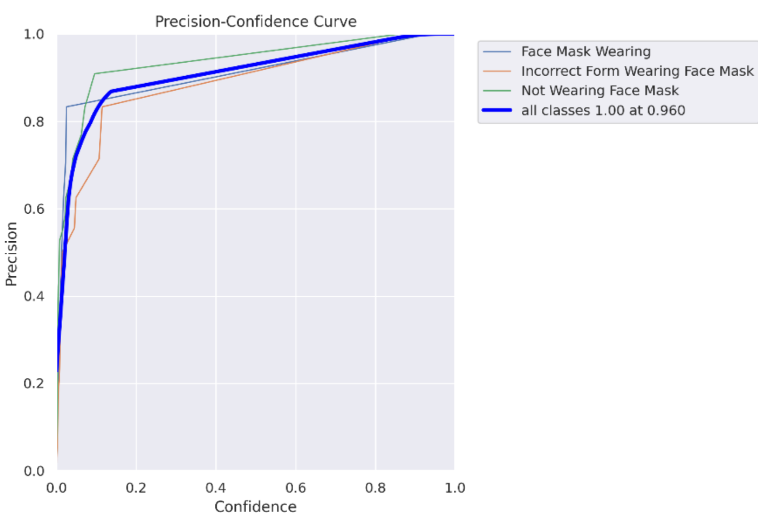


Figure 6.7 In Collected dataset 20 epoch YOLOv8s Model

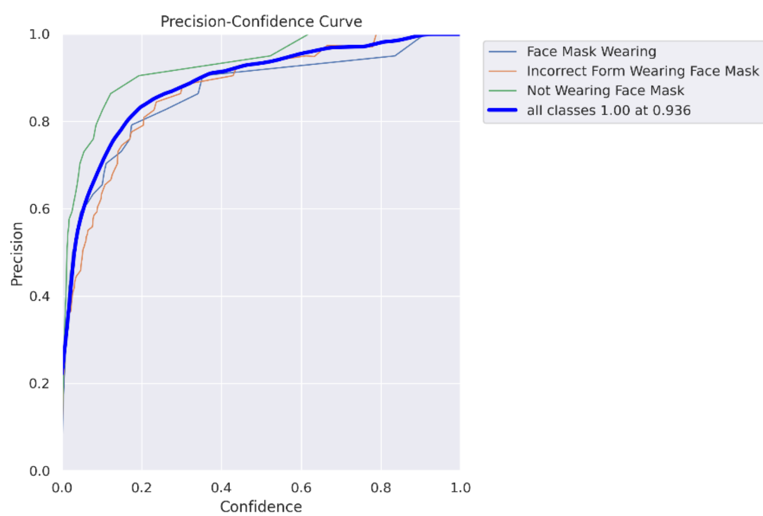


Figure 6.8 In Public Dataset 50 epoch YOLOv8m Model

[13] https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/
[14] https://github.com/ultralytics/ultralytics/issues/5122

6.4.2 Recall-Confidence Curve

Recall-Confidence Curve is a valuable tool for analyzing the model's handling of false negatives, optimizing confidence thresholds for recall-driven tasks, and identifying potential calibration issues. By understanding and interpreting this curve alongside other metrics like the PC curve, you can gain a deeper understanding of your model's overall performance and make informed decisions to improve its effectiveness. The RC Curve is a graph that plots the recall (true positive rate) of your model's predictions on the x-axis, and the confidence score of those predictions on the y-axis. Each point on the curve represents the recall achieved at a specific confidence threshold.

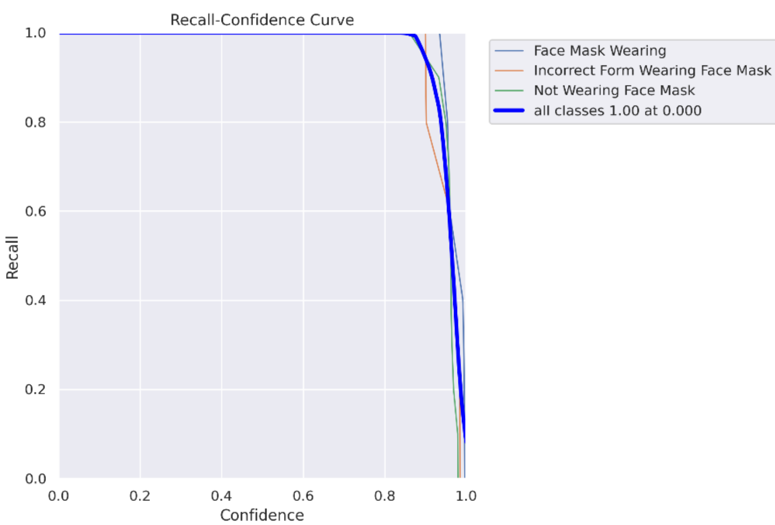


Figure 6.9 In Collected Dataset 20 epoch YOLOv8s Model

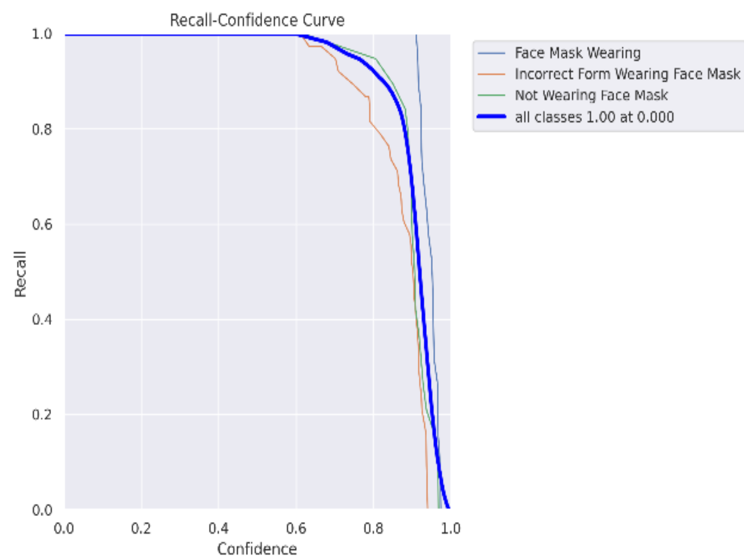


Figure 6.10 In Public Dataset 50 epoch YOLOv8m Model

6.4.3 Precision-Recall Curve

The precision-recall curve is used for evaluating the performance of binary classification algorithms. It is often used in situations where classes are heavily imbalanced. Also like ROC curves, precision-recall curves provide a graphical representation of a classifier's performance across many thresholds, rather than a single value e.g., accuracy, f-1 score[15]. The Precision-Recall Curve (PR Curve) is a fundamental tool in the world of classification tasks, visualizing the critical trade-off between precision and recall as the model's confidence threshold varies. By

[15] https://medium.com/@douglaspsteen/precision-recall-curves-d32e5b290248

understanding this curve, you can gain valuable insights into your model's performance and make informed decisions about its deployment.

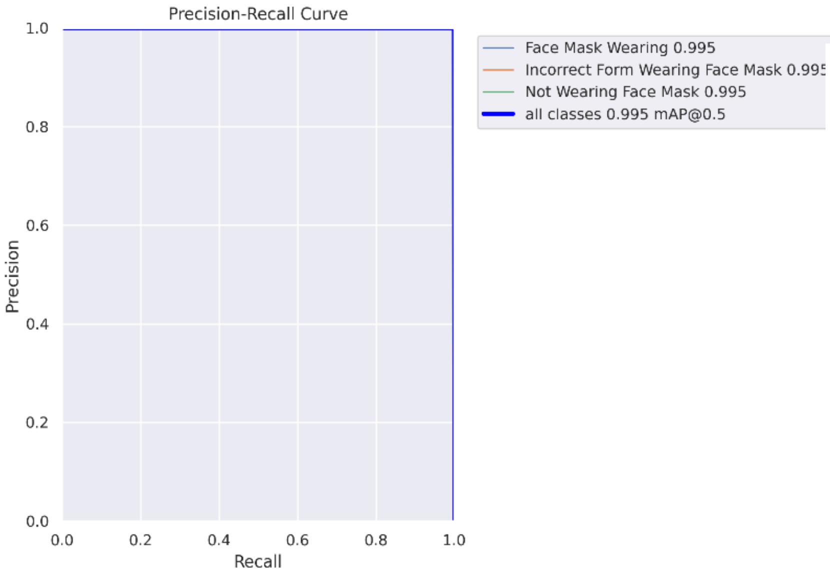


Figure 6.11 In Collected Dataset 20 epoch YOLOv8s Model

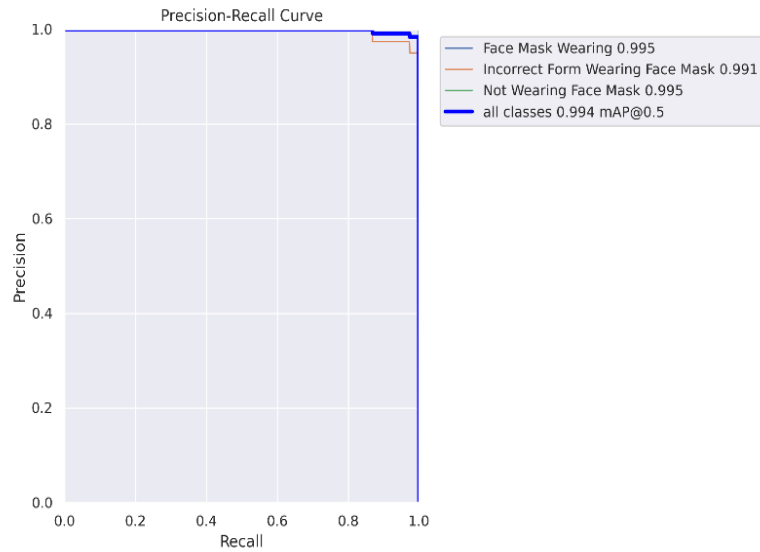


Figure 6.12 In Public Dataset 50 epoch YOLOv8m Model

6.4.4 F1_curve

The F1_curve, closely related to the Precision-Recall (PR) curve, provides another insightful perspective on your model's classification performance. While the PR curve focuses on the trade-off between precision and recall, the F1_curve highlights the balanced performance of your model across different confidence thresholds. F1_curve is a valuable tool for evaluating your model's classification performance, focusing on its ability to strike a balance between precision and recall. By interpreting the curve effectively and considering other metrics, you can make informed decisions about choosing the optimal confidence threshold, improving model calibration, and ultimately enhancing your model's effectiveness for your specific task.

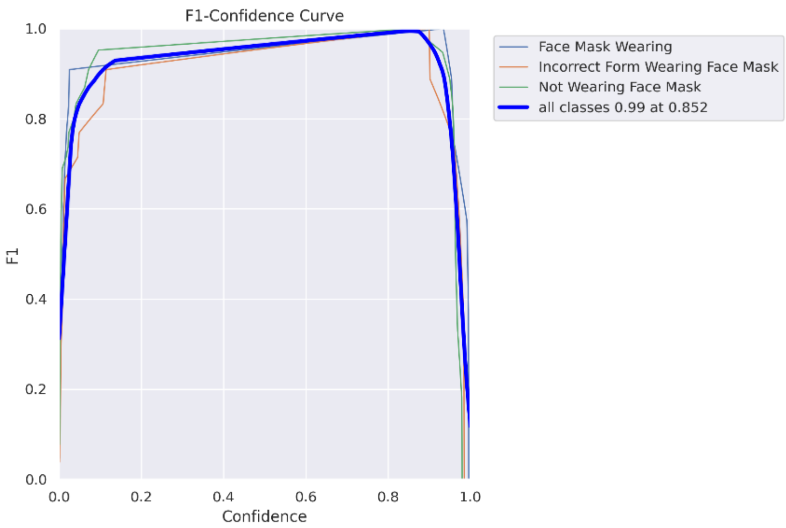


Figure 6.13 In Collected Dataset20 epoch YOLOv8s Model

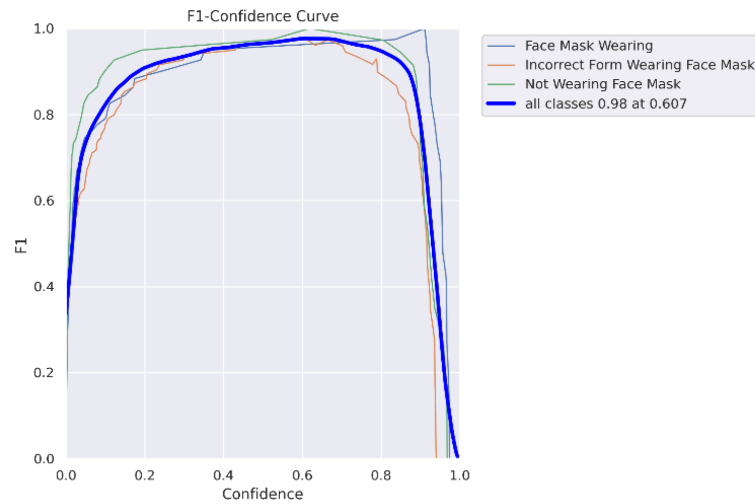


Figure 6.14 In Public Dataset 20 epoch YOLOv8m Model

6.5 Models Performance and Evaluation Result

The performance and evaluation results of YOLOv8s, YOLOv8m, and YOLOv8x-based models on face mask protocol detection the confusion matrix, and other performance metrics are used in testing and training models performance select local datasets and other global datasets. In this study, the performance of selected models with local dataset and global dataset is discussed to identify the most performing model based on the result.

6.5.1 Face Mask Protocol Detection Model Result

Box_loss:- represents the average loss calculated for the predicted bounding boxes during the training process.

Cls_loss:- refers to the classification loss calculated during the training process. It measures how well the model is correctly classifying the objects within the detected bounding boxes.

Dfl_loss:- refers to the Distribution Focal Loss calculated during the training process. This metric specifically measures the performance of the model's bounding box regression with a focus on addressing class imbalance issues.

Metrics/precision (B):- refers to the precision score calculated for the best model performance achieved during training.

Metrics/recall (B):- refers to the recall metric calculated at the box level. This means it measures the percentage of true positive bounding boxes (correctly identified objects) out of all ground truth bounding boxes.

Metrics/mAP50 (B):- is a specific metric used to evaluate the model's performance on object detection tasks.

Metrics/mAP50-95(B):- refers to a specific metric used to evaluate the performance of the model in object detection tasks.

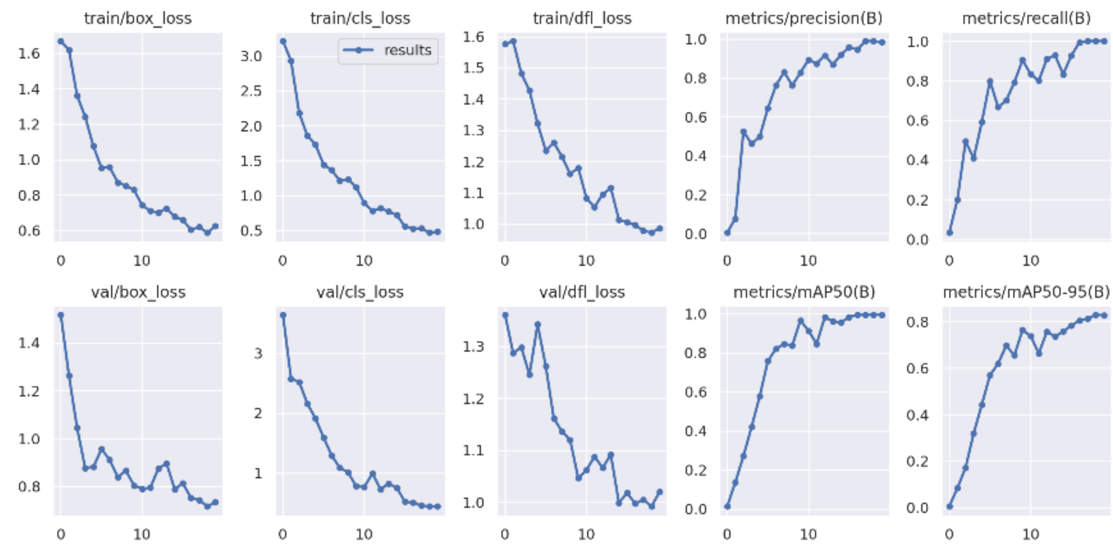


Figure 6.15 In Collected Dataset 20 epoch YOLOv8s Model

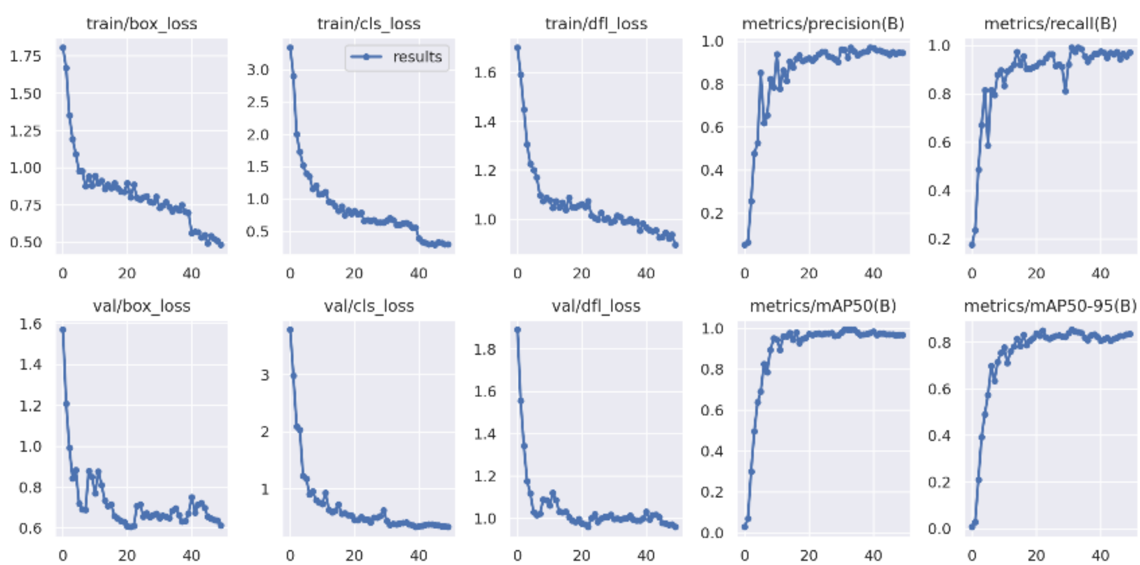


Figure 6.16 In Public Dataset 50 epoch YOLOv8m Model

6.5.2 Face Mask Protocol Detection Confusion_Matrix

In Chapter Three Confusion Matrix is described but in this chapter, we delve into the analysis of confusion matrices to uncover the performance of YOLOv8s, YOLOv8m, and YOLOv8x models in face mask protocol detection. Specifically, we examine their behavior on both local and global datasets to reveal potential differences in model performance across diverse data distributions.

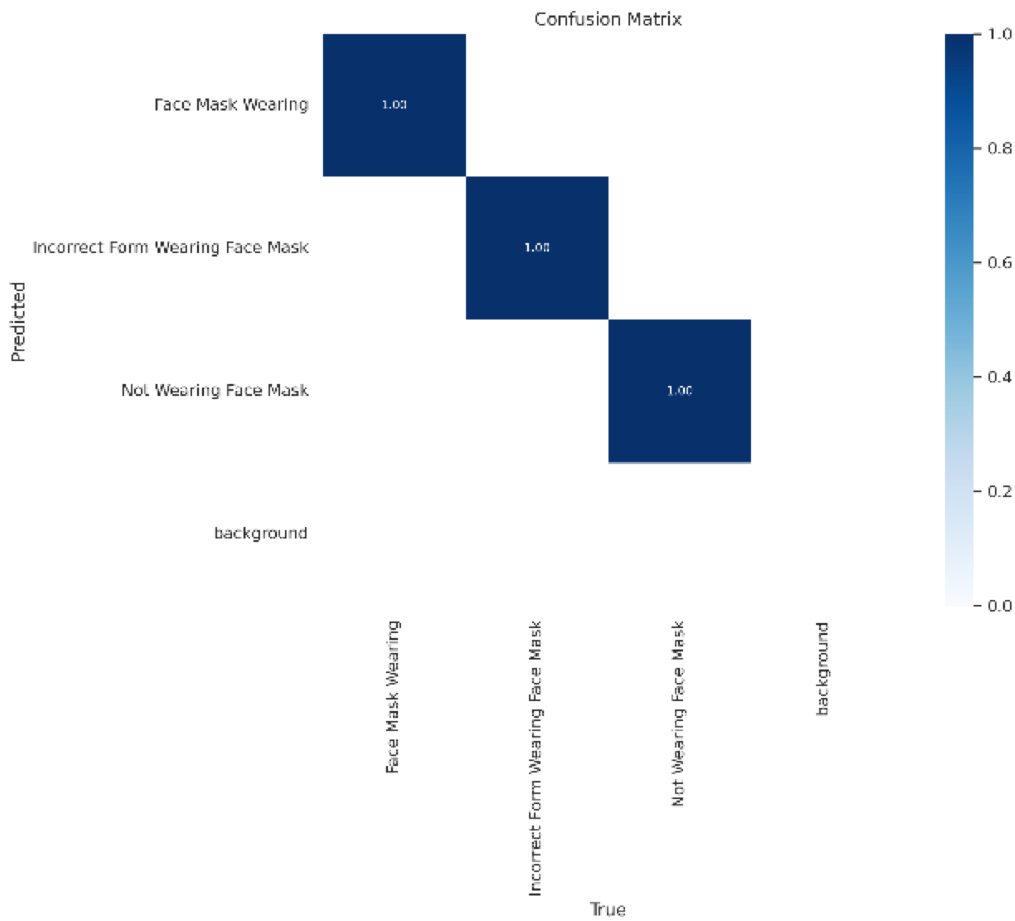


Figure 6.17 In Collected Dataset 20 epoch YOLOv8s Model

Figure 6.18 In Public Dataset 50 epoch YOLOv8m Model

6.5.3 Face mask protocol detection Performance metrics reports

Performance metrics reports are essential tools for evaluating the effectiveness of various models. They provide valuable insights into strengths, weaknesses, and areas for improvement, enabling informed decision-making and optimization. As shown in the table, among the three models, YOLOv8s, YOLOV8m, and YOLOV8x and facemask protocol detection performance metrics such as model types, dataset types, epoch, Precision-Confidence, Precision-Recall, and F1 score. The experimental results are shown in the following table.

Table 6.1 In Collected Dataset Performance metrics reports

Model Types	Epochs	Precision Confidence	F1 Confidence	Precision-Recall mAP
YOLOv8s	10	0.975	0.85	0.950
YOLOv8m	10	0.916	0.89	0.977
YOLOv8x	10	0.840	0.88	0.912
YOLOv8s	**20**	**0.960**	**0.99**	**0.995**
YOLOv8m	20	0.907	1.00	0.995
YOLOv8x	20	0.876	1.00	0.995
YOLOv8s	50	0.912	0.99	0.995
YOLOv8m	50	0.876	1.00	0.995
YOLOv8x	50	0.913	1.00	0.995

Table 6.2 In Public Dataset Performance metrics reports

Model Types	Epochs	Precision Confidence	F1 Confidence	Precision-Recall mAP
YOLOv8s	10	0.994	0.82	0.840
YOLOv8m	10	0.998	0.95	0.976
YOLOv8x	10	1.000	0.85	0.955
YOLOv8s	20	0.924	0.97	0.991
YOLOv8m	20	0.956	0.95	0.989
YOLOv8x	20	0.997	0.95	0.984
YOLOv8s	50	0.948	0.93	0.988
YOLOv8m	**50**	**0.936**	**0.98**	**0.994**
YOLOv8x	50	0.940	0.97	0.992

6.6 Tuning Hyperparameters Result

Hyperparameter tuning results refer to the outcomes of face mask protocol detection to achieve the best possible performance on a given task. There are two types of datasets local and global datasets using for the experiment we used three machine-learning algorithms YOLOv8s, YOLOv8m, and YOLOv8x. Training takes up 80% of the time, while testing takes up 20%. In the local dataset, the best performance will be in the below table.

Table 6.3 In the collected dataset best Performance

Model Types	Epochs	Precision Confidence	F1 Confidence	Precision-Recall mAP
YOLOv8s	**20**	**0.960**	**0.99**	**0.995**

In the global dataset, the best performance will be in the below table.

Table 6.4 In the Public dataset best Performance

Model Types	Epochs	Precision Confidence	F1 Confidence	Precision-Recall mAP
YOLOv8m	**50**	**0.936**	**0.98**	**0.994**

6.7 Discussion

At the beginning of this study, three research questions were addressed. Hence, this section tries to answer those questions.

Answer for question 1: What is the impact of dataset type and diversity on the performance of facemask detection models?

In this experiment, when a black person wears a black facemask, the model may produce incorrect predictions. This could occur due to difficulties in distinguishing between the dark-colored mask and the individual's skin tone, leading to inaccurate detection by the model. The results of this experiment will be included in Appendix E.

Answer for question 2: What are the necessary features to prepare an effective facemask protocol dataset in facial annotation?

In this experiment, we selected some dataset preprocessing models but we selected LabelIng Multi-modal data annotation used to build the preprocessing techniques and an effective face mask protocol dataset preprocessing methods. Labeling Multi-modal data annotation in chapter 4.4 described how it is working.

Answer for question 3: Which deep learning algorithm is more accurate for the prediction of Facemask protocol detection?

For facemask protocol detection, we selected two models in local dataset and the global dataset. In the experiment result if using the local dataset the higher performance will be the YOLOv8s model and the good result in the global dataset higher performance will be the YOLOv8m model. In chapter six all experiments will be in the local dataset and global dataset described.

The good results observed in the collected dataset after training a YOLOv8s model for 20 epochs and in public dataset after training, a YOLOv8m model for 50 epochs can be attributed to several factors:

Model Architecture: YOLOv8 is a highly efficient and accurate object detection model known for its state-of-the-art architecture. It incorporates advanced features such as CSPNet, FPN+PAN, and PANet, which enhance feature extraction and object localization capabilities.

Dataset Quality: The collected dataset likely contains high-quality annotations with accurately labeled bounding boxes and class labels. A high-quality dataset is essential for training a robust object detection model, as it provides the necessary information for the model to learn and generalize well to unseen data.

Sufficient Training Duration: Training the model for collected dataset 20 epochs and public dataset 50 allows it to iteratively learn from the dataset multiple times. This extended training duration enables the model to converge to a more optimal solution, capturing complex patterns and variations present in the data.

Effective Training Parameters: The choice of training parameters, such as learning rate, batch size, and optimization algorithm, plays a crucial role in model training. Properly tuning these parameters can lead to faster convergence and better performance. The selected parameters during training are likely effective in facilitating model learning and convergence.

Validation and Fine-tuning: The model may have been validated on a separate validation dataset, and hyperparameters may have been fine-tuned based on validation performance. This iterative process ensures that the model generalizes well to unseen data and achieves good performance across different datasets.

Overall, the combination of a strong model architecture, high-quality dataset, effective training parameters, regularization techniques, data augmentation, and validation procedures likely contributed to the good results observed in the collected dataset after training the YOLOv8s model for 20 epochs and YOLOv8m model for 50 epochs for public dataset.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

A facemask is crucial for choosing the right type and ensuring its effectiveness in its intended use. By prioritizing proper coverage, filtration, fit, comfort, and durability, we can maximize the benefit of facemasks in protecting individuals and communities from the spread of airborne illnesses. Facemask protocol detection has emerged as a significant technology in response to the global pandemic, aiming to curb the spread of infectious diseases. This technology has made advancements in recent years, offering promising solutions for public health and safety.

The goal of this study was to use machine-learning approaches to build a facemask protocol detection, and this research was to develop, test, and evaluate machine-learning methods for predicting facemask wearing, incorrect form wearing, and not wearing a facemask. To carry out the study successfully, it was essential to detect the facemask protocol. Bounding box, passing class balance the dataset, model training using YOLOv8s, YOLOv8m, and YOLOv8x model testing, and finally comparing models using evaluation metrics are also methods used to implement and design for facemask protocol detection. We used collected datasets and downloaded them from Kaggle global datasets for this research. To balance the dataset; we resampled it using a combination of oversampling and under-sampling algorithms. Each balanced severed dataset class is divided into three assigning classes: face mask-wearing, incorrect form wearing, and not wearing a face mask. The experiment was conducted on three dataset types, obtaining three models for each dataset. When compared to YOLOv8s, YOLOv8m, and YOLOv8x, the YOLOv8s model produced greater test accuracy on the local dataset, and in the global dataset, YOLOv8m will be more good accuracy. In the local dataset, Precision Recall mAP is 0.995, F1 Confidence is 0.99, and Precision Confidence is 0.960, in 20 epochs. In the global dataset, Precision Recall mAP is 0.994, F1 Confidence is 0.98, and Precision Confidence is 0.936, in 50 epochs.

7.2 Future Work

This study successfully leverages machine learning for facemask protocol detection, fulfilling its key objectives. Building on these results, we propose the following recommendations as future avenues for researchers. These suggestions can spark further exploration and implementation of prototype machine-learning applications in other domains.

- ❖ In our research project, we utilized four types of datasets: correctly worn masks, chin-only wearers who are not wearing mouth and nose, chin-and-mouth are wearing but not nose wearers, and non-mask wearers. We plan to expand this in future work to include scenarios like masks worn on the forehead or not covering the chin, mouth, and nose. Additionally, we will investigate recommendations for proper mask-wearing forms.
- ❖ We acknowledge the limitations of our current model's performance on individuals wearing black facemasks that wearing black person. We are committed to improving our model's accuracy and inclusivity in future iterations through diverse datasets and bias mitigation techniques.

REFERENCES

Arora, R. (2020). Covid-19 Face Mask Detector. *International Journal of Engineering Applied Sciences and Technology*, *5*(4), 407–409. https://doi.org/10.33564/ijeast.2020.v05i04.064

Ather, B., & Edemekong, P. F. (2019). Airborne Precautions. *StatPearls Publishing*, *01*.

Bell, C. (2021). Introduction to the Internet of Things. *Beginning IoT Projects*, 3–30. https://doi.org/10.1007/978-1-4842-7234-3_1

Bowen, L. E. (2010). *Does That Face Mask Protect You ?*

Chithra, K., & Santhanam, T. (2018). A novel denoising technique for mixed noise removal from grayscale and color images. *Journal of Theoretical and Applied Information Technology*, *96*(3), 626–642.

Computing, S. I., & Technology, C. (2021). *Face Mask Detection Using CNN*. 118–122. https://doi.org/10.3233/APC210022

Das, A., Ansari, M. W., & Basak, R. (2021). *Covid-19 Face Mask Detection Using*. https://doi.org/10.1109/INDICON49873.2020.9342585

Das, A., Wasif Ansari, M., & Basak, R. (2020). Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. *2020 IEEE 17th India Council International Conference, INDICON 2020*. https://doi.org/10.1109/INDICON49873.2020.9342585

Goh, Y., Tan, B. Y. Q., Bhartendu, C., Ong, J. J. Y., & Sharma, V. K. (2020). Brain , Behavior , and Immunity The face mask : How a real protection becomes a psychological symbol during Covid-19 ? *Brain Behavior and Immunity*, *88*(June), 1–5. https://doi.org/10.1016/j.bbi.2020.05.060

Haton, J. P. (2006). A brief introduction to artificial intelligence. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *9*(PART 1), 8–16. https://doi.org/10.3182/20060522-3-fr-2904.00003

Hays, J. (2022). Computer Vision Computer Vision : VisionLab. *Computer Vision*, *May*, 1–31.

Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, *11*(7).

https://doi.org/10.3390/machines11070677

Hussain, S., Yu, Y., Ayoub, M., Khan, A., Rehman, R., Wahid, J. A., & Hou, W. (2021). *applied sciences IoT and Deep Learning Based Approach for Rapid Screening and Face Mask Detection for Infection Spread Control of COVID-19*. *December 2019*.

Ike, J. D., Bayerle, H., Logan, R. A., & Parker, R. M. (2020). Face Masks: Their History and the Values They Communicate. *Journal of Health Communication*, *25*(12), 990–995. https://doi.org/10.1080/10810730.2020.1867257

Jain, K. (2015). Machine Learning Basic Concepts . *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2015/06/machine-learning-basics/

Jiang, X., Gao, T., Zhu, Z., & Zhao, Y. (2021). Real-time face mask detection method based on yolov3. *Electronics (Switzerland)*, *10*(7), 1–17. https://doi.org/10.3390/electronics10070837

Kamiri, J., & Mariga, G. (2021). Research Methods in Machine Learning: A Content Analysis. *International Journal of Computer and Information Technology(2279-0764)*, *10*(2). https://doi.org/10.24203/ijcit.v10i2.79

Kumar, A., Kaur, A., & Kumar, M. (2019). Face detection techniques: a review. *Artificial Intelligence Review*, *52*(2), 927–948. https://doi.org/10.1007/s10462-018-9650-2

Ławrynowicz, A., & Tresp, V. (2014). Introducing machine learning. *Perspectives on Ontology Learning*, *18*(January), 35–50. https://doi.org/10.1007/978-3-030-67626-1_8

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Liang, M., Gao, L., Cheng, C., Zhou, Q., Patrick, J., Heiner, K., & Sun, C. (2020). E ffi cacy of face mask in preventing respiratory virus transmission : A systematic review and meta-analysis. *Travel Medicine and Infectious Disease*, *36*(March), 101751. https://doi.org/10.1016/j.tmaid.2020.101751

Loy, C. C. (2021). Face Detection. *Computer Vision*, 429–434. https://doi.org/10.1007/978-3-030-63416-2_798

Mask, F., & Using, D. (2021). *FACE MASK DETECTION USING AI AND IOT A PROJECT*

REPORT Submitted By BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING INSTITUTE OF ROAD AND TRANSPORT TECHNOLOGY,... June. https://doi.org/10.13140/RG.2.2.23151.15521

Mathew, A., Amudha, P., & Sivakumari, S. (2021). Deep learning techniques: an overview. *Advances in Intelligent Systems and Computing*, *1141*(August 2020), 599–608. https://doi.org/10.1007/978-981-15-3383-9_54

Matuschek, C., Moll, F., Fangerau, H., Fischer, J. C., Zänker, K., Van Griensven, M., Schneider, M., Kindgen-Milles, D., Knoefel, W. T., Lichtenberg, A., Tamaskovics, B., Djiepmo-Njanang, F. J., Budach, W., Corradini, S., Häussinger, D., Feldt, T., Jensen, B., Pelka, R., Orth, K., … Haussmann, J. (2020). The history and value of face masks. In *European Journal of Medical Research* (Vol. 25, Issue 1). https://doi.org/10.1186/s40001-020-00423-4

Pandey, V. K., Gupta, V. K., & Kumar, S. (2022). *Face Mask Detection Using Convolutional Neural Network*. 951–954. https://doi.org/10.1109/icac3n53548.2021.9725689

Patel, K. K., Patel, S. M., & Scholar, P. G. (2016). Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. *International Journal of Engineering Science and Computing*, *6*(5). https://doi.org/10.4010/2016.1482

Saisupriya, N. P., & Rashmi, S.Parthasarathy, G. (2021). Face mask detection using CNN. *Advances in Parallel Computing*, *38*, 118–122. https://doi.org/10.3233/APC210022

Sethi, S., Kathuria, M., & Kaushik, T. (2020). *Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID- 19 . The COVID-19 resource centre is hosted on Elsevier Connect , the company ' s public news and information . January*.

Suresh, K., Mb, P., & Bhuvan, S. (2021). *Face Mask Detection by using Optimistic Convolutional Neural Network*. 1084–1089. https://doi.org/10.1109/ICICT50816.2021.9358653

Suresh, K., Palangappa, M. B., & Bhuvan, S. (2021). Face Mask Detection by using Optimistic Convolutional Neural Network. *Proceedings of the 6th International Conference on*

Inventive Computation Technologies, ICICT 2021, 1084–1089. https://doi.org/10.1109/ICICT50816.2021.9358653

Tegegne, T., Balcha, H. B., & Beyene, M. (2019). Internet of Things Technology for Agriculture in Ethiopia: A Review. In *Communications in Computer and Information Science* (Vol. 1026, Issue May 2020). Springer International Publishing. https://doi.org/10.1007/978-3-030-26630-1_20

Tian, Y. (2020). Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm. *IEEE Access*, *8*, 125731–125744. https://doi.org/10.1109/ACCESS.2020.3006097

Todt, E., & Krinski, B. A. (2019). *Introduction CNN Layers CNN Models Popular Frameworks Papers References Convolutional Neural Network-CNN*. 1–68.

Venkateswarlu, I. B. (2021). *Face mask detection using MobileNet and Global Pooling Block*. 0–4.

Wiersinga, W. J., & Prescott, H. C. (2020). What Is COVID-19? *JAMA - Journal of the American Medical Association*, *324*(8), 816. https://doi.org/10.1001/jama.2020.12984

Yegnanarayana, B. (1994). Artificial neural networks for pattern recognition. *Sadhana*, *19*(2). https://doi.org/10.1007/BF02811896

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object Detection with Deep Learning: A Review. In *IEEE Transactions on Neural Networks and Learning Systems* (Vol. 30, Issue 11). https://doi.org/10.1109/TNNLS.2018.2876865

Arora, R. (2020). Covid-19 Face Mask Detector. *International Journal of Engineering Applied Sciences and Technology*, *5*(4), 407–409. https://doi.org/10.33564/ijeast.2020.v05i04.064

Ather, B., & Edemekong, P. F. (2019). Airborne Precautions. *StatPearls Publishing*, *01*.

Bell, C. (2021). Introduction to the Internet of Things. *Beginning IoT Projects*, 3–30. https://doi.org/10.1007/978-1-4842-7234-3_1

Bowen, L. E. (2010). *Does That Face Mask Really Protect You ?*

Chithra, K., & Santhanam, T. (2018). A novel denoising technique for mixed noise removal from grayscale and color images. *Journal of Theoretical and Applied Information Technology*, *96*(3), 626–642.

Computing, S. I., & Technology, C. (2021). *Face Mask Detection Using CNN*. 118–122. https://doi.org/10.3233/APC210022

Das, A., Ansari, M. W., & Basak, R. (2021). *Covid-19 Face Mask Detection Using*. https://doi.org/10.1109/INDICON49873.2020.9342585

Das, A., Wasif Ansari, M., & Basak, R. (2020). Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. *2020 IEEE 17th India Council International Conference, INDICON 2020*. https://doi.org/10.1109/INDICON49873.2020.9342585

Goh, Y., Tan, B. Y. Q., Bhartendu, C., Ong, J. J. Y., & Sharma, V. K. (2020). Brain , Behavior , and Immunity The face mask : How a real protection becomes a psychological symbol during Covid-19 ? *Brain Behavior and Immunity*, *88*(June), 1–5. https://doi.org/10.1016/j.bbi.2020.05.060

Haton, J. P. (2006). A brief introduction to artificial intelligence. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *9*(PART 1), 8–16. https://doi.org/10.3182/20060522-3-fr-2904.00003

Hays, J. (2022). Computer Vision Computer Vision : VisionLab. *Computer Vision*, *May*, 1–31.

Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, *11*(7). https://doi.org/10.3390/machines11070677

Hussain, S., Yu, Y., Ayoub, M., Khan, A., Rehman, R., Wahid, J. A., & Hou, W. (2021). *applied sciences IoT and Deep Learning Based Approach for Rapid Screening and Face Mask Detection for Infection Spread Control of COVID-19*. *December 2019*.

Ike, J. D., Bayerle, H., Logan, R. A., & Parker, R. M. (2020). Face Masks: Their History and the Values They Communicate. *Journal of Health Communication*, *25*(12), 990–995. https://doi.org/10.1080/10810730.2020.1867257

Jain, K. (2015). Machine Learning Basic Concepts . *Analytics Vidhya*.

https://www.analyticsvidhya.com/blog/2015/06/machine-learning-basics/

Jiang, X., Gao, T., Zhu, Z., & Zhao, Y. (2021). Real-time face mask detection method based on yolov3. *Electronics (Switzerland)*, *10*(7), 1–17. https://doi.org/10.3390/electronics10070837

Kamiri, J., & Mariga, G. (2021). Research Methods in Machine Learning: A Content Analysis. *International Journal of Computer and Information Technology(2279-0764)*, *10*(2). https://doi.org/10.24203/ijcit.v10i2.79

Kumar, A., Kaur, A., & Kumar, M. (2019). Face detection techniques: a review. *Artificial Intelligence Review*, *52*(2), 927–948. https://doi.org/10.1007/s10462-018-9650-2

Ławrynowicz, A., & Tresp, V. (2014). Introducing machine learning. *Perspectives on Ontology Learning*, *18*(January), 35–50. https://doi.org/10.1007/978-3-030-67626-1_8

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Liang, M., Gao, L., Cheng, C., Zhou, Q., Patrick, J., Heiner, K., & Sun, C. (2020). E ffi cacy of face mask in preventing respiratory virus transmission : A systematic review and meta-analysis. *Travel Medicine and Infectious Disease*, *36*(March), 101751. https://doi.org/10.1016/j.tmaid.2020.101751

Loy, C. C. (2021). Face Detection. *Computer Vision*, 429–434. https://doi.org/10.1007/978-3-030-63416-2_798

Mask, F., & Using, D. (2021). *FACE MASK DETECTION USING AI AND IOT A PROJECT REPORT Submitted By BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING INSTITUTE OF ROAD AND TRANSPORT TECHNOLOGY,... June*. https://doi.org/10.13140/RG.2.2.23151.15521

Mathew, A., Amudha, P., & Sivakumari, S. (2021). Deep learning techniques: an overview. *Advances in Intelligent Systems and Computing*, *1141*(August 2020), 599–608. https://doi.org/10.1007/978-981-15-3383-9_54

Matuschek, C., Moll, F., Fangerau, H., Fischer, J. C., Zänker, K., Van Griensven, M., Schneider, M., Kindgen-Milles, D., Knoefel, W. T., Lichtenberg, A., Tamaskovics, B., Djiepmo-Njanang, F. J., Budach, W., Corradini, S., Häussinger, D., Feldt, T., Jensen, B., Pelka, R., Orth, K., … Haussmann, J. (2020). The history and value of face masks. In *European Journal of Medical Research* (Vol. 25, Issue 1). https://doi.org/10.1186/s40001-020-00423-4

Pandey, V. K., Gupta, V. K., & Kumar, S. (2022). *Face Mask Detection Using Convolutional Neural Network*. 951–954. https://doi.org/10.1109/icac3n53548.2021.9725689

Patel, K. K., Patel, S. M., & Scholar, P. G. (2016). Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. *International Journal of Engineering Science and Computing*, *6*(5). https://doi.org/10.4010/2016.1482

Saisupriya, N. P., & Rashmi, S.Parthasarathy, G. (2021). Face mask detection using CNN. *Advances in Parallel Computing*, *38*, 118–122. https://doi.org/10.3233/APC210022

Sethi, S., Kathuria, M., & Kaushik, T. (2020). *Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID- 19 . The COVID-19 resource centre is hosted on Elsevier Connect , the company ' s public news and information . January*.

Suresh, K., Mb, P., & Bhuvan, S. (2021). *Face Mask Detection by using Optimistic Convolutional Neural Network*. 1084–1089. https://doi.org/10.1109/ICICT50816.2021.9358653

Suresh, K., Palangappa, M. B., & Bhuvan, S. (2021). Face Mask Detection by using Optimistic Convolutional Neural Network. *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, 1084–1089. https://doi.org/10.1109/ICICT50816.2021.9358653

Tegegne, T., Balcha, H. B., & Beyene, M. (2019). Internet of Things Technology for Agriculture in Ethiopia: A Review. In *Communications in Computer and Information Science* (Vol. 1026, Issue May 2020). Springer International Publishing. https://doi.org/10.1007/978-3-030-26630-1_20

Tian, Y. (2020). Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm. *IEEE Access*, *8*, 125731–125744. https://doi.org/10.1109/ACCESS.2020.3006097

Todt, E., & Krinski, B. A. (2019). *Introduction CNN Layers CNN Models Popular Frameworks Papers References Convolutional Neural Network-CNN*. 1–68.

Venkateswarlu, I. B. (2021). *Face mask detection using MobileNet and Global Pooling Block*. 0–4.

Wiersinga, W. J., & Prescott, H. C. (2020). What Is COVID-19? *JAMA - Journal of the American Medical Association*, *324*(8), 816. https://doi.org/10.1001/jama.2020.12984

Yegnanarayana, B. (1994). Artificial neural networks for pattern recognition. *Sadhana*, *19*(2). https://doi.org/10.1007/BF02811896

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object Detection with Deep Learning: A Review. In *IEEE Transactions on Neural Networks and Learning Systems* (Vol. 30, Issue 11). https://doi.org/10.1109/TNNLS.2018.2876865

APPENDIX

Appendix A: Sample Code

```
[2] import numpy as np
    import pandas as pd
    from sklearn.metrics import accuracy_score
```

```
[3] import os
    HOME = os.getcwd()
    print(HOME)
```

```
/content
```

```
import torch
print(torch.version.cuda)
```

```
12.1
```

```
[5] # Pip install method (recommended)

    !pip install ultralytics==8.0.20

    from IPython import display
    display.clear_output()

    import ultralytics
    ultralytics.checks()
```

```
Ultralytics YOLOv8.0.20 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 26.3/78.2 GB disk)
```

```
[6] from ultralytics import YOLO

    from IPython.display import display, Image
```

```
[8] from google.colab import drive
    drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[9] %cd /content/drive/MyDrive/data
```

```
/content/drive/MyDrive/data
```

```
!ls
```

```
data.yaml  runs  test  test.yaml  train  Valid  yolov8m.pt  yolov8s.pt  yolov8x.pt
```

```
[ ] %cd /content/drive/MyDrive/data
```

Appendix B: Training Sample

```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:04<00:00,  4.90s/it]
                   all         24         24     0.0616      0.139     0.0566     0.0152
      Face Mask Wearing         24          6          0          0     0.0367     0.0148
Incorrect Form Wearing Face Mask         24         12      0.154       0.25     0.0983     0.0179
    Not Wearing Face Mask         24          6     0.0309      0.167     0.0347     0.0129

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       2/50      0.69G      1.972      3.678      1.788         24        224: 100% 3/3 [00:06<00:00,  2.03s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:03<00:00,  3.68s/it]
                   all         24         24     0.0613      0.222     0.0491      0.014
      Face Mask Wearing         24          6     0.0261      0.167     0.0336     0.0143
Incorrect Form Wearing Face Mask         24         12      0.139      0.333      0.084     0.0152
    Not Wearing Face Mask         24          6     0.0191      0.167     0.0298     0.0126

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       3/50     0.843G      2.065      3.598      1.821         32        224: 100% 3/3 [00:06<00:00,  2.32s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:03<00:00,  3.10s/it]
                   all         24         24     0.0562      0.111     0.0561     0.0192
      Face Mask Wearing         24          6          0          0     0.0362     0.0175
Incorrect Form Wearing Face Mask         24         12       0.13      0.167     0.0886      0.017
    Not Wearing Face Mask         24          6     0.0387      0.167     0.0435     0.0231

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       4/50     0.843G      1.898      3.196      1.824         31        224: 100% 3/3 [00:08<00:00,  2.79s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:03<00:00,  3.15s/it]
                   all         24         24     0.0664      0.139      0.093     0.0421
      Face Mask Wearing         24          6          0          0     0.0742     0.0376
Incorrect Form Wearing Face Mask         24         12      0.118       0.25     0.0969     0.0236
    Not Wearing Face Mask         24          6     0.0813      0.167      0.108     0.0651

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       5/50     0.843G      1.614      2.295      1.692         28        224: 100% 3/3 [00:08<00:00,  2.77s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:03<00:00,  3.08s/it]
                   all         24         24       0.62      0.222      0.224      0.118
      Face Mask Wearing         24          6          1          0       0.26       0.13
Incorrect Form Wearing Face Mask         24         12      0.135        0.5      0.152     0.0672
    Not Wearing Face Mask         24          6      0.725      0.167      0.259      0.157

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       6/50     0.843G      1.427      2.098      1.589         30        224: 100% 3/3 [00:07<00:00,  2.37s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:04<00:00,  4.08s/it]
                   all         24         24      0.469      0.444      0.238      0.124
      Face Mask Wearing         24          6          1          0      0.316      0.199
Incorrect Form Wearing Face Mask         24         12      0.193      0.498      0.157      0.079
    Not Wearing Face Mask         24          6      0.213      0.833      0.241     0.0954

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       7/50     0.849G      1.482      2.006      1.671         30        224: 100% 3/3 [00:06<00:00,  2.14s/it]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1 [00:04<00:00,  4.67s/it]
                   all         24         24      0.461      0.482      0.372      0.228
      Face Mask Wearing         24          6          1      0.314      0.505      0.312
Incorrect Form Wearing Face Mask         24         12      0.164      0.333      0.156      0.074
```

Appendix C: The Best Performance Image Result in the Collected Dataset

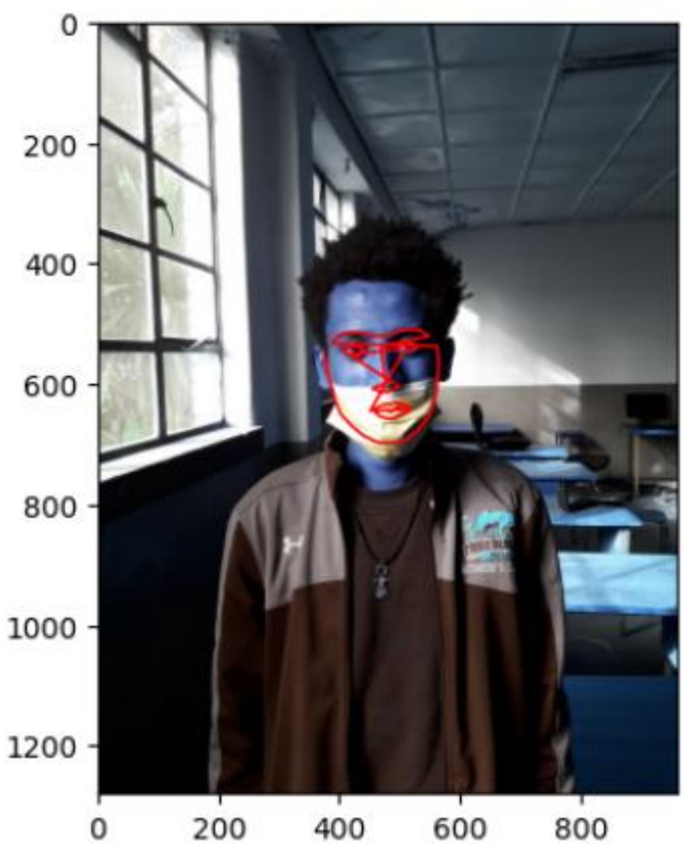
In 20 epoch YOLOv8s Model

Appendix D: The Best Performance Image Result in the Public Dataset

In 50 epoch YOLOv8m Model

Appendix E: On Research Question one Experiment result

Appendix F: Face Landmark Experiment Result



Appendix G: Text based detection results

```
image 1/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000259_1_000259_FEMALE_23.jpg: 224x192 1 Face Mask Wearing, 59.7ms
image 2/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000259_2_000259_FEMALE_23.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 10.8ms
image 3/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000259_3_000259_FEMALE_23.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.6ms
image 4/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000259_4_000259_FEMALE_23.jpg: 224x192 1 Not Wearing Face Mask, 7.5ms
image 5/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000299_1_000299_MALE_22.jpg: 224x128 1 Face Mask Wearing, 58.1ms
image 6/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000299_2_000299_MALE_22.jpg: 224x128 1 Face Mask Wearing, 1 Incorrect Form Wearing Face Mask, 6.9ms
image 7/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000299_3_000299_MALE_22.jpg: 224x128 1 Incorrect Form Wearing Face Mask, 7.0ms
image 8/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000299_4_000299_MALE_22.jpg: 224x128 1 Not Wearing Face Mask, 6.9ms
image 9/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000352_1_000352_NONE_25.jpg: 224x192 1 Face Mask Wearing, 8.1ms
image 10/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000352_2_000352_NONE_25.jpg: 224x192 1 Face Mask Wearing, 1 Not Wearing Face Mask, 7.4ms
image 11/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000352_3_000352_NONE_25.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 1 Not Wearing Face Mask, 7.5ms
image 12/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000352_4_000352_NONE_25.jpg: 224x192 1 Not Wearing Face Mask, 7.4ms
image 13/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000388_1_000388_MALE_21.jpg: 224x192 1 Face Mask Wearing, 7.5ms
image 14/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000388_2_000388_MALE_21.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.5ms
image 15/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000388_3_000388_MALE_21.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.3ms
image 16/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000388_4_000388_MALE_21.jpg: 224x192 1 Not Wearing Face Mask, 7.7ms
image 17/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000393_1_000393_MALE_18.jpg: 224x192 1 Face Mask Wearing, 1 Incorrect Form Wearing Face Mask, 7.5ms
image 18/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000393_2_000393_MALE_18.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.4ms
image 19/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000393_3_000393_MALE_18.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 8.6ms
image 20/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000393_4_000393_MALE_18.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.5ms
image 21/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000396_1_000396_FEMALE_22.jpg: 224x192 1 Face Mask Wearing, 7.7ms
image 22/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000396_2_000396_FEMALE_22.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.6ms
image 23/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000396_3_000396_FEMALE_22.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 8.0ms
image 24/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000396_4_000396_FEMALE_22.jpg: 224x192 1 Not Wearing Face Mask, 7.5ms
image 25/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000399_1_000399_MALE_21.jpg: 224x192 1 Face Mask Wearing, 7.6ms
image 26/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000399_2_000399_MALE_21.jpg: 224x192 1 Face Mask Wearing, 1 Incorrect Form Wearing Face Mask, 7.6ms
image 27/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000399_3_000399_MALE_21.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 1 Not Wearing Face Mask, 7.5ms
image 28/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000399_4_000399_MALE_21.jpg: 224x192 1 Not Wearing Face Mask, 7.5ms
image 29/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000401_1_000401_MALE_26.jpg: 224x192 7.4ms
image 30/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000401_2_000401_MALE_26.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.5ms
image 31/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000401_3_000401_MALE_26.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.3ms
image 32/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000401_4_000401_MALE_26.jpg: 224x192 1 Not Wearing Face Mask, 7.4ms
image 33/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000409_1_000409_NONE_19.jpg: 224x192 1 Face Mask Wearing, 7.5ms
image 34/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000409_2_000409_NONE_19.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 9.4ms
image 35/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000409_3_000409_NONE_19.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.5ms
image 36/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/000409_4_000409_NONE_19.jpg: 224x192 1 Not Wearing Face Mask, 7.4ms
image 37/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/001305_1_001305_MALE_44.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.4ms
image 38/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/001305_2_001305_MALE_44.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.6ms
image 39/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/001305_3_001305_MALE_44.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 1 Not Wearing Face Mask, 7.5ms
image 40/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/001305_4_001305_MALE_44.jpg: 224x192 1 Not Wearing Face Mask, 7.4ms
image 41/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/20220629_163018.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.2ms
image 42/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/20220629_163038.jpg: 224x192 1 Incorrect Form Wearing Face Mask, 7.2ms
image 43/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/20220629_163054.jpg: 224x192 1 Not Wearing Face Mask, 7.2ms
image 44/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/20220629_163116.jpg: 224x192 1 Not Wearing Face Mask, 7.3ms
image 45/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/IMG_20221015_144326_665.jpg: 224x128 1 Not Wearing Face Mask, 7.2ms
image 46/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/IMG_20221015_144328_372.jpg: 224x128 1 Incorrect Form Wearing Face Mask, 6.7ms
image 47/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/IMG_20221015_144330_635.jpg: 224x128 1 Incorrect Form Wearing Face Mask, 6.7ms
image 48/48 /content/drive/MyDrive/yolov8_dataset/data/test/images/IMG_20221015_144333_131.jpg: 224x128 1 Incorrect Form Wearing Face Mask, 1 Not Wearing Face Mask, 6.7ms
```