MASKED FACE DETECTION

A PROJECT REPORT

submitted by

ABHINAV U (WYD18CS005)
ARCHANA UNNI (WYD18CS011)
SAMAYA SHAJI (WYD18CS042)
WASEEM ASHRAF (WYD18CS059)

to

the APJ Abdul Kalam Technological University
in partial fulfilment of the requirements for the award of the Degree

of

Bachelor of Technology
in
Computer Science and Engineering

Department of Computer Science and Engineering

Government Engineering College, Wayanad
Thalappuzha – 670644

JUNE, 2022

DECLARATION

I, on behalf of authors of the report: Abhinav U , Archana Unni, Samaya Shaji, Waseem Ashraf, hereby declare that the project report " Masked Face Detection " submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University , Kerala is a bonafide work done by us under supervision of Prof. Riji R. This submission represents our ideas in our own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also invoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Thalappuzha
Date: 30-05-2022

Abhinav U
Archana Unni
Samaya Shaji
Waseem Ashraf

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOVERNMENT ENGINEERING COLLEGE, WAYANAD
THALAPPUZHA – 670644

CERTIFICATE

This is to certify that the report entitled **‘ Masked Face Detection ’** submitted by **Abhinav U (WYD18CS005), Archana Unni (WYD18CS011) , Samaya Shaji (WYD18CS015), Waseem Ashraf (WYD18CS059)** to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor Technology in Computer Science and Engineering is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor(s)

Department Seal

Project Coordinator(s) Head of the Department

ACKNOWLEDGEMENTS

We owe our heartfelt gratitude to Almighty for all the blessings showered on us during this thesis work. We are extremely grateful to the Principal Dr. Anitha V S, Government Engineering College Wayanad, for providing the necessary facilities. We take this opportunity to express our sincere thanks to Associate Professor Dr.Gilesh P, Head of the Computer Science and Engineering Department, for his valuable suggestions. We also thank the project coordinators Prof. Shabeer K P, Assistant Professor,Department of Computer Science and Engineering, Prof. Binatha C, Assistant Professor, Department of Computer Science and Engineering, for their guidance throughout this project. Their positive approach had offered incessant help in all possible ways from the beginning. We take immense pleasure in expressing our humble note of gratitude to our project guide Prof. Riji R, Assistant Professor, Department of Computer Science and Engineering, for her remarkable guidance in the course of completion of the project. We also extend our thanks to faculty members and our friends for their moral support for completing the project. Our acknowledgment would not be complete without acknowledging our gratitude to our beloved parents who have been the pillars of support and constant encouragement throughout this project.

ABSTRACT

Face recognition is one of the most common biometric authentication methods as its feasibility while convenient use. Recently, the COVID-19 pandemic is dramatically spreading throughout the world, which seriously leads to negative impacts on people's health and economy. Wearing masks in public settings is an effective way to prevent viruses from spreading. However, masked face recognition is a highly challenging task due to the lack of facial feature information.So there is a need to develop a facial recognition software that can recognize faces even if the person is wearing a face mask. This project aims at developing a system that can overcome this problems by using FaceNet which is a face recognition pipeline that can learn to map from faces to a position in a multidimensional space where the distance between points can directly correspond to a measure of face similarity.We have trained the model on CASIA Webface dataset and used Labelled Faces in the Wild(lfw) dataset to test the model. The model has obtained an accuracy of 96.4 +- 0.61 % and an Area Under the Curve(AUC) 0f 0.994. We have employed a variety of methods for preparing the dataset which are discussed in detail later in this report.

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

COVID-19 has brought a drastic shift in pace of life of every person on this planet.We have lost countless people due the pandemic that shows no signs of ending anytime soon. This had lead to necessity to adapt our life to live with covid. COVID-19 is known be highly infectious and quickly spread through societies through air by direct human-to-human contact. Healthcare specialists have come with a variety of methods to fight against this pandemic such as social distancing, taking vaccines , encourage contactless services and ofcourse wear masks. Wearing masks in public spaces is necessary to prevent spread of covid-19. Also wearing masks is one of the lowest cost effective way to safeguard ourselves. This however will drastically affect the performance of face recognition systems which has now become an integral part of our daily life from unlocking the phone to accessing public transportation system in some developed counties, face recognition systems are re-changing every aspect of life day by day. But as masks cover a large portion of the face including key facial landmark features like lips and nose which results in not being able to extract a large number of features.Therefor it is necessary to upgrade existing face recognition systems so that it work even if the person is wearing a face mask.

1.1 CURRENT SYSTEM

The current system works by extracting a number of facial features from key facial landmarks like lips and nose. Face masks cover a large portion of these landmarks thereby reducing efficiency of existing systems.

1.2 PROPOSED SYSTEM

Our system attempts to overcome the limitations introduced by facemasks to existing face recognition systems. Our proposed methods consist of mainly three stages face recognition, face embedding and face recognition using inception resnet V1.

Before training the model we also need to prepare the dataset. We create a new dataset of CASIA Webface in which we place masks on people faces as the original CASIA Webface dataset did not contain masked images of people that can be used for training.

Once the system has been trained we run the model on smaller systems so that have less computational power.

1.3 FEASIBILITY

In terms of the feasibility study, our project 'Masked Face Recognition' is feasible after training.Once trained it can be run even on a laptop with low GPU.

1.3.1 Technical Feasibility

The proposed system needs libraries like tensorflow, keras, sktlearn,opencv,numpy, dlib, in python and application used for this is Pycharm , and hence it is technically feasible.

1.3.2 Economical & Financial Feasibility

The expense is mainly for the computer , graphical processing unit and RAM for implementing the system and for Internet connection. Otherwise,all the libraries for developing the project is completely free of cost. But the system required to train would cost minimum of 2 lakh Rupees

1.3.3 Schedule Feasibility

In this Project, the time required to train the model is very long .Each team member is assigned to develop different sections in parallel till sequential execution is required. As we started the project early we could complete the project on schedule even with days long training time.

1.3.4 Resource Feasibility

To train the model we need high graphical processing power , so we used a good system with 64 GB of RAM and GPU of RTX 3080Ti.Also a storage space of 500GB of SSD is recommended for staring and working on the dataset. Obtaining the training resource is a great hurdle for our project.

1.4 PROCESS MODEL

We selected the agile model as the process model for our project. This model is a way to manage the project by breaking it up into several phases. It involves quick response to any changes with the help of adaptive planning, joint elaboration of requirements, rationalization of tasks performed by the development team and step by step software development with strict time frames . Once the work begins, teams cycle through a process of planning, executing, and evaluating.

- **Requirement Analysis:** In this phase,the requirements of the users are collected and all these requirements are documented in the software requirements specification(SRS). And Planned the time and effort needed to build the project. Based on this requirements technical and economic feasibility is evaluated.

- **Design Document & prototype:**Good definitions for the requirements are formed in this phase and high level UML diagrams are used to show how it will apply to the project.This design helps to clearly understand hardware and software requirements and helps in defining overall system architecture.

- **Development And Integration:**The development begins after the design phase is completed. Each unit is developed and tested separately for its functionality. The individual units which are tested are integrated in this phase.

- **Testing:** In this phase,After integration the entire system is tested and looks for the bug.

- **Deployment And Feedback:** In this phase,the product is issued for the user's work environment. Maybe some issues may arise from the client-side. We have to fix those issues. Maintenance is done to provide these changes.

CHAPTER 2
REQUIREMENT ANALYSIS

2.1 METHOD OF REQUIREMENT ELICITATION

The main purpose of requirement elicitation is to upgrade existing facial recognition sysytem to recognize faces even with masks. This includes study machine learning models,interaction , online search, and group discussion. The team have gathered at every project hour to bring maximum output.

2.1.1 System Study

Researched various types of machine learning models to find out which can solve our problem with maximum accuracy and can be implemented on any small device.

2.1.2 Online Search

Find the software requirements for the creation of the application. Learnt MTCNN, tensorflow and environment creation from various online courses.Also we sought out help from alumnis and teachers.

2.1.3 Group Discussion

Group discussion with the team members, colleagues and guide gives new idea for improving the proposed model. Did brainstorming to understand the design and implementation of the software.

2.1.4 Testing

We tested the model with faces of team members, guide and friends by entering our faces to the database and testing the model with and without masks.

2.2 USER REQUIREMENTS

- Recognize faces.
- Recognize face with masks.
- Identify if the person has worn a mask or not.

2.3 PROJECT REQUIREMENTS

2.3.1 Software requirements

- Pycharm and Conda is required to develop and train our machine learning model.
- The code written using python programming language.
- Libraries like dlib, tensorflow, keras , opencv, numpy are to downloaded.
- Different conda environments with different tensorflow version are set up.

CHAPTER 3
DESIGN AND IMPLEMENTATION

System design is the earliest stage of the project in which we define the architecture, product design, modules, interfaces, and data for a system to satisfy the requirements specified in the requirement specification phase of the project development. The purpose of this chapter is to is outline the design of the project titled 'Masked Face Detection'.

3.1 ARCHITECTURAL DESCRIPTION

The Architecture comprises of the algorithms and modules used in this project. It gives a view of the entire system highlighting the relevant features and ignoring unnecessary details. The system is an machine learning model that recognize faces even if the user is wearing a mask. The admin can add and maintain database consisting of user's faces.

Our working is pretty straight-forward. After adding an user's face to the database , the system can instantly recognize the person even if the person is wearing a mask. The system can also detect masks without adding an user to the database, this can be also used as face mask detection at public places.

3.1.1 Decomposition Description

The software is to follow a combination of iterative and incremental approach for the development. The process model selected for development is the Agile Based Development. The system is divided into modules based on the functionality of the system.The modules present in the software are:

1. **Dataset** : Dataset Collection is most important module in our system. Without adequate dataset of various people we cannot train the model efficiently. For this we have used CASIA Webface dataset which contain 494,414 face images of 10,575 real identities collected from the web. After aligning and cleaning the dataset, we created masked face dataset of CASIA Webface. Both these dataset are used to train the model. We test the accuracy and efficiency of the model using Labelled Faces in the Wild(LFW). The data set contains more than 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured.

2. **Face alignment using MTCNN** :Face detection is very necessary step for a face recognition system to be a reliable one. In this project we use Multi-task Cascaded Convolutional Networks (MTCNN) which is a strong face detector offering high detection scores. MTCNN integrates both recognition and alignment tasks using multitask learning. MTCNN is mainly based on 3 separate CNN models: P-Net, R-Net and O-Net. But before that we take a image and resize it to different scales in order to build an image pyramid, which is the input of the following three-staged cascaded network.

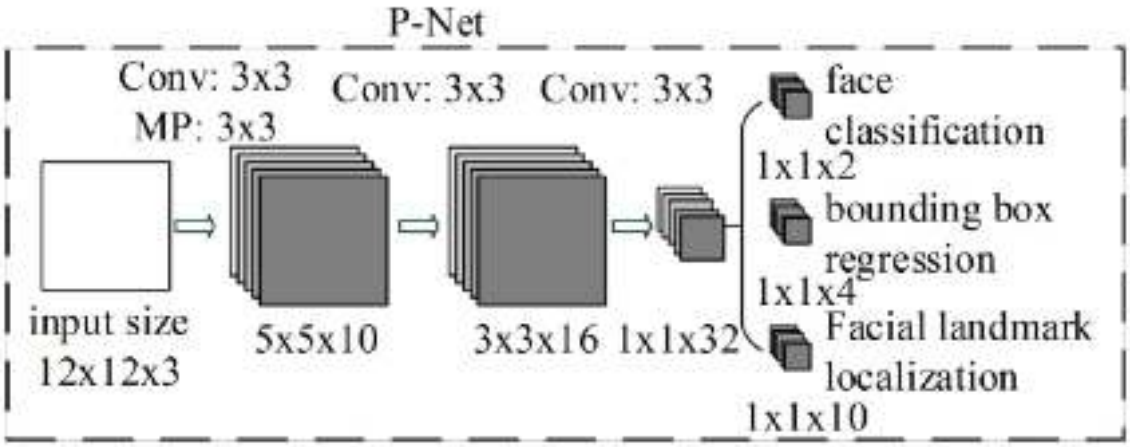

Figure 3.1: Proposal Network

- **The Proposal Network (P-Net)** Short for Proposal Network. P-Net is used to find all candidate windows and their bounding boxes. This stage is fully convolutional network as it does not use a dense layer as a part of its architecture. It's input size is 12 x 12 x 3.

- **The Refine Network (R-Net)** All the candidates obtained from P-Net are fed to R-Net. This network is a CNN, not a FCN like the one before since there is a dense layer at the last stage of the network architecture. The R-Net further reduces the number of candidates, performs calibration with bounding box regression and employs non-maximum suppression (NMS) to merge overlapping candidates.

  The R-Net outputs whether the input is a face or not, a 4 element vector which is the bounding box for the face, and a 5 element vector for facial landmark localization.

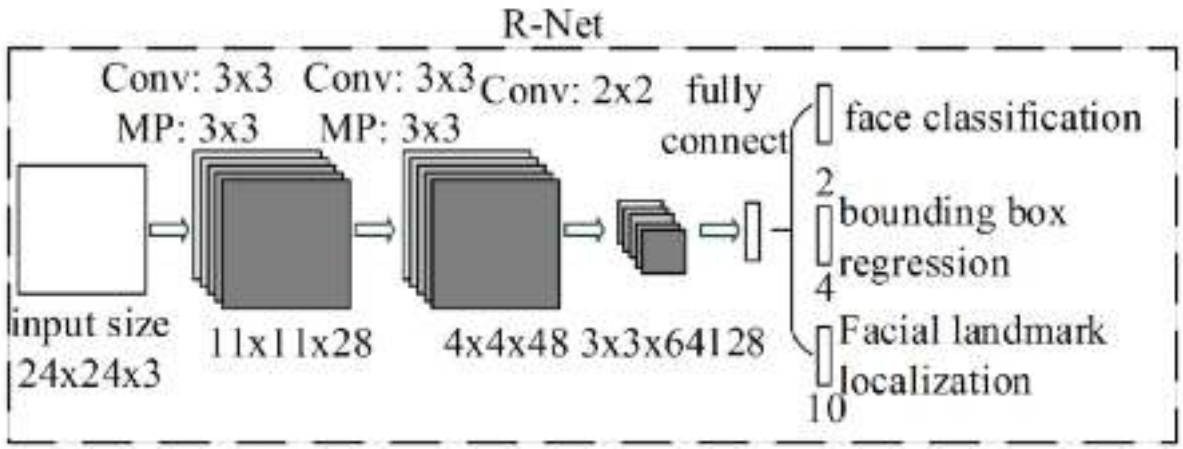

Figure 3.2: The Refine Network

- **The Output Network (O-Net)** This stage is similar to the R-Net, but this Output Network aims to describe the face in more detail and output the five facial landmarks' positions for eyes, nose and mouth. But for our project we only intend to do face alignment using MTCNN, so we discard the 5 element vector for facial facial landmark localization.

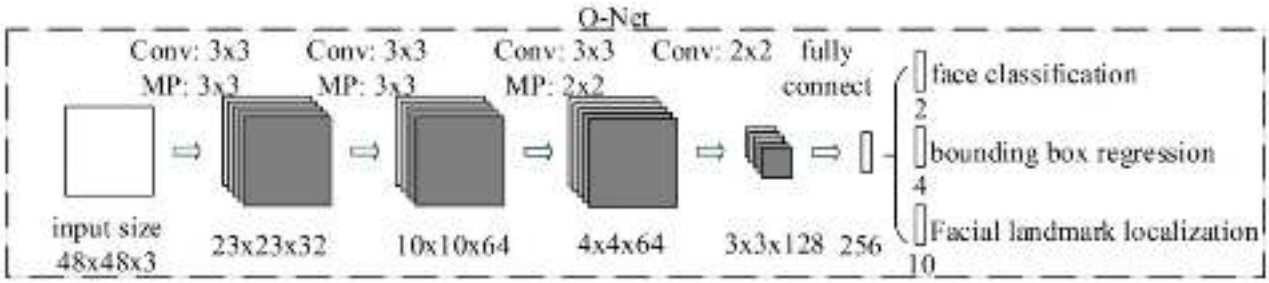

Figure 3.3: The Output Netwotk

So by using MTCNN , we complete 3 tasks :

- Face Classification
- Bounding box regression

- Facial landmark localization

3. **Clean the dataset** : The main aim of Data Cleaning is to identify and remove errors and duplicate data, in order to create a reliable dataset. This improves the quality of the training data for analytics and enables accurate decision-making.

   Data In the CASIA Webface dataset that we are using , there are 10575 classes ie. folders. Even if we check 100 folders a day, it would take us more than 100 days to finish the task.

   - Select images one by one in the same class as the target image.Others are regarded as reference images.
   - Calculate average distance between target image and reference image.
   - Set distance threshold.
   - Remove images whose average distance surpass the threshold.

4. **Creating a Masked dataset** We are doing this to train out model to recognize faces with face mask.

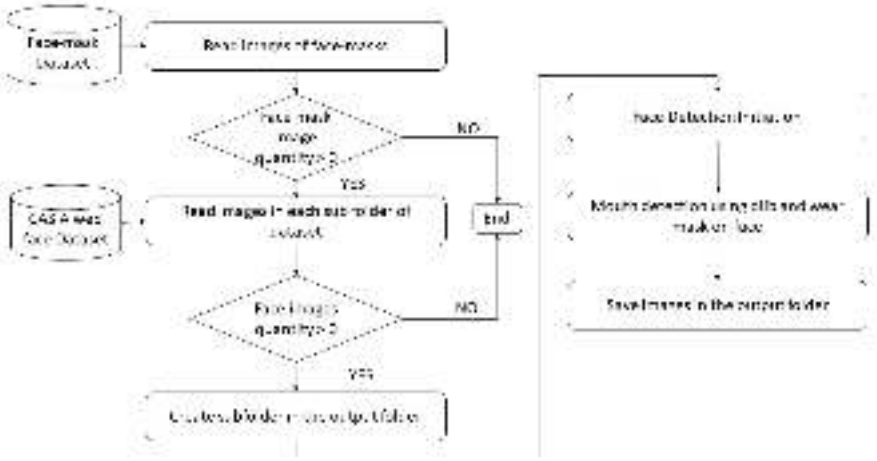

Figure 3.4: Overview of wearing mask

- **Prepare the mask image** The mask image should be a PNG image because PNG image has 4 channels, the additional fourth channel is used to describe the transparency.
- **Using random masks** We use random varieties of face mask images as people wear different types of masks. This makes sure that our model is able to recognize different types of masks.
- **Detect mouth and place masks**
  - Use Dlib face detection library to find the face
  - Use Dlib_68_face_landmarks to find the mouth part.
  - We would then take the mouth part as our ROI(Region Of Interest)
  - The face mask must be resized to same as our ROI.
  - We then place our randomly selected face mask on the ROI

Figure 3.5: Overview of wearing mask

3.1.2 Dependency Description

Data Flow Diagram is a graphical representation of the "flow" of data through an information system, modeling its process aspects. In this Data Flow Diagram, we can see how the user display model works. Flow diagrams, in general, are usually designed using simple symbols. It is a way of representing a flow of data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Here, we have 2-levels to represent the data flow.

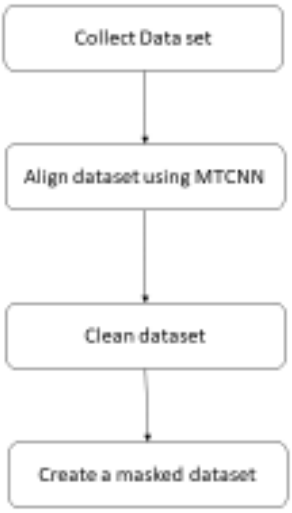

Figure 3.6: An overview training process

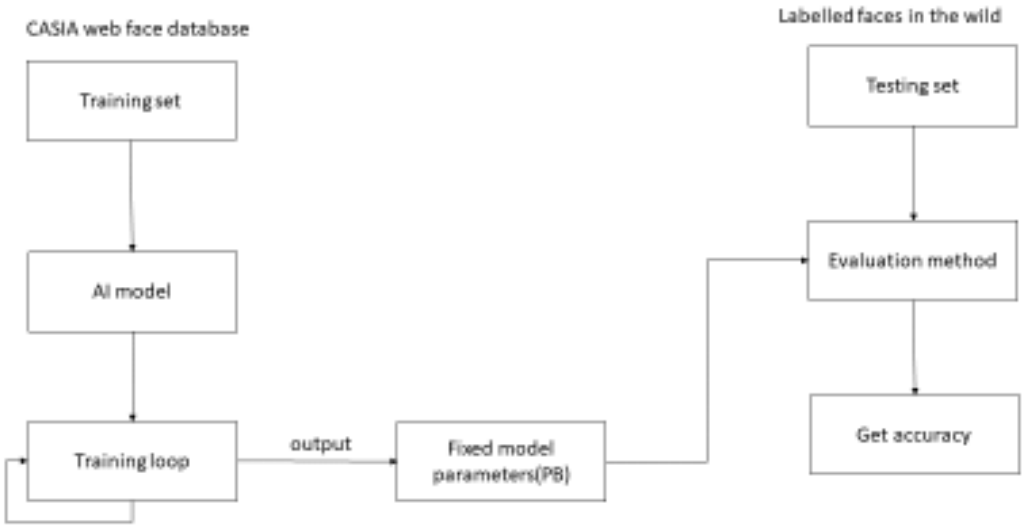

Figure 3.7: An overview training process

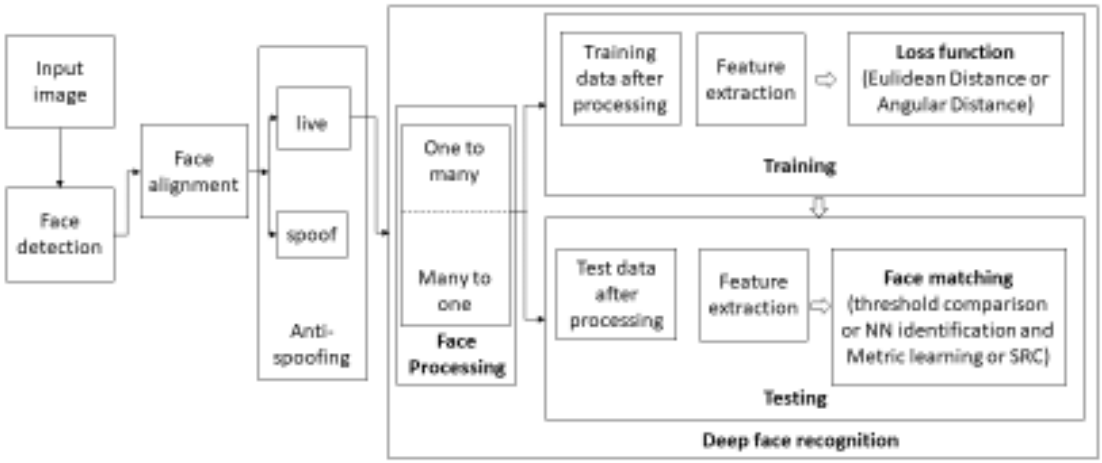

Figure 3.8: Data Flow Diagram Level-1

0-level is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data.

In 1-level DFD, a context diagram is composed of process. In this, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into the process. Fig. 3.5 Illustrates how we create a masked dataset for training including random selection of masks and detection of mouth

It also checks if a given folder is empty and terminate the process.

We can also see that the masked images are saved in different folder fro efficient training.

3.1.3 User Interface

The user interface is a window that live streams feed from the camera module on to the screen. User does not have to do anything to recognize faces as the system automatically detects and recognize faces.

Figure 3.9: Without Mask recognition

Figure 3.10: Masked Face recognition

If a user is not wearing a mask , then the bounding box around their face is in RED color. This also works even if there are multiple persons in the frame. An admin can easily identify person with no masks from a remote monitor at ease.

Whereas if a user is wearing a mask, the bounding box used to identify him will be in GREEN color .By using this color system the admin can easily get the count of masked and non-masked person in a given area.

If the user is already registered in the database by the admin, their names will be shown along side their face.

There are two functional keys in the system. If you press the letter 's' the system allows you capture an user's real time face image. The system then prompts to enter the captured

user's name. Once the name is entered , the system updates this data to the user database. So the next time when the same user is present in fromt of the camera, the system will identify them by their name along with their mask.

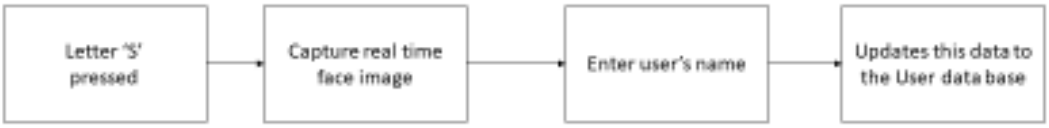

Figure 3.11: Enter new user

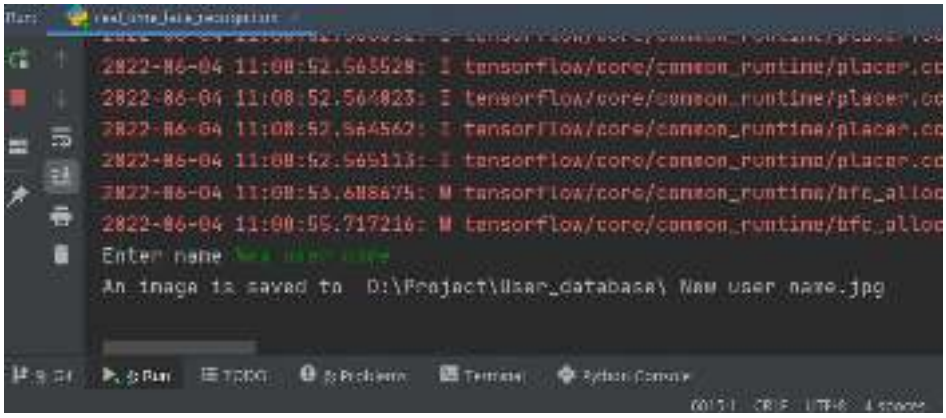

Figure 3.12: Adding new user

Figure 3.13: Command to close system

3.1.4 Requirement Based Testing

Figure 3.14: Masked recognition

Figure 3.15: Face recognition

As the figure clearly illustrates we can see that system works perfectly and correctly in recognizing an user with and without mask. The sytem also identifies if a person is wearing a mask or not. Thus we can see that the system satisfies all the requirements.

CHAPTER 4
CODING

4.1 INTRODUCTION

The development of the system is done using Pycharm IDE and Python programming language.

4.2 PYCHAM AND PYTHON

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. PyCharm is available in three editions:

- Community (free and open-sourced)
- Professional (paid)
- Edu (free and open-sourced)

Features

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages

- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for scientific tools like Matplotlib, NumPy and SciPy

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum. Python has a reputation as a beginner-friendly language, replacing Java as the most widely used introductory language because it handles much of the complexity for the user. Python is great for backend web development, data analysis, artificial intelligence, and scientific computing. Python Syntax similar to the English language, with a mathematical influence, Python is built for readability. Google TensorFlow, as well as other libraries (scikit-learn, Keras), establish a foundation for AI development because of the usability and flexibility it offers Python users. These libraries, and their availability, are critical because they enable developers to focus on growth and building. These are the major resons why we choose to build our project in Pycharm using python programming language.

4.3 DIRECTORY STRUCTURE OF THE PROJECT

- **Create dataset with face mask :** This folder contains the program to align images from the CASIA-Webface dataset so that we get a more useful dataset to train our model. Alignment is done using Multi Tasked Cascaded Convolution Network. The folder also contain tranparent png images of face masks in its 'mask_img' folder
- **mask_wear_classpy :** This python file contains the code to detect and wear face

masks. We Dlib library in python download by 'pip install

- **Libraries :** Our project code is composed of numerous libraries and packages. Some packages are pre-installed with Pycharm while others are downloaded during development using pip or conda.

- **tools.py :** Is a python in which coded many useful functionalities like model_restore_ from_pb(), img_removal_by_embed(), delete_dir_with_no_img() for restoring model, removing mislabelled images and deleting empty libraries.

- **facenet_train.py :** This is the main training module of our project. Weights required for our model is obtained after running this module.

- **pb_model_select_num==15.pb :** This is the trained model weights that we are using for our real time face recognition. This model is well trained and take a bit longer time to execute.

- **User database** This folder contains images of faces of colleagues and teachers that we have collected through the program and verified that our model worked well.

4.4 WORKING OF THE DEPLOYMENT:

We deploy the project through the python file real_time_face_recognition.py. This program imports two function modules from face_alignment.py and tools.py called FaceMaskDetection and model_restore_from_pb.

4.4.1 FaceMaskDetection

```
def __init__(self,pb_path,margin=44,GPU_ratio=0.1):
```

Figure 4.1: User Database

```
#  var
node_dict = {'input': 'data_1:0',
             'detection_bboxes': 'loc_branch_concat_1/concat:0',
             'detection_scores': 'cls_branch_concat_1/concat:0'}
conf_thresh = 0.8
iou_thresh = 0.7


#  anchors config
feature_map_sizes = [[33, 33], [17, 17], [9, 9], [5, 5], [3, 3]]
anchor_sizes = [[0.04, 0.056], [0.08, 0.11], [0.16, 0.22],
[0.32, 0.45],[0.64, 0.72]]
```

```
anchor_ratios = [[1, 0.62, 0.42]] * 5

id2class = {0: 'Mask', 1: 'NoMask'}


anchors = self.generate_anchors(feature_map_sizes, anchor_sizes,

anchor_ratios)

#  model initature_map_sizes, anchor_sizes, anchor_ratios)

anchors_exp = np.expand_dims(anchors, axis=0)

# model restore from pb file

sess, tf_dict = model_restore_from_pb(pb_path, node_dict,

GPU_ratio = GPU_ratio)

tf_input = tf_dict['input']

model_shape = tf_input.shape  # [N,H,W,C]

print("model_shape = ", model_shape)

img_size = (tf_input.shape[2].value,tf_input.shape[1].value)

detection_bboxes = tf_dict['detection_bboxes']

detection_scores = tf_dict['detection_scores']
```

- model_restore_from_pb() restores the model weights so that we can easily recognize face shapes and mask position. Also this model was trained with masks images so the system can recognize different types of face masks.

- self.generate_anchors() : These are anchors that can define embeddings in our case to relative to a reference point.

4.4.2 model_restore_from_pb

- restores the model weights so that we can easily recognize face shapes and mask position. Also this model was trained with masks images so the system can recognize different types of face masks.

```
def model_restore_from_pb(pb_path,node_dict,GPU_ratio=None):
    tf_dict = dict()
    with tf.Graph().as_default():
        config = tf.ConfigProto(log_device_placement=True,
                                allow_soft_placement=True,
                                )
     if GPU_ratio is None:
         config.gpu_options.allow_growth = True
     else:
         config.gpu_options.per_process_gpu_memory_fraction = GPU_ratio
     sess = tf.Session(config=config)
     with gfile.FastGFile(pb_path, 'rb') as f:
            graph_def = tf.GraphDef()
            graph_def.ParseFromString(f.read())
            sess.graph.as_default()

            #----issue solution if models with batch norm
```

```
    for node in graph_def.node:
        if node.op == 'RefSwitch':
            node.op = 'Switch'
            for index in range(len(node.input)):
                if 'moving_' in node.input[index]:
                    node.input[index] = node.input[index] + '/read'
        elif node.op == 'AssignSub':
            node.op = 'Sub'
            if 'use_locking' in node.attr: del node.attr
            ['use_locking']
    tf.import_graph_def(graph_def, name='')


sess.run(tf.global_variables_initializer())
for key,value in node_dict.items():
    try:
        node = sess.graph.get_tensor_by_name(value)
        tf_dict[key] = node
    except:
        print("node:{} does not exist in the graph".format(key))
return sess,tf_dict
```

CHAPTER 5
PERFORMANCE EVALUATION

Performance evaluation is important to make sure the project is carrying out it's duties correctly. We have to check the working condition of the system and making sure that it works properly. Measuring effectiveness is not tangible or easy to grasp and it also takes longer to determine

5.1 EVALUATION SCENARIO

5.1.1 Scenario based testing

Scenario based testing is the testing of a by the install the system and test the input face image and detect the face who is he/she and also detect who wore. In terms of the the project the following scenarios were tested

(i) face detection

This is tested by all the images detecting the faces correctly and show the name of the person in the screen.

(ii) Masked face detection

This is tested by all the images that wore the masks and detect the masked face and show the name of the person in the screen accurately .

(iii) Detect if a person is wearing a mask or not

This is tested by the faces of person and put the mask on person without mask and alignment if mask is proper to faces.

5.1.2 Load Testing

Masked face detection can be used by Admin of the system. Time taken to load is directly proportional to size of User Database. For testing we took a database of over 85,000 images , and it took about 9 mins to load. The system provide quick real time result all the time. The system run smoothly all the time with a normal GPU like NVIDIA MX150 which now commonly available in laptops.

5.1.3 Install/Uninstall testing

There is a need for create the Environment for training. Training time for one epoch is about 5400 seconds and we have about 100 epochs. The application consume about 380 MB and also consume about 1.5GB GPU .It works well with high GPU. The system can be unistalled simply by deleting the folder.

5.2 RESULTS

The system works well and act as a detecting the faces and also masked faces .After scenario based evaluation, it is confirmed that all the features of the system works well as needed. Images of the faces added successfully after creating the account or directory and the adding name details are successfully updated in the database.

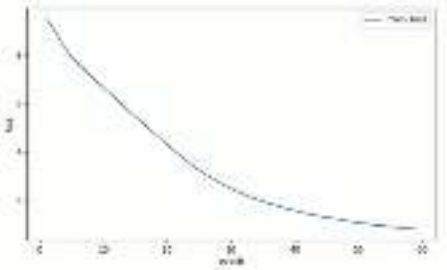

Figure 5.1: Train vs loss

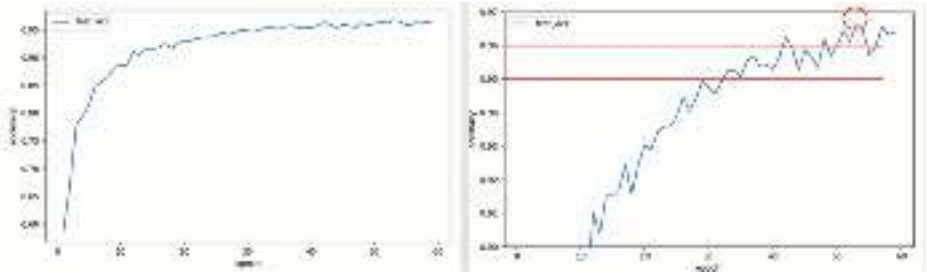

Figure 5.2: Accuracy as epoch increases

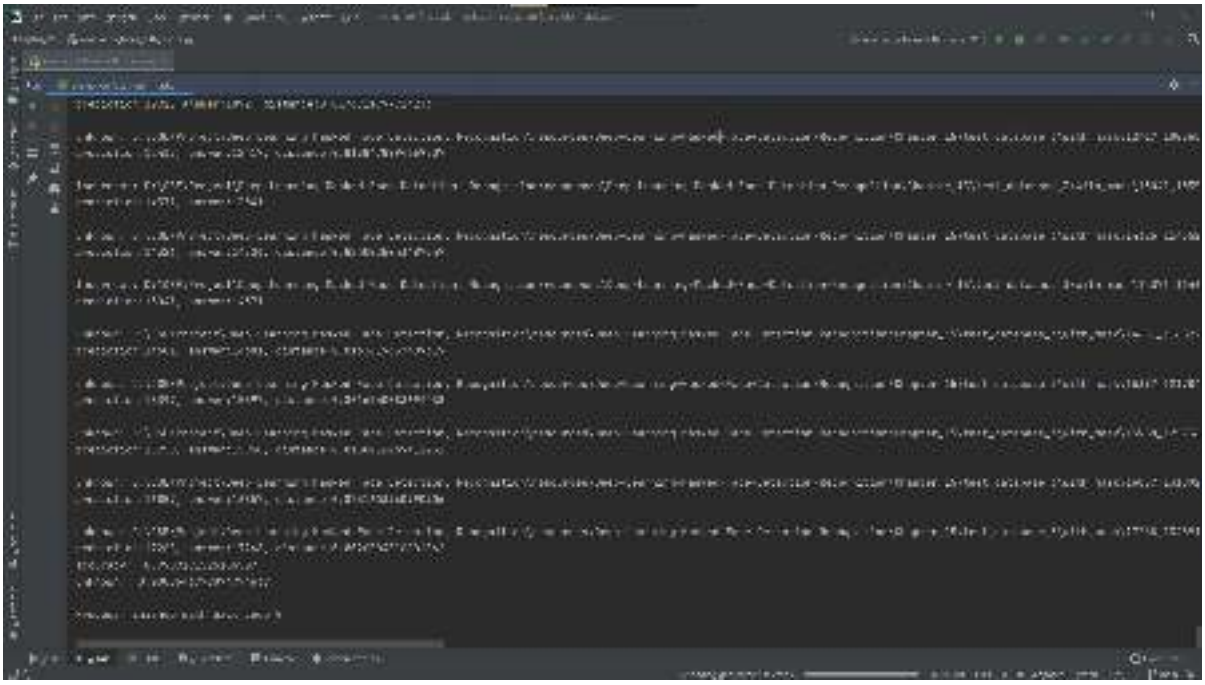

Figure 5.3: Calculated accuracy

CHAPTER 6
DOCUMENTATION

6.1 INTRODUCTION

This section contains documentation of the product "Masked face detection ". This project aims at developing a platform where in Shopping complex, institutions and also in face locking system Where we can detect our faces with and without mask

We plan to develop a system this project to reduce the time for detecting faces when wearing mask

6.2 INSTALLATION

This system can install through a TensorFlow and python packages by pip and setup the environment.

6.3 WORKING WITH THE PRODUCT

This section outlines the main users of our proposed system and the details of various functions that the system provides.

6.3.1 User

Only show the faces to the camera module. And detect their Face with and without mask.

6.3.2 Admin

Admin is responsible for handling of user database.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

We intent to design a platform that are suitable for shopping mall , institution , public transportation systems etc. to fulfill for all the facial recognition co-ordinate and detect the faces with and without mask.

Masked face detection software can also improve real time recognising faces when wearing mask.

7.2 ADVANTAGES

- Recognize faces with masks
- Detect if a person is wearing a mask or not.
- Can work on low power GPU
- Easy to update user database

7.3 LIMITATIONS & FUTURE EXPANSIONS

Use case for this project is limitless as people are using face recognition as apart of their daily life. We plan on making the user database dynamically accessible.We also plan on decreasing the system size and making the algorithms and models faster. This might also make system process run on lower resources.

REFERENCES

1. Rishabh garg(15 Aug 2016) "Face Alignment: Deep multi-task learning", published in Towards Data Science

2. Vu HN, Nguyen MH, Pham C. "Masked face recognition with convolutional neural networks and local binary patterns". Appl Intell (Dordr). 2022;52(5):5497-5512. doi: 10.1007/s10489-021-02728-1. Epub 2021 Aug 14. PMID: 34764616; PMCID: PMC8363871.,

3. J. Prinosil and O. Maly, "Detecting Faces With Face Masks," 2021 44th International Conference on Telecommunications and Signal Processing (TSP), 2021, pp. 259-262, doi: 10.1109/TSP52935.2021.9522677.

4. P. Hofer, M. Roland, P. Schwarz, M. Schwaighofer and R. Mayrhofer, "Importance of different facial parts for face detection networks," 2021 IEEE International Workshop on Biometrics and Forensics (IWBF), 2021, pp. 1-6, doi: 10.1109/IWBF50991.2021.9465087.