A Thesis/Project/Dissertation Report

on

TITLE OF THESIS/PROJECT/DISSERTATION

Submitted in partial fulfillment of the

requirement for the award of the degree of

B.TECH CSE

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. A Arul Prakash
Assistant Professor

Reviewer
Mr. G nagarajan
Gnagarajan@galgotiasuniversity.edu.in

Group ID – BT3144

S. No	Enrolment Number	Admission Number	Student Name	Degree / Branch	Sem
1	19021011931	19SCSE1010791	Rahul kapri	B.tech/CSE	5
2	19021011283	19SCSE1010079	Amit Yadav	B.tech/CSE	5

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING /
DEPARTMENT OF COMPUTERAPPLICATION
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
December, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"FACE DETECTION APPLICATION FOR COVID-19"** in partial fulfillment of the requirements for the award of the B.TECH CSE submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name… Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

RAHUL KAPRI,19SCSE1010791

AMITYADAV,19SCSE1010079

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

Name

Designation

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Rahul kapri – 19SCSE1010791. Amit Yadav, 19SCSE1010079 has been held on 16/10/2021 and his/her work is recommended for the award of B. TECH SCSE.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

Acknowledgement

I would like to express my special thanks of gratitude to my Group mate as well as my teacher Mr. A Arul Prakash who gave me the golden opportunity to do this wonderful project on the topic FACE DETECTION APPLICATION FOR COVID-19 which also helped me in doing a lot of research and I came to known about so many new things I am really thankful to them.
Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Abstract

As one of the most successful applications of image analysis and understanding face recognition has recently gained significant attention especially during the past several years There are at least two reasons for such a trend is the wide range of commercial and law enforcement applications and the second is the availability of feasible technologies after 2 years of research .

COVID-19 pandemic has rapidly affected our day-to-day life disrupting the world trade and movements. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customers to wear masks correctly to avail of their services. Therefore, face mask detection has become a crucial task to help global society. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn. The proposed method detects the face from the image correctly and then identifies if it has a mask on it or not. As a surveillance task performer, it can also detect a face along with a mask in motion. The method attains accuracy up to 95.77% and 94.58% respectively on two different datasets. We explore optimized values of parameters using the Sequential Convolutional Neural Network model to detect the presence of masks correctly without causing over-fitting.

Existing Problem: Challenges in face detection, are the reasons which reduce the accuracy and detection rate of face detection. These challenges are complex background, too many faces in images, odd expressions, illuminations, less resolution, face occlusion, skin color, distance and orientation etc. Odd expressions Human face in an image may have odd expressions unlike normal, which is challenge for face detection.

Proposed Solution: Kairos Offers a wide variety of image recognition solutions through their API. Their API endpoints include identifying gender, age, emotional depth, facial recognition in both photo and video, and more. Trueface.ai One flaw with some facial recognition APIs is that they are unable to differ-entiate between a face and a picture of a face. TrueFace.ai solves that problem with their ability to do spoof detection through their API.

Tools and Technology Used : Face detection is a computer technology that determines the location and size of a human face in the digital image. The facial features are detected and any other objects like trees, buildings and bodies are ignored from the digital image. It can be regarded as a specific case of object-class detection, where the task is finding the location and sizes of all objects in an image that belongs to a given class

Results: Facial recognition uses computer-generated filters to transform face images into numerical expressions that can be compared to determine their similarity. These filters are usually generated by using deep "learning," which uses artificial neural networks to process data.

Conclusion and Future Scope : In recent years face detection has achieved considerable attention from researchers in bio metrics, pattern recognition, and computer vision groups. There is countless security, and forensic applications requiring the use of face recognition technologies Currently many companies providing facial biometric in mobile phone for purpose of access.

List of Figures

List of Table

Table of Contents

I. CHAPTER-1

Introduction

According to the World Health Organization (WHO)'s official Situation Report – 205, coronavirus disease 2019 (COVID-19) has globally infected over 20 million people causing over 0.7million deaths. Individuals with COVID-19 have had a wide scope of symptoms reported – going from mellow manifestations to serious illness. Respiratory problems like shortness of breath or difficulty in breathing is one of them.
To curb certain respiratory viral ailments, including COVID-19, wearing a clinical mask is very necessary. The public should be aware of whether to put on the mask for source control or aversion of COVID-19. Potential points of interest of the utilization of masks lie in reducing vulnerability of risk from a noxious individual during the "pre-symptomatic" period and stigmatization of discrete persons putting on masks to restraint the spread of virus.

Face mask detection involves in detecting the location of the face and then determining whether it has a mask on it or not. The issue is proximately cognate to general object detection to detect the classes of objects. Face identification categorically deals with distinguishing a specific group of entities i.e. Face. It has numerous applications, such as autonomous driving, education, surveillance, and so on [5]. This paper presents a simplified approach to serve the above purpose using the basic Machine Learning (ML) packages such as TensorFlow, Keras, OpenCV and Scikit-Learn
.
Moreover, the FR system is providing vast benefits compared to other biometric security solutions such as palmprints and fingerprints. The system captures biometric measurements of a person from a specific distance without interacting with the person. In crime deterrent applications, this system can help many organizations identify a person who has any kind of criminal record or other legal issues. Thus, this technology is becoming essential for numerous residential buildings and corporate organizations.

KEYWORDS

1. Machine Learning: Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.
2. Open-CV: Open-CV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems.
3. Eigen Face: Eigenfaces is a method that is useful for face recognition and detection by determining the variance of faces in a collection of face images
4. Fisher Face: Fisher face is one of the popular algorithms used in face recognition, and is widely believed to be superior to other techniques.
5. LBPH: The Local Binary Pattern Histogram (LBPH) algorithm is a face recognition algorithm based on a local binary operator, designed to recognize both the side and front face of a human.

Tools and Technology Used

Tools: OS/LINUS/MAX

VISUAL STUDIO CODE/BRACKET/PYCHARM

PYTHON/OPEN-CV/WEBCAM/LIBRARY

Face detection is a computer technology that determines the location and size of a human face in the digital image. The facial features are detected and any other objects like trees, buildings and bodies are ignored from the digital image. It can be regarded as a specific case of object-class detection, where the task is finding the location and sizes of all objects in an image that belongs to a given class. Face detection is a computer technology that determines the location and size of a human face in the digital image. The facial features are detected and any other objects like trees, buildings and bodies are ignored from the digital image. It can be regarded as a specific case of object-class detection, where the task is finding the location and sizes of all objects in an image that belongs to a given class. Due to the significant development of machine learning, the computing environment, and recognition systems, many researchers have worked on pattern recognition and identification via different biometrics using various building mining model strategies. Some common recent works on FR systems are surveyed here in brief. Deng et al. proposed additive angular margin loss (Arc Face) to accomplish face acknowledgment. The proposed Arc Face has an unmistakable geometric understanding as a result of the specific correspondence to geodesic separation on a hypersphere. They also introduced the broadest exploratory assessment against the FR method utilizing ten FR datasets.

1 **TensorFlow**: TensorFlow is an open source framework developed by Google researchers to run machine learning, deep learning and other statistical and predictive analytics workloads.[5] Like similar platforms, it's designed to streamline the process of developing and executing advanced analytics applications for users such as data scientists, statisticians and predictive modelers. Google has also used the framework for applications that include automatic email response generation, image classification and optical character recognition, as well as a drug discovery application that the company worked on with researchers from Stanford University.

2 **Keras**: While deep neural networks are all the rage, has been a barrier to their use for developers new to machine learning. There have been several proposals for improved and simplified high-level APIs for building neural network models, all of which tend to look similar from a distance but show differences in closer examination.

Keras (is one of the high level neural networks APIs) a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation.functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

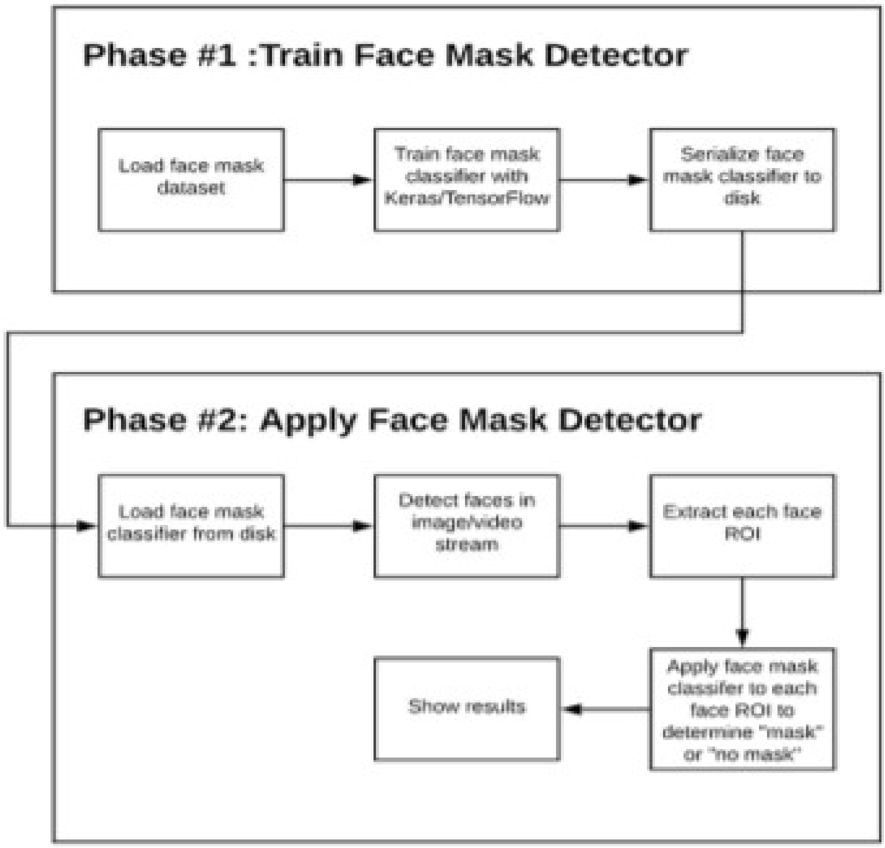
3. **OpenCV:** OpenCV (Open source computer vision library) is a library of programming functions mainly aimed at real-time computer vision. The library is cross platform. OpenCV was built to provide a common infrastructure for computer vision and applications. [9]The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, track moving objects, etc.

4. **Deep Learning**: Deep learning is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

Deep learning methods aim at learning feature hierarchies with features from higher level of the hierarchy formed by the composition of lower level features. Automatically learning features at multiple levels of abstraction .

5. **Phases:** In order to train a custom face mask detector, we broke our project in two distinct phases, each with its own respective sub-steps:

1. Training: Here we have focused on loading our face mask detection dataset from disk, training a model (using Keras/TensorFlow) on this dataset, and then serializing the face mask detector to disk.
2. Deployment: Once the face mask detector is trained, we then moved on to loading the mask detector, performing face detection, and then classifying each face as with mask or without mask.



6 **Dataset**: The dataset we have used in the project is created by Prajna Bhandary. This dataset consists of 1,376 images belonging to two classes:
● with mask: 690 images
● without mask: 686 images

The goal is to train a custom deep learning model to detect whether a person is or is not wearing a mask. To create this dataset, Prajna had the ingenious.
● Taking normal images of faces
● Then creating a custom computer vision Python script to add face masks to them, thereby creating an artificial (but still real-world applicable) dataset Facial landmarks are applied to face to build the dataset of people wearing masks. Facial landmarks allow us to automatically infer the location of facial structures, including eyes, eyebrows, nose, mouth, jawline.

7. **Implementation**: We trained the images of people not wearing a face mask. The next step is to apply face detection. Here we've used a deep learning method to perform face detection with OpenCV. Then the next step is to extract the face ROI with OpenCV and NumPy slicing. And from there, we apply facial landmarks, allowing us to localize the eyes, nose, mouth, etc. Next, we need an image of a mask. This mask will be automatically applied to the face by using the facial landmarks. We will repeat this process for all the input images. If you use a set of images to create an artificial dataset of people wearing masks, you cannot "reuse" the images

The set of TensorFlow. Eras imports allow for:

1. Data augmentation
2. Loading the MobilNetV2 classifier (we will finetune this model with pre-trained ImageNet weights)
3. Building a new fully-connected (FC) head
   Pre-processing
4. Loading image data

We prepared MobileNetV2 for fine tuning. Fine-tuning is a strategy I nearly always recommend to establish a baseline model while saving considerable time. Fine-tuning setup is a three-step process:

1. Load MobileNet with pre-trained ImageNet weights, leaving off head of network
2. Construct a new FC head, and append it to the base in place of the old head
3. Freeze the base layers of the network. The weights of these base layers will not be updated during the process of backpropagation, whereas the head layer weights will be tuned.

During training, we have applied on-the-fly mutations to our images in an effort to improve generalization. This is known as data augmentation, where the random rotation, zoom, shear, shift, and flip parameters are established. The imutils paths implementation helped us to find and list images in our dataset. And we have used matplotlib to plot our training curves.

Face mask detector training accuracy/loss curves demonstrate high accuracy and little signs of overfitting on the data. We're now ready to apply our knowledge of computer vision and deep learning using Python, OpenCV, and TensorFlow/Keras to perform face mask detection

II. CHAPTER -2

LITERATURE REVIEW

This section gives an overview on the major human face recognition techniques that apply mostly to frontal faces, advantages and disadvantages of each method are also given. The methods considered are eigenvalues (eigenfeatures), neural networks, dynamic link architecture, hidden Markov model, geometrical feature matching, and template matching. The approaches are analyzed in terms of the facial representations they used.
90.9% for the multi modal biometric versus 71.6% for the ear and 70.5% for the face.

In face detection method, a face is detected from an image that has several attributes in it. According to , research into face detection requires expression recognition, face tracking, and pose estimation. Given a solitary image, the challenge is to identify the face from the picture. Face detection is a difficult errand because the faces change in size, shape, color, etc. and they are not immutable. It becomes a laborious job for opaque image impeded by some other thing not confronting camera, and so forth. Authors in think occlusive face detection comes with two major challenges: unavailability of sizably voluminous datasets containing both masked and unmasked faces, and exclusion of facial expression in the covered area. Utilizing the locally linear embedding (LLE) algorithm and the dictionaries trained on an immensely colossal pool of masked faces, synthesized mundane faces, several mislaid expressions can be recuperated and the ascendancy of facial cues can be mitigated to great extent. According to the work reported in convolutional neural network (CNNs) in computer vision comes with a strict constraint regarding the size of the input image. The prevalent practice reconfigures the images before fitting them into the network to surmount the inhibit. Aside from handwashing, the use of facemasks is also valuable in infectious disease control, especially in circumventing droplet transmission. For example, the effectiveness of surgical masks and N95 masks in blocking the transmission of SARS are 68% and 91%, respectively. Facemasks, when fitted properly, effectively disrupt the forward momentum of particles expelled from a cough or sneeze, preventing disease transmission. Even if the facemasks are ill-fitting, they are still able to interrupt the particles and airborne viruses sufficiently, such that these pathogens do not reach the breathing zones of people nearby.

Project Design

1. System design

The major requirement for implementing this project using python programming language along with Deep learning, Machine learning, Computer vision and also with python libraries. The architecture consists of Mobile Net as the backbone, it can be used for high and low computation scenarios. We are using CNN Algorithm in our proposed system.

Implementation:

We have four modules

1. Datasets Collecting: We collect no of data sets with face mask and without masks. we can get high accuracy depends on collecting the number of images.
2. Datasets Extracting: We can extract the features using mobile net v2 of mask and no mask sets
3. Models Training: We will train the model using open cv, keras (python library).
4. Facemask Detection: We can detect Pre processing image and also detect via live video. If people wear mask, it will permit them, if not then it will give the buzzer to wear mask to prevent them from virus transmission.

The effects of COVID-19 on the global economy can be seen with the naked eye, as the confinement of people in the homes brings with it less production and slows down the commercial dynamism. However, it should be noted that in situations of health crisis such as the one that continues to be experienced, it is relevant to put people's health before any productive activity. That is why biosecurity measures and social distancing protocols have been implemented to limit the spread of this dangerous virus. As well as the capacity in public institutions, industries and other establishments has been limited, highlighting the so-called telework (in certain cases). Thus, companies have implemented various methodologies, strategies, and techniques to protect the integrity and health, both when entering and staying in face-to-face work sessions. As previously mentioned, CNN have been an important technological tool during this pandemic. Although most approaches have been taken towards the diagnosis of the disease, monitoring and prevention has also been covered.

Today, the use of a personal face mask is a mandatory preventive measure. Keeping the mouth, nose, and cheeks covered has now made people only recognizable by their eyes, eyebrows, and hair, which is a problem for the human eye, which tends to find similarities in several faces that have similar features. This problem also affects computer systems, as facial recognition systems are now very common. They are used to unlock the smartphone, access sensitive applications, and to enter certain places. Current systems usually process information from the entire face of the person, which is why technology must adapt to these new conditions. All this is done with the purpose of maintaining the biosecurity of the user, but giving them the opportunity to continue with the activities as naturally as possible.

The literature has shown that there are systems that seek to identify whether people use it properly. These works have had very good results. However, facial recognition using biosecurity material has not yet been explored. All of this motivated the present investigation, in which a detection system with two approaches is presented. The first is to develop a face classifier, starting from a database of people with and without a mask. The second describes a facial recognition algorithm in controlled environments, which allows for personnel to be identified automatically, without removing the face mask. This can be implemented as an access system to an institution or a home, but at a low cost. This is ensured by using open-source programming software and simple features that reduce computational expense. For this reason, the possibility of improving the adaptability of current facial recognition systems, in the face of new circumstances, has been established as a starting hypothesis.

2. Requirements

The programming language used here is Python. For optimal operation, a high-processing equipment (GPU) is needed. However, we received no external financing, so we chose to work with free Google servers, which are available in Google Colab. Another of the essential requirements is to have the necessary databases in order to carry out the training and obtain the classification and recognition models. Taking into account that building the database of these databases requires a high investment of time when working with artificial intelligence and especially with convolutional neural networks.

Additionally, it is necessary to develop a consent form for the people who will allow for taking photographs for the facial recognition algorithm database. This is necessary because there are currently no databases for the recognition of people with face masks.

For this, it is necessary to rely on Art 6.1d of the European General Data Protection Regulation (RGPD), in connection with article 46.3 of the LOU. Here, it is mentioned that the data of a person will be treated in accordance with the exercise of public powers conferred on the universities as responsible for the treatment of the data of the students. As well as biometric data ((article 9.2.a) of the RGPD), in which consent will be needed so that it can be part of the exams where facial recognition techniques are applied. The collection, filing, processing, distribution, and dissemination of these information data will require the authorization of the owner and the mandate of the law.

3. System Development

It is proposed to design a system that is capable of identifying a person's face, even if it is with or without a mask. For the system to work properly, it is necessary to use two databases: the first is for classifier training and consists of a large number of images of people who wear a face mask and others who do not. The second is used for training the facial recognition system, and here there are people with and without the biosafety material (face mask). The input data are obtained either from an image, or a video and the architecture used is MobileNet, with the aim of having a better precision and robustness. This project is divided into three stages, which are described below.

3.1. First Stage

This stage focuses on finding the location and dimension of one or more faces, regardless of whether or not they wear a mask, within an image. For this, the OpenCV Deep Learning-based face detection model is used and, as a result, the region of interest (ROI) is obtained, which contains data such as the location, width, and height of the face.

3.2. Second Stage

A diagram of the operation of the second stage in **Figure 1. This** is where the classifier training is performed to detect faces with a mask and without a mask. For this, the "Real-World-Masked-Face-Dataset" database available on Git-Hub is used. Unzipping the files makes available a large number of images of people of Asian origin wearing a mask. From this database, the training of the classifier of the first stage is carried out.

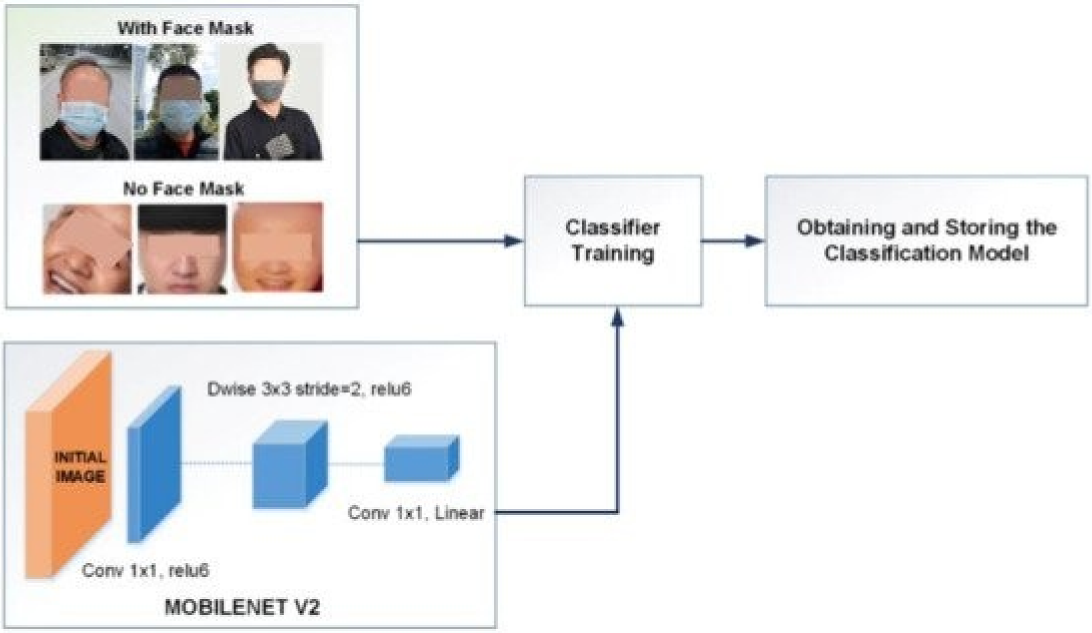


Figure 1. Classifier training block diagram.

For the classifier, MobileNet V2 architecture is used, as it uses smaller models with a low latency and low parameterization power. This improves the performance of mobile models in multiple tasks and benchmarks, resulting in a better accuracy. It also retains the simplicity and does not require any special operator to classify multiple images and various detection tasks for mobile applications. MobileNetV2 is 35% faster at object detection compared with the first version, when combined with Single Shot Detector Lite. It is also more accurate and requires 30% fewer parameters, as the Bottleneck encodes the intermediate inputs and outputs, as shown in **Figure 2**.

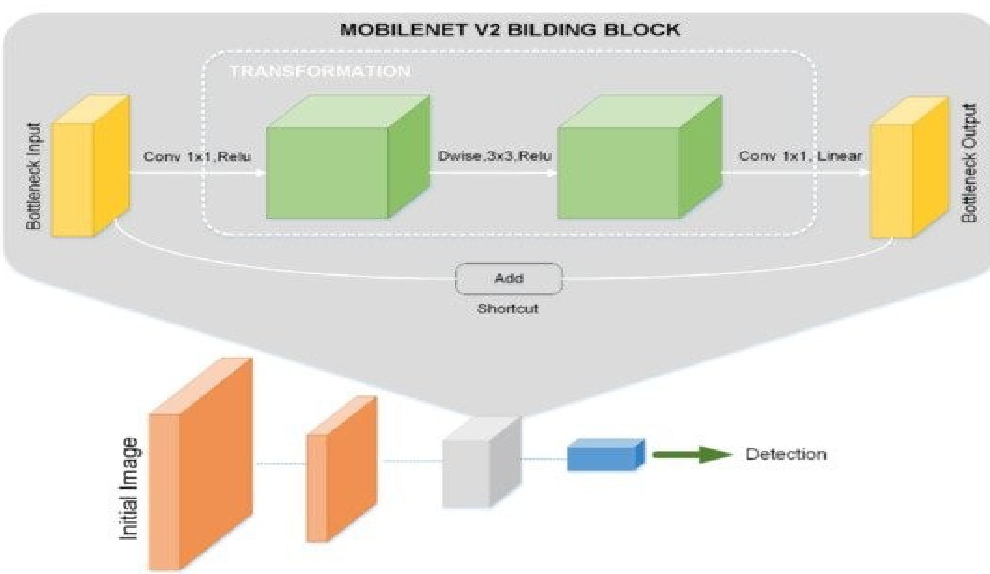


Figure 2. Architecture of MobileNetV2.

At the same time, the inner layer encapsulates the model's ability to transform of lower-level concepts (pixels) to higher-level descriptors (image categories). This is available in the "TensorFlow" library in Python, and through "Transfer Learning", a change is made in the last layer of the convolutional neural network. It should be noted that this architecture is selected because it is a model that does not require excessive computational expense and is therefore efficient in terms of processing speed. For the training of the first classifier presented in **Figure 1**, the input data of the neural network come from a scaling of the color images to a size of 224 × 224 pixels. The architecture used comprises a max-pooling layer (7 × 7), a flatten layer; a hidden layer of 128 neurons with a "ReLu" activation function, a Dropout of 0.5, an output layer with two neurons, and a Softmax activation function. Regarding the training of the classifier, the configurations are used.
Settings used in the classifier training.

At this point (first and second stage together), we have information about where the face is and whether or not it has a face mask. Therefore, it is possible to pass this information to the third stage, which is where the recognition task is carried out. **Figure 3** shows a diagram of the operation of the second stages.

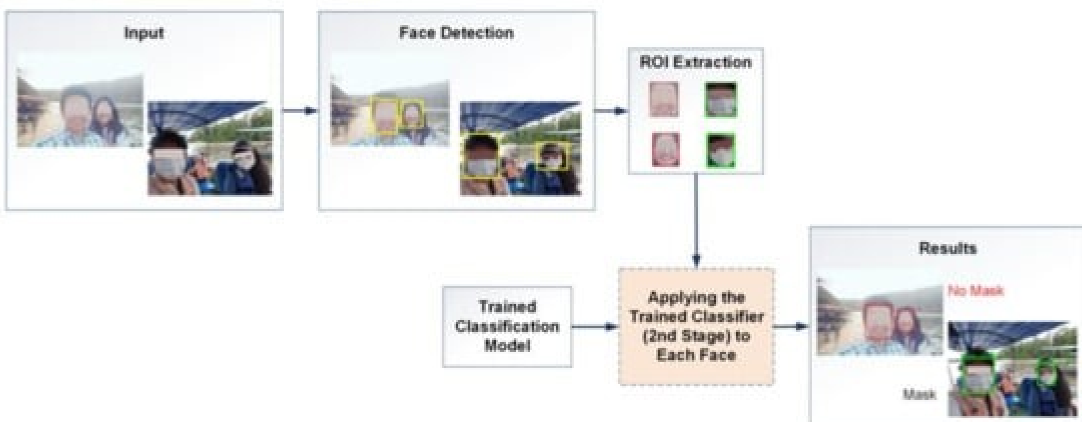


Figure 3. Block diagram for face detection with and without a face mask.

3.3. Third Stage

Once the face of the person has been identified, in the third stage, facial recognition is carried out, for which a set of own observations is used that is built based on the faces of various people. For the construction of the set of observations, a balance is sought in terms of gender, namely, five women and five men from whom the images are obtained. **Figure 4** shows the set of faces using a mask and **Figure 5** shows without a face mask.

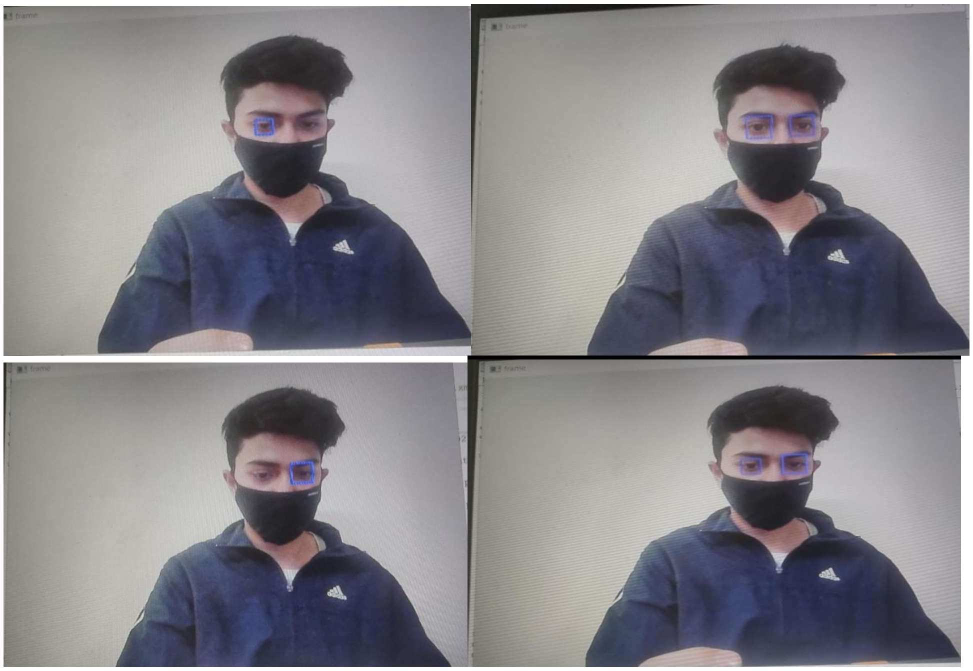


Figure 4. Set of observations using a face mask.

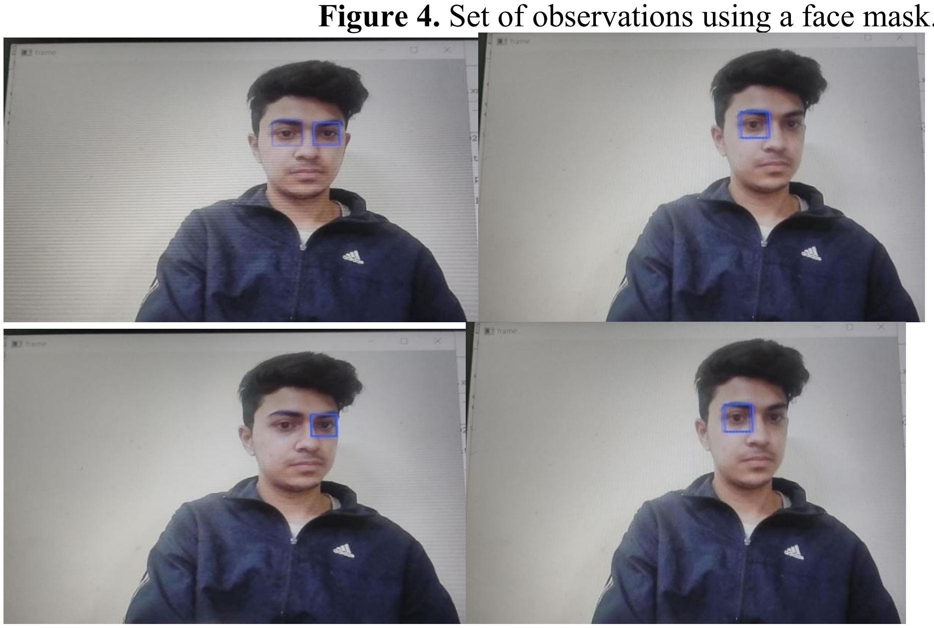


Figure 5. Set of observations without using a face mask.

1. Training of Facial Recognition Models

The procedure to obtain the images is as follows: (i) during a week, daily videos of the face of each individual are obtained, seeking to capture different angles and different environmental conditions (lighting changes). (ii) From the videos obtained, images are captured at different moments in order to build a set of observations with images. (iii) The images where the face is not found in the entirety are eliminated. At the end of the procedure, a total of 13,359 images are obtained; 7067 (52.9%) with a face mask and 6292 (47.1%) without a face mask. In practice, acquiring so many images of a face requires a short video recording of a few seconds showing different views of the face. Regarding the identification of the images used for the recognition of people, the images have been labeled from left to right, as follows:

- Women: W1, W2, W3, W4, and W5 (Vicky, Mela, Damaris, Ale, and Yaritza, respectively).
- Men: M1, M2, M3, M4, and M5 (Oscar, Jorge, Pablo, John, and Jonathan respectively).

In this way, when a person is recognized in an image, the name information can be accessed and placed in the image. Therefore, once the necessary data have been obtained, the recognition model is trained. The two facial recognition models follow the architecture of **Figure 6**, which is briefly explained below. To do this, from this set of observations, the facial recognition model has two approaches:

1. The first model aimed at recognizing people using a face mask.
2. The second model aimed at recognizing people not using a face mask.

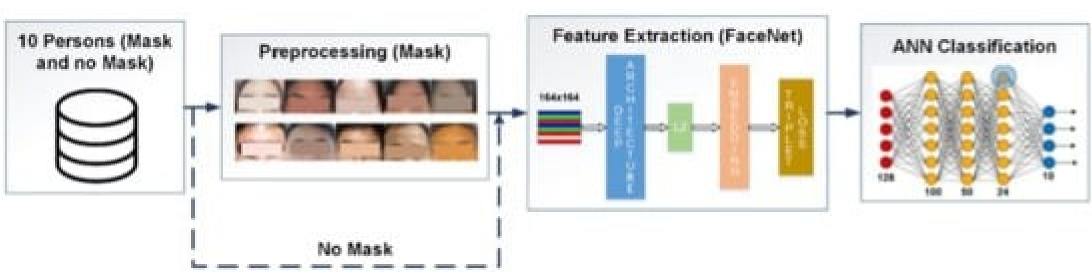


Figure 6. Block diagram of the training of the two face recognition models (mask and no mask).

Database: Set of observations of the faces of people using a mask and without using a mask for both approaches of the third stage.

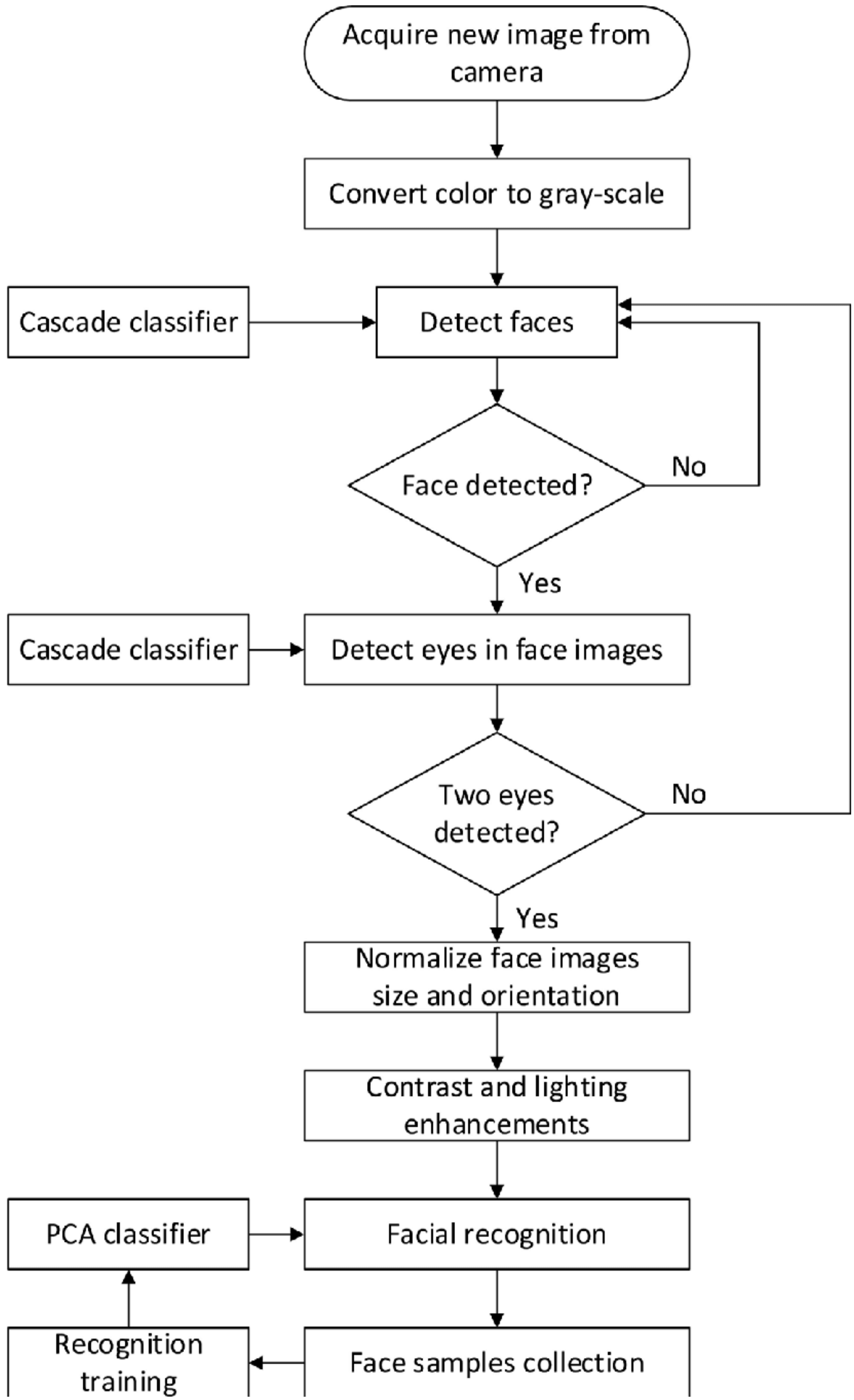
Preprocessing: For the facial recognition model of people using a face mask, only 3/5 of the upper part of the face is extracted. This in order to discriminate the mask that the person is using (as in the experimental tests, this information is not useful for the recognition model). Whereas, for the facial recognition model without a face mask, the image of the full face is used. In both cases, the resulting image is scaled to a size of 164 × 164 pixels.

Characteristics extraction: The FaceNet model is used to extract the most essential characteristics of the face. This model extracts the most essential features from the input image, in this case a face, and returns a vector of 128 features. The input of the network is an image with a human face and, using a deep metric learning technique, it calculates the metric to generate vectors of real characteristics **[17].** This network is a model belonging to PyTorch, which can generate neural networks similar to Caffe **[18].** The image is inserted into the network, then passes through the neural network and obtains the embedding of each face represented by $f(x) \in R^d$. This method attempts to ensure that the image of a specific person (x_i^a) is closer to all of the images of the same person (x_i^p) and away from images of other people (x_i^n). Equation (1) shows the calculation of the loss (L), where α is the margin applied between positive and negative pairs [21]. It receives an image of 164 × 164 pixels as the input data, and a vector of characteristics of 128 elements called "face embedding" is obtained at the output.

Flow chart

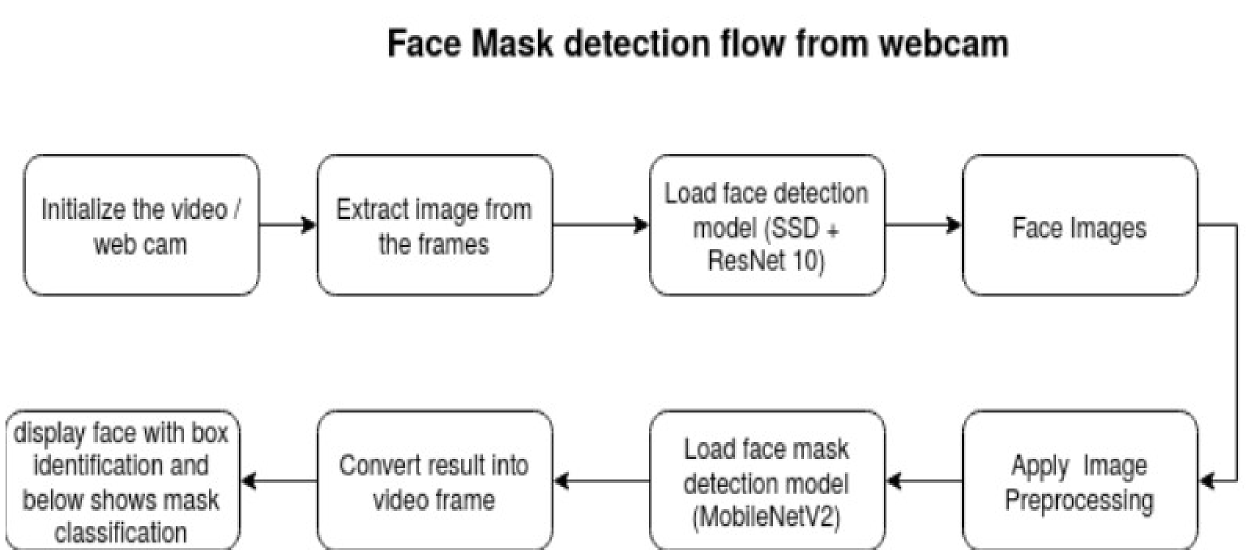
The flow chart shown below is the data representation of the face detection without mask. In this flow chart the following steps are followed:

1. Firstly, acquiring new images from the camera, whenever the user opens the camera and start the program. The camera will capture the image of the user and then stores it in the data set for further detection.
2. The camera after capturing the image it converts the color of the image into grey-scale. So that the camera can understand it properly and stores it in the data set.
3. If the image is already stored in the data set or the user is new to the program, the program will detect the face by the help of the cascade classifier which is a built in library of python.
4. If the user is in front of the camera the output screen will show, the face is detected. And if the user is not standing in front of the camera the output screen will show that no face is detected.
5. After detecting the face , the algorithm search for the eye by using haarcascade xml file.
6. The program checks for two eye .
7. After collecting data through PCA it will store in the data.


Acquire new image from camera
Convert color to gray-scale
Cascade classifier
Detect faces
Face detected?
No
Yes
Cascade classifier
Detect eyes in face images
Two eyes detected?
No
Yes
Normalize face images size and orientation
Contrast and lighting enhancements
PCA classifier
Facial recognition
Recognition training
Face samples collection

Mask detection

1. Firstly acquiring new images from the camera , whenever the user opens the camera and start the program. The camera will capture the image of the user and then stores it in the data set for further detection .
2. Now extracting the images form the frame.
3. Loading the face detection model(SSD + ResNet) .
4. The camera after capturing the image it converts the color of the image into grey-scale. So that the camera can understand it properly and stores it in the data set.
5. Applying the face detection process for the user standing in front of the camera.
6. Load the mask detection model (mobileNet).
7. Convert the result into the video frame.
8. Displaying face with the box identification and below shows mask classification.

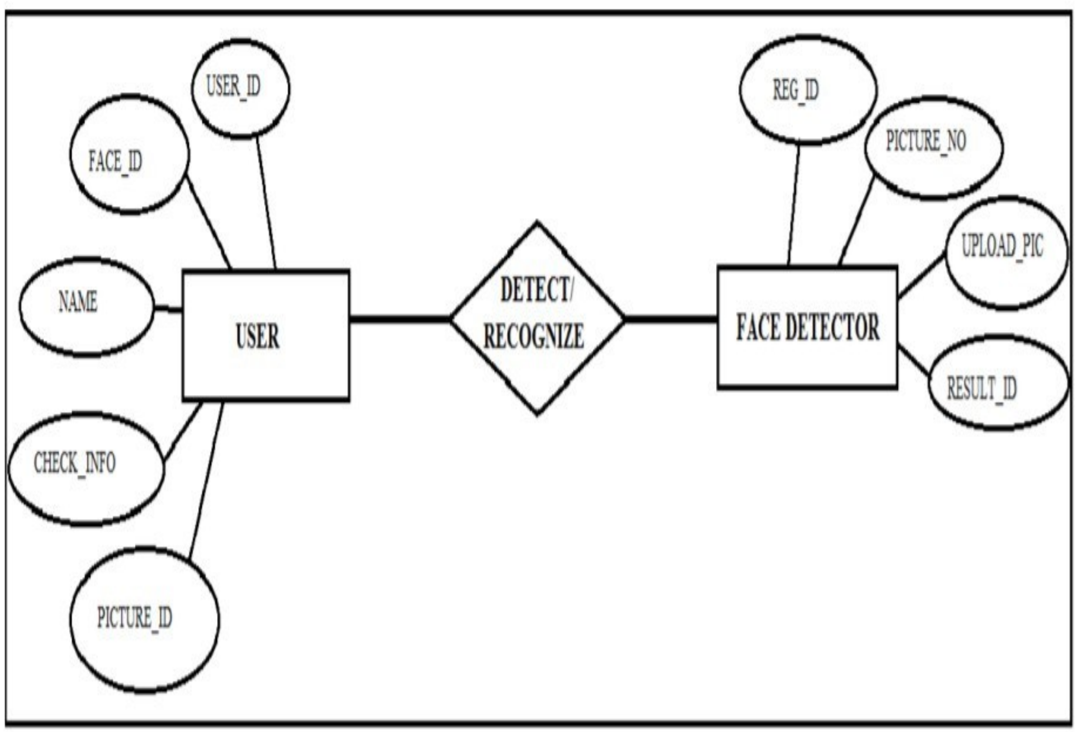


E-R diagram

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

b. ENTITY RELATIONSHIP DIAGRAM:

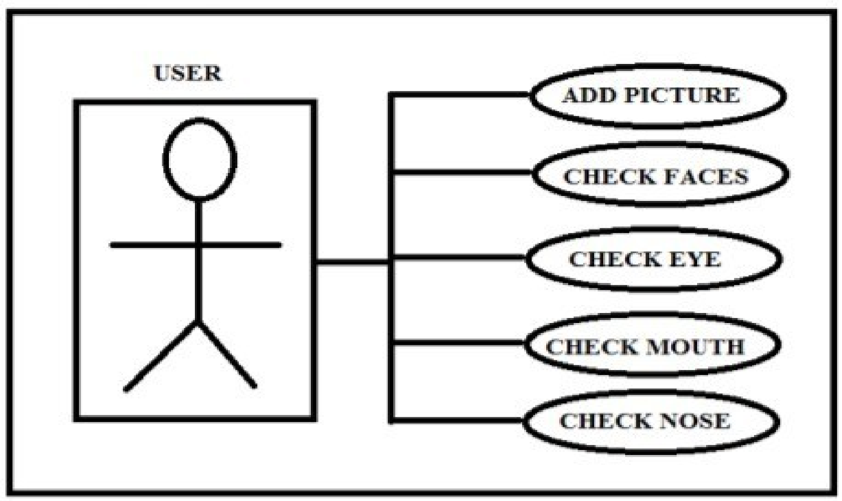


Use Case Diagram

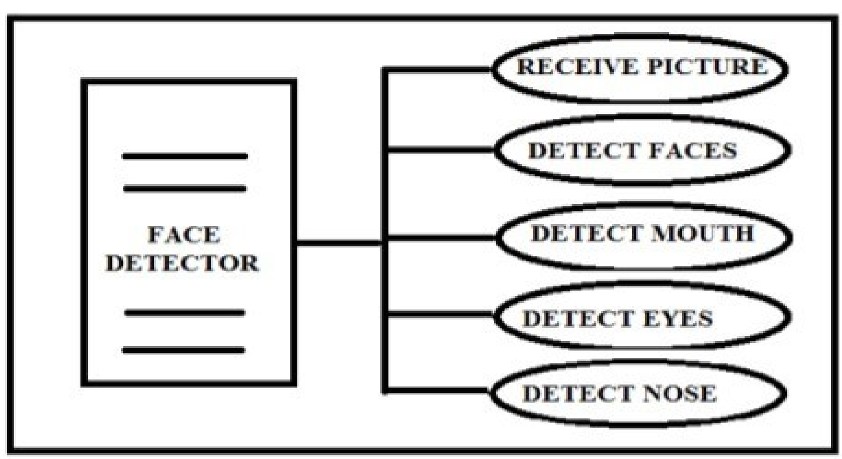
A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

c. USE CASE DIAGRAM:

User Module:



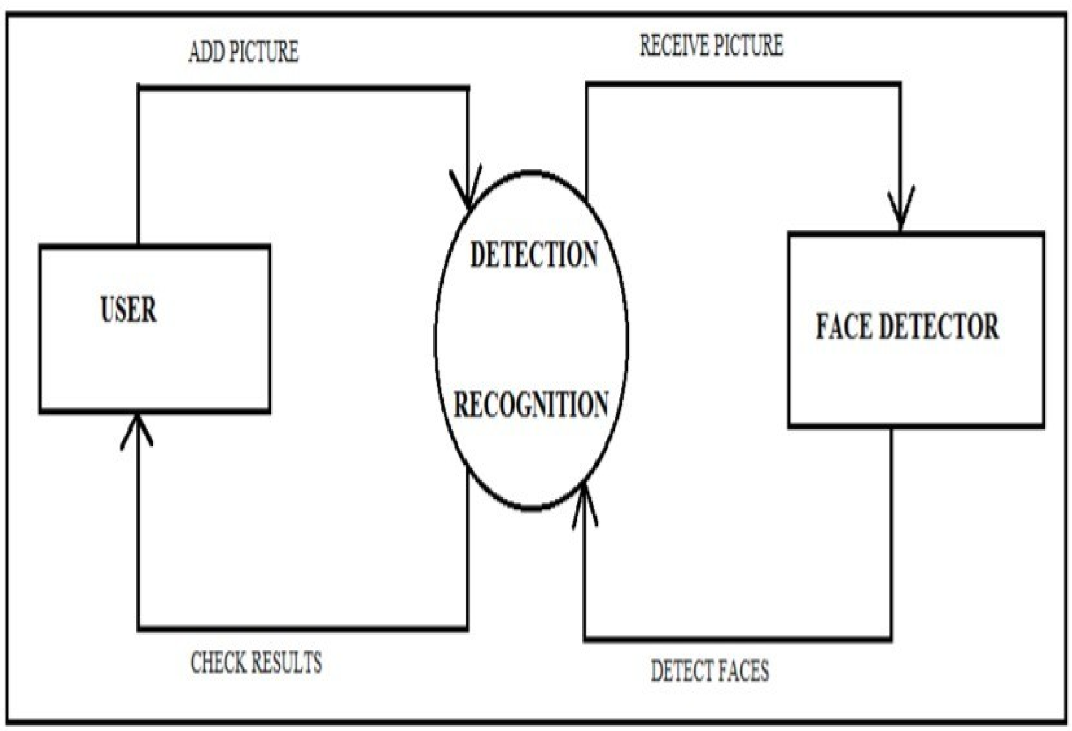
Software Module:



Data Flow Chart Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

a.DATA FLOW DIAGRAM:



RELATED WORK

Face detection is defined as the procedure that has many applications like face tracking, pose estimation or compression. Face detection is a two-class problem where we have to decide if there is a face or not in a picture. This approach can be seen as a simplified face recognition problem.

AdaBoost: Adaboost is an algorithm for constructing a strong classifier as a linear combination. Adaboost, short for Adaptive Boosting, is a machine learning algorithm. It is a meta-algorithm and can be used in conjunction with many other learning algorithms to improve their performance. Adaboost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers. Adaboost generates and calls a new weak classifier in each of a series of rounds. For from a set of training images. This method can be used for both face detection and face locations. In this method, a standard face (such as frontal) can be used. The advantages of this method are that it is very simple to implement the algorithm, and it is easily to determine the face locations such as nose, eyes, mouth, etc. based on the correlation values.

ALGORITHMS OF FACE DETECTION

Haar- like feature: Haar-like wavelets are binary rectangular representations of 2D waves. A common visual representation is by black (for value minus one) and white (for value plus one) rectangles. The square above the 0-1- interval shows the corresponding Haar-like wavelet in common black-white representation. The rectangular masks used for visual object detection are rectangles tessellated by black and white smaller rectangles. Those masks are designed in correlation to visual recognition tasks to be solved, and known as Haar like feature each call, a distribution of weights is updated that indicates the importance of examples in the data set for the classification. On each round, the weights of each incorrectly classified example are increased, and the weights of each correctly classified example are decreased, so the new classifier focuses on the examples which have so far eluded correct classification.

Survey

Dataset on Face mask Recognition. The dataset below is taken from the project which signifies the accuracy and recognize a face with mask and without mask.

Type of Database	Dataset	Scale	#Face	Masked face image	Occlusion
Masked Faced Detection Database	FDB	2845	5171	-	-
	MALF	5250	11931	-	RIGHT
	CALEBA	20000	2022599	-	-
	WIDERFACE	32203	19400	-	RIGHT
Faced masked datasets	MAFA	30811	37824	37824	RIGHT
	RMFRD	95000	9200	9200	RIGHT
	SMFRD	85000	500	5100	RIGHT

Precision table of the data collected from the project to recognize. This table shows the precision and FI-score of the program which helps in identifying the accuracy of the project that how fast and efficient the dataset and the code is to identify a face with mask or without mask.

	Precision	Recall	F1-Score
W4	0.99	0.98	0.98
W3	1.00	1.00	1.00
M4	0.99	1.00	0.99
M5	1.00	0.99	0.99
M2	0.99	1.00	1.00
W2	1.00	1.00	1.00
M1	0.99	1.00	1.00
M3	1.00	1.00	1.00
W1	0.97	0.99	0.98
W5	1.00	1.00	1.00
Accuracy			1.00
Macro avg	0.99	0.99	0.99
Weighted avg	1.00	1.00	1.00

Precision table

Comparison of proposed with recent face mask detection m which shows the difference between face detection and mask detection after collecting the general data from face detection.
Some of the data collected from projects are-

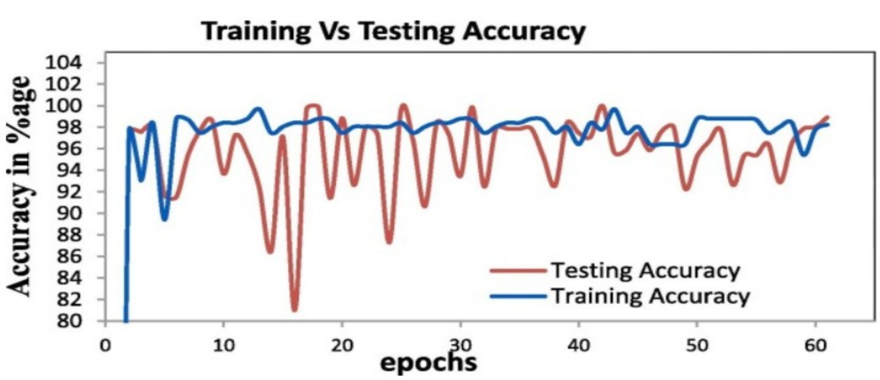
1. Retina facemask based on model mobileNet.
2. Retina facemask based on ResNet.
3. Proposed model on ResNet.

Model	Face Detection		Mask Detection	
	Precious(%)	Recall(%)	Precsion(%)	Recall(%)
Ratina Face Mask based on Mobiile Net	83	95.6	82.3	89.1
Retina Face Mask based on Res Net	91.9	96.3	93.4	94.5
Proposed model based on ResNet	99.2	99.0	98.92	98.24

Comparison of proposed model.

Trianing data set

The graph below Trainig accuracy and testing accuracy between the face and mask detecion dataset .
This graph data indicates the real time data fluctuation by the action performing by the subject (person) with removing mask and wearing mask . The red line shows the testing accuracy
And the blue line shows the training accuracy .



Training data

Comparison between MobileNet -SSD ,ResNet50 and their various comparison based on random Vs Hard/soft complexity of test data .

The table below shows the data compared between mobileNet and ResNet on the parameters .some of the parameter are –
1.Random split .
2. Soft/hard split .
3. image complexity prediction.
4. mask detection Time.

Comparision Parameter	Mobile Net—SSD				
	100-0%	75-25%	50-50%	25-75%	0-100%
Random split(mAP)	0.8868	.9095	.9331	.09650	.9899
Soft/hard Split(mAP)	.80688	.9224	.9631	.9892	.9899
Mask detection time(ms)	.05	.192	3.08	5.07	6.02
Total Computation Time(ms)	.05	1.97	3.13	5.12	6.02

Comparison parameter.

III. CHAPTER -3

Functionality/Working of Project

Module Description

Challenges in face detection, are the reasons which reduce the accuracy and detection rate of face detection. These challenges are complex background, too many faces in images, odd expressions, illuminations, less resolution, face occlusion, skin color, distance and orientation etc.

Odd expressions Human face in an image may have odd expressions unlike normal, which is challenge for face detection.

Kairos Offers a wide variety of image recognition solutions through their API. Their API endpoints include identifying gender, age, emotional depth, facial recognition in both photo and video, and more.

Trueface.ai One flaw with some facial recognition APIs is that they are unable to differ-initiate between a face and a picture of a face. TrueFace.ai solves that problem with their ability to do spoof detection through their API.

The face is considered the most critical part of the human body. Research shows that even a face can speak, and it has different words for different emotions. It plays a crucial role in interacting with people in society.

It conveys people's identity and thus can be used as a key for security solutions in many organizations. The facial recognition (FR) system is increasingly trending across the world as an extraordinarily safe and reliable security technology. It is gaining significant importance and attention from thousands of corporate and government organizations because of its high level of security and reliability.

Performance Analysis for Face Detection

In order to verify the effectiveness of the feature-guided attention network, the detector with the attention network removed is used as a test baseline (baseline), and Face++ proposed in [17] is used as a comparison method. Baseline adopts the feature fusion method consistent with FPN to build the model

In order to more intuitively understand the attention module on the performance of the face detector, Figure 5 gives visualized face prediction results. Figure 5(a) is a face image, and input it to Baseline,

In the selected detection image, the shape and scale of the face are quite different, especially the gray level of the face and the adjacent background is similar. According to the maximum analysis of the response map, these appearance changes cannot get the accurate boundary, which leads to ConvNet, KSDD, and FACEILD cannot get the accurate face.

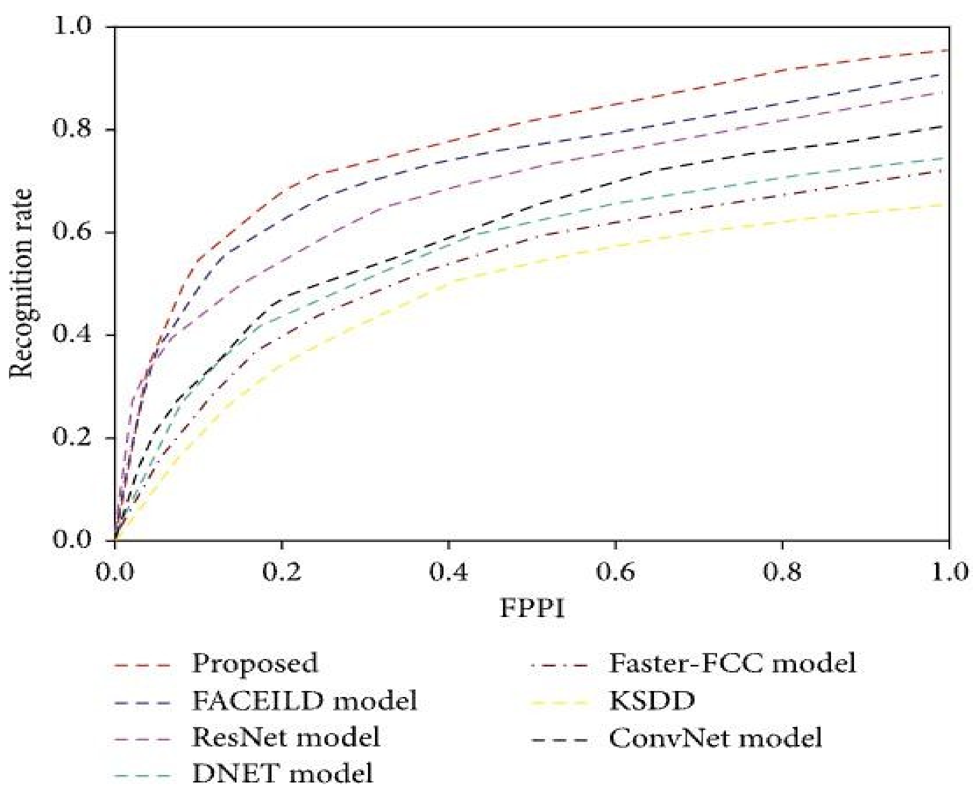
Experimental analysis

In order to evaluate the performance of the face detection and recognition algorithm based on the visual attention-guided mechanism proposed in this paper, LFW (labeled faces in the wild database), CMUFD database (CMU face detection database) [23], and UCFI database (UCD color face image) [25] are used as face detection and recognition data sets.

At present, the video surveillance intelligent analysis system has been able to detect and recognize the face with different scales. However, existing algorithms have a large number of false detection for face objects under partial occlusion mainly.

Generalization Analysis

In order to verify the generalization performance of the proposed method, the proposed method was trained on the LFW training set, and cross-data set experiments were performed on the CMUFD database. The heavy subset consists of face objects with a height greater than 50 pixels and a visible range of [0.20, 0.65]. As shown in Figure 6, FPPI represents the statistical results of face detection and recognition algorithms at different detection rates. In order to facilitate comparison in different deep networks, the experiment mainly discusses the detection results of each algorithm when FPPI = 1 for analysis.



IV. CHAPTER -4

Results

We have successfully trained our model and tested it on a real time face using the laptop's camera. Our face mask detector correctly labeled the person's face as either 'Mask' or 'No Mask'. As you can see in this image that the face is labeled as 'Mask' when the person is wearing a mask and labeled as 'No Mask' when the person is not wearing a mask.

The experiments carried out have the purpose of demonstrating the potential use of the system, so tests are carried out using the recall metrics, Precision, F1-score, and the corresponding macro avg and weighted avg. The objective of using these metrics is to evaluate the system from different perspectives. Recall and precision indicate the ability of the model to correctly detect true positives. Recall also considers the false negatives detected, and the precision of the false positives detected by the model. False positives, in this case of face detection with masks, occur when an object is labeled as a face. For example, the system frames a plant as a face with a face mask, as it is false that a plant is a positive face. The reasons this can occur in our system are numerous, which is why hard work is needed in order to collect a large database so that the model being trained can better distinguish the desired classes (faces). False negatives occur when a face is not detected in the first stage, because the face has covered areas that make classification difficult; in this proposal, this initial classifier is a pre-developed tool. On the other hand, the F1-score provides a global measure of the system's performance, it is a combination of precision and recall (in a single value), with 0 being low performance and 1 being the best possible performance (all cases detected correctly).

By considering the macro avg metric, sd can get a general idea of the average of all of the experiments, while the weighted avg establishes an average measure of all of the experiments, but considering the number of observations of each class. Thus, in the event that a class has a higher score, the final weighted avg score will not be affected by it, but will give a value of importance to each score depending on the amount of observation. When considering these metrics, what is sought is to verify the robustness of the method by classifying both classes. In the second stage, the face classifier training with a mask and faces without a mask takes a period of approximately 10 h, and in the third stage the extraction of face embeddings takes approximately 5 h and the ANN-based classifier training takes 10 min._

Implementation

```
import cv2
import numpy as np
from keras.models import load_model
model=load_model("./mode2-010.h5")

labels_dict={0:'without mask',1:'mask'}
color_dict={0:(0,0,255),1:(0,255,0)}

size = 4
webcam = cv2.VideoCapture(0) #Use camera 0

# We load the xml file
classifier =
cv2.CascadeClassifier('/home/shivam/.local/lib/python3.6/site-packages/
cv2/data/haarcascade_frontalface_default.xml')

while True:
    (rval, im) = webcam.read()
    im=cv2.flip(im,1,1) #Flip to act as a mirror

    # Resize the image to speed up detection
    mini = cv2.resize(im, (im.shape[1] // size, im.shape[0] // size))

    # detect MultiScale / faces
    faces = classifier.detectMultiScale(mini)

    # Draw rectangles around each face
    for f in faces:
        (x, y, w, h) = [v * size for v in f] #Scale the shapesize backup
        #Save just the rectangle faces in SubRecFaces
        face_img = im[y:y+h, x:x+w]
        resized=cv2.resize(face_img,(150,150))
        normalized=resized/255.0
        reshaped=np.reshape(normalized,(1,150,150,3))
        reshaped = np.vstack([reshaped])
        result=model.predict(reshaped)
        #print(result)

        label=np.argmax(result,axis=1)[0]
```

```
        cv2.rectangle(im,(x,y),(x+w,y+h),color_dict[label],2)
            cv2.rectangle(im,(x,y-40),(x+w,y),color_dict[label],-1)
                                cv2.putText(im,    labels_dict[label],    (x,    y-
cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2)

        # Show the image
        cv2.imshow('LIVE',   im)
        key = cv2.waitKey(10)
        # if Esc key is press then break out of the loop
        if key == 27: #The Esc key
            break
    # Stop video
    webcam.release()

    # Close all started windows
    cv2.destroyAllWindows()
```

test.py - C:\Users\Rahul\Downloads\face-mask-detector-project\test.py (3.8.3)

File Edit Format Run Options Window Help

```python
import cv2
import numpy as np
from keras.models import load_model
model=load_model("./mode2-010.h5")

labels_dict={0:'without mask',1:'mask'}
color_dict={0:(0,0,255),1:(0,255,0)}

size = 4
webcam = cv2.VideoCapture(0) #Use camera 0

# We load the xml file
classifier = cv2.CascadeClassifier('/home/shivam/.local/lib/python3.6/site-packages/cv2/data/haarcascade_frontalface_default.xml')

while True:
    (rval, im) = webcam.read()
    im=cv2.flip(im,1,1) #Flip to act as a mirror

    # Resize the image to speed up detection
    mini = cv2.resize(im, (im.shape[1] // size, im.shape[0] // size))

    # detect MultiScale / faces
    faces = classifier.detectMultiScale(mini)

    # Draw rectangles around each face
    for f in faces:
        (x, y, w, h) = [v * size for v in f] #Scale the shapesize backup
        #Save just the rectangle faces in SubRecFaces
        face_img = im[y:y+h, x:x+w]
        resized=cv2.resize(face_img,(150,150))
        normalized=resized/255.0
        reshaped=np.reshape(normalized,(1,150,150,3))
        reshaped = np.vstack([reshaped])
        result=model.predict(reshaped)
        #print(result)

        label=np.argmax(result,axis=1)[0]

        cv2.rectangle(im,(x,y),(x+w,y+h),color_dict[label],2)
        cv2.rectangle(im,(x,y-40),(x+w,y),color_dict[label],-1)
        cv2.putText(im, labels_dict[label], (x, y-10),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2)

    # Show the image
    cv2.imshow('LIVE',   im)
    key = cv2.waitKey(10)
    # if Esc key is press then break out of the loop
    if key == 27: #The Esc key
```

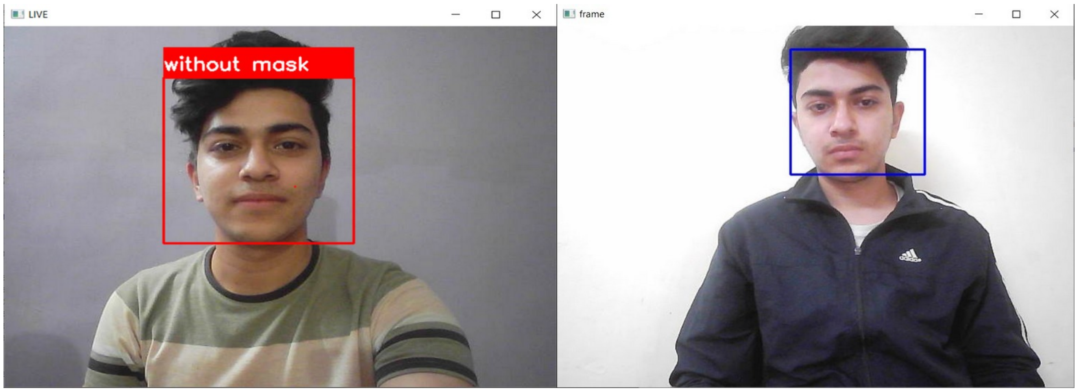
Ln: 16 Col: 21

Output

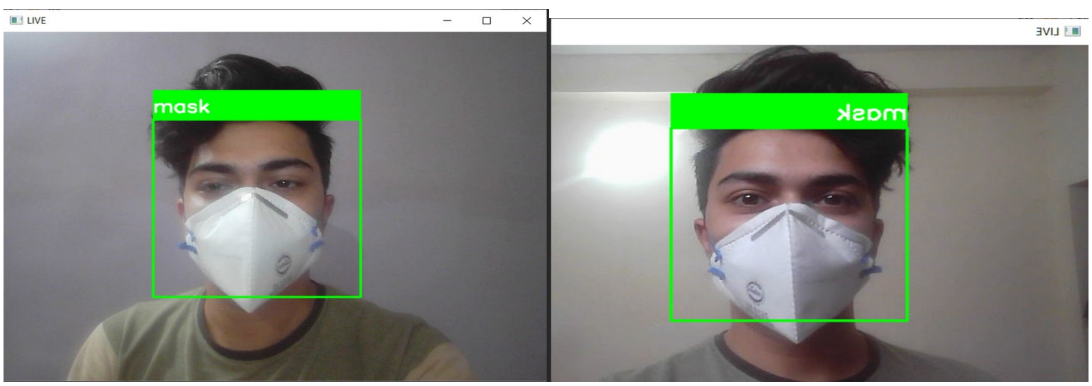
The output below shows: -

1. First the program will run for testing the presence of the face.
2. The program will test the eye and face for the person standing in front of the camera.
3. After detecting the eye and the face it will test for the mask and without mask.
4. If the program detects the face, it will show a green outline to show the presence of the mask.
5. And if the mask is not present then a red outline will show.

Detecting Face and Eye for training the model dataset.



Face detection without mask



Face detection Mask Detection

Conclusion and Future Scope

In recent years face detection has achieved considerable attention from researchers in bio metrics, pattern recognition, and computer vision groups. There is countless security, and forensic applications requiring the use of face recognition technologies

Currently many companies providing facial biometric in mobile phone for purpose of access. In future it will be used for payments, security, healthcare, advertising, criminal identification etc.

Due to the significant development of machine learning, the computing environment, and recognition systems, many researchers have worked on pattern recognition and identification via different biometrics using various building mining model strategies. Some common recent works on FR systems are surveyed here in brief.
Wang proposed a large margin cosine loss (LMCL) by reformulating the Soft-Max loss as a cosine loss by L2 normalizing the two highlights and weight vectors to evacuate outspread varieties and using the cosine edge term to expand the choice edge in precise space. They achieved the highest between-class difference and lowest intraclass fluctuation via cosine choice edge augmentation and normalization.

Tran proposed a disentangled representation learning-generative adversarial network (DR-GAN) with three different developments. First, the encoder-decoder structure of the generator permits DR-GAN to gain proficiency with a discriminative and generative portrayal, including picture blending. Second, the portrayal is unraveled from other face varieties—for example, through the posture code given to the decoder and posture estimation in the discriminator.

Masi proposed to build prepared information sizes for face acknowledgment frameworks: domain explicit information development. They presented techniques to enhance realistic datasets with critical facial varieties by controlling the faces in the datasets while coordinating inquiry pictures presented by standard convolution neural systems. Jonnathann presented a comparison between profound learning and conventional AI strategies (for example, artificial neural networks, extreme learning machine, SVM, optimum-path forest, KNN) and deep learning. For facial biometric acknowledgment, they concentrated

References

1. Health Organization. Global and Alert Response (GAR), Disease outbreaks by year [online] [Accessed April 20 2013]. Available at: http://www.who.int/csr/don/archive/year/en/index.html .

2. Burgess A, Horii M. Risk, ritual and health responsibilisation. Japan's 'safety blanket' of surgical face mask-wearing. *Sociol Health Illn.* 2012;34:1184–98. [PubMed] [Google Scholar]

3. Müller MA, Raj VS, Muth D, et al. Human corona virus EMC does not require the SARS-coronavirus receptor and maintains broad replicative capability in mammalian cell lines. *MBio.* 2012;3:e00515–12. [PMC free article] [PubMed] [Google Scholar]

4. Chan JF, Li KS, To KK, et al. Is the discovery of the novel human betacoronavirus 2c EMC/2012 (HCoV-EMC) the beginning of another SARS-like pandemic? *J Infect.* 2012;65:477–89. [PMC free article] [PubMed] [Google Scholar]

5. Wiemken TL, Peyrani P, Ramirez JA. Global changes in the epidemiology of community-acquired pneumonia. *Semin Respir Crit Care Med.* 2012;33:213–9. [PubMed] [Google Scholar]

6. Ministry of Health Singapore. Top 4 Conditions of Polyclinic Attendances [online] [Accessed August 9 2012]. Available at: https://www.moh.gov.sg/content/moh_web/home/statistics/Health_Facts_Singapore/Top_4_Conditions_of_Polyclinic_Attendances.html .

7. Jefferson T, Foxlee R, Del Mar C, et al. Physical interventions to interrupt or reduce the spread of respiratory viruses: systematic review. *BMJ.* 2008;336:77–80. [PMC free article] [PubMed] [Google Scholar]

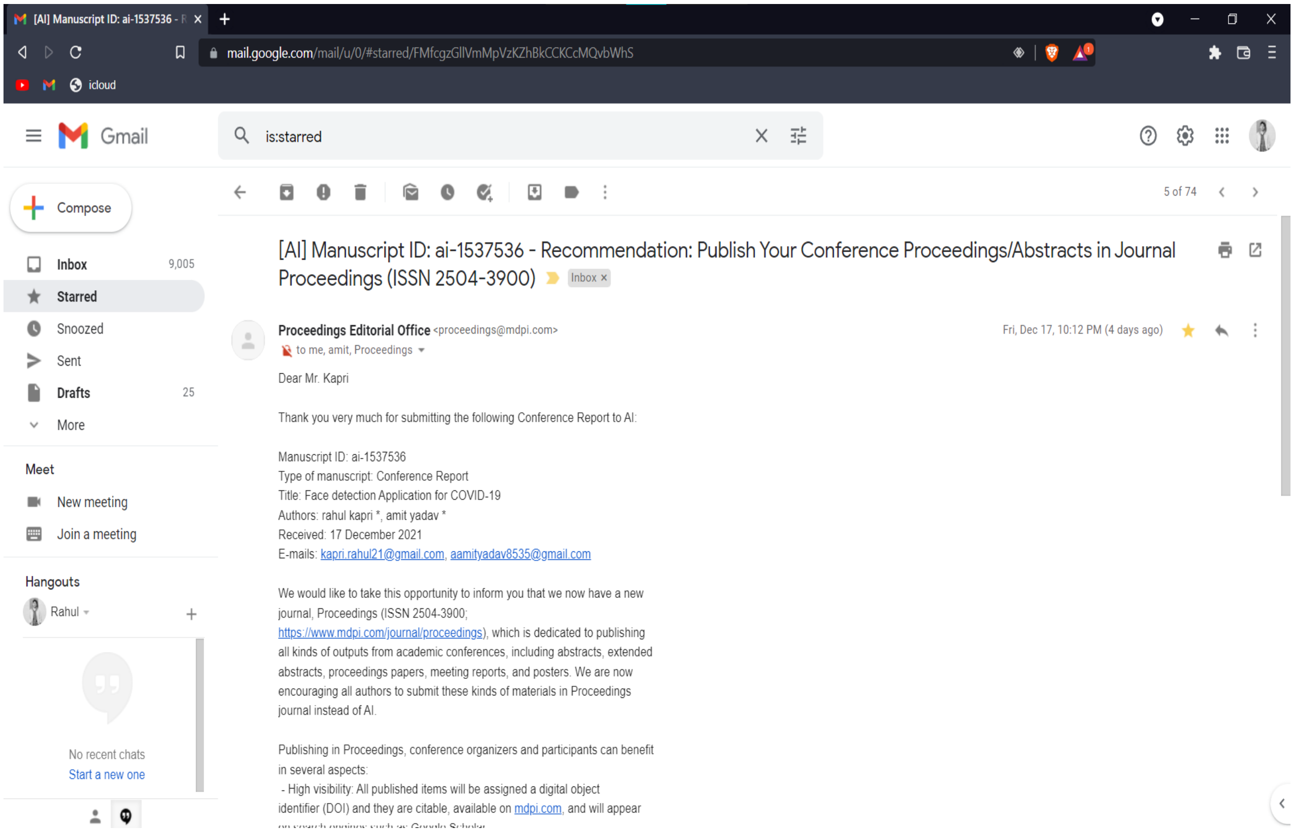
8. Aiello AE, Perez V, Coulborn RM, et al. Facemasks, Hand hygiene, and influenza among young adults: a randomized intervention trial. *PLoS One.* 2012;7:e29744. [PMC free article] [PubMed] [Google Scholar]

9. Aledort JE, Lurie N, Wasserman J, Bozzette SA. Non-pharmaceutical public health interventions for pandemic influenza: an evaluation of the evidence base. *BMC Public Health.* 2007;7:208. [PMC free article] [PubMed] [Google Scholar]

10. Kiliç S, Gray GC. Nonpharmaceutical Interventions for Military Populations During Pandemic Influenza. *Turk Silahli Kuvvetleri Koruyucu Hekim Bul.* 2007;6:285–90. [PMC free article] [PubMed] [Google Scholar]

11. Luby SP, Agboatwalla M, Feikin DR, et al. Effect of handwashing on child health: a randomised controlled trial. *Lancet.* 2005;366:225–33. [PubMed] [Google Scholar]
12. White CG, Shinder FS, Shinder AL, Dyer DL. Reduction of illness absenteeism in elementary schools using an alcohol-free instant hand sanitizer. *J Sch Nurs.* 2001;17:258–65. [PubMed] [Google Scholar]
13. Jefferson T, Del Mar C, Dooley L, et al. Physical interventions to interrupt or reduce the spread of respiratory viruses. *Cochrane Database Syst Rev.* 2010;1:CD006207. [PubMed] [Google Scholar]
14. Furuya H. Risk of transmission of airborne infection during train commute based on mathematical model. *Environ Health Prev Med.* 2007;12:78–83. [PMC free article] [PubMed] [Google Scholar]
15. Eastwood K, Durrheim D, Francis JL, et al. Knowledge about pandemic influenza and compliance with containment measures among Australians. *Bull World Health Organ.* 2009;87:588–94. [PMC free article] [PubMed] [Google Scholar]
16. van der Sande M, Teunis P, Sabel R. Professional and home-made face masks reduce exposure to respiratory infections among the general population. *PLoS One.* 2008;3:e2618. [PMC free article] [PubMed] [Google Scholar]
17. Tang JW, Liebner TJ, Craven BA, Settles GS. A schlieren optical study of the human cough with and without wearing masks for aerosol infection control. *J R Soc Interface.* 2009;6(suppl 6):S727–36. [PMC free article] [PubMed] [Google Scholar]
18. Taylor M, Raphael B, Barr M, et al. Public health measures during an anticipated influenza pandemic: Factors influencing willingness to comply. *Risk Manag Healthc Policy.* 2009;2:9–20. [PMC free article] [PubMed] [Google Scholar]
19. MacIntyre CR, Cauchemez S, Dwyer DE, et al. Face mask use and control of respiratory virus transmission in households. *Emerg Infect Dis.* 2009;15:233–41. [PMC free article] [PubMed] [Google Scholar]
20. Quah SR, Hin-Peng L. Crisis prevention and management during SARS outbreak, Singapore. *Emerg Infect Dis.* 2004;10:364–8. [PMC free article] [PubMed] [Google Scholar]
21. Rosenstock IM. Why people use health services. *Milbank Meml Fund Q.* 1966;44:94–127. [PubMed] [Google Scholar]

22. Tang CS, Wong CY. Factors influencing the wearing of facemasks to prevent the severe acute respiratory syndrome among adult Chinese in Hong Kong. *Prev Med.* 2004;39:1187–93. [PMC free article] [PubMed] [Google Scholar]
23. Kuo PC, Huang JH, Liu MD. Avian influenza risk perception and preventive behaviour among traditional market workers and shoppers in Taiwan: practical implications for prevention. *PLoS One.* 2011;6:e24157. [PMC free article] [PubMed] [Google Scholar]
24. Lau JT, Kim JH, Tsui HY, Griffiths S. Anticipated and current preventive behaviours in response to an anticipated human-to-human H5N1 epidemic in the Hong Kong Chinese general population. *BMC Infect Dis.* 2007;7:18. [PMC free article] [PubMed] [Google Scholar]
25. Lau JT, Griffiths S, Choi KC, Lin C. Prevalence of preventive behaviours and associated factors during early phase of the H1N1 influenza epidemic. *Am J Infect Control.* 2010;38:374–80. [PMC free article] [PubMed] [Google Scholar]


[AI] Manuscript ID: ai-1537536 - Recommendation: Publish Your Conference Proceedings/Abstracts in Journal Proceedings (ISSN 2504-3900)
Inbox
Proceedings Editorial Office <proceedings@mdpi.com>
to me, amit, Proceedings
Fri, Dec 17, 10:12 PM (4 days ago)
Dear Mr. Kapri
Thank you very much for submitting the following Conference Report to AI:
Manuscript ID: ai-1537536
Type of manuscript: Conference Report
Title: Face detection Application for COVID-19
Authors: rahul kapri *, amit yadav *
Received: 17 December 2021
E-mails: kapri.rahul21@gmail.com, aamityadav8535@gmail.com
We would like to take this opportunity to inform you that we now have a new journal, Proceedings (ISSN 2504-3900; https://www.mdpi.com/journal/proceedings), which is dedicated to publishing all kinds of outputs from academic conferences, including abstracts, extended abstracts, proceedings papers, meeting reports, and posters. We are now encouraging all authors to submit these kinds of materials in Proceedings journal instead of AI.
Publishing in Proceedings, conference organizers and participants can benefit in several aspects:
- High visibility: All published items will be assigned a digital object identifier (DOI) and they are citable, available on mdpi.com, and will appear
mail.google.com/mail/u/0/#starred/FMfcgzGllVmMpVzKZhBkCCKCcMQvbWhS
is:starred
Compose
Inbox 9,005
Starred
Snoozed
Sent
Drafts 25
More
Meet
New meeting
Join a meeting
Hangouts
Rahul
No recent chats
Start a new one
5 of 74