VOL. 17, NO. 10, MAY 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

FACE MASK DETECTION AND IDENTIFICATION OF INDIVIDUALS BY USING CNN ALGORITHM

P. Saleem Akram, T. Oam Chandra Laasya, G. Lakshmi Aishwarya and N. Alekhya
Department of E.C.E., KLEF (Deemed to be University), Vaddeswaram, Guntur (D.T), Andhra Pradesh, India
E-Mail: saleemakramp@gmail.com

ABSTRACT

The COVID19 pandemic has had a significant impact on people's social lives. Due to this pandemic, almost every office, institution, organization in the world suffered a great deal from being practically closed. The World Health Organization (W.H.O) recommended everyone wear a mask whenever they step outside or in a public place. Therefore, it is mandatory to cover your face with a mask at public places, social gatherings, etc. Facemask detection has recently become one of the most important tasks to help society. The advancement of technology has proven that deep learning has shown its effectiveness in recognition and classification through image processing. There are many face detection models created by using several algorithms and techniques. Find whether a person has puton a mask properly or not and identify that person who didn't puton a mask properly employing their age and gender. The combination of the face mask detection module and age & gender detection module is used. In our paper, the Haar cascade classifier was implemented to detect faces from the input images in the face mask recognition module. We train this module using CNN. We can recognize faces in this model using the Voila Jones technique and Haar-like features. The face detection module and age & gender detection module is trained by using a Convolutional neural network. A model trained by Tal Hassner and Gil Levi is used to implement Age and Gender detection; an alert sound will be a part of the outcome if the person is not wearing a mask properly. For the facemask detection module, the dataset is taken from Kaggle; images of people wearing masks and not wearing masks are gathered from different sources and formed into a dataset to train the model. In this paper, we used the Adience dataset to train age & gender detection and a dataset from Kaggle containing pictures of people's faces with and without a mask. The model attains an accuracy of 93.42 %for face mask detection and an accuracy of 91.23% for Age and Gender detection.

Keywords: haar-cascade classifier, OpenCV, keras, convolution neural network (CNN), TensorFlow, deep learning.

INTRODUCTION

As per the World Health Organization's (WHO) official Situation Report, COVID-19 has afflicted over 250 million people worldwide, resulting in over 5 million deaths [1]. Covid 19 is an infectious disease caused by the SARS-CoV-2 virus, causing respiratory issues like breathing or windiness. Quite a few people may experience mild to moderate respiratory illness, infected by the inevitable coronavirus. However, people with medical conditions and elderly ones are more likely to suffer from severe illnesses. Senior individuals having lung sickness can have genuine confusion from COVID-19 as they have all the earmarks of being in more danger. As per WHO guidelines, the Government made it necessary for all people to wear masks when going out in public places. Hence, Facemask detection has turned into an urgent errand in present worldwide society [2-3]. Face mask Detection includes recognizing the area of the Face and afterward deciding if the person is wearing the mask properly or not. Face mask detection is the same as an object detection system in which a system detects a particular class of objects. We have imported packages like Keras, TensorFlow, OpenCV to detect the faces. It has numerous applications in various fields such as self-driving cars, industries, surveillance, etc. To identify the person who is not Wearing masks properly, we use age, and gender detection is vital for various real-time applications, such as social identification, biometrics, identity verification, video surveillance, and many more. In the age and gender detection module, the age and gender of the faces in an image are classified. There are various works proposed for age and sexual orientation, the expectation in the beyond quite a while. The prior works were essentially dependent on available made elements [4], separated facial pictures followed by a classifier. Detected Face in face mask detection model is taken as input to age and gender detection module. An Integer is returned by an age classifier that represents the person's age range in the input image. There are nine possible age ranges, so the classifier returns an integer from 0 to 8. The gender classifier returns a binary result where 1 represents female, and 0 represents the male. This model utilized the fundamental Machine Learning (ML) frameworks like TensorFlow, Keras, OpenCV [5]. A convolutional neural network is used to train both modules. Haar cascade features are used to detect the Face from an image and then processed to get the desired output. The package winsound enables the production of an audio output if the detected Face from the image is not wearing a mask properly.

RELATED WORK

Before we put forth the proposed method, we have reviewed some related work on face mask detection and Age-Gender detection. Due to the recent outbreak of COVID-19, wearing a mask has become mandatory. So it isn't easy to detect a face that is covered with a mask. To detect a face, the main criteria required is the facial features to detect the Face as referred to in [6-7]. In recent years, grayscale conversions have played a crucial role in

VOL. 17, NO. 10, MAY 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

detecting face/object in an image or a frame taken from the video as referred to in [8]. To convert the input image (RGB image) to a greyscale image, we use grayscale conversion. By converting the image to grayscale, the features will be more visible, and it will be easy for the model to detect the faces. After grayscale conversion, the features are extracted using CNN, the model is trained as per the research in [9]. As for age and gender detection, these areas have been for decades, and various approaches were introduced to detect the age and gender of a person. In the initial attempts of age and gender detection, constrained images were taken as input when tested, and high accuracy was obtained, leading to difficulties in the real-time scenarios as referred to in [10-11]. Until 2015, many research papers have explained either age detection or gender detection, but not the combination of both the modules. In 2015, [12] introduced an architecture that could detect the age and gender of a person and explained the methodology straightforwardly.

DATASET

To design this model, we require two different datasets for the Face mask detection module and the age & gender detection module.

Figure-1. Samples from dataset includes people wearing mask.

The Kaggle dataset has been used for the face mask detection module [13]. The images in this dataset are collected from various sources, out of which 1,916 people are wearing masks as shown in abovefigure1, while 1,919 are not wearing masks as shown below in as for Age and Gender detection, we will be using the Adience dataset shown in above Figure-2. This dataset consists of different facial images, including various real-world imaging conditions like noise, pose, lighting, and appearance. It's not easy to train the model for detecting the age and gender of an unknown person as different people have different features. Various photographs from Flickr's albums have been gathered. Their area, a total of 26,580 photos of over 2000+ subjects shown in below Figure-3 in nine different age ranges and they are (0-2,4-6,8-13-15-20,21-24,25-32,38-43,48-53, 60& above) and Gender of the person is classified as male or female.

Figure-2. Samples from dataset includes people not-wearing mask.

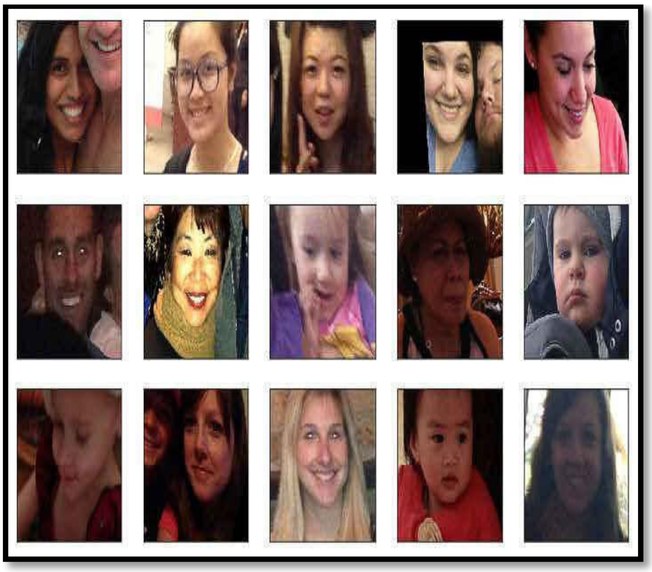

Figure-3. Adience dataset for age and gender detection.

INCORPORATED PACKAGES

Open CV: It's an open-source Computer Vision and Machine Learning library, as the name suggests. As preferred in [14-15], OpenCV allows users to understand how images and videos are stored and manipulated.

MobileNetV2: MobileNetV2 focuses on the cutting-edge execution of models on different tasks. It is a feature extractor for identifying and segmenting objects. It's used as a productive structural block for depthwiseseparable convolutions. It improves precision.

Caffe: Yangqing Jia developed Caffe, a framework for constructing image recognition applications, as part of his Ph.D. Caffe grew to accommodate various machine learning algorithms and many other types of neural networks as more people contributed to its development and one of them is [16].

Keras: Other products are focused on performance and range of capabilities, but Keras is focused on modularity and ease of development. As per the research in [17-18], Keras's ease of use is due to its fast API and straightforward collection of functions. This open-source library is used to build a neural network model with a simple and improved high-level API. The commonly used neural network components in Keras are the layers, activation functions, and various tools, which make it easy to work with image data.

Tensorflow: TensorFlow is an open-source system created by Google specialists to run A.I., profound learning. A detailed explanation is referred to in [19] regarding this library. It's intended to smooth out the most common way of creating and executing progressed investigation applications.

NumPy: Python has a handful of libraries; one of them is NumPy. Functions for working with matrices, Fourier transformations, and linear algebra are also included. In Python, we have lists that operate like arrays, although they are inefficient in processing images.

Matplotlib: Matplotlib is a superb Python visualizing package for 2D array graphs. It is a multi-platform visual analytics software. One of the advantages of visualization is that it enables us to see big amounts of information in easily understandable images. Line, bar, scatter, histogram, and far more plots are provided in Matplotlib.

METHODOLOGY

In this research, the overall architecture is divided into two modules: face mask detection module and age & gender detection module. The first step in the Face mask recognition module is to train the model with a suitable dataset to forecast if a person has worn a mask appropriately.

a) Face Mask Detection Module

In the face Mask Detection module, we prepare the data to fit the model, extract the facial features, etc., to train the model.

A. Data Processing

The process of converting given data from a given format to the desired form makes it more meaningful and informative. Data can be in the form of tables, pictures, recordings, diagrams, and so on. Tasks like data transformation, data cleaning, data reduction, Data integration, etc. were carried out in data processing. Data processing is done because the real world data is generally incomplete, noisy, and inconsistent.

a. Data Visualization: It is the graphical representation of information and data. It translates the data into visual contents such as maps, graphs, etc. Data visualization makes it easier to understand or identify the patterns, trends in large data sets. The dataset is divided into two categories.

The variable categories: ['with_mask', 'without_mask']

By distinguishing the categories, we can find the number of labels. As there are two categories, where labels will be set to[0,1]. Therefore, the mapped variable will be: {'with_mask': 0, 'without_mask': 1}

b. RGB to Gray Conversion: Converting an RGB image to a grayscale image is called grayscaling. It reduces model complexity, allowing the algorithms to work efficiently; Dimensionality reduction is done by converting RGB images with three channels and three dimensions to a 1D greyscale image. This function *"cv2.COLOR_BGR2GRAY"* converts the input image to a grayscale image.

c. Image Reshaping: In CNN, all input images should be in the same size and format using the function cv2.resize(), all the images are resized into 100x100.The pixel range in the images is normalized between 0 to1.

d. Viola-Jones Algorithm: This method looks for specific traits in multiple smaller subregions of a face and tries to recognize a face by looking for them in each part of the region. It must check various positions and scales because an image can contain many faces of various sizes. For the method, Viola and Jones employed Haar-like properties to recognize faces. In this module, we used the haar cascade frontal face classifier.

B. Training of Model

(a) CNN architecture: Convolutional Neural Network Architecture: In deep learning, a convolutional neural network (CNN or ConvNet) is a type of neural network which is most frequently used to explore visual representational mechanisms. CNN takes an input image and assigns importance to different features in images in the form of weights and bias. In this paper, the CNN model is trained on images containing faces with and without masks so that the trained model can classify new images of faces whether the Face has a mask or not. CNN architecture consists of 5 different layers: convolution layer, pooling layer, fully connected input layer, fully connected layer, fully connected output layer. The first component of the convolutional neural network is the convolution layer. Convolution is the process of applying a filter continuously through the pixels of the image. As real-time data consists of non-linear features such as the transition between the pixels, we use Rectified Linear Unit(ReLU) to increase the non-linearity. The second component of CNN is the pooling layer; it is done using another filter in which the maximum value from the convolution result is the feature map. This layer is applied to each feature map separately to create a new set of pooled feature maps. Generally,

The pooling layer is used to reduce the dimensionality, extract the dominant features, and noise suppression. Images are flattened to a single vector at a fully connected layer as input to the SoftMax layer. A fully connected layer assigns weights to the previous layers' features and eventually outputs a suitable classification.

b. Dataset splitting and CNN model preparation

Generally, we split the dataset into training and testing data to train the model. '*train_test_split'* is used to split the dataset. If the *test_size* value is set to 0.1, 10% of data from the dataset is taken for testing and 90% of the data undergoes the training process.

Then, the pictures in the training dataset and the testing dataset are fitted to the Sequential model. In order to get better accuracy and avoid overfitting, we are preparing our model for 20 epochs. Training accuracy across each epoch can be found in Figure-8.

b) Age & Gender Detection Module

The person's face in an image is taken as input to the age and gender detection module. The age classifier returns an integer representing the person's age range, and the gender classifier returns an integer representing the gender of the person.

A. Facial Landmark Detection and Face Alignment: Face detection is next to facial landmark detection and face alignment. The image reprocessing method generally utilizes five landmark detection models: a front model, two long models, and two full-profile models in a multiview open-source face landmark method. These five models were each given one of the facial positions to work on. The face alignment phase runs on five facial landmark models and all recognized faces.

B. CNN Architecture: Our CNN has six layers, 4 of which would be convolutional and two fully connected. Four completely linked layers are utilized for the feature extraction process, each with its own sets of criteria, such as the number of criteria and each filter's kernel size. The detailed layers ate referred to in the above CNN architecture.

C. Training Details: The age group classification will sort unfiltered facial feature images into nine age categories, while the gender classification will sort them into two gender groups. At last, we fine-tuned the network using the test dataset's training data (OIU-Adience). The fine-tuning let the CNN catch up on the dataset's distribution, peculiarities, and bias, resulting in improved performance. The dataset was separated into two parts: 90percent for training and 10percent for testing.

D. Age Group Classification: We implemented an L2 weight decay of 0.0005 to classify unprocessed face pictures into the relevant age group after setting the initial learning algorithm to 0.0001 to allow the model can train further. We use Adam optimizer to help our model generalize and forecast accurately for each of the parameters. Each of the parameters. Figure.9 is the graphical representation of the training accuracy to detect the Face of a person.

E. Gender Classification: In learning the gender classifier, we fix the learning rate to a minute value, the L2 weight degradation to 0.0005, and the momentum term to 0.9. To update all of the training models, we are using Stochastic Gradient Descent (S.G.D.). A graph was used to represent the model's training accuracy, which detects the gender of a person, as shown in Figure-10.

IMPLEMENTATION

For our video input, we'll use OpenCV to accomplish real-time face identification for Livestream via webcam. These frames are types of images that are combined and formed into a video. It will be simple to recognize faces in a Livestream if the input is in the form of still images. We extract the major elements of our Face (such nose, eyes, eyebrows, lips, etc.) known as descriptors from the image/frame by converting it into a grayscale image. This conversion of grayscale makes it easy for different algorithms to perform efficiently. Haar cascade approach (an object detection tool used to recognize items in an image/video) determines the person's face. In this algorithm, there is a face cascade object with a method known as detectMultiScale(). The word MultiScale informs the classifier to detect the Face from different regions of an image in varying sizes. They construct rectangles across the image/subregions. Then this frame/image is received as an argument, and the classifier is performed over the image. If we resize a larger face to a smaller one, it makes the algorithm task easier for detection. For resizing the image, an initial value of 1.06 would be reasonable, which means the size of the image will be reduced by 5%.

After performing all the parameters through the detectMultiScale() function, we obtained all the detections for the final image. These detections are saved using pixel coordinates. The coordinates define the detection in the top-left corner, and the identified faces are circumscribed by rectangles of different widths and heights. Now to display the detected faces, we use the rectangle () function from OpenCV. This function draws a rectangle across the picture and requires the top-left and bottom-right detection corners' coordinates. The function rectangle () has arguments like the original image(frame), coordinates (top-left, bottom-right), the color of the rectangle, and thickness of the vertices of the rectangle. For this model, we have used green color for the rectangular box if the person has put on the mask or the color of the rectangular box will be red. Before displaying the output, if the person is wearing the mask, it simply displays "MASK" with a green color border faceframe and accuracy. If he is not wearing a mask, this image/frame is taken as an input for the second part of the model, which is Age and Gender detection. In this part, we will detect the age and gender of the person not wearing the mask. The internal states of layers and their parameters are described in the Caffe model. If the person in the image did put on a mask properly, an audible alarm and a text on the border face frame with the title "No mask" will be shown accurately. Since we are accessing the inbuilt webcam until any key is

VOL. 17, NO. 10, MAY 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

hit,frames are taken continuously. If it's not a video, we'll have to wait, so we'll use cv2's *Waitley()* and then break.

BLOCK DIAGRAM

The face mask detection module's detailed method is depicted in Figure 4. In the below block diagram, the video input is taken in a live stream via webcam.

Figure-5 describes the detailed block diagram of the age & gender of a person is applied to the image of a person with nomask. The output obtained from the face mask detection model is classified into two parts for further observation. One is where the person wears his mask correctly, and the other one is where he/she doesn't wear their mask properly. In case, if a person did not put on the mask properly, that image is taken as an input for detecting the age and gender of that person.

As we can see in Figure-5, the output occurred from the face mask detection module in the case if there is no mask found on the face of the person, it is taken as an input for the Age and gender detection module where different tools were implemented.

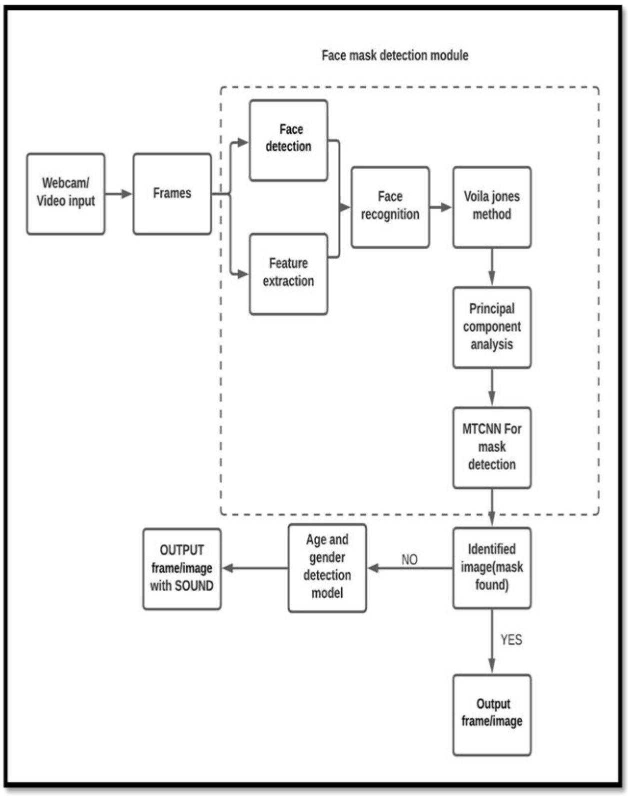

Figure-4. Block diagram for the monitoring system.

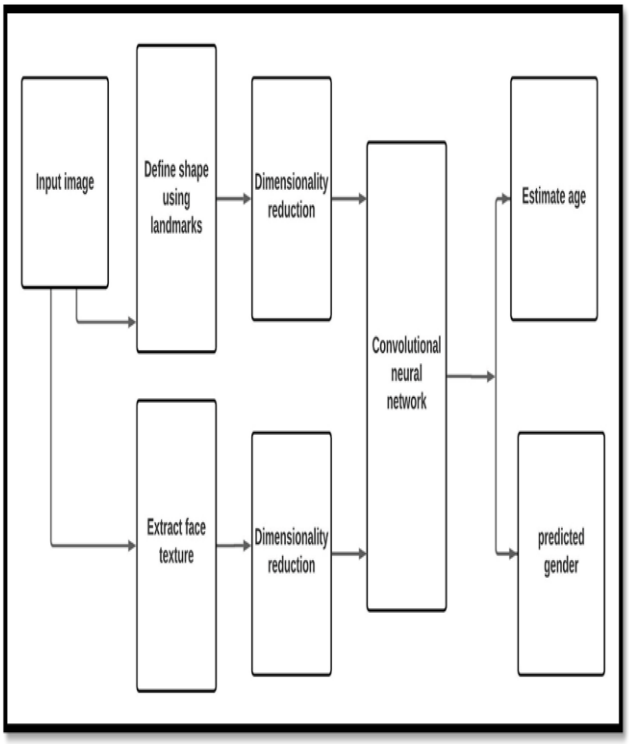

Figure-5. Block diagram for Age-gender detection module.

RESULTS

The results are shown below. Figure-6 describes the person who puts on a mask, and along with that, it displays the accuracy. The mask's precision determines how much of the face it covers. The accuracy displayed in Figure-6 is 99.98%, which means the person is completely wearing the mask.

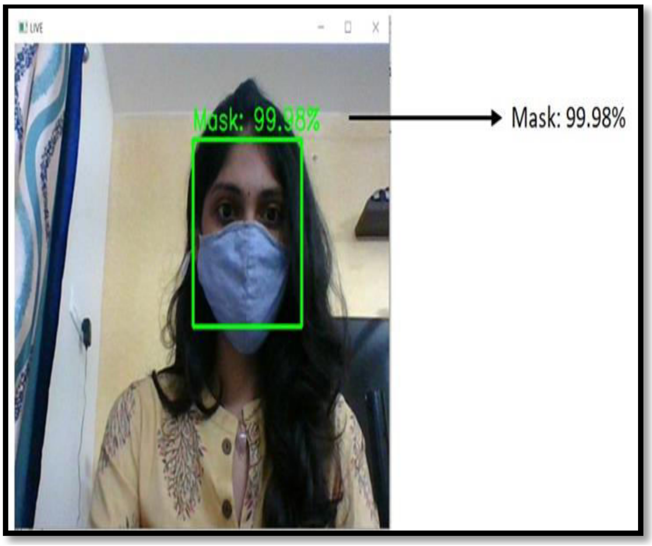

Figure-6. The result shown here is the person wearing a mask along with accuracy.

Figure-7 describes the person not wearing a mask. As the person is not wearing the mask, his age and

VOL. 17, NO. 10, MAY 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

gender are also detected. In Figure-7, we can observe that the person's age is (21-24) and the gender is also displayed. Along with the output displayed with age and gender, it also gives an alert in the form of a sound.

After implementing the architecture of CNN, the age and gender of a person are detected. Once the age and gender are also detected for the person not wearing a mask, an alert is generated in an audio format to intimates that the person hasn't put on his mask. The alert is in the form of a proper format whenever the person is not wearing a mask. Not only does giving an alert, but it also detects the age and gender of the person, as we can observe in Figure-7, as shown below.

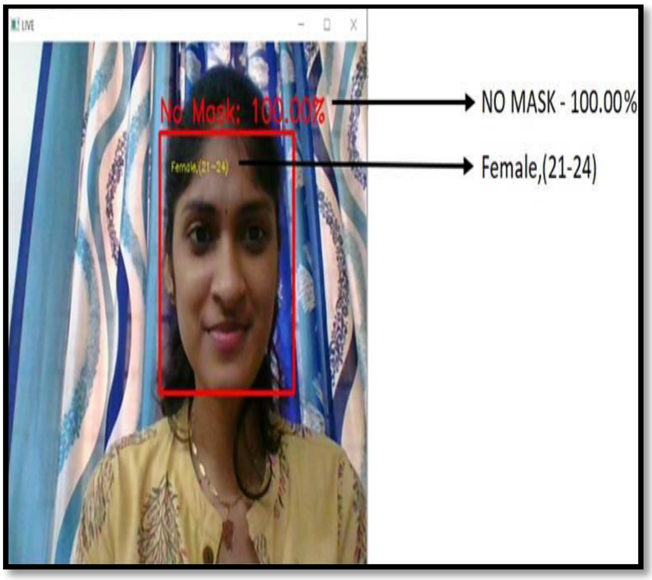

Figure-7. The result shown here is the person without a mask, along with that it displays the age and gender of the person.

The results for the training accuracy for face mask detection is shown in the Figure-8. In our research we have also acquired results for the training accuracy for age and gender detection the form of a graphical representation as shown below. For both age and gender module, the training accuracy is obtained individually. The result displayed below in the Figure-8 is the training accuracy for the face mask detection model where it is represented in the form of graph using matplotlib library. In the below graph, the x-axis defines the number of epochs and y-axis defines the training loss/accuracy. Along with these, the red line present in the graph stands for the training loss/accuracy and the blue line represents the validation loss. As we can observe that there is a slight overfitting, giving an efficient accuracy the graph in Figure-9 presents the exact age accuracy using O.I.U. Audience dataset. The x-axis represents the epoch number in this graph, and the y-axis presents the exact training loss. The red line defines the accuracy of exact training loss, and the blue represents the exact loss of validation. In the case of gender detection, the exact training accuracy is represented graphically.

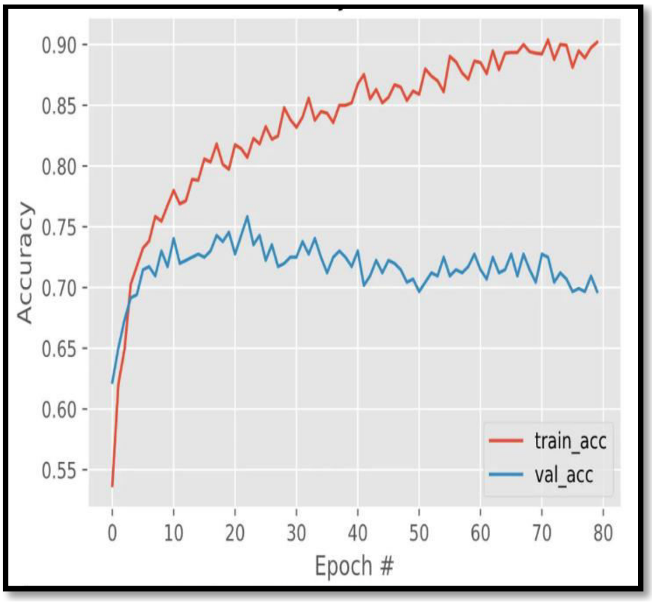

Figure-8. Graphical representation of the training accuracy for face mask detection.

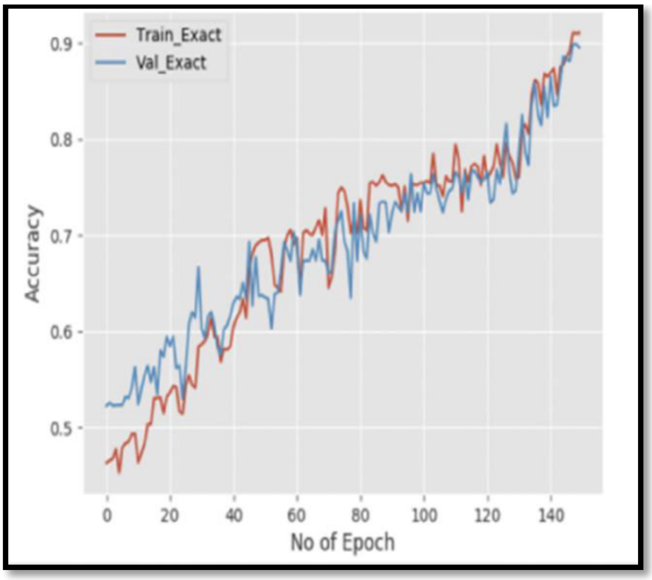

Figure-9. Graphical representation of the training accuracy for age detection.

As shown in Fig**ure-**10, the graph's x-axis is the number of an epoch, and the y-axis contains the training loss. The red line in the graph represents the exact validation loss obtained from the gender detection model using the Adience dataset.

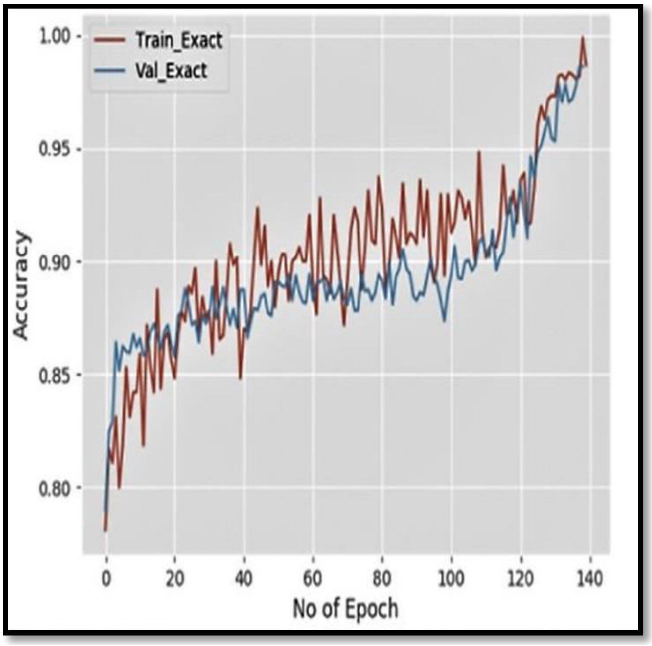

Figure-10. Graphical Representation of the training accuracy for Gender detection.

CONCLUSIONS

In our research, a model was built to determine whether a person has put on a mask properly or not. If the person is not wearing a mask, it detects their age and gender and produces an alarm sound. The engineering of the framework comprises Keras, Tensorflow, and OpenCV. This model's image categorization approaches yielded a positive outcome. To train a model, we considered two separate datasets. The dataset for face mask identification in this research was compiled from various sources and included various photos of faces with and without masks. The dataset used to train the age, and gender detection module is the Adisen Dataset.

We can recognize faces in this model using the Voila Jones technique and Haar-like features. To recognize face masks, this model employs a convolutional neural network architecture. This model allows detecting the age and gender of the person if in case of not wearing the mask properly. As a last reminder, an alert was included whenever a person did not have a mask. By using winsoundly, we have provided an alert in the form of a sound. We employed CNN, which is one of the top image classification systems, to improve accuracy. The idea behind taking up this research was to create awareness among people as most of them have stopped wearing the mask. Wearing a mask not only to protect yourself from the harmful virus but to help yourself get adapted to a healthy lifestyle.

REFERENCES

[1] Das A., Ansari M.W. and Basak R. 2020, December.Covid-19 Face Mask Detection Using Tensor Flow, Keras and OpenCV. In 2020 IEEE 17th India Council International Conference (INDICON) (pp. 1-5). IEEE.

[2] Meena D. and Sharan R. 2016, December. An approach to face detection and recognition. In 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE) (pp. 1-6). IEEE.

[3] Ojala T., Pietikainen M. and Maenpaa T. 2002. Multiresolution grayscale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence. 24(7): 971-987.

[4] Ghosh S., Das N. and Nasipuri M. 2019. Reshaping inputs for convolutional neural network: Some common and uncommon methods. Pattern Recognition. 93, pp.79-94.

[5] Eidinger E., Enbar R. and Hassner T. 2014. Age and gender estimation of unfiltered faces. IEEE Transactions on Information Forensics and Security. 9(12): 2170-2179.

[6] Levi G. and Hassner T. 2015. Age and gender classification using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 34-42).

[7] Mordvintsev A. and Abid K. 2013. Opencv-python tutorials. Obtenido de https://opencvpythontutroals.readthedocs.io/en/latest/ py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc. html.

[8] Das A., Ansari M.W. and Basak R. 2020, December. Covid-19 Face Mask Detection Using Tensor Flow, Keras and OpenCV. In 2020 IEEE 17th India Council International Conference (INDICON) (pp. 1-5). IEEE.

[9] Ari N. and Ustazhanov M. 2014, September. Matplotlib in Python. In 2014 11th International Conference on Electronics, Computer and Computation (ICECCO) (pp. 1-6). IEEE.

[10] Yamashita R., Nishio M., Do R.K.G. and Togashi K. 2018. Convolutional neural networks: an overview and application in radiology. Insights into imaging. 9(4): 611-629.

[11] Arel I., Rose D.C. and Karnowski T.P. 2010. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. IEEE computational intelligence magazine. 5(4): 13-18.

[12] Shukla N. and Fricklas K. 2018. Machine learning with TensorFlow. Shelter Island, Ny: Manning.

[13] Beyeler M. 2017. Machine Learning for OpenCV. Packt Publishing Ltd.

[14] Gulli A. and Pal S. 2017. Deep learning with Keras. Packt Publishing Ltd.

[15] Cengil E., Çınar A. and Özbay E. 2017, October. Image classification with caffe deep learning framework. In 2017 International Conference on Computer Science and Engineering (UBMK) (pp. 440-444). IEEE.

[16] Vander Plas J. 2016. Python data science handbook: Essential tools for working with data. O'Reilly Media, Inc.

[17] Jain A. 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 12: 2825-2830.

[18] Krause I. and Sebastian II, R., Capstone Project: Real Time Musical Audio Detection, Conversion & Transcription.

[19] Ashok G.L.P, P. Saleem Akram, etc. 2020. Implementation of smart home by using packet tracer. International Journal of Scientific and Technology Research. 9(2): 678-685.