An Integrated Facemask Detection with Face Recognition and Alert System using MobileNetV2

Gopinath Pranav Bhargav[1], Kancharla Shridhar Reddy[1], Alekhya Viswanath[1], B Abhi Teja[1] and Akshara Preethy Byju[1]

[1] Amrita University, Kollam, Kerala, India - 690546

Abstract. COVID-19 pandemic has impacted the lives of individuals, organizations, markets, and the whole world in a way that has changed the functioning of all the systems. To get going, some try to adapt to working online, children started studying online and people started ordering food online. While this is still going on, there are many people whose jobs demand physical presence at workplaces and they have no choice but to be exposed to the virus while keeping our society functioning. People are needed to adapt to the new "normal" by practicing social distancing and wearing masks. Wearing masks is the most effective means of prevention of Covid-19. To ensure this, we built a web application that aims at keeping people advised to wear masks constantly with the help of an integrated facemask detection and face recognition system. The proposed system initially detects whether the person in the real-time video feed is wearing a mask or not and then recognizes the face of the person if they are not wearing a mask. Finally, the proposed system alerts that specific violator to wear a mask through an auto-generated email to his personal email id. The application also allows the admin and the violators to log in and access the list of fines levied along with photo evidence.

Keywords: Facemask detection, Face recognition, Covid-19, Data Augmentation.

1 Introduction

In 2020, Covid-19 has been declared a global pandemic and it mainly spreads through physical contact by means of an infected person's saliva, respiratory droplets, etc [1][2]. A healthy person is infected when these droplets are inhaled or come in contact with the person's eyes, nose, or mouth. As a precautionary measure to contain the spread of the virus, lockdowns were imposed throughout the world. Due to this educational institutions, universities and workplaces had to shut down. Students and employees could not attend their colleges or workplaces. This had a pretty drastic effect on the country's economy. People had a hard time working from home. The World Health Organisation (WHO) suggested wearing masks at all times and practicing social distancing as the most effective measures of staying safe against infection.

Recently, vaccination drives have begun and several people are being vaccinated. Gradual efforts are being made worldwide to remove lockdown. Very soon institutions will be allowed to function at their normal pace and capacity. This means many people will come in contact with each other in such places, every day. Even now, there is a need to follow safety protocols, since the risk of being infected by the virus is still there due to the advent of multiple variants of CoronaVirus [3]. Someone has to make sure that, even after places like universities, schools, offices, etc. are opened, the people in it still follow the safety measures.

To make sure that everyone in a workplace or any institution follows safety protocols at all times, a team of people might have to monitor them continuously. For these people to be there at all times, to keep check of safety measures is very difficult and it further puts them at the risk of infection. To address this problem, our proposed system can be deployed and used in places where Closed-Circuit-Television (CCTV) cameras are installed. Fortunately, most organizations have cameras installed in many different locations in their infrastructure. Our proposed system detects people without masks via surveillance camera so if a person is found not wearing a facemask, their image is captured and stored in the database. Facial recognition is run on the image to identify the person. If the person's details are present in the organization's database, he or she will be identified and an email is sent to the registered email id of the identified person by the proposed system automatically, stating that they have violated the safety protocol by not wearing a mask.

The proposed system consists of two models: i) a Face Mask Detection Model and ii) a Face Recognition Model. In the proposed system, these two models are integrated together to identify the people who are not wearing the masks and alert them directly without human intervention. The Face Mask Detection Model, checks for a facemask in an image or live video stream from surveillance cameras using transfer learning in neural networks. It is then trained using a Convolutional Neural Network (CNN). The dataset used for face mask training consists of 3918 images of real people's faces.

The second model is the Face Recognition Model, this model as the name suggests is used to identify faces with the help of an amazingly simple Python library, which is called "face_recognition". This library is built using dlib's state-of-the-art face recognition which again employs Convolutional Neural Network (CNN). Dlib is a modern toolkit containing several machine learning algorithms and it is coded in C++. The face-recognition library is built with deep learning and has been trained to obtain an accuracy of 99.38% on the Labeled Faces in the Wild (LFW) benchmark. Labeled Faces in the Wild (LFW) [4] is a database created by the University of Massachusetts, the database was created for studying unconstrained face recognition. The database consists of 13,233 images of 5,749 different people.

2 Related Works

In recent times, there has been a lot of Covid-19 related research in all the fields like biotechnology [5], data science [6], etc, suggesting various solutions to tackle this pandemic and for being ready for any such future adversities. Among them, one of the most researched areas is developing novel methodologies for facemask detection In several countries, as people are adjusting to the new normal after lifting the lockdowns, wearing masks is mandated in public places. It is a herculean task to physically achieve the monitoring of people wearing masks by security personnel. This task also puts them at risk of infection. Hence, recent progressive studies and optimization techniques have been researched and published to help face this problem.

In [7], a face mask detection model is proposed that uses MobileNetV2 and single shot multibox detector (SSD) as a framework for classification that achieves a good accuracy of 0.9264. This model also employs Open-CV DNN for face detection. Their work also provides a contextual understanding of the convolution layer, pooling layer, linear bottleneck. In [8], a mask detection system was proposed that runs in real-time to detect faces and checks if they are wearing face masks using optimistic CNN. This model detects faces from an image and determines if it should be classified as 'no_mask' or 'mask' using Machine Learning (ML) libraries like Keras, TensorFlow, Scikit-Learn, and OpenCV. The training of the model was done using a Simulated Masked Face Dataset (SMFD). It also makes use of image augmentation technique training and testing for when there exists a problem of restricted data.

In [9], a hybrid model built based on classical and deep machine learning consists of two components, one for feature extraction and the other is for classification processing using 3 approaches. For feature extraction, Resnet50 was used and for classification processing of mask approaches like i) support vector machine (SVM) algorithms, ii) decision tree, and iii) ensemble were used. Simulated Masked Face Dataset (SMFD) for SVM algorithms, Labeled Faces in the Wild (LFW) for and Real-World Masked Face Dataset (RMFD) are the three datasets used. SVM learning algorithm achieved 99.49% accuracy in SMFD. RMFD [10] achieved 99.64% of accuracy, LFW[4] achieved 100% of testing accuracy. In [11] the model was built by fine-tuning the pre-trained state-of-the-art deep learning model, InceptionV3 [12]. A simulated Masked Face Dataset (SMFD) was used to train and test the InceptionV3 model. For better training and testing, the Image augmentation technique was also implemented to meet the limited availability of data. This model achieved an accuracy of 99.9% during training and 100% during testing as claimed by the authors.

3 Methodology

In a broader abstraction, there are four steps in the implementation of our proposed system. i) Detecting faces from the live video feed, ii) Facemask detection, iii) Face recognition, and iv) sending alerts to the violator and admin, as shown in Fig.1. Firstly, the video feed is obtained using OpenCV modules. Next, faces need to be detected. A face detection model called faceNet was used to accommodate face detection. These detected faces, i.e, the face locations are then used by the Facemask detection model to detect whether the face has a mask or not. For the Facemask detection model, we used libraries like Tensorflow, Keras and made use of CNN. The dataset is split into a training set and a test set (validation set) with the training set being 80% and the rest for the testing set. Tensorflow and Keras were used for the preprocessing and training of the model. For training the model, we used

MobileNetV2, which is a Convolution Neural Networks architecture. MobilenetV2 [13] works as an inverted residual structure where the residual connections are between the bottleneck layers. After training the model, face detection and facemask detection are implemented on the real-time feed, and based on the confidence, the probability percentage for the existence of the mask in the frame is displayed.

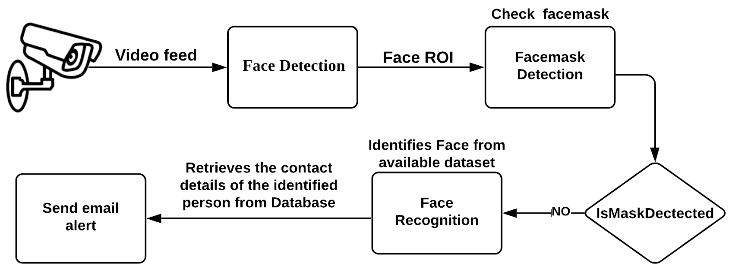

Fig. 1. The architecture of the proposed system.

In the Face Recognition model, we have adopted the face_recognition module from the python library. The library is built on dlib's state-of-the-art face recognition. Though the library is originally written in C++, it has easy-to-use Python bindings. The responsibility of the Face Recognition model involves the conversion of the detected face's training images into image encodings, and then they are stored in an encodings_list. Then it converts the 'to be identified' image into image encoding, then using the face_recognision library, we compare the 'to be identified' image encoding with encodings in the encoding list. By comparing the encodings we can thus identify the person in the image.

When the people in the live feed are detected without the mask, i.e, the violators, it is the job of the Face Recognition model to identify and recognize the violator's face. If the violator's image was used in the training set while training the model, the person's name is returned. This name is used to run MySQL query to retrieve the contact information of the person like Roll no, in case of University, or Employee ID, in case of corporate usage, etc, and Email Id. A mail is sent to this email id along with the photo for proof to the violator. If the face of the violators is unknown, then a notification email is sent to the administrator's email id along with the photo. These details can also be checked by the admin on the application's web page, through admin login.
As we discussed, the proposed system consists of two main modules: i) The Face Mask Detection Model and ii) The Face Recognition Model.

3.1 Facemask Detection Model

The Facemask detection model (Fig.2) mainly has four steps:

1. **Dataset and pre-processing:** The dataset used, curated from few sources like Kaggle.com, google images etc, consists of 2 sub-datasets. One is "with_mask" which is a dataset of 1915 images of people with masks and the other is "without_mask" containing 1918 images of people without wearing

masks. A sample of both datasets is shown in Fig. 3. During the preprocessing phase, all the images are cropped to dimensions (224,224) for uniformity. The images are then converted into numeric encodings and set as NumPy arrays for ease of use and for faster calculation. Labels are also one-hot-encoded.

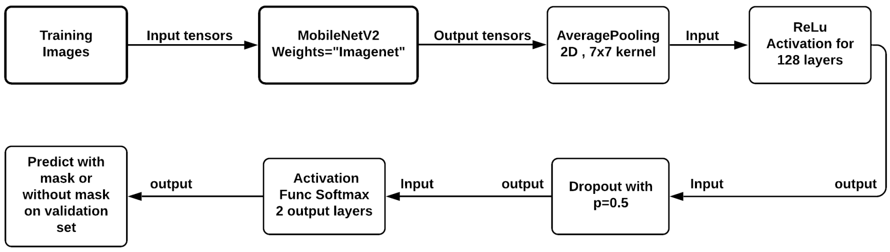

Fig. 2. The architecture of the Facemask detection model

Fig. 3. Sample Images of "with_mask" and "without_mask" datasets

2. **Data augmentation:** Data augmentation is a very beneficial tool in this regard and can be used to augment already present images to solve this issue. In this technique, methods like rotation, flipping, zooming, shifting, shearing, etc. are applied to the images. So, in short, the images from the dataset are used for generating numerous versions of similar images. In this model, ImageDataGenerator from Keras is used for the data augmentation process. Samples of the augmented or generated mages from our dataset are shown in Fig. 4.

Fig. 4. A sample of augmented images from the dataset

3. **Training and testing the model:** The dataset is split into training and testing sets in the ratio of 0.8 i.e, 80% of the data is training set and the rest is testing set. The testing set is used for validation. With the use of data augmentation, the total no of images is also increased aiding in training the model better. Training the model is done in two steps. Training a head model and a base model. The Head model is trained with the help of the weights of the “ImageNet” database instead of starting from null. ImageNet is a large dataset consisting of over 14 million labeled images belonging to more than 20,000 classes. Hence, ImageNet weights are widely used for transfer learning models that are based on image classification. After the base model, a head model is trained on top of the base model.

   For the head model, there are a few notable layers involved like convolution layer, pooling layer, dropout layer, and non-linear layer. The convolution layer, which is the fundamental layer of CNN, is used for feature extraction and it uses sliding window techniques for the generation of feature maps. For the pooling layer, we used average pooling, which takes the average of all the values that are currently in the region under computation and takes this value as the output for the matrix value of that cell. Then a dropout layer is used for reducing the overfitting of the model. We specify a ratio for dropout which defines the likelihood for a neuron to be dropped. A nonlinear function such as Rectified Linear Unit (ReLU) is applied to 128 hidden layers and then Softmax activation function is applied to 2 output layers.

4. **Implementing the model:** After successfully training, testing, and evaluating the accuracy of the model, we move to implement it on the live data i.e, video stream. We use OpenCV to capture live video frames and detect the face in the frame with the help of a face detection module. Now, the frames of the detected faces are fed to the Facemask detection model to classify the frame as either “with mask” or “without mask”.

3.2 The Face recognition model

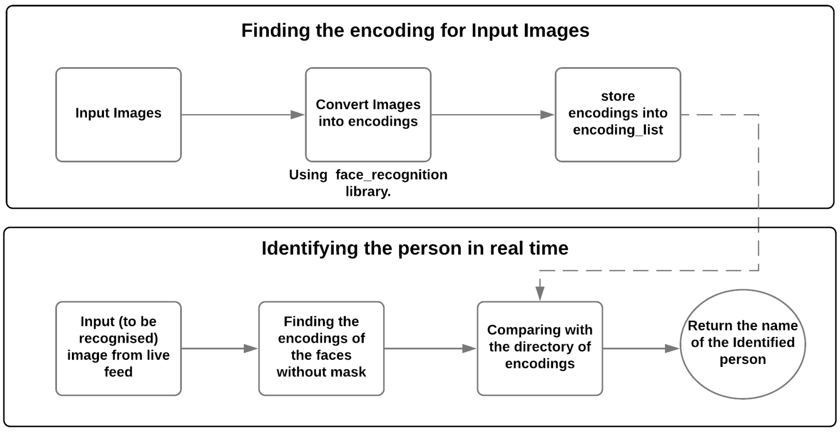

Fig. 5. Face recognition architecture

1. **Create training dataset:** If the model is to be trained to recognize all the people in an institution then one picture of each person with the name of the image labeled as their actual name is to be used to create the dataset as shown in Fig. 6.

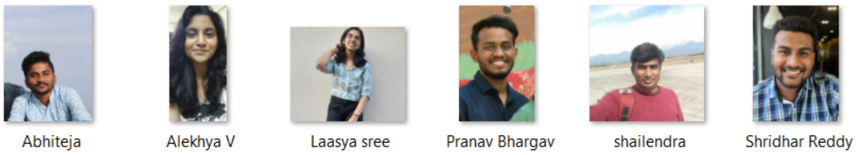

Fig. 6. Image dataset for Face Recognition

2. **Converting the image to encodings:** Once the image dataset is loaded, then, the images are converted to encodings using the 'face_recoginition' library. Each encoding is 128-dimension face encoding for each face in the image. These encodings are stored in an encode_list. With this training, the part is over.

3. **Implementing the model:** To recognize a face from an image, first it has to be converted to encoding and then use 'face_recognition.compare_faces' to identify the person. As claimed in the documentation of the face recognition library it has an accuracy of 99.38% on the Labeled Faces in the Wild (LFW) benchmark.

3.3 Database

The images captured during the face mask detection process need to be stored, thus we need to create a database. The details of all the users also need to be maintained in order to send them an email when they are detected not wearing the mask. To update the database in real-time we used python and PHP to connect to the database and add details.

3.4 Web Application

Our web application has a very intuitive design and accommodates login and sign-up facilities for users and admin. This application helps admins to monitor the violators and allows users to check their fine details.

4 Results

The MobileNetV2 model, with pre-trained weights of "ImageNet" and trained on the facemask dataset is evaluated based on the results of the validation set. The evaluation metrics used are Accuracy and loss of training and validation sets (Fig.7), classification report (Fig.8), and confusion matrix (Fig.9).

The graph (in Fig.7), shows the accuracies and losses of training and validation set with respect to epochs. During the first 5 epochs, training loss and validation loss were quite high and they progressively reduced. After 10 epochs values started becoming more stable. Accuracy increased from 82.7 % to 98% in the first four epochs and after 12 epochs, the accuracy gained from 99.3% to 99.44% at the end of 20 epochs

Fig. 7. Training & validation losses and accuracies

The classification report in Fig.8 shows the values of precision, recall, f1- score, accuracy, etc. Precision is the ratio of True Positives to the sum of True Positives and False Positives. The model gets a precision of 0.98 for predicting "with_mask" and a precision of 1.00 for "without_mask". Similarly, Recall is the ratio of True Positives to the sum of True Positives and False Negatives. The model gets a recall value of

1.00 for “with_mask” and 0.98 for “without_mask”. F1-score is a weighted harmonic mean of recall and precision. It is used to check if we are correctly identifying real threats and are not disturbed by false alarms. F1-score for both “with_mask” and ‘without_mask” is 0.99 which is a very good score.

	precision	recall	f1-score	support
with_mask	0.98	1.00	0.99	383
without_mask	1.00	0.98	0.99	384
accuracy			0.99	767
macro avg	0.99	0.99	0.99	767
weighted avg	0.99	0.99	0.99	767

Fig. 8. Classification Report

The confusion matrix shown in Fig.9, is plotted with the help of a heatmap from the seaborn library to represent the 2D matrix data. The model successfully identified 380+ True positives and 380+ true negatives. 1 image is false positives and 6 images are False negatives.

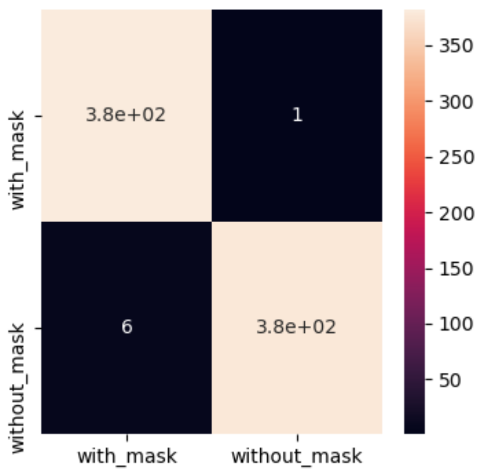

Fig. 9. Confusion Matrix

5 Conclusion

The Covid-19 pandemic has really changed the way we approach our daily jobs and tasks. Wearing masks is a mandatory safety protocol. To ensure the safety of the people of an organization, university, hospitals, and various workplaces which need the admission of certain people on a regular basis, we have proposed a system to automatically detect people not wearing masks and notify them and also the administrator of the organizations with the help of emails. We used Computer Vision, MobileNetV2, Myphpadmin, and face recognition to monitor, detect, store and alert people to help ensure that they wear masks all the time. The performance of our proposed system with 99.44% accuracy clearly outperforms most of the existing works. The proposed system can also be implemented in public establishments, airports, railway stations, etc.

6 References

1. Rahmani, A. and Mirmahaleh, S., 2021. Coronavirus disease (COVID-19) prevention and treatment methods and effective parameters: A systematic literature review. *Sustainable Cities and Society*, 64, p.102568.
2. Leung, N., Chu, D., Shiu, E., Chan, K., McDevitt, J., Hau, B., Yen, H., Li, Y., Ip, D., Peiris, J., Seto, W., Leung, G., Milton, D. and Cowling, B., 2020. Respiratory virus shedding in exhaled breath and efficacy of face masks. *Nature Medicine*, 26(5), pp.676-680.
3. Koyama, T., Weeraratne, D., Snowdon, J. and Parida, L., 2020. Emergence of Drift Variants That May Affect COVID-19 Vaccine Development and Antibody Treatment. *Pathogens*, 9(5), p.324.
4. Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, University of Massachusetts, Amherst, Technical Report 07-49, October 2007.
5. Shohag, M., Khan, F., Tang, L., Wei, Y., He, Z. and Yang, X., 2021. COVID-19 Crisis: How Can Plant Biotechnology Help?. *Plants*, 10(2), p.352.
6. Teodoro Alamo, D. G. Reina, Pablo Millán, “Data-Driven Methods to Monitor, Model, Forecast and Control Covid-19 Pandemic: Leveraging Data Science, Epidemiology and Control Theory”, arXiv:2006.01731 [q-bio.PE].
7. Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P. and Hemanth, J., 2021. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustainable Cities and Society*, 66, p.102692.
8. Suresh, K., Palangappa, M. and Bhuvan, S., 2021. Face Mask Detection by using Optimistic Convolutional Neural Network. *2021 6th International Conference on Inventive Computation Technologies (ICICT)*,.
9. Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, Nour Eldeen M. Khalifa, A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic, Measurement, Volume 167, 2021, 108288, ISSN 0263-2241, https://doi.org/10.1016/j.measurement.2020.108288.
10. RMFD: Z. Wang, et al., Masked face recognition dataset and application, arXiv preprint arXiv:2003.09093, 2020
11. Jignesh Chowdary G., Punn N.S., Sonbhadra S.K., Agarwal S. (2020) Face Mask Detection Using Transfer Learning of InceptionV3. In: Bellatreche L., Goyal V., Fujita H., Mondal A., Reddy P.K. (eds) Big Data Analytics. BDA 2020. Lecture Notes in Computer Science, vol 12581. Springer, Cham. https://doi.org/10.1007/978-3-030-66665-1_6

12. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, Zbigniew Wojna: Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826.
13. Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen: MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520