Face Mask Detection

Shubharthak Sangharasha
CSE-AIML
Chandigarh University
Delhi, India
20bcs6872@cuchd.in

Aashish Kalra
CSE-AIML
Chandigarh University
Rajasthan, India
20bcs6852@cuchd.in

Harshil Jain
CSE-AIML
Chandigarh University
Faridabad, India
20bcs6877@cuchd.in

Abstract—**The novel Coronavirus had brought a new normal life in which the social distance and wearing of face masks plays a vital role in controlling the spread of the virus. But most people are not wearing face masks in public places which increases the spread of viruses. Hence to avoid such situations we have to scrutinize and make people aware of wearing face masks. Humans cannot be involved in this process, due to the chance of getting affected by corona. Hence here comes the need for artificial intelligence (AI), which is the main theme of our project. Our project involves the identification of persons wearing face masks and not wearing face masks in public places through using image processing or OpenCV and other AI techniques for giving alerts through our UI to that person that they need to wear a mask. The object detection algorithms are used for the identification of persons with and without wearing face masks.**

Our Research Paper is proposing:- Detection of the Face, Also accepts multiple Faces. Check whether the person is wearing a face mask or not. If a person is wearing a face mask, then show a green box else raise the alarm. If a person is not wearing a face mask for more than 30 seconds then send the person's ROI to the owner's mail.

Keywords—Coronavirus, Covid-19, Machine Learning, Face Mask Detection, Convolutional Neural Network, TensorFlow, OpenCV

I. Introduction

The novel coronavirus covid-19 had brought a new normal life. 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 a 0.7million deaths [1]. 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 are one of them. Elder people having lung disease can possess serious complications from COVID-19 illness as they appear to be at higher risk [2]. Some common human coronaviruses that infect public around the world are 229E, HKU1, OC43, and NL63. Before debilitating individuals, viruses like 2019-nCoV, SARS-CoV, and MERS-CoV infect animals and evolve into human coronaviruses [3]. Persons having respiratory problems can expose anyone (who is in close contact with them) to infective beads. Surroundings of a tainted individual can cause contact transmission as droplets carrying the virus may withal arrive on adjacent surfaces [4]. India was struggling to get out of this virus attack and the government implemented a lockdown for the long way. Lockdown placed pressure on the global economy. So, the government gave relaxations in lockdown. WHO declared that maintaining social distance and wearing a mask is necessary. Potential points of interest of the utilization of masks lie in reducing the vulnerability of risk from a noxious individual during the "pre-symptomatic" period and stigmatization of discrete persons putting on masks to restrain the spread of the virus. WHO stresses prioritizing medical masks and respirators for health care assistants[4]. The biggest support that the government needs after relaxation is social distancing and wearing of masks by the people. But many people are getting out without a face mask, this may lead to the increase in the spread of the covid-19 virus in the country. According to Economic Times, India has stated that "90 per cent Indians are aware, but only 44 per cent are wearing a mask". This survey points out that people are aware but they are not wearing the mask due to some discomfort in wearing and carelessness. This may result in the easy spreading of covid-19 in public places. So, it is important to make people wear masks in public places. In densely populated regions it is difficult to find the persons not wearing the face mask and warn them. Hence, we are using image processing techniques for the identification of persons wearing and not wearing face masks. The real-time images from the camera are compared with the trained dataset and detection of wearing or not wearing a mask is done. The trained dataset is made by using the machine learning technique which is the deciding factor of the result. The algorithm created using a trained dataset will find the persons with and without wearing face masks.

Face mask detection involves detecting the location of the face and then determining whether it has a mask on it or not. The issue is approximately 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

II. Ease of Use and related work

The System which we are proposing we are trying to eliminate the cons of the existing system by making it more beneficial and reliable.

In the current system, the accuracy of the system is very low and inefficient and even it doesn't guide or warn our user to wear the mask, in our project our UI will guide the user to wear the mask and even our program will send the mail to the authorised committee which will tell that this person is not wearing any mask by sending the picture as well.

The System which we are proposing needs very low maintenance and it is automated, we just need to run the program and then it will handle everything else on its own and by this, our work will be done with the least effort and effectively.

We also added a feature that we can add a video or an image and our program can tell which person is wearing a mask and which person is not wearing a mask with an accuracy percentage. We are also trying to add some other warning devices which will warn the person that he/she is not wearing a mask like a beeping device or a LED which will turn Red when a person is not wearing any mask.

If someone tries to get inside without wearing any mask then the machine will take the picture of that person and send it to the authorized committee in real-time and will start beeping and our UI will tell him/her that they need to wear the mask to control the spreading of Coronavirus.

In this system, a face is detected from an image that has several attributes in it. According to [21], 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 images impeded by some other thing not confronting the camera, and so forth. Authors in [22] think occlusive face detection comes with two major challenges: 1) unavailability of sizably voluminous datasets containing both masked and unmasked faces, and 2) 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 [11], a convolutional neural network (CNN) 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 inhibition.

Here the main challenge of the task is to detect the face from the image correctly and then identify if it has a mask on it or not. To perform surveillance tasks, the proposed method should also detect a face along with a mask in motion

As we previously stated that for now, our project is only software-based but for future purposes, our project will use a camera to identify and recognise faces and their face mask, and we are also using a UI guidance system, which will necessitate the use of a speaker. We'd also need a device to maintain all these things and where our software will be loaded; an ideal choice would be a Raspberry Pi or an Arduino Board, but as we're simply submitting software, we won't be utilizing one. Also, our programme is OS independent as long as all dependencies are loaded. We developed it on our personal computers. We've also tested on a variety of operating systems, including Windows, Ubuntu, etc.

For the software, the most basic requirement is python, as the entire software is built on it. We would also require pip to install additional packages. A large number of Python modules would be required like TensorFlow and Keras Modules for face building modules and making predictions. Google Text to Speech (gTTS), Computer Vision (cv2), and Smtplib Module for sending Emails to the authorized committee. The most fundamental need is Github. GitHub is a version management and collaboration tool for programming. For receiving the response from the owner, we will use the web framework Flask, which is developed in Python and we are also using .caffeemodel and .prototxt file for face detection and we'll also need a dataset for training our data. We have used different text editors and IDEs like spyder, emacs, vscode, jupyter-notebooks for writing code.

In the current system for face mask detection, the efficiency is very low if a person is wearing a mask on the mouth but not covering their nose then also it doesn't warn that person but, in our project, if the efficiency is less than 75% our UI will start warning that person that they need to wear the mask properly and if he/she still doesn't wear mask properly then we will click the photo of that person and send the authorized committee for taking the further actions.

Furthermore, there is no automation for doing all the things adjacently, kind of, there was always a need for a person to maintain all these features adjacently and there are many places where committee assigns a person for this work only to report them who is wearing the mask and who is not which will increase the handwork and that person need to do everything on his own like warning people that they need to wear the mask and all so our project is solving every problem we might face in the future as well. There is one more problem that the existing system is facing, i.e., the real-time pictures and the reporting of that pictures on time, the existing program takes at least 10-15 minutes for reporting that picture and in the meantime, that person might leave or wears a mask so that is also a concerning problem that we need to warn them in the real-time. The existing systems are controlled manually.

Also, existing systems are very complex and hard to maintain like we need to fetch details of every minute from the existing project and then we can proceed, we also face these problems in our existing system–

- No Automated facial recognition.
- No real-time information gathering and sending to the authorized committee.

III. METHODOLOGY

Two datasets have been used for experimenting with the current method. Dataset 1 [16] consists of 1376 images in which 690 images with people wearing face masks and the rest 686 images with people who do not wear face masks. Fig. 1 mostly contains a front face pose with a single face in the frame and with some type of mask having white colour only.

Fig. 1. Samples from Dataset 1 including faces without masks and with masks

Dataset 2 from Kaggle [17] consists of 853 images and its countenances are clarified either with a mask or without a mask. In fig. 2 some face collections are head turn, tilt and slant with multiple faces in the frame and different types of masks having different colors as well.

Fig. 2. Samples from Dataset 2 including faces without masks and with masks

INCORPORATED PACKAGES

A. TensorFlow

TensorFlow, an interface for expressing machine learning algorithms, is utilized for implementing ML systems into fab- rication over a bunch of areas of computer science, including sentiment analysis, voice recognition, geographic information extraction, computer vision, text summarization, information retrieval, computational drug discovery and flaw detection to pursue research [18]. In the proposed model, the whole Sequential CNN architecture (consists of several layers) uses TensorFlow at the backend. It is also used to reshape the data (image) in the data processing.

B. Keras

Keras gives fundamental reflections and building units for the creation and transportation of ML arrangements with high iteration velocity. It takes full advantage of the scalability and cross-platform capabilities of TensorFlow. The core data structures of Keras are layers and models [19]. All the layers used in the CNN model are implemented using Keras. Along with the conversion of the class vector to the binary class matrix in data processing, it helps to compile the overall model.

C. OpenCV

OpenCV (Open Source Computer Vision Library), an open-source computer vision and ML software library, is utilized to differentiate and recognize faces, recognize objects, group movements in recordings, trace progressive modules, follow eye gestures, track camera actions, expel red eyes from pictures taken utilizing flash, find comparative pictures from an image database, perceive the landscape and set up markers to overlay it with increased reality and so forth [20]. The proposed method makes use of these features of OpenCV in resizing and colour conversion of data images.

IV. The Proposed Method

A. Data Processing

Data preprocessing involves the conversion of data from a given format to a much more user-friendly, desired and meaningful format. It can be in any form like tables, images, videos, graphs, etc. This organized information fit in with an information model or composition and captures the relationship between different entities [6]. The proposed method deals with image and video data using Numpy and OpenCV

a) *Data Visualization:* Data visualization is the process of transforming abstract data into meaningful representations us-ing knowledge communication and insight discovery through encodings. It is helpful to study a particular pattern in the dataset [7].

The total number of images in the dataset is visualized in both categories – 'with mask' and 'without a mask'.

The statement *categories=os.listdir(data path)* categorizes the list of directories in the specified data path. The variable *categories* now look like: ['with mask', 'without a mask']

Then to find the number of labels, we need to distinguish those categories using *labels=[i for i in range(len(categories))]*. It sets the labels as: [0, 1]

Now, each category is mapped to its respective label using *label dict=dict(zip(categories,labels))* which at first returns an iterator of tuples in the form of zip object where the items in each passed iterator are paired together consequently. The mapped variable *label dict* looks like: *{*'with mask': 0, 'without a mask': 1*}*

b) Conversion of RGB image to Gray image: Modern descriptor-based image recognition systems regularly work on grayscale images, without elaborating the method used to convert from color-to-grayscale. This is because the color-to-grayscale method is of little consequence when using robust descriptors. Introducing nonessential information could increase the size of training data required to achieve good performance. As grayscale rationalizes the algorithm and diminishes the computational requisites, it is utilized for extracting descriptors instead of working on color images instantaneously [8].

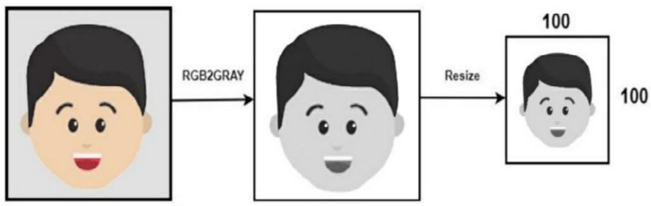


Fig. 3. Conversion of an RGB image to a Gray Scale image of 100 x 100 size

We use the function *cv2.cvtColor(input image, flag)* for changing the color space. Here the flag determines the type of conversion [9]. In this case, the flag *cv2.COLOR BGR2GRAY* is used for gray conversion.

Deep CNNs require a fixed-size input image. Therefore we need a fixed common size for all the images in the dataset. Using *cv2.resize()* the grayscale image is resized into 100 x 100.

c) Image Reshaping: The input during relegation of an image is a three-dimensional tensor, where each channel has a prominent unique pixel. All the images must have an identically tantamount size corresponding to the 3D feature tensor. However, neither images are customarily coextensive nor their corresponding feature tensors [10]. Most CNNs can only accept fine-tuned images. This engenders several problems throughout- out data collection and implementation of the model. However, reconfiguring the input images before augmenting them into the network can help to surmount this constraint. [11].

The images are normalized to converge the pixel range between 0 and 1. Then they are converted to 4 dimensional arrays using *data=np.reshape(data,(data.shape[0], img size,img size,1))* where 1 indicates the Grayscale image. As, the final layer of the neural network has 2 outputs – with mask and without mask i.e. it has categorical representation, the data is converted to categorical labels.

B. Training of Model

b) Building the model using CNN architecture: CNN has become ascendant in miscellaneous computer vision tasks [12]. The current method makes use of Sequential CNN.

The First Convolution layer is followed by Rectified Linear Unit (ReLU) and MaxPooling layers. The Convolution layer learns from 200 filters. Kernel size is set to 3 x 3 which specifies the height and width of the 2D convolution window. As the model should be aware of the shape of the input expected, the first layer in the model needs to be provided with information about the input shape. Following layers can perform instinctive shape reckoning [13]. In this case, *the input shape* is specified as *data.shape[1:]* which returns the dimensions of the data array from index 1. Default padding is "valid" where the spatial dimensions are sanctioned to truncate and the input volume is non-zero padded. The activation parameter to the Conv2D class is set as "relu". It represents an approximately linear function that possesses all the assets of linear models that can easily be optimized with gradient-descent methods. Considering the performance and generalization in deep learning, it is better compared to other activation functions [14]. Max Pooling is used to reduce the spatial dimensions of the output volume. Pool size is set to 3 x 3 and the resulting output has a shape (number of rows or columns) of: shape of output = (input shape - pool size + 1) / strides), where strides has default value (1,1) [15].

As shown in the figure below, the second Convolution layer has 100 filters and the Kernel size is set to 3 x 3. It is followed by ReLu and MaxPooling layers. To insert the data into CNN, the long vector of input is passed through a Flatten layer which transforms the matrix of features into a vector that can be fed into a fully connected neural network classifier. To reduce overfitting a Dropout layer with a 50% chance of setting inputs to zero is added to the model. Then a Dense layer of 64 neurons with a ReLu activation function is added. The final layer (Dense) with two outputs for two categories uses the Softmax activation function.

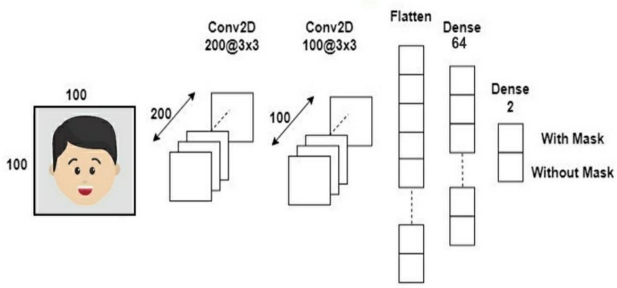


Fig. 4. Convolutional Neural Network architecture

The learning process needs to be configured first with the compile method [13]. Here "adam" optimizer is used. *categorical cross-entropy* which is also known as multiclass log loss is used as a loss function (the objective that the model tries to minimize). As the problem is a classification problem, metrics are set to "accuracy".

c) Splitting the data and training the CNN model: After setting the blueprint to analyze the data, the model needs to be trained using a specific dataset and then to be tested against a different dataset. A proper model and optimized train_test_split help to produce accurate results while making a prediction. The test size is set to 0.1 i.e. 90% data of the dataset undergoes training and the rest 10% goes for testing purposes. The validation loss is monitored using ModelCheck- point. Next, the images in the training set and the test set are fitted to the Sequential model. Here, 20% of the training data is used as validation data. The model is trained for 20 epochs (iterations) which maintains a trade-off between accuracy and chances of overfitting. Fig. 5 depicts a visual representation of the proposed model.

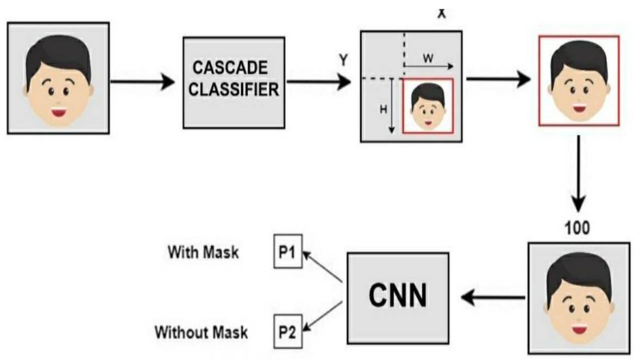


Fig. 5. Overview of the Model

V. Result and analysis

The model is trained, validated and tested upon two datasets. Corresponding to dataset 1, the method attains accuracy up to 95.77% (shown in fig. 7). Fig. 6 depicts how this optimized accuracy mitigates the cost of error. Dataset
2 is more versatile than dataset 1 as it has multiple faces in the frame and different types of masks having different colors as well. Therefore, the model attains an accuracy of 94.58% on dataset 2 as shown in Fig. 9. Fig. 8 depicts the contrast between training and validation loss corresponding to dataset 2. One of the main reasons behind achieving this accuracy lies in *MaxPooling*. It provides rudimentary translation invariance to the internal representation along with the reduction in the number of parameters the model has to learn. This sample-based discretization process down-samples the input representation consisting of an image, by reducing its dimensionality. The number of neurons has the optimized value of 64 which is not too high. A much higher number of neurons and filters can lead to worse performance. The optimized filter values and pool size help to filter out the main portion (face) of the image to detect the existence of the mask correctly without causing over-fitting.

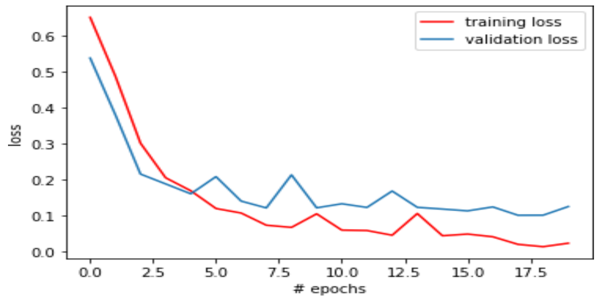


Fig. 6. # epochs vs loss corresponding to the dataset

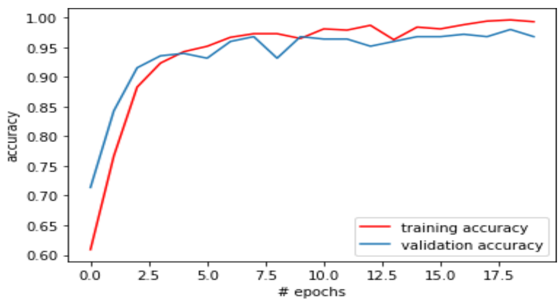


Fig. 7. # epochs vs accuracy corresponding to dataset 1

The system can efficiently detect partially occluded faces either with a mask or hair or hand. It considers the occlusion degree of four regions – nose, mouth, chin and eye to differentiate between an annotated mask or face covered by hand. Therefore, a mask covering the face fully including the nose and chin will only be treated as "with mask" by the model.

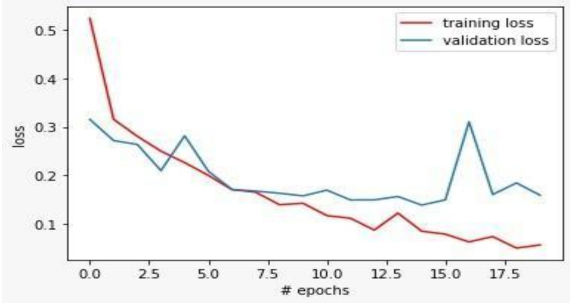


Fig. 8. # epochs vs loss corresponding to dataset 2

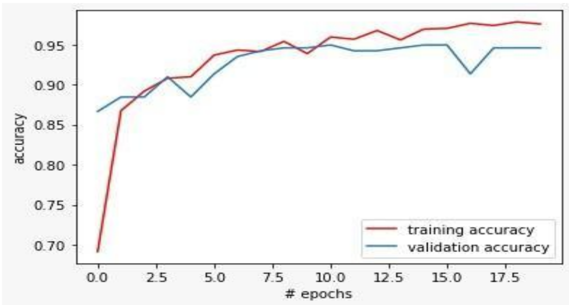
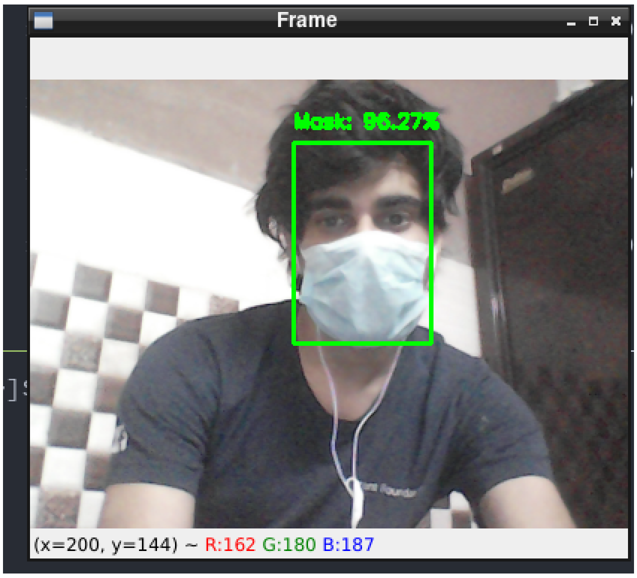


Fig. 9. # epochs vs accuracy corresponding to dataset 2

The main challenges faced by the method mainly comprise varying angles and lack of clarity. Indistinct moving faces in the video stream make it more difficult. However, following the trajectories of several frames of the video helps to create a better decision – "with mask" or "without a mask".

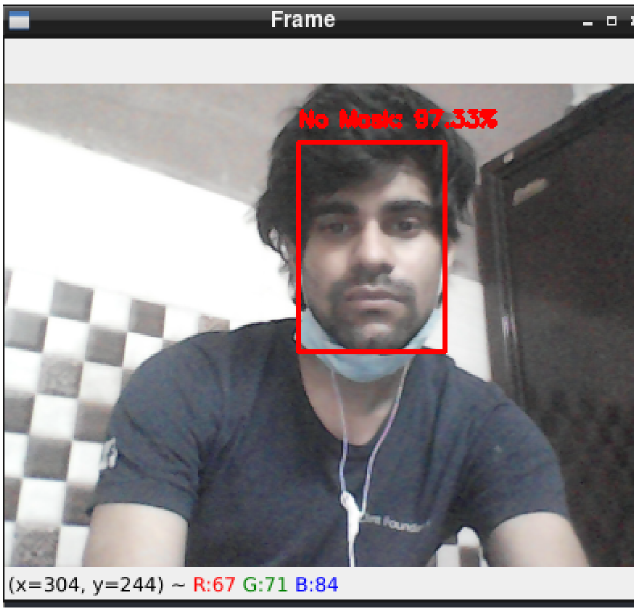
This endeavor necessitates a great deal of new knowledge, such as Deep Learning and AI. We'll also need to learn Flask (a web framework), and we'll have to study a lot of documentation. This is the skill we will need the most in the future, as new technologies emerge daily, and not everything has well-formatted tutorials. As a result, we learnt a crucial skill: learning new skills through documentation. We've used OpenCV to recognise cameras and faces, therefore it's the library we've used the most. We will also use Speech Recognition for taking information like the reason of coming from the visitor at the door. But the main concept here is all about Transfer Learning and Tensorflow implementation. It almost took a lot of time for us to learn and understand the idea and concept. However, Thanks to python for making it simpler for us to understand it more easily.

The result while predicting is as follows

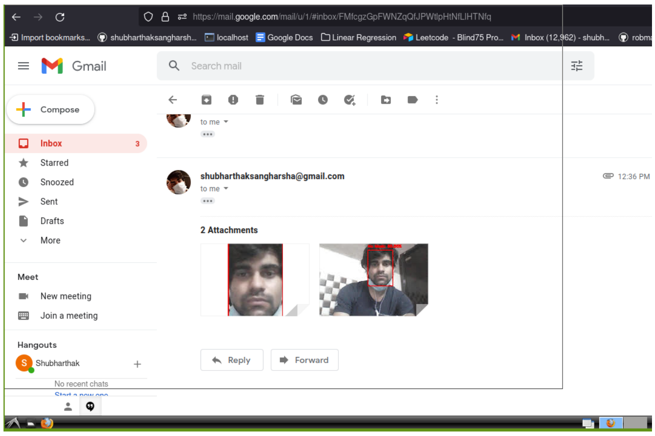


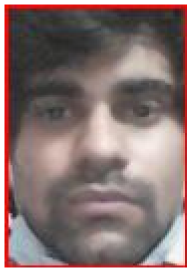
The model is predicting whether the person is wearing a mask or not. In the above picture, The model is predicting the accuracy of wearing the mask and change Region of Interest as green color.

While, in the following image the person is not wearing a mask. The model is predicting the accuracy of not wearing the mask and change region of interest as red color.



If the person is not wearing mask for 30 seconds, then it will capture the picture of the person and send it to the authorized mail.





Here, it is capturing the region of interest and send it to the given authorized id.

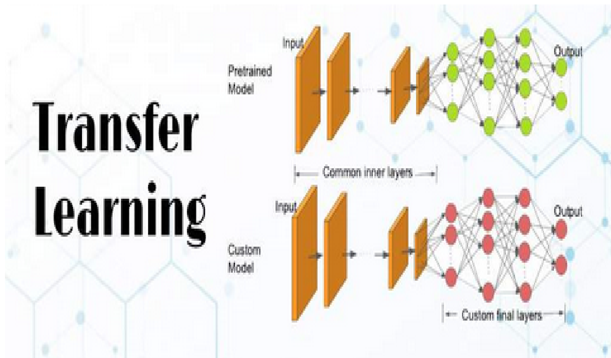


Figure 10: Transfer Learning Diagram

The next most important step here is to determine the accuracy of our model. We have used MobileNETv2 which is the best CNN Architecture.

The accuracy we got using This architecture is well amazing.

See the following confusion matrix.

	precision	recall	f1-score	support
with_mask	0.99	0.99	0.99	433
without_mask	0.99	0.99	0.99	386
accuracy			0.99	819
macro avg	0.99	0.99	0.99	819
weighted avg	0.99	0.99	0.99	819

Figure 11:Confusion matrix

There is no hardware-based functionality. It's similar to a camera in that it can see outside, but it requires an Arduino/ raspberry pi, to do this functionality. However, if one can insert this functionality it can become the best device for awareness of wearing masks in public restaurants/ shops/ universities and even hospitals.

To determine how well our proposed system worked, we must test it on several aspects, including facial mask detection accuracy, cost, time consumption, reliability, and security.

We have done significant advancements from the existing technology, as we make the system more reliable. pocket friendly and have implemented AI in the needed field, so as we know AI is the future so we are trying to solve people's problems using AI and deep learning.

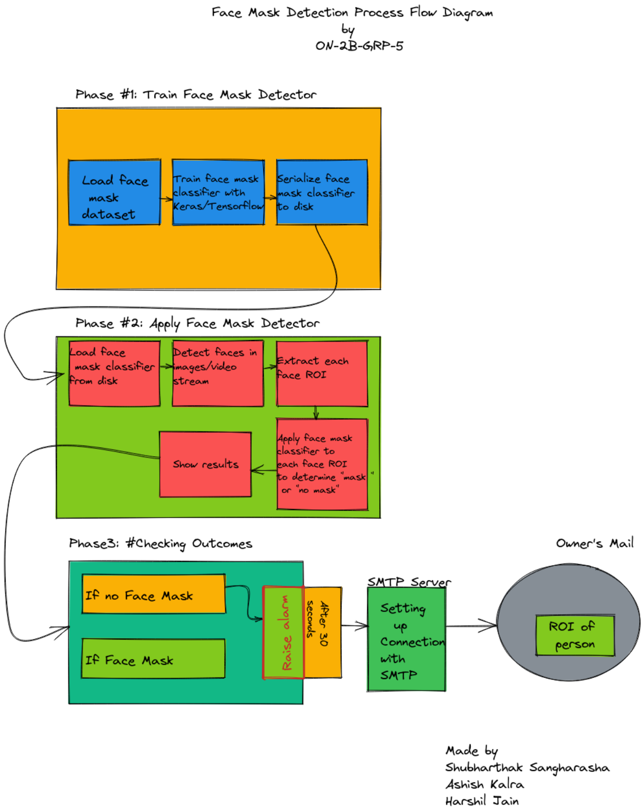


Figure 12: Flow diagram

VI. Conclusions

In this paper, we briefly explained the motivation of the work at first. Then, we illustrated the learning and performance tasks of the model. Using basic ML tools and simplified techniques the method has achieved reasonably high accuracy. It can be used for a variety of applications. Wearing a mask may be obligatory soon, considering the Covid-19 crisis. Many public service providers will ask the customers to wear masks correctly to avail of their services. The deployed model will contribute immensely to the public health care system. In future, it can be extended to detect if a person is wearing the mask properly or not. The model can be further improved to detect if the mask is virus prone or not i.e. the type of the mask is surgical, N95 or not.

References

[1] W.H.O., "Coronavirus disease 2019 (covid-19): situation report, 205". 2020

[2] "Coronavirus Disease 2019 (COVID-19) – Symptoms", Centers for Disease Control and Prevention, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/symptoms- testing/symptoms.html. 2020.

[3] "Coronavirus — Human Coronavirus Types — CDC", Cdc.gov, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/types.html. 2020.

[4] W.H.O., "Advice on the use of masks in the context of COVID-19: interim guidance", 2020.

[5] M. Jiang, X. Fan and H. Yan, "RetinaMask: A Face Mask detector", arXiv.org, 2020. [Online]. Available: https://arxiv.org/abs/2005.03950. 2020.

[6] B. Suvarnamukhi and M. Seshashayee, "Big Data Concepts and Tech- niques in Data Processing", International Journal of Computer Sci- ences and Engineering, vol. 6, no. 10, pp. 712-714, 2018. Available: 10.26438/ijcse/v6i10.712714.

[7] F. Hohman, M. Kahng, R. Pienta and D. H. Chau, "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers," in IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 8, pp. 2674-2693, 1 Aug. 2019, doi: 10.1109/TVCG.2018.2843369.

[8] C. Kanan and G. Cottrell, "Color-to-Grayscale: Does the Method Matter in Image Recognition?", PLoS ONE, vol. 7, no. 1, p. e29740, 2012. Available: 10.1371/journal.pone.0029740.

[9] Opencv-python-tutroals.readthedocs.io. 2020. Changing Colorspaces — Opencv-Python Tutorials 1 Documentation. [online] Available at:https://opencv-python-tutroals.readthedocs.io/en/latest/py tutorials/ py imgproc/py colorspaces/py colorspaces.html. 2020.

[10] M. Hashemi, "Enlarging smaller images before inputting into convo- lutional neural network: zero-padding vs. interpolation", Journal of Big Data, vol. 6, no. 1, 2019. Available: 10.1186/s40537-019-0263-7 . 2020.

[11] S. Ghosh, N. Das and M. Nasipuri, "Reshaping inputs for con- volutional neural network: Some common and uncommon meth- ods", Pattern Recognition, vol. 93, pp. 79-94, 2019. Available: 10.1016/j.patcog.2019.04.009.

[12] R. Yamashita, M. Nishio, R. Do and K. Togashi, "Convolutional neural networks: an overview and application in radiology", Insights into Imaging, vol. 9, no. 4, pp. 611-629, 2018. Available: 10.1007/s13244- 018-0639-9.

[13] "Guide to the Sequential model - Keras Documentation", Faroit.com, 2020. [Online]. Available: https://faroit.com/keras-docs/1.0.1/getting-started/sequential-model-guide/. 2020.

[14] Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S., 2020. Activation Functions: Comparison Of Trends In Practice And Research For Deep Learning. [online] arXiv.org. Available at: https://arxiv.org/abs/1811.03378. 2020.

[15] K. Team, "Keras documentation: MaxPooling2D layer", Keras.io, 2020. [Online]. Available: https://keras.io/api/layers/pooling layers/ max pooling2d/. 2020.

[16] "prajnasb/observations", GitHub, 2020. [Online]. Available: https://github.com/prajnasb/observations/tree/master/experiements/data. 2020.

[17] "Face Mask Detection", Kaggle.com, 2020. [Online]. Available: https://www.kaggle.com/andrewmvd/face-mask-detection. 2020.

[18] "TensorFlow White Papers", TensorFlow, 2020. [Online]. Available: https://www.tensorflow.org/about/bib. 2020.

[19] K. Team, "Keras documentation: About Keras", Keras.io, 2020. [On- line]. Available: https://keras.io/about. 2020.

[20] "OpenCV", Opencv.org, 2020. [Online]. Available: https://opencv.org/. 2020.

[21] D. Meena and R. Sharan, "An approach to face detection and recognition," 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, 2016, pp. 1-6, doi: 10.1109/ICRAIE.2016.7939462.

[22] S. Ge, J. Li, Q. Ye and Z. Luo, "Detecting Masked Faces in the Wild with LLE-CNNs," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 426-434, doi: 10.1109/CVPR.2017.53.