REAL-TIME FACE MASK DETECTOR

Pranav KN[1] , Pramod Mahajan H[2], Praveen V[3], Sai Kiran[4],
Dr. M Vinoth Kumar[5]

[1]BE Student, Department of Information Science and Engineering, Dayananda Sagar Academy of Technology and Management, Karnataka, India

[2]BE Student, Department of Information Science and Engineering, Dayananda Sagar Academy of Technology and Management, Karnataka, India

[3]BE Student, Department of Information Science and Engineering, Dayananda Sagar Academy of Technology and Management, Karnataka, India

[4]BE Student, Department of Information Science and Engineering, Dayananda Sagar Academy of Technology and Management, Karnataka, India

[5] Associate Professor, Department of Information Science and Engineering, Dayananda Sagar Academy of Technology and Management, Karnataka, India

ABSTRACT

The COVID-19 pandemic has quickly influenced our everyday lives, disrupting commerce, and movements around the world. The wearing of a mask to cover the face has become a modern normal. In the near future, several public utility providers will be asking consumers to wear masks appropriately to make use of their facilities. Face mask identification has therefore become a key factor in helping global society. Using basic Machine Learning packages such as TensorFlow, Keras, and OpenCV, this paper offers a simpler approach to achieving this objective. The proposed method correctly recognizes the face from the image and then identifies whether or not it has a mask on it. As a surveillance task officer, a face may also be identified along with a mask in motion. On two separate datasets, the technique achieves accuracy of up to 95.77 percent and 94.58 percent, respectively. We analyze optimum parameter values using the Sequential Convolutional Neural Network model to appropriately detect the existence of masks without overfitting.

Keywords: *Face Mask Detection, TensorFlow, Keras, and OpenCV..*

1. INTRODUCTION

Globally, 178,699,697 confirmed cases of COVID-19 were reported to the WHO as of 14:07 GMT, 19 June 2021, including 3,869,020 deaths [1]. Individuals with COVID19 have reported a broad spectrum of symptoms, from mild signs to extreme illnesses. One of them is respiratory conditions, such as shortness of breath or breathing difficulties. Seniors with pulmonary disease can have considerable complications of COVID-19 because they are more at risk [2]. The 229E, HKU1, OC43 and NL63 are some common human coronaviruses [3] which affect the humankind globally. Viruses like 2019-nCoV, SARS-CoV, and MERS-CoV are infecting animals, evolving into human coronaviruses until they reach humans. Persons with breathing conditions can expose anyone to contagious beads (who is in close contact with them). The environment of a stained individual will cause transmission of the touch, as droplets with the virus can contact their adjacent surfaces [4].

It is necessary to wear a clinical mask to curb such respiratory viral ailments, like COVID-19. The public should be aware of whether the mask should be put on for source control or COVID-19 aversion. Potential points of significance for the use of masks are to reduce the vulnerability of the noxious individual during the "pre-symptomatic" time and to stigmatize discreet people wearing masks to restrict the spread of the virus. WHO emphasizes that health care assistants prioritize surgical masks and respirators [4]. Therefore, in today's global society, face mask identification has become a critical mission.

Face mask detection requires the detection of the position of the face and then the recognition of whether or not it is masked. The problem is abut cognizing the general identification. Face recognition categorically deals with distinguishing a particular group of entities , i.e. face. It has many uses such as autonomous driving, schooling , monitoring, etc [5]. This paper provides a brief overview of the approach to the above motive using the basic packages of Machine Learning(ML) such as TensorFlow, Keras and OpenCV.

The remainder of the paper is structured as follows: Section II discusses related work associated with the detection of face masks. Section III addresses the nature of the dataset used. Section IV presents the details of the packages incorporated to build the proposed model. Section V gives an overview of our method. In section VI, the findings and analysis of the experiments are presented. Section VII concludes and draws the line towards future works.

2. RELATED WORK

In various application domains for object detection and recognition, there have been several advances in machine learning over the years. Most of the works typically focus on the restoration of photographs and facial recognition for identity verification. But the main objective of this work is to identify people who do not wear masks in public places to monitor the further transmission of COVID-19. A face is identified from an image with several attributes during the face detection process. According to [21], face recognition research requires recognition of gesture and poses. Since the size, shape, color, etc. of the faces differ and are not eternal, face recognition is a complicated errand.

Authors in [22] claim that there are two major challenges to occlusive facial detection: 1) the absence of significantly large datasets comprising both masked and unmasked faces and 2) the exclusion of facial expression in the region protected

According to the published work[11], the input picture has size limit by convolutional neural networks (CNNs) in computer vision. The prevalent technique restores the images to overcome the inhibition before they are inserted into the network. Here, the main challenge is to identify the face properly and then determine whether or not it is masked. The proposed method could also detect a face alongside a mask in motion to carry out surveillance activities. Nag and others built a face recognition based door access control in the IoT area[4]. The OpenCV functionality is used to detect and classify the faces of known people automatically and thus to monitor door access. P. Hu proposed a fog computing-based face detection and recognition system that aims to offload the face recognition task from the cloud to fog nodes[5].

The Principal Component Analysis (PCA) algorithm was introduced by Md.Sabbir Ejaz et al.[6] for masked and un-masked face recognition. It was noted that PCA is effective with an accuracy of 96.25 percent in recognizing faces without a mask, but its accuracy is reduced to 68.75 percent in identifying faces with a mask. Li et al.[7] used YOLOv3 for face recognition, which is focused on the darknet-19 deep learning network architecture, in which the WIDER FACE and Celebi databases were used for training, and the FDDB database was used later for evaluation. An accuracy of 93.9% was achieved by this model. Few other works are designed to distinguish individuals with or without a face mask. Two types of facial images have been considered for the training of the model by these methods: with mask and without a mask. In terms of the framework and model chosen to construct the model, the built systems differ.

Based on the above context, it is clear that very limited numbers of research papers have been published to date, particularly for mask detection, while further improvements on existing methods are needed. Therefore, we propose a method to achieve this objective by using some simple Machine Learning packages such as TensorFlow, Keras, OpenCV, in order to contribute to further improvements in face mask recognition in the battle against COVID-19.

3. DATASET

To experiment with the current system, two datasets were used. Dataset 1[16] consists of 1376 images containing 690 images of individuals wearing face masks and the remaining 686 images of individuals not wearing face masks. Fig 1 largely includes a front face pose with a single face in the frame and with the same kind of mask having white color only.

Fig-1: Dataset 1 Samples Including Faces Without Masks And Masks

Kaggle's Dataset 2[17] consists of 853 images and its presence is explained either by a mask or without a mask. Some face collections in Fig 2 are head turn, tilt, and slant with several faces in the frame and also various kinds of masks with different colors.

Fig -2: Dataset 2 Samples Including Faces Without Masks And Masks

4. INCORPORATED PACKAGES

4.1 TensorFlow

Tensorflow: In order to incorporate ML systems into production across a variety of computer science fields, an interface for expressing machine learning algorithms is used, including sentiment analysis, speech recognition, spatial knowledge extraction, computer vision, text summary, information retrieval, computational drug discovery and fault detection studies [18]. In the proposed model, TensorFlow is used at the backend of the entire Sequential CNN architecture (consisting of many layers). It is also used to reshape the data(image) in the data processing

4.2 Keras

Keras offers fundamental reflections and construction units with high iteration velocity for the formation and transportation of ML arrangements. It takes full advantage of TensorFlow's scalability and cross-platform capabilities. Layers and models are the central data structures of Keras [19]. All the layers used are implemented using Keras in the CNN model. It helps to compile the overall model alongside the conversion of the class vector to the binary class matrix in the data processing.

4.3 OpenCV

OpenCV (Open Source Computer Vision Library), an open-source software library for computer vision and ML, is used to differentiate and recognize faces, recognize objects, group movements in videos, trace progressive modules, follow eye motions, monitor camera actions, expel red eyes from flash images, locate comparative images from an image database, perceive the landscape and set up markers to overlay it with increased reality and so forth [20]. The proposed approach allows the use of these OpenCV features in data image resizing and color conversion.

5. THE PROPOSED METHOD

The proposed method consists of a cascade classifier and a pre-trained CNN which contains two 2D convolution layers connected to layers of dense neurons. The algorithm for face mask detection is as follows:

Algorithm 1: Face Mask Detection

Input: Dataset including faces with and without masks

Output: Categorized image depicting the presence of face mask

For each image in the dataset do

- Visualize the image in two categories and label them
- Convert the RGB image to Gray-Scale image
- Resize the gray-scale image into 100 x 100
- Normalize the image and concert it into 4 dimensional array

End

For building the CNN model do

- Add a convolution layer of 200 filters
- Add the second convolution layer of 100 filters
- Insert a flatten layer to the network classifier
- Add a dense layer of 64 neutrons
- Add the final dense layer with 2 outputs for 2 categories

End

Split the data and train the model

Fig-3: Face Mask Detection Algorithm

5.1 Data Processing

Data preprocessing involves converting data to a much more user-friendly, desired, and meaningful format from a specified format. It can be in any form, such as tables, photos, images, maps, etc. Such ordered data fits into a model or composition of information and captures relationships between various entities [6]. Using Numpy and OpenCV, the proposed method deals with image and video data.

5.1.1 Data Visualization:

Data visualization is the method of translating abstract information by encoding it into concrete representations using knowledge exchange and insight exploration [7].

In both types,' with mask' and 'without mask', the total number of images in the dataset is visualized.

The statement categories = os.listdir(data path) categorizes the directory list in the specified data path. Variable types now look like: ['with mask,' 'without mask']

Then to count the amount of labels, we need to differentiate those categories using labels=[i for I in range(len(categories))]. The labels are set as: [0, 1]

Each category is now mapped to its respective label using dict=dict(zip(categories, labels)), which initially returns a tuple iterator in the form of a zip object, where the items are then paired together in each passed iterator. The mapped variable label dict looks like: {‘with mask’: 0,‘without mask’: 1}

5.1.2 Conversion of RGB image to Gray image:

Modern descriptor-based image recognition systems continue operating regularly on grayscale images, without elaborating the method used to convert from color to grayscale. This is because the color-to-grayscale approach has no impact on the use of robust descriptors. The inclusion of non-essential information could increase the amount of training data needed for successful results to be achieved. As grayscale rationalizes the algorithm and decreases the computational requirements, instead of operating on color images instantaneously [8], it is used to extract descriptors

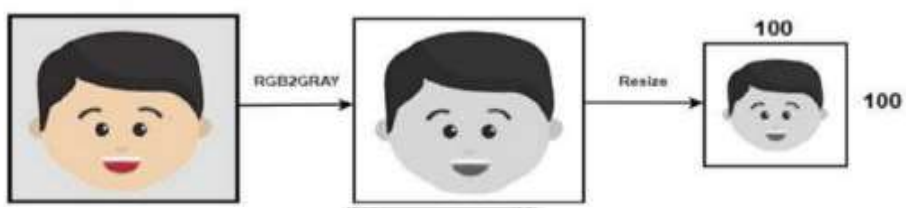

Fig-4: Converting An RGB Image To A Gray Scale Image Of Size:100 X 100

For changing the color space, we use the function cv2.cvtColor(input image, flag). The flag here specifies the form of the conversion [9]. The flag cv2.COLOR BGR2GRAY is used for gray conversion in this case. Deep CNNs require a fixed-size input image. For all the images in the dataset, we, therefore, need a fixed, common scale.

The gray scale image is resized into 100 x 100 using cv2.resize().

5.1.3 Image Reshaping:

A three-dimensional tensor is input during image relegation, where each channel has a prominent unique pixel. All the images must have the same tantamount size corresponding to the 3D feature tensor. Nevertheless, neither images nor their corresponding feature tensors are typically co-extensive. Most CNNs are only able to accept

A three-dimensional tensor is input during image relegation, where each channel has a prominent unique pixel. All the images must have the same tantamount size corresponding to the 3D feature tensor. Nevertheless, neither images nor their corresponding feature tensors are typically co-extensive. Most CNNs are only able to accept Images that are fine-tuned [10]. Throughout data collection and model implementation, this engenders many issues. However, it can help to resolve this limitation by reconfiguring the input images before augmenting them into the network [11].

The images are normalized such that the range of pixels between 0 and 1 converges. Then they are transformed using data=np.reshape(data,(data.shape[0],img size,img size,1)) where 1 indicates the Grayscale image. As there are two outcomes in the final neural network layer, the data is changed into categorical labels with a mask and no mask., i.e. it has categorical representation.

5.2 Training of Model:

5.2.1 Building the model using CNN architecture:

In different computer vision functions, CNN has become ascendant [12]. The current approach uses Sequential CNN. The First Convolution layer is followed by the Rectified Linear Unit (ReLU) and the MaxPooling layers. The Convolution layer learns from 200 filters. The size of the kernel is set to 3 x 3, which determines the height and width of the window of a 2D convolution. As the model should be aware of the expected input shape, it is important to provide information about the input shape to the first layer of the model. Instinctive shape reckoning can be performed by the following layers [13]. In this case, the input shape is specified as data.shape[1:], which returns the data array dimensions from index 1. Default padding is "valid" where the spatial dimensions are authorized to truncate and the input volume is non-zero padded. The activation parameter is set as 'relu' for the Conv2D class. It represents an approximately linear function that possesses all the assets of linear models that can be easily optimized by gradient-descent methods.

Considering the efficiency and generalization of deep learning, it is better compared to other activation functions [14]. In order to minimize the spatial dimensions of the output stream, Max Pooling is used.

The size of the pool is set to 3 x 3 and the resulting output has the shape (number of rows or columns) of: shape of output = (input shape-pool size + 1)/strides), where the strides have the default value (1,1) [15].

As shown in fig 5, the second layer of Convolution has 100 filters and the size of the kernel is set to 3 x 3. This is followed by the ReLu and MaxPooling layers. To insert data into CNN, a long input vector is passed through a flatten layer that transforms a 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. A dense layer of 64 neurons with ReLu activation function is then introduced. The final layer (Densus) with two outputs for two groups uses the Softmax activation function.

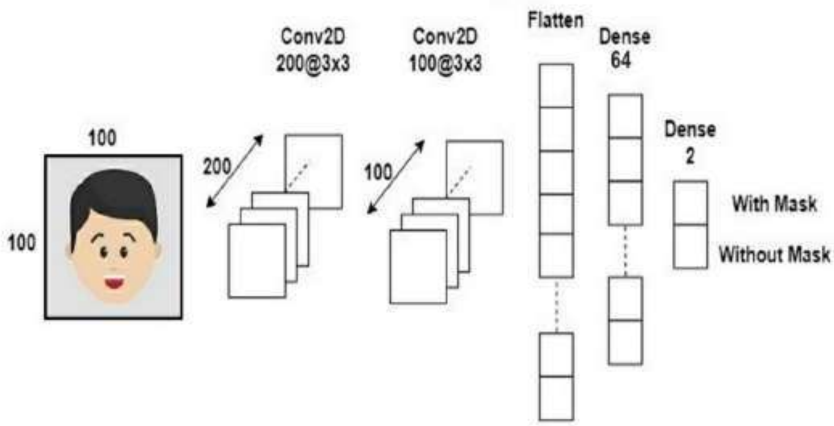

Fig-5: Convolutional Neural Network Architecture

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

5.2.2 Splitting the data and training the CNN model:

The model needs to be trained using a particular dataset after setting the blueprint to analyze the data and then to be tested against a different dataset. A proper model and an optimized split of the train test help to generate precise results when creating a prediction. The test size is set to 0.1, i.e. 90% of the dataset data is trained and the remaining 10% is used for testing purposes. Using ModelCheckpoint, the validation loss is tracked. Next, the images in the training set and the test set are placed on the Sequential model. Twenty percent of the training data here is used as validation data. For 20 epochs (iterations), the model is trained, maintaining a trade-off between accuracy and chances of overfitting.

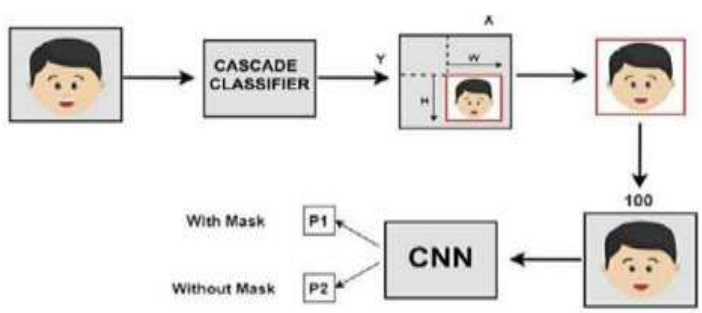

Fig-6: Portrays The Proposed Model's Visual Representation

6. RESULT

Two datasets are used to train, validate, and test the model. The methodology achieves accuracy of up to 95.77 percent when applied to dataset 1. Dataset 2 is more adaptable than dataset 1 because it includes many faces in the frame as well as several types of masks in various colors. As a result, on dataset 2, the model achieves an accuracy of 94.58 percent. Max Pooling is one of the most important factors in reaching this level of accuracy. It gives the internal representation rudimentary translation invariance while also reducing the amount of parameters the model must learn. This sample-based discretization procedure reduces the dimensionality of the input representation, which is an image. The optimal number of neurons is 64, which is not excessively large. The use of a large number of neurons and filters can result in poor performance. The optimal filter settings and pool size aid in filtering out the main portion (face) of the image in order to correctly detect the presence of a mask without over-fitting. With a mask, hair, or a hand, the system can effectively detect partially occluded faces. To distinguish between annotated mask and face covered by hand, it examines the degree of occlusion in four regions: nose, mouth, chin, and eye. As a result, a mask that covers the entire face, including the nose and chin, will only be considered "with mask" by the model. The method's key limitations include shifting perspectives and a lack of clarity. It's made more difficult by the indistinct shifting faces in the video stream. Following the paths of numerous frames of the film, on the other hand, aids in making a more informed selection - “with mask” or “without mask.”

7. CONCLUSION

The purpose for the work was briefly discussed in this paper in the beginning. After which we talked about the model in terms of learning and performance. It can be used for a range of different purposes. In future, in view of the Covid 19 crisis, wearing masks could be necessary. In order to take advantage of their programs, several public service providers will require customers to correctly wear masks. The new model would make a significant contribution to the national healthcare system. The identification of whether a person wears the mask correctly can be extended in the future. The design can be further improved to see whether the mask is susceptible or not to the virus, including the surgical type of the mask, N95 or not.

8. 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 andPrevention,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 Techniques in Data Processing”, International Journal of Computer Sciences and Engineering, vol. 6, no. 10, pp. 712714,2018. 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?”,PLoSONE, 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 Documentation.[online]Availableat:https://opencvpythontutorials.readthedocs.io/en/latest/pytutorials/pyimgproc/pycolorspaces/pycolorspaces.html.2020.

[10] M. Hashemi, “Enlarging smaller images before inputting into convolutional 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 convolutional neural network: Some common and uncommon methods”, Pattern Recognition, vol. 93, pp. 7994,2019. 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.W.H.O., “Coronavirus disease 2019 (covid19): situation report, 205”.2020