International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12– No.40, June 2023 – www.ijais.org

Real-Time Face Mask Detection using Deep Learning

Md. Towhidul Islam Robin
Senior Lecturer
Dept. of CSE, Stamford University
Bangladesh

Md. Samiul Islam
Lecturer
Dept. of CSE, Stamford University
Bangladesh

Ahmed Abdal Shafi Rasel
Senior Lecturer
Dept. of CSE, Stamford University
Bangladesh

Mehedi Hasan
Lecturer
Dept. of CSE, Stamford University
Bangladesh

ABSTRACT

COVID-19 has made a severe impact on the world's economy, health, education sector, and so on. Although the infection rate of the COVID-19 virus has dropped significantly in recent days, there is no scope to ignore precautionary measures like wearing masks in public. In this paper we proposed, a real-time Deep Learning based face detection system using a balanced dataset of over 6000 images divided into two classes named 'With Mask' and 'Without Mask. The existing studies do not consider the data where people wear masks but not correctly, hence violating the safety measures. Our dataset is well-curated to detect people who do not wear masks properly. Our proposed CNN model not only overperformed other existing studies in terms of accuracy but also requires significantly less memory and time compared to other existing models. Our model achieved an accuracy rate of about 98.5% and it requires only 20 Mega Bytes of memory to deploy the model.

General Terms

Artificial Intelligence, Image Segmentation, Machine Learning Models.

Keywords

Covid-19, Deep Learning, CNN, Face Mask

1. INTRODUCTION

According to a recent report by World Health Organization (WHO), more than 600 million people are infected with coronavirus [1]. It is a critical phenomenon how an individual can protect him/herself from this deadly virus. A study showed that covering the nose and mouth with a surgical mask can limit the spread of coronavirus [2][3]. Wearing a mask is simple, yet very effective than other complex and expensive measures. Various public services are required to implement of safety measures like safe distancing, no masks no service, vaccination, etc. It is a necessity to provide an automated mask detection system to ensure prompt service with safety. It also enables the feature of maintaining safety measures in crowded places without any help from dedicated personal assistance. Besides, an alarm-based face detection system can provide sudden awareness to the people to put on their masks properly to avail of any particular services. For this reason, face mask detection emerged as a critical image processing problem that requires a substantial level of accuracy and preciseness.

To detect a face mask, we first need to localize the human face and then the mask position. Detecting the face is a subtask to this problem whereas detecting the mask is similar to finding a particular object detection task. The problem becomes more challenging due to different background colors, different mask shapes, unwanted objects in front of the face, etc. Object detection using manually crafted features like the histogram of Gradients, and scale-invariant feature. transform is showing less effectiveness than the deep learning-based approach. Deep learning models can run features without the help of any prior information on feature extractor methods. Deep learning models can detect features either in one stage or two stages. In our proposed method, we used a single shot detector (SSD) which relies on the single neural network [4]. Our system uses multi-scale detection relies on SSD. It organizes the features from high to low-level information as the model layer deepens to increase the overall performance [5].

2. LITERATURE REVIEW

Face detection problem is not only helpful for assisting covid-19 related safety but also important for multiclass object detection under the image processing domain. Different authors used various classical machine learning-based models as well as pre-trained deep learning models. The most notable models are KNN, SVM, CNN,MobileNetV2, and VGG-19 [6].

Jiang et al. [7] proposed a cascaded CNN framework called Retina Facemask to detect face masks on a human face. The dataset used for this model is a combination of both real and synthetic face mask images with almost 56,000 instances. The particular model is able to handle the loss of face details and also claimed to perform well in obscure images. Retina Face Mask obtained an accuracy of about 94.50%.

Joshiet al. [8] proposed two models to detect face masks. One model uses Retina Face Mask and another model used the Cascaded CNN framework. They captured images from video frames and converted them into fixed-size images to feed into themodel. They achieved an accuracy of nearly 82% while using Retina Face Mask and got 94% precision while using cascaded CNN.

Militante et al. [9] developed a real-time Raspberry PI-based face detection system with instigated deep learning model. Their data set contains 25,000 images with a fixed pixel resolution. The CNN model shows an accuracy rate of 96%. The hardware part of the system is implanted using Raspberry PI with an integrated alarm system.

Sethi et al. [10] proposed an ensemble-based deep learning model to detect fac face masks with low inference time. The authors also mentioned about bounding technique in order to get better localization performance. They use three pre-trained models: RestNet50,MobileNet, and RestNet50 respectively. The model shows the best accuracy while using RestNet50 with an accuracy rate of 98.2%.

Kodali et al. [11] introduced a deep learning-based mask detection system that works with any web camera without any

API. The images are converted to grayscale images in data preprocessing steps. Convolutional Neural Network (CNN) is used to train the model. The model achieved accuracy of 96%.

Table 1. Summary Of Existing Study

DL Model	**Data Set** **Size**	**Accuracy**	**Hardware** **Implementation**	**Ref.**
CNN	56000	94.50%.	No	[7]
CNN	Not Mentioned	82%.	No	[8]
CNN	25000	96.5%	Yes	[9]
RestNet50	7553	98.2%	Yes	[10]
CNN	Not Mentioned	96%	Yes	[11]

3. METH0DOLOGIES AND PROPOSED METHOD

We have used a convolutional Neural Network as our base model to train. The model must be trained on a carefully curated dataset, as will be covered in more detail later in this section. Convolution Neural Network Layers (CNN) serves as the model's underlying architecture and are used to generate various layers and to carry out image processing-related classification tasks in contrast with our system where face mask classification is done. Additionally, tools like OpenCV, Keras, TensorFlow, and Video-Stream libraries are utilized.

3.1 Convolutional Neural Network (CNN)

Image processing and segmentation-related task require more complex feature extraction methods. CNNs generally contains one or more convolutional layer which is being used for identifying more complex and intricate features from images. Convergence and pooling are the two main processes that are frequently used in CNN. Convolution can be performed using many filters by removing features from the dataset, i.e., feature map, while their spatial information is preserved. Pooling sometimes called subsampling. One can reduce the dimensionality by the convolution, extracted features (feature maps) were produced operation. Multiple levels of processing are applied to each input image before earlier, there was convolution and pooling of many types and filters. to the fully connected levels and being transmitted.

3.2 Data Collection

To create a dataset, images from an open source [12] are used. Applying data improvement techniques can increase the size of datasets. We have used 10000 images to train our model. The images are kept in a single folder called "dataset," which has two subfolders called "With Mask" and "Without Mask," each of which holds 40% and 60% of the total number of photos. If the test data makes up 20% and the training data 80%. A variety of techniques are used to build bounding boxes, also referred to as "data annotations," around a region of interest. In the proposed system, labeling images as "with mask" "without a mask" or "incorrect mask" will be done using the labeling technique.

3.3 Data Augmentation

Data augmentation is a process of increasing the data set by adding more samples with different shapes, angles, and rotations in order to ensure more robust performance from the training dataset [13]. To enhance the performance and generalizability of the model, the training dataset is expanded using image data. The Image-Data-Generator class in the Keras deep learning toolkit supports the augmentation of image data. By doing so, we can enhance image data using shift, flip, range, rotation, and zoom. A sample image augmentation is shown at Fig 1:

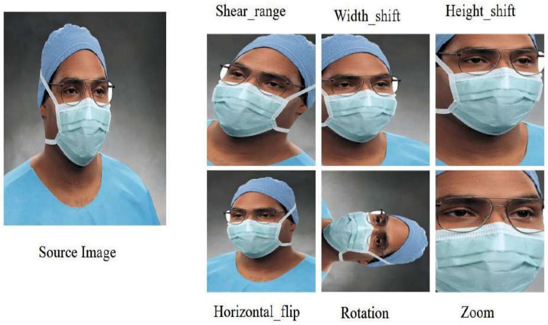

Fig 1: Image Augmentation [13]

3.4 Data Preprocessing

The quality of the data significantly affects the performance of the model. In the initial screening, we identified damaged images and remove them from the original dataset. It is necessary to lower the resolutions of the images in order to ensure optimum run time without compromising the performance of the model. We converted each of the images to a uniform resolution of 224 x 224. The images are assigned to two classes. A single image is converted into a NumPy array to faster the processing time and calculation. Additionally, the input feature is utilized. After that, the data augmentation technique is used to both increase the size and quality of the training dataset. When necessary, the function Image-Data-Generator is utilized. Various iterations of the same image can be created by varying the zoom, rotation, and flip parameters. Over-fitting is a common problem when the number of data is too low, hence we increase the data artificially to get rid of this problem to make the model more generalized. Images are selected randomly from the collected data and therefore split into training data and testing maintaining a ration of 80% in the train data set and 20% in test data set. In order to maintain an equal proportion of data both in the original data set and test-train split data set we choose the stratify value 17.

3.5 Model Training

CNN architecture is generally a combination of a 2D convolutional layer, a pooling layer to pick up the value for the best feature, an activation layer for triggering the neuron followed by a fully connected layer. There are five Conv2D layers in the proposed model. In the pooling layer, a fixed-size filter is used to find the best feature value. The process can be done in two ways either select the max value or the average value from the kernel filter. This feature map is a 2D matrix input and collected as sown in. Fig. 3. which represents a systematic block diagram of our proposed neural network.

The size of the feature map is reduced when layers are pooled. As a result, there are fewer trainable parameters, which allows for quick calculations without sacrificing key components. Max pooling and average pooling are the two primary types of pooling operations that can be performed. The mean of each number in that region is computed, however, using average pooling. Nodes with activation functions are found at the end of or between neural networks (layers). They control whether the neuron fires or not. The model learning performance heavily depends on the activation function choice. This selection of activation function is crucial for hidden layers as well as the output layers. The SoftMax function is generally used in the output layer for a multiclass problem which determines class out from the probability distribution, a set of real number values. For problems involving multi-class categorization, the latter option is favored. In comparison to the function, ReLU delivers greater efficacy and a more extensive level of learning. The FC layers are applied after all Convolutional layers have been completed. These layers aid in categorizing images into binary and multiclass categories. The SoftMax activation function is the preferred option in these layers for generating probabilistic results.

The proposed system is divided into three separate steps. In first steps, image data is prepared and processed in order to feed into the models. Image labeling is also done in this step. In the second step, preprocessed data is fed into the convolutional neural network model. A summary of the proposed CNN architecture and a summary of hypermeters used in the experiment are shown in Table 2. and Table 3. respectively.

Table 2. Model Summary

Layer	Output Shape	No. Of Parameter
Conv2D	(None, 148, 148, 100)	2800
MaxPooling2D	(None, 74, 74, 100)	0
Conv2D	(None, 72, 72, 100)	90100
MaxPooling2D	(None, 36, 36, 100)	0
Flatten	(None, 129600)	0
Dropout	(None, 129600)	0
Dense	(None, 50	6480050
Dense	(None, 2)	102

Table 3. Hyper Parameters Used In The Model

Parameter Name	Details
Learning Rate	0.0001
Epochs	100
Batch Size	32
Optimize	Adam
Loss Functio	Binary Cross Entropy

Our proposed CNN model has a total number of 6,573,052 trainable parameters.

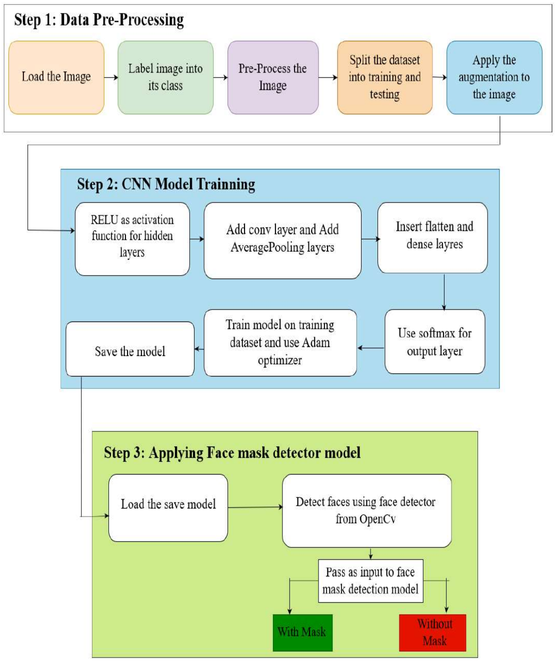
Step 1: Data Pre-Processing
Load the Image
Label image into its class
Pre-Process the Image
Split the dataset into training and testing
Apply the augmentation to the image
Step 2: CNN Model Trainning
RELU as activation function for hidden layers
Add conv layer and Add AveragePooling layers
Insert flatten and dense layres
Use softmax for output layer
Train model on training dataset and use Adam optimizer
Save the model
Step 3: Applying Face mask detector model
Load the save model
Detect faces using face detector from OpenCv
Pass as input to face mask detection model
With Mask
Without Mask

Fig 2: Work Flow of Proposed Model

4. EXPERIMENT RESULT AND ANALYSIS

The dataset employed in this investigation comprises of 10,000 images with and without masks on the faces. The model's accuracy of 98.51% after 100 epochs is significantly higher than that of many other neural networks used for face detection. We also offer a graphical representation of Validation Loss, Training Loss, Validation Accuracy, and Training Accuracy, which is beneficial for making better Validation judgments. Fig. 3: And Fig. 4: below show the output from the live-streaming video that was evaluated by this model. The model can assess the videos and can identify those wearing masks and those who are not. The model was evaluated on a wide range of unique photos. The rectangle red box shows that the person is not wearing a mask, whereas the that the classifier does a decent job of determining whether or not a mask is on a person's face. It should be observed that the F1 scores for the two classes (with mask and without mask) are the same, indicating that the model is performing well. Additionally, it is possible to confirm that the model appropriately fits both classes by looking at the weighted average, regardless of the quantity of data for each class.

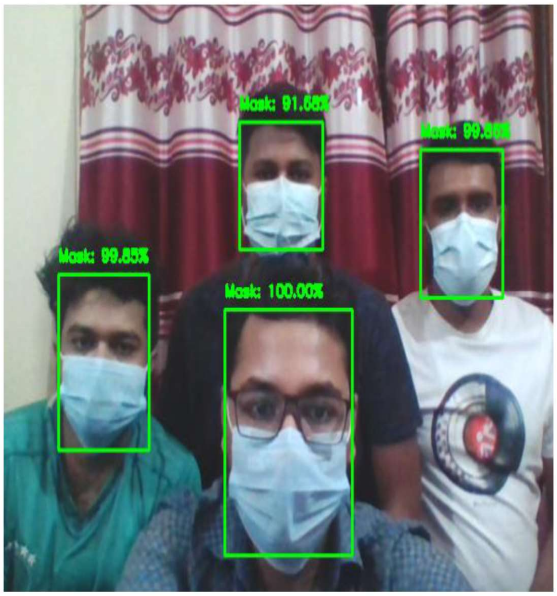

Fig 3: Face Mask Detector With Mask

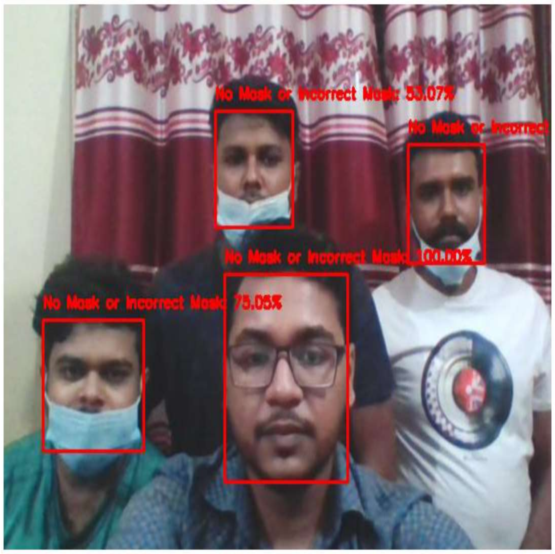
No Mask or Incorrect Mask: 53.07%
No Mask or Incorrect
No Mask or Incorrect Mask: 100.00%
No Mask or Incorrect Mask: 75.05%

Fig 4: Face Mask Detector Without Mask

Table 3. Classification Report

	Precision	Recall	F1 Score
With Mask	0.98	0.97	0.97
Without Mask	0.97	0.97	0.98
Macro Average	0.97	0.97	0.98
Weighted Average	0.97	0.97	0.98

In Fig. 5:, which shows training accuracy, the model achieved nearly about 98% accuracy after 100 epoch. The model loss rate decreases significantly as the number of epoch increases. The model loss curve is shown in Fig. 6:.

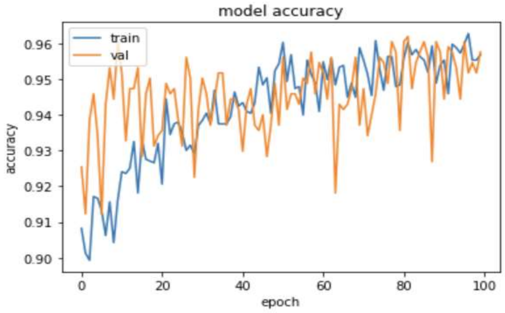

Fig 5: Training and Validation Accuracy

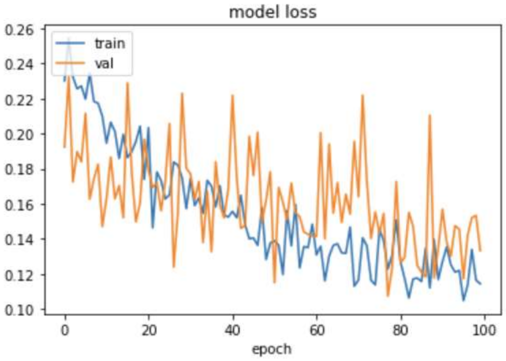

Fig 6: Training and Validation Loss

5. CONCLUSION

In order to stop the COVID-19 pandemic from spreading, we developed an automated system to identify whether a person properly wears masks or not from the live video stream. The accuracy produced by the trained model is roughly 98.50%. Our proposed model outclasses pre-trained models used for similar tasks.

6. REFERENCES

[1] Kabagenyi A, Wasswa R, Nannyonga BK, Nyachwo EB, Kagirita A, Nabirye J, Atuhaire L, Waiswa P. "Factors Associated with COVID-19 Vaccine Hesitancy in Uganda: A Population-Based Cross-Sectional Survey." Int J Gen Med. 2022;15:6837-6847.

[2] Y. Cheng, N. Ma, C. Witt, S. Rapp, P. S. Wild, M. O. Andreae, U. P¨oschl, and H. Su, "Face masks effectively limit the probability of SARS-CoV-2 transmission," Science, 2021

[3] S. Feng, C. Shen, N. Xia, W. Song, M. Fan, and B. J. Cowling, "Rational use of face masks in the COVID-19 pandemic," The Lancet Respiratory Medicine, 2020.

[4] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008, pp. 1–8.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788

[6] M. S. Ejaz and M. R. Islam, "Masked face recognition using convolutional neural network," 2019 Int. Conf. Sustain. Technol. Ind. 4.0, STI 2019, vol. 0, pp. 1–6, 2019, doi: 10.1109/STI47673.2019.9068044.

[7] M. Jiang, X. Fan, and H. Yan, "Case cade framework for masked face detection" 2020. [Online]. Available: http://arxiv.org/abs/2005.03950.

[8] Sathiyanathan, N. "Deep Learning Framework to Detect Face Masks from Video Footage" Journal of Global Research in Computer Science 9.9 (2018): 01-04

[9] S. V. Militante and N. V. Dionisio, "Real-Time Facemask Recognition with Alarm System using Deep Learning," 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC), 2020, pp. 106-110.

[10] R Shilpa Sethi, Mamta Kathuria, Trilok Kaushik, "Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread"Journal of Biomedical Informatics, Volume 120, 2021,103848, ISSN 1532-0464.

[11] R. K. Kodali and R. Dhanekula, "Face Mask Detection Using Deep Learning," 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021, pp. 1-5R.

[12] Matthias, Daniel & Managwu, Chidozie. (2021). Face mask detection application and dataset. Journal of Computer Science and Its Application. 27. 10.4314/jcsia.v27i2.5.

[13] "Image Augmentation" [Online]. Available: https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/ [Accessed: 20-JUN-2022].

[14] Huang Y, Qiu C, Wang X, Wang S, Yuan K (2020) A compact convolutional neural network for surface defect inspection. Sensors 20(7):1974