AUTOMATIC DOOR UNLOCKING SYSTEM BASED ON FACEMASK DETECTION

Shiva G Ramakrishnan
Student, Department of Information Technology, PSG College of Technology, Coimbatore, Tamil Nadu, India.
shivarkrish17@gmail.com

Somasundaram S K,
Assistant Professor, Department of Information Technology, PSG College of Technology, Coimbatore, Tamil Nadu, India.
sks.it@psgtech.ac.in

Ahalya K
Student, Department of Information Technology, PSG College of Technology, Coimbatore, Tamil Nadu, India.
18i461@psgtech.ac.in

Rishi R
Student, Department of Information Technology, PSG College of Technology, Coimbatore, Tamil Nadu, India.
17i333@psgtech.ac.in

ABSTRACT:

There is not really any spot on the planet that hasn't been influenced by novel Covid either monetarily or wellbeing based. The medical care framework is going through an emergency. It is fundamental for individuals to wear masks out in the open spots to decrease the level of transmission of CoVID-19. The proposed framework utilizes deep learning, Keras, TensorFlow and OpenCV to identify face masks. This model is cost-effective to deploy and thus it tends to be utilized for security purposes. A Raspberry Pi based door unlocker circuit is sent alongside the model and this can go about as an extra element alongside security checks and help in checking if the individual entering the premises is wearing a mask or not. This model uses SSDMNV2 approach where a Single Shot multibox Detector is utilized to distinguish faces and a MobileNetV2 design goes about as a system for the classifier. The model gives us an accuracy of 0.98 and F1-score of 0.98.

General Terms:

Face mask detection, framework, keras, deep learning, algorithms.

Key Terms:

MobileNetV2, OpenCV, Bottleneck, Convolutional Neural Network, Data Augmentation.

1.Introduction:

Coronavirus lastingly affects in excess of 100 nations on the planet since December 2019. It was pronounced a pandemic by the World Health Organization (WHO) on 11 March 2020 as it had influenced around 114 nations around then. As of now, the second wave of the destructive infection is progressing in numerous nations and there are numerous reports of changes emerging from the infection in numerous pieces of the world. There have been some new forward leaps in the clinical business and antibodies have been made as of late. Yet at the same time the strength of antibodies against the infection is problematic since there have been numerous instances of individuals getting tainted even after inoculation. Besides, the immunizations aren't made for every one of the mutations of the infection and consequently it isn't protected to say that the pandemic is finished. FaceMask detection has become a moving application since the emerge of CoVID-19, which requests an individual to wear facemasks , follow social separating and use hand sanitizers to wash hands at regular intervals. Despite the fact that an individual is influenced by the infection, there is a likelihood that the individual doesn't show any symptoms. However, this doesn't change the way that he/she can be a mode of transmission. So the Centers for Disease Control and Prevention (CDCP) demand each individual over the age of 2 years to wear facemasks in public places.

Face mask detection has gone up to be a fascinating issue with regards to the field of image processing. Because of the presence of many progressed AI and deep learning models, the issue is by all accounts invalidated completely. Today is more applicable on the grounds that it tends to be utilized progressively management and investigation. With the progression of AI, deep learning and neural network, high exactness and accuracy can be accomplished. A model named SSDMNV2 has been proposed in this paper. SSDMNV2 performs efficiently in separating images with mask from images without front facing mask set up. This model has been deployed with a Raspberry Pi circuit for real-time applications.

2.Related Works:

Meanwhile numerous ongoing frameworks have been created for CoVID-19 from one side of the planet to the other. Numerous scientists and investigators have utilized gray scale images for this in the new past[1]. In Convolutional Neural Network-based model, face detection models gain straightforwardly from the client's information and afterward apply a few deep learning calculations on it (Ren, He, Girshick, and Sun, 2015). In the year 2007, Li, Lin, Shen, Brandt, and Hua (2015) concocted Cascade CNN. In Yang, Yan et al. (2015), Yang et al. concocted highlights accumulation of faces in the face detection model. In additional exploration works, Ojala et al. (2002) redesigned the AlexNet design for tweaking the image dataset. While some were totally based on design ID models, having beginning data of the face model while others were utilizing AdaBoost [2], which was an incredible classifier for preparing purposes. At that point came the Viola-Jones Detector, which gave a leap forward in face detection innovation, and continuous face detection got conceivable. It faced different issues like the direction and splendor of the face, making it difficult to capture. So fundamentally, it neglected to work in dull and faint light. Accordingly, scientists began looking for another elective model that could without much of a stretch distinguish faces just as masks on the face. Previously, numerous datasets for face detection were created to frame an impression of face mask detection models. Prior datasets comprised of images got in regulated environmental factors, while late datasets are built by taking on the web images like WiderFace[23],IJB-A[4],MALF [5], and CelebA [6]. Explanations are accommodated presentfaces in these datasets when contrasted with before ones. Huge datasets are substantially more required for improving preparing and testing information and perform certifiable applications in an easier manner. In Convolutional Neural Network-based characterization, face locator models gain straightforwardly from the client's information and afterward apply a few deep learning calculations on it [10]. In the year 2007, Li et al. [11] concocted Cascade CNN. In[5], Yang et al. concocted highlights total of faces in the face detection model. In additional examination works, Farfade et al. [1] overhauled the AlexNet engineering for adjusting the image dataset. As innovation progressed, further CNN-based 3D models fired coming up; one was proposed by Li et al. [11]. It was a learning structure for face mask detection models. A few different works were done in the circle of posture recognition, gender assessment and so on.

In spite of the fact that there are many face detection and face mask detection models accessible , there are sure downsides to some of them wherein the dataset needed to prepare the model is extremely gigantic and henceforth builds the size of the model sent. This disadvantage can be overwhelmed by utilizing the procedure proposed below in the upcoming sections.

3. Proposed Methodology

To predict whether an individual has worn a mask effectively, the underlying stage is train the model utilizing a legitimate dataset. Insights regarding the Dataset have been examined beneath in Section 3.1. Subsequent to preparing the classifier, an exact face detection model is needed to recognize faces, with the goal that the SSDMNV2 model can group if the individual is wearing a mask. The task in this paper is to raise the accuracy of mask detection without being too expensive.

For doing this task, the DNN module was utilized from OpenCV, which contains a 'Single Shot Multibox Detector' (SSD) (Liu et al., 2016) object detection model with ResNet-10 (Anisimov and Khanova, 2017) as its backbone architecture.

The following classifier uses a pre-trained model MobileNetV2 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) to predict whether the person is wearing a mask or not. The approach used in this paper is depicted in Fig 1.

3.1 Dataset Used and Preprocessing

The images were gotten by utilizing Bing API. They required preprocessing prior to going to the subsequent stage. In the preprocessing step, the image is changed into a grayscale image in light of the fact that the RGB shading image contains such an excess of repetitive data that isn't required for face mask detection. RGB shading image put away 24 digit for every pixel of the image. Then again, the grayscale image put away 8 digit for every pixel and it contained adequate data for arrangement. At that point, we reshaped the images into (64×64) shape to keep up consistency of the info images to the engineering. At that point, the images are normalized and after normalization, the estimation of a pixel dwells in the reach from 0 to 1. Normalization helped the learning calculation to learn quicker and caught essential highlights from the image.

3.2 Data Augmentation

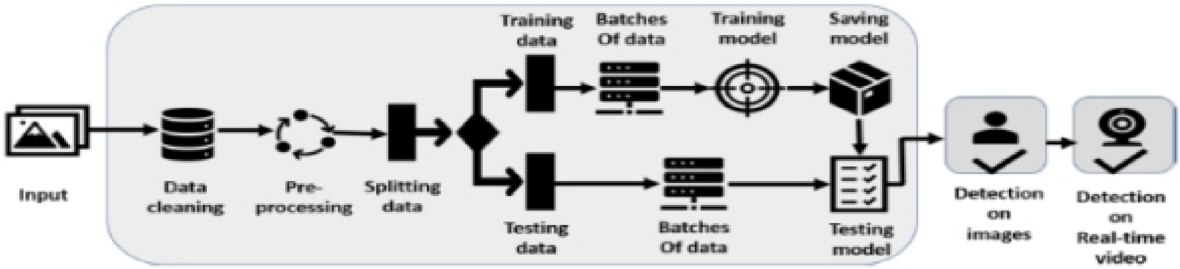

Fig 1: Flow Diagram of SSDMNV2 Model

For viable and proficient expectation utilizing SSDMNV2 model, it requires an enormous measured dataset to be utilized. However, since we have just set number of images, we perform data augmentation ie. Techniques like pivot, zooming, moving, shearing, and flipping the image are utilized for producing various variants of a comparable picture. In the proposed model, image augmentation is utilized for the information increase measure. A capacity image information age is made for image increase, which returns test and train groups of data.

3.3 Face detection using OPEN-CV DNN

This model is set up on the 'Single Shot Multi-box Detector' (SSD) and utilizations 'ResNet-10' design as the base-model. The SSD is like YOLO wherein it makes just a solitary effort to identify numerous faces around there. Two forms of the model are made accessible by OpenCV:

i.)CaffeImplementation (Floating point 16 rendition)

ii.)Original TensorFlow Implementation (8-bit quantized rendition)

Here, we've utilized the CaffeImplementation. For this, the Caffemodel and prototxt documents were stacked utilizing cv2.dnn.readNet("path/to/prototxtfile", "way/to/caffemodelweights"). Subsequent to applying the face detection model, we get the quantity of faces identified, the area of their bouncing boxes, and the certainty score in those expectations. These outputs are then utilized as input for the face mask classifier. Utilizing this way to deal with recognize faces takes into consideration constant detection absent a lot of asset utilization. It can likewise distinguish faces with various orientations and various sizes.

3.4 Classifying images using MobileNetV2

MobileNetV2 is a Deep Neural Network that has been proposed for the grouping issue. Pre-trained weights of ImageNet were stacked from TensorFlow. At that point the base layers are frozen to keep away from weakness of already learned features. At that point new trained layers are added, and these layers are prepared on the gathered dataset so it can decide the highlights to arrange a face wearing a mask from a face not wearing a mask. At that point the model is adjusted, and afterward the loads are saved. Utilizing pre-prepared models evades pointless computational expenses and helps in exploiting effectively one-sided loads without losing already learned features.

3.4.1 Architecture Development

The learning model depends on CNN which is helpful for pattern recognition from images. The network involves an input layer, a few hidden layers and an output layer. The hidden layers comprise of different convolution layers that learn reasonable channels for significant feature extraction from the given examples. The features separated by CNN are utilized by numerous dense neural networks for classification purposes. The design of the created network is delineated in Table I and fig.2. The engineering contains three sets of convolution layers each followed by one max pooling layer. This layer diminishes the spatial size of the portrayal and along these lines decreases the quantity of parameters.

Subsequently, the computation is streamlined for the network. At that point, a flatten layer reshapes the data into a vector to take care of into the dense network. Three sets of dense and dropout layers learn parameters for classification. The dense layer includes a progression of neurons every one of them learn non direct features. The dropout layer keeps the network from over-fitting by exiting units. At last, a dense layer containing two neurons differentiates the classes.

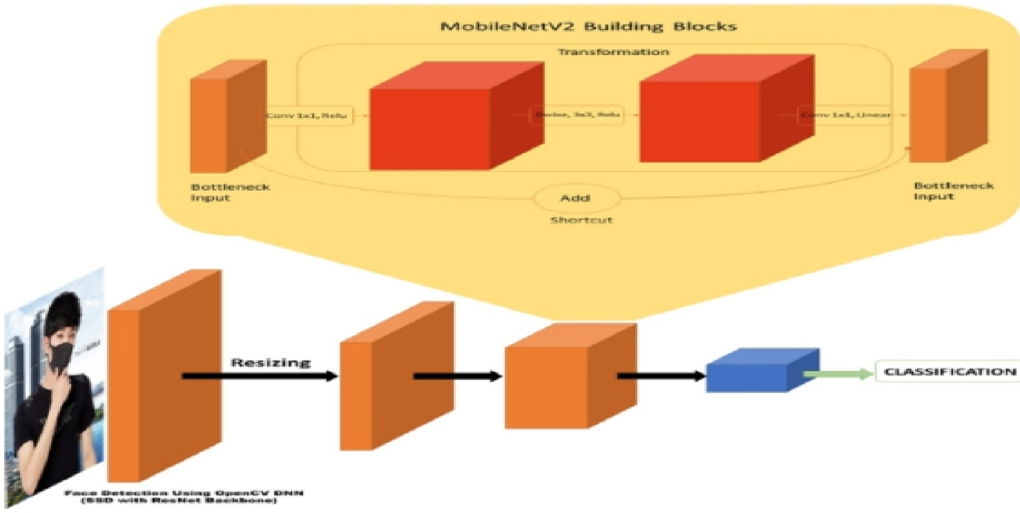

Fig 2: The architecture of the network

Table 1: The architecture description of the developed network

Layer	Type	Kernel	Kernel Size	Output Size
1	Convolution2D	32	(3×3)	(62×62×32)
2	Convolution2D	32	(3×3)	(60×60×32)
3	MaxPooling2D	-	(2×2)	(30×30×32)
4	Convolution2D	32	(3×3)	(28×28×32)
5	Convolution2D	32	(3×3)	(26×26×32)
6	MaxPooling2D	-	(2×2)	(13×13×32)
7	Convolution2D	32	(3×3)	(11×11×32)
8	Convolution2D	32	(3×3)	(9×9×32)
9	MaxPooling2D	-	(2×2)	(4×4×32)
10	Flatten	-	-	512
11	Dense	-	-	100
12	Dropout	-	-	100
13	Dense	-	-	30
14	Dropout	-	-	30
15	Dense	-	-	10
16	Dropout	-	-	10
17	Dense	-	-	2

3.5 Algorithms Used in the proposed work

Algorithm explaining the complete process is as follows:

Algorithm1: Preprocessing and training the dataset

INPUT: Images along with their pixel values.

OUTPUT: Trained Model

Step 1: Start

Step 2: Load the images along with their pixel values.

Step 3: Process the images and convert it into a 1D array.

Step 4: Load the filenames and labels.

Step 5: Perform data augmentation and split data into training and testing batches.

Step 6: Load MobileNetV2 model from Keras and train it on training batches. Compile the trained data using Adam Optimizer.

Step 7: Save the model for future use.

Step 8: Stop

Algorithm2: Deployment of Face Mask Detector

INPUT: Choice of deployment and Files (Optional).

OUTPUT: Classification in Real-time.

Step 1: Start

Step 2: Load saved classifier from disk and face detector from OpenCV.

Step 3: Load the real-time feed from OpenCV

Step 4: Read the feed frame by frame.

Step 5: Apply face detection model to Detect faces in Frames read in real-time

Step 6: Detect the faces and crop face to bounding box coordinates from face detection model

Step 7: Get predictions from the face classifier model.

Step 8: Use a counter variable that increases if face mask is worn or resets to zero if face mask is not worn. If the counter variable reaches a particular value (say 36, which is the value used in our deployed program) or greater than that, the raspberry pi opens the door.

Step 9: Stop

3.6 Circuit Connection

To deploy it in real time, a circuit simulation is done using raspberry Pi. First the Raspbian OS is booted into the Raspberry Pi. Then the circuit is constructed using the circuit diagram (Fig.3) which is explained in the upcoming sub-sections.

3.6.1 Components Required

i) Raspberry Pi 4

ii) Monitor

iii) USB Type C cable

iv) Micro HDMI cable

v) Jumper Wires

vi) Mouse + Keyboard

vii) Breadboard

viii) Micro SD Card

ix) 5V Optocoupler Relay.

x) 12VDC solenoid door lock.

3.6.2 Circuit Explanation

A raspberry pi 4 is used for this project. It uses a Micro-HDMI port for projection and a Type C port for powering up. Hence, a Micro-HDMI cable and Type C cable are used for the respective tasks. The GPIO pins of the Raspberry Pi can give output upto 3.3 V. But the door lock solenoid is 12V. Hence an external power supply and a relay is used to operate the lock. The complete circuit diagram is given in fig. 3

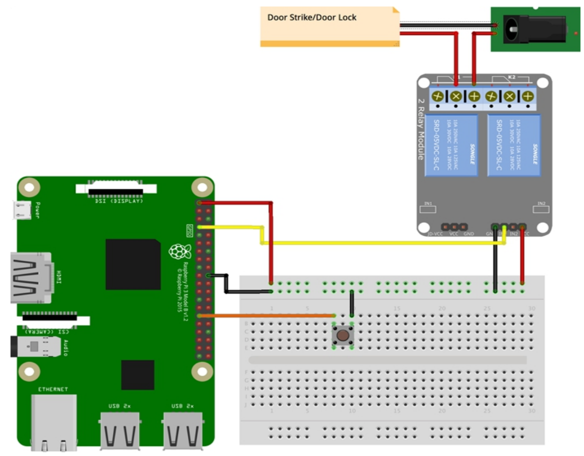

Fig 3: Door Unlocker Circuit

3.7 Deployment

A testing program is made to run on loop in the Raspberry Pi. A testing program is made to run on circle in the Raspberry Pi. A Pi Cam or some other USB Camera is connected to the Raspberry Pi from which it catches the video transfer frame-by-frame. In each frame, it is predicted if the face mask is worn or not utilizing the trained model. Assuming the face mask is identified, it is ensured whether the expectation is directly by checking if the mask is worn for a specific nonstop number of edges (Counter Value). On the off chance that this rule is fulfilled, the Raspberry Pi opens the solenoid door lock. The Counter Value can be increased or decreased. Assuming it is increased, the sensitivity increments yet in addition builds the odds of blunder if the worth is diminished past a specific edge and the other way around.

4. Experimental results

Every one of the test preliminaries have been led on a PC prepared by an Intel i7-8750H processor (4.1 GHz), 8 GB of RAM with 1050timax-Q with 2 GB of VRAM. The trained model has given a precision of 98% and has been exceptionally compelling in detection and perceiving human faces with and without a face mask. The model is trained for 20 epochs for improving outcomes as demonstrated in fig. 4.

The accuracy of the model is measured using F1–score .

It is calculated as,

$$F1 = 2 * (precision * recall) / (precision + recall) \quad (1)$$

where,

```
Epoch 1/20
102/102 [==============================] - 45s 351ms/step - loss: 0.6336 - accuracy: 0.6823 - val_loss: 0.1864 - val_accuracy: 0.9621
Epoch 2/20
102/102 [==============================] - 34s 336ms/step - loss: 0.1790 - accuracy: 0.9513 - val_loss: 0.1119 - val_accuracy: 0.9719
Epoch 3/20
102/102 [==============================] - 34s 336ms/step - loss: 0.1094 - accuracy: 0.9673 - val_loss: 0.0856 - val_accuracy: 0.9731
Epoch 4/20
102/102 [==============================] - 34s 336ms/step - loss: 0.0899 - accuracy: 0.9705 - val_loss: 0.0798 - val_accuracy: 0.9731
Epoch 5/20
102/102 [==============================] - 34s 337ms/step - loss: 0.0728 - accuracy: 0.9764 - val_loss: 0.0641 - val_accuracy: 0.9756
Epoch 6/20
102/102 [==============================] - 34s 336ms/step - loss: 0.0607 - accuracy: 0.9814 - val_loss: 0.0596 - val_accuracy: 0.9768
Epoch 7/20
102/102 [==============================] - 34s 336ms/step - loss: 0.0601 - accuracy: 0.9824 - val_loss: 0.0620 - val_accuracy: 0.9756
Epoch 8/20
102/102 [==============================] - 34s 337ms/step - loss: 0.0461 - accuracy: 0.9853 - val_loss: 0.0584 - val_accuracy: 0.9768
Epoch 9/20
102/102 [==============================] - 34s 337ms/step - loss: 0.0401 - accuracy: 0.9870 - val_loss: 0.0695 - val_accuracy: 0.9731
Epoch 10/20
102/102 [==============================] - 34s 336ms/step - loss: 0.0506 - accuracy: 0.9817 - val_loss: 0.0540 - val_accuracy: 0.9780
Epoch 11/20
102/102 [==============================] - 34s 334ms/step - loss: 0.0291 - accuracy: 0.9918 - val_loss: 0.0465 - val_accuracy: 0.9792
Epoch 12/20
102/102 [==============================] - 34s 336ms/step - loss: 0.0342 - accuracy: 0.9909 - val_loss: 0.0530 - val_accuracy: 0.9792
Epoch 13/20
102/102 [==============================] - 34s 335ms/step - loss: 0.0368 - accuracy: 0.9877 - val_loss: 0.0447 - val_accuracy: 0.9792
Epoch 14/20
102/102 [==============================] - 34s 336ms/step - loss: 0.0339 - accuracy: 0.9889 - val_loss: 0.0400 - val_accuracy: 0.9805
Epoch 15/20
102/102 [==============================] - 34s 336ms/step - loss: 0.0288 - accuracy: 0.9890 - val_loss: 0.0400 - val_accuracy: 0.9817
Epoch 16/20
102/102 [==============================] - 34s 337ms/step - loss: 0.0307 - accuracy: 0.9883 - val_loss: 0.0455 - val_accuracy: 0.9829
Epoch 17/20
102/102 [==============================] - 35s 338ms/step - loss: 0.0279 - accuracy: 0.9894 - val_loss: 0.0452 - val_accuracy: 0.9817
Epoch 18/20
102/102 [==============================] - 34s 337ms/step - loss: 0.0307 - accuracy: 0.9904 - val_loss: 0.0411 - val_accuracy: 0.9805
Epoch 19/20
102/102 [==============================] - 34s 337ms/step - loss: 0.0252 - accuracy: 0.9925 - val_loss: 0.0430 - val_accuracy: 0.9817
Epoch 20/20
102/102 [==============================] - 34s 337ms/step - loss: 0.0283 - accuracy: 0.9905 - val_loss: 0.0427 - val_accuracy: 0.9829
```

Fig 4: Training the Model

precision = (True Positives) / (True Positives + False Positives) – (2)

recall = (True Positives) / (True Positives + False Negatives) –(3)

Fig 5: Training Loss and Accuracy

	precision	recall	f1-score	support
with_mask	0.98	0.99	0.98	433
without_mask	0.99	0.97	0.98	386
accuracy			0.98	819
macro avg	0.98	0.98	0.98	819
weighted avg	0.98	0.98	0.98	819

Fig 6: Accuracy of the model

The figure 5 and 6 represents the Training Loss and Accuracy & Accuracy of the model respectively.

5. Conclusion

In the proposed face mask detection model named SSDMNV2, both the training and improvement of the image dataset, which was separated into classifications of individuals having masks and individuals not having masks have been done effectively. The strategy of OpenCV deep neural networks utilized in this model produced productive outcomes. With the above module conveyed into a solitary mechanized framework, it can furnish with a superior possibility of expectation of weakness to contamination and can likewise help in lessening the spread of the disease. The accuracy of the model can be expanded further via training the model with more images of various orientations and various sorts of faces in both classes(with mask and without mask). Also, extra features can be added to the proposed framework relying upon the necessity. For instance, an IR based non-contact temperature sensor can be added which screens the temperature of the individual entering the premises. Hence, a trustable framework which can foresee with high exactness can be constructed. The framework can be fabricated savvy and strong without necessitating that much support.

6. ACKNOWLEDGEMENTS

Our thanks to the experts who have contributed towards the development of the paper.

7. REFERENCES:

[1] Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(7), 971-987.
[2] Kim, T. H., Park, D. C., Woo, D. M., Jeong, T., & Min, S. Y. (2011, October). Multi-class classifier-based adaboost algorithm. In *International Conference on Intelligent Science and Intelligent Data Engineering* (pp. 122-127). Springer, Berlin, Heidelberg.
[3] Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5525-5533).
[4] Klare, B. F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., & Jain, A. K. (2015). Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1931-1939).
[5] Yang, B., Yan, J., Lei, Z., & Li, S. Z. (2015, May). Fine-grained evaluation on face detection in the wild. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)* (Vol. 1, pp. 1-7). IEEE.
[6] Klare, B. F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., & Jain, A. K. (2015). Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1931-1939).
[7] Jones, P., Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *University of Rochester. Charles Rich*.
[8] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multi-task cascaded convolutional networks. *IEEE Signal Processing Letters*, *23*(10), 1499-1503.
[9] Chen, D., Ren, S., Wei, Y., Cao, X., & Sun, J. (2014, September). Joint cascade face detection and alignment. In *European conference on computer vision* (pp. 109-122). Springer, Cham.
[10] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).
[11] Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2015). A convolutional neural network cascade for face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5325-5334).
[12] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.
[13] Maria Boncy J N, Shanthi D, Somasundaram S K (2018), "Mobility Based Self-Assessment Monitoring System For Diabetes Patients Using IoT", International Journal of Pure and Applied Mathematics, vol. 118, no. 20, pp. 3705 – 3714.
[14] Anisimov, D., & Khanova, T. (2017, August). Towards lightweight convolutional neural networks for object detection. In2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS) (pp. 1-8). IEEE.
[15] Dey, S.K., Howlader, A., & Deb, C (2021). A Multi-phase Face Mask Detection Model to Prevent Person-To-Person Transmission of SARS-CoV-2. In *Proceedings of the International conference on Trends in Computational and Cognitive Engineering*(pp. 603-613). Springer, Singapore.