

# Deep Learning-Based Face Mask Detecting System: An Initiative Against COVID-19



Komal Saini and Nikhil Marriwala 

**Abstract** In this era of pandemic, wearing mask and taking precaution is a must, and this paper provides a way of detecting mask with the help of deep learning. Deep learning is an important part of machine learning. OpenCV, an important part of Python, is used for real-time image detection. This is a very decent system which can be applied at various platforms and can help in slowing down the transmission of coronavirus.

**Keywords** Deep learning · Python · OpenCV · ROI (region of interest) · Adaboost · Cascade Classifier

## 1 Introduction

Coronavirus disease 2019 (COVID-19) is an infectious disease transmitted by a newly discovered virus naming coronavirus. Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is the reason for coronavirus disease which is an ongoing global health emergency. Most of the infected people experience mild to moderate scale of respiratory illness, and many may recover without even requiring any unusual treatment. People with some previous disease like cardiovascular disease, cancer, sugar or any elder age disease are more prone to get infected by this virus.

The finest way to stop or sluggish the transmission of this virus is to be well aware about this virus, the sickness it causes and how it may spread. Protecting oneself and others from infection is very essential and can be done by taking precautions using an alcohol-based sanitizer frequently or washing our hands and not touching your face [1].

---

K. Saini (✉) · N. Marriwala  
Department of Electronics and Communication Engineering, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, Haryana, India

N. Marriwala  
e-mail: [nmarriwala@kuk.ac.in](mailto:nmarriwala@kuk.ac.in)

This virus is spreading by droplets of sneeze form nose and saliva when any infected person coughs or sneezes, so it is very important to practice respiratory etiquette (e.g. by covering face while sneezing and coughing). This virus can impact the mental health in some cases [2].

COVID hit the world suddenly and left us in a jeopardy. It is very necessary in these times that everyone should follow protocols provided by government. Wearing mask, sanitizing and washing hands is a must. But some people seem to just ignore all these and got infected because of their irresponsible actions.

## 2 Approach

For this project, the approach followed is very simple. There is a need of some utilities to be installed which will make the task very easy. The data is taken, then, it is preprocessed, and then, training of the convolutional neural network (CNN) with the data is done. A multi-layer neural network serves well for this purpose. A CNN is a system with different layers of convolution kernels that operate with any original dataset, special features according to the need from dataset are extracted, and thus, it is a very powerful tool for computer vision tasks [3]. Then, the trained model attached with a video stream builds a face recognition system. The next step is testing, and the task is done (Fig. 1).

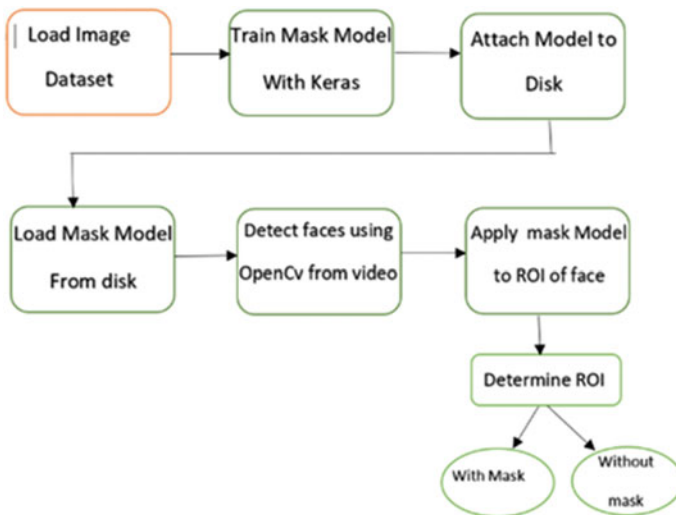


Fig. 1 Flow chart of the face mask detector

### 3 Deep Learning

Deep learning is a part of machine learning. This system aims at learning the feature extraction from higher levels of the hierarchy, and the extracted features can be used for training the CNN. Automatically the extracted features are used and layer by layer help in building CNN. This method can also be used in complex problems [4]. Therefore, deep learning helps in learning representation of the data (Fig. 2).

Deep learning network is similar to a multilevel information-distillation operation, in this any information that goes through a number of filters and comes out increasingly purified (that is, useful with regard to some task) [5].

#### Utilities used:

- TensorFlow
- Keras
- Imutils
- OpenCV Python.

#### 3.1 TensorFlow

TensorFlow is an interface that can be used for algorithm of machine learning. This can be used in number of areas like voice recognition, text summarization, information extraction and many more. Using this for our face mask detection system would be a very good choice [6].

#### Keras

Keras gives a basic reflection, and this is a part of TensorFlow. It has many useful libraries that can ease the task of mask detector model making [7].

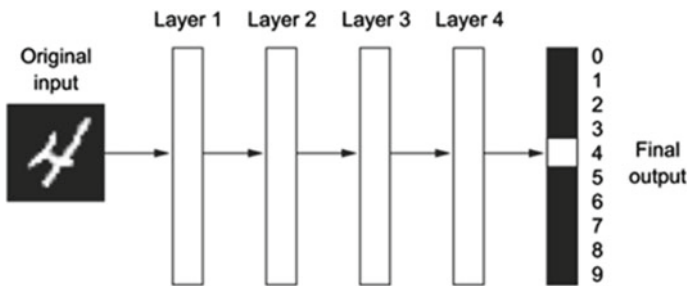


Fig. 2 Deep learning model

### 3.2 Preprocessing

For preprocessing, some images of people wearing mask and some of people not wearing mask are collected to make dataset. These images can be collected from Google or any other open-source image library. Then, these images split in two folders naming “with mask” and “without mask”.

Now in training code, there are two list, i.e. data [] and labels [], the data [] list contain images, and label [] will be having the labels, i.e. “with” or “without” mask. After this, the images are converted into arrays because deep learning model works on arrays only. For converting images to arrays, an `image_to_array` function from Keras is the best choice. Keras is an open-source library in Python for deep learning. This contains many such functions which can ease the process of working with deep learning.

The images are now in machine understandable form (array), but labels are still in alphabetic order. For labels to be in machine understandable form, a label binarizer function from Sklearn would work well, which have many unsupervised and supervised learning algorithms and which can convert labels in binary form.

Next step is splitting the dataset into two sets, and those are train and test. As dataset is small, so `ImageDataGenerator` function is used to expand the dataset. This function increases the dataset as it is able to convert many images of a single image, i.e. rotated, flatted, etc. After this, the dataset will be increased and help in making a better model.

## 4 Training

For training of the model, MobileNet, which is similar to CNN in working but it is faster and uses lesser parameters, is the best choice. MobileNet models are skilled in TensorFlow through using RMSprop with an asynchronous gradient descent which is similar to Inception V3 [8]. CNN is a deep learning algorithm which takes input, assigns importance and is able to differentiate one from other. It is a convolutional neural network architecture which seeks to perform quite well on mobile devices. Here, MobileNet is a kind of convolutional neural network that was designed for mobile and embedded vision applications. They normally hinge on a streamlined architecture that practices a depth-wise separable convolutions to shape lightweight deep neural networks that can have low latency for mobile and embedded devices [9]. By using MobileNet, it is easier to create a head model and a base model for the module, “relu” which is a go to activation function can be used for creating head model.

For training purpose, the code uses the learning rate, the epochs and bit size, and this can be chosen according to the required accuracy, which totally depends on the user. If training rate is less, the system would be more accurate.

The image dataset should be converted into a similar pattern. A desired pixel size can be chosen, and all image data is converted into that size. Like (x, y) can be used for resizing height and width. So that the data is symmetric and is uniform.

After resizing the pixels, max pooling is done, which is nothing but a method of extracting the prominent features and avoiding the unnecessary details. It will extract the necessary details from data. Now, this code is compiled and run, the model will get trained, and a plot of accuracy and loss is plotted.

After training a model which will be able to detect a mask on a human face is generated. But the task is not done yet, this model will only detect mask but for detecting face there is a need to create a program using OpenCV in which can detect a human face and use mask detection model with it to detect face mask.

## 5 Face Detection

The model generated using Keras for mask detection will be coped with a face detection model using OpenCV. The following shown is the AdaBoost algorithm, which is used to build this system.

### 5.1 AdaBoost

In AdaBoost, all weights are prepared equally, but with each next round, the weights of imperfectly classified examples are enlarged so that the weak learner in model is forced to focus on the hard examples in the trading set [10].

For  $m = 1$  to  $M$ .

1. Hand-pick and abstract  $k_m$  from the pool of classifiers that minimizes

$$W_e = \sum_{y_i \neq k_m} \omega_i^{(m)}$$

2. Set the weight of the classifier

$$\alpha_m = \frac{1}{2} \ln \left( \frac{1 - e_m}{e_m} \right)$$

3. Apprise the weights of the data points for next iteration  
If  $k_m(x_i)$  is a miss, set

$$\omega_i^{(m+1)} = \omega_i^{(m)} e^\alpha = \omega_i^{(m)} \sqrt{\frac{1 - e_m}{e_m}}$$

Otherwise

$$\omega_i^{(m+1)} = \omega_i^{(m)} e^{-\alpha} = \omega_i^{(m)} \sqrt{\frac{e_m}{1 - e_m}}$$

## 5.2 OpenCV

Open-source computer vision library (OpenCV) is a useful library, because it has a number of functions which can be helpful for real-time operations, It is established by Intel. It mainly is good with real-time operations because of the libraries it have [11]. OpenCV contains various functions and utilities that appear to be well suited for real-time operations [12].

The architecture of deep learning consists of a number of layers; all are bound to different functions. This layer system is always evolved in the process a system recognition. Hence, deep learning laterally with the face detection can work as the deep layer learning model [13]. Therefore, in face detection area, deep learning usually works in two domains; those are discovering face in a frame and then recognizing it [12].

Python language is a prevailing programming languages and is used all across the world, and this language can be a very decent choice for face recognition task. Python is talented to support a wide variety of third-party tools which make Python a lot more easier to use and motivate the consumers to continue with [14]. Both recognizing and detecting of face can be of ease by using Python and OpenCV [12]. OpenCV practices face detector called “Haar Cascade classifier”. It takes an input image, usually from the camera or real-time video frame, and then, it checks whether it is human face or not and also its location [15].

For detecting any human face, OpenCV can be useful with camera function and image display. In Python, there are a number of libraries like FaceNet which can help in detecting a human face. **FaceNet** can be demarcated as a system that directly learns a planning from face images to a Real-time face detection by using OpenCV, Keras and Python (Fig. 3).

The approach for detecting human face follows extracting the ROI of face using NumPy slicing. NumPy can be defined as a Python library that is used for working with array. It is able to provide high performance multidimensional arrays. NumPy is called an open-source library from Python.

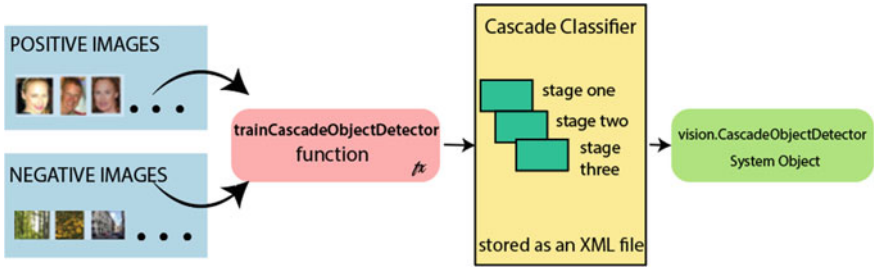


Fig. 3 Cascade classifier

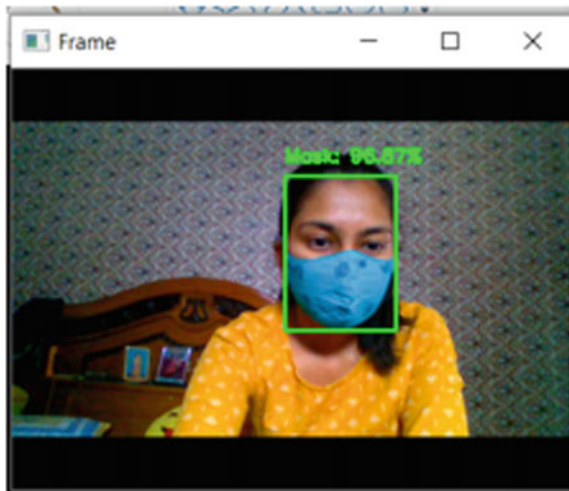
## 6 Result

After successful completion of training and testing the codes, the result would be amazing. After compilation is done, a frame appears as shown below. The window captures real-time frames and detects face, along with that it shows a box around face with the prediction percentage of surety of a mask or not.

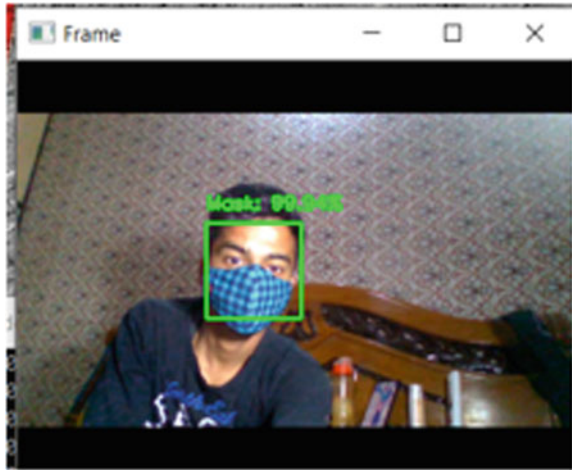
### With Mask

These are the results with mask, taken from different angles and distances. The results do not depend on colour of mask. The model works well even in less light. Figures 4 and 5 is taken from a closer distance, but Fig. 6 is taken from a farther distance. Also, it can be seen in Fig. 7 that more than one person can also be detected by this model.

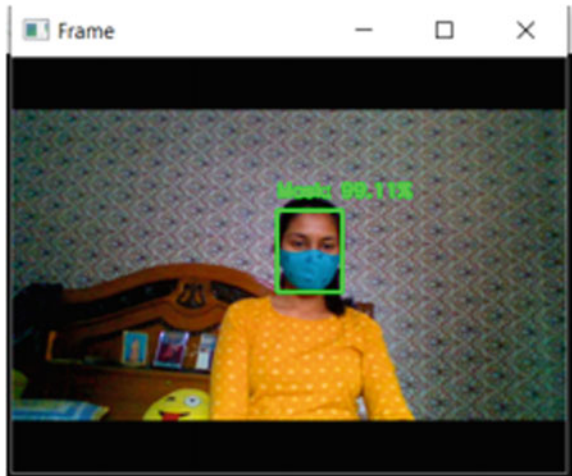
Fig. 4 Image with mask (female)



**Fig. 5** Image with mask (male)



**Fig. 6** Image with mask from a distance

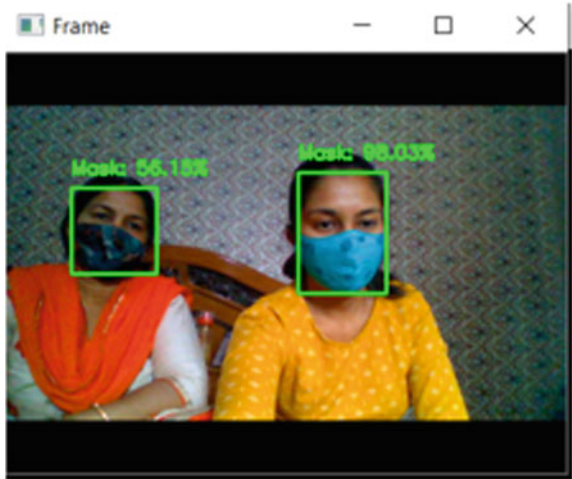


**Without Mask**

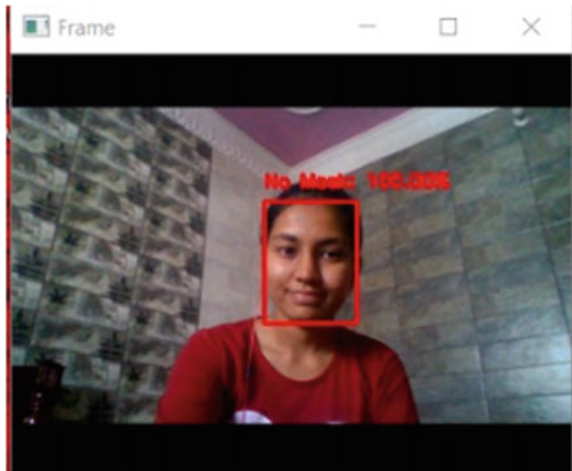
These are the results without mask, taken from different angles, lighting and distances (Figs. 8, 9, 10, 11). This model also works in low lights as shown in Figs. 12 and 13. Also, wearing mask properly is necessary as in Fig. 9 nose is not covered so the model is indicating no mask (Figs. 14 and 15).



**Fig. 7** Image of two people with mask at a time



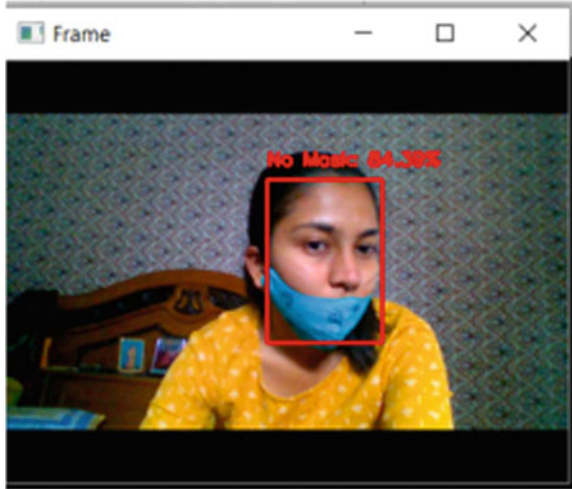
**Fig. 8** Image without mask



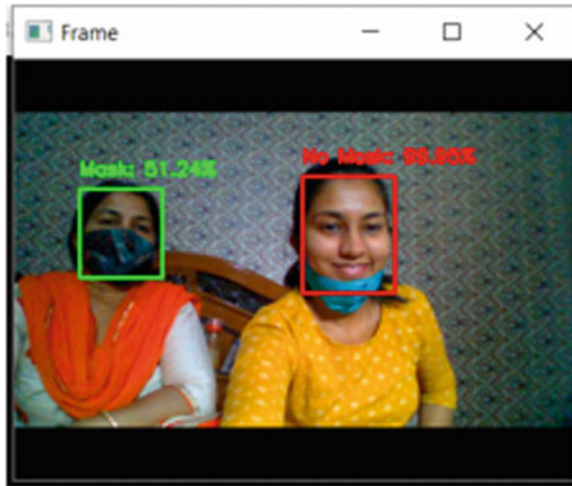
## 7 Conclusions

Face mask detection systems can be simply associated with several of the top technological corporations and industries and can make the work of face mask detection a lot relaxed. This tool is convenient because of the presence of python programming language which is very easy to use and OpenCV. The projected system is very convenient and can be proved very much helpful in a number of places, as this system is easy to build and implement.

**Fig. 9** Image with half mask on



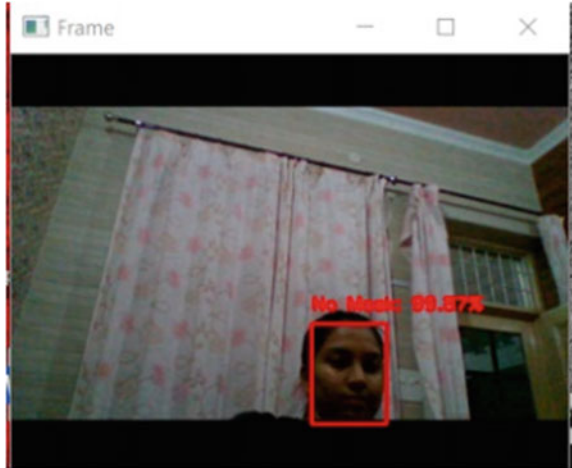
**Fig. 10** Image with one person wearing mask and one not at a same time



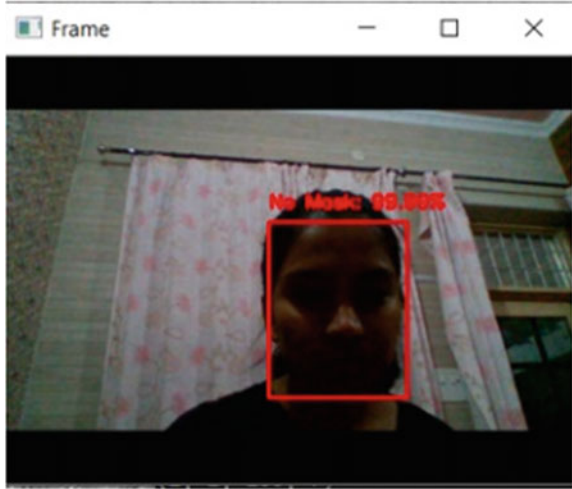
**Fig. 11** Image with one person wearing mask and one not at a same time



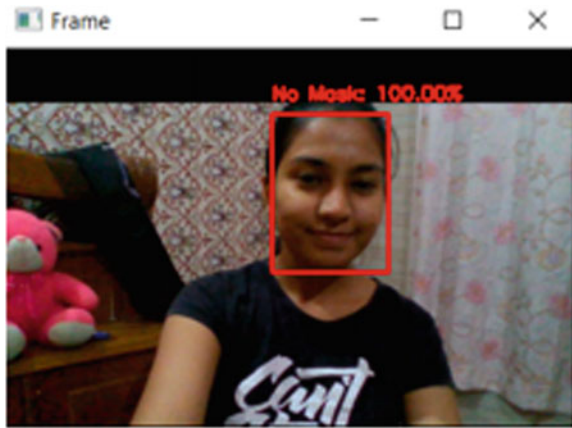
**Fig. 12** Image of person not wearing mask at a distance



**Fig. 13** Image with no mask in less light



**Fig. 14** Image with no mask in proper light



**Fig. 15** Image with no mask in proper light



## References

1. Yuen KS, Ye ZW, Fung SY, Chan CP, Jin DY (2020) SARS-CoV-2 and COVID-19: The most important research questions. *Cell Biosci* 10(1):1–5. <https://doi.org/10.1186/s13578-020-00404-4>
2. Torales J, O’Higgins M, Castaldelli-Maia JM, Ventriglio A (2020) The outbreak of COVID-19 coronavirus and its impact on global mental health. *Int J Soc Psychiatry* 66(4):317–320. <https://doi.org/10.1177/0020764020915212>
3. Rahman MM, Manik MMH, Islam MM, Mahmud S, Kim JH (2020) An automated system to limit COVID-19 using facial mask detection in smart city network. *IEMTRONICS 2020—Int. IOT, Electron. Mechatronics Conf Proc.* <https://doi.org/10.1109/IEMTRONICS51293.2020.9216386>.
4. Vinitha V, Velantina V (2020) Covid-19 Facemask detection with deep learning and computer vision. *Int Res J Eng Technol* 7(8):3127–3132
5. Feng S, Shen C, Xia N, Song W, Fan M, Cowling BJ (2020) Rational use of face masks in the COVID-19 pandemic. *Lancet Respir Med* 8(5):434–436. [https://doi.org/10.1016/S2213-2600\(20\)30134-X](https://doi.org/10.1016/S2213-2600(20)30134-X)
6. Batagelj, Peer P, Štruc V, Dobrišek S (2021) How to Correctly Detect Face-Masks for COVID-19 from Visual Information? *Appl Sci* 11(5):2070. <https://doi.org/10.3390/app11052070>.
7. Sinha D, El-Sharkawy M (2019) Thin MobileNet: an enhanced mobilenet architecture. In: 2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference UEMCON 2019, pp 0280–0285. <https://doi.org/10.1109/UEMCON47517.2019.8993089>.
8. Michele A, Colin V, Santika DD (2019) Mobilenet convolutional neural networks and support vector machines for palmprint recognition. *Procedia Comput Sci* 157:110–117. <https://doi.org/10.1016/j.procs.2019.08.147>
9. Goswami K, Sowjanya AM (2020) Detection of face mask through machine learning 11:450–4
10. Wang R (2012) AdaBoost for feature selection, classification and its relation with SVM, a review. *Phys Procedia* 25:800–807. <https://doi.org/10.1016/j.phpro.2012.03.160>
11. Goyal K, Agarwal K, Kumar R (2017) Face detection and tracking: using OpenCV. In: *Proceedings of the International Conference on Electronics and Communication and Aerospace Technology ICECA 2017*, vol 2017-Janua, no 4, pp 474–478. <https://doi.org/10.1109/ICECA.2017.8203730>.

12. Khan M, Chakraborty S, Astya R, Khepra S (2019) Face detection and recognition using OpenCV. In: Proceedings of 2019 International Conference on Information and Communication Technology and Systems ICCIS 2019, vol 2019-Janua, pp 116–119. <https://doi.org/10.1109/ICCCIS48478.2019.8974493>.
13. Kota VM, Manoj Kumar V, Bharatiraja C (2020) Deep Learning—a review. IOP Conf Ser Mater Sci Eng 912(3). <https://doi.org/10.1088/1757-899X/912/3/032068>.
14. Sharma A, Khan F, Sharma D, Student FY, Area SI (2020) Python : the programming language of fut Int J Innov Res Technol 6(12):115–118
15. Naveenkumar M (2016) OpenCV for Computer Vision Applications, no. March 2015, 2016