IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 7.047 ⁜ Vol. 10, Issue 4, April 2022

DOI: 10.17148/IJIREEICE.2022.10409

FACE MASK DETECTION USING PYTHON AND TENSERFLOW

Vaishnavi Khamankar[1], Neha Kawadkar[2], Chetana Hedaoo[3], Vishal Gupta[4],

Dr. Prabhakar D. Khandait[5]

Dept. of Electronic Engineering. K.D.K. College of Engineering, Nagpur[1-4]

HOD, Dept. of Electronic Engineering. K.D.K. College of Engineering, Nagpur[2]

Abstract: Corona Virus is a big threat to humanity. Now, the whole world is struggling to reduce the spread of Corona virus. Wearing masks is one of the practices that help to control the spread of the virus according to the world health organization. However, ensuring all people wear facemask is not an easy task.

In this paper, we propose a simple and effective model for real- time monitoring using the convolution neural network to detect whether an individual wears a face mask or not. The model is trained, validated, tested upon two datasets. Corresponding to dataset 1, the accuracy of the model was 95.77% and, it was 94.58% for dataset 2.

Keywords: Face mask, Convolution neural network, deep learning, TensorFlow

1. INTRODUCTION

COVID-19 or Corona virus is responsible for producing an atmosphere of terror as it can transmit through the respiratory system. Currently, there is neither medicine nor vaccine to fight against this virus. Therefore, the only options people have to maintain are the social distancing, wash hands regularly, and wear a mask. According to the World Health Organization (WHO)'s, official Situation Report – 205, Corona virus disease 2019 (COVID-19) has globally infected over 20 million people causing over 0.7 million deaths [1]. Individuals with COVID-19 have had a wide scope of symptoms reported like shortness of breath or difficulty in breathing. Elder people having lung disease are at higher risk [2] of getting corona virus than most. The importance of wearing masks lie in reducing vulnerability of risk from a noxious individual during the "pre-symptomatic" period to restrain the spreading of the virus.

WHO stresses on prioritizing medical masks and respirators for health care assistants [4]. Therefore, face mask detection has become a crucial task in the present situation. Face mask detection involves detection of the location of the face and then determines whether it has a mask on it or not. The issue is proximately close to general object detection to detect the classes of objects. Face identification 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]. Deep learning has been used to find out who is not wearing the facial mask using Convolutional neural network (CNN). The remainder of this paper is organized as follows. Section 2 introduces related work. Section 3 describes the datasets used to train the proposed model. Section 4 outlines the incorporated packages. Section 5 and Section 6 contain all the details and the analysis concerning the algorithm. Section 7 concludes the paper.

2. RELATED WORK

In face detection method, a face is detected from an image that has several attributes on 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 image impeded by some other thing not confronting camera, and so forth. Authors in [22] think occlusive face detection comes with two major challenges: first, unavailability of sizably voluminous datasets containing both masked and unmasked faces, second, 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], convolutional neural network (CNNs) 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. In [23], a robust and efficient technique for liveness

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 7.047 ⁜ Vol. 10, Issue 4, April 2022
DOI: 10.17148/IJIREEICE.2022.10409

detection was proposed. The authors used the deep learning DeBNet approach for feature extraction and classification. In [24], the authors used SVM for proposing a machine learning based face detection and recognition system. The proposed model was used to detect the faces of students for monitoring their activities during online examinations. The proposed system used feature vectors from the input images for detecting the faces in a faster manner. In [25], a multi-task deep learning method calledF-DR Net for recognizing and detecting was used.

3. DATASETS

Two datasets have been used in the model. Dataset 1 [16] consists of 1376 images in which 690 images with peoplewearing face masks and the rest 686 images without face masks. Fig. 1 mostly contains front face pose with single face and with same type and color of mask (white only).

3.1

Fig. 1 Samples from Dataset 1 with faces with and without masks [16]

Dataset 2 from Kaggle [17] consists of 853 images and itscountenances are clarified either with or without a mask. In Fig. 2 some face collections are head turn, tilt and slantwith multiple faces in the frame with different types and colors of masks.

Fig. 2 Samples from Dataset 2 including faces without masks and with different types and colors of masks [17]

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 7.047 ⁜ Vol. 10, Issue 4, April 2022

DOI: 10.17148/IJIREEICE.2022.10409

4. INCORPORATED PACKAGES

4.1 TensorFlow

TensorFlow [18], an interface for expressing machine learning (ML) algorithms, is utilized for implementing ML systems into various 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]. The proposed model, the whole Sequential CNN architecture (consists of several layers) uses TensorFlow at backend. Itis also used to reshape the data in the data processing.

4.2 Keras

Keras [19] gives fundamental reflections and building units for 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, the conversion of the class vector to the binary class matrix in data processing, helpsto compile the overall model.

4.3 OpenCV

OpenCV (Open-Source Computer Vision Library) [20], is 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 gesture, track camera actions, expel red eyes from pictures taken utilizing flash, find comparative pictures from an image database, perceive 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 color conversion of data images.

5. THE PROPOSED MODEL

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 facemask
For each image in the dataset do
Visualize the image in two categories and labelthem
Convert the RGB image to Gray-scale imageResize the gray-scale image into 100 X 100 Normalize the image and convert 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 filtersInsert a Flatten layer to the network classifier Add a dense layer of 64 neurons
Add the final dense layer with 2 outputs for 2 categories
End
Split the data and train the model

5.1 Data Pre-Processing

Data preprocessing involves conversion of data from a given format to much more user friendly, desired and meaningful format. It can be tables, images, videos, graphs, etc. This organized information fit in with information model and captures relationship between different entities [6]. The proposed method deals with image and video datausing Numpy and OpenCV [20].

Conversion of RGB image to Grayimage

Modern descriptor-based image recognition systems regularly work on gray scale images, without conversion

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 7.047 ⁜ Vol. 10, Issue 4, April 2022
DOI: 10.17148/IJIREEICE.2022.10409

from color to gray scale. This is because converting the color to gray scale 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 gray scale rationalizes the algorithm and diminishes thecomputational requisites, it is utilized for extracting descriptors instead of working on color images [8].

Fig. 3: Conversion of a RGB image to a Gray Scale image of 100 x 100 size [8]

We use the function cv2.cvtColor(input image, flag) for changing the color space. Here 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 gray scale image is resized into 100 x 100as shown in Fig. 3.

5.1.1 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 identically size corresponding to 3D feature tensor. However, neither images are customarily coextensive nor their corresponding feature tensors [10]. Most CNNs can only accept fine-tuned images. This introduces several problems throughout data collection and implementation of model that can be solved by reconfiguring the input images before augmenting them into the network [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 Gray scale 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.

5.2 Training of Model

5.2.1 Data Mapping

Data visualization is the process of transforming abstract data to meaningful representations using knowledge communication and insight discovery through encodings.It is helpful to study particular pattern in the dataset [7]. The total number of images in the dataset is visualized in both categories – ‘with mask’ and ‘without mask’. The statement categories=os.listdir(data path) categorizes the list of directories in the specified data path. The variable categories now looks like: [‘with mask’, ‘without 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 is paired together consequently. The mapped variable label dict looks like: (‘with mask’: 0, ‘without mask’: 1)

5.2.2 Splitting the data and training the CNNmodel

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 Model Checkpoint. Next, the images in the training set and the test set are fitted to the Sequentialmodel. 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.4 depicts visual representation of the proposed model.

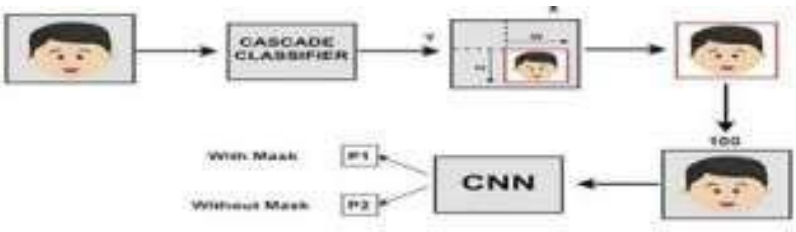

Fig. 4: The Proposed Model

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 7.047 ⁜ Vol. 10, Issue 4, April 2022
DOI: 10.17148/IJIREEICE.2022.10409

5.2.3 Building the model using CNNarchitecture

The current method makes use of Sequential CNN [12]. 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 input shape. The following layers can perform instinctive shape reckoning [13]. In this case, 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 Fig. 5, thesecond Convolution layer has 100 filters and Kernel size isset 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 matrix of features into a vector that can be fed into a fully connectedneural 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 with two outputs for two categories uses the SoftMax activation function.

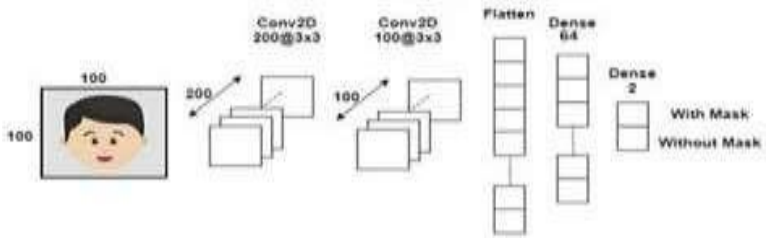

Fig. 5: CNN Architecture [12]

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 asmulticlass log loss is used as a loss function (the objectivethat the model tries to minimize). As the problem is a classification problem, metrics is set to "accuracy".

6. RESULTS AND ANALYSIS

The model is trained, validated and tested upon two datasets. Corresponding to dataset 1, the method attains accuracy up to 95.77% as 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 with different colors. 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

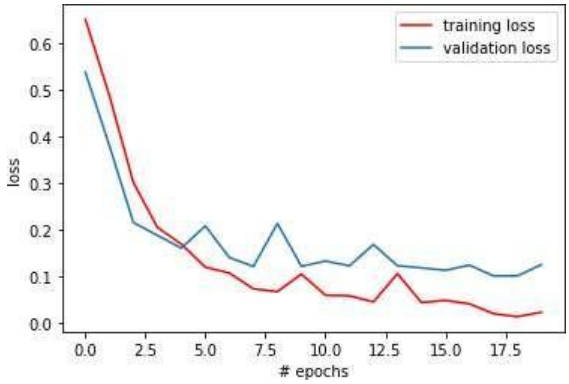

Fig. 6: # epochs vs. loss corresponding to dataset 1

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 7.047 ⁜ Vol. 10, Issue 4, April 2022

DOI: 10.17148/IJIREEICE.2022.10409

parameters the model has to learn. This sample-based discretization process down-samples the input representation consisting of image, by reducing its dimensionality. Number of neurons has the optimizedvalue 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 face in order to detect the existence of mask correctly without causing over-fitting.

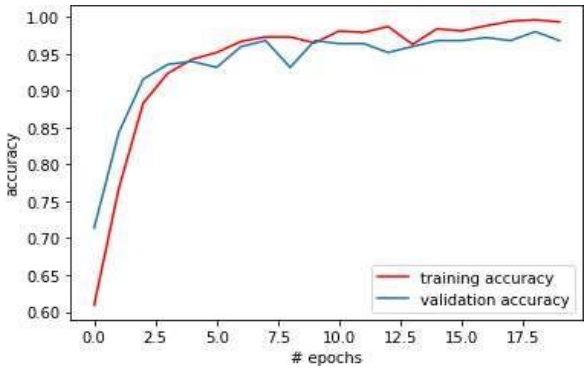

Fig. 7: # epochs vs. accuracy corresponding to dataset 1

The system can efficiently detect faces that are partially occluded (either with a mask or hair or hand). Based on the occlusion degree of four regions (nose, mouth, chin and eye) it differentiates between annotated mask and face covered by hand. Therefore, a mask covering the face fully including nose and chin will only be treated as "with mask" by the model.

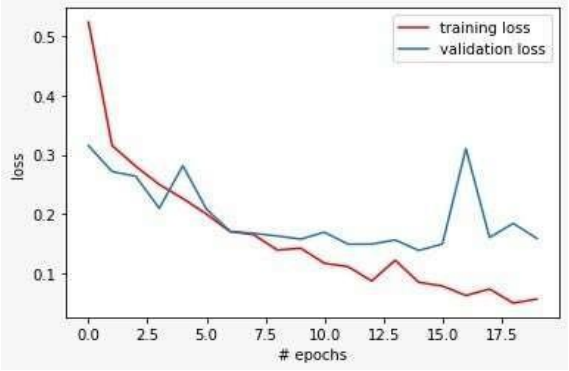

Fig. 8: # epochs vs. loss corresponding to dataset 2

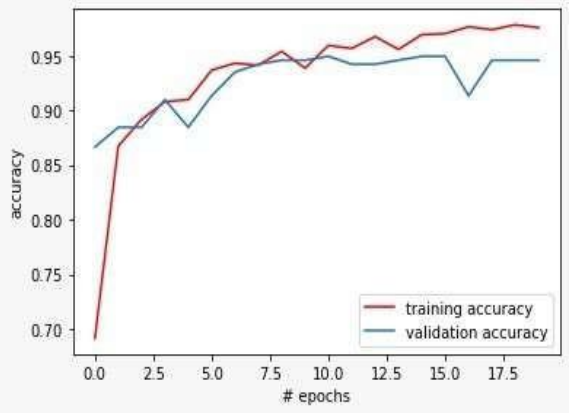

Fig. 9: # epochs vs. accuracy corresponding to dataset 2

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

DOI: 10.17148/IJIREEICE.2022.10409

7. CONCLUSION

Wearing a face mask all the time is difficult and exhausting task but is obligatory since Covid-19 crisis because face mask can help in controlling the outspread of the virus. Many public service providers ask the customers to wear masks in order to fulfill their services.

In this paper, we briefly explained the motivation of the work at first. Then, we illustrated the learning and performance task of the model. Using basic ML tools and simplified techniques the method has achieved reasonably high accuracy.

In future, the model can be extended to detect if a person will wear the mask properly (as instructed by WHO) and also to detect the type of mask.

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/symptomstesti ng/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. 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] Availableat: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 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. 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- 18-0639-9.

[13] "Guide to the Sequential model - Keras Documentation",Faroit.com, 2020. [Online]. Available: ttps://faroit.com/keras-docs/1.0.1/gettingstarted/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/experie ments/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. [Online]. 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, pp. 1-6, 2016, doi: 10.1109/ICRAIE.2016.7939462.

DOI: 10.17148/IJIREEICE.2022.10409

[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, pp. 426-434, 2017, doi: 10.1109/CVPR.2017.53.

[23] S. Garg, S. Mittal, P. Kumar and V. Anant Athavale, "DeBNet: Multilayer Deep Network for Liveness Detection in Face Recognition System," 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN),pp. 1136-1141, 2020, doi:10.1109/SPIN48934.2020.9070853.

[24] M. Geetha, R. S. Latha, S. K. Nivetha, S. Hariprasath, S. Gowtham and C. S. Deepak, "Design of face detection and recognition system to monitor students during online examinations using Machine Learning algorithms," 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021, pp. 1-4, doi: 10.1109/ICCCI50826.2021.9402553.

[25] L. Pang, Y. Ming and L. Chao, "F-DR Net:Face detection and recognition in One Net," 2018 14th IEEE International Conference on Signal Processing (ICSP), pp. 332-337, 2018,doi: 10.1109/ICSP.2018.8652436.