Facemask Detector in Surveillance for COVID-19

Sai Vignesh Ramisetty, D Madhumita, K Yashwanth Chowdary

***Abstract:** Due to this unexpected pandemic we are going on these days, wearing a face mask became mandatory to save ourselves as well as others from the virus. But it is difficult to monitor every citizen whether he is wearing a mask or not. But it is very important. So, to overcome this problem we came up with a solution to monitor every citizen using a deep learning concept. So, we are developing a face mask detector with opencv/keras. This helps us to easily identify the persons wearing masks or not which helps us in taking safety measures according to it. We tried using different types of platforms such as mobilev2net and resnet architecture but the accuracy of resnet architecture is more compared to the other architecture.*

***Keywords:** This helps us to easily identify the persons wearing masks or not which helps us in taking safety measures according to it.*

I. INTRODUCTION

Developing this face mask detector is not only the way but we should develop it in a more portable way which can be used in any area. So, to make it portable we are using a Resnet classifier. This is the only way to make this project portable and user friendly. Basically, there are two phases in this face mask detector. Phase1: Train the face mask detector. Phase2: Apply face mask detector. Phase 1 is the basic step in which we will train the project using the datasets and train it to vary between different types of images. This will be the testing and comparison set to the real images. In phase 2, the given image is compared with the stored dataset and gives us the original output whether the person is wearing a mask or not. The detailed explanation is in phase 2 the face mask classifier is loaded from the disk, and then the camera detects the image and extracts the ROI of each face. This ROI is applied to each face mask classifier to detect the mask. Finally, it gives the output.

II. OBJECTIVE

We aim to design a hands-free entry system using a face mask detector in surveillance to combat the further spread of the virus. This will ensure to reduce the transmission of pathogens on high-touch surfaces, like door handles, and to prevent entry in a community area without a face mask. To accomplish this task, we will be fine-tuning the Resnet architecture, which will help us in training hundreds of layers quickly and making sure that there won't be a drop in the training percentage.

III. LITERATURE REVIEW

[1] The vision towards computers is changing every 13 months. During this change using the computer technology in the neuro-scientific field also came into light in which this face detection is also one of the parts. [2] Algorithms based on principle component analysis are the bases of the numerous studies in neuroscience and algorithmic face detection literature. Eigenvector is one of the main concepts in differentiating and grouping the data to train the device. [3] Every face is treated as a two-dimensional face rather than three dimensional to make the identifying easier. But these images can be identified in any colour but the image should not be tinted with green colour. [4] A view-based multiple-observer eigenspace technique is proposed for use in face recognition under variable pose. In addition, a modular eigenspace description technique is used which incorporates salient features such as the eyes, nose and mouth, in an eigen feature layer. This modular representation yields higher recognition rates as well as a more robust framework for face recognition. An automatic feature extraction technique using feature Eigen templates. [5] For face detection, Haar-Cascades were used and for face recognition Eigenfaces, Fisherfaces and Local binary pattern histograms were used. The difference between face detection and identification is, face detection is to identify a face from an image and locate the face. Face recognition is making the decision" whose face is it?", using an image database. The technologies available in the Open-Computer-Vision (OpenCV) library and methodology to implement them using Python.[6] Human face localization and detection is often the first step in applications such as video surveillance, human computer interface, face recognition and image database management. Locating and tracking human faces is a prerequisite for face recognition and/or facial expressions analysis, although it is often assumed that a normalized face image is available. In this paper we intend to implement the Haar-Classifier for Face detection and tracking based on the HaarFeatures. This is done using Haar Classifier through Raspberry Pi BCM2835 CPU processor which is a combination of SoC with GPU based Architecture. SimpleCV and OpenCV libraries are used for face detection and tracking the head poses position.

Manuscript received on July 16, 2021.
Revised Manuscript received on July 19, 2021.
Manuscript published on July 30, 2021.
* Correspondence Author

Sai Vignesh Ramisetty*, Studying B.Tech (4th year), Department of Electronics and Communication Engineering (Spec in IOT & Sensors), Vellore Institute of Technology, Vellore (Tamil Nadu), India. Email: vigneshramisettyrsv1234@gmail.com

D Madhumita, Studying B.Tech (4TH year), Department of Electronics and Communication Engineering (Spec in IOT & Sensors), Vellore Institute of Technology, Vellore (Tamil Nadu), India. Email:madhu2892000@gmail.com

K Yashwanth chowdary, Studying B.Tech (4th year), Department of Electronics and Communication Engineering (Spec in IOT & Sensors), Vellore Institute of Technology, Vellore (Tamil Nadu), India. Email: yashwanth.chowdary2018@vitstudent.ac.in

IV. SOFTWARE REQUIREMENTS

Tensorflow & Keras: Tensorflow is used to achieve following

- Data augmentation
- Loading the MobilNetV2 classifier
- Building a new fully-connected (FC) head
- Pre-processing
- Loading image data

Sklearn: sklearn is used for binarizing class labels, segmenting our dataset, and printing a classification report.

Imutils: This helps in listing the paths of images inside the dataset.

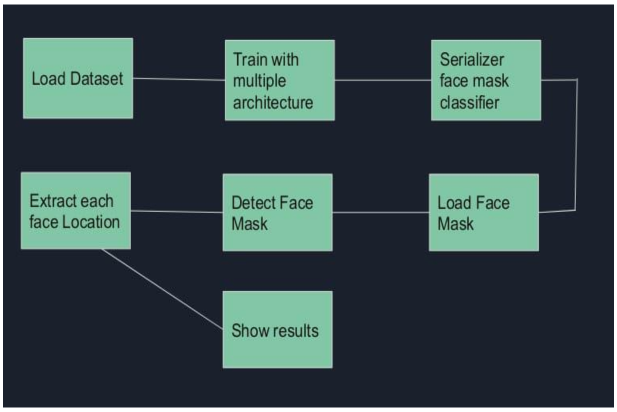
V. PROPOSED SYSTEM

We will take an input from Real-time video as an input and this video is processed using the algorithm developed by us. We use the testing algorithm to detect if the person identified in the image is wearing a mask or not this then gives a digital output which includes the status of the person and the accuracy of the output. This output will be displayed on the screen.

VI. WORKING

A dataset containing images will be used to train and test the code. This dataset contains 690 images with people wearing masks and 600 images with people without a mask . 80% of the dataset is used for training the algorithm and the remaining 20% is used for testing. In the training at first, we will be loading the complete dataset of images as a list. We will be using two arrays in which one of the arrays will contain the data(images) and the other one contains the label(names) of the image. Then both arrays are changed into a numpy array. Now we split the data set into training and testing parts. We have to arrange every image in the dataset according to our requirements so we will be using different syntaxes to alter the dimensions of every image now using the Resnet architecture which was designed to enable hundreds or thousands of convolutional layers. Here we'll focus on loading our face mask detection dataset from disk, training a model (using Keras/TensorFlow) on this dataset, and then serializing the face mask detector to disk. Using this resnet we will create a base model which is used to train the algorithm. This base model is used in prediction of different images in the dataset and the label which is used in the second array is matched with these images. This is going to help in serializing the data so this will end the training part of the system. Coming to the testing part a Real-time video will be given as an input. A video is nothing but a combination of frames so in this testing part we will extract frame from this Real-time video every 1 second this frame is going to be used for mask detection once the frame is extracted the algorithm will identify the person in the image and bounds the face of the person in a rectangular box. When an frame is extracted it is converted into data which is then stored into an array if the value of the array happens to be greater than 0 it means that the face of the person is identified by the algorithm. According to the output if the person is wearing a mask he will be bound with a green box. If not he will be bound with a red box every image will be compared with a set of images in the data set and the output is given according to the results. So, the value stored in the array will help us identify the face of the person in the image as soon as this happens the detection of the mask takes place so if the person is wearing a mask it's going to print yes along with its accuracy percentage.

VII. BLOCK DIAGRAM

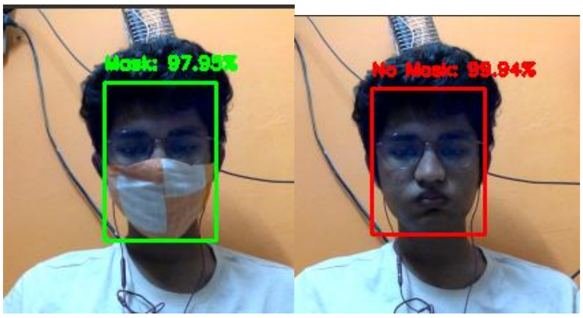

Load Dataset
Train with multiple architecture
Serializer face mask classifier
Extract each face Location
Detect Face Mask
Load Face Mask
Show results

VIII. RESULTS

We have got an accuracy of 89%. With 1:4 ratio of testing: training data.

	precision
with mask	0.89
without mask	0.90
accuracy	0.89

These are our simulations for the same:



IX. CONCLUSION

This paper concludes the development and working of face mask detector using mobile v2 classifier and this is very portable and the accuracy and precision are much better than compared to other systems. By further developing this concept we can add this to the entrance which automatically opens the door on detecting the mask.

REFERENCES

1. S. Wang Chen, Horby Peter W, Hayden Frederick G, Gao George F. A novel corona virus epidemic of global concern for health. It's the Lancet. 2020;395(10223):470-473. 10.1016 / S0140-6736(20)30185-9.
2. Matrajt L, Leung T. Evaluating the efficacy of social distancing strategies to postpone or flatten the curve of coronavirus disease. Emerg Infect Dis, man. 2020:

3. Chen, S., Zhang, C., Dong, M., Le, J., R., M., 2017b. Using rankingcnn for age estimates, in: IEEE Conference on Computer Vision and Pattern Recognition(CVPR)
4. Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residues and linear bottlenecks," IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510-4520. 2018.
5. C.Fu, W.Liu, A.Ranga, A. Tyagi, A. Berg, "DSSD: deconvolutional single shot detector model," arXiv preprint arXiv:1701.06659, (2017)
6. Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Type Pyramid Networks for Object Detection," IEEE Conference Proceedings on Computer Vision and Pattern Recognition, pp. 2117-2125. 2017.
7. GOODFELLOW, I., BENGIO, Y., & COURVILLE, A. (2016). Deep learning process. Chapter 6.
8. S. S. Farfade, M. J. Saberian, and L. Li. Multi-view face recognition using deep convolutional neural networks. In ACM ICMR, pages 643–650, 2015
9. Masita, K. L., Hasan, A. N., and Satyakama, P., 2018. Pedestrian identification by mean of R-CNN object detector. IEEE Latin American Conference on Computational Intelligence (Nov. 2018). DOI=10.1109/LA-CCI.2018.8625210.
10. R. Girshick, “Fast R-CNN,” in Proc. IEEE International Conference Computer Vision, Dec. 2015, pp. 1440–1448.
11. S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with area proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.
12. Liu, H., Chen, Z., Li, Z., and Hu, W. 2018. A powerful method of pedestrian detection based on YOLOv2. Mathematical Engineering Issues(Dec. 2018). DOI=https://doi.org/10.1155/2018/3518959.
13. P. Viola and M. Jones, "Fast object detection using an enhanced cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I.
14. X. Zhu and D. Ramanan. Face detection, pose estimation,and landmark position in the wild. In IEEE CVPR, pages 2879–2886, 2012
15. Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). MobileNets: Effective Convolutional Neural Networks for Mobile Vision Applications.
16. Zhu, X. 2006. Semi-supervised study of learning literature survey. Computer Science, University of Wisconsin-Madison 2(3):4.
17. Facial Recognition using OpenCV, Shervin EMAMI ,Senior Systems Software Engineer Mobile Computer Vision Team, NVIDIA, AUSTRALIA. Valentin Petruț SUCIU IT&C Security Master Department of Economic Informatics and Cybernetics Bucharest University of Economic Studies ROMANIA
18. Computation and performance aspects of pca-based face-recognition algorithms Hyeonjoon moon, Department of Electrical and Computer Engineering, state university of new york, buffalo,amherst, NY 14260, USA P jonathan phillips, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA.
19. Eigenfaces and recognition Matthew turk, Vision and modelling group, the media laboratory,Massachusetts Institute of Technology. Alex Pentland, Vision and modelling group, the media laboratory,Massachusetts Institute of Technology.
20. View-Based and Modular Eigenspaces for Face Recognition Alex Pentland, Perceptual Computing Group, The Media Laboratory Massachusetts Institute of Technology 20 Ames St., Cambridge, MA 02139 Baback Moghaddam, Perceptual Computing Group, The Media Laboratory Massachusetts Institute of Technology 20 Ames St., Cambridge, MA 02139 Thad Starner, Perceptual Computing Group, The Media Laboratory Massachusetts Institute of Technology 20 Ames St., Cambridge, MA 02139
21. Face Detection & Face Recognition using open Computer Vision Classifiers LAHIRU DINALANKARA, August 4, 2017 Robotic Visual Perception and Autonomy Faculty of Science and Engineering Plymouth University,
22. Real Time Face Detection and Tracking Using OpenCV Prof.PY Kumbhar, Department of Electronics and Telecommunication Engineering, Walchand Institute of Technology, Solapur, Maharashtra Mohammad Attaullah, Department of Electronics and Telecommunication Engineering, Walchand Institute of Technology, Solapur, Maharashtra Shubham Dhere, Department of Electronics and Telecommunication Engineering, Walchand Institute of Technology, Solapur, Maharashtra Shivkumar Hipparagi, Department of Electronics and Telecommunication Engineering, Walchand Institute of Technology, Solapur, Maharashtra

AUTHORS PROFILE

Sai Vignesh Ramisetty, Presently studying BTech (4th year), Electronics and Communication Engineering (Spec in IOT & Sensors) in Vellore Institute of Technology, Vellore.

D Madhumita, Presently studying BTech (4TH year), Electronics and Communication Engineering (Spec in IOT & Sensors) in Vellore Institute of Technology, Vellore.

K Yashwanth chowdary, presently studying BTech (4th year), Electronics and Communication Engineering (Spec in IOT & Sensors) in Vellore Institute of Technology, Vellore.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication