IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

REAL TIME FACE MASK DETECTION AND ACCURACY PREDICTION

Computation using OpenCV and TensorFlow

[1] Reddy Harithasri,[2] Samardi Preethi,[3] Sanapala Anusha,[4] Sappa Prameela,

[5] Dr.P.Himakeerthi

[1,2,3,4]Students, [5]Project Guide,
Computer Science and Systems Engineering (CS&SE)
Andhra University College Of Engineering For Women (AUCEW), Visakhapatnam, Andhra Pradesh, India

Abstract: Wearing a face mask appears to be important to both the person and other people to restrict the spread of a virulent disease that can be communicated via sputtering in the context of a virulent sickness which can be transmitted via sputtering. During the COVID-19 pandemic, people all around the world were required to wear masks to limit the danger of virus transmission. According to research, properly wearing a face mask can effectively reduce transmission and save lives. The identification of face masks has made significant progress in the domains of image processing and computer vision since the start of the Covid-19 pandemic. Face recognition models have been built using a number of algorithms and methodologies. This is a deep learning model design using Python, TensorFlow, Keras, MobileNet, and openCV in this project. The main goal is to utilise a trained AI model and a collection of masked and unmasked photos to determine whether or not someone is wearing a mask during a webcam Livestream. Because it is relatively resource efficient to deploy, this model can be employed for safety considerations

Index Terms **– Open cv, mask Detection,OpenCV Library,Image Recognition,Tensor Flow, Convolutional Neural Network,python,machine learning**

INTRODUCTION: The COVID-19 outbreak has startled the world since the year began, and the year 2020 has thrown mankind a mind-boggling series of catastrophes, the most life-changing of which is the COVID-19 pandemic. COVID-19 has urged for strict measures to be taken to prevent the spread of disease, which has an impact on many people's health and life. People are doing everything they can to safeguard themselves and society, from basic grooming to medical treatments; face masks are one type of personal protective equipment. Face masks are worn when people leave their houses, and officials make it a point to ensure that people wear them in groups and public places. To guarantee that citizens follow this essential safety principle, a policy should be developed. A technology that detects face masks can be used to verify this. Face mask detection is the process of recognising whether or not someone is wearing a mask. To detect the existence of a mask on a face, the first step is to detect the face, which divides the technique into two parts: detecting faces and detecting masks on those faces. Face detection is a type of object detection that can be useful in a variety of scenarios, including defence, biometrics, and law enforcement. Several detector systems have been created and are currently in use around the world. However, much of this research still has to be improved; a better, more precise detector is required because the world cannot afford any more corona cases. The goal of this project is to develop a face mask detector that can distinguish between those who are wearing masks and those who aren't. Facenet is used for face detection, and a neural network is used to detect the presence of a face mask in this study's proposed methodology. Using the adam optimizer, the programme processes pictures, videos, and live video streams.

NEED OF THE STUDY

Face mask detection refers to detect whether a person is wearing a mask or not. In fact, the problem is reverse engineering of face detection where the face is detected using different machine learning algorithms for the purpose of security, authentication and surveillance. Masks are one component of a package of prevention and control measures to limit the spread of COVID-19. In addition to mask wearing, these measures include avoiding crowded spaces, avoiding poorly ventilated spaces and improving ventilation in indoor spaces, keeping a distance, hand hygiene, respiratory etiquette - covering your mouth and nose with a bent elbow or a tissue when you cough or sneeze, getting vaccinated and staying up to date with booster doses.

RESEARCH METHODOLOGY

SYSTEM ANALYSIS

HARDWARE REQUIREMENTS:

- Processor 4.0GHz
- RAM 2 GB
- Hard Disk 30 GB
- Keyboard 104 Keys
- Display VGA

SOFTWARE REQUIREMENTS:

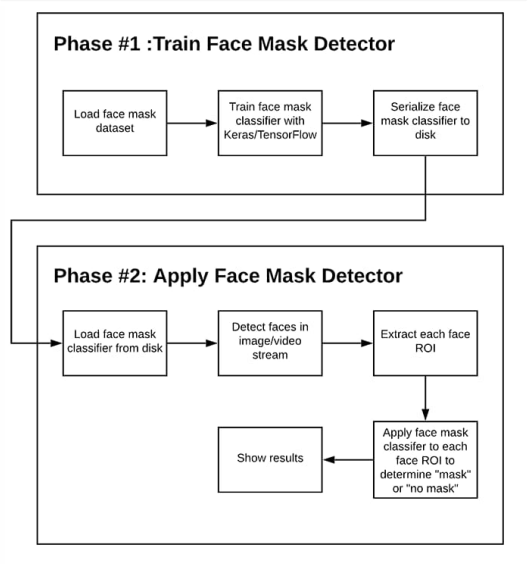
- Operating Systems Windows/Linux
- Language Python
- Editor Spyder
- Interpreter Python Interpreter

NON-FUNCTIONAL REQUIREMENTS:

1. Execution qualities: Efficiency
2. Evolution qualities Testability
3. Extensibility
4. Scalability
5. Usability
6. Reliability
7. Performance
8. Supportability
9. Implementation

The project is developed in 2 phases:

- ➢ Phase 1: training the Face mask detector
- ➢ Phase 2: Applying the face mask detector



Under Phase 1:

Under this phase various steps like creation of data set , pre-processing of data, trainingthe model are included.

1. DATASET

   This model was trained on a data set that included two types of images: with mask and without mask. Images were gathered from a variety of sources, including Kaggle, Google Images, and a few open source image libraries, to build this data set.

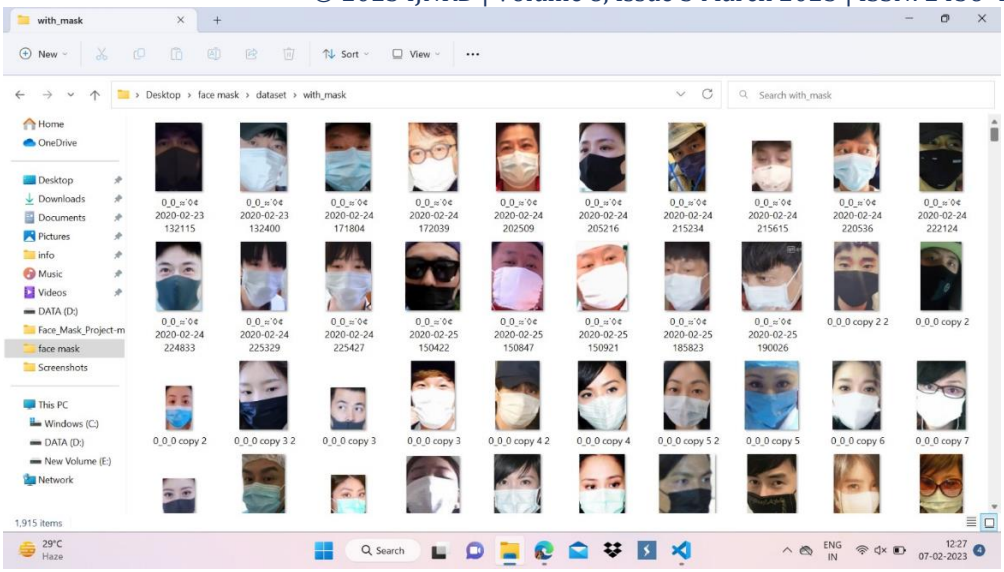


Figure: dataset with mask

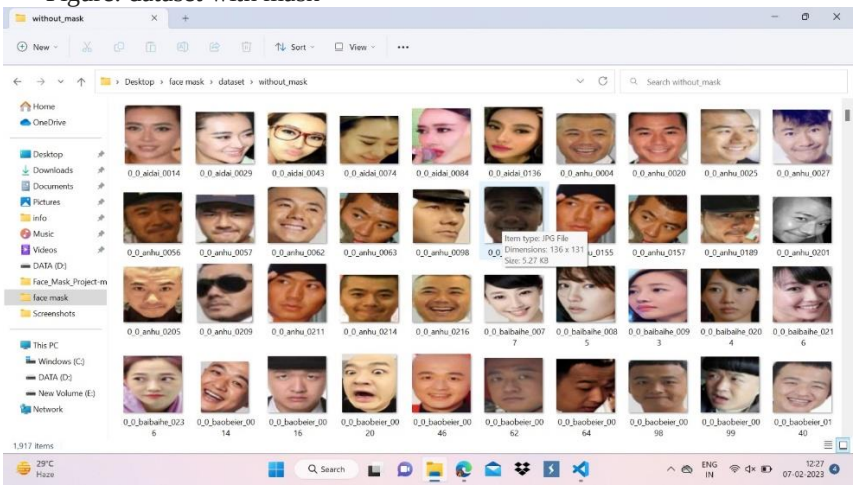


FIGURE: dataset without mask

2. DATA PRE-PROCESSING

The data pre-processing is one of the major steps of this project. The steps as mentioned below was applied to all the raw input images to convert them into clean versions, which could be fed to a neural network machine learning model. • Converting all the images in the data set to (224 x 224) pixels using the load image function of keras. • The images are then converted into Numpy arrays using image to array function of keras. • The images are appended to label list and then convert them into arrays as deep learning model work with arrays. • The data set is then split into training and testing set.Here it is in the ratio 80:20. 3. TRAINING • The training image generator for data augmentation is constructed, by which mul- tiple images can be created using a single image by adding properties such as flip,rotate thus increasing the dataset. • To train the model CNN with a little modification has been used. In our model, once the input image is processed into an array, it is being sent into MobileNet instead of convolution layer, followed by pooling. • It is then flattened and the output is given as masked or unmasked. MobileNet is being used in this model as it is very fast in processing compared to Cnn and it uses lesser parameters. • The initial weights of the imagenet model are going to be changed for yielding better results and accuracy. The pooling is done with a poolsize(7,7) and then is flattened. A dense layer with 12.8 neurons is added. Dropout is used to avoid non linear use cases. • The optimizer being used in the model is Adam optimizer which is a go to for image datasets. • After training of the model, a classification report is generated and saved using the predict method and the graph is plotted between training loss and accuracy using matplotlib. • The trained model is saved and applied on the face detection model in next phase.

3. TRAINING

• The training image generator for data augmentation is constructed, by which mul- tiple images can be created using a single image by adding properties such as flip,rotate thus increasing the dataset.

• To train the model CNN with a little modification has been used. In our model, once the input image is processed into an array, it is being sent into MobileNet instead of convolution layer, followed by pooling.

• It is then flattened and the output is given as masked or unmasked. MobileNet is being used in this model as it is very fast in processing compared to Cnn and it uses lesser parameters.

• The initial weights of the imagenet model are going to be changed for yielding better results and accuracy. The pooling is done with a poolsize(7,7) and then is flattened. A dense layer with 12.8 neurons is added. Dropout is used to avoid non linear use cases.

• The optimizer being used in the model is Adam optimizer which is a go to for image datasets.

• After training of the model, a classification report is generated and saved using the predict method and the graph is plotted between training loss and accuracy using matplotlib.

• The trained model is saved and applied on the face detection model in next phase.

Under Phase 2:

Under this phase the mask detector model that is generated in phase1 is applied on the facenet model to detect mask during livestream.

1. APPLYING THE MODEL IN LIVESTREAM
2. • To apply the model in the camera i.e livestream, firstly the face should be de- tected using readnet function of dnn module in open CV library and the previously trained model must be loaded.
3. • Every frame in the video is resized to 400 pixels. The model detects the face and predicts if the person is wearing a mask or not and mentions it using the color green for mask and red for no mask with the help of color=(b,g,r) of opencv. • An additional function of percentage of prediction of the position of the mask is also added to the model. • Thus the model returns prediction and location.

V. ALGORITHMS AND LIBRARIES:

Deep Convolutional Neural Network (DCNN):

DCNN is not just a deep neural network with many hidden layers, but actually, it is a deep network that mimics the way the human brain's visual cortex processes and recognizes images [21]. A given input image is processed by assigning relevant weights (learnable parameters) to various parts of the image and then making distinctions between the various characteristics. It is substantially less preprocessing and time-consuming than other classification methods when compared to DCNN. While traditional techniques necessitate the creation of filters by manually, DCNN can learn to create these filters with sufficient training. The DCNN architecture that we used in our research is depicted in Figure 4. These five layers include convolutional layers, average-pooling layers, and one fully connected layer. Convolutional layers and avg-pooling layers are included in this network. For every convolution layer, the layer is convolved with their respective kernel size and after every convolution; Rectified Linear Unit (ReLU) activation function is added. ReLU is used for filtering information that propagates forward through the network. After every convolution, the avg-pooling operation takes place. The fully connected layer also known as the classification layer includes the flattening process. Flattening converts the matrix found after the last avg-pooling into a single column matrix for inputting it to the final output layer. At the last output layer, a "softmax" activation function is used that predicts a multinomial probability distribution

MobileNetV2:

MobileNetV2 is a Google-based developed architecture that is pertained on 1.4 million images of 1000 classes [19]. It is an advanced DCNN architecture that performs well on mobile devices. In MobileNetV2, we do not have to train the model from scratch, we only change the last output layers according to our domain. The architecture of MobileNetV2 is based on its previous version (i.e., MobilenetV1). To preserve the information, it introduced a new structure named "inverted residual." The problem of information destroying in convolution blocks by a nonlinear layer applies the technique of Depthwise Separable Convolution (DSC) by using a linear bottleneck layer [22]. Figure 5 shows the basic architecture of MobileNetV2.

Python:

It has a large and broad library and provides a rich set of modules and functions for rapid application development. GUI Programming Support: Graphical user interfaces can be developed using Python.

Opencv :

It stands for Open Source Computer Vision Library. This library consists of around 2000+ optimized algorithms that are useful for computer vision and machine learning.

TensorFlow:

It is a Python-friendly open-source library for numerical computation that makes machine learning and developing neural networks faster and easier TensorFlow allows developers to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes.

VI. SYSTEM TESTING:

Testing is centered on the following items:

Valid Input: identified classes(image formats/video formats) of valid input must be accepted. Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

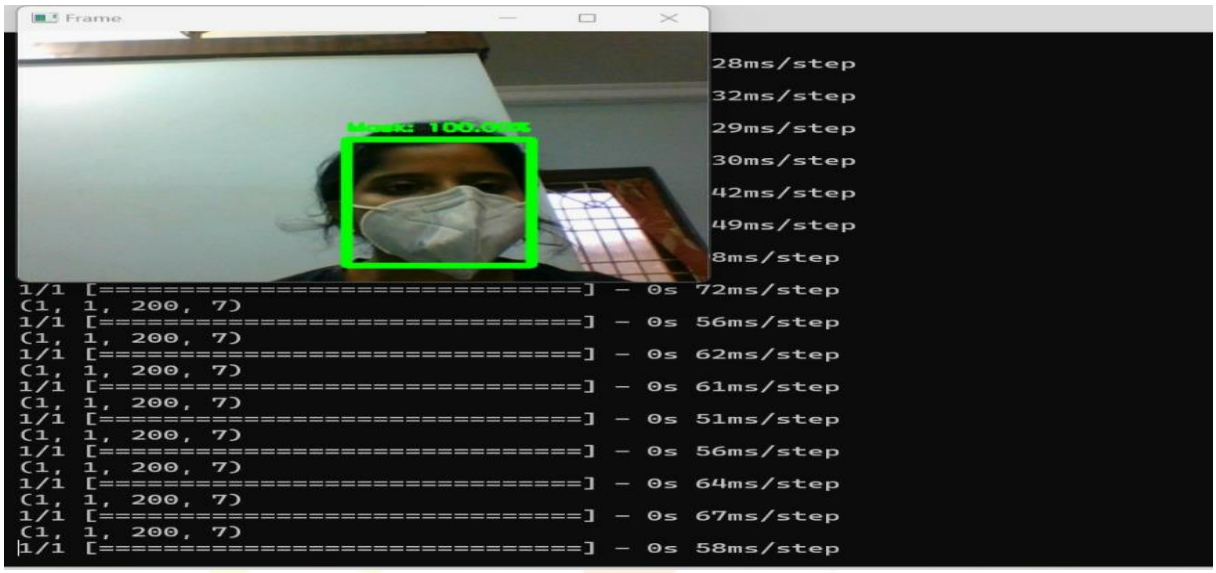
Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of different tests are focused on requirements, key functions, or special test cases.

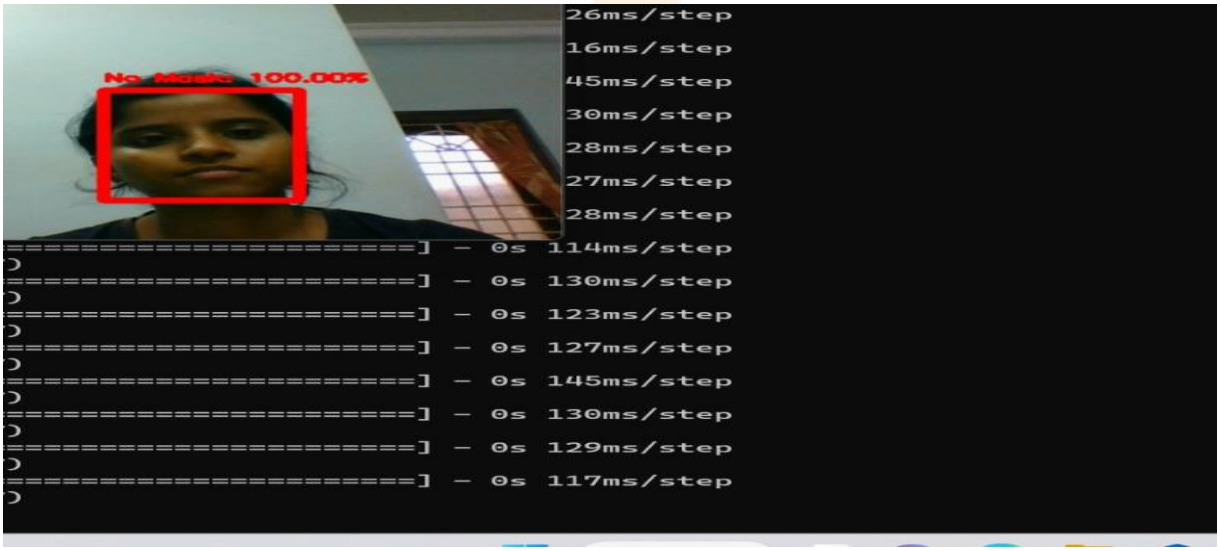
In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before all types of testing are complete, additional tests are identified and the effective value of current tests is determined.

VII. RESULTS AND DISCUSSION

Based on the implementation, the output of the module is expected to distinguish people wearing masks properly and those who are not wearing masks. After training and testing the model is implemented to check whether the people are wearing face masks accurately in the real time video. By making slight changes in the weights while train- ing the efficacy and relevance of the result have improved. This model has been further tested using live video streams and it has shown highly accurate results. The model is built with the MobileNetV2 architecture, which is computationally efficient and lightweight, making it easier to deploy in embedded devices .The dataset is partitioned into training and testing sets by retaining a rational proportion of different classes. The dataset contains 3833 samples in total, with 80 percent of them being used in the training process and 20percent in the testing phase. There are 3066 and 767 images in the training and research datasets, respectively. The built architecture is trained for 40 epochs. The file which was given as an input for validation was successfully analyzed and the results obtained were accurate. The output shows how accurately the people in the live video stream is wearing their face mask.



The figure above shows a person wearing mask properly



The figure above shows a person wearing no mask.

VIII. CONCLUSION

Based on the implementation, the output of the module is expected to distinguish people wearing masks properly and those who are not wearing masks. After training and testing the model is implemented to check whether the people are wearing face masks accurately in the real time video. By making slight changes in the weights while train- ing the efficacy and relevance of the result have improved. This model has been further tested using live video streams and it has shown highly accurate results. The model is built with the MobileNetV2 architecture, which is computationally efficient and lightweight, making it easier to deploy in embedded devices .The dataset is partitioned into training and testing sets by retaining a rational proportion of different classes. The dataset contains 3833 samples in total, with 80 percent of them being used in the training process and 20percent in the testing phase. There are 3066 and 767 images in the training and research datasets, respectively. The built architecture is trained for 40 epochs. The file

which was given as an input for validation was successfully analyzed and the results obtained were acurate. The output shows how accurately the people in the live video stream is wearing their face mask.

IX. ACKNOWLEDGEMENT

Without mentioning the individuals who made it possible and whose persistent leadership and commitment crown all the efforts with success, the satisfaction that comes with the successful completion of any endeavor would be lacking. We are grateful to the administration of our college and our project advisor, Dr.P.Himakeerthi mam for providing us with the tools we needed to complete the project. We appreciate the professionalism and internal assistance provided by our project guide, who assisted us in turning the project into a success. We would like to take this time to thank everyone who has supported us in any way in getting our effort to where it is today. Finally, I want to express my sincere gratitude to my family for their unwavering support and assistance throughout the academic years and for their continued support and encouragement for the completion of the project.

X. REFERENCES

[1] A. Rota, M. S. Oberste, S. S. Monroe, W. A. Nix, R. Campagnoli,J. P. Icenogle, S. Penaranda, B. Bankamp,K.Maher, M.-h. Chenet al., “Characterization of a novel coronavirus associated with severe acute respiratory syndrome,”science,vol. 300, no. 5624, pp. 1394–1399, 2003.

[2] Vinitha, V., Velantina, V. (2020). Covid-19 Facemask Detection With Deep Learning and Computer Vision. Int. Res. J. Eng. Technol, 7(8), 3127-3132.

[3] Z. A. Memish, A. I. Zumla, R. F. Al-Hakeem, A. A. AlRabeeah, and G.

[4] M. Stephens, “Family cluster of middleeast respiratory syndrome coronavirus infections,”New England Journal of Medicine, vol. 368, no. 26, pp.2487–2494, 2013.

[5] C. Li, Y. Diao, H. Ma and Y. Li., 2008. “A Statis-tical PCA Method for Face Recognition”, Available from: https://ieeexplore.ieee.org/abstract/document/4740022/citationscitations

[6] Y. Liu, A. A. Gayle, A. Wilder-Smith, and J. Rocklo¨v, “The reproductive number of covid-19 is higher compared to sars corona virus,” Journal of travelmedicine, 2020. [7] Melissa P. Johnston. 2014. “Secondary Data Analysis: A Methodof which the time Has Come” Available from: http://www.qqml-journal.net/index.php/qqml/article/view/169/170