PROJECT BASED LEARNING II ON

FACE MASK DETECTION

SUBMITTED TO THE SAVTRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

Second Year (Computer Engineering)

BY

RASHMI INGLE	**Roll No: 20U223**
SHASHANK JAGTAP	**Roll No: 20U224**
MOUSIN MULANI	**Roll No: 20U234**
AKANKSHA MULIK	**Roll No: 20U235**

Under the guidance of
Prof. Deepa Kulkarni

Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

STES'S SMT. KASHIBAI NAVALE COLLEGE OF ENGINEERING

VADGAON BK, OFF SINHGAD ROAD, PUNE 411041

SAVTRIBAI PHULE PUNE UNIVERSITY

2020 - 21

Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

STES'S SMT. KASHIBAI NAVALE COLLEGE OF ENGINEERING

VADGAON BK, OFF SINHGAD ROAD, PUNE 411041

CERTIFICATE

This is to certify that the project report entitles

"FACE MASK DETECTION"

Submitted by

RASHMI INGLE	**Roll No: 20U223**
SHASHANK JAGTAP	**Roll No: 20U224**
MOUSIN MULANI	**Roll No: 20U234**
AKANKSHA MULIK	**Roll No: 20U235**

is a bonafide work carried out by them under the supervision of Prof. Deepa Kulkarni and it is submitted towards the partial fulfillment of the requirement of SAVTRIBAI PHULE PUNE UNIVERSITY, Pune for the award of the degree (Computer Engineering)

(Prof. Deepa Kulkarni)
Guide
Department of Computer Engineering

(Dr. P. N. Mahalle)
Head,
Department of Computer Engineering

Seal/Stamp of the College

(Dr. A. V. Deshpande)
Principal,
Smt. Kashibai Navale College of Engineering
Pune – 4

Place:
Date:

CONTENTS

1. ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without acknowledging those who have made it possible and to those whose constant encouragement and guidelines has been a source of inspiration throughout the course of this project.

We are very much thankful to our Project Guide **Prof. Deepa Kulkarni** and Head of the Department **Dr. P. N. Mahalle** , for providing their valuable suggestions and constant supervision without which carrying out this project would not have been possible.

We express our sincere gratitude to our **Principal Dr. A. V. Deshpande** for helping us getting this project completed.

Name of students

RASHMI INGLE **Roll No: 20U223**
SHASHANK JAGTAP **Roll No: 20U224**
MOUSIN MULANI **Roll No: 20U234**
AKANKSHA MULIK **Roll No: 20U235**

2. ABSTRACT

COVID-19 pandemic has rapidly affected our day- to-day life disrupting the world trade and movements. Face masks have become a vital element of our daily lives. So, wearing them is important for safety and to control the spread. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customersto wear masks correctly to avail of their services. Therefore,face mask detection has become a crucial task to help global society. This paper presents a simplified approach to achievethis purpose using some basic Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn. The proposed method detects the face from the image correctly and then identifies if it has a mask on it or not. The method attains maximum accuracy.

3. LIST OF FIGURES

4. INTRODCUTION

According to the World Health Organization (WHO)'s official Situation Report – 205, coronavirus disease 2019(COVID-19) has globally infected over 20 million people causing over 0.7 million deaths. Individuals with COVID- 19 have had a wide scope of symptoms reported – goingfrom mellow manifestations to serious illness. Respiratory problems like shortness of breath or difficulty in breathing is one of them. Elder people having lung disease can possess serious complications from COVID-19 illness as they appear to be at higher risk. Persons having respiratory problems can expose anyone (who is in closecontact with them) to infective beads. Surroundings of a tainted individual can cause contact transmission as droplets carrying virus may withal arrive on his adjacent surfaces.

To curb certain respiratory viral ailments, includingCOVID-19, wearing a clinical mask is very necessary. The public should be aware of whether to put on the mask for source control or aversion of COVID-19. Potential points of interest of the utilization of masks lie in reducing vulnerability of risk from a noxious individual during the "pre-symptomatic" period and stigmatization of discrete persons putting on masks to restraint the spread of virus. WHO stresses on prioritizing medical masks and respirators for health care assistants. Therefore, face mask detection has become a crucial task in present global society.

Face mask detection involves in detecting the location of the face and then determining whether it has a mask on it or not. This project report presents a simplified approach to serve the above purpose using the basic Machine Learning (ML) packages such as TensorFlow, Keras, OpenCV.

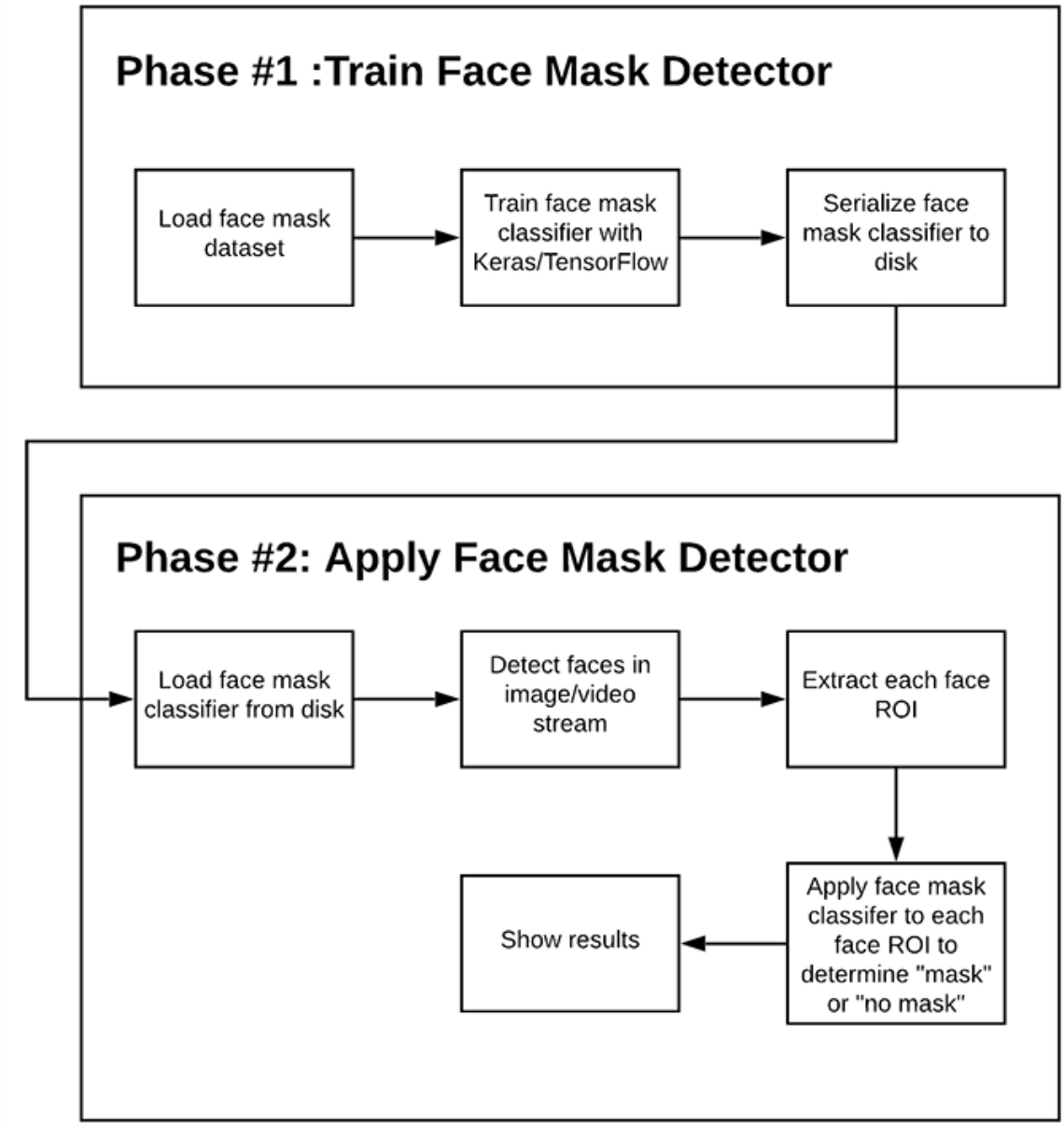
5. MOTIVATION

As the coronavirus is spreading rapidly, the World Health Organization (WHO) has suggested to maintain social distance and to wear facemasks. Due to this everyone should strictly follow the required precautions in order to avoid any possible further spread of the virus. To ensure this we have thought of implementing this as our project to check whether a person is wearing a mask or not. Many other people are working on such projects to improve and make advancements in order to make this concept more effective.

6. PROBLEM STATEMENT

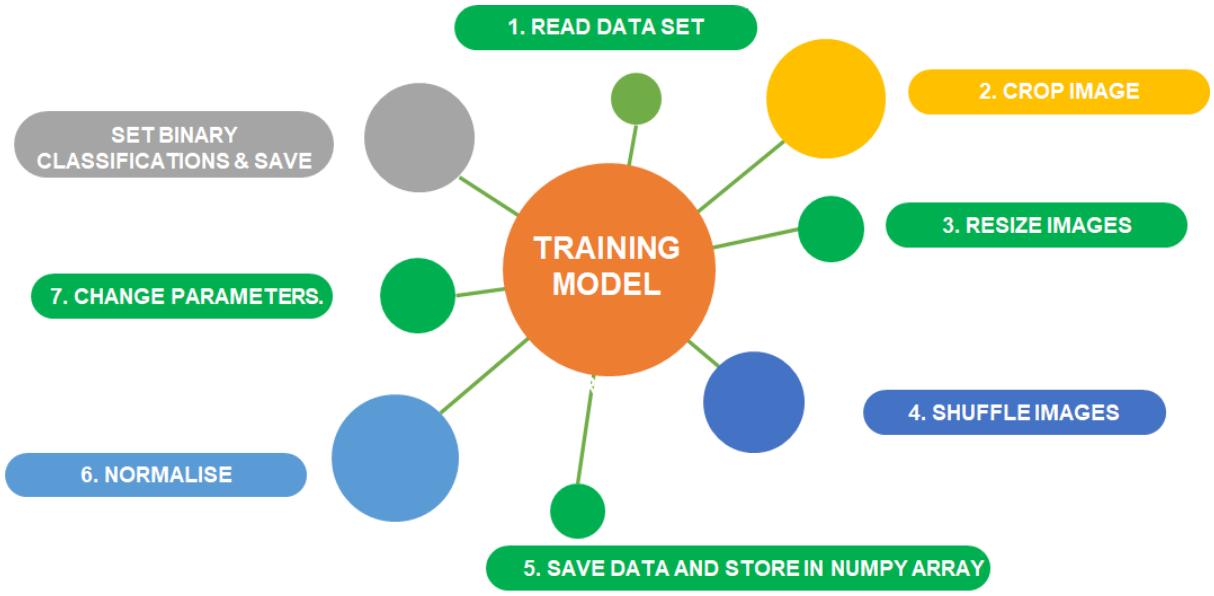
To detect whether a certain person on camera is wearing a mask or not. Providing data sets to a learned model which can identify a person's face in a given image. Dataset Types: People without masks & same People with masks. By passing any image dynamically, it can detect if a person is wearing a mask or not.

7. SYSTEM MODEL

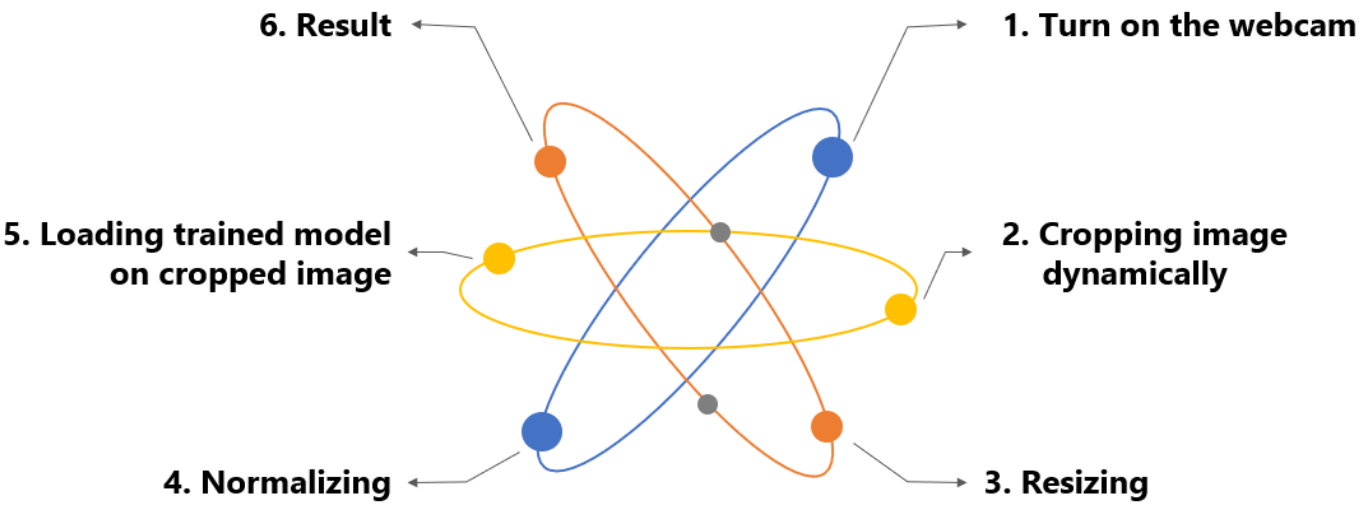


8. METHODOLOGY

Part 1 - Training



Part 2 - Facecam(Actual Identification)



9. SYSTEM ARCHITECTURE

A) Dataset

Two datasets have been used for experimenting the current method.

Dataset 1 consists of 1376 images in which 690 images with people wearing face masks and the rest 686 images with people who do not wear face masks. Fig. 1 mostly contains front face pose with single face in the frame and with same type of mask having white color only. Dataset 2 from Kaggle consists of 853 images and its countenances are clarified either with a mask or without a mask. In fig. 2 some face collections are head turn, tilt and slant with multiple faces in the frame and different types of masks having different colors as well.

B) Incorporated Packages

1. TensorFlow

TensorFlow, an interface for expressing machine learning algorithms, is utilized for implementing ML systems into fabrication over a bunch of 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]. In the proposed model, the whole Sequential CNN architecture (consists of several layers) uses TensorFlow at backend. It is also used to reshape the data (image) in the data processing.

2. Keras

Keras 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 .

All the layers used in the CNN model are implemented using Keras. Along with the conversion of the class vector to the binary class matrix in data processing, it helps to compile the overall model.

3. OpenCV

OpenCV (Open Source Computer Vision Library), an opensource 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.

C) Data Processing

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

D) *Data Visualization:*

Data visualization is the pro- cess of transforming abstract data to meaningful representa- tions using knowledge communication and insight discovery through encodings. It is helpful to study a particular patternin the dataset.

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 theitems in each passed iterator is paired together consequently. The mapped variable *label dict* looks like: *{*'with mask': 0, 'without mask': 1*}*

E) *Conversion of RGB image to Gray image:*

Modern descriptor-based image recognition systems regularly workon grayscale images, without elaborating the method used to convert from color-to-grayscale. This is because the color- to-grayscale 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 grayscale rationalizes the algorithm and diminishes the computational requisites, it is utilized for extracting descriptors instead of working on color images instantaneously.

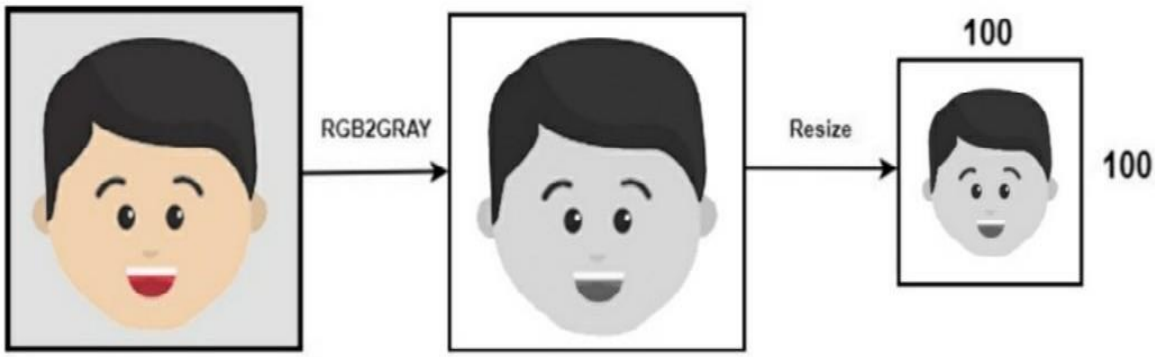


Fig.. Conversion of a RGB image to a Gray Scale image of 100x100 size

We use the function *cv2.cvtColor(input image, flag)* for changing the color space. Here flag determines the type of conversion

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 resizedinto 100 x 100.

F) *Image Reshaping:*

The input during relegation of an image is a three-dimensional tensor, where each channel has aprominent unique pixel. All the images must have identically tantamount size corresponding to 3D feature tensor. How- ever, neither images are customarily coextensive nor their

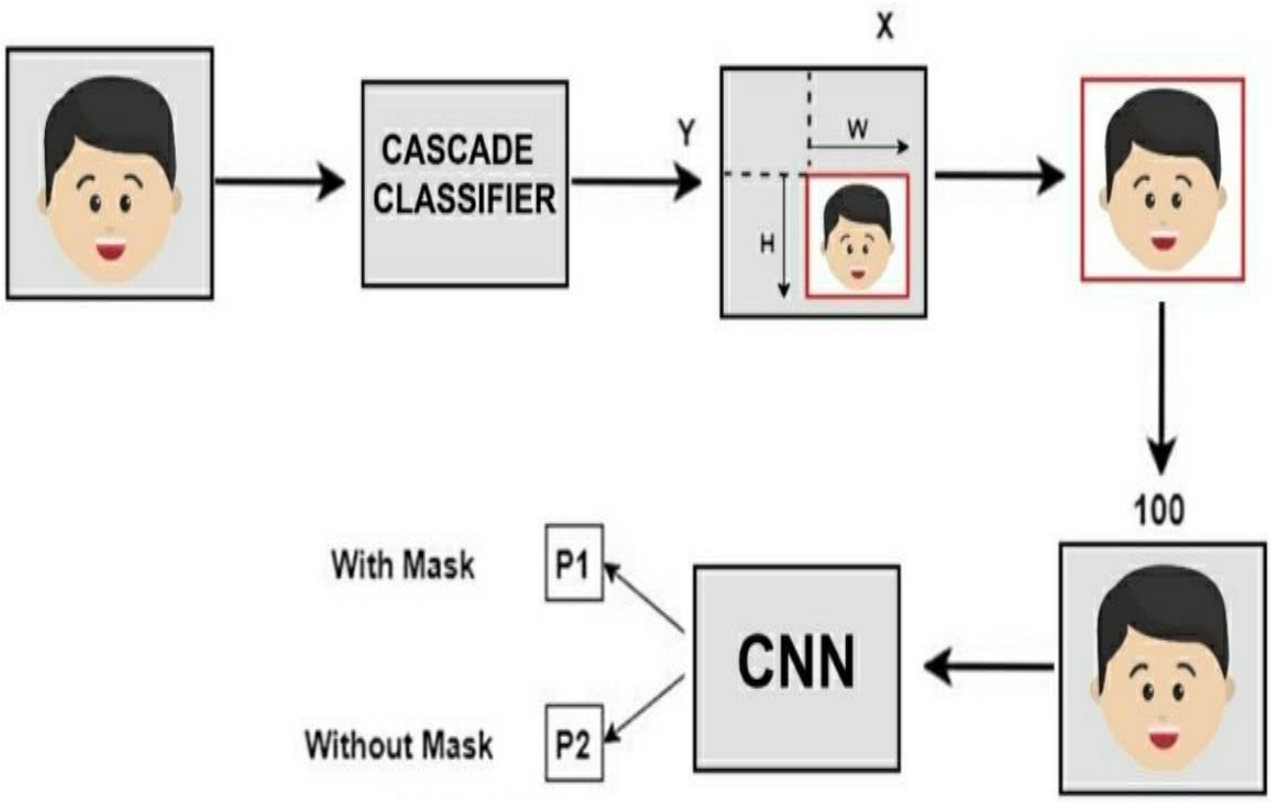
corresponding feature tensors [10]. Most CNNs can only accept fine-tuned images. This engenders several problems throughout data collection and implementation of model. However, reconfiguring the input images before augmenting them into the network can help to surmount this constraint. [11].

The images are normalized to converge the pixel range between 0 and 1. Then they are converted to 4 di- mensional arrays using *data=np.reshape(data,(data.shape[0],img size,img size,1)*) where 1 indicates the Grayscale 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.

G) Training of Model

Splitting the data and training the CNN model: After setting the blueprint to analyze the data, the modelneeds to be trained using a specific dataset and then tobe tested against a different dataset. A proper model andoptimized *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 monitoredusing *ModelCheckpoint*. Next, the images in the training set and the test set are fitted to the Sequential model. Here, 20%of the training data is used as validation data. The model is trained for 20 epochs (iterations) which maintains a trade-offbetween accuracy and chances of overfitting.

10. SYSTEM DIAGRAM



11. SYSTEM REQUIREMENT

Anaconda Navigator

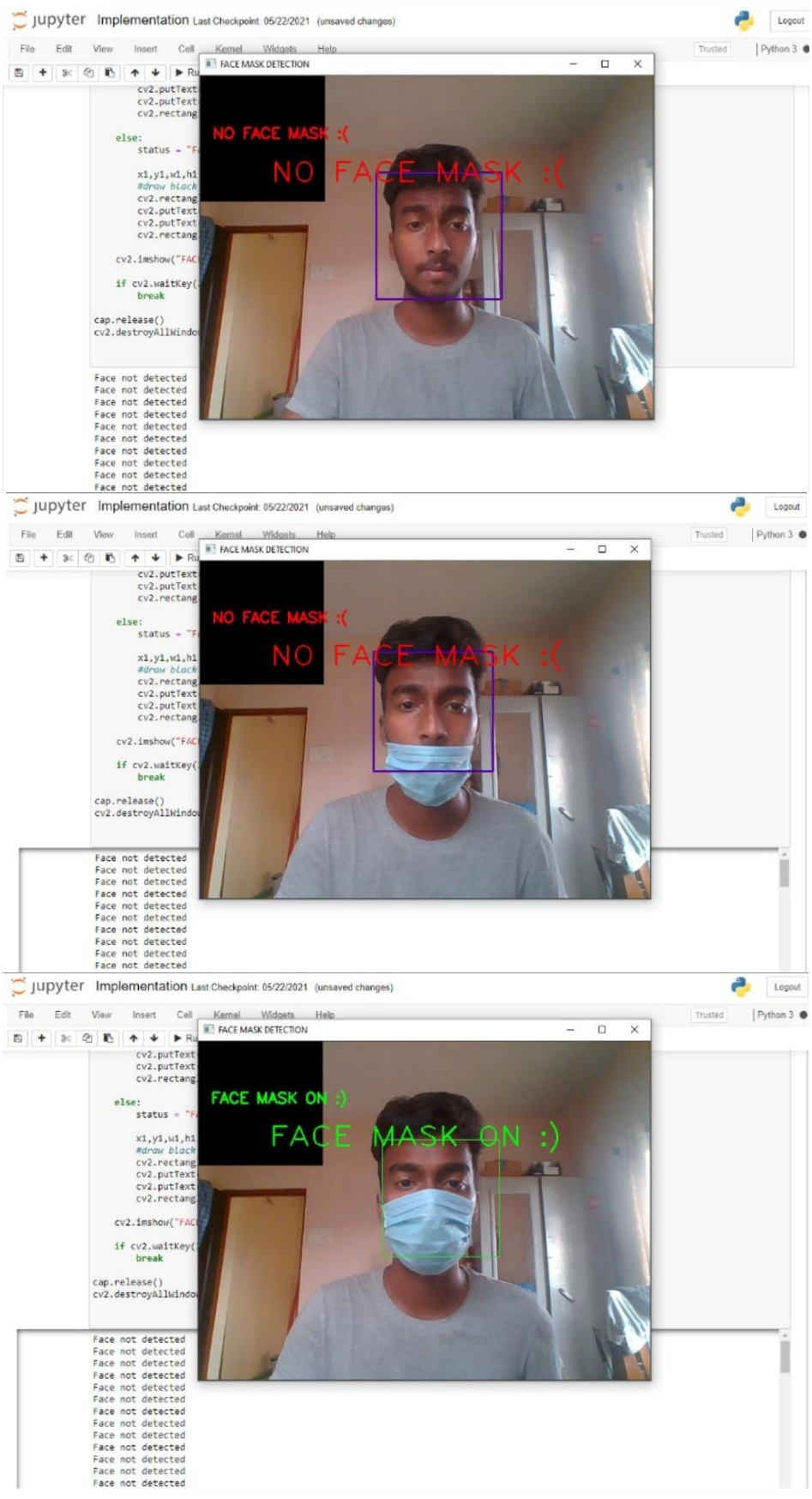
Jupyter Notebook

Python 3

Libraries

- Open CV (for overlaying of text & face re cognition, we use CV-2)
- Tensor-flow(For transfer learning, keras)
- NumPy array(For storing data)
- Import OS(to link to our computer)

12. OUTPUT


jupyter Implementation Last Checkpoint: 05/22/2021 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
FACE MASK DETECTION
NO FACE MASK :(
NO FACE MASK :(
else:
status = "F
x1,y1,w1,h1
#draw black
cv2.rectang
cv2.putText
cv2.putText
cv2.rectang
cv2.imshow("FAC
if cv2.waitKey(
break
cap.release()
cv2.destroyAllWindo
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
jupyter Implementation Last Checkpoint: 05/22/2021 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
FACE MASK DETECTION
NO FACE MASK :(
NO FACE MASK :(
else:
status = "F
x1,y1,w1,h1
#draw black
cv2.rectang
cv2.putText
cv2.putText
cv2.rectang
cv2.imshow("FAC
if cv2.waitKey(
break
cap.release()
cv2.destroyAllWindo
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
jupyter Implementation Last Checkpoint: 05/22/2021 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
FACE MASK DETECTION
FACE MASK ON :)
FACE MASK ON :)
else:
status = "F
x1,y1,w1,h1
#draw black
cv2.rectang
cv2.putText
cv2.putText
cv2.rectang
cv2.imshow("FAC
if cv2.waitKey(
break
cap.release()
cv2.destroyAllWindo
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected
Face not detected

13. RESULTS

The model is trained, validated and tested upon two datasets. Corresponding to dataset 1, the method attains accuracy up to 95.77% (shown in fig. 7). Fig. 6 depicts how this optimized accuracy mitigates the cost of error. Dataset2 is more versatile than dataset 1 as it has multiple faces in the frame and different types of masks having different colors as well. 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 correspondingto dataset 2. One of the main reasons behind achieving this accuracy lies in *MaxPooling*. It provides rudimentarytranslation invariance to the internal representation along with the reduction in the number of parameters the modelhas to learn. This sample-based discretization process down-samples the input representation consisting of image, by reducing its dimensionality. Number of neurons has the optimized value 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 outthe main portion (face) of the image to detect the existenceof mask correctly without causing over-fitting.

The system can efficiently detect partially occluded faces either with a mask or hair or hand. It considers the occlusion degree of four regions – nose, mouth, chin and eye to differentiate between annotated mask or 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.

14. CONCLUSION AND FURTHER WORK

In this project report, 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. It can be used for a variety of applications. Wearing a mask may be obligatory in the near future, considering the Covid-19 crisis. Many public service providers will ask the customers to wear masks correctly to avail of their services. The deployed model will contribute immensely to the public health care system. In future it can be extended to detect if a person is wearing the mask properly or not. The model can be further improved to detect if the mask is virus prone or not i.e. the type of the mask is surgical, N95 or not.

15. REFERENCES

[1] W.H.O., "Coronavirus disease 2019 (covid-19): situation report, 205".2020"Coronavirus Disease 2019 (COVID-19) – Symptoms",Centers for Disease Control and Prevention, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html. 2020.

[2] "Coronavirus — Human Coronavirus Types — CDC", Cdc.gov, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/types.html. 2020.

[3] W.H.O., "Advice on the use of masks in the context of COVID-19: interim guidance", 2020.

[4] M. Jiang, X. Fan and H. Yan, "RetinaMask: A Face Mask detector",

[5] 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 ComputerSciences 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 Analyticsin 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 convolu- tional 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 con- volutional neural network: Some common and uncommon meth- ods", 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 intoImaging, vol. 9, no. 4, pp. 611-629, 2018. Available: 10.1007/s13244-018-0639-9.

[13] "Guide to the Sequential model - Keras Documentation", Faroit.com, 2020. [Online]. Available: https://faroit.com/keras-docs/1.0.1/getting- started/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.