FACE MASK DETECTION

Using Machine Learning and Deep Learning

Affiliated to GGSIP University, New Delhi
Approved by AICTE & Council of Architecture

Project Report

Submitted By

Name	Enrollment No
Saiyam Jain	41218002717
Mayank Goyal	35118002717
Deepak Singh	42918002717
Abhishek Aswal	43318002717

Under the guidance of

Ms. Upasna Joshi

Assistant Professor

Department of Computer Science & Engineering

Delhi Technical Campus

<u>Declaration By the Students</u>

We student(s) of B.Tech in Computer Science & Engineering branch of Delhi Technical Campus hereby declare that the project entitled **<u>"Face Mask Detection using Machine Learning and Deep Learning"</u>** which is submitted by us to Department of computer science and engineering, delhi technical campus, Greater Noida, in partial fulfillment of requirement for the award of the degree of Bachelor of Technology in Computer Science And Engineering, has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

The Authors attest that permission has been obtained for the use of any copy righted material appearing in the Dissertation / Project report other than brief excerpts requiring only proper acknowledgement in scholarly writing and all such use is acknowledged.

Name of the Student(s)

Saiyam Jain

Mayank Goyal

Deepak Singh

Abhishek Aswal

Place – **<u>Greater Noida</u>**

Date – **<u>08-June-2021</u>**

CERTIFICATE OF ORIGINALITY

On the basis of declaration submitted by **Saiyam Jain, Mayank Goyal, Deepak Singh, Abhishek Aswal**, student(s) of B.Tech(CSE), I hereby certify that the project titled **“Face Mask Detection using Machine Learning and Deep Learning”** which is submitted to, DELHI TECHNICAL CAMPUS, Noida, in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is an original contribution with existing knowledge and faithful record of work carried out by him/them under my guidance and supervision.

To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Date: 08-June-2021

Ms. Upasna Joshi

(Name and Signature of Guide)

Department of Computer Science & Engineering

Delhi Technical Campus, Greater Noida

ACKNOWLEDGEMENT

Firstly, we would like to express our special thanks of gratitude to our teacher and guide Ms. Upasna Joshi who gave us the golden opportunity to do this wonderful project on the topic "Face Mask Detection using Machine Learning and Deep Learning ", which also helped us in doing a lot of Research and gain knowledge about many new things. She encouraged us to work on this project and we are really thankful to her. We are also grateful to our college for giving us the opportunity to work with them and providing us the necessary resources for the project. We are immensely grateful to all involved in this project as without their inspiration and valuable suggestion it would not have been possible to develop the project within the prescribed time.

CONSENT FORM

This is to certify that we Saiyam Jain, Mayank Goyal, Deepak Singh and Abhishek Aswal, students of B. Tech CSE of 2017-2021 presently in the VIII Semester at DELHI TECHNICAL CAMPUS, Greater Noida give our consent to include all our personal details, for all accreditation purposes.

Saiyam Jain - 41218002717

Mayank Goyal - 35118002717

Deepak Singh - 42918002717

Abhishek Aswal - 43318002717

Place – **New Delhi**

Date – **08-June-2021**

DECLARATION FORM (Health, Safety & Plagiarism)

We, students of B. Saiyam Jain, Mayank Goyal, Deepak Singh and Abhishek Aswal, batch 2017-2021, Department of Computer Science and Engineering, Delhi Technical Campus, GGSIP University, New Delhi, hereby declare that we have gone through project guidelines including policy on health and safety, policy on plagiarism etc.

Date: 8Th June’2021

Name and Signatures of the Students

Saiyam Jain

Mayank Goyal

Deepak Singh

Abhishek Aswal

TABLE OF CONTENT

Sr. No.	Content	Page No.
1	INTRODUCTION	8
2	LITERATURE SURVEY	9
3	BACKGROUND STUDY	11
4	METHODOLOGY	14
5	RESULTS	16
6	TEST OUTPUTS	18
7	LIMITATION & FUTURE WORK	19
8	CONCLUSION	20
9	REFERENCES	21

INTRODUCTION

Corona Virus was originated in Wuhan, China at the end of 2019. Since then, it has been spreading like a wild fire in a forest. Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness. The best way to prevent and slow down transmission is to be well informed about the COVID-19 virus, the disease it causes and how it spreads. Protect yourself and others from infection by washing your hands or using an alcohol based rub frequently and not touching your face. The COVID-19 virus spreads primarily through droplets of saliva or discharge from the nose when an infected person coughs or sneezes, so it’s important that you also practice respiratory etiquette (for example, by coughing into a flexed elbow). Millions have been affected and around 37,96,473 have unfortunately passed away as on 08th of June 2021, almost a year since this virus came to existence. People who have this illness can take up to 2 weeks to cure, with the risk of having to suffer additional medical problems causedby it. Children and old people have proved to be at the highest risk to contract the disease, which may even result in death. Hence, it has been made a priority to contain the virus than to cure it. The virus spreads through the air, transmitted by one person to another not only by touch, but also by speaking and coughing. The concern was put forward to WHO(World Health Organization) which suggested that face masks and social distancing is the answer to it, until a cure is invented. Putting a face mask on can reduce the risk of getting infected by a great extent, notonly to the one wearing it but also to the others that he comes in contact with. Wearing masks every time we go out is something we can do with little effort thatcan effectively save lives, and that is precisely why it is in so much demand at this point of time.

We propose a Face Mask Detection project that consists of 2 phases, namely training and deployment. The first stage detects human faces, while the second stage uses deep learning to firstly, identify the ROI(Region Of Interest) being the person's face and secondly identify the faces detected in the first stage as either 'With Mask' or 'Without Mask' and draws boundary of colors either green or red, depending on the output. The project takes JPG and PNG files as inputs, but it has also been tested on videos. The project can give accurate results if set up with a CCTV camera to track people without masks to ensure the safety and wellbeing of others, thus help controlling the spread of the virus.

LITERATURE SURVEY

Object detection is one of the trending topics in the field of image processing and computer vision. Ranging from small scale personal applications to large scale industrial applications, object detection and recognition is employed in a wide range of industries. Some examples include image retrieval,security and intelligence, OCR, medical imaging and agricultural monitoring.

In object detection, an image is read and one or more objects in that image are categorized. The location of those objects is also specified by a boundary called the bounding box. Traditionally,researchers used pattern recognition to predict faces based on prior face models.

A breakthrough face detection technology then was developed named as Viola Jones detector that was an optimized technique of using Haar, digital image features used in object recognition. However, it failed because it did not perform well on faces in dark areas and non-frontal faces. Since then, researchers are eager to develop new algorithms based on deep learning to improvethe models.

Deep learning allows us to learn features with end to end manner and removing the need to use prior knowledge for forming feature extractors.

Like object detection, face detection adopts the same architectures as one-stage and two-stage detectors, but in order to improve face detection accuracy, more face-like features are being added. However, there is occasional research focusing on face mask detection. Some already existing face mask detectors have been modeled using OpenCV, Pytorch Lightning, MobileNet, RetinaNet and Support Vector Machines.

In this project, embedded devices are used for deployment that could reduce the cost of manufacturing. MobileNetV2 architecture is used as it is a highly efficient architecture to apply on embedded devices with limited computational capacity such as Google Coral, NVIDIA Jetson Nano. This project performed well, however, if a large portion of the face is occluded by the mask, this model could not detect whether a person is wearing a mask or not.

The dataset used to train the face detector did not have images of people wearing face masks as a result, if thelarge portion of faces is occluded, the face detector would probably fail to detectproperly. To get rid of this problem, they should gather actual images of people wearing masks rather than artificially generated images.

BACKGROUND STUDY

Machine Learning

Machine Learning or ML is a study of computer algorithms that learns and enhance automatically through experience. It seems to be a subset of artificial intelligence. A machine learning algorithm builds a mathematical model based on "training data", in order to make decisions or predictions without being explicitly programmed to do so.

Machine learning algorithms are used in a variety of applications from email filtering to computer recognition, where it is difficult or impossible to develop general skills to perform the required tasks. These studies are closely related to computer statistics, which focus on computer-generated domain. The data prediction and mining is a coherent field of study, focusing on the analysis of experimental data by unsupervised learning. In its application to business problems, machine learning is also called predictive analytics.

Supervised learning, also known as supervised machine learning, is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately. This occurs as part of the cross validation process to ensure that the model avoids overfitting or underfitting. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox. Some methods used in supervised learning include neural networks, naïve bayes, linear regression, logistic regression, random forest, support vector machine (SVM), and more.

Machine learning is a fast-growing trend in the health care industry, thanks to the advent of wearable devices and sensors that can use data to assess a patient's health in real time. The technology can also help medical experts analyze data to identify trends or red flags that may lead to improved diagnoses and treatment.

Deep Learning

Deep learning methods aim to learn feature hierarchies with a high-level hierarchy which is structured by the construction of lower-level features. Automated learning at multiple levels of extraction allows a system to learn complex tasks to do input mapping directly from data to output, without relying entirely on man-made features. Deep learning algorithms capture unspecified structure inside the input distribution to find better characterization frequently at multiple levels, with high-level learning features in the context of low-level features.

Inputs and outputs are in-depth study of the analog Excel problem domain. Meaning, they are not some size in table format, but they are pixel data, text data documents or data from audio files. Deep learning empower logical and mathematical models to find representations of data with numerous levels of abstraction, multiple processing layers.

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing.

However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.

When deep learning is used to detect fraud, it will leverage several signals, such as IP address, credit score, retailer, or sender, to name a few. In the first layer of its artificial neural network, it will analyze the amount sent. In a second layer, it will build on this information and include the IP address, for example. In the third layer, the credit score will be added to the existing information, and so forth until a final decision is made.

OpenCV

OpenCV is a library which is use to develop computer based real-time applications. It majorly focuses on analysis including features like image processing, video capture and object detection and face detection.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

We use the OpenCV library to execute infinite loops using our webcam, which detects faces using cascade classifications. The library has over 2000 optimized and advance algorithms for computer vision based machine learning. These algorithms can be used for face detection and recognition, object detection, classifying human movements in video, tracking camera actions, tracking objects, taking 3D objects, adaptive thresholding and assembling together to produce high resolution image. It can also be useful in finding similar images from the database, removal of red eyes from photos taken with flash, follow the facial movements, and add tags to transition with advanced reality. It is continuously adding new modules to the latest algorithms from machine learning.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV’s deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDAand OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

Tensorflow

Tensor Flow is a standalone and open-source software library for Dataflow for a variety of tasks and a wide variety of programming. It is also used for machine learning applications such as the Symbolic Mathematics Library, and Neural Networks. TensorFlow is a great system for handling all aspects of a machine learning system. However, this class focuses on using the unique Tensor Flow API to train and deploy machine learning models.

TensorFlow provides a collection of workflows with intuitive, high-level APIs for both beginners and experts to create machine learning models in numerous languages. Developers have the option to deploy models on a number of platforms such as on servers, in the cloud, on mobile and edge devices, in browsers, and on many other JavaScript platforms. This enables developers to go from model building and training to deployment much more easily. Easy model building Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging. Robust ML production anywhere Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.

Powerful experimentation for research

A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

How Google uses TensorFlow

TensorFlow is used by many internal Google products and teams including: Search, Gmail, Translate, Maps, Android, Photos, Speech, YouTube, and Play.

We used TensorFlow and Keras to train the classifier to automatically identify if a person is wearing a mask. Since reference implementation runs on single devices, TensorFlow is able to runs on multiple Processing Units and GPUs having extensions regarding general use.

Keras

Keras is an API for high level neural networking. It follows best practices to reduce the major burden and provides consistent and flexible APIs that reduce the number of user actions required for normal usage situations and provide clear and actionable error messages. It is written in Python programming language and has a large developer community and support.

Keras includes several implementations of commonly used neural-network architecture, such as hosting devices to simplify the coding required to write layers, targets, optimizers, activation tasks, and an intensive neural network. It make easy to work with image and text data. The Keras models are easily deployable among various platforms.

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling.

Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep-learning models on clusters of Graphics processing units (GPU) and tensor processing units (TPU).

MobileNet V2

MobileNet is a Convolution Neural Network architecture model for various categorical classification and object detection work. This architecture is easily executable on mobile devices with a high rate of accuracy when compare to other light weighted CNN architectures. Also, it is ideal for mobile devices that do not have GPUs and highly embedded computational efficiency. It is significantly faster and accurate on results. It is also well suited for web or browsers as the browser has limitations on computing, graphic processing and storage.

We have used the MobileNetV2 architecture, for it computational efficiency, making it easy to set up models for embedded systems (Raspberry Pi, Google Corel, Jetson, Nano, etc.).

Depthwise Separable Convolution is introduced which dramatically reduce the complexity cost and model size of the network, which is suitable to Mobile devices, or any devices with low computational power. In MobileNetV2, a better module is introduced with inverted residual structure. Non-linearities in narrow layers are removed this time. Each layer has batch normalization and the activation function is ReLU6. However, the output of the projection layer does not have an activation function applied to it. Since this layer produces low-dimensional data, the authors of the paper found that using a non-linearity after this layer actually destroyed useful information.

METHODOLOGY

This proposed model focuses on identifying face mask in a person through an image or video stream with the help of Deep Learning and Machine Learning using Keras, TensorFlow, OpenCV and the Scikit-Learn library. We have designed our model in two phases:

1. Training (Training the model on the dataset using TensorFlow & Keras)

Image pre-processing:

To create this dataset, following steps have been taken Taking normal images of faces.

Then creating a custom computer vision Python script to add face masks to them, thereby creating an artificial (but still real-world applicable) dataset

Facial landmarks allow us to automatically infer the location of facial structures, including:

- Eyes
- Eyebrows
- Nose
- Mouth
- Jawline

To use facial landmarks to build a dataset of faces wearing face masks, we needed to first start with an image of a person not wearing a face mask.

From there, we apply face detection to compute the bounding box location of the face in the image.

The next step is to apply face detection. Here we’ve used a deep learning method to perform face detection with OpenCV.

Once we know where in the image the face is, we can extract the face Region of Interest (ROI).

The next step is to extract the face ROI with OpenCV and NumPy slicing.

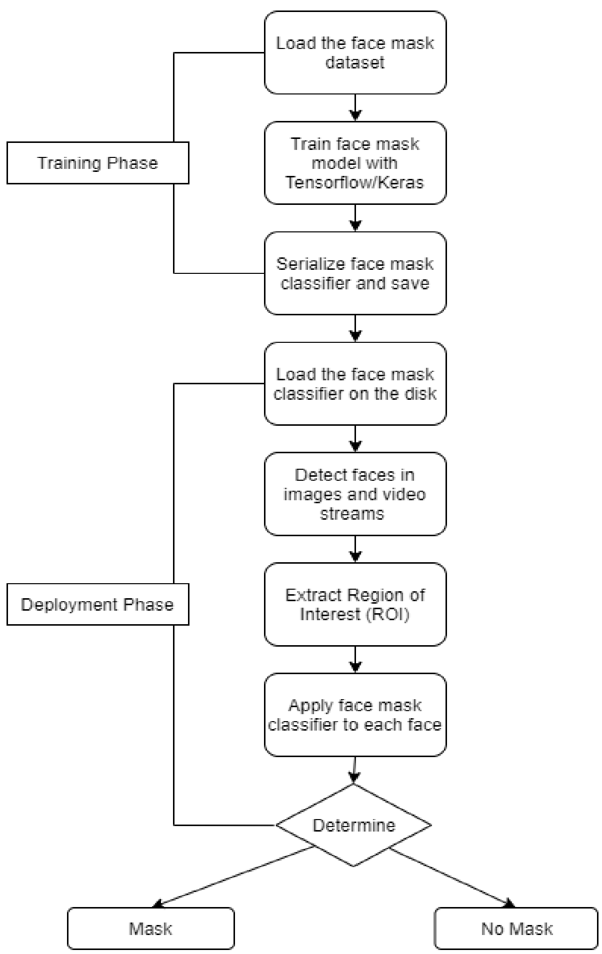
And from there, we apply facial landmarks, allowing us to localize the eyes, nose, mouth, etc.

Then, we detect facial landmarks using lib so that we know where to place a mask on the face. Next, we need an image of a mask (with a transparent background).

This mask will be automatically applied to the face by using the facial landmarks (namely the points along the chin and nose) to compute where the mask will be placed. The mask is then resized and rotated, placing it on the face.

We can then repeat this process for all of our input images, thereby creating our artificial face mask dataset.

2. Deployment (Loading the trained model and applying detector over images/livevideo stream)



Flowchart showing the training and deployment phase.

The dataset is loaded first in the training phase. Training and modeling are streamlined during the training phase. After serializing face mask classifier to the disk, model is loaded to detect the face mask on the images or real-time video.

The model will calculate the ROI (Region of Interest) for the determination. We then compute bounding box value for a particular face and ensure that the box falls within the boundaries of the image. We then determine the class label based on predictions returned by the mask detector model and colors are assigned for interpretation. The “Green” will be for with mask and “Red” will be for without mask. Once all detection is executed we will display the output.

We have used MobileNetV2 architecture which is a accurate and efficient and can be applied to embedded devices. We have specified the constants i.e. Initial Learning rate to be 1e-4, batch size to be 32 and no of epochs to train the model as 20. After preprocessing, the input images are resized as 224 x 224 x 3 pixels and then compiled and evaluated on the test set. Face detection is similar to what was discussed earlier. A special frame is held in place by a stream and re-shaped. Then the face mask detection is processed. The results are displayed on the screen after post processing.

RESULTS

We tried using three different base models for detecting ‘mask’ or ‘no mask’. The exercise was done to find the best fit model in our scenario. The evaluation process consists of first looking at the classification report which gives us insight towards precision, recall and F1 score. The equations of these three metrics are as follows:

Precision = True Positives / Positives + False Positives

Recall = True Positives / Positives + False Negatives

Accuracy = True Positives + True Negatives / Positives + Negatives

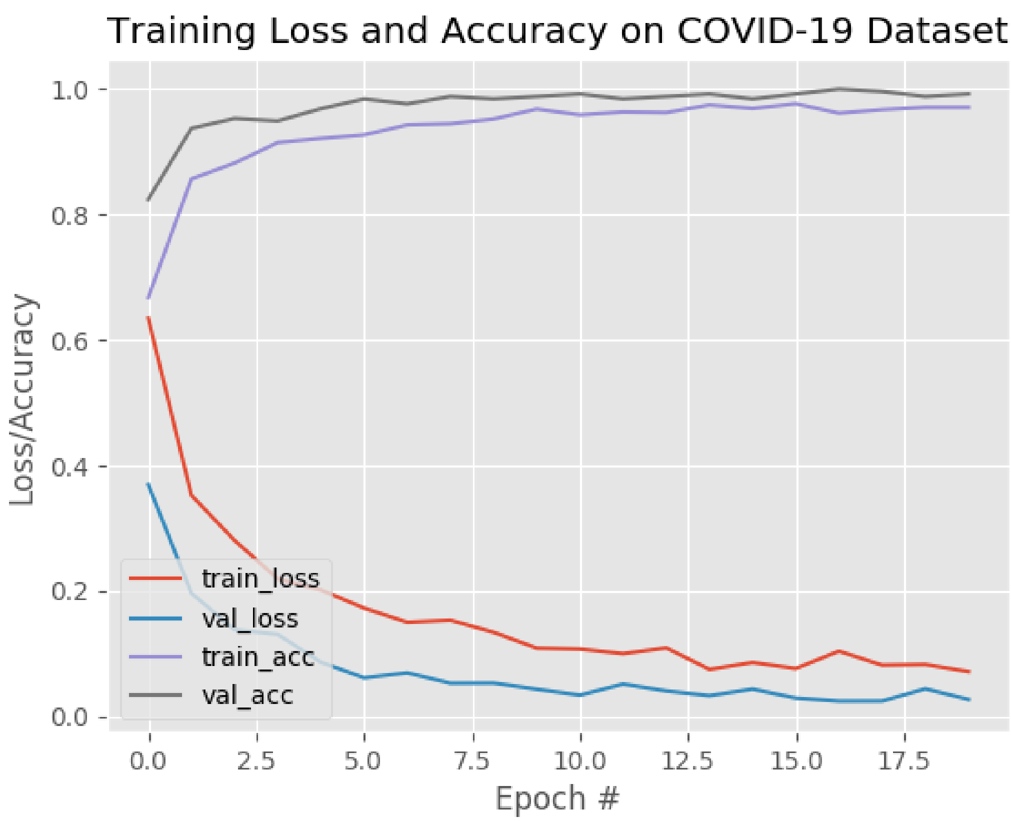
Using these three metrics, we can conclude which model is performing most efficiently. The second part consists of plotting the train loss, validation loss, train accuracy and validation accuracy which also proves helpful in choosing a final model.

MobileNet V2 Loss Function

MobileNetV2 is an architecture of bottleneck depth-separable convolution building of basic blocks with residuals. It has two types of blocks. The first one is a residual block with stride of 1. Second one is also residual block with stride 2 and it is for downsizing.

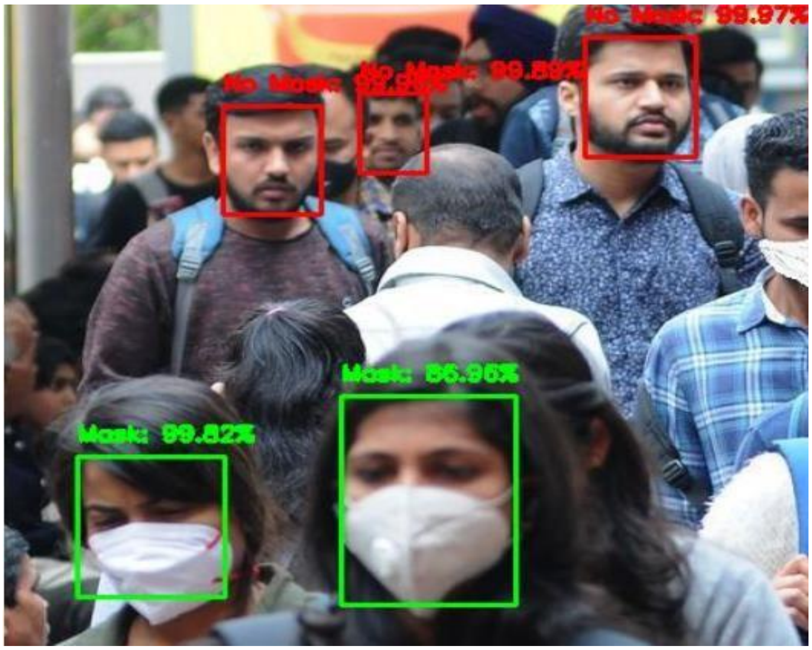
Evaluation Table for Trained Model

	Precision	Recall	F1-Score	Support
With mask	0.99	0.93	0.94	134
Without mask	0.99	0.96	0.94	134
Accuracy			0.98	274
Macro Avg.	0.98	0.98	0.98	274
Weighted Avg	0.98	0.08	0.98	274

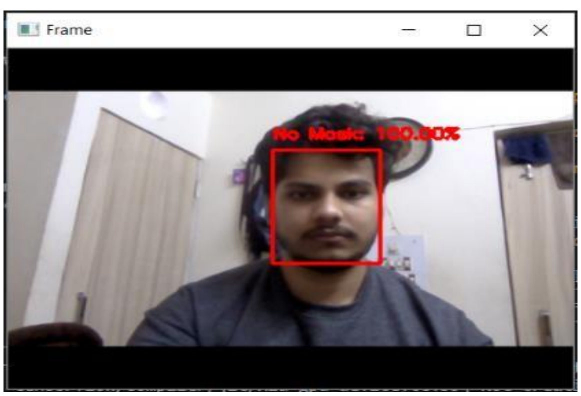


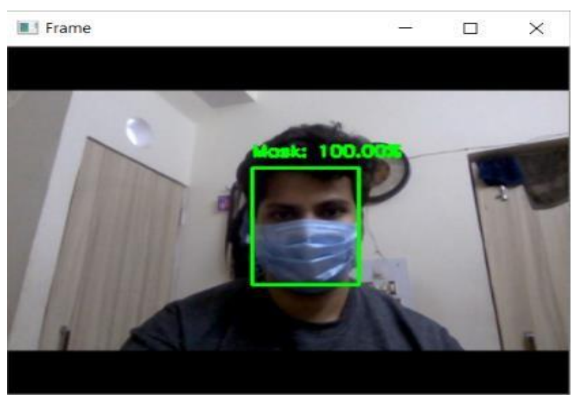
TEST OUTPUTS

As we can see the accuracy obtained by our trained face mask detector model is ~98%. We have got the following output from our model



Output of Face Mask Detector in Uploaded Image





Output of Face Mask Detector in Real time video stream

LIMITATION & FUTURE WORK

There were not many challenges faced but the two problems that were time consuming and made the tasks tedious are discussed as follows. One was the excessive data loading time in Google Colab Notebook while loading the dataset into it. Since the runtime restarting refreshes all the cells, the cell for dataset loading took most of the time while running. Secondly, the access problem in Google Colab Notebook: it did not allow the access of webcam which posed a hurdle in testing images and live video stream through Google Colab Notebook. Therefore, we had to run the code locally on the computer through which we tested the code on the live video stream.

In this work, a deep learning-based approach for detecting masks over faces in public places to curtail the community spread of Coronavirus is presented. The proposed technique efficiently handles occlusions in dense situations by making use of an ensemble of single and two-stage detectors at the pre-processing level. The ensemble approach not only helps in achieving high accuracy but also improves detection speed considerably. Furthermore, the application of transfer learning on pre-trained models with extensive experimentation over an unbiased dataset resulted in a highly robust and low-cost system. The identity detection of faces, violating the mask norms further, increases the utility of the system for public benefits.

Finally, the work opens interesting future directions for researchers. Firstly, the proposed technique can be integrated into any high-resolution video surveillance devices and not limited to mask detection only. Secondly, the model can be extended to detect facial landmarks with a facemask for biometric purposes.

More than fifty countries around the world have recently initiated wearing face masks compulsory. People have to cover their faces in public, supermarkets, public transports, offices, and stores. Retail companies often use software to count the number of people entering their stores. They may also like to measure impressions on digital displays and promotional screens. We are planning to improve our Face Mask Detection tool and release it as an open-source project. Our software can be equated to any existing USB, IP cameras, and CCTV cameras to detect people without a mask. This detection live video feed can be implemented in web and desktop applications so that the operator can see notice messages. Software operators can also get an image in case someone is not wearing a mask.

Furthermore, an alarm system can also be implemented to sound a beep when someone without a mask enters the area. This software can also be connected to the entrance gates and only people wearing face masks can come in.

The developed system can detect the live video streams but does not keep a record. Unlike the CCTV camera footage the admin can not rewind, play or pause it. As whenever a strict system is imposed people always try to break it. Hence when a person is detected with no mask, the head of the organization can be notified via mail that so and so person entered without mask. The proposed system can be integrated with databases of respective organizations to keep a record of the person who entered without mask. With more complex functions a screenshot of the person’s face can also be attached to keep it as a proof.

Scope for improvement

To improve our face mask detection model further, we should have used real images instead of artificially generated ones but it was not possible due to the Corona virus itself.

While our artificial dataset worked well in this case, there's no substitute for the real thing.

Secondly, we should also gather images of faces that may "confuse" our classifier into thinking the person is wearing a mask when in fact they are not — potential examples include shirts wrapped around faces, bandana over the mouth, etc.

Unfortunately, due to busy schedules and deadlines we were not able to do the same, but we will be looking forward to it in the future.

As we dug deep for improvement in our project, we found that the problem with this approach is that a face mask, by definition, obscures part of the face. If enough of the face is obscured, the face cannot be detected, and therefore, the face mask detector will not be applied.

To circumvent that issue, you should train a two-class object detector that consists of a with_mask class and without_mask class.

Combining an object detector with a dedicated with_mask class will allow improvement of the model in two respects.

First, the object detector will be able to naturally detect people wearing masks that otherwise would have been impossible for the face detector to detect due to too much of the face being obscured.

Secondly, this approach reduces our computer vision pipeline to a single step — rather than applying face detection and then our face mask detector model, all we need to do is apply the object detector to give us bounding boxes for people both with_mask and without_mask in a single forward pass of the network.

CONCLUSION

We created a face mask detector using Deep Learning, Keras, TensorFlow and OpenCV. We trained it to distinguish between people wearing mask and people not wearing a mask We have used MobileNet V2 classifier with the ADAM optimizer for the best result.

The model is tested with photos and real-time video streams. It detected the face from the images/videos and extracts each individual's face and apply the face mask classifier to it.

As the technology is booming with emerging trends therefore the novel face mask detector which can possibly contribute to public healthcare. The model is trained on an authentic dataset. We used OpenCV, tensor flow, keras and CNN to detect whether people were wearing face masks or not. The models were tested with images and real-time video. The accuracy of the model is achieved and, the optimization of the model is a continuous process and we are building an accurate solution by tuning the hyper parameters. This specific model could be used as a use case for edge analytics. By the developing this system, we can detect if the person is wearing a face mask and allow their entry would be of great help to the society.

Since, we have used the MobileNet V2 architecture we can easily deploy our model to the embedded systems such as Raspberry Pi, Jetson, Google Coral, Nano etc. This can help be very helpful for the society and can possibly contribute to the public healthcare.

REFERENCES

[1] D. Chiang., "Detect faces and determine whether people are wearing mask, 2020. [2] G. Jignesh Chowdary, Narinder Singh Punn, Sanjay Kumar Sonbhadra, Sonali Agarwal "Face Mask Detection using Transfer Learning of InceptionV3",2020

[2] P. Khandelwal, A. Khandelwal, S. Agarwal," Using computer vision to enhance safety of workforce in manufacturing in a post covid world" 2020

[3] Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Peiet al., "Masked face recognition dataset and application" arXiv preprint arXiv:2003.09093, 2020.

[4] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review", IEEE transactions on neural networks and learning systems, vol. 30, no. 11, pp. 3212–3232, 2019.

[5] Kumar, A. Kaur, and M. Kumar, "Face detection techniques: a review,"Artificial Intelligence Review, vol. 52, no. 2, pp. 927–948, 2019.D.-H. Lee, K.-L. Chen, K.-H. Liou, C.-L. Liu, and J.-L. Liu, "Deep learning and control algorithms of direct perception for autonomous driving, 2019.

[6] Qin B. and Li D. , Identifying face mask wearing condition using image super-resolution with classification network to prevent COVID-19.doi: 10.21203/rs.3.rs-28668/v1

[7] Erhan, D., Szegedy, C., Toshev, A., and Anguelov, D. (2014). "Scalable object detection using deep neural networks," in Computer Vision and Pattern Recognition Frontiers in Robotics and AI

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014.

[9] https://www.worldometers.info/coronavirus/coronavirus-death-toll/ for the statistics related to the Corona virus.