A Thesis/Project/Dissertation Report

on

Real Time Face-Mask Detection System

Submitted in partial fulfillment of the

requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. V. Arul
Assistant Professor
Department of Computer Science

Submitted By

19SCSE1010687 - Hrithik Gupta

19SCSE1010445 - Kartavya Chauhan

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER - 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"Real Time Face-Mask Detection System"** in partial fulfillment of the requirements for the award of the Galgotias University-submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **JULY-2021 to DECEMBER-2021**, under the supervision of **Mr.V. ARUL, Assistant Professor**, **Department of Computer Science and Engineering** of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Hrithik Gupta, 19SCSE1010687
Kartavya Chauhan, 19SCSE1010445

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr.V.Arul,
Assistant Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **19SCSE1010687 - HRITHIK GUPTA, 19SCSE1010 - KARTAVYA CHAUHAN** has been held on ________________ and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:

Place:

Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them. We are highly indebted to Mr. V. Gokul Rajan for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards the member of Galgotias University for their kind co-operation and encouragement which help us in completion of this project.

We would like to express our special gratitude and thanks to industry persons for giving me such attention and time. Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

Abstract

COVID-19[1] pandemic has rapidly affected our day-to-day life disrupting the world trade and movements. Wearing a protective face mask has become a new normal. In the near future, many public service providers will ask the customers to 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 achieve this 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. As a surveillance task performer, it can also detect a face along with a mask in motion. We explore optimized values of parameters using the Sequential Convolutional Neural Network model to detect the presence of masks correctly without causing over-fitting.

Keywords: Deep Learning, Computer Vision, OpenCV, TensorFlow, Keras.

TABLE OF CONTENTS

S.No	Particulars	Page No
1	Introduction	1-3
2	Literature Survey	4 - 9
3	Incorporated Packages	10-15
3	Functionality/Working	16-20
4	Result and Discussion	21
5	Conclusion and Future scope	22
7	References	23

Introduction

Coronavirus disease 2019 (COVID-19) has infected over 20 million individuals worldwide, resulting in over 0.7 million fatalities, according to the World Health Organization's official Situation Report – 205. COVID19 patients have reported a wide range of symptoms, ranging from mild signs to major sickness. One of them is respiratory issues such as shortness of breath or trouble breathing. COVID-19 infection can cause major consequences in elderly patients with lung disease, since they appear to be at a higher risk. 229E, HKU1, OC43, and NL63 are some of the most prevalent human corona-viruses that infect people all over the world. Viruses like 2019-nCoV, SARS-CoV[2], and MERS-CoV infect animals and develop into human corona-virus before infecting humans. Infectious beads can be spread by people with respiratory difficulties to everyone who comes into touch with them. Contact transmission can occur in the environment of a tainted person, since virus-carrying droplets may land on his neighbouring surfaces.

Wearing a clinical mask is essential for preventing some respiratory viral infections, such as COVID-19. The public should know whether to wear the mask for source control or COVID-19 aversion. The use of masks has the potential to reduce vulnerability to danger from a noxious individual during the "pre-symptomatic" stage, as well as stigmatize those who use masks to prevent the transmission of virus.

Medical masks and respirators for health care assistants are prioritized by the WHO. Face mask detection has therefore become a critical job in today's global society.

Since the end of 2019, infectious coronavirus disease (COVID-19) has been reported for the first time in Wuhan, and it has become a public damage fitness issue in China and even worldwide. This pandemic has devastating effects on societies and economies around the world causing a global health crisis. It is an emerging respiratory infectious disease caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). All over the world, especially in the third wave, COVID-19 has been a significant healthcare challenge. Many shutdowns in different industries have been caused by this pandemic. In addition, many sectors such as maintenance projects and infrastructure construction have not been suspended owing to their significant effect on people's routine life .

By now, the virus has rapidly spread to the majority of the countries worldwide. The last statistics (04/05/2021) provided by the World Health Organization (WHO) show 152,543452 confirmed cases and 3198528 deaths. According to the centers for Disease Control and Prevention (CDC), coronavirus infection is transmitted predominantly by respiratory droplets produced when people breathe, talk, cough, or sneeze with common droplet size 5–10'm but aerosol emission increases when humans speak and shout loudly.

Therefore, to prevent rapid COVID-19 infection, many solutions, such as confinement and lockdowns, are suggested by the majority of the world's governments. However, this COVID-19 management inefficacy can be additionally explored with game-theoretic scenarios beyond the public goods game. In particular, some researchers have focused on the hesitancy of governments in enacting difficult but necessary virus containment measures (e.g., stay-at-home orders and lockdowns), as well as noncooperation for reasons other than free riding. For instance, it has been argued that because strict stay-at-home measures can greatly impact people's livelihoods, the cost of staying home (coupled with lockdown fatigue) can end up outweighing the risk of infection from going out. As individual-level decisions have a direct impact on the society-level effectiveness of stay-at-home orders, governments may refrain from implementing them because of anticipated low rates of compliance, especially from socioeconomically disadvantaged individuals who do not have the luxury of staying home. Some governments may have also been hopeful that herd immunity from recoveries and vaccinations would allow them to avoid imposing such unpopular measures altogether .

With rising numbers of cases and stretched health facilities, as well as the lack of a vaccine throughout 2020 and difficulties associated with achieving herd immunity for COVID-19 , government inaction became increasingly unviable.

It is true that COVID-19 is a global pandemic and affects several domains. Nevertheless, it created a path for researchers in computer science. We have seen multiple research topics, such as creating new automatic detection methods of COVID-19 and detecting people with or without masks. Considering that there are some errors in the results of the early laboratory tests and their delays, researchers focused on different options .Therefore, the application of advanced artificial intelligence (AI) techniques coupled with chest radiological imaging (computed tomography (CT) and X-ray) can lead to a more accurate detection of the COVID-19 and can help to control the problem of loss of specialized physicians in isolated villages.

Literature Survey

Face identification models were previously constructed utilizing edge, line, and centre near features, with patterns detected from those features. These methods are used to locate binary patterns on a local level. These methods are particularly successful for dealing with gray-scale photographs, and they involve relatively little computational work.

AdaBoost is a regression-based classifier that will fit a regression function to the original data set, even if some incorrectly classified items are changed during back propagation to improve the results. The Viola Jones Detector presented a real-time object model that could be used to identify various object classifications.Harr-like features are similar to convolutions in that they are used to determine if a certain feature is present in the picture. When picture brightness fluctuates, this model fails to operate, and it also performs poorly when images are rotated.

Convolutional networks are mostly used to solve classification issues. There are several different CNN architectures, such as the VGG-16. This design is made up of two convolution layers, each having a kernel of size 224.

Malathi, J's [3] major focus was on detecting forgeries photographs used in various locations, such as social media and other places where publicity was desired. Various strategies are provided in this study to detect aspects of a counterfeit picture, such as image spicing and copy move attack, which may be dealt with by employing correlation analysis to find duplicate features. Patelet et al. established a

model for determining the grade of iron ore by extracting characteristics from mining sample material. It is critical to determine the ore's quality.
SVR is a support vector regressor that is used to measure ore quality in real time. They retrieved 280 characteristics for object recognition in this method, and the SFFS model was constructed using SVR.

Satapathy, Sandeep Kumar, and colleagues[4] created a model to detect number plates, which is a critical issue that aids police in pursuing numerous criminal cases. The authors employed an OCR-based technique to identify characters in the number plate, which were then saved and processed in a client-server paradigm to obtain the owner's information. Pathaket developed a multi-dimension biometric authentication system that works well in low-light situations. The accuracy of the system was increased by employing an entropy-based CNN. Medical plant detection is becoming a major issue that will aid regular people in detecting spices. The authors of this research presented a CNN-based approach. It was educated using medical leaf photos and can now more reliably identify medical plants. Human posture detection is an essential scientific topic that is receiving a lot of interest these days.

CNN employed a joint angular displacement technique to improve its ability to record 3D motion sign language in real time, which may be applied to a variety of real-time applications these days. Patel, Ashok Kumar, and colleagues devised a methodology for determining the grade of iron ore by extracting characteristics from mining sample material. It is critical to determine the ore's quality. SVR is a support vector regressor that is used to measure ore quality in real time. They retrieved 280 characteristics for object recognition in this method, and the SFFS model was constructed using SVR.

Deep learning is an important breakthrough in the AI field. It has recently shown enormous potential for extracting tiny features in image analysis. Due to the COVID-19 epidemic, some deep learning approaches have been proposed to detect patients infected with coronavirus. In this context, and unlike bacterial pneumonia, many other types of lung infections caused by viruses are called viral pneumonia. These viruses, such as the COVID-19, infect the lungs by blocking the oxygen flow, which can be life-threatening. This motivated researchers to develop many frameworks and schemes based on AI tools in the fight against this dangerous virus. Hence, we divide this section into two sections to provide an in-depth overview of the proposed techniques.

Deep Learning Tools and CXR Image-Based COVID-19 Detection

Radiography is a technique used to quantify the functional and structural consequences of chest diseases, to provide high-resolution images on disease progression. The aim of this scheme is to provide clinical decision support for healthcare workers and also for researchers. Hence, performance results, as well as the accuracy value of about 91.34%, and the other metrics in terms of recall, precision, and F1-score, prove the efficiency of the method. In the same context, Ozturk et al., introduced a new automatic COVID-19 detection model using CXR images denoted by the “DarkCovidNet.” It is used to provide correct diagnosis for both a binary classification (COVID-19 VS no findings) and a multiclass classification (COVID-19 VS pneumonia VS no findings). For binary classes, the classification accuracy produced by this model is about 98.08%, but, for multiclass cases, the accuracy is 87.02%. To validate their initial screening, radiologists can use the model to assist them. This model can be employed also via cloud to screen patients immediately. As a solution to the shortage of radiologists, this method can be used in remote places especially in countries affected by COVID-19. The most important advantage of this method is that such models can be used to diagnose supplementary chest-related diseases such as tuberculosis and pneumonia. However, the proposed work fits well into the COVID-19 detection phase, but to ensure its efficiency and model reliability, the authors may augment the dataset and retrain the proposed model. Their model, called COVID-Net, is open source and accessible to the general public. The test accuracy achieved by this model is 93.3%. Therefore, this model makes predictions which can assist clinicians in improving screening, transparency, and trust.

Deep Learning Tools and CT Image-Based COVID-19 Detection

Computed tomography scan or CT scan is a medical imaging technique utilized in radiology in order to get detailed images of the body for diagnosis purposes. Accurate and fast COVID-19 screening is achievable using CT scan images. Various works have been carried out in this context. In the dataset, we find 349 images corresponding to patients with COVID-19 and 463 images corresponding to patients without COVID-19. These images were divided into three sets: 80% of them for training set, 10% for validation, and 10% for testing. From the different techniques presented in this work, we cite CTnet-10, which is a self-developed model having an accuracy of 82.1%. We can also cite VGG-16, ResNet-50, InceptionV3, VGG-19, and DenseNet-169, having an accuracy of 89%, 60%, 53.4%, 94.52%, and 93.15%, respectively. The accuracy of VGG-19 is the best as compared to other models. To predict the results, CTnet-10 takes only 12.33 ms. This method is well-organized. It is useful for doctors, especially in mass screening. All the automatic diagnosis methods presented previously can be used by doctors for COVID-19 screening.

Methods Using CXR and CT Images

Combining two types of images in one dataset is an effective method to detect a disease. In this context, in [21], Sedik et al. presented two deep learning models: CNN and ConvLSTM. To simulate the models, two datasets are assumed. The first dataset includes CT images while the second set includes X-ray images. Each dataset contains COVID-19 and non-COVID-19 image categories.

The first model based on CNN includes five convolutional layers (CNVLs) accompanied by five pooling layers (PLs). Two layers (fully connected layer (FC) and classification layer) make up the classification network. The second model is a hybrid one. It combines ConvLSTM and CNN at the same time.

The classification network, too, is in the first model. To reduce the complexity of the planed deep learning structure, training, validation, and testing are the three phases that make up the two modalities. An optimization methodology is necessary in the training. To minimize the errors between the real and the estimated targets, Sedik et al. used the Adam optimizer. This type of model needs to be held carefully. The proposed models are evaluated by measuring accuracy, Matthews correlation coefficient (MCC), and F1-score. The specificity, negative predictive value (NPV), sensitivity, and positive predictive value (PPV) are considered also in the evaluation process.

The models were tested four times: firstly, on the dataset containing CT images with 288 COVID-19 and 288 normal images, this dataset is augmented by diverse rotations and operations of scaling, and the number of COVID-19 and normal images becomes 2880 and 2880, respectively; secondly, on the dataset containing X-ray images, this dataset includes two distinct augmented subsets, and each subset, named “augmented dataset A” and “augmented dataset B,” contains 304

COVID-19 and 304 normal images; thirdly, on the dataset, named COVID-19 radiography dataset, containing COVID-19 and viral pneumonia X-ray images; and fourthly, on a combined dataset, which combines both X-ray and CT images in the two cases, normal and COVID-19.

When the models were examined on CT images, the dataset is split into a training set (70%) and a testing set (30%). They were trained on 40 epochs. The testing accuracy for the CNN model and for the ConvLSTM was the same, equal to 99%. This is due to their methodical design and the nature of images. And when they are tested on the augmented dataset A, the testing accuracy was 99% for the first model and 100% for the second. However, when they were tested on the augmented dataset B, the testing accuracy was 100% for the first model and 99% for the second model. As for testing on the combined dataset, containing both X-ray and CT images, the testing accuracy was 99% for the first model and 98% for the second model. Finally, when they were tested on the radiography dataset, the testing accuracy of the first model was 95%, but 88% for the second model.

We can consider this scenario as a challenging one, because it is called for differentiating between two diseases (COVID-19 and pneumonia) with a high closeness in features. The proposed models achieved the same accuracy of 99% when they were tested on X-ray and CT images, while, in previous works, they achieved a range of 95% to 98% and a range of 83% to 90.1%, for X-ray and CT images, respectively. Therefore, the two proposed models can be considered as efficient COVID-19 detection systems.

Incorporated Packages

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 [. 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.

TensorFlow allows developers to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor.

TensorFlow provides all of this for the programmer by way of the Python language. Python is easy to learn and work with, and provides convenient ways to express how high-level abstractions can be coupled together. Nodes and tensors in

TensorFlow are Python objects, and TensorFlow applications are themselves Python applications.

The actual math operations, however, are not performed in Python. The libraries of transformations that are available through TensorFlow are written as high-performance C++ binaries. Python just directs traffic between the pieces, and provides high-level programming abstractions to hook them together.

TensorFlow applications can be run on most any target that's convenient: a local machine, a cluster in the cloud, iOS and Android devices, CPUs or GPUs. If you use Google's own cloud, you can run TensorFlow on Google's custom TensorFlow Processing Unit (TPU) silicon for further acceleration. The resulting models created by TensorFlow, though, can be deployed on most any device where they will be used to serve predictions.

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 [19]. 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.

Keras runs on top of open source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK). Theano is a python library used for fast numerical computation tasks. TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks.

Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

Keras user experience

1. **Keras is an API designed for humans**
   Best practices are followed by Keras to decrease cognitive load, ensures that the models are consistent, and the corresponding APIs are simple.
2. **Not designed for machines**
   Keras provides clear feedback upon the occurrence of any error that minimizes the number of user actions for the majority of the common use cases.
3. **Easy to learn and use.**
4. **Highly Flexible**
   Keras provide high flexibility to all of its developers by integrating low-level deep learning languages such as TensorFlow , which ensures that anything written in the base language can be implemented in Keras.

OpenCV

OpenCV (Open Source Computer Vision Library), 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 . The proposed method makes use of these features of OpenCV in re sizing and color conversion of data images.

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both **academic** and **commercial** use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

Applications of OpenCV: There are lots of applications which are solved using OpenCV, some of them are listed below

- face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- object recognition
- Medical image analysis
- Movies – 3D structure from motion
- TV Channels advertisement recognition

OpenCV Functionality

- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (objdetect, features2d, nonfree)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu)

Image-Processing

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it.
If we talk about the basic definition of image processing then **"Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality".**

Digital-Image :
An image may be defined as a two-dimensional function f(x, y), where x and y are spatial(plane) coordinates, and the amplitude of fat any pair of coordinates (x, y) is called the intensity or grey level of the image at that point.
In another word An image is nothing more than a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function f(x, y) at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be.
Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image.
Image processing basically includes the following three steps:

1. Importing the image
2. Analysing and manipulating the image.

The Proposed Face-Detection System Model

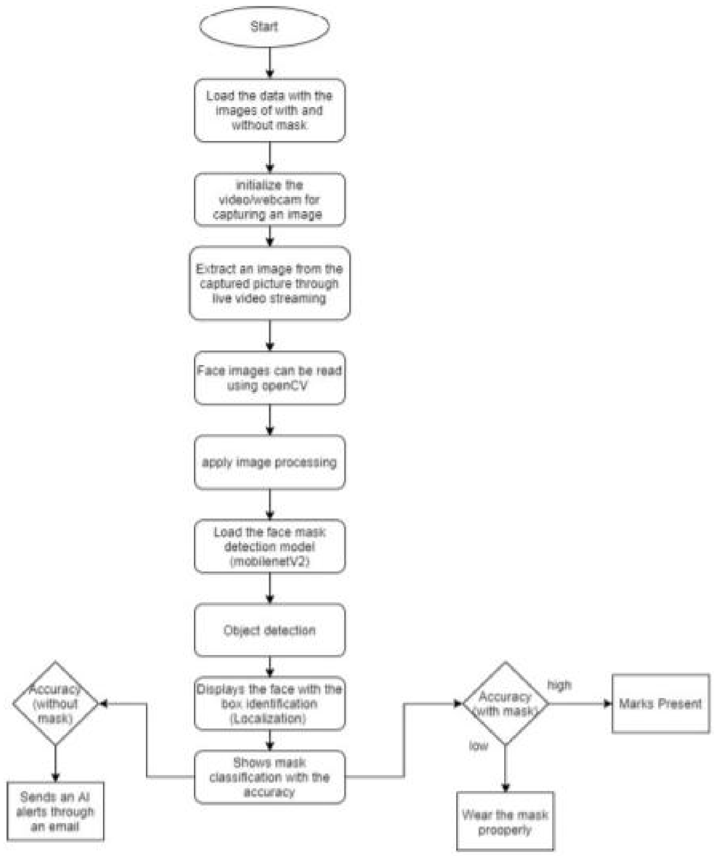
The model proposed here is designed and modeled using python libraries namely Tensorflow, Keras and OpenCV. The model we used is the MobileNetV2 of Convolutional neural network. The method of using MobileNetV2 is called using Transfer Learning. Transfer learning is using some pre trained model to train your present model and get the prediction which saves time and makes using training the different models easy.

We tune the model with the hyper parameters : learning rate, number of epochs and batch size. The model is trained with a dataset of images with two class, with mask and without mask.

(i) Training the model with the taken dataset.

(ii) Deploying the model In the paper we have developed a model using the above mentioned libraries.

We have tested the model for different conditions with different hyper parameters, for which the results are mentioned in the next section. First we feed the dataset in the model, run the training program, which trains the model on the given dataset. Then we run the detection program, which turns on the video stream, captures the frames continuously from the video stream with an anchor box using object detection process. This is passed through the MobileNetV2 model layers which classifies the image as with or without mask. If the person is wearing a mask, a green anchor box is displayed and red if not wearing a mask with the accuracy for the same tagged on the anchor box.



The face mask recognition system uses AI technology to detect the person with or without a mask. It can be connected with any surveillance system installed at your premise. The 3 authorities or admin can check the person through the system to confirm their identity. The system sends an alert message to the authorized person if someone has entered the premises without a face mask. The accuracy rate of detecting a person with a face mask is 95-97% depending on the digital capabilities. The data has been transferred and stored automatically in the system to enable reports whenever you want.

Functionality/Working

Face mask recognition is achieved in this study using a machine learning algorithm and the MobileNetv2 picture classification approach. MobileNetV2 is a technique based on Google's Convolutional Neural Network (CNN) that has been enhanced in terms of performance and efficiency.

Data Collecting

The collection of data is the first step in developing the Face Mask Recognition model. The dataset collects information on persons who use masks and those who don't.The model will distinguish between persons who are wearing masks and those who are not.

Pre-processing

This study utilizes 1915 data with a mask and 1918 data without a mask to develop the model. The image is cropped at this point until the only visible thing is the object's face. The data must then be labelled. The obtained data was divided into two groups: with and without a mask. The data is divided into two categories after it has been labelled.

The pre-processing step occurs before the data is trained and tested. Pre-processing consists of four steps: shrinking picture size, turning the image to an array, pre-processing input using MobileNetV2, and performing hot encoding on labels. Due to the efficacy of training models, picture scaling is a significant pre-processing step in computer vision. The model will perform better if the picture is smaller.

The next step is to create an array from all of the photographs in the dataset. The picture is transformed into an array so that the loop function may call it. The picture will then be used to pre-process input with MobileNetV2. Because many machine learning algorithms cannot directly act on data labeling, the final step in this phase is to execute hot encoding on labels.

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

a) Data Visualization: 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 a particular pattern in 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 the items in each passed iterator is paired together consequently. The mapped variable label dict looks like: {'with mask': 0, 'without mask': 1}

b) Conversion of RGB image to Gray image:

Modern descriptor-based image recognition systems regularly work on grayscale images, without elaborating the method used to convert from color-to-grayscale. This is because the colorto-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. Conversion of a RGB image to a Gray Scale image of 100 x 100 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 resized into 100 x 100.

c) 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 tantamount size corresponding to 3D feature tensor. However, neither images are customarily coextensive nor their corresponding feature tensors. 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.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 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.

Building the Model

The next step is to construct the model. Building of the training image generator for augmentation, the basic model with MobileNetV2, adding model parameters, compiling the model, training the model, and storing the model for future prediction are the six processes in generating the model.

CNN has become ascendant in miscellaneous computer vision tasks. The current method makes use of Sequential CNN. 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. Following layers can perform instinctive shape reckoning . 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 .

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).The second Convolution layer has 100 filters and Kernel size is set 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 connected neural 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 (Dense) with two outputs for two categories uses the Softmax activation function

Implementing the model

The model implemented in the video. The video is read frame by frame, and then the face detection algorithm is used. If a face is spotted, the programme moves on to the next step. Reprocessing will be done out on recognized frames containing faces, including shrinking the picture size, converting to the array, and Pre-processing input using MobileNetV2.

Predicting input data from the stored model is the next step. Predict the input picture that has been processed using a model that has already been created. In addition, the video frame will be tagged with whether the individual is wearing a mask or not, as well as the predicted percentage.

Result and Discussion

The model is trained, validated and tested upon a dataset with 1915 images with mask and 1918 images without mask. The approach achieves an accuracy of up to 99.77 percent, demonstrating how this optimal precision reduces the cost of mistake. MaxPooling[6] is one of the primary reasons for obtaining this level of precision. It adds rudimentary translation in variance to the internal representation while also reducing the number of parameters that the model must learn.

The system can recognize partly obstructed faces with a mask, hair, or hand with high accuracy. It evaluates the degree of occlusion of four places – nose, mouth, chin, and eye – to distinguish between annotated mask and hand-covered face. As a result, a mask that completely covers the face, including the nose and chin, will only be considered as "with mask" by the model.

CONCLUSION AND FUTURE SCOPE

In conclusion, this research proposes a face mask identification algorithm based on machine learning. Following the training, validation, and testing phases, the model can accurately predict the percentage of persons using face masks in specific cities.

This study can be an easy move for authorities to use more unstructured data resources for more data-based mitigation, evaluation, prevention, and action planning against COVID-19 in the name of the statistical organization that needs to move quickly to adopt and take advantage of machine learning and new digital data resources.

In this paper, we briefly discussed the work's motivation. The model's learning and performance task was then demonstrated. The method has attained a reasonable level of accuracy using basic machine learning tools and simplified methodologies. It may be used for a wide range of purposes.

Given the Covid-19 situation, wearing a mask may become mandatory in the near future. Many government agencies will require clients to wear masks appropriately in order to use their services. The implemented approach will make a significant contribution to the public health care system. It might be extended in the future to detect whether or not a person is wearing the mask properly. The model should be enhanced further to recognize whether the mask is virus-prone or not, i.e. whether it is surgical, N95, or not.

REFERENCES

[1]. Coronavirus disease (COVID-19)- “Symptoms of COVID-19”: https://www.cdc.gov/coronavirus/2019-ncov/symptomstesting/symptoms.html.

[2]. SARS-CoV-”Severe acute respiratory syndrome (SARS)”: https://www.cdc.gov/sars/index.html.

[3]. Krishnaveni, G., B. Lalitha Bhavani, and NVSK Vijaya Lakshmi. "An enhanced approach for object detection using wavelet based neural network." Journal of Physics: Conference Series. Vol. 1228. No. 1. IOP Publishing, 2019.

[4]. Pathak, Mrunal, Vinayak Bairagi, and N. Srinivasu. "Multimodal Eye Biometric System Based on Contour Based E-CNN and Multi Algorithmic Feature Extraction Using SVBF Matching."International Journal of Innovative Technology and Exploring Engineering.

[5]. MobileNetV2- “Inverted Residuals and Linear Bottlenecks”: https://arxiv.org/abs/1801.04381.

[6]. Max Pooling-”A Gentle Introduction to Pooling Layers for Convolutional Neural Networks”: https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/.