A Project/Dissertation ETE REVIEW Report

on

COVID PREVENTION SYSTEM

Submitted in partial fulfillment of the requirementfor the award of the degree of

Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING

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

Under TheSupervision

Mrs. N Gayathri

ASSISTANT PROFESSOR

Submitted By

MOHIT SHARMA - 19SCSE1010285

DAKSH - 19SCSE1010350

SCHOOL OF COMPUTING SCIENCE AND

ENGINEERING DEPARTMENT OF COMPUTER SCIENCE

AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA2021-22

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 **"COVID PREVENTION SYSTEM"** in partial fulfillment of the requirements for the award of the project review 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 **Mrs. N Gayathri, Assistant Professor**, Department of Computer Science and Engineering of School of Computing Science and Engineering , Galgotias University, Greater Noida

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

19SCSE1010285- MOHIT SHARMA

19SCSE1010350- DAKSH

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

Mrs. N Gayathri

Assistant Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of 19SCSE1010285 MOHIT SHARMA, 19SCSE1010350 DAKSH 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:
GreaterNoida

Fall 2021 – 2022

Contents

CHAPTER 1

Abstract

As we know Covid-19 pandemic has affected both health and life style of the people, not only this the global economy also affected badly .We are still trying many things to find origin of virus but with the precaution. And all people are advised to take vaccine but only vaccination will not gives a full guaranty to prevent people lives, so we have to defend ourselves with the spread of the viruses .For that mask and social distancing are main key factor that are also recommended by the government and the public health agencies. As people are not habitual of wearing the mask so many times people forget to wear mask at public places that is one of the main reason for spreading of the virus. Sometime we see that at crowded places like metro station and malls and universities , two or three guards are always there to check if people are wearing mask or not and other guard for checking thermal temperature and also telling people to maintain a social distance. So there are lot of problem in this because at crowed places like at metro station people are in queue for checking their temperature then it is somehow hard to maintain social distance or if there is a infected person found in queue then all other surrounded people can become infected .So we decided to contribute to the public health sector by making a complete system that will check if they are wearing a mask on their face properly or not with that it will also check thermal temperature of the people through the cameras and checks if someone is not violating social distancing rule without forming the queue for checking . This will prevent people from the infected people and also save time of people and now maximum one guard is needed for monitoring .To make this project practically we are taking help of machine learning and deep learning. We will be using face detection and recognition algorithms that will detect the faces of people and will give probability of someone is wearing mask or not or maintaining safe distance or not and also temperature .Programming language will be python. For face detection we are using YOLO v4(You look only once) which support Convolutional neural network .So our process flow will be like that first we import all libraries and after that we will build neural network and after that training will be done on a model and then testing the model. After that system will become ready to deploy on cloud.

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

CHAPTER 2

Literature Review

In one Shot Detector mechanism is used for the object detection purpose. This project contains face mask detector can be used in many areas like Public malls, Bus stand and other heavy traffic related places to monitor the People and to avoid the increment of the problem of COVID by checking who is to follow the rules of precaution and who is not doing.
The algorithm is used to capture the faces expression of the people and check whether they are wearing mast or not. It is equally effective for both individual and group for detection. Our face mask detection system can decrease the number of enforcement agents on the ground.

***Figure 1**. Result image*

During the evaluation the model will detect their mask with the percentage accuracy and we find some error then we again set hyper parameter and evaluate again.
So if it find there is no mask on that person's face then the application will send sms or do the broadcast for wearing the mask..

Why is object detection important?

Face mask detection is very useful in this pandemic situation as some people are not following the rules of wearing mask. And if they come in contact to an infected person they harms them as well there family and other people.
So if it find there is no mask on that person's face then the application will send sms or do the broadcast for wearing the mask..

Why is object detection important?

Face mask detection is very useful in this pandemic situation as some people are not following the rules of wearing mask. And if they come in contact to an infected person they harms them as well there family and other people.Manual Monitoring is very difficult for officers to see that whether the people are wearing mask or not. So in our technique, We are using webcam to detect people faces and to stop from virus transmission.

• It has fast and high accurate system.

• It will save a lot of time of people not need to stand up in queue for checking of temperature.

• We can keep peoples safe from viruses by following protocols.

•.It is a complete system that need maximum one worker to monitor

Problem Formulation

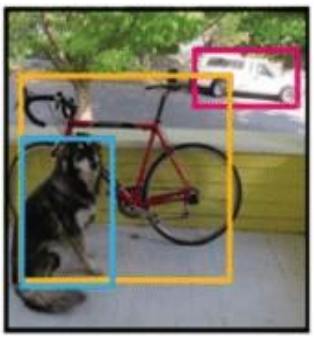
In this time, we all are facing this pandemic situation. We all know that how important is to take precautions in the present condition. Because if we will not take precautions so we have to suffer as well as our belongings because if we came in a contact to an infected person and we are not following the precautions so we will also be infected.

So, this problem can be solved using this Detection System. By using this system it will be easy to detect somebody has wear mask on his face or not. This system actually can used in many areas like bus stand , metro railway station etc where it will check who is following the rules and who are disobeying .

Figure 2*. Work flow image*

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 [1], 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 improve the 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. There are various methods of object detection based on deep learning which are divided into two categories: one stage and two stage object detectors

Figure 1: Bounding boxes in an image

Two stage detectors use two neural networks to detect objects, for instance region-based convolu- tional neural networks (R-CNN) and faster R-CNN. The first neural network is used to generate region proposals and the second one refines these region proposals; performing a coarse-to-fine de-tection. This strategy results in high detection performance compromising on speed. The seminal work R-CNN is proposed by R. Girshick et al. [2]. R-CNN uses selective search to propose some candidate regions which may contain objects. After that, the proposals are fed into a CNN model to extract features, and a support vector machine (SVM) is used to recognize classes of objects. However, the second stage of R-CNN is computationally expensive since the network has to detect proposals on a one-by-one manner and uses a separate SVM for final classification. Fast R-CNN

[3] solves this problem by introducing a region of interest (ROI) pooling layer to input all proposalregions at once. Faster RCNN [4] is the evolution of R-CNN and Fast R-CNN, and as the name implies its training and testing speed is greater than those of its predecessors. While R-CNN and Fast R-CNN use selective search algorithms limiting the detection speed, Faster R-CNN learns theproposed object regions itself using a region proposal network (RPN).

On the other hand, a one stage detector utilizes only a single neural network for region proposals and for detection; some primary ones being SSD (Single Shot Detection) [5] and YOLO (You OnlyLook Once) [6]. To achieve this, the bounding boxes should be predefined. YOLO divides the image into several cells and then matches the bounding boxes to objects for each cell. This, however, is not good for small sized objects. Thus, multi scale detection is introduced in SSD which can detectobjects of varying sizes in an image. Later, in order to improve detection accuracy, Lin et. al [7] proposes Retina Network (RetinaNet) by combining an SSD and FPN (feature pyramid network) toincrease detection accuracy and reduce class imbalance. One-stage detectors have higher speed but trades off the detection performance but then only are preferred over two-stage detectors.

Like object detection, face detection adopts the same architectures as one-stage and two-stage de- tectors, 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 facemask detectors have been modeled using OpenCV, Pytorch Lightning, MobileNet, RetinaNet and Support Vector Machines. Here, we will be discussing two projects. One project used Real World Masked Face Dataset (RMFD) which contains 5,000 masked faces of 525 people and 90,000 normalfaces [8]. These images are 250 x 250 in dimensions and cover all races and ethnicities and are unbalanced. This project took 100 x 100 images as input, and therefore, transformed each sample image when querying it, by resizing it to 100x100. Moreover, this project uses PyTorch then they convert images to Tensors, which is the base data type that PyTorch can work with. RMFD is im- balanced (5,000 masked faces vs 90,000 non-masked faces). Therefore, the ratio of the samplesin train/validation while splitting the dataset was kept equal using the train test split function of sklearn. Moreover, to deal with unbalanced data, they passed this information to the loss function toavoid unproportioned step sizes of the optimizer. They did this by assigning a weight to each class,according to its representability in the dataset. They assigned more weight to classes with a small number of samples so that the network will be penalized more if it makes mistakes predicting the label of these classes. While classes with large numbers of samples, they assigned to them a smallerweight. This makes their network training agnostic to the proportion of classes. The weights for each class were chosen using the formula below:

$$Class\ Weight = 1 - \frac{Class\ Cardinality}{Cardinalities\ of\ all\ classes}$$

To load the data efficiently this project used the data loader. For instance, in this project, they used the PyTorch lighting, and to load them for training and validation they divided data into 32 batches and assigned the works of loading to the 4 number of workers, and this procedure allowed them to perform multi-process data loading. Like most of the projects, this project also used Adam optimizer. If any Model has a high rate of learning, it learns faster, but it bounces a lot to reach theglobal minima and may diverge from the global minima. However, a small learning rate may take considerably lower time to train, but it reaches to the global minima. If the loss of the model declines quickly for any learning rate, then that learning rate would be the best learning rate. However, it seems that this project considered the 0.00001 learning rate would be the best for their model so thatit could work efficiently. To train the model they defined a model checkpointing callback where they wanted to save the best accuracy and the lowest loss. They tried to train the model for 10 epochs and after finding optimal epoch, they saved the model for 8 epochs to test on the real data. To get rid of the problem of occlusions of the face which causes trouble face detectors to detect masksin the images, they used a built-in OpenCV deep learning face detection model. For instance, the Haar-Cascade model could be used but the problem of the Haar-Cascade model is that the detection frame is a rectangle, not a square. That is why, without capturing the portion of the background, theface frame can fit the entirety of the face, which can interfere with the face mask model predictions.

In the second project [9], a dataset was created by Prajna Bhandary using a PyImageSearch reader.This dataset consists of 1,376 images belonging to all races and is balanced. There are 690 images with masks and 686 without masks. Firstly, it took normal images of faces and then created a customized computer vision Python script to add face masks to them. Thereby, it created a real- world applicable artificial dataset. This method used the facial landmarks which allow them to detect the different parts of the faces such as eyes, eyebrows, nose, mouth, jawline etc. To use the facial landmarks, it takes a picture of a person who is not wearing a mask, and, then, it detects the portion of that person's face. After knowing the location of the face in the image, it extracted the face Region of Interest (ROI). After localizing facial landmarks, a picture of a mask is placed into

the face. 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 theface detector did not have images of people wearing face masks as a result, if the large portion of faces is occluded, the face detector would probably fail to detect properly. To get rid of this problem, they should gather actual images of people wearing masks rather than artificially generated image.

The major goal of this research is to stop the spread of the corona virus; currently, a vaccine will work in lieu of her, but this will not be enough to prevent us. Masks and hand sanitizers are two items that keep us from getting into contact with viruses again. However, some people forget to wear masks or do not use masks in public places for various reasons, which affects not only them but also others. As a result, our project will recognise those who are wearing masks and make a public announcement to remind them to do so. So, in this case, we're utilising machine learning to detect and analyse faces.

While already many people are persuaded of the interest for facial protective mask, as suggested by the World Health Organization (WHO, 2020) and Studies in Science conducted by (N. Leung et al, 2020), (S. Zhou et al, 2018) and (M. Sande et al, 2008), One may note that many people don't wear masks for protection from the virus (see various sample data in Figure 1). As a result of these findings, public health educators and others began to launch mask-wearing prevention initiatives. Sensitizing individuals on the need of wearing a mask through the distribution of preventative posters and sketches (African Union, 2020) – (L. Colart et al, 2020). In our situation, we propose to assist these projects and public health sectors by establishing a real-time video-based research system and a linked interactive resource dedicated to assessing face mask wear using a webcam. As we know Corona Virus pandemic has affected both health and global economy very badly .We tried many things to stop the spread of virus but nothing works. At the end vaccination was our last hope but only vaccination does not helping us to come out from this pandemic, we should defend ourselves with the spread of the viruses .For that mask and social distancing are key factors that are recommended by government and public health agencies, sometimes people don’t wear mask at public places so we decide to make a project that will recognize the faces of the people and check if they are wearing a mask on their face or not so by doing this we are we are contributing to public health sector.To make this project workfull we are taking help of machine learning. We will be using face detection and recognition algorithms that will detect the faces of people and there will be some libraries that give probability if some is wearing mask or not .In this project programming language will be python. And mention before we will be taking help of opencv and keras , both are machine learning libraries. For face detection Convolutional neural network/haar-cascades are quite famous so we can use one of these both .So our process flow will be like, first we import all libraries and after that we will build neural network and after that training will be done on a model and then testing the model. After that our project will become ready to deploy the project on

cloud then set the model to the hardware means camera and it starts recognizing the face mask through camera**.** For object detection, a single Shot Detector mechanism is used. This project includes a face mask detector that may be utilised in a variety of settings such as public malls, bus stops, and other high-traffic areas to monitor people and prevent the spread of COVID by determining who is following the precautionary regulations and who is not.
The system is used to capture people's facial expressions and determine whether or not they are wearing a mast. It is equally good in detecting both individuals and groups. The number of enforcement agents on the ground could be reduced thanks to our face mask detecting system.
During the evaluation, the model will detect their mask with a percentage accuracy. If there is a mistake, we will adjust the hyper parameter and re-evaluate.
If the application detects that there is no mask on that person's face, it will send an SMS or broadcast requesting that the mask be worn.
Face mask detection is highly essential in this pandemic circumstance because some people do not wear masks according to the rules. And if they get into contact with an infected person, it is harmful to them, their family, and others. Officers have a tough time determining whether or not persons are wearing masks when using manual monitoring. As a result, with our method, we use a webcam to detect people's faces and prevent virus transmission.

- It has fast and high accurate system.
- This system can be used in ATMs, Banks etc.
- We can keep peoples safe from viruses.

- It gives buzzer sound to wear mask.

Artifacts are typically identified by their distinguishing qualities. There are numerous characteristics that identify a human face from a variety of other artefacts. It locates faces and then uses structural traits such as eyes, nose, and mouth to recognise a face by extracting structural characteristics such as eyes, nose, and mouth. In most cases, a statistical classifier was used to differentiate between face and non-facial regions. Human faces, on the other hand, have distinct textures that can be utilised to differentiate them from other objects. Furthermore, the feature edge can aid in the separation of artefacts from the face. In the next part, we'll show how to use OpenCV to implement a feature-based strategy.

Object detection problems aim to locate and classify objects in an image [12]. The face mask and physical distance detection are classified as object detection problems. Object detection algorithms have been developing in the past 2 decades. Since 2014, deep-learning usage in object detection has driven remarkable breakthroughs, improving accuracy and detection speed [13].

Object detection is divided into two categories. One-stage detection, such as You Only Look Once (YOLO), and two-stage detection, such as Region-Based Convolutional Neural Networks (R-CNN). Two-stage detectors have higher localization and object recognition accuracy, while the one-stage detectors achieve greater inference speed [14]. R-CNN or regions with CNN features (R-CNNs), introduced by Girshick et al. [15], implements four steps; First, it selects several regions from an image as object candidate boxes, then rescales them to a fixed size

image. Second, it uses CNN for feature extraction of each region. Finally, the features of each region are used to predict the category of boundary boxes using the SVM classifier [15, 16]. However, feature extraction of each region is resource-intensive and time-consuming since candidate boxes have overlap, making the model perform repetitive computation. Fast R-CNN handles the issue by taking the entire image as the input to CNN to extract features [17]. Faster R-CNN speeds up the R-CNN network by replacing selective search with a Region Proposal Network (RPF) to reduce the number of candidate boxes [18,19,20]. Faster R-CNN is a near-real-time detector [18].

Face mask detection identifies whether a person is wearing a mask or not in a picture [21, 22]. Physical distance detection first recognizes people in a picture then identifies the real distance between them [23]. Since the beginning of COVID-19 pandemic, studies have been conducted to detect face masks and physical distancing in the crowd. Jiang et al. [21] employed a one-stage detector, called RetinaFaceMask, that used a feature pyramid network to fuse the high-level semantic information. They added an attention layer to detect the face mask faster in an image. They achieved a higher accuracy of detection comparing with previously developed models. Militante & Dionisio [24] used the VGG-16 CNN model and achieved 96% of accuracy to detect people who wear a face mask or not. Rezaei & Azarmi [25] developed a model based on YOLOv4 to detect face mask wearing and physical distancing. They trained their model on some accessible big databases and achieved the precision of 99.8% for a real-time detection. Ahmed et al. [23] employed YOLOv3 to detect people in the crowd then used the Euclidian distance to detect the physical distance between two people. They also used a tracking algorithm to track people who violate the physical distancing in a video. They achieved an initial accuracy of 92% and then increased the accuracy by applying transfer learning to 98%.

One major problem in training of deep-learning models for construction sites is data deficiency. Using drones in construction sites is one of the methods to address this issue. In construction projects, drones capture real-time video and provide aerial insights that help catch problems immediately. Research studies have collected large-scale image data from drones and applied deep-learning techniques to detect objects. Asadi & Han [26] developed a system to improve data acquisition from drones in construction. Mirsalar Kamari and Ham [10] used drones' image data in a construction site to quickly detect areas vulnerable to the wind in the job site. Li et al. [27] applied deep-learning object detection networks on video captured from drones to track-moving objects. The data obtained from drones can be used to detect workers and identify whether they are wearing face masks and practice physical distancing. In this paper, to address the data-deficiency problem, we used different data augmentation techniques. A critical question during the novel coronavirus disease (COVID-19) pandemic is the effectiveness of the social distancing policies adopted by US states and localities in bending the curve. Although these policies take a variety of forms—such as imposing shelter-in-place orders; restricting dine-in at restaurants;

closing nonessential business such as bars, entertainment venues, and gyms; banning large social gatherings; and closing public schools—their effectiveness depends critically on the cooperation of the public. For example, although California's first-in-the-nation shelter-in-place order carries threats of fines and incarceration, its effectiveness fundamentally relies on social pressure.1 Compliance with social distancing orders appears to be related to local income, partisanship, and political beliefs in the US, and compliance with self-quarantines is related to potential losses in income in Israel.2–4

Some epidemiological models forecast the eventual number of COVID-19 cases and fatalities based on untested assumptions about the impact of social distancing policies in contemporary society. The widely cited Imperial College London model assumes that contact outside the home, school, or workplace declines by 75 percent; school contact rates are unchanged; workplace contact rates fall by 25 percent; and household contact rates rise by 25 percent.5 Another study assumes that social distancing measures will reduce the average contact rate by 38 percent, based on evidence from the 1918 influenza pandemic.6

In the US, the literature on models of social distancing during the COVID-19 pandemic is evolving rapidly; at the time of writing, we were aware of several working papers that examined the consequences of social distancing policies. Recent work found statistically significant effects of stronger measures (such as shelter-in-place orders) on movement, using difference-in-differences methods and state-level data from Google.7 Similar findings have been obtained in a study with SafeGraph mobility data,8 although a different study using PlaceIQ and SafeGraph data found that strong measures were not important.9 Another paper used synthetic control methods to show that California's shelter-in-place order markedly reduced COVID-19 cases.1 A study of shelter-in-place orders across the US also found a reduction in cases, as well as higher rates of staying home full time.10 Other authors used interrupted time-series methods and found that early statewide social distancing measures were associated with decreases in states' COVID-19 growth rates, but later shelter-in-place orders did not lead to further reductions.11

At issue is not whether isolation works to limit the spread of disease but, rather, whether the particular government restrictions designed to encourage social distancing in the US reduced spread relative to simply providing information and recommendations. Individuals may voluntarily engage in avoidance behavior, such as washing hands or wearing masks, once they fully perceive the risks of contagion.12,13 Critics of more stringent government measures highlight Sweden's less intrusive response to COVID-19, although Sweden's strategy is increasingly being questioned.14 Rigorous empirical research is needed to determine the impacts of the various aspects of governments' responses in the US.

Our work, which leveraged both state and county policy variation and used a flexible event study method that allowed for effects to vary across measures and over time, estimated the impacts of four types of social distancing measures on confirmed COVID-19 case growth rates through April 27, 2020. The reduced-form approach captures any potential pathways driven by these mandates, including complementary avoidance behaviors the public may engage in if these orders provide an informational shock in addition to increasing social distancing.

The unit of observation was daily US counties or county equivalents. Although there are 3,142 US counties, official COVID-19 records report New York City as a whole instead of dividing it into five counties, reducing this number to 3,138. Our data set tracked counties over the course of fifty-eight days from March 1, 2020, to April 27, 2020, leading to a sample size of 182,004. We chose March 1 as the start date because no new cases were reported in the entire US on most days in January and February. We chose the April 27 end date to coincide with the first removal of one of the four types of restrictions we analyzed (the reopening of restaurants and other entertainment facilities in Georgia).15 Each county observation was weighted by population, using 2018 estimates from the Department of Agriculture's Economic Research Service.16

We examined the daily growth rate in confirmed COVID-19 cases at the county level, which originated from the COVID-19 Dashboard provided by the Johns Hopkins Center for Systems Science and Engineering. This repository contains data on COVID-19 cases worldwide, collected from a range of sources including government and independent health institutions.17

The daily exponential growth rate was calculated as the natural log of cumulative daily COVID-19 cases minus the log of cumulative daily COVID-19 cases on the prior day. We chose this functional form because epidemiological models predict exponential growth in the absence of intervention. Percentage growth in cases is identical to percentage growth in cases per capita because reported county populations did not vary during the sample period. The growth rate was multiplied by 100 and can be read as percentage-point changes. In computing the growth rate, we followed a recent COVID-19 study and added 1 to the case counts to avoid dropping counties that started with zero cases.18 The data on the timing of state and local government social distancing interventions were gathered from a host of sources and made available by Johns Hopkins University.19 Part A of the online appendix explains a few corrections we made to the dates and provides a list of state- and county-level policies used in the analysis.20

We focused on four government-imposed interventions: shelter-in-place orders, public school closures, bans on large social gatherings, and closures of entertainment-related businesses. For large gatherings we used the date of the first prohibition that was at least as restrictive as five hundred people. Most of the bans were much more restrictive: 95 percent of the time (in our population-weighted sample) the prohibition extended to fifty people. For entertainment-related businesses, we used the date of the

first closure of either restaurant dining areas (including bars) or gyms and entertainment centers. Ninety-six percent of the time, if one such prohibition was in place, the other was in place as well.

We included control variables related to the availability of COVID-19 tests. The same data repository that provides case counts also includes daily counts of positive, negative, and pending tests in each state on each day, which we added together.17 To mirror our measure of cases, we converted this testing variable to the exponential daily growth rate of cumulative tests performed. Because COVID-19 test results are generally not available immediately, we also included the one-day lag of this growth rate. Further lags (out to ten days) were considered but were always statistically insignificant, so we did not include them. Most states did not report any pending tests, meaning that they did not officially record tests until the results were obtained. This likely explains the lack of a longer lag between testing growth and case growth.

We estimated the relationship between social distancing policies and the exponential growth rate of confirmed COVID-19 cases, using an event study regression with multiple policies. Statistical analysis was conducted using Stata MP, version 15. This approach is akin to difference-in-differences but is more flexible, as it interacts the policy variables with multiple indicators of time since implementation, thereby tracing out the evolution of the policy effects over time.21

For each of the four policies, we included seven variables: whether the policy was implemented one to five, six to ten, eleven to fifteen, sixteen to twenty, or more than twenty days before a given sample day and whether it will be implemented five to nine or ten or more days after that day. Implementation on the current day through four days later was, therefore, the reference group. If a county never adopted the policy, each of these variables was set to 0 throughout the sample period.

An event study model is particularly useful for studying the impact of social distancing policies on COVID-19 cases for two reasons. First, after accounting for the incubation period and time between the onset of first symptoms and a positive test result, such policies likely affect official cases after a considerable lag only.22 In addition, the inclusion of variables reflecting future implementation allows for an analysis of prepolicy trends. As it is not plausible for policies that have not yet been implemented to causally affect current cases, finding such associations could suggest misspecification. For instance, one might expect counties with rapidly growing case counts to be the most likely to enact these measures, leading to a reverse-causal relationship between current cases and future policies that would be detected by our model.

Each policy was implemented at least ten days after the start of the sample period and at least 20 twenty days before the end. Therefore, each policy contributes to the

identifying variation for all coefficients except those for implementation more than twenty days before the specified sample day and ten or more days after the specified sample day. Because the estimated policy effects at those two catch-all periods could partially reflect compositional changes, they should therefore be interpreted with more caution than the estimates for the other time intervals.

In addition to the testing controls discussed here, the model also included fixed effects for geography and time. County fixed effects accounted for the likelihood that even aside from differences in policies, case growth rates may have varied because of a number of county characteristics. These characteristics include population density and residents' education, political orientation, and age.3,4 Fixed effects for each day in each of the nine US census divisions (522 fixed effects in total) allowed for flexible underlying trends in growth rates that could vary in different parts of the country, helping account for the staggered nature of the outbreak across locations.23 We report 95 percent confidence intervals, with standard errors robust to heteroskedasticity and clustered by state, which is the level of most of the policy variation. Part B of the appendix provides the formal notation for the event study model.20.

There were several limitations to our analysis. Official COVID-19 case counts are known to understate the true prevalence of the disease, as they do not include asymptomatic carriers, those who are not ill enough to seek medical care, and those who are unable to obtain a test because of supply constraints.1 Nonetheless, confirmed case counts are crucial to the Trump administration's "Opening Up America Again" plan, which proposes either a "downward trajectory of documented cases within a 14-day period" or a "downward trajectory of positive tests as a percent of total tests within a 14-day period (flat or increasing volume of tests)" as criteria for loosening social distancing measures.24 Moreover, to the extent that testing shortages led to only the sickest individuals receiving tests, official case counts can loosely be interpreted as the prevalence of moderate-to-severe illness, a relevant metric for policy purposes.

A related caveat is that, ideally, we would like to be able to control more precisely for access to testing. Available data allowed us to control for the number of tests performed at only the state, rather than county, level. However, most of our policy variation is at the state level, so controlling for state-level testing should go a long way toward alleviating bias. In addition, the number of tests performed is not an ideal measure of the ease of obtaining a test because it also reflects the level of illness in the community.

Also, we might ideally want to estimate a richer econometric model. It would be interesting to trace out the timing of impacts more exactly and study the policies' interactions with each other or with county characteristics. Future work should also examine the impacts of other social distancing policies such as closing public parks and beaches, the requirement to wear masks in public, restrictions on visitors in

nursing homes, state announcements of first cases or fatalities, and federal government actions such as prohibiting international travel.9 However, it is difficult to include numerous correlated policy variables without reducing precision to the point at which statistical inference is uninformative.

Finally, as is typical of observational data analyses, we could not rule out all possible threats to causal inference. Numerous possible confounders could vary across time and space, including the other policies mentioned here, informal encouragement by government officials to wear masks or improve hygiene, changing business practices, and social norms regarding distancing. That said, including census division by day and county fixed effects in our model and examining prepolicy trends helped us push in the direction of causality. The number of confirmed COVID-19 cases in the US grew rapidly during the sample period, going from just 30 on March 1 to 978,047 on April 27. Part C of the appendix shows the number of counties with any COVID-19 cases on each day.20 On March 1 the vast majority of counties had zero cases; across all days, 49 percent of unweighted county-by-day observations were zero. However, counties with zero cases tended to have low populations, so our population weights limited the influence of these counties on the results.

Exhibit 1 illustrates the coverage of the US population by social distancing policies over time. Shelter-in-place orders were generally the last policy to be implemented, and adoption of them was uniformly lower than for the other policies. On March 1 no jurisdiction had implemented all four measures. By March 22 nearly 25 percent of the US population was covered by all of the measures. This rose to approximately 65 percent by March 29 and 95 percent by April 7, when the last shelter-in-place order took effect. Relative to the reference category of zero to four days before implementation, shelter-in-place orders led to statistically significant ($p<0.01$) reductions in the COVID-19 case growth rate of 3.0 percentage points after six to ten days, 4.5 percentage points after eleven to fifteen days, 5.9 percentage points after sixteen to twenty days, and 8.6 percentage points from twenty-one days onward. Because the model held constant the other types of policies, these estimates should be interpreted as the additional effect of shelter-in-place orders beyond the effects of shutting down schools, large social gatherings, and entertainment-related businesses. This additional effect may come either from the requirement or strong advisement to shelter in place aside from essential activities or from the accompanying closure of any nonessential businesses that remained open. We did not observe any statistically significant placebo effects of shelter-in-place orders in the periods before implementation, which gives credence to a causal interpretation of our main results. If anything, the pre trend appears to point upward, which would make our estimates in the postpolicy period conservative.

What is Neural Network ?

An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process

information.The key element of this paradigm is the novel structure of the information processing system. It is composed of large no. of highly interconnected processing element (neurons) working in union to solve specific problems. ANN‘s like peopling, learning by example. An ANN is configured for a specific application. such as pattern recognition or data classification, through a learning process .Learning in a Biological system involves adjustments to the synaptic connections that exist between the neuron.

Why use Neural Network ?

Neural network with their remarkable ability to derive meaning from complicated or imprecise data can be use to extract pattern and detect trend that are too complex to be noticed by either human or other computer techniques. A trained neural network can be thought of as an “expert” in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer “what if” questions. Other Advantages Include: • Adaptive Learning: An ability to learn how to do tasks based on the data given for training or initial experience. • Self-Organization:An ANN can create its own organization or representation of the information it receives during learning time. • Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability. • Fault Tolerance Via Redundant Information coding: partial destruction of network leads to thecorresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Implementation of HCR :- HCR works in stages as preprocessing, segmentation, feature extraction and recognition using neural network. Preprocessing includes series of operations to be carried out on document image to make it ready for segmentation. During segmentation the document image is segmented into individual character or numeric image then feature extraction technique is applied on character image. Finally feature vector is presented to the selected algorithm for recognition.Here this extracted features are provided to NN for recognition of character.

Machine Learning Classifiers

A classifier in machine learning is an algorithm that automatically orders or categorizes data into one or more of a set of “classes.” One of the most common examples is an email classifier that scans emails to filter them by class label: Spam or Not Spam.Machine learning algorithms are helpful to automate tasks that previously had to be done manually. They can save huge amounts of time and money and make businesses more efficient. Machine learning classifiers are used to automatically analyze customer comments (like the above) from social media, emails, online reviews, etc., to find out what customers are saying about your brand.Other text analysis techniques, like topic classification, can automatically sort through customer service tickets or NPS surveys, categorize them by topic (Pricing, Features, Support, etc.), and route them to the correct department or employee.

- Classifier: An algorithm that maps the input data to a specific category.

- Classification model: A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.
- Feature: A feature is an individual measurable property of a phenomenon being observed.
- Binary Classification: Classification task with two possible outcomes. Eg: Gender classification (Male / Female)
- Multi-class classification: Classification with more than two classes. In multi class classification each sample is assigned to one and only one target label. Eg: An animal can be cat or dog but not both at the same time
- Multi-label classification: Classification task where each sample is mapped to a set of target labels (more than one class). Eg: A news article can be about sports, a person, and location at the same time.

Artificial Intelligence :- The idea of reading Handwriting characters, digits, and words by computer systems can be argued to be an imitation of a human being. In other words, such a system can be argued that they use artificial intelligence to read handwriting from images or any Handwriting source, Artificial intelligence refers to intelligence that is demonstrated by machines The term is used to describe computer or machines that can mimic "cognitive" functions that are associated with the human mind. Artificial intelligence allows the machine tolearn from experience, adjust to new data (inputs), and perform tasks that can be performed byhumans.

Machine Learning :-Machine learning technology is inspired by psychology and biology that focus on learning from a set of data. The central assumption is that machines can learn to perform given tasks by learning from data. A machine learning model is provided with trainingdata that is specific to the given problem domain and the solution to each instance of the problem. That way, the model learns how to solve certain problems based on learning.

Artificial Neural Network (ANN) :-Artificial Neural Network (ANN) refers to information processing paradigm or computing systems that are inspired by biological neural networks that constitute the human brain . The systems are not identical to the biological neural systems, but they are designed to process information the same way the human brain and animal brain process information . The networks are composed of many interconnected neurons working in unison to achieve specific goals . Just like the human brain, ANN learns from example. Hence, an ANNcan be configured for an application, such as data classification or character recognition throughthe learning process. The learning process involves adjusting the system to a connection . The artificial neural network comprises a network of multiple simple processors, each with a small amount of local memory.

Deep Neural Network:- The neural network has layers of units where each layer takes some value from the previous layer. That way, systems that are based on neural networks can compute inputs to get the needed output[4] . The same way neurons pass signals around the brain, and values are passed from one unit in an artificial neural network to another to perform the required computation and get new value as output . Neural Processing Unit (NPU) architectures dedicated to energy-efficient DNN acceleration became essential. Despite the fact that training phase of DNN requires precise number representations, many researchers proved that utilizing smaller bit-precision is

enough for inference with low-power consumption. This led hardware architects to investigate energy-efficient NPU architectures with diverse HW-SW co- optimization schemes for inference. This chapter provides a review of several design examples of latest NPU architecture for DNN, mainly about inference engines The united are layers, forming a system that starts from the layers used for imputing to layer that is used to provide the output. The layers that are found between the input and output layers are called the hidden layer. The hidden layers refer to a deep neural network that is used for computation of the values inputted in the input layer, h enough training data, the deep neural network can be able to perform any function that a neural network is supposed to do. It is only possible if the neural network has enough hidden layers, although the smaller deep neural network is more computationally efficient than a more extensive deep neural network.

Convolutional Neural Networks(CNN):-CNN stands for Convolutional Neural Networks that are used to extract the features of the images using several layers of filters[10]. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers. The convolution layers are generally followed by maxpool layers that are used to reduce the number of features extracted and ultimately the output of the maxpool and layers and convolution layers are flattened into a vector of single dimension andare given as an input to the Dense layer (The fully connected network).

convolutional network were inspired by biological processes[9][10][11][12] in that the connectivity pattern between neurons resembles the organization of the

animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive

fields of different neurons partially overlap such that they cover the entire visual field.CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns to optimize

the filters (or kernels) through automated learning, whereas in traditional algorithms these filters are hand-engineered. This independence from prior knowledge and human intervention in feature extraction is a major advantage.

Convolutional neural network is used as a classifier for classifying the handwritten character from the input image. PAPER NAME FEATURE USED CLASSIFIERS ACCURACY “Efficient Offline Cursive Handwriting Word Recognition[5]” Zones and upper and lower profile of the word Minimum Distance Classifier and the SVM 80.76% “Handwritten Word Recognition Using MLP based Classifier: A Holistic Approach[6] Longest run features MLP classifier. 83% “Unconstrained Handwritten Word Recognition Using a Combination of Neural Networks[7]”, Feature extractor based on non□supervised clustering Feed-forward (FF) network, FF-SOM network 86.5% A Hidden Markov Model for Alphabet□Soup Word Recognition[8]” Joint boosting technique Hidden markov model 85% Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks[9] The hierarchical structure Multidimensional recurrent neural networks 91% A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers[15]. A CNN consists of three major components which are convolutional layer, pooling layer and output layer. The activation function that is commonly used with CNN is ReLU which stands for Rectified Linear Unit. Convolution layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights

and a small region they are connected to in the input volume. The pooling layer is a form of nonlinear down sampling. Max pooling is the most common which partition the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum. ReLU applies the non-saturating activation function . It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer. A rectified linear unit has output 0 if the input is less than 0, and raw output otherwise. Its value is obtained based on the formula which is as follows: $f(x)= \max(x,0)$ The softmax function is often used in the final layer of a neural network-based classifier. The softmax function squashes the outputs of each unit to be between 0 and 1, just like a sigmoid function. But it also divides each output such that the total sum of the outputs is equal to 1. The output of the softmax function is equivalent to a categorical probability distribution. Thus, softmax function calculates the probabilities distribution of the event over ‘n’ different events.

Hidden Markov Model (HMM) has been used in many handwriting recognition systems as a primary modeling component. HMM is a statistical Markov model that is used in a system that is supposed to assume the Markov process . It can be considered as the most straightforward dynamic Bayesian network. Hidden Markov Models are class pf probabilistic graphical models used for predicting a sequence of hidden variables from a set of observed variables More specifically, you only know observational data and not information about the states. In other words, there’s a specific type of model that produces the data (a Markov Model) but you don’t know

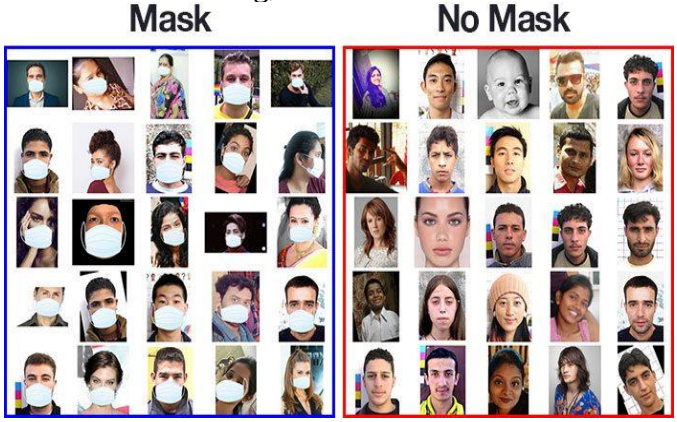
what processes are producing it. You basically use your knowledge of Markov Models to make an educated guess about the model's structure.

CHAPTER 3

Proposed System

Dataset:

The dataset which we have used consists of 3835 total images out of which 1916 are of masked faces and 1919 are of unmasked faces. All the images are actual images extracted from Bing Search API, Kaggle datasets and RMFD dataset. From all the three sources, the proportion of the images is equal. The images cover diverse races i.e Asian, Caucasian etc. The proportion of masked to unmasked faces determine that the dataset is balanced.
We need to split our dataset into three parts: training dataset, test dataset and validation dataset. The purpose of splitting data is to avoid overfitting which is paying attention to minor details/noise which is not necessary and only optimizes the training dataset accuracy. We need a model that performs well on a dataset that it has never seen (test data), which is called generalization. The training set is the actual subset of the dataset that we use to train the model. The model observes and learns from this data and then optimizes its parameters. The validation dataset is used to select hyperparameters (learning rate, regularization parameters). When the model is performing well enough on our validation dataset, we can stop learning using a training dataset. The test set is the remaining subset of data used to provide an unbiased evaluation of a final model fit on the training dataset. Data is split as per a split ratio which is highly dependent on the type of model we are building and the dataset itself. If our dataset and model are such that a lot of training is required, then we use a larger chunk of the data just for training which is our case. If the model has a lot of hyperparameters that can be tuned, then we need to take a higher amount of validation dataset. Models with a smaller number of hyperparameters are easy to tune and update, and so we can take a smaller validation dataset.
In our approach, we have dedicated 80% of the dataset as the training data and the remaining 20% as the testing data, which makes the split ratio as 0.8:0.2 of train to test set. Out of the training data, we have used 20% as a validation data set. Overall, 64% of the dataset is used for training, 16% for validation and 20% for testing.



The dataset we used has a total of 3835 pictures, with 1916 masked faces and 1919 unmasked faces. All of the photos were taken from the Bing Search API, Kaggle datasets, and the RMFD dataset. The proportion of photos in each of the three sources

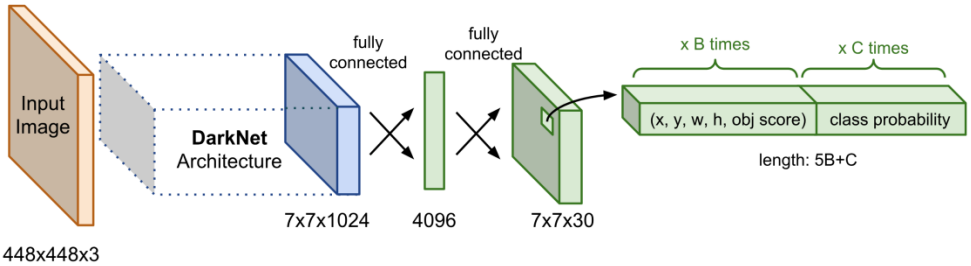
is the same. The photographs depict a variety of races, including Asians, Caucasians, and others. The percentage of masked to unmasked faces in the dataset determines whether it is balanced. The dataset we used has a total of 3835 pictures, with 1916 masked faces and 1919 unmasked faces. All of the photos were taken from the Bing Search API, Kaggle datasets, and the RMFD dataset. The proportion of photos in each of the three sources is the same. The photographs depict a variety of races, including Asians, Caucasians, and others. The percentage of masked to unmasked faces in the dataset determines whether it is balanced.

Our dataset must be divided into three parts: training dataset, test dataset, and validation dataset. The goal of splitting data is to avoid overfitting, which is when you pay attention to little details/noise that isn't necessary and simply improves the accuracy of the training dataset. The term "generalisation" refers to a model's ability to perform well on a dataset it has never seen before (test data). The training set is a subset of the dataset used to train the model. This data is observed and learned by the model, which then optimises its parameters. Hyperparameters are chosen from the validation dataset (learning rate, regularisation parameters). We can end learning with a training dataset once the model performs well enough on our validation dataset. The test set is the last collection of data used to evaluate a final model fit on the training dataset in an unbiased manner. The split ratio, which is highly reliant on the sort of model we're creating and the dataset itself, is used to split data. If our dataset and model necessitate a considerable amount of training, we employ a larger portion of the data solely for training, as in our example. If the model contains a large number of hyperparameters that can be tweaked, we'll require a larger validation dataset. We can use a smaller validation dataset since models with fewer hyperparameters are easier to adjust and update.
In our technique, we used 80% of the dataset for training and 20% for testing, resulting in a split ratio of 0.8:0.2 for train to test data. We used 20% of the training data to create a validation data set. The dataset is used for training 64 percent of the time, validation 16 percent of the time, and testing 20 percent of the time.

ARCHITECTURE

YOLO Architecture

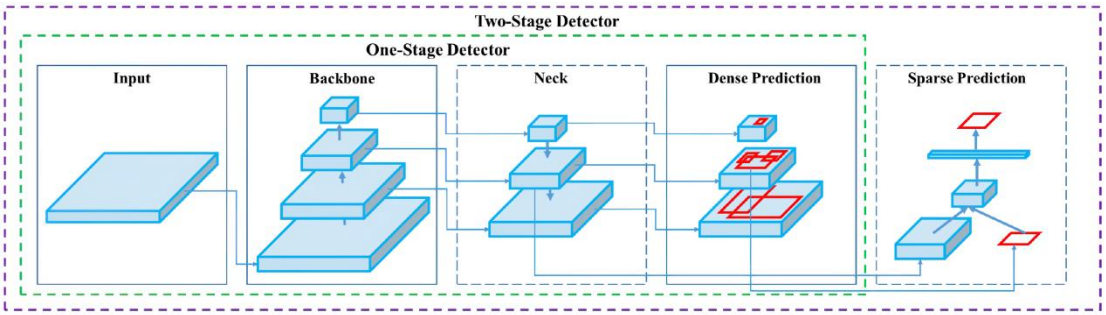
The Yolo algorithm stands for You Only Look Once, this algorithm is a state of art, which works on a real-time system, build on deep learning for solving various Object Detection as well as Object Tracking problems. The architecture of Yolo can be observed from the below Fig .



It can be observed from the above figure that the architecture contains the Input image layers which are responsible for taking the inputs that would be passed to further layers, input can be any image depending upon the use cases. Along the input layer comes the DarkNet Architecture, this is an open-source neural network for which framework is created with the help of C & CUDA, this framework features YOLO for object detection & object tracking.

Further, the architecture consists of the flattened layer which is densely connected with the convolutional layer which is also densely connected to pass the data from each node to other nodes in the architecture, similarly, this is passed to the output layer which gives 4-part values, those 4 parts describe the predicted value for the bounding box, denoted by x, y, w, h, along with the object detection score plus the probability of the predicted class. This YOLO is part of the One-Shot object detector family which is accurate & fast, there is also a Two-Shot object detector.

Two-Shot object detectors which are popular are R-CNN, Fast R-CNN, and Faster R-CNN, these algorithms are accurate in obtaining the results based on certain use cases but are slow as compared to that of Yolo, You Only Look Once is an algorithm that looks at the image at a single glance and based on that look predicts the bounding boxes related to certain classes, classes can be anything ranging from Dog to Car, or Gun to Tanks, this special feature makes Yolo stand out from others. Different types of object detectors based on a shot can be observed in Fig below.



From the above figure, we can find out different components, there are 4 different types of components

- Input The input to the detector can be an image or video based on the use cases specified in the research.
- Backbone The backbone of the object detector contains models, these models can be ResNet, DenseNet, VGG.
- Neck The neck in the detector acts as an extra layer, which goes in parallel to the backbone & the head.
- Head The head is the network that is in charge of the detection of objects based on bounding boxes

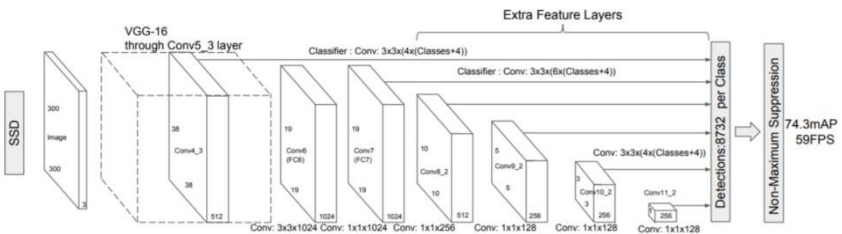
Covid Prevention System is a machine learning based system that is using the OpenCV and Tensorflow like python libraries to implement Yolo algorithm for object detection .Talking about the camera architecture it will be having two IR sensors one is to transmit radiation and another for receiving the signal that come after reflecting

the surface so these sensors are for detecting the heat from a body so they will detect thermal temperature of the object and there will be a other camera that is used to capture face of object and detecting whether mask is or not or if it has mask then whether he/she wearing in right manner or not ,same camera will also measure distance between the people and a small flash light below all camera for night vision.

SSD Architecture

The working of the Single Shot Detector algorithm relies on an input image with a specified bound- ing box against the objects. The methodology of predicting an object in an image depends upon very renowned convolution fashion. For each pixel of a given image, a set of default bounding boxes (usually 4) with different sizes and aspect ratios are evaluated. Moreover, for all the pixels, a confidence score for all possible objects are calculated with an additional label of 'No Object'. This calculation is repeated for many different feature maps.
In order to extract feature maps, we usually use the predefined trained techniques which are used for high quality classification problems. We call this part of the model a base model. For the SSD, we have VGG-16 network as our base model. At the training time, the bounding boxes evaluated are compared with the ground truth boxes and in the back propagation, the trainable parameters are altered as per requirement. We truncate the VGG-16 model just before the classification layer and add feature layers which keep on decreasing in size. At each feature space, we use a kernel to produce outcomes which depicts corresponding scores for each pixel whether there exists any object or not and the corresponding dimensions of the resulting bounding box.
VGG-16 is a very dense network having 16 layers of convolution which are useful in extracting features to classify and detect objects. The reason for the selection is because the architecture

consists of stacks of convolutions with 3x3 kernel size which thoroughly extract numerous feature information along with max-pooling and ReLU to pass the information flow in the model and adding non linearity respectively from the given image. For additional nonlinearity, it uses 1x1 convolution blocks which does not change the spatial dimension of the input. Due to the small size filters striding over the image, there are many weight parameters which end up giving an improved performance.
The block diagram [10] shows the working functionality of SSD. At the input end, we can see the VGG-16 being used as the base model. Some additional feature layers are added at the end of the base model to take care of offsets and confidence scores of different bounding boxes. At end part of the figure, we can see the layers being flattened to make predictions for different bounding boxes. At the end, non maximum suppression is used whose purpose is to remove duplicate or quite similar bounding boxes around same objects. There may be situations where the neighboring pixel also

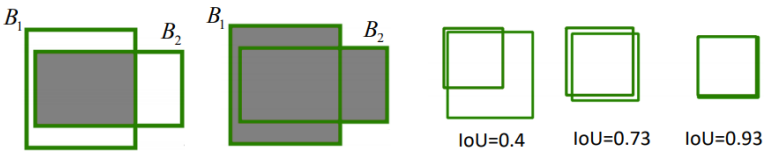


predicts a bounding box for an object with a bit less confidence which is finally reject.

The problem can be solved in two parts: first detecting the presence of several faces in a given im- age or stream of video and then in the second part, detect the presence or absence of face mask on face. In order to detect the face, we have used the OpenCV library. The latest OpenCV includes a Deep Neural Network (DNN) module, which comes with a pre-trained face detection convolutional neural network (CNN). The new model enhances the face detection performance compared to the traditional models. Whenever a new test image is given, it is first converted into BLOBS (Binary Large Object refers to a group of connected pixels in a binary image) and then sent into the pre- trained model which outputs the number of detected faces. Every face detected comes out with a level of confidence which is then compared with a threshold value to filter out the irrelevant detec- tions. After we have the faces, we need to evaluate the bounding box around it and send it to the second part of the model to check if the face has a mask or not.
The second part of the model is trained by us using a dataset consisting of images with mask and without mask. We have used Keras along with Tensorflow to train our model. First part of the training includes storing all labels of the images in a Numpy array and the corresponding images are also reshaped (224, 244, 3) for the base model. Image augmentation is a very useful technique because it increases our dataset with images with a whole new perspective. Before inputting, we performed the following image augmentations randomly: rotations up to 20 degrees, zooming in and out up to 15%, width or height shift up to 20%, up to 15 degrees shear angle in the counterclockwise direction, flip inputs horizontally and points outside the boundaries of the inputs are filled from the nearest available pixel of the input. For the image classification, it is now a common practice to use transfer learning which means using a model which has been pre-trained on millions of labels before and it has been tested that this method results in significant increase in accuracy. Obviously, the assumption here is that both the problems have sufficient similarity. It uses a well-structured and deep neural network that has been trained on a large amount of data set. Due to somewhat same nature of the problem, we can use the same weights which have the capability to extract features and later in the deep layers, convert those features to objects.

Training

At the training time, for each pixel, we compare the default bounding boxes having different sizes and aspect ratios with ground truth boxes and finally use Intersection over Union (IoU) method to select the best matching box. IoU evaluates how much part of our predicted box match with the ground reality. The values range from 0 to 1 and increasing values of IoU determine the accuracies in the prediction; the best value being the highest value of IoU. The equation and pictorial description of IoU is given



as follow:

$$IoU(B, B) = \frac{B1 \cap B2}{B1 \cup B2}$$

Hyperparameters

A hyperparameter is a parameter or a variable we need to set before applying an algorithm into a dataset. These parameters express the "High Level" properties of the model such as its complexity or how fast it should learn. Hyperparameters are fixed before the actual training process begins. They can be divided into two categories: optimizer hyperparameters and model hyperparameters.

Optimizer parameters help us to tune or optimize our model before the actual training process starts. Some common optimizer hyperparameters are as follows. Learning rate is a hyperparameter that controls how much we are adjusting the weights of our neural network with respect to the gradient. Mini-batch size is a hyperparameter that influences the resource requirements of the training and impacts training speed and number of iterations. Epochs are the hyperparameters that determine the frequency of running the model. One epoch is when an entire dataset is passed forward and backward through the neural network only once. Model hyperparameters are parameters that are more involved in the architecture or structure of the model. They help us to define our model complexity based on the different layers like the input layer, hidden layer, and output layer of a neural network.

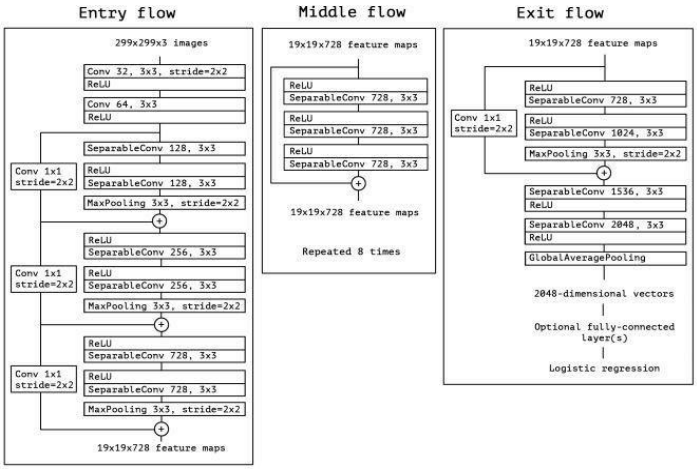
Initially, we trained with different values of hyperparameters by changing one and keeping the other constant and noted down the results in each case. We selected the hyperparameters that produced better performance through evaluation metrics. We have chosen the hyperparameters as follows: initial learning rate is taken as 0.0001, batch size is taken to be 32 and number of epochs as 20. In our case, the target size is also one of the hyperparameters which we kept (224, 224, 3) as it is default input.

Testing

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. 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. The results of different choices are shown below

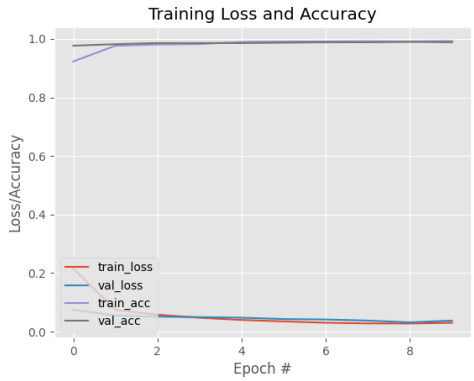
Xception

The complete form of Xception is "Extreme Inception". Basically, the Xception architecture [11] is designed with depth wise separable convolution layers with residual connections. These convo- lutional layers are linearly stacked with each other. This architecture is easy to define and easy to modify. If we compare it with Inception V2 and V3 then Inception V2 and V3 are much more complex to define.



Loss function

The loss of the overall detection problem can be broken into two main subsets: localization loss and confidence loss. The localization loss is just the difference between the default predicted bounding box and ground truth bounding box (g). For a given center (cx, cy), we try to alter the width and height of the box such as to



decrease the loss. Respectively, the equations for the localization and confidence losses can be defined.

CHAPTER 4

Required Tools

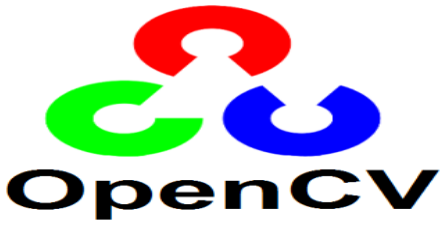
a. What is Python?

It's often used for constructing websites and applications, task automation, data analysis, and data visualisation, and its powerful typing and active binding make it appealing for both application development and writing. Python is easy to learn and use, and it lowers system maintenance expenses. For the object detection in this project, we will use the OpenCV python library. Pip, the Python package installer, can be used to install OpenCV. "pip3 install opencv-python" is the command to use. Because pip is the package installer in the Python language, we'll use pip3 to install OpenCVPython is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

b. What is OpenCV?

OpenCV is a widely used open-source library for handling computer vision challenges. It is mostly a library for dealing with computer vision. The quickest approach to install OpenCV to Python is using pip, assuming you have Python 3 pre-installed on your system. You can do so by entering the command line below into your command prompt. It primarily focuses on image processing and image capture. Then you can see whether or not the person in front of the camera is wearing a mask at the time.



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.

c. Tensor Flow

TensorFlow is a free and open source software library for machine learning and artificial intelligence. It's an open source Machine Learning platform that's encrypted. It can be used to train the model and deal with complex problems It contains various libraries and packages that assist you in quickly completing your project.

TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation) based simulations. Best of all, TensorFlow supports production prediction at scale, with the same models used for training.

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.

d. What is Keras?

Keras is a high-level TensorFlow API for creating neural networks. TensorFlow is a deep learning API developed by Google. It's written in Python and is used to make neural network implementation simple. It also aids in the computation of numerous backend neural networks

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.

FLOW CHART

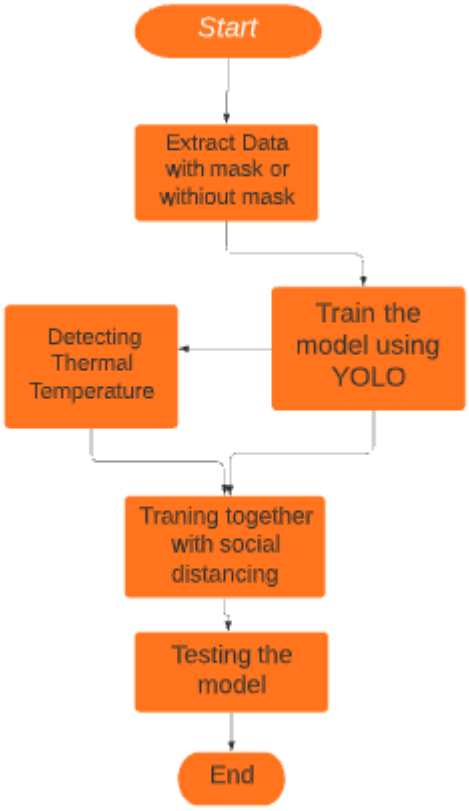


Figure 4. *Flow Chart*

Single Shot Multibox Detector (SSD) is an acronym for Single Shot Multibox Detector. It's a technology that uses a single deep neural network to recognise objects in photos. In a nutshell, it's used to detect objects in an image. SSD outperforms existing object detectors like YOLO and Faster R-CNN in both speed and accuracy by adopting a VGG-16 architecture as its basis architecture. The SSD architecture is depicted in the diagram below. Because training an SSD model from scratch requires a lot of data, I've used OpenCV to import pre-trained weights (Caffe Face Detector Model).
It has numerous other applications, such as in driverless vehicles, animals, and so forth.
cvNet = cv2.dnn.readNetFromCaffe('../input/caffe-face-detector-opencv-pretrained-model/architecture.txt','../input/caffe-face-detector-opencv-pretrained-model/weights.caffemodel')

g. Function

- The JSON Function retrieves the data of the bounding box in the training dataset from a json file.
- Gamma correction, or simply gamma, is a nonlinear operation used in video and still image systems to encode and decode luminance or tristimulus values. To put it another way, it's utilised to add some light to an image. If gamma is less than one, the image will be darker, and if gamma is greater than one, the image will be brighter.

Covid Prevention System will be a complete system that will detect face masks and social distancing and also for the temperature. So when people will take entry into the building or metro stations then the camera will detect object/person and then it will

check for face mask on their mask and show a label round the person that will show up whether how much accurately system has detected face masks . Same case with the social distancing a label will surround the object ,now when any rule violates then it will start to show red or otherwise green. For thermal temperature detection it will start to display the body temperature of that particular person on top right of label as soon as person start detecting in the video.
So cameras system will give two output one for thermal temperature and another for face mask detection and model will check is mask accurately wear or not if not then a buzzer sound will come and if social distancing is small than some threshold then it will also give sound of buzzer and in display it will show red labels, same for the temperature when it crosses some threshold temperature then it will show red label and buzzer sound can be hear so that person can go for further testing manually.

Working flow

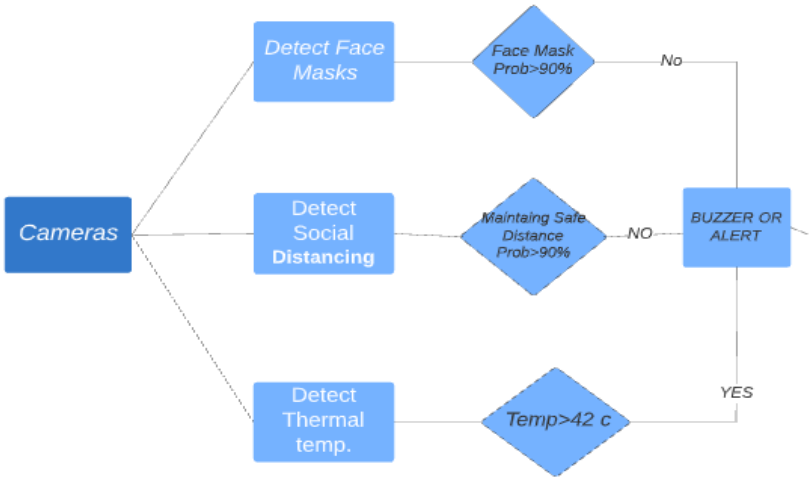


Figure 5. *Working Chart*

Covid Prevention System will be a complete system that will detect face masks and social distancing and also for the temperature. So when people will take entry into the building or metro stations then the camera will detect object/person and then it will check for face mask on their mask and show a label round the person that will show up whether how much accurately system has detected face masks . Same case with the social distancing a label will surround the object ,now when any rule violates then it will start to show red or otherwise green. For thermal temperature detection it will start to display the body temperature of that particular person on top right of label as soon as person start detecting in the video.
So cameras system will give two output one for thermal temperature and another for face mask detection and model will check is mask accurately wear or not if not then a buzzer sound will come and if social distancing is small than some threshold then it will also give sound of buzzer and in display it will show red labels, same for the temperature when it crosses some threshold temperature then it will show red label and buzzer sound can be hear so that person can go for further testing manually.

Although there has been a confusion between the two terms Image Classification andObject Detection, often meaning them to be the same, they are completely different. ImageClassification performs identification of objects in images while Object Detection performsidentification of the objects including its location in the images. Both of these terms arewidely popular in Computer Vision tasks (Merkulova, Shavetov, Borisov and Gromov,2019). They can be used in every field possible such as healthcare, defence, sports, andvarious other industries.The next question that arises is whether Object Detection and Tracking are the sameterms or not. Yes, Object Detection and Tracking are two very similar terms in the way theyare functioned. They are basically designed to perform the same functionality but with a littledifference. Object Detection is used to detect objects present in an image or in multipleimages where an object is stationary while Object Tracking performs detection on videos,that is, it keeps a track of the following object detected while it is moving (Porikli andYilmaz, 2012). A video is a combination of fast-moving frames and thus identification of theobjects and their location from every frame is performed by Object Tracking.Object detection can be stated as a fundamental problem in the computer vision and theimages domain. It intends to detect objects in the video that belong to specific classes such ashumans, vehicles, and more. The deep neural network models like CNN have dominated thebenchmarks of object detection. Pre-trained models like the MS COCO7, has more than 896Kobjects and over 123K images in the training and validation set and almost 80K images withmore than 80 categories in the testing set. With the help of supervised learning techniqueslike data augmentation, these models are trained.The best approach to building a model to perform object detection is with the help of aSliding Window technique (Glumov, Kolomiyetz and Sergeyev, 1995). The sliding windowapproach is where an image is divided into particular sizes and regions and then according tothe region, the respective classes are classified as shown in Figure 1. Here, in this research,the sliding window will detect all the people present in the video footage and form abounding box around it. A CNN classifier will state a confidence value where it willrepresent on the certainty that the window contains an individual or not. Then for each andevery region, a CNN will be passed which will extract the features and then further pass it onto the classifier and the regressor.

The state-of-the-art approach of this paper is capable of deploying pre-trained models.Pre-trained models can achieve good results to detect 2D object even with the different imagequalities, angles, and camera models. Hence, this paper uses state-of-the-art pre-trained deeplearning models that are used in detecting pedestrians and their face masks. These models canbe used in the monitoring of social distancing as explained in the following sections.A Social Distancing Detection and a Face Mask Detection model is trained and deployedto identify whether people are violating the social distancing measures and wearing a mask ornot. YOLOv4 along with DBSCAN clustering is used for recognizing the potential intruders.The face mask classification model uses the train and deploy the model in orderto identify the people whether they are with or without a mask. The project description isgiven below.

You Only Look Once Version 38 is a real-time, state-of-the-art object detectionsystem which is pre-trained on the COCO9dataset. It has a resolution of

416*416 and is usedin this research for the purpose of obtaining bounding boxes of individuals in the videos.8 https://pjreddie.com/darknet/yolo/9https://cocodataset.org/#download12There are many ways for gaining the position of an individual such as obtaining thecenter of the bounding box, that is, the midpoint of the bounding box. Other options includeOpenCV's bird's eye perspective. This project focuses on the first method to obtain thedistances between every individual. First, the model detects individuals in the frames andtheir faces. It then puts a red or green bounding box around the individual and his/her face todetermine if they are safe or not and wearing a mask or not. In order to reduce the complexityof the project, the regions that do not contain the object can be discarded. The process ofextraction of such regions is called Region Proposals. These algorithms are proposed so thatthe Regions of Interest (ROIs) can be selected. Therefore, one of the best approaches to ROIcan be found by deploying the Region-based Convolutional Neural Network (R-CNN) (Liaoet al., 2019).

Dual Shot Face Detector is utilized in this research to detect the individuals' faces. It is a method which derives from SSD and offers a Feature Enhance Module (FEM) to transfer original features to expand the single shot to the dual shot detector (Li et al., 2019). Conventional face detectors such as the MTCNN (Xiang and Zhu, 2017) or the Haar- Cascades (Tej Chinimilli et al., 2020) are not useful for this research as it lacks in detecting faces that are in low-resolutions or faces that are covered. DSFD is a bit complex and heavy on the pipeline but it delivers accurate results. It is widely used where detections are in a more large-scale ranged-orientations. As this research is working on video frames, the probability of encountering blurred faces is high and thus with the help of DSFD none of them will be missed. The blurriness could occur due to various reasons such as the face being out of focus or any random rapid movements or noise during the capture of the video.

A somewhat modified ResNet50 model whose layers have been pre-trained on ImageNet is used to classify individuals based on whether they are with a mask or without a mask. Figure 2 depicts the basic pipeline behind the working of the Social Distancing and the Face Mask Classification model. The basic methodology behind this algorithm is to first, divide the video into frames and detect people and their faces in every frame, individually. Later on, the frames are combined which then again forms a video. It works as follows:

1. Capture the video.

2. Read the video by dividing it into a number of frames.

3. Else, detect persons in each frame and get the bounding boxes around them with the help of YOLOv3.

a. If it reaches to EOF, stop.

4. Further, get the positions of the people with the help of DBSCAN to detect where the clusters are forming.

5. While detecting persons, detect their faces with the help of the DSFD model to detect whether they have masks on or not.

6. With the help of bounding boxes on the person and their faces, measure the distance between them and detect masks on them.

7. Create a results board on top of the video and display the results.

8. Create an output stream and then show the results.

9. Do this for every frame till it reaches to end of figure.

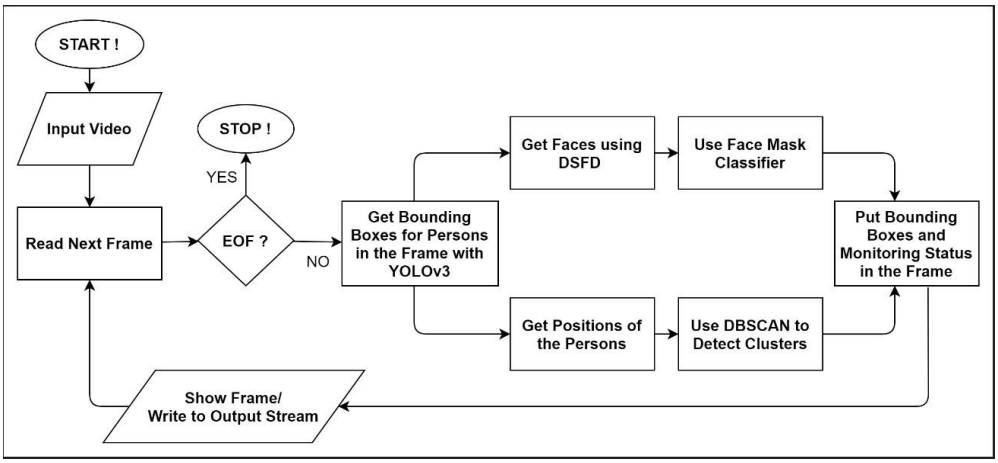


Figure : Social Distancing and Face Mask Detection Pipeline

CHAPTER 5

Result and Analysis

The method shows accuracy up to 95.77% and 94.58% respectively on two different datasets. We search optimized resultof parameters using the Sequential Convolutional Neural Netwzork model to detect the presence of masks correctly without causing over-fitting.

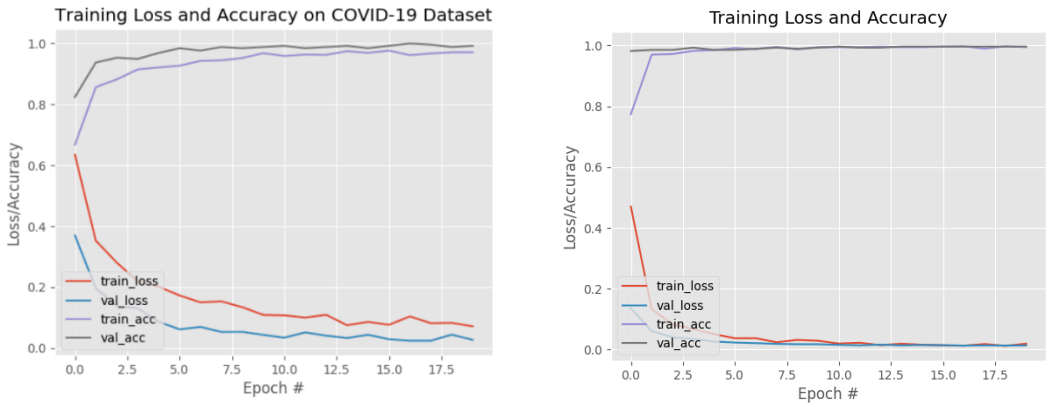


Figure 6. *Graphs*

COVID-19 face mask detector training with accurate guesture or loss curves demonstrate high accuracy and little signs of overfitting on the data. Now our project done for doing our science of computer vision and deep learning using **Python**, **OpenCV**, and **TensorFlow** and **Keras** to perform face mask detection. As we can see, we are obtaining ~99% accuracy on our test set in our project.
And the tools and language that are useful in making the project are easily available on the online sites for free.

Figure 7. *Thermal Graph*

A video feed is used to test the output. A face with a mask on is represented by a yellow colour rectangle drawn across the face, with the likelihood of being correct stated. A face without a mask is also denoted by a red rectangle drawn across it, as well as the likelihood of it being correct.
For detecting temperature we are measuring by taking a threshold so that will be 41 C . And image will come like this by accurately finding the temperature

Figure 8. *Face mask and social detection*

Detailed study of the Face mask detection System, analysing and installation of the tools and finding of the best algorithm for mask detection.
Initially, we are knowing to do a detailed study of the object detection so that we can get the idea of it still we will be seeing for the languages and tools that are required for making the project. Now we will be analysing the tools and languages after that we will be installing them on our system. And then we will be looking for the best\algorithm so that it will be easy for the implementation of the object detection.
Finally, we will be implementing the Face mask detection system using machine learning packages, tools, language and algorithm.

Future Work

At the outset of 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 techniques. It can 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 correctly in order to use their services. The implemented model will make a significant contribution to the public health care system. It could be extended in the future to detect whether or not a person is wearing the mask properly. The model should be enhanced further to recognise whether the mask is virus-prone or not, i.e. whether it is surgical, N95, or not.

A technology is being developed for detecting whether someone is using a face mask in a video selfie. Various types of conventional masks and acquisition settings were

used to test various analysis scenarios. The efficiency of the exploited face and face-feature detectors is critical to the performance of the developed technique. Wearing glasses had no harmful effects in this investigation. Rigid masks appear to be desirable since they limit the chances of incorrect face alignment. The created prototype can be especially useful in this case. As a result, we've suggested a promising face mask recognition model. By allowing individuals to validate the face mask via their webcam, a proof of concept as well as a development basis are provided to help reduce the spread of COVID-19. Furthermore, as a conformance aspect of mask wearing, this self-checking of mask wearing could be utilised by monitoring-related apps. Future research could look towards training a deep learning model with specific face-feature categories or correctly and poorly worn mask categories to develop very robust detectors. A technology is being developed for detecting whether someone is using a face mask in a video selfie. Various types of conventional masks and acquisition settings were used to test various analysis scenarios. The efficiency of the exploited face and face-feature detectors is critical to the performance of the developed technique. Wearing glasses had no harmful effects in this investigation. Rigid masks appear to be desirable since they limit the chances of incorrect face alignment. The created prototype can be especially useful in this case. As a result, we've suggested a promising face mask recognition model. By allowing

References

[1]. A.Kumar,A.Kaur,M.Kumar,Face detection techniques: review, Artificial intelligence review,volume.52 no.2pp.927- 928,2019.D.H.Lee,K.-L.CHEN,K.H Liou,C.Liu,andJ.Liu,Deep learning and control algorithms of direct perception for autonomous driving,2019.

[2]. Guangchengwang, yumiaoMasked face recognition data sets and application National natural science foundation of china 2020

[3]. Raza Ali, Saniya Adeel,Akhyar Ahmed Face Mask Detector July 2020

[4]. Z.-Q. Zhao, P. Zheng, S.-t.Xu, and X. Wu, Object detection with deep learningIEEE transactions on neural networks and learning systems 2019.

[5]. https://www.ijert.org/covid-19-facemask-detection-with-deep-learning-and-computer-vision

[6]. https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/