Real Time Mask Control Identification Model for Government Control Enforcement

Bandr AlSwyan
Rempee Kalia

American University
4400 Massachusetts Ave, NW

CSC-693-001

April 26, 2021

Table of Contents

1. Abstract

From 2020 to 2021 and further into the future, there is not a comfortable feeling anymore to walk outside or socialize with people, without the daunt of getting sick. With the rise of COVID-19, society has adapted to something that can hold us over until the Covid-19 virus has passed: face masks. The idea of face masks predicts one of the easier tools that will prevent the Covid spread. Evolving sciences have shown that the reduction of airborne transmissions will happen with face masks which will stop the spread of the virus. The governments have been forced to impose regulations which includes wearing of face masks in the public spaces, this has posed a major challenge to the existing facial recognition models, therefore new algorithms with increased accuracy and efficiency as well as reduced bias are needed to help security officers that have been relying on facial recognition using computer vision to enforce law and order to dispense their duties seamlessly and without danger.

Reflecting on this the idea that masks are one of the most important factors in controlling outbreaks is powerful in terms of data and technology. This is one of the reasons why artificial intelligence and the detection of images is important to track the movement of face mask wearing. There's several events in social times where face recognition needs to be precise in order to collect the right census. Types of social gatherings, protests, or national ceremonies can attract the attention of many people, and to detect whether everyone is wearing a mask needs to be quite efficient. Here, there are several questions that can be raised: Who will detect face masks? How should face masks be worn around the world? Who will control this? What type of technology would support this? We can lightly conclude that masks aren't the only thing that can stop the virus from spreading, but a large amount of scientific consensus states that they are extremely efficient in the reduction of the infection.The project is therefore seeking to give governments, police enforcement law and authority the tool to identify and potentially assist locating individuals who are not wearing face masks during the time of COVID-19 in public places therefore putting lives of fellow citizens in danger.

Currently, there are not any modern trained models that can identify people of the color in an accurate manner that is made to face track persons wearing mask; We are therefore using machine learning to identify a dataset of around 6000 images (of people who are wearing face masks and people who are not) from different colors, genders & ages to make the model un-biased which will analyze images of people efficiently without a bias in regards to pigmentation and or gender. We understand that there has been several ethical criticism of the existing facial recognition technologies in regards to gender and racial biases; to address this, we are using images from different regions of the world and focusing on a collection from different races of the world to understand the culturalism of the face mask mandate. The project uses the following software development technologies; Python, Tensorflow, Caffee DNN libraries, OpenCV, MobileNet and Keras. These will understand the simplicity of the model, in addition to detecting who is wearing a face mask and who is not, in order to have a better understanding of mask-wearing in a society's role.

The project sought to achieve the following objectives; (a) To enhance/add features in the development of the government mandate of Covid-19 face mask wearing. (b) To develop new ways that the method can be effective in everyday/long term living life in various cultures. (c) To

analyze the effect on gender and age of people [enhance the bias] result to be accurate). And (d) To compare different cultures with Covid-19 policies in their respective countries.

To realize these objectives, we will create a solution that will compile with a new policy of mask wearing where there will be several pros and cons. A sub-hypothesis that arises is: The future of Covid-19 can be decreased significantly with the mask mandates set in place, but will everyone follow the rule of mask wearing? In addition, how do we know that everyone is wearing a mask? This raises a question of who is actually wearing a mask during the COVID-9 pandemic. These areas have arisen in the research experiment and that will help us identify what it takes to build an AI model for government control.

Conclusively within the abstract, this project report describes the capstone-project for facial recognition and tracking using AI models; the project is intended to design a tool that can be used by the law enforcement across various countries to assist in enforcing the COVID-19 control measures specifically the wearing of face masks in the public spaces.. The environment in which the tool will run on is a python application powered by a Nero machine learning network that the user will be able to interact with by viewing the possibility percentage if a person in front of the camera is wearing a face mask or not. The face mask detector did not make use of any morphed masked images dataset. The model is proven to be accurate in mask detection, and is computationally efficient because of the use of the MobileNetV2 architecture hence making it easier to deploy the model to embedded systems (Raspberry Pi, Google Coral, etc.). This system can therefore be recommended and used in real-time tools or apps which require face-mask detection for safety purposes in addition to the outbreak of Covid-19 pandemic. Tagging this note, it is foreseen that face masks will be in society for at least another five years until mass cases are reduced. Therefore, many mask strategies are in the works of advancing this method for the future of our world. Research is showing that countries who follow the mask mandate are showing less severe results of the Covid-19 outbreaks. Within this, we will propose our topic of "Real Time Mask Identification Model for Government Control". Today and the future to come will merely rely on the government for guidance and support of tackling this virus, and we believe our project will well define the importance of artificial intelligence and governmental relations.This project can be integrated with embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed. Moving forward, it is in our hope that much improvement will be done to improve the political or ethical landscape of the application of AI to increase public awareness and support so as to harness the innumerable benefits we can get from it including such tools as facial recognition and detection for security and healthcare measures.

2. Introduction

In December 2019, SARS CoV -2 (COVID-19) cases were reported in Wuhan, China. Following 2020, the viral spread was so rapid that it was declared as a global pandemic implying that it became a public health threat of an international (global) scope. This necessitated the intervention of national and international public health organizations; and new guidelines were released including the mandatory (in some countries) wearing of face masks to prevent the spread of the airborne viral disease that could lead to cross infection.

Various governments therefore issued guidelines and directives to the health officers and security (police) officers to help the public in ensuring masking in public spaces as well as social distancing. Society has adapted to a new lifestyle of how to protect themselves, but this has also been met with resistance, some intentional while some are unintentional; such as deliberate wearing of face masks the wrong way or not wearing the face masks at all. To ensure that the citizens observe their civil duty of protecting themselves and protecting others, this project would provide a face tracking and face detection tool that is simple with reduced person to person contact & biases to enable the public to be safe as well as protected from the spread of the virus.

The purpose of this capstone project is to give the governments, police enforcement law and authority the tool to identify and potentially assist locating individuals who are in falls wearing medical face masks in the time of crisis and such COVID-19. The need for this technology arises in its simplicity as there is no current application available out there or they are either complex or not modern enough to provide the complete potential and use with the right reach. The project applies and is supported with huge dataset collected from different regions and of different gender and race to train the model that will reflect better understanding of the application thinks and interact in the face tracking in live video recordings of masked and unmasked in the target public spaces.

Face tracking serves a vital role in numerous applications such as visual surveillance, video indexing, facial expression recognition or human computer interaction [1]. In these applications it is needed to detect the faces, track them from one frame to another and analyze the tracks, for example to make sense of their behavior. In the modest form, a tracker makes an educated guess of the face trajectory by tracing its position in every single frame of the sequence. While this information may be satisfactory for some applications (e.g. identifying the presence of an intruder), other applications involve additional data, like knowing the positioning, extension or even the clear-cut contour of the faces at each frame (e.g. facial expressions recognition).

Our project will cover another scenario where the tracking and segmentation of faces along video sequences determine who is wearing mask, who isn't, as well as the position of the mask that will be labeled as correct or not correct by the application using color codes as well as detect the different skin colors of people (race). A facial recognition system is a computer app for routinely verifying or identifying a person from a digital video or an image frame from a video source. One of the ways we've chosen to accomplish this is by paralleling selected facial features from the captured or recorded image and a facial database. It is typically used in security systems and can be compared to other biometric such as fingerprint or eye iris recognition systems; it has also found use in medicine and healthcare to identify, monitor patients and as well to diagnose genetic, and behavioral conditions and medical conditions [2].

This paper will discuss a strategy that combines the mean shift algorithm for trailing with a representation of the images in terms of areas uniform in color [3]. As will be dispersed in the following sections of the project, mean shift can be applied as a robust and flexible algorithm for tracking, needing marginal training and computational resources. Sequentially, the use of regions enables a vigorous approximation of object and background models, as well as the exact definition of the face shape [4]. We also used in the python environment the following technologies; powered by Neural Networks machine learning and micro technologies such as, Caffee Libraries: A deep learning framework that is made with expression, speed and modularity. TensorFlow: Google detection model zoo that focuses on light-weight models training reflection. Quality is measured with model speed (slow, medium and fast) and model performance (mAP — mean average precision). OpenCV: multiprocessing Optimized performance webcam connection with stable FPS (Frame Per Second), and Keras: artificial neural networks, convolutional and recurrent neural networks.

This project is organized as follows; First, the Review section, we review the elementary mean shift algorithm and its application in object tracking. In the third section, Design Requirements/Details of Project, we detail our region-based approach as applied in this project, in the following section, Feasibility Discussion, we will discuss the feasibility of the project in areas such as politics, ethics, manufacturability and technical aspects of the project and what it means for the community, we shall then discuss the final implementation where we shall discuss some intimate details with occasional examples of the project as well as some calculations applied in the project; we shall then discuss the results and conclusions of the project as well as outline the recommendations obtained during the activity.

3. Review

This section provides an overview of similar projects and provides background material. Discussing the context and history of the project by describing and explaining the question/issue/problem, how it has been handled and what work has been done in the past. Include literature search results for the overall problem and context rather than the options for component parts.

This section will review available related works on facial recognition as well as masked facial recognition and data augmentation technologies;

The global pandemic that was declared in early 2020 of COVID-19 has continuously led to the mass cultural change which has been receiving mixed reactions; one of the shift in the culture is in the line of attire; the introduction of mandatory mask in public spaces; this has affected lives as well as technology use especially the face detection technologies; therefore the need to review and address the need for useful machine learning models that can detect and recognize as well as track both masked and unmasked faces as well as different color shades of the skin.

3.1 History of Face Recognition

The face recognition technology began as early as 1977 with the first automated system being introduced by Kanade using a feature vector of human faces; in 1983, two computer scientists, Kirby and Sirovich introduced the principal component analysis (PCA) for feature extraction. By applying the PCA, Turk and Pentland Eigenface was developed in 1991 and is considered a gross milestone in technology. Local binary pattern analysis for texture recognition was introduced in 1904 and was improved upon facial recognition afterwards by incorporating Histograms (LBPH). In 1996 Fisherface was advanced using Linear discriminant analysis (LDA) for dimensional reduction and can identify faces in different illumination conditions, that was an issue in Eigenface method. Jones and Viola introduced face detection techniques using ADABoost and HAAR cascades. In 2007 which is a wide range since the last significant advancement, a face recognition technique was developed by Skarbek and Naruniec using Gabor Jets that are similar to mammalian eyes. In this project, Eigenface, Fisherface and LBPH are used for face recognition and HAAR cascades are used for face detection.

3.2 Facial Recognition

In recent years, face recognition has garnered much attention and research on it has rapidly increased by not only computer engineers but also neuroscientists, since it has plenty potential applications in healthcare (identification and primary diagnosis), computer vision communication and automatic access control system. Particularly, face detection is an indispensable part of face recognition as the first step of programmed face recognition. However, face detection is not forthright because it has lots of variations of image appearance, such as pose variation (front, non-front), image orientation, occlusion, illuminating condition and facial expression [5].

Numerous innovative methods have been suggested to resolve each disparity listed above. For instance, the template-matching methods [6], [7] are applied in face localization and detection by computing the correspondence of an input image to a typical (standard) face pattern. The feature invariant approaches are used for feature detection [8], [9] of mouth, eyes, nose, ears, etc. The appearance-based methods are applied in face detection with EigenFace [10], [5], [1], neural network [8] [11], and information theoretical methodology [12]. Nonetheless, implementing the approaches altogether is still a great test. Such is the test that this project will face when it considers detection of masked faces as well as racial/ skin color variation.

To solve this, the project uses color segmentation to detect skin colors in colored pictures; While the input color image is typically in the RGB format, these methods typically use color components in the color space, such as the YIQ or HSV formats. That is due to the fact that RGB components are subject to the lighting conditions therefore the face detection may fail if the lighting condition fluctuates. Among multiple color spaces, this project used YCbCr components since it would save the computation time. In the YCbCr color space, the luminance information is contained in the Y component; and, the chrominance information is in the Cb and Cr components. Therefore, the luminance information can be effortlessly de-embedded. The RGB components were converted to the YCbCr components using the following formula:

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cb = -0.169R - 0.332G + 0.500B$$

$$Cr = 0.500R - 0.419G - 0.081B$$

In the skin color detection course, each pixel was classified as skin or non-skin on the basis of its color components. The detection window for skin pigment was determined based on the standard deviation and mean of Cb and Cr component, acquired using the training faces in the input images.

Similar to person-to-person communication that occurs in human social activities, automatic machine learning based facial expression recognition (FER) technology plays an import role in artificial intelligence (AI) by helping to comprehend the internal states of others, which can be used in a range of fields including security, psychology, health, human computer interaction and entertainment [13]. Recent years, deep learning based FER methods have already achieved reasonable accuracy [4] [5]. As the COVID-19 pandemic has now spread across the world, before the virus is eliminated, it can be foreseen that almost everyone will have to wear a face mask post pandemic while interacting in public [14].

3.4 Facial Recognition Using Mean Based Mean-Shift Algorithm (MBSM-A)

Mean shift is an iterative, non-parametric technique for finding out the mode of a density distribution signified by a set of samples [10]. Mean Shift Algorithm is one of the clustering algorithms that is attributed with the highest density points or mode value as the primary factor for developing ML. It is a kind of unsupervised machine learning algorithm. The algorithm operates on the concept of Kernel Density Estimation known as KDE. It is as well-known as the mode seeking algorithm. The Kernel is attributed with mathematical computation related to weightage to the data points. There are chiefly two common kernel functions associated with the mean Shift Algorithm for example the flat kernel and Gaussian Kernel. This algorithm is regularly used for computer vision and image segmentation.

Let *S* be a finite set in an n-dimensional Euclidean space *X,* the sample data. The sample mean with kernel *K* at a point $x \in X$ is defined as:

$$m(x) = \frac{\sum_{s \in S} K(s-x)w(s)s}{\sum_{s \in S} K(s-x)w(s)} \quad (1)$$

where *K* defines a zone of influence for *x* and *w(x)* is a weight function. The difference $m(x) - x$ is termed mean shift. The idea is to calculate the sample means for a condensed set of points $T \subset X$ and transfer the points in *T* towards their mean, until convergence. That is, if $m(T) = \{m(t): t \in T\}$, the mean shift procedure iterates and advances *T* until it finds a static point $T = m(T)$. Officially, a kernel *K* is a function defined in terms of its profile function $k: [0, \infty] \rightarrow R$, a non-increasing, non-negative and integrable function such that $K(x) = k (||x||2)$. The mean shift algorithm pursues the modes of the density estimation *q(x)* calculated with another kernel *H* which is termed as the shadow kernel of *K*:

$$q(x) = \sum_{s \in S} H(s - x)w(s) \quad (2)$$

The two kernels must satisfy the correlation $h'\,0\,(r) = -ck(r)$, where h and k are the profiles of H and K, in that order, $r = ||s-x||$ and $c > 0$ is some constant. This association guarantees that the mean shift vector $m(x)-x$ is in the gradient bearing of the density estimate $q(x)$, and has an adaptive step size, i.e., it changes fast when it is far from the mode and in short steps when it is near the mode. Two kernels K classically used are the unit Gaussian kernel and unit flat kernel the, whose shadows are the Gaussian kernels and Epanechnikov, respectively [15].

3.5 Mean Shift for Tracking

In object tracking the objective is to track the position of an object at each frame in the system. The evolving set T, consequently, comprises one point and the object centroid. In this perspective, a sample relates to the spatial coordinates of a pixel x, and has an associated sample weight $w(x)$, which delimits how probable it is that the pixel x fits to the object. The mean shift algorithm searches for the mode of the kernel density $q(x)$ figured with these weights.

Typically, weights are determined using a color-based object appearance model. For example, [6] works with a histogram of object colors and histogram back projection is employed to assign to each pixel the odds associated with its color, while [16] computes the weights with a degree of histogram similarity among object and background color distributions. Nonetheless, the mean shift strategy can be used to sample weight images computed with additional features besides color, like background subtraction results, texture similarity, etc. [17]. A particular enactment of the algorithm necessitates the definition of the kernel (scale and shape), the weight function and the shape, a model for the object, and extension of the final tracked object. The kernel scale is an indispensable parameter to the performance of the algorithm [17]. If the scale is excessively large, the search window might comprise background points that bear a resemblance to the object model, hence leading to an overestimation of the object size. An excessively large window could make the tracker converge to a region between compound modes instead of converging to just one of them. On the other hand, if the scale is exceedingly small, the shifts may well move within a flat zone of probability around the mode, leading to lessened object localization.

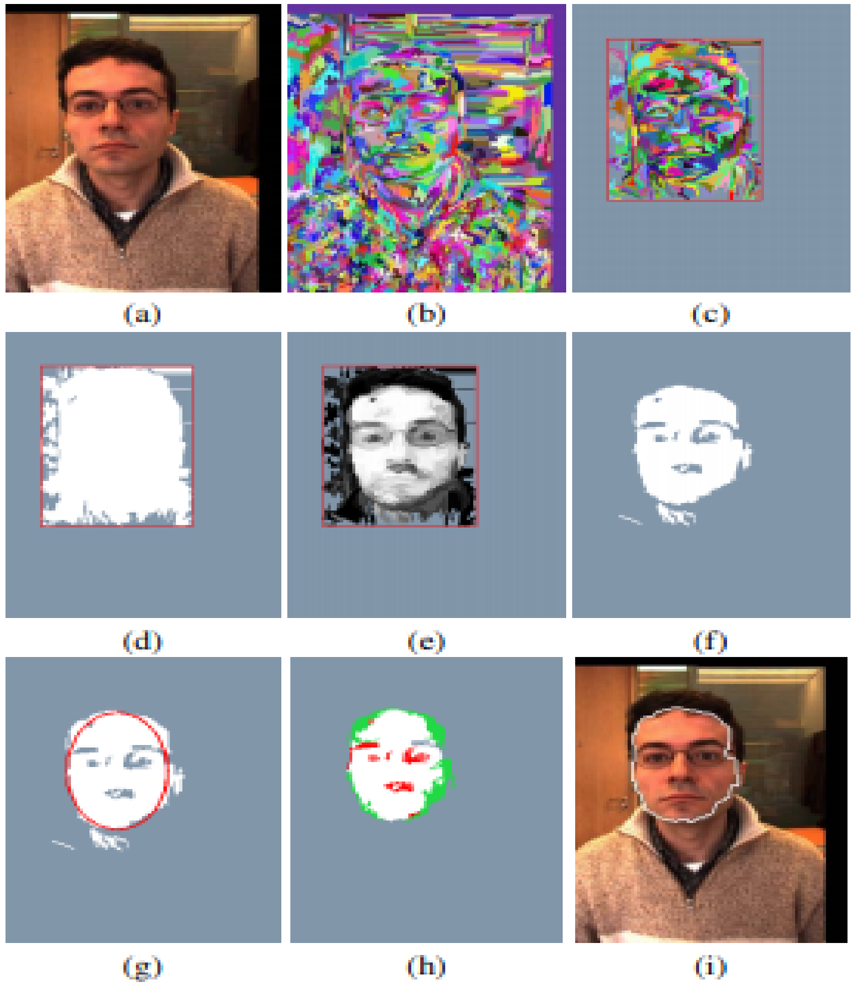


Figure 1. (a) Original image, (b) Partition, (c) Fitted partition, (d) Kernel (fitted partition mask (e) Weight image, (f) Initial object mask, (g) Fitted ellipse, (h) Final object mask (i) Smoothed object contour

4. Design Requirements

4.1 Specifications And Requirements For The Project

The project specification includes the following requirements; enhanced features to reduce the bias and improve efficiency of the system, developed innovative ways to make the project useful for a long time in the future, and to develop a method that can be reused in the future especially features in the AI development model that would add variety to how mask detection is done, This includes the assistant of the voice helper, possible color features of masks and the uncertainty (or confidence level) of detecting if an image has a mask on, to enhance the bias algorithms to ensure that there is reduced racial and or gender discrimination by training the models using different using large image database drawn from various regions namely Africa, India, and Middle East.

The end users of this tool will be the public and it has a significant cost benefit to the government, like any other innovative ideas in technology, the application will increase efficiency in people management as well as empowering and educating the public on the importance of putting on a face mask as this is the one way of containing COVID19; by using this technology, the government will be able to save on resources used to deploy law enforcement and security officers at the target strategic public spaces as the application will be helping in the real time monitoring and also be advising the users on proper use of masks. The project is a software that is meant to be compatible with security cameras and checkpoint face recognition and identification computers; the installation is expected to be cheaper awhile the results are remarkable thanks to the large pool of data from Africa, Middle East and Asia which tried to consider the skin color and gender, this should help to at least address the ethical concerns about the gender and racial biases seen in the existing facial recognition tools.

The models are using unsupervised learning technique therefore the machine is able to advance in its intelligence by itself; it is considerably safe as since it uses live 3D and 2D video analysis to recognize the faces as well as enable face tracking; the data being entirely biometric, there is minimal risk of hackers using the person's information to infiltrate their privacy or exploit them; facial recognition is considered one of the safest way of digital signature that is difficult for hackers to Target for cyber attack.

The application software is built on open source technologies, majorly and the only cost that the users and developers will worry about is the hardware and operations cost which isn't that much high; facial recognition is legal and provisionally ethical especially in the case of this project since it has addressed the key ethical and political issues that the technology has been facing, and that is racism and gender discrimination.

It is in our efforts and interest that in the future we will be able to address the new arising concern about ageism.We designed the project to meet the needs of the public and address the innovation gap or bias in respect to recognition of the masked faces and different skin colors; using open computer vision library, this activity was met with immense success. The difference between face identification/recognition and detection is that face detection is to identify a face from an image and locate the face while face recognition entails making the decision of whose face it is using an image database.

4.2 Functional Decomposition

The project uses deep learning (AI) for facial recognition; to do this, the models are trained using the mean shift procedure with a region based approach to identify and recognize faces; the algorithm that is developed is majorly region-based to allow for the precision in the distinction between two skin colors and different facial landmarks or regions; the approach is as an extension of the mean shift tracking algorithm and is used in reliance on the use of homogeneous image partition.

Objects are represented by sets of regions, which are used in building explicit models of object and background. These models allow the Bayesian estimation of pixel probabilities, which are therefore used de novo to define the weight images during the tracking reiterations and finally, together with a face shape model, to segment the final face. The precise definition of face contours populates a mechanism for adapting the kernel size while tracking faces through variations in scale, and for updating the object and background models.

The algorithm works with pixels that exist within a sub-image defined by a rectangular search window W and a partition P of the image into homogeneous regions in color. At every frame, the height and width of the search window are also the height and width of the bounding box of the object found in the previous frame, scaled by a fixed constant factor throughout the process. The window size is the same for every iteration in a frame.

The kernel is dictated by all the regions R in partition P that are completely included in W:

$$K(x) = \left\{\frac{1}{0}\right\} if x \in \left\{R \in \frac{P}{R} \subset W\right\} 0 \ otherwise \qquad (3)$$

An example of image partition, fitted partition and kernel is presented in figures 1(b),(c) and (d) respectively, where the factor used to scale the previous object size is set to 1.4, and the image partition has 500 regions.

The analysis of these images also takes into account the weight function of the objects and therefore render the final shape. To detect the wearing the face masks, MobileNet was used to train the models; the following parameters were taken into considerations;

4.3 Simulated and Real masked Data sets

For us to train masked FR models on publicly available data-sets, we firstly construct a masked FER dataset based on the Labeled Faces in the Wild (LFW) dataset [19]. We annotate the LFW dataset manually according to three types of facial masking (Mask, No mask, Wrong Wear), which all contain five types of facial orientations (up, left, center, right, down). Some pictures where the images were so blurry or a different clothing covered the face e.g., the hijab was difficult to distinguish whether the masks were on; are removed and 6000 out of 13000 samples are selected from the LFW dataset to finally obtain a LFW-FR dataset. MBSM-A is then used to process all samples in the LFW-FR dataset by putting a mask on the faces to obtain an M-LFW-FR dataset finally. LFW-FR and M-LFW-FR are released for academic study on either unmasked or masked FR.[1]

In order to train masked FR models that take facial orientation into consideration, we constructed a private KDDI-FR dataset. The facial features (Mask, No mask, Wrong Wear) of 12 subjects (5 females, 7 males) are photographed with 5 facial orientations (up, left, center, right, down) for each expression. Then, a total of 17236 samples are collected with 3447 samples for each orientation category and 1149 samples for each expression. MBSM-A is also

[1] https://pan.baidu.com/s/1ULHV6ShpnPUzn5eqPUYu0w Password: q7cf

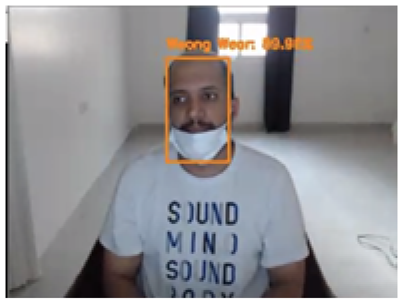
used to process all the samples in the KDDI-FR dataset and finally a masked RWMFRD-FR dataset (M- RWMFRD-FR) is obtained.

In order to test FR models trained on the above-mentioned datasets, we also constructed a real-world masked FER test dataset for model evaluation. We manually crawled 6000 masked facial expression figures (2274 natural, 1730 positive, 1996 negative) from the Internet by searching keywords, such as "smile, face, with mask" or "angry, face, with mask". The obtained real-world masked FR dataset for the test is called the M-FER-T dataset.

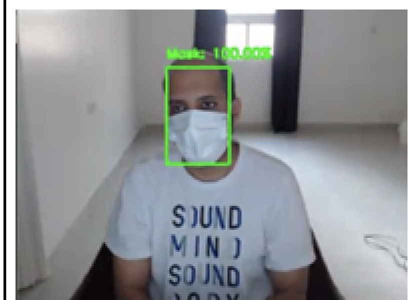
FR Data sets	Process methods	Train	Validation	Test
LFW -FR	Unmasked	0.9	0.84	0.3
	front view mask only	0.72	0.66	0.32
M-LFW-FR	front view mask only, face cut	0.76	0.7	0.45
	front/side view masks, face cut	**0.79**	**0.72**	**0.52**
Real-world masked face recognition dataset -FR, (RWMFRD-FR)	Unmasked	0.99	0.94	0.3
	front view mask only	0.54	0.52	0.29
M- RWMFRD-FR	front view mask only, face cut	0.7	0.68	0.43
	front/side view masks, face cut	**0.98**	**0.94**	**0.49**

The model use the data sets to train and test the Mask Recognizer, Voice Verification, recognition of masks, and finally be able to label the interfaces where the Red identifies “no mask”, the yellow identifies “mid-mask” wrong wear and the green identifies a “mask” on the face. As shown below, the confidence level, in percentage, is all identified

No Mask



Wrong Wear



Mask

Figure (3) The facial recognition and identification of masking status whether no mask, wrong wear or masked.

We added audio to the facial recognition to alert the users on the status change and also advise the user; this would decrease the need for manpower manning the checkpoints in places with reasonable levels of civilization. In addition this can come in handy with people that have minimum vision or similar disabilities.

4.4 Feasibility Discussion

The project should be considered to be environmentally friendly, this is evident by the nature of its availability to the public as an open source project; this means that developers who would like to work on similar project or improve on this project will be able to save energy as most of the work is already done, and such is the beauty of open source projects.

The project uses mostly open source technologies including Python and the respective libraries. Therefore the ease of manufacturing is increased as it can find many other applications even in the post COVID-19 pandemic in healthcare and learning institutions and even in ATM and Banks [17] and with this, the technology still proves that it is indispensable when it comes to economic feasibility; based on the current emergency, the change in the way of life that led to wearing face masks. Face masks cover up a significant portion of what facial recognition needs to identify and detect people essentially threatening the future of a multi million-dollar industry unless the technology can learn to recognize people beyond the coverings [18]; Meaning, current innovations enhance masked face recognition and tracking will gain more popularity in the future and this will also aid in increased recognition of faces in blurred images or part of faces as well as in mood recognition.

From a technological perspective, the project is much feasible in its pursuit and can be easily integrated into the already existing technologies (software and hardware). The approach also should provide the future developers with a direction towards eliminating biases and address the ethical issues being raised against these novel innovations.

Many facial recognition systems have a harder time identifying African-American, Asian and native groups than Caucasians, according to a study released by the National Institute of Standards and Technology [12]. There were higher rates of false positives among these groups when it came to confirming whether a photo matched another image of the same person in a database, the study says. This is known as one-to-one matching and is often used for tasks like unlocking a phone or checking a passport.

The subject of Artificial Intelligence (AI) as a whole have outlining bias from Ethicists; Facial Recognition haven't been left behind; in as much as we want to make work easier, our pure intention to do good may not just be so interpreted; for instance academicians wrote a journal in China and a facial recognition project before they were asked to pull it down since it had been used, allegedly, by the authorities to undermine the rights of a minority Muslim community in their country[15]; the intention here is that the government will not use this tool to infringe the human rights, which could lead to the public's disapproval whereif it increases the weight of the arms of government crushing on them.

The project is addressing a sensitive issue of public health; and health is a fundamental human right and with that also comes the ethical considerations of health such as, every patient has a say and autonomy in the choice of treatment they want and should be able to receive it whenever he proves the need. Nations did not democratically vote to put on the face mask, it came in as a guideline and slowly transformed into a regulation which makes face masks in public to be mandatory [14].

Facial recognition companies and organizations have for a sustained period used people's pictures without consent to train their algorithms. Civil liberty advocates contend that this technology threatens free speech and privacy, warning as well that there are almost no laws to prevent abuse of the surveillance tools. Hence, facial recognition has faced criticism for obtaining data without consent for facial-recognition algorithms to work desirably, they have to be trained and tested on large data sets of objects and images, ideally captured numerous times under different lighting conditions (illumination) and at distinct angles. Around the 1990s and 2000s, scientists generally recruited volunteers to pose for these photos — but most now collect facial images without asking permission [15]. This could at some point lead to uproar from the public, but the truth is that no opportunity comes without obligation, if the public still wish to enjoy new novel technological advancements, they should understand that nothing can be achieved without sacrificing a few of their data, their images, to train these models for the common good.

Questions whether with informed consent taking pictures of vulnerable populations have popped up from time to time and some have been turned down. The truth remains that there need to be ethical checkpoint on the practice of Facial recognition, and data should be carefully used in such a manner that no life is put to danger when the data are exposed or leaked; in the search for much data, researchers should improve efforts of generating simulated databases to train their models so as to avoid ethical conflicts [15].

5 Final Implementation

5.1 Training the mask recognizer

The facial recognition is trained on two models, one is to detect the face then the second is to recognize the face detected in the first case. Open computer vision uses ML to search for faces within a picture starting from the left; usually for faces, there is a voluminous supply of classifiers, around 6000 classifiers and above. The algorithm runs many tests to confirm the true positive or negative but with a degree of error limits. OpenCV uses cascades to shorten the time needed to assess the classifiers therefor braking the problem into a number of stages; for each block from the left, it does a brief and quick test; if that passes it does a slightly more detailed test, and the series continues; the algorithm will only detect a face if all these stages pass.

In order to train a custom face mask detector, we need to break our project into two distinct phases, each with its own respective sub-steps after creating the FM dataset; (I) *Training*: Here we'll focus on loading our face mask detection dataset from disk, training a model (using Keras/TensorFlow) on this dataset, and then serializing the face mask detector to disk (ii) *Deployment*: Once the face mask detector is trained, we can then proceed to loading the mask detector, doing face detection, then classifying each face as with_mask or no_mask and wrong_wear

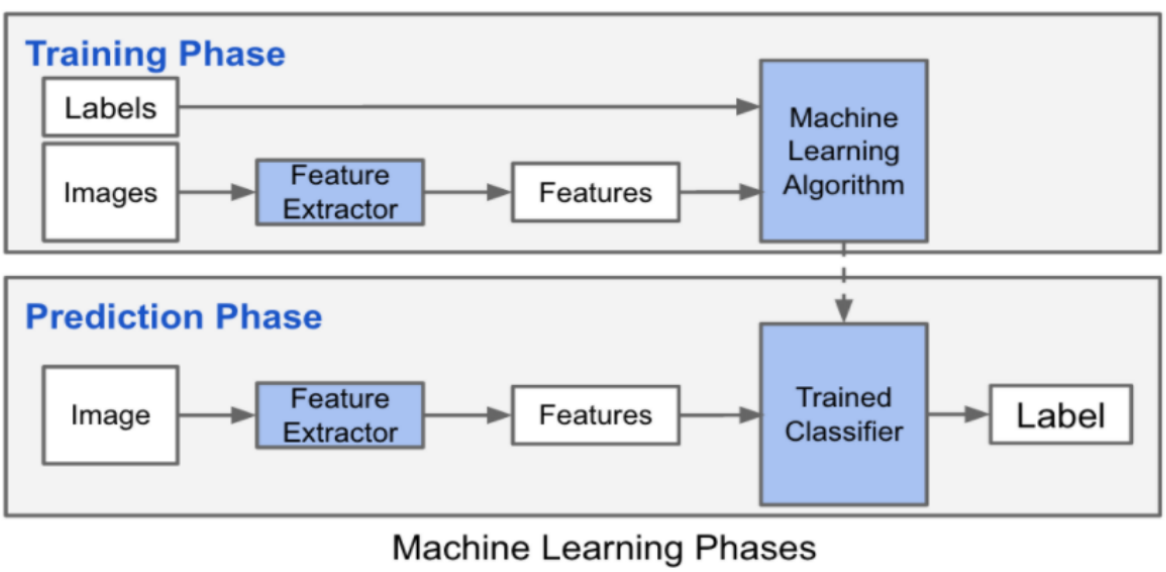


Figure 4: The Training and Prediction Phase

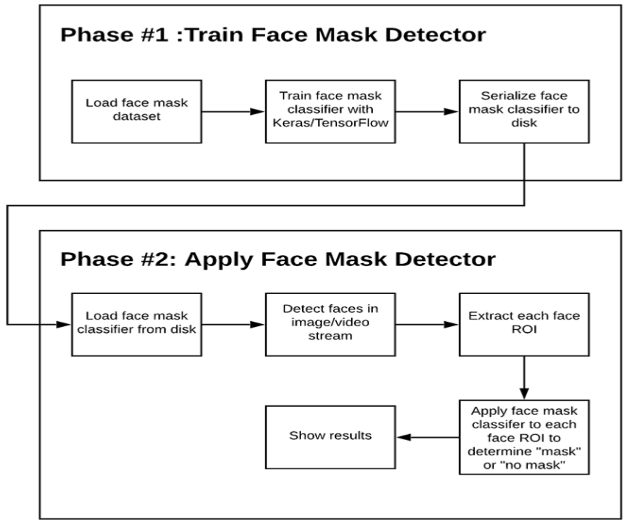


Figure 5: The ML phases

Below is a review of the sample codes that was crucial in detecting faces and training the mask organizer;

- **train_mask_detector.py:** Accepts our input dataset and fine-tunes MobileNetV2 upon it to create our mask_detector.model . A training history plot.png containing accuracy/loss curves is also produced
- **detect_mask_image.py**: Performs face mask detection in static images
- **detect_mask_video.py:** Using your webcam, this script applies face mask detection to every frame in the stream

5.2 Implementing face mask detector training script with Keras and TensorFlow

To use Keras and TensorFlow to train a classifier to *automatically* detect whether a person is wearing a mask or not and whether he is wearing it right we fine-tuned the Mobile-Net V2 architecture; note that to highly efficient can be embedded to devices with minimum computational capacity e.g Google Coral.

Deploying our face mask detector to embedded devices could reduce the cost of manufacturing such face mask detection systems, thus why we choose this architecture.

To train the mask detector, the following packages and dependencies are installed using PIP and imported to the train_mask_detector.py as bellow;

```
# import the necessary packages
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
```

```python
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import argparse
import os
```

The sets we used of tensorflow.keras imports allow for:

a) Preprocessing the data,

b) Data augmentation,

c) Loading image data,

d) Building a new fully-connected(FC) head,

e) Loading the MobilNetV2 classifier (we will fine-tune this model with pre-trained ImageNet_weights).

To binarize the class labels, scikit-learn was used; scikit-learn was also used to segment our dataset, and print a classification report. My imutils paths implementation will help us to find and list images in our dataset. And use matplotlib to plot our training curves.To realize the above, we parsed the following command line arguments;

```python
# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-d", "--dataset", required=True,
help="path to input dataset")
ap.add_argument("-p", "--plot", type=str, default="plot.png",
help="path to output loss/accuracy plot")
ap.add_argument("-m", "--model", type=str,
default="mask_detector.model",
help="path to output face mask detector model")
```

```
args = vars(ap.parse_args())
```

The arguments we created include –dataset, which is the path to the input dataset of the faces and and those with mask as well as with mask but not adorned right, --model which is the path to the resulting serialized face mask classification model and –plot as the path to our output training history plot which was generated using matplotlib.

We then defined our deep learning hyper-parameters as below:

```
# initialize the initial learning rate, number of epochs to train for,
# and batch size
INIT_LR = 1e-4
EPOCHS = 20
BS = 32
```

We defined our hyperparameter constants including initial learning rate, number of training epochs, and batch size. Later, we applied a learning rate decay schedule, which is why it was named the learning rate variable INIT_LR.

At this juncture our system was ready for the training data loading and pre-processing;

```
# grab the list of images in our dataset directory, then initialize
# the list of data (i.e., images) and class images
print("[INFO] loading images...")
imagePaths = list(paths.list_images(args["dataset"]))
data = []
labels = []
# loop over the image paths
for imagePath in imagePaths:
# extract the class label from the filename
label = imagePath.split(os.path.sep)[-2]
# load the input image (224x224) and preprocess it
image = load_img(imagePath, target_size=(224, 224))
image = img_to_array(image)
image = preprocess_input(image)
# update the data and labels lists, respectively
data.append(image)
labels.append(label)
# convert the data and labels to NumPy arrays
data = np.array(data, dtype="float32")
labels = np.array(labels)
```

After taking all of the imagePaths in the dataset then initialize the data and labels variables; we then loop over the imagePaths and load it while preprocessing the images.

Pre-processing steps include resizing to *224×224* pixels, conversion to array format, and scaling the pixel intensities in the input image to the range *[-1, 1]* (via the preprocess_input convenience function). The pre-processed images are then appended together with their associated labels to the data and labels lists accordingly.

We then went further to encode our labels partition the dataset and prepare for data augmentation as served by the code snippet below;

```
# perform one-hot encoding on the labels
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)
# partition the data into training and testing splits using 80% of
# the data for training and the remaining 20% for testing
(trainX, testX, trainY, testY) = train_test_split(data, labels,
test_size=0.20, stratify=labels, random_state=42)
# construct the training image generator for data augmentation
aug = ImageDataGenerator(
rotation_range=20,
zoom_range=0.15,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.15,
horizontal_flip=True,
fill_mode="nearest")
```

when we run the code above; the one-hot encoding on the labels means that our data will appear as below;

```
$ python train_mask_detector.py --dataset dataset
  [INFO] loading images...
  -> (trainX, testX, trainY, testY) = train_test_split(data, labels,
  (Pdb) labels[500:]
  array([[1., 0.],
  [1., 0.],
  [1., 0.],
  ...,
  [0., 1.],
  [0., 1.],
  [0., 1.]], dtype=float32)
  (Pdb)
```

Using scikit we segment our data for 80% training and 20% trial/testing.

During training, we apply on-the-fly mutations to our images in an attempt to improve generalization. This is known as data augmentation, where the random rotation, zoom, shear, shift, and flip parameters are established. We use the aug object at training time.

5.3 Preparing MobileNetV2 for fine-tuning

This is a three step process that involves loading the MobileNetV2 with pre-trained ImageNet weights leaving off the head of the network; constructing a new FC head, and appending it to the base of the previous head; we then freeze the base of the network.

```
# load the MobileNetV2 network, ensuring the head FC layer sets are
# left off
baseModel = MobileNetV2(weights="imagenet", include_top=False,
input_tensor=Input(shape=(224, 224, 3)))
# construct the head of the model that will be placed on top of the
# the base model
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)
# place the head FC model on top of the base model (this will become
# the actual model we will train)
model = Model(inputs=baseModel.input, outputs=headModel)
# loop over all layers in the base model and freeze them so they will
# *not* be updated during the first training process
for layer in baseModel.layers:
layer.trainable = False
```

With our data ready and model architecture in order for fine-tuning, we're now head to compiling and training our face mask detector network:

```
# compile our model
print("[INFO] compiling model...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy", optimizer=opt,
metrics=["accuracy"])
# train the head of the network
print("[INFO] training head...")
H = model.fit(
aug.flow(trainX, trainY, batch_size=BS),
steps_per_epoch=len(trainX) // BS,
validation_data=(testX, testY),
validation_steps=len(testX) // BS,
epochs=EPOCHS)
```

In the above snippet we compile using categorical cross-entropy, and launch the face mask training where our data augmentation object provides the batches of mutated image data.
Once the training is complete, the resulting data is evaluated on the test set.

```
# make predictions on the testing set
print("[INFO] evaluating network...")
predIdxs = model.predict(testX, batch_size=BS)
# for each image in the testing set we need to find the index of the
# label with corresponding largest predicted probability
predIdxs = np.argmax(predIdxs, axis=1)
```

```
# show a nicely formatted classification report
print(classification_report(testY.argmax(axis=1), predIdxs,
target_names=lb.classes_))
# serialize the model to disk
print("[INFO] saving mask detector model...")
model.save(args["model"], save_format="h5")
```

After evaluation and training of the serialization of the resulting data models to disk, we plot our accuracy and loss curve as below;

```
 # plot the training loss and accuracy
N = EPOCHS
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")
plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")
plt.plot(np.arange(0, N), H.history["accuracy"], label="train_acc")
plt.plot(np.arange(0, N), H.history["val_accuracy"], label="val_acc")
plt.title("Training Loss and Accuracy")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend(loc="lower left")
plt.savefig(args["plot"])
```

We then use Keras, Tensorflow, and Deep learning to train our face mask detector by executing the command line; $ python train_mask_detector.py --dataset dataset.

5.4 Implementing face mask detector for images with OpenCV

Now that the dataset training is good, we focus on the implementation of recognition, our detect_mask_image.py script requires three Keras /TensorFlow imports to (1) load our MaskNet model and (2) pre-process the input image. Open computer vision (OpenCV) is required for display and image manipulations. We loaded our face mask and face detector classifiers models. With our deep learning now committed to memory, we loaded a pre-processed image and input image. We then made a copy of the image upon loading and grabbed framed dimensions for future scaling and display purposes.

Preprocessing is handled by the OpenCV's blobFromImage function; we then resized the images to 300 x 300 pixels and performed mean subtraction. We performed facial detection then recognition and ensured, in the process that they met the confidence threshold before we extract the faceROIs. We computed the bounding box for particular faces and ensured that the box falls within the image's boundary.We then run the ROI through our MakNet model then annotate and display the results.

5.5 Voice verification

The audio assistant was added to issue instructions and guidelines to the end user in the form of notifications based on the three statuses.

The voice verification works on the principle that a person looks at the system, his real-time image is captured by the camera, the image gets classification after which the system

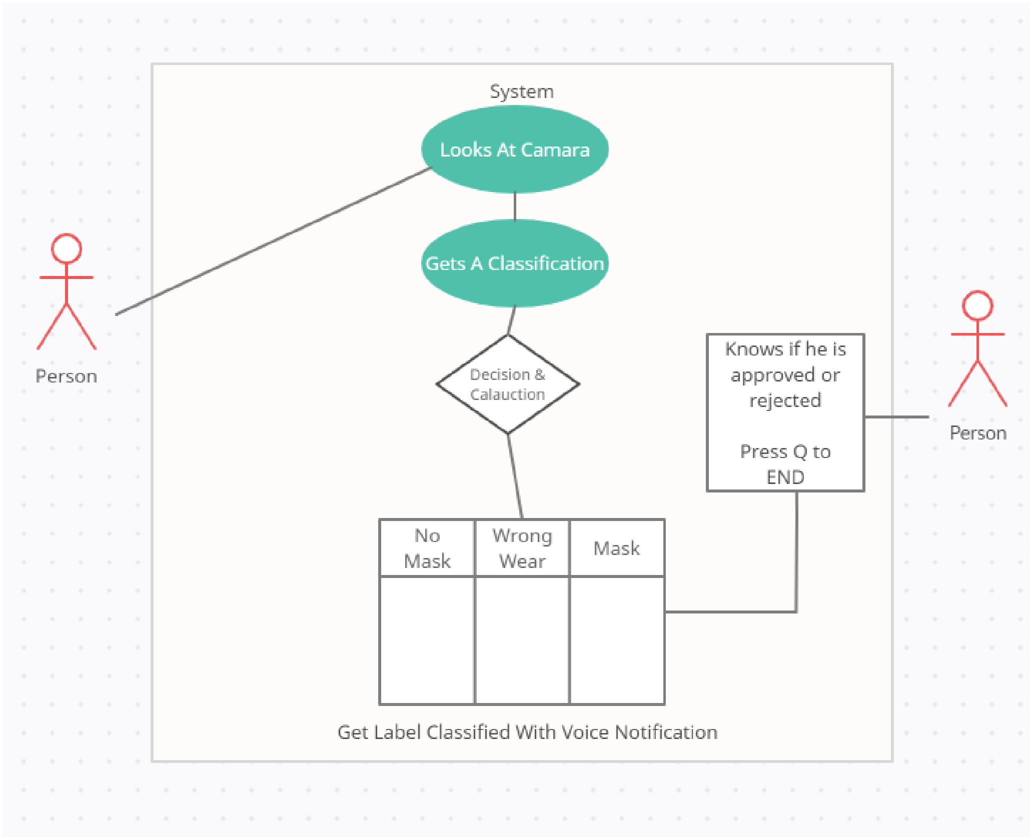


Figure 6: *Voice verification flowchart*

makes decision and calculation and relays one of the three options (No mask, Wrong wear, Mask) where the labels are classified along with respective voice notification and give an output to the person to know whether he is approved, that is mask on, or rejected that is, they are either a No mask or a Wrong wear. The flowchart below represents the process at a glance;

6 Results

The tool was able to recognize the persons with masks on and also to identify those that were with masks on but not adorned properly as well as those without masks.

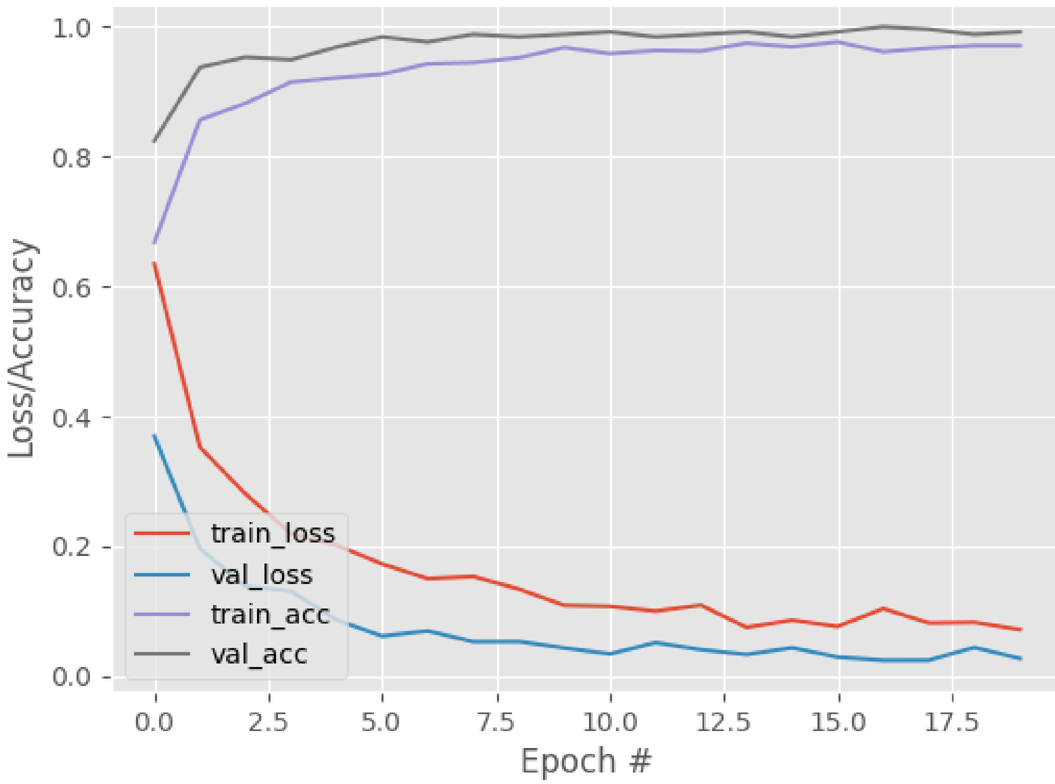
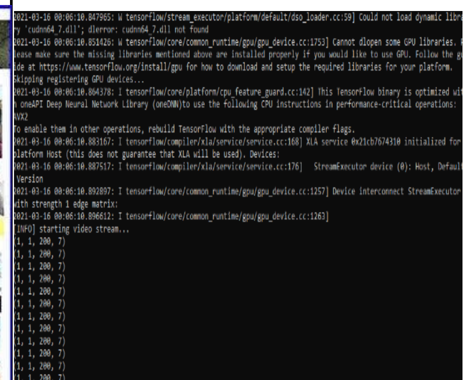


Figure 7: ***Training Loss and Accuracy on the dataset.*** *Face mask detector training accuracy/loss curves demonstrate high accuracy and little signs of over-fitting on the data.*

The plot shows a 99% accuracy on our test sets; which is a good thing. Additionally, the algorithm was able to perform the task at a record speed. The computational cost brought by the use of the partition is dependent to some extent on the size of the object being tracked. The average number of frames processed per second is 10, for a set of 20 sequences with faces of different sizes.

Our observations showed that some masked faces could not be detected because they were occluded by the face mask therefore mapping the facial locations to map and detect a face and recognize whether it's masked or not masked or wrong wear. This also emphasized how the model would be able to reflect on types of face masks or colors that are similar in the picture. The algorithms were pieced together to map every frame of our webcam stream therefore able to detect more than a face all at the same time.The model performed well on validation and training therefore the impressive percentage accuracy. The reduction in loss after a few iterations implies the model was performing just fine; since the data we have though diverse isn't that big, the nature of the models that is, unsupervised learning, should help define the learning curve with increasing volume of data to be processed.

The logical design for the face detection was designed to cope with speed as well by ensuring that at least one face was detected, if not empty press were returned. Performance of predictions in the batch proved to be more efficient. The figures below represent sample outputs from the make face detection;

No Mask	Mask	Terminal execution output
		

```
2021-03-16 00:06:10.847965: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic libra
ry 'cudnn64_7.dll'; dlerror: cudnn64_7.dll not found
2021-03-16 00:06:10.851426: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1753] Cannot dlopen some GPU libraries. P
lease make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the gu
ide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2021-03-16 00:06:10.864378: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized wit
h oneAPI Deep Neural Network Library (oneDNN)to use the following CPU instructions in performance-critical operations:
AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-03-16 00:06:10.883167: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x21cb7674310 initialized for
platform Host (this does not guarantee that XLA will be used). Devices:
2021-03-16 00:06:10.887517: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default
 Version
2021-03-16 00:06:10.892897: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1257] Device interconnect StreamExecutor
with strength 1 edge matrix:
2021-03-16 00:06:10.896612: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1263]
[INFO] starting video stream...
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
```

Figure 8: Sample output and datasets.

The final output was related to the faces labeled as with_mask, wrong_wear and without make; an audio was linked to the labels and gave voice notification on each end result in respect to one out of three choices recognized. With this recognition, it came to the conclusion that the model was about to detect whether a photo had a mask on or not. The regions of the photos were well scattered out within the data, but still worked to bring a conclusion to the set and resulted in identifying “masks” and “no masks”.

6.1 Overfitting

Figure 9: Training Loss and Accuracy Graph (Overfitting Graph)

The plot above shows the model showing a pattern of where the data was displaying overfitting patterns. This problem usually occurred because our data has a form of biases(from many parameters) that it holds. The form of regularization during the training phase of the model used a method that was iterative. the method has updated within the model to make the data a better fit. Because of these reasons, when we added new data, the better fit was recognized and affected the model's performance on the data set making it overfit.

To deal with the overfitting or decreasing this in the near future, we would want to stop the data training during the test of the new set, even the ones that include false mask wearing to ensure that it is not a bias model. This would reduce the number of overfitting and would make the network fit to the right level wanted. We believe that the best strategy would be to make the network break at the right level yet more efficient in identifying the types of photos so that when real-time is running, it will be able to detect the type of mask, no matter whether its the same tone as skin color, or other discrepancies are happening. While the data overfitting appears to be high in the graph above, a reset of a more diverse training method would be able to fix this upon reinitiating the training. This allows the training to stop in the right space between test error and testing error just as seen below.

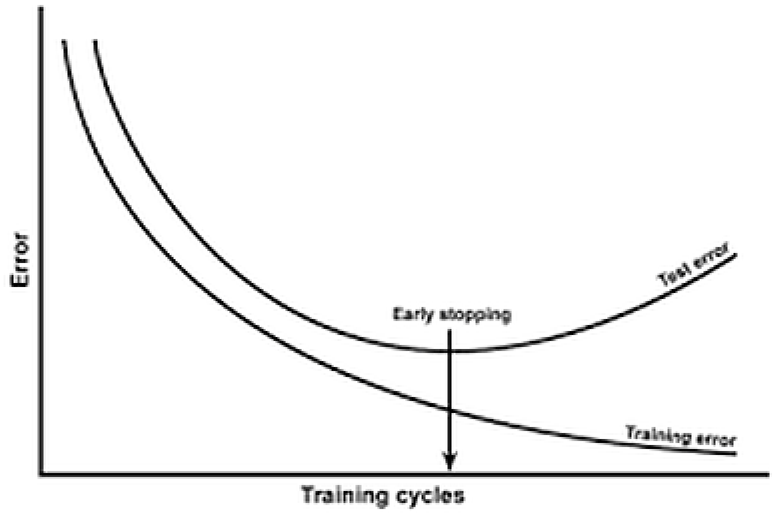


Figure 10: Training Cycle Early Break

6.2 Social and Ethical Challenges

While masks are one of the most effective ways of tackling the pandemic, it still takes the public as a whole to oblige to the situation that is affecting millions of people, even today. Masking wearing can bring political and other controversial statements and we can't agree that all countries are going to act the same. Throughout our data collection, we focused on the regions of Africa, Middle East, and North America. Here, the data shows that mask wearing is indeed a different aspect in many countries and that it has many conflicts regarding what type of mask should be worn. Our model has tested these different types of races and cultures to determine one thing: is the mask on or off? Within our findings, we have seen how the pandemic has affected people socially and the ethical changes that the government was obliged to make for the safety of millions of people.

There are specific countries where mask wearing has been successful such as Taiwan, Hong Kong, and South Korea, but we wanted to look at more politically challenged counties where the government was in the middle of some sort of administrative change or facing difficulties with cases. In the U.S, there was a record of more than 200,000 deaths [19] and the new cases were continuing to rise. Therefore, we tested our model to investigate how the mask wearing trend was looking. The data collection of the U.S. showed that the masks were very different and somewhat of a trend like. For people to essentially wear a mask, most were decorated, nude colored, or showed some kind of political message. If we compared this to our non-targeted data which was mainly from Asia, we could see that the masks were almost all the same on each person. The ethical challenges that the U.S. was facing was the mandate of people wearing the mask. With this type of situation in North America, our model would be most efficient because of the lack of mask wearing by society.

A note from the pandemic is that "50% of person to person transmission of the virus may occur through asymptomatic individuals". [19] The question is that almost all adults are in the spectrum of wearing a mask. This includes the senior status as well as young adults, where children are not obliged to. This made it difficult in tight cities where the highest rates were occurring per capita. The spread of the virus was predominant in populations in the metropolitan areas. By having a model such as the Face Recognition, we can conclude that researching groups per region will better understand these groups and target a more genuine public health message to promote practices. This helps facilitate greater understanding for experimentation for data on gender and race. One sub-conclusion that we found in our data collection is that women are more likely to appear with customized face masks from the U.S. than men. Meaning, the genders are comparatively selective of their appearance for North American.

7 Conclusions

With the advent of coronavirus, the public is in need of solutions to contain the spread of COVID-19; among the current guidelines is the wearing of face masks which is mandatory in some countries; the current face recognition tools were built before COVID-19 and the models did not take into account the wearing of face masks; in addition to the other false positives that has been detected with the facial recognition models especially when it comes t the detection of females and the people of color, the project is offering a viable solution to the public, and if taken well and put into good and responsible use, should be able to reduce the cost of operation of public healthcare by the government; additionally this tool can find use in security checkpoints to help advise the citizen to wear a mask and wear it properly; it will also help the security officers in people management in the public.

The program is support by Python applications powered with Neural network machine learning that the user will be able to interact with by viewing the possibility percentage if the person in front of the camera is wearing a face mask or not with a high accuracy that is intended to increase with increasing faces to train the models as it is learning through unsupervised learning.The face mask detector did not make use of any morphed masked images dataset. The model is accurate in that effect, and is computationally efficient thanks to the use of the MobileNetV2 architecture hence making it easier to deploy the model to embedded systems (Google Coral, etc.).

This system is therefore recommended for use in real-time tools and apps which require face-mask detection for safety purposes in regards to the outbreak of Covid-19 pandemic, it can be integrated with embedded security systems for application in airports, railway stations, offices, schools, and other public places to ensure that public safety guidelines are followed. Additionally, moving forward, it is in our hope that much improvement will be done to improve the political/ ethical landscape of the application of AI to increase public awareness and support so as to harness the innumerable benefits we can get from it including such tools as facial recognition and detection for security and healthcare measures.

The project has presented a 99% accuracy which our test models, and if more training can be done, is at it’s best result and operational since it uses unsupervised learning technique of machine learning. If there is more data available to train the models, this could provide further accuracy that we desire and would reduce the bias in facial recognition that is associated with illumination, and skin color, as well as facial regions. The mean-shift algorithm came in handy for faster and relatively high accuracy; in the future it is prudent to examine the use of shape information to reduce leakages in the final shape contour and to cope faster to variations in scale.

Moving forward we shall work on small improvements such as combining object with a dedicated with_mask class to enhance natural detection of people wearing masks that otherwise wouldn't be possible due to too much of the face being obscured and also reduce the computer vision pipeline to a unit step rather than sequentially applying face detection then mask recognition; this would increase much efficiency and speed and we look forward to make such improvements.

7.1 GitHub Repository

For the best representation of our report and project, we have hosted all files and sources using Github Repo that will hold all of the code and data to be found. There are several files for the model that support the code and how to execute it properly so that the model can detect the mask wearing. We will host an informational github repo that shows the findings and analysis of the AI detection (This report). The data will be also available for review where it will show a variation of the different regional races photos that this model chose to focus on throughout the testing phases.

In future cases for this model, it's possible to host the model somewhere else just by bundling or moving the code from its original repository. The repository is accessible through this link: https://github.com/balswyan/senior-capstone-spring-2021.

8 References

[1] R. C. Verma, C. Schmid, and K. Mikolajczyk,, "“Face detection and tracking in a video by propagating detection probabilities,," *IEEE. Trans. on Pattern Analysis and Machine Intelligence,* vol. 25, no. 10, p. 1215–1228, October 2003.

[2] J. P. Nicole Martinez-Martin, "What Are Important Ethical Implications of Using Facial Recognition Technology in Health Care?," *AMA Journal of Ethics,* vol. 21, no. 2, p. 22, 2019.

[3] Verónica Vilaplana, David Varas, "FACE TRACKING USING A REGION-BASED MEAN-SHIFT ALGORITHM WITH ADAPTIVE OBJECT AND BACKGROUND MODELS," in *Technical University of Catalonia (UPC),* , Barcelona, Spain, 2009.

[4] AnkurBansal, MukeshAgarwal, Anima Sharma, Anindya Gupta, "A Review Paper on FACIAL RECOGNITION," *International Journal on Recent and Innovation Trends in Computing and Communication,* vol. 1, no. 4, pp. 224 - 228, 2013.

[5] Inseong Kim, Joon Hyung Shim, and Jinkyu Yang, "Face detection," [Online]. Available: http://web.stanford.edu/class/ee368/Project_03/Project/reports/ee368group02.pdf. [Accessed 2021].

[6] I. Craw, D. Tock, and A. Bennett,, "Finding face features,," in *Proc.of 2nd European Conf*, 1992.

[7] A. Lanitis, C. J. Taylor, and T. F. Cootes, "An automatic face identification system using flexible appearance models," vol. 5, pp. 393-401, 1995.

[8] T. K. Leung, M. C. Burl, and P. Perona, "“Finding faces in cluttered scenes using random," in *IEEE int’l Conf. Computer Vision*, 1995.

[9] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* pp. 696-710, July 1997.

[10] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.

[11] T, Agui, Y. Kokubo, H. Nagashi, and T. Nagao, "Extraction of face recognition from monochromatic photographs using neural networks," in *Automation, Robotics, and Computer Vision*, 1992.

[12] M. S. Lew, "Information theoretic view-based and modular face detection,," in . *2nd Int’l Conf Automatic Face and Gesture Recognition*, 1996.

[13] Ligang Zhang, Brijesh Verma, Dian Tjondronegoro, and Vinod Chandran, "Facial Expression Analysis under Partial Occlusion: A survey," *Comput,* vol. 51, no. 2, pp. 1-49, 2018.

[14] Joan Mateu, "Which countries have made wearing face masks compulsory?," Aljazeera, 17 August 2020. [Online]. Available: https://www.aljazeera.com/news/2020/8/17/which-countries-have-made-wearing-face-masks-compulsory. [Accessed 19 April 2021].

[15] R. V. Noorden, “The ethical questions that haunt facial-recognition research,” *Nature News*, 18-Nov-2020. [Online]. Available: https://www.nature.com/articles/d41586-020-0 3187-3. [Accessed: 22-Apr-2021].

[17] J. Kobielus, “Will Facial Recognition Thrive in the Post-Pandemic Economy?,” *InformationWeek*, 30-Jul-2020. [Online]. Available: https://www.informationweek.com/big-data/ai-machine-learning/will-facial-recognition-thrive-in-the-post-pandemic-economy/a/d-id/1338491#:~:text=A 2019 study by Markets,of 16% through that year. [Accessed: 22-Apr-2021].

[18] A. Ng, “Your face mask selfies could be training the next facial recognition tool,” *CNET*, 19-May-2020. [Online]. Available: https://www.cnet.com/news/your-face-mask-selfies-could-be-training-the-next-facial recognition-tool/. [Accessed: 22-Apr-2021].

[19] Gary Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Tech. rep. (10 2008)

[20] Haischer, Michael H et al. “Who is wearing a mask? Gender-, age-, and location-related differences during the COVID-19 pandemic.” *PloS one* vol. 15,10 e0240785. 15 Oct. 2020, doi:10.1371/journal.pone.0240785