**ABSTRACT**

The end of 2019 witnessed the outbreak of Coronavirus Disease 2019 (COVID-19), which has continued to be the cause of plight for millions of lives and businesses even in 2020.As the world recovers from the pandemic and plans to return to a state of normalcy, there is a wave of anxiety among all individuals, especially those who intend to resume in-person activity. Studies have proved that wearing a face mask significantly reduces the risk of viral transmission as well as provides a sense of protection. However, it is not feasible to manually track the implementation of this policy. Technology holds the key here. We introduce a Deep Learning based system that can detect instances where face masks are not used properly.

As the pandemic slowly settles and such sectors become eager to resume in-person work, individuals are still skeptical of getting back to the office. 65% of employees are now anxious about returning to the office (Woods, 2020). Multiple studies have shown that the use of face masks reduces the risk of viral transmission as well as provides a sense of protection. However, it is infeasible to manually enforce such a policy on large premises and track any violations. Computer Vision provides a better alternative to this. Using a combination of image classification, object detection, object tracking, and video analysis, we developed a robust system that can detect the presence and absence of face masks in images as well as videos.

**CONTENT**

1. Abstract ………2
2. Introduction ………3
3. Literature Survey ………5
4. Existing System ……...12
5. Proposed System ………14
6. Problem Definition ………15
7. Objectives ………16
8. Methodology ………17
9. References ………20

**CHAPTER 1**

**PREAMBLE**

**1.1 GENERAL INTRODUCTION**

Rapid advancements in the fields of Science and Technology have led us to a stage where we are capable of achieving feats that see medium probable a few decades ago. Technologies in fields like Machine Learning and Artificial Intelligence have made our lives easier and provide solutions to several complex problems in various areas. Modern Computer Vision algorithms are approaching human-level performance in visual perception tasks. From image classification to video analytics, detections robust to the noise due to motion blur. This system can then be integrated with an image or video capturing device like a CCTV camera, to track safety violations, promote the use of face masks, and ensure a safe working environment. Amid the global crisis, new demand has emerged in the market, and that is of face mask detection. It is one such technology capable of detecting a face with a mask and verifying that person’s identity. It incorporates an AI-based pattern recognition system that uses biometric data of individuals. It extracts facial features and classifies them in different categories. Besides, it can also identify people without masks by generating an alarm or a notification to notify security or officials. They can see who has not covered faces with masks through software, mobile app, device, or a website.

The face recognition app or software uses biometrics to map the facial features from any image or video, by comparing it with a database of known faces. Moreover, the facial recognition market is also expected to grow at a value of USD 9.06 billion by 2024. Government initiatives and increasing demand for surveillance systems to enhance security will increase the facial recognition software’s adoption. So, this is about facial recognition without masks!

**1.2 PROBLEM STATEMENT**

Although the problem statement is similar (detecting face mask usage), the reality of images obtained from real-world sources like CCTV or surveillance cameras can be much harsher. We focused on [this particularcamera](http://camaras.vera.com.uy/camara/34) during most of the development, but we also tested it with other sources, like the streams shown at the beginning of the post.

Can you imagine the main issues that we’ll be facing? To enumerate some:

* **Size of the images**. Faces are drastically smaller, and much less clear.
* **Varying angles.** People are rarely looking straight to the camera — they look in every other possible angle.
* **Lack of clarity.** Often, it’s very difficult — or not possible at all — to tell if the person is wearing a mask or not from a single still frame.

**1.3 OBJECTIVES**

* Develop a novel object detection method that combines one-stage and two-stage detectors for accurately detecting the object in real-time from video streams with transfer learning at the back end.
* Improved affine transformation is developed to crop the facial areas from uncontrolled real-time images having differences in face size, orientation and background. This step helps in better localizing the person who is violating the facemask norms in public areas offices.
* Creation of unbiased facemask dataset with imbalance ratio equals to nearly one.
* The proposed model requires less memory, making it easily deployable for embedded devices used for surveillance purposes. To Detect whether a person is wearing mask or not. To make a efficient and less complex model that can work in low computational devices like CCTV cameras, mobile phones

**1.4 SCOPE**

Our work aims to a develop technique that can accurately detect mask over the face in public areas (such as airports. railway stations, crowded markets, bus stops, etc.) to curtail the spread of Coronavirus and thereby contributing to public healthcare. Further, it is not easy to detect faces with/without a mask in public as the dataset available for detecting masks on human faces is relatively small leading to the hard training of the model.

**1.5 LIMITATIONS**

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

**1.6 METHODOLOGY**

The major requirement for implementing this project using python programming language along with Deep learning, Computer vision and also with python libraries. The architecture consists of Mobile Net as the backbone, it can be used for high and low computation scenarios. We are using CNN Algorithm in our proposed system.

* We have four modules
  1. Datasets Collecting: We collect no of data sets with face mask and without masks. We can get high accuracy depends on collecting the number of images.
  2. Datasets Extracting: We can extract the features using mobile net v2 of mask and no mask sets
  3. Models Training: We will train the model using opencv, keras (python library).
  4. Facemask Detection: We can detect Preprocessing image and also detect via live video. If people wear mask, it will permit them, if not then it will give the identification to wear mask to prevent them from virus transmission.

**CHAPTER 2**

**LITERATURE SURVEY**

* **“Rapid object detection using a boosted cascade of simple feature”** **by** **P. Viola and M. Jones**

This work is distinguished by three key contributions. The first is the introduction of a new image representation called the "integral image" which allows the features used by our detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers. The third contribution is a method for combining increasingly more complex classifiers in a "cascade" which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. The cascade can be viewed as an object specific focus-of-attention mechanism which unlike previous approaches provides statistical guarantees that discarded regions are unlikely to contain the object of interest. In the domain of face detection the system yields detection rates comparable to the best previous systems. Used in real-time applications, the detector runs at 15 frames per second without resorting to image differencing or skin color detection.

Rich feature hierarchies for accurate object detection and

semantic segmentation

Rich feature hierarchies for accurate object detection and

semantic segmentation

* **“Rich feature hierarchies for accurate object detection and semantic segmentation” by R. Girshick, J. Donahue, T. Darrell, and J. Malik**

A large convolutional neural network trained for whole-image classification on ImageNet be coaxed into detecting objects in PASCAL. The framework combines powerful computer vision techniques for generating bottom-up region proposals with recent advances in learning high-capacity convolutional neural networks. We call the resulting system R-CNN: Regions with CNN features. The same framework is also competitive with state-of-the-art semantic segmentation methods, demonstrating its flexibility. Beyond these results, we execute a battery of experiments that provide insight into what the network learns to represent, revealing a rich hierarchy of discriminative and often semantically meaningful features.

* **“Focal loss for dense object detection” by P.Goyal, R.Girshick & P.Dollar**

The highest accuracy object detectors to date are based on a two-stage approach popularized by R-CNN, where a classifier is applied to a sparse set of candidate object locations. In contrast, one-stage detectors that are applied over a regular, dense sampling of possible object locations have the potential to be faster and simpler, but have trailed the accuracy of two-stage detectors thus far. We discover that the extreme foreground-background class imbalance encountered during training of dense detectors is the central cause. We propose to address this class imbalance by reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples. Our novel Focal Loss focuses training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training. To evaluate the effectiveness of our loss, we design and train a simple dense detector we call RetinaNet. Our results show that when trained with the focal loss, RetinaNet is able to match the speed of previous one-stage detectors while surpassing the accuracy of all existing state-of-the-art two-stage detectors.

* **“Rational use of face mask detection in Covid-19 Pandemic” by S.Feng, C.Shen, N,Xia.**

Since the outbreak of severe acute respiratory syndrome

coronavirus 2 (SARS-CoV-2), the virus that caused

coronavirus disease 2019 (COVID-19), the use of face

masks has become ubiquitous in China and other

Asian countries such as South Korea and Japan. Some

provinces and municipalities in China have enforced

compulsory face mask policies in public areas; however,

China’s national guideline has adopted a risk-based

approach in oﬀering recommendations for using face

masks among health-care workers and the general

public. We compared face mask use recommendations

by diﬀerent health authorities (panel). Despite the

consistency in the recommendation that symptomatic

individuals and those in health-care settings should use

face masks, discrepancies were observed in the general

public and community settings

Since the outbreak of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), the virus that caused coronavirus disease 2019 (COVID-19), the use of face masks has become ubiquitous in China and other Asian countries such as South Korea and Japan. Some provinces and municipalities in China have enforced compulsory face mask policies in public areas; however, China’s national guideline has adopted a risk-based approach in oﬀering recommendations for using face masks among health-care workers and the general public. We compared face mask use recommendations by diﬀerent health authorities (panel). Despite the consistency in the recommendation that symptomatic individuals and those in health-care settings should use face masks, discrepancies were observed in the general public and community settings

* **“Imagenet: A large-scale hierarchical image database” by J.Deng, W.Dong, & R.Socher.**

The explosion of image data on the Internet has the potential to foster more sophisticated and robust models and algorithms to index, retrieve, organize and interact with images and multimedia data. But exactly how such data can be harnessed and organized remains a critical problem. We introduce here a new database called ldquoImageNetrdquo, a large-scale ontology of images built upon the backbone of the WordNet structure. ImageNet aims to populate the majority of the 80,000 synsets of WordNet with an average of 500-1000 clean and full resolution images. This will result in tens of millions of annotated images organized by the semantic hierarchy of WordNet. This paper offers a detailed analysis of ImageNet in its current state: 12 subtrees with 5247 synsets and 3.2 million images in total. We show that ImageNet is much larger in scale and diversity and much more accurate than the current image datasets. Constructing such a large-scale database is a challenging task. We describe the data collection scheme with Amazon Mechanical Turk. Lastly, we illustrate the usefulness of ImageNet through three simple applications in object recognition, image classification and automatic object clustering. We hope that the scale, accuracy, diversity and hierarchical structure of ImageNet can offer unparalleled opportunities to researchers in the computer vision community and beyond

.

* **“Going deeper with convolutions” by C.Szegedy, W.Liu**

In our case, the word deep” is used in two different meanings: ﬁrst of all, in the sense that we introduce a new level of organization in the form of the “Inception module” and also in the more direct sense of increased network depth. In general, one can view the Inception model as a logical culmination of while taking inspiration and guidance from the theoretical work by Arora et al]. The beneﬁts of the architecture are experimentally veriﬁed on the ILSVRC 2014 classiﬁcation and detection challenges, on which it signiﬁcantly outperforms the current state of the art. we will focus on an efficient deep neural network architecture for computer vision, code named Inception, which derives its name from the Network in network paper by Lin et al in conjunction with the famous “we need to go deeper” internet

* **“Deep residual learning for image recognition” by K.He, S.Ren**

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers---8x deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers. The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

* **“Wider face: A face detection benchmark” by S.Yang, P.Luo**

Face detection is one of the most studied topics in the computer vision community. Much of the progresses have been made by the availability of face detection benchmark datasets. We show that there is a gap between current face detection performance and the real world requirements. To facilitate future face detection research, we introduce the WIDER FACE dataset, which is 10 times larger than existing datasets. The dataset contains rich annotations, including occlusions, poses, event categories, and face bounding boxes. Faces in the proposed dataset are extremely challenging due to large variations in scale, pose and occlusion, as shown in Fig. 1. Furthermore, we show that WIDER FACE dataset is an effective training source for face detection. We benchmark several representative detection systems, providing an overview of state-of-the-art performance and propose a solution to deal with large scale variation.

* **“Detecting masked faces in the wild with lle-cnns” by S.Ge, J.Li**

With the rapid development of machine learning methods, the problem of face detection seems to be well addressed yet. For example, the face detector proposed in [17] achieves an average precision of 98.0% on the public image benchmark AFW [37] by using the cascaded Convolutional Neural Networks, while the speed of some face detectors can reach up to 35 FPS [1] or even 400 FPS [21]. in some recent works such as [7, 23, 32], it is still necessary to construct large datasets and develop effective and efficient models for masked face detection. Toward this end, this paper presents a dataset for masked face detection, which is denoted as MAFA. The dataset consists of 30, 811 Internet images, in which 35, 806 masked human faces are manually annotated. In the annotation process, we ensure that each image contains at least one face occluded by various types of masks, while the six main attributes of each masked face, including locations of faces, eyes and masks, face orientation, occlusion degree and mask type, are manually annotated and cross-checked by nine subjects. The dataset will be released soon on the Internet, which we believe can facilitate the development of new face detectors in the future.

* **“Object detection with deep learning” by X.Wu**

Traditional object detection methods are built on handcrafted features and shallow trainable architectures. Their performance easily stagnates by constructing complex ensembles which combine multiple low-level image features with high-level context from object detectors and scene classifiers. With the rapid development in deep learning, more powerful tools, which are able to learn semantic, high-level, deeper features, are introduced to address the problems existing in traditional architectures. These models behave differently in network architecture, training strategy and optimization function, etc. In this paper, we provide a review on deep learning based object detection frameworks. Our review begins with a brief introduction on the history of deep learning and its representative tool, namely Convolutional Neural Network (CNN). Then we focus on typical generic object detection architectures along with some modifications and useful tricks to improve detection performance further. As distinct specific detection tasks exhibit different characteristics, we also briefly survey several specific tasks, including salient object detection, face detection and pedestrian detection. Experimental analyses are also provided to compare various methods and draw some meaningful conclusions. Finally, several promising directions and tasks are provided to serve as guidelines for future work in both object detection and relevant neural network based learning systems.

* **“Face detection techniques: a review” by A.Kumar & A.Kaur**

Face plays a major role in social intercourse for conveying identity and feelings of a person. Human beings have not tremendous ability to identify different faces than machines. So, automatic face detection system plays an important role in face recognition, facial expression recognition, head-pose estimation, human–computer interaction etc. Face detection is a computer technology that determines the location and size of a human face in a digital image. Face detection has been a standout amongst topics in the computer vision literature. This paper presents a comprehensive survey of various techniques explored for face detection in digital images. Different challenges and applications of face detection are also presented in this paper. At the end, different standard databases for face detection are also given with their features. Furthermore, we organize special discussions on the practical aspects towards the development of a robust face detection system and conclude this paper with several promising directions for future research.

* **“Rapid object detection using a boosted cascade of simple features” by P.Voila & M.Jones**

This work is distinguished by three key contributions. The first is the introduction of a new image representation called the “Integral Image” which allows the features used by our detectorto be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features from a larger set and yields extremely efficient classifiers. The third contribution is a method for combining increasingly more complex classifiersin a “cascade” which allows background regions of the image to be quickly discarded while spending more computation on promising object-like regions. The cascade can be viewed as an object specific focus-of-attention mechanism which unlike previous approaches provides statistical guarantees that discarded regions are unlikely to contain the object of interest. In the domain of face detection the system yields detection rates comparable to the best previous systems. Used in real-time applications, the detector runs at 15 frames per second without resorting to image differencing or skin color detection

**CHAPTER 3**

**SYSTEM ANALYSIS AND DESIGN**

**3.1 EXISTING SYSTEM**

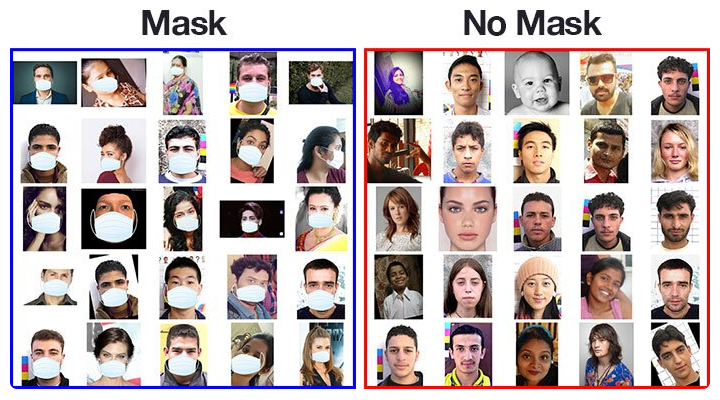
So far, many countries have used technology-based solutions to overcome the pandemic loss. Several developed countries are employing GPS technology to monitor the movements of the infected and suspected individuals, provides a survey of different emerging technologies, including Wi-fi, Bluetooth, smartphones, and GPS, positioning (localization), computer vision, and deep learning that can play a crucial role in several practical Face mask detection scenarios. Some researchers utilize drones and other surveillance cameras to detect crowd gatherings and to get assured people are wearing mask or not.

**3.2 PROPOSED SYSTEM**

The pre-trained face detection model seems to work great for this case, and it detects faces even when they are partially covered by masks. So, no need to re-train anything for the first model (object detector, SSD).

As there was no pre-trained classifier to distinguish faces with and without masks, Adrian trained this model with a dataset provided by one of his readers (to download it, refer to the [**post**](https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/)). This reader*-*of*-*your*-*dreams collaborator created this dataset with nearly ~1400 images in a very clever and effective way: he took a dataset with regular faces, and artificially added masks to some of the images. He leveraged other computer vision techniques to automatically place the masks over the faces.

Some of the images in the dataset are shown below:



Even though this is a synthetic dataset and it’s built with a single mask type, it seems to generalize pretty good for other kind of masks . It does a great job for this use case, where the user —or users, since it supports multiple faces— is placed in front of the camera.

**Detecting people and faces**

To have a better idea of how the components will fit, let’s briefly describe how tracking works:

1. The tracking algorithm lives during the whole video sequence, and is updated step by step, with each new frame that is processed.
2. At each step, it receives a list of points, and tries to match each point with previously detected points. For that, it can estimate the new position that each point should have, according to its estimated speed (e.g: using a Kalman Filter, details later).
3. When it detects that certain points (e.g: corresponding to a person’s eyes) seem to have “survived” for a couple steps, you can use it to say: this is a single person and it’s moving from here to there.

### Robustly finding people in frames

* With the brief explanation above, it’s clear that this tracking algorithm only works if we can **consistently detect positions during the entire sequence**. In other words, if some points are not detected for many frames and then they show up in another place, this tracker won’t be able to match them as belonging to the same person.
* So there’s an unavoidable problem with tracking just faces: **what if the person looks in another direction and the face is not visible for many frames?**
* We could attempt to detect heads in any position instead of just faces, to then classify them in mask/no mask/back side. After discussing these ideas with our R&D team, who had been working on some tracking solutions for different video analytics applications — from estimating people flow for the retail industry, to vehicle tracking in traffic — the recommendation was to first try with [pose estimation](https://en.wikipedia.org/wiki/Articulated_body_pose_estimation). Apart from being a well-studied problem, there are several pre-trained working implementations and datasets.
* There are some trade-offs, though. In general, these models tend to be slower than object detectors. But we think there are many reasons that justify using it anyway. Using the person’s key points, we can consistently track them across frames, despite changes in orientation and position.

Let’s see how pose estimation looks like in our camera.



**CHAPTER 4**

**SYSTEM REQUIREMENT**

**4.1 HARDWARE REQUIREMENT**

* Processor: AMD ryzen 3300u
* Hard disk storage: SSD 256GB
* Memory: 8GB DDR4
* Monitor: 15’ LED
* Input devices: Keyboard and Mouse
* Internet connection: 1Mbps

**4.2 SOFTWARE REQUIREMENT**

* Operating system: Windows 10
* Programming language: Python Programming Language version 3.7
* Graphics: Radeon 2gb Graphics card and DirectX version 11
* IDE and SHELL: jupyter notebook, windows powershell 5.1
* Libraries: Tensorflow/keras, opencv, numpy, os, sklearn, imutils
* ModelsandOptimizer**:** mobilenetv2 and Adam optimizer

**CHAPTER 5**

**IMPLEMENTATION**

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.

1. **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
2. **Deployment:** Once the face mask detector is trained, we can then move on to loading the mask detector, performing face detection, and then classifying each face as with mask or without mask

**We have four modules**

* **Datasets Collecting**: We collect no of data sets with face mask and without masks. we can get high accuracy depends on collecting the number of images .
* **Datasets Extracting**: We can extract the features using mobile net v2 of mask and no mask sets
* **Models Training**: We will train the model using open cv, keras (python library).
* **Facemask Detection**: We can detect Preprocessing image and also detect via live video . If people wear mask, it will permit them, if not then it will give the buzzer to wear mask to prevent them from virus transmission.

**HOW DOES OUR MODEL WORK?**

So here are steps representing working of our model, this is actual practical part we have implemented. This steps and sub-steps shows the journey from how our own custom model was created using mobilenetv2 and it is divided into two parts i.e Training part and deployment part

* **TRAINING**

1. **LOAD DATA SET**

If the Dataset already contains rows; the incoming data from the data source is merged with the existing rows. The Load method can be used in several common scenarios, all centered around getting data from a specified data source and adding it to the current data container (in this case, a Dataset ).

1. **TRAIN MODEL USING TENSERFLOW**

A machinelearningmodel is a function with learnable parameters that maps an input to a desired output. The optimal parameters are obtained by training the model on data. Training involves several steps: Getting a batch of data to the model. It allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for: Classification**,** Perception**,** Understanding**,** Discovering**,** PredictionandCreation

1. **SERIALIZE MODEL TO DISK**

The most common method is to serializethemodelusingsomeparticularformataftertraining, and desterilize that model in the production environment. In Python, there are several language-specific serialization formats based on pickle. Alternatives include more agnostic exported formats

* **DEPLOYING OUR MODEL WITH IMAGES AND VIDEOS**

1. **LOAD MODEL FROM DISK**

When saving the model and its layers, the Saved Model format stores the class name, **call function**, losses, and weights (and the config, if implemented).

A Keras model consists of multiple components:

* + The architecture, or configuration, which specifies what layers the model contain, and how they're connected.
* A set of weights values (the "state of the model").
* An optimizer (defined by compiling the model).
* A set of losses and metrics (defined by compiling the model or calling add\_loss() or add\_metric())

1. **DETECT FACES IN IMAGES/VIDEO STREAMS**

In most cases, the face detection system will confirm detection by overlaying a rectangle on each face in the scene displayed on the camera's LCD. Some systems use a different color to identify the face that will be used as a focusing target. Tracking facilities are also provided by some recent systems. The algorithm might then attempt to detect eyebrows, the mouth, nose, nostrils and the iris. ... The methods used in face detection can be knowledge-based, feature-based, template matching or appearance-based

.

1. **EXTRACT ROI OF EACH FACE**

Extraction of ROI (Region-Of-Interest) in dermatosis images can be used in content-based image retrieval (CBIR). Image segmentation takes an important part in it.

Using OpenCV for efficiently extracting ROI from images

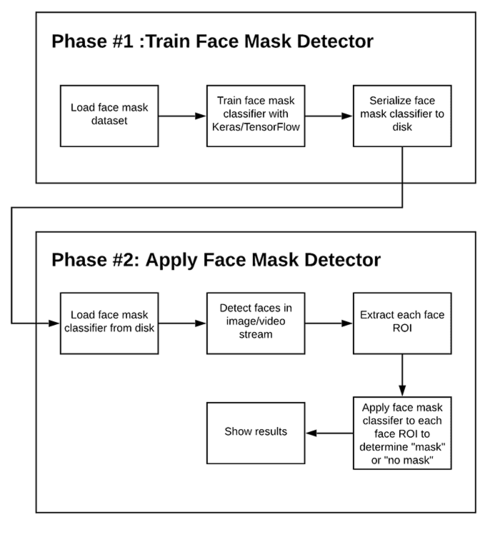
1. Convert the RGB image to gray-scale using “cvtColor()”
2. Remove noise from the gray-scale image by applying a blurring function “GaussianBlur()”
3. Finally applying the “Canny()” function to the blurred image to obtain the edges.
4. **APPLY OUR MODEL TO EACH FACE ROI**

Here in each image or video stream the model should be applied and the region of interest is facial landmark allow us to automatically to infer the location of facial structures, including

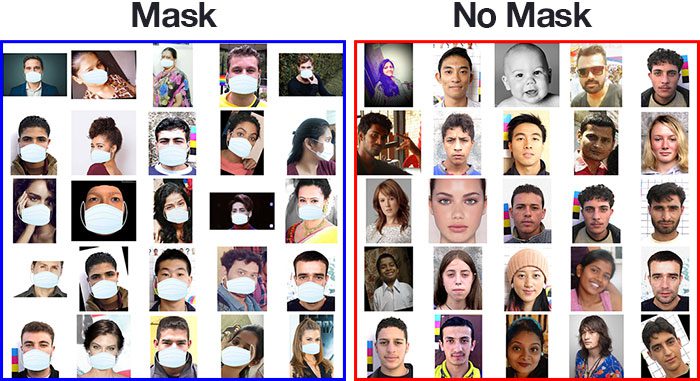
* Eye
* Eyebrows
* Nose
* Mouth
* Jawline

1. **CHECK IF PERSON WEARING MASK OR NOT AND DISPLAY RESULT**

* In the last step, we use the OpenCV library to run an infinite loop to use our web camera in which we detect the face using the Classifier. The code webcam = cv2.VideoCapture(0) denotes the usage of webcam.
* The model will predict the possibility of each of the two classes ([without\_mask, with\_mask]). Based on which probability is higher, the label will be chosen and displayed around our faces.
* **TWO-PHASE COVID-19 FACE MASK DETECTOR**



### OUR COVID-19 FACE MASK DETECTION DATASET



**Figure:** A face mask detection dataset consists of “with mask” and “without mask” images.

This dataset consists of **1,376 images** belonging to two classes:

* with mask: 690 images
* without mask: 686 images

**Our goal is to train a custom deep learning model to detect whether a person is or is not wearing a mask.**

**How was our face mask dataset created?**

* Best case scenario
* Worst case scenario

**STEPS TO CREATE DATASET**

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



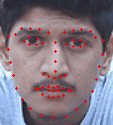
1. From there, we apply face detection to compute the bounding box location of the face in the image:



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



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



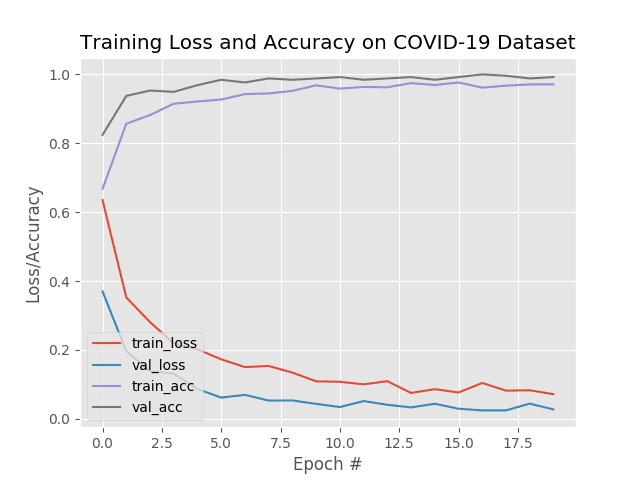
1. Next, we need an image of a mask (with a transparent background) such as the one below:



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



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



**Implementing our COVID-19 face mask detector for images with OpenCV**

Now that our face mask detector is trained, let’s learn how we can:

1. Load an input image from disk
2. Detect faces in the image
3. Apply our face mask detector to classify the face as either with mask or without mask

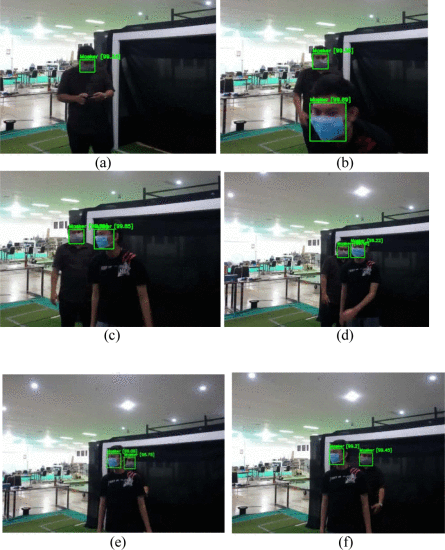
**CHAPTER 6**

**INTERPRETATION OF RESULTS**

This section will present the experiment results of the face mask detection in real-time application and that has already been installed. The first experiment which is depicted on Fig. 4 has been done as the first trial before it is implemented for the moving person. Fig. 4(a) illustrates the face detector that detected the single user wearing a face mask accurately even it has some disturbance in the area. As for Fig. 4(b), the user was added slowly from below the camera and the detector was able to detect the mask properly. When the users are standing close to each other as seen on Fig. 4 (c)-(f), this system was also able to detect the face mask even if the user was surrounded by many objects with similar color.

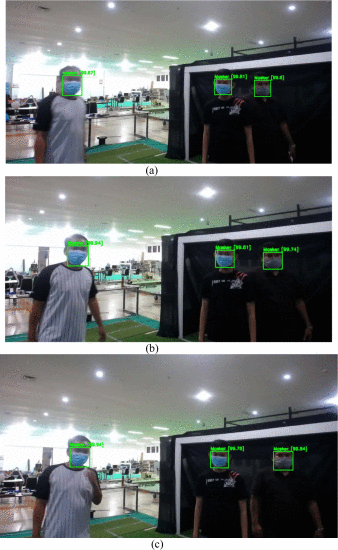
After did the trial with no error, we are ready to verify this device with more user. As seen on Fig. 5(a)-(c), we added the user into three people with different types of face mask such as surgical and fabric face mask. Each person was standing in different position to verify the performance of face mask detection. From the picture it is verify that the face detected remains steady in detecting face mask of the users even the lighting was in different brightness. To make different in brightness, we turned off and on the lamp at our lab as to test the feature of this system.

On the other hand, the experiment of detecting a non-wearing mask is presented on Fig. 6(a)-(c). On Fig. 6(a) the first user who wears a white T-shirt attempted to pull off his mask, and the mask detector was able to distinguish the non-wearing mask and mask-wearing user. Fig. 6(b) also presented the mask detection precisely. Moreover, on Fig. 6(c) the first and third users tried to take off their mask, and the mask detector detected the face mask condition steadily.



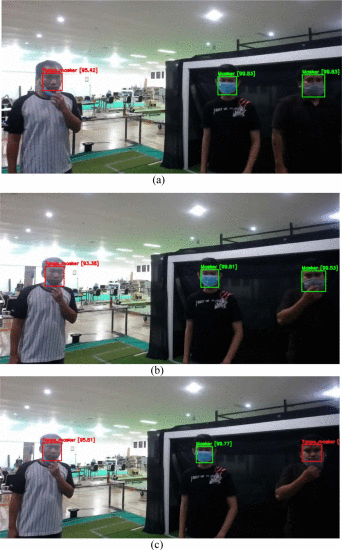
**Fig. 4.**

The face mask detector detected the user who wear the face mask (a) alone in the frame, (b) detected the new user, (c)-(f) detected the face mask where the user was close to each other.



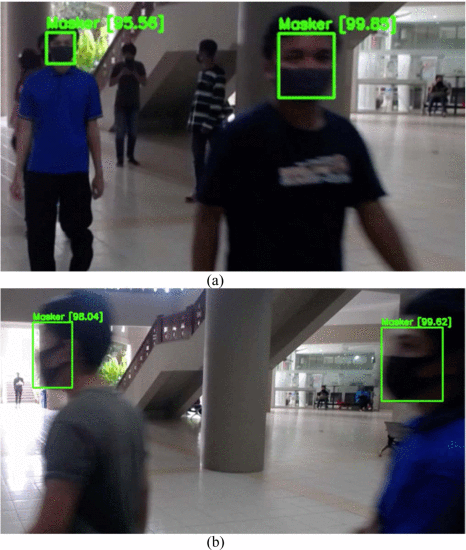
**Fig. 5.**

The detector detected multiple people who are wearing a face mask with different position from each other.



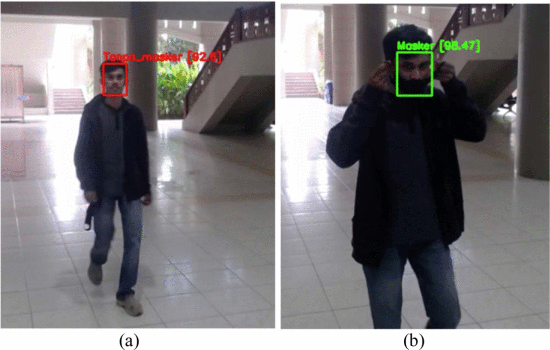
**Fig. 6.**

The face mask detection detected the non-wearing user with different angle of pose.



**Fig. 7.**

The face detection detected the fabric face mask while (a) the user moving towards to the device detector, (b) the user moved away from the device.



**Fig. 8.**

The face-mask detected the (a) a non-wearing face mask user, (b) a wearer of face mask

When the user attempted not to wear the face mask properly, represented on Fig. 8, this device will show that the user needs to wear the mask . Fig. 8 (a) illustrates the non-wearing user of face mask and Fig. 8 (b) for wearers of face mask. From all these experiments, the face detector which is built by algorithm is able to detect and distinguish a non-wearing and a wearing-mask user properly in every different situation such as lighting, mess up area, and clean area.

CONCLUSION AND FUTURE SCOPE

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

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

BIBLOGRAPHY

* 1. World Health Organization et al. Coronavirus disease 2019 (covid-19): situation report, 96. 2020. - Google Search. (n.d.).
  2. Social distancing, surveillance, and stronger health systems as keys to controlling COVID-19 Pandemic, PAHO Director says - PAHO/WHO | Pan American Health Organization. (n.d.).
  3. Garcia Godoy L.R. Facial protection for healthcare workers during pandemics: a scoping review, BMJ. Glob. Heal. 2020;5(5) doi: 10.1136/bmjgh-2020-002553.
  4. Eikenberry S.E. To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic.
  5. Wearing surgical masks in public could help slow COVID-19 pandemic’s advance: Masks may limit the spread diseases including influenza, rhinoviruses and coronaviruse
  6. Inamdar M., Mehendale N. Real-Time Face Mask Identification Using Facemasknet Deep Learning Network
  7. R. Yamashita, M. Nishio, R. Do and K. Togashi, "Convolutional neural networks: an overview and application in radiolog
  8. C. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall, "Activation Functions: Comparison Of Trends In Practice And Research For Deep Learning