Media Engineering and Technology Faculty
German University in Cairo

Detection of Incorrectly Worn Face Masks

Bachelor Thesis

Author:	Donia Ali Mohamed
Supervisors:	Assoc. Prof. Seif Eldawlatly
Submission Date:	1 August, 2021

Media Engineering and Technology Faculty
German University in Cairo

Detection of Incorrectly Worn Face Masks

Bachelor Thesis

Author:	Donia Ali Mohamed
Supervisors:	Assoc. Prof. Seif Eldawlatly
Submission Date:	1 August, 2021

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowlegement has been made in the text to all other material used

Donia Ali Mohamed
1 August, 2021

Acknowledgments

I sincerely would like to thank my supervisor, Assoc. Prof. Seif Eldawlatly for his patience, continuous guidance, and help; I would not achieve this without his guidance. Thanks to my family and friends for their continuous support.

Abstract

COVID-19 was discovered in December 2019 in Wuhan, China. It is the coronavirus that causes severe acute respiratory syndrome (SARS) (SARS-CoV-2). The illness emerged as a result of the rapid development of unexplained pneumonia cases. The World Health Organization (WHO) declared the Public Health Emergency of International Concern epidemic on January 30, 2020. The globe surpassed 100,000 cases of the virus on March 7, 2020. By March 11, 2020, the epidemic had progressed from an epidemic to a pandemic. The World Health Organization recommended wearing face masks to limit the spread of COVID-19. As a result, various governments have adopted wearing a face mask in public areas as mandatory. Furthermore, public service providers do not allow customers to utilize their services unless they wear a face mask. What is happening in the world draws researchers' attention to exploring computer vision techniques and machine learning to develop mask detection techniques to help in monitoring public behavior and in applying the COVID-19 pandemic containment. In this thesis, we tried different image processing and machine learning techniques for mask detection and classification. We first tried different existing algorithms for face detection and evaluated their accuracy in detecting faces, especially faces with masks. Then we tried different approaches to detect and classify face masks, starting from image processing techniques to machine learning models. After that, we evaluated these approaches. Finally, we construct a complete pipeline to detect mask presence, then recognize whether the face masks are worn correctly or not. We evaluated the accuracy of both Multi-Task Cascaded Convolutional Neural Network and Haar cascade classifiers on detecting faces. We found that Multi-Task Cascaded Convolutional Neural Network has an average accuracy of 97.6% in detecting masked faces. In contrast, the Haar cascade classifier has an average accuracy of 84.8% in detecting masked faces. So, Multi-Task Cascaded Convolutional Neural Network is more precise in detecting faces, especially the masked ones. For mask detection, we tried the edge detection approach and segmentation with machine learning approaches. We found that the segmentation approach is more accurate where it has an accuracy of 93% to differentiate between faces with masks and faces without masks. In contrast, the edge detection method has an accuracy of 66% in the non-masked faces and an average accuracy of 59.2% in the masked faces. We tried the nose and mouth detection approach for mask detection and classification with the Support Vector Machine classifier and machine learning technique as Support Vector Machine. We found that the nose and mouth detection approach has an accuracy of 92% for detecting correctly worn face masks. And accuracy of 91%, 87%, 81% and 80% for detecting Incorrectly worn face masks where

the chin is exposed, the nose is exposed, nose and mouth are exposed, and no face masks respectively. We had two Support Vector Machine models for the used machine learning approach: A 3-class Support Vector Machine model and A pipeline of 2-class Support Vector Machine models. The purpose of the 3-class Support Vector Machine model is to classify faces with correct worn face masks, faces with incorrectly worn face masks, and faces without face masks. We found that this model has an accuracy of 97%. The pipeline of 2-class Support Vector Machine models is constructed from a Support Vector Machine model to detect face masks and another Support Vector Machine model to classify face masks if detected. The first model has an accuracy of 98% to differentiate between faces with mask and faces without mask. The second model has an accuracy of 98% to differentiate between correctly worn face masks and Incorrectly worn face masks.

Contents

Chapter 1

Introduction

1.1 Motivation

In December 2019, Wuhan, China, saw the outbreak of a novel and very infectious primary atypical (viral) pneumomnia. The new illness, Coronavirus disease 2019 (COVID-19), was eventually shown to be caused by a previously undiscovered zoonotic coronavirus known as SARS-CoV-2. To help limit the spread of this new coronavirus, the World Health Organization (WHO), medical experts, and governments around the world now recommend that people wear face masks if they have respiratory symptoms, are caring for people who have symptoms, or otherwise interact with large groups of people.

1.2 Objective

After these developments, researchers responded to that by giving attention to developing a face mask detection system using computer vision and artificial intelligence. So, our objective in this thesis is to recognize if face masks are not used properly using image processing and machine learning methods.

1.3 Thesis contribution

In this thesis, we started by exploring face detection techniques. We used the Haar cascade classifier and Multi-task Cascaded Neural Network for detecting faces either with a mask or without. Our results reveal that Multi-Task Cascaded Convolutional Neural Network (MTCNN) is more accurate than Haar Cascade Classifiers (Haar) in recognizing faces, particularly those with face masks worn correctly or wrongly. Compared to Haar, which has an average accuracy of 84.8% in detecting masked faces, MTCNN has an average accuracy of 97.6% in recognizing masked faces.

Then we explored different image processing and machine learning approaches to detect and classify face masks, such as edge detection methods, segmentation, cascade classifiers for nose and mouth detection, and Support Vector Machine classifier.

We tried edge detection methods and segmentation with machine learning using the Support vector machine (SVM) classifier approach for face mask detection. The edge detection method has a 66% accuracy in non-masked faces and a 59.2% average accuracy in masked faces. While the segmentation approach has a 93% accuracy in distinguishing between faces with masks and faces without masks.

We tested nose and mouth detection along with the SVM classifier and machine learning approach as SVM classifier as face mask detection and classification approaches. The accuracy for detecting correctly worn face masks in the nose and mouth detection approach was 92%. Furthermore, for incorrectly worn face masks with the chin exposed, the accuracy was 91%. Furthermore, for incorrectly worn face masks with the nose exposed, the accuracy was 87%. Finally, the accuracy for incorrectly worn face masks where the nose and mouth are exposed was 81%, while the accuracy for no face masks was 80%. The accuracy of the SVM model in differentiating between incorrectly worn face mask images (nose and mouth covered) and correctly worn face mask images (nose, mouth, and chin are covered) was 96%. We had two models for the machine learning technique employing the SVM classifier: a 3-class SVM model and a pipeline of 2-class SVM models. The 3-class model had a 97% accuracy in classifying faces with correct worn face masks, faces with incorrectly worn face masks, and faces without face masks. The pipeline of 2-class SVM models includes an SVM model to detect face masks and another SVM model to classify face masks if they are detected. The initial model differentiates between faces with and without masks with a 98% accuracy. The second model distinguishes correctly worn face masks from incorrectly worn face masks with a 98% accuracy.

And among different approaches for mask detection and classification, we can find that a trained Support Vector Machine model is more accurate in solving our problem. Also, using SVM along with the nose and mouth detection approach has high premises in real-time.

1.4 Thesis outline

The thesis is organized as follows. In chapter (2), we review details about COVID-19 and the theoretical explanations of the used techniques. Chapter (3) describes the used datasets, different proposed approaches, and the used evaluation metrics. In chapter (4), we display the results of the proposed techniques. Finally, the thesis is concluded and summarized in chapter (5).

Chapter 2

Background

2.1 COVID-19

COVID-19 [1] is a contagious disease that belongs to the severe acute respiratory syndrome (SARS) family. COVID-19 was first observed in Wuhan, China, in December 2019. It is known as the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The sudden appearance of unexplained pneumonia cases in Wuhan led to the emergence of COVID-19. The Wuhan Huanan Seafood Wholesale Market was suspected to be the cause of the COVID-19 epidemic in China. On 1 January 2020, the market was closed for the sanitation and disinfection of the environment. Several important zoonotic viruses have been found in bats as hosts. These include coronaviruses linked to human SARS outbreaks in 2002 and the Middle East respiratory disease (MERS) outbreak in 2013. So, bats were suspected of being the source of COVID-19. On 30 January 2020, the World Health Organization (WHO) called the outbreak a Public Health Emergency of International Concern. On 7 March 2020, the world reached 100,000 COVID-19 cases. On 11 March 2020, COVID-19's status was changed from an epidemic to a pandemic. COVID-19 was found to spread easily among people and continuously more cases have been discovered over time in different countries. When a person who is infected coughs, sneezes, talks or breathes, the COVID-19 spreads into minute liquid particles from their mouth or nose. According to current data, the virus transmits mostly amongst people who are near one another, within one metre (short-range). When virus-containing aerosols or droplets are inhaled or come into direct contact with the eyes, nose, or mouth, a person might get infected. The virus can also spread in poorly vented and/or congested interior environments, where people prefer to spend more time. Touching an object or a surface with the virus on it, then touching one's mouth, nose or eye can lead to infection with COVID-19. COVID-19 symptoms may appear two to 14 days after exposure.The most common symptoms are fever, fatigue, and dry cough. It was found that losing taste and smell, sore throat, headache, aches and pains, a rash on the skin, or discolouration of fingers or toes, diarrhoea, and conjunctivitis are less common symptoms. The WHO organization recommended some safety precautions, such as maintaining social distance by keeping 1 meter apart between people, wearing face masks and making them a part of normal life, washing hands regularly and thoroughly, either using soap or rubbing them using an alcohol-based hand rub, coughing into a bent elbow or tissue, avoiding touching one's eyes, nose, and mouth, cleaning and disinfecting surfaces continuously, especially those which are regularly touched, avoiding crowded places, etc. Different organizations started to find vaccines for COVID-19 in early December 2020. On December 31, 2020, the Pfizer/BioNtech Comirnaty vaccine was listed on the WHO Emergency Use Listing (EUL). On February 16, 2021. the WHO Emergency Use Listing was given for the SI-I/Covishield and AstraZeneca/AZD1222 vaccines (designed by AstraZeneca/Oxford and produced by the State Institute of India and SK Bio, respectively). Johnson & Johnson's Janssen/Ad26.COV 2.S was approved for WHO Emergency Use Listing on March 12, 2021. The Moderna COVID-19 vaccine (mRNA 1273) was approved for WHO Emergency Use Listing on April 30, 2021. Beijing Bio-Institute of Biological Products Co. Ltd., a subsidiary of China National Biotec Group, manufactures the Sinopharm vaccine (CNBG). On June 1, 2021, the Sinovac-CoronaVac was offered for the WHO Emergency Use Listing. According to the World Health Organization, there are about 190,671,330 confirmed COVID-19 cases and about 4,098,758 deaths around the world. And a total of 3,436,534,998 vaccine doses have been administered.

2.1.1 Face masks

With the rapid spread of COVID-19, Wearing a face mask was one of the proposed solutions to limit the infection. There are different types of face masks such as the medical masks, respirator masks and the fabric masks. Different countries has made wearing a face mask mandatory in public places. Also public service provider don't allow people to use their services without wearing a face mask. Face masks should be worn in a correct way covering nose, mouth,and chin as shown in Fig(2.2).

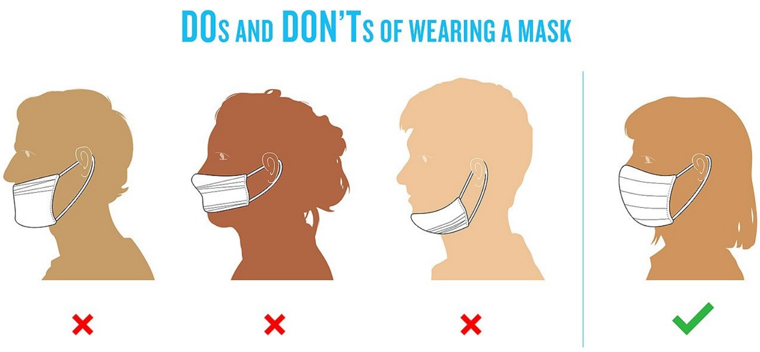


Figure 2.1: How to correctly wear a face mask (Adapted from [2])

2.2 Face and Facial Landmarks Detection

Face detection is a type of computer technology. This technology's goal is to recognize faces in digital photographs. Face detection has numerous applications in various fields such as security, law enforcement, personal safety, biometrics, etc [3]. It has progressed from simple computer vision techniques to more complex machine learning approaches to more sophisticated artificial neural network algorithms. As a result, computer scientists are very interested in detecting faces and facial landmarks such as the mouth, nose, and eyes [4].

2.2.1 Cascade classifiers

Haar Cascade classifiers

In 2001, Paul Viola and Michael Jones proposed an effective object detection method using Haar feature-based cascade classifiers in their paper[5]. They used a machine learning approach by training a cascade function on a large number of positive and negative images. For face detection, the model is trained on a large number of images of faces (positive images) and images without faces (negative images). The next step is extracting Haar features that are shown in Fig (2.2). Each feature is a value computed by subtracting the summation of the bright pixel under the white rectangle from the summation of the dark pixels under the black rectangle, and by doing this for every feature, it needs too much computation. To solve this, they used an integral image approach to decrease computation time. The selection of the best features out of 160000+ features is achieved by the Adaboost approach to select a subset of features to make a strong classifier. In the last step comes the use of cascade classifiers, where taking a 24x24 window and applying all 6000 features on it will be time-consuming. So, by applying one feature at a time, if it succeeds, the next one is applied, and so on. If the window passes all the stages, then it is a face region. But if it fails at some point, the window is then rejected. Also Haar cascade classifier can be used in facial landmarks detection like nose, mouth, and eyes. The architecture of Haar cascade calssifier is shown in Fig (2.3).

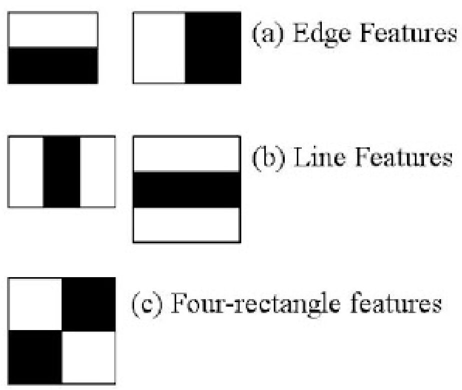


Figure 2.2: Haar-features source

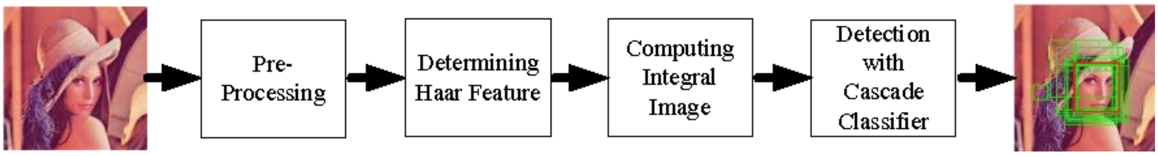


Figure 2.3: Haar feature-based cascade architecture (Adapted from [6])

2.2.2 Multi-Task Cascaded Convolutional Neural Network

Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning neural network [7]. This class of neural networks is often used for classification and computer vision tasks. Before CNNs feature extraction methods, that are used in object detection, was manual and time-consuming. Convolutional neural networks, on the other hand, use concepts from linear algebra, notably matrix multiplication, to discover patterns inside an image, making them more scalable for image classification and object recognition applications. CNNs have three main layers which are the Convolutional layer, Pooling layer, and Fully-connected (FC) layer in order. The CNN becomes more complicated with each layer, recognizing larger areas of the image [8]. The Convolutional layer is the core building block in CNNs as it encounters the majority of computations. when an input image is fed to the first layer (convolutional layer), The feature detector known as a kernel or a filter will move across the receptive fields of the image in order to check the presence of the features. The Convolution Operation's goal is to extract high-level features from the input picture, such as edges. According to the type of padding, whether it is valid padding or same padding, the produced result is either reduced in dimensionality as compared to the input or increased/ same in size with the input. Finally, the convolutional layer transforms the image to a numerical value, which the neural network can analyse and extract significant patterns from. After that comes the Pooling layer which is also known as the down sampling layer. The Pooling layer is responsible for decreasing the Convolved Feature's spatial size which decreases computational power to process the data, improve efficiency, and limit the risk of overfitting. There are two main types of pooling which are max pooling, and average pooling. After the first two-layer, the output is going to be flattened. Finally comes the Fully Connected Layer (FC Layer) which is responsible for the classification task using the Softmax Classification technique. The convolutional and pooling layers use ReLu functions. There are various architectures of CNNs available like LeNet, AlexNet, VGGNet, ResNet, GoogLeNet, ZFNet, and LeNet-5 which is known as the classic CNN architecture. Fig (2.4) illustrates a sample CNN architecture.

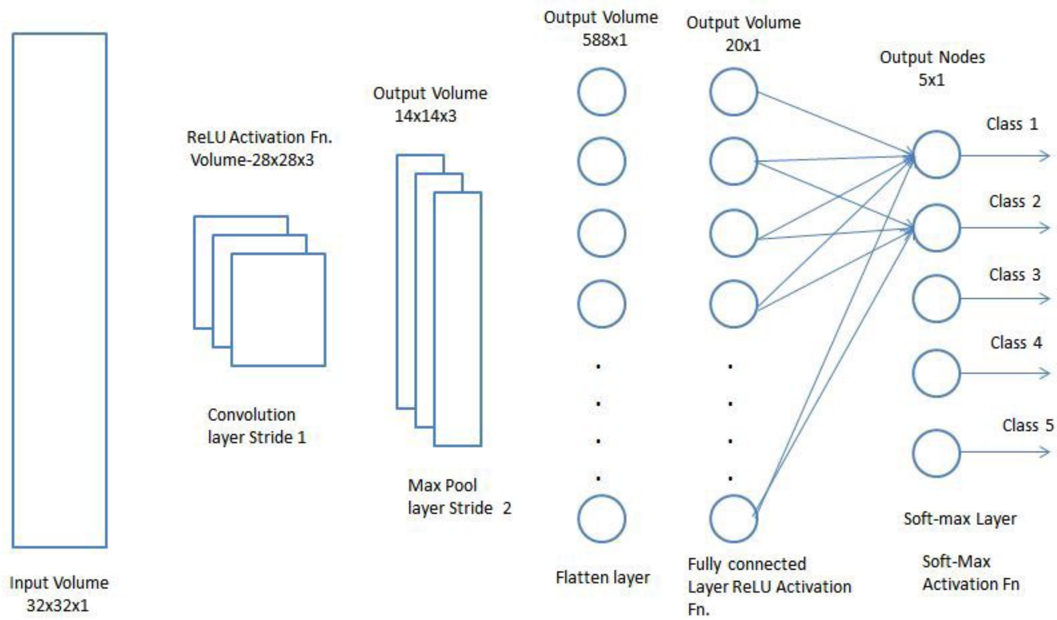
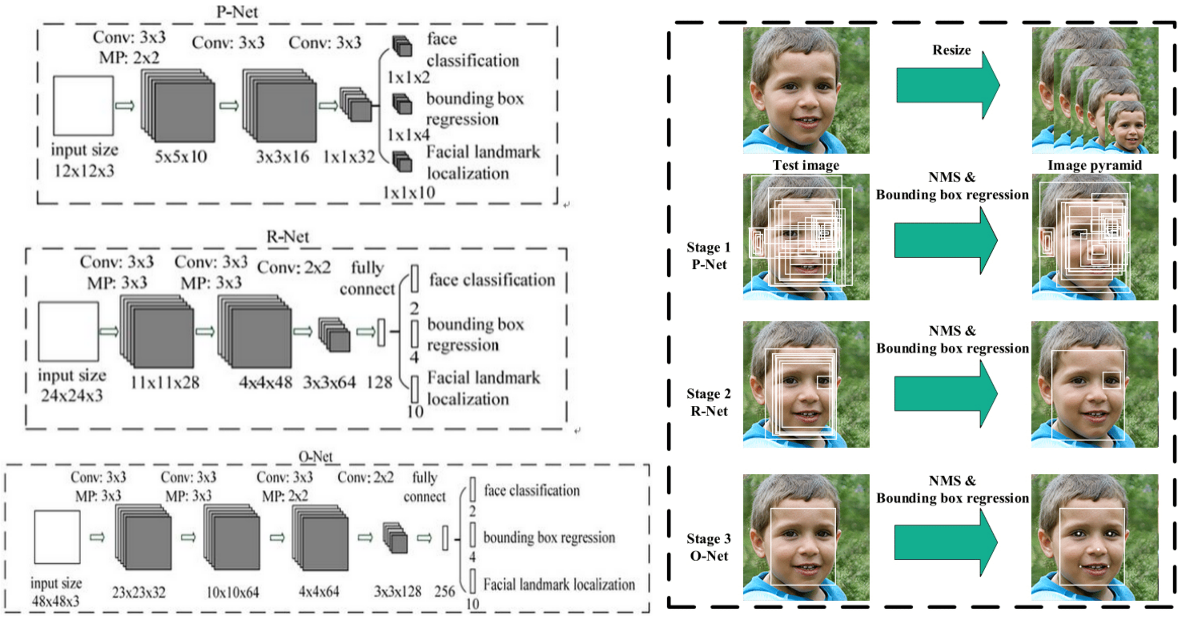


Figure 2.4: The architecture of a basic CNN (Adapted from [9])

Multi-Task Cascaded Convolutional Neural Network

Multi-Task Cascaded Convolutional Neural Network (MTCNN) is a modern approach for detecting faces proposed by Kaipeng Zhang, et al. in the 2016 paper [10]. It is a 3-Stage neural network detector with a cascade structure. The 3 neural networks are Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net). First, the input image is rescaled to different sizes called an image pyramid which will be the input to the neural networks. Then the Proposal Network works on the image pyramid and proposing candidate facial regions and their bounding boxes. After that, A non-maximum suppression (NMS) is employed to merge highly overlapped candidates proposed by P-Net. The candidates are then fed to the second network which is R-Net. The purpose of the Refine network is to filter candidate bounding boxes by rejecting a large number of false candidates with the help of NMS. Finally, comes the purpose of the O-network (Output) which is proposing the face bounding box and the five facial landmarks. So, MTCNN is called a multi-task network as each of the three models in the cascade (P-Net, R-Net, and O-Net) are trained on three tasks, and make three types of predictions which are face classification, bounding box regression, and facial landmark localization. Fig (2.5) illustrates the MTCNN architecture and how it works.



(a) The architecture of P-Net, R-Net, and O-Net (b) Visualisation of how MTCNN works

Figure 2.5: Multi Task Cascaded Convolutional Neural Network (Adapted from [11])

2.3 Image processing

Image processing is the method of processing digital images using a digital computer through algorithms. It is a subfield of signal processing where the input is a digital image and the output is either an image or features of the image. Image processing is widely used in many fields, such as the medical field, transmission, and encoding, video processing, machine/Robot vision, etc.

2.3.1 Canny edge detector

In 1986, John F. Canny [12] developed the popular Canny Edge Detection algorithm. Canny edge detector is a multi-step process. The grayscale of the input image goes through the noise reduction stage, gradient calculation stage, non-maximum suppression stage, and hysteresis thresholding stage. In the first stage, a 5×5 Gaussian filter is applied to smooth the image and get rid of the noise. Smoothing the image will ignore the unwanted details and focus on the image structure. The equation for a Gaussian filter kernel of size (2k+1)×(2k+1) is as follows:

$$k(x, y) = \frac{1}{2\Pi\sigma^2} exp\left(- \frac{x^2 + y^2}{2\sigma^2} \right) \tag{2.1}$$

Where x and y represents respective distances to the horizontal and vertical center of the kernel and σ represents standard deviation of the kernel.

In the second stage, gradient orientation and magnitude are computed. Applying a Sobel filter in both horizontal and vertical directions to get the first derivative in the horizontal direction (G_x) and the vertical direction (G_x). So, the edge gradient and direction for each pixel are computed as follows:

$$Edge\_Gradient(G) = \sqrt{G_x + G_y} \text{ and } f(x) = tan^{-1}(\frac{G_x}{G_y}) \tag{2.2}$$

Then comes the computation of edge gradient and direction for each pixel. In the third stage, the goal is to thin the edges using a non-maximum suppression process. By examining the gradient magnitude G and orientation θ at each pixel in the image. Then comparing every pixel to the 3×3 neighborhood surrounding it and determining in which direction the orientation is pointing. So, if the pixel is pointing towards the north or south, then the north and south magnitudes are examined. But if the orientation is pointing towards the east or west, then the east and west pixels are examined. If the magnitude of the center pixel is larger than both the pixels it is being compared to, then the magnitude should be preserved; otherwise, it should be discarded. Finally, comes the hysteresis thresholding stage. The goal of this stage is to remove regions that are not technically edges. Two threshold values are needed for this stage, which are minVal and maxVal. If the gradient intensity of an edge is greater than maxVal, then this edge is a certain edge. If the gradient intensity of an edge is less than minVal, then this edge is definitely not an edge, so this region is discarded. But if the gradient intensity of an edge lies in a range between minVal and maxVl, the classification will be based on the connectivity. If the pixel is connected to "sure-edge" pixels, then it is considered as a part of the edge. Otherwise, the pixel is discarded.

2.3.2 Segmentation

Image segmentation is the process of dividing a digital image into segments (groups of pixels). The goal of segmentation is to group together pixels that are similar in some ways and distinguish groups of pixels that are different. So, the representation of an image is simplified or changed into something more meaningful and easier to analyze. Image segmentation is used in different practical applications, such as object detection, recognition tasks, machine vision, medical imaging, etc. There are different approaches to image segmentation, such as threshold-based segmentation, region-based segmentation, edge-based segmentation, and more. In threshold-based segmentation, every pixel is compared to a threshold, ending with two groups of pixels. A group has pixels larger than the threshold and a group has pixels less than the threshold. The selection of the threshold can be manual or automatic based on prior knowledge or information about image features. There are different algorithms for threshold-based segmentation, such as the basic thresholding algorithm, histogram-based thresholding, etc. The Triangle threshold and the Otsu threshold are examples of histogram-based thresholds.

In the Triangle threshold [13], the goal is to construct the a line between the maximum value of the histogram and the minimum value of the histogram. Then, the distance is calculated between the line and the histogram and the maximum distance between the line and histogram is the threshold point.

In region-based segmentation, pixels are divided into different regions having similar characteristics. There are two approaches to region-based segmentation, which are region-growing methods and region-splitting and merging methods [14].

2.4 Machine learning

Machine Learning (ML) is a subfield of artificial intelligence and computer science that allows computers to emulate the way that humans learn by focusing on the use of data and algorithms. A formal definition of machine learning is that "a computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E." Tom Mitchell, 1997 [15]. There are four basic machine learning approaches, which are supervised learning, semi-supervised learning, unsupervised learning, and reinforcement learning. In supervised machine learning, the training data is labeled, where for every feature x there is a label y. Supervised learning is used in different methods, such as support vector machines, neural networks, linear regression, logistic regression, and more. In unsupervised machine learning, the training data is not labeled, and the machine tries to learn by itself by trying to find connections between data. Unsupervised learning is used in image and pattern recognition, data analysis, cross-selling strategies, and customer segmentation. Unsupervised learning is used in different methods, such as clustering (k-means), visualization and dimensionality reduction (principal component analysis), association rule learning (Apriori), neural networks, and more. Semi-supervised learning is an in-between machine learning type where a small portion of training data is labeled. The machine is learning by the impact of the taken action, whether it should be rewarded for this positive action or penalized for this negative action.

2.4.1 Support vector machine

Support Vector Machine (SVM) is a supervised machine learning model [16]. It is used for classification and regression problems, but is mostly used for classification problems. In classification problems, the goal of SVM is to find the appropriate N-dimensional hyperplane that distinctly separates data points, where N is the number of features in the data. The choice of the appropriate hyperplane depends on the accuracy of separating the data points. So the hyperplane with the largest margin from the data points of the different classes is the desired one, where the margin is the smallest distance between the decision boundary and any of the input vectors. The decision boundary is boundary separating the examples of different classes Support vectors are data points that affect the orientation and position of the hyperplane. The equation of the hyperplane is given as follows:

$$wx - b = 0 \tag{2.3}$$

Where w is a real-valued vector whose dimension is equal to input feature vector x, b is a real number, and the expression wx means $w_1x_1 + w_2x_2 + \ldots + w_nx_n$, and n is the number of dimensions of the feature vector x. The predicted label for some input feature vector x is given as follows:

$$y = sign(wx - b) \tag{2.4}$$

Where $sign$ is a mathematical operator that takes the input value and returns $+1$ if the value of the input is positive or -1 if the value of the input is negative.

The goal of SVM is to benefit from the dataset and find optimal w^* and b^* for parameters w and b. Then by finding these optimal values the model $f(x)$ can be defined as follows:

$$f(x) = sign(w^*x - b^*) \tag{2.5}$$

So, some constraints are needed to be satisfied to find w^* and b^*. These constraints are as follows:

$$wx_i - b \geq +1 \quad if \quad y_i = +1. \tag{2.6}$$

$$wx_i - b \leq -1 \quad if \quad y_i = -1. \tag{2.7}$$

The objective of SVM is to maximize the margin $\frac{1}{\|w\|}$ by minimizing the the Euclidean norm of w which will lead to a better generalization. Where the Euclidean norm of w is denoted by $\| w \|$ and is given by:

$$\sqrt{\sum_{i=1}^{n} (w_i)^2} \quad \text{and} \quad \| w \|^2 = w^T w \tag{2.8}$$

The objective function is given as follows:

$$Min \quad \frac{1}{2}\| w \|^2 \quad subject\ to\ the\ constraints \quad y_i(wx_i - b) \geq 1 \tag{2.9}$$

There are two types of SVM which are linear SVM ,and non linear SVM . In the linear SVM classifier, data are linearly separable and the dataset can be classified into two classes by using a single straight line. In non linear SVM classifier, data cannot be separated linearly and the idea is to find a hyperplane in higher dimension that can separate data points. This idea is called kernel trick and a kernalized SVM is used. There are different kernel functions available such as linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid. The RBF kernel function for two points or samples x_i and x_j can be computed as follows:

$$K(x_i, x_j) = exp(\frac{-1}{2\sigma^2}\|x_i - x_j\|^2) \tag{2.10}$$

Where, σ is our hyperparameter and variance and $\|x_i - x_j\|^2$ is the Euclidean Distance between the two points x_i and x_j. The kernel trick is that SVM does not actually have to perform the actual transformation on the data points to the new higher dimensional space. Fig (2.6) illustrates An example of an SVM model for 2-dimensional feature vectors.

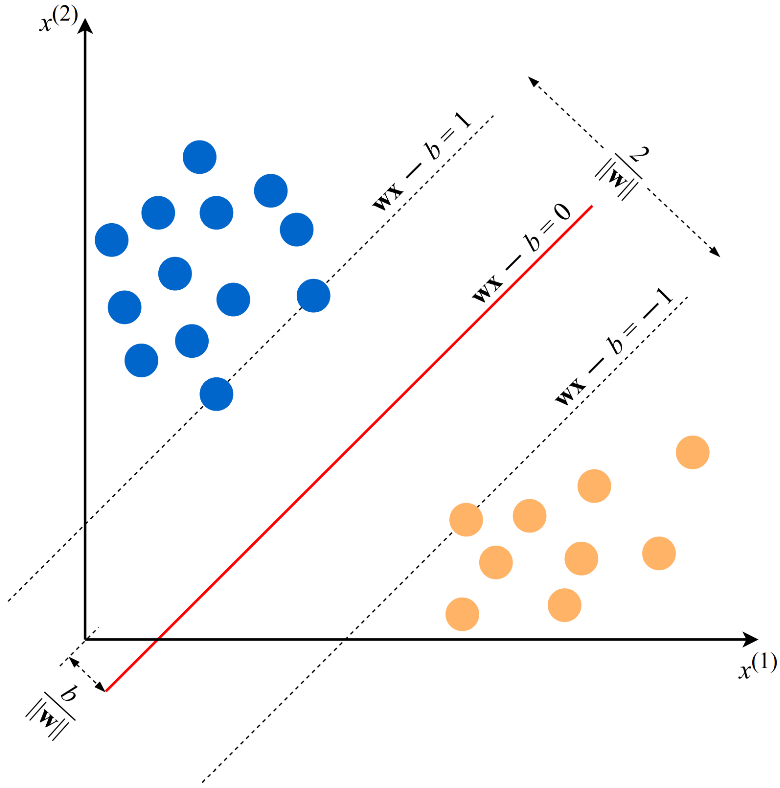
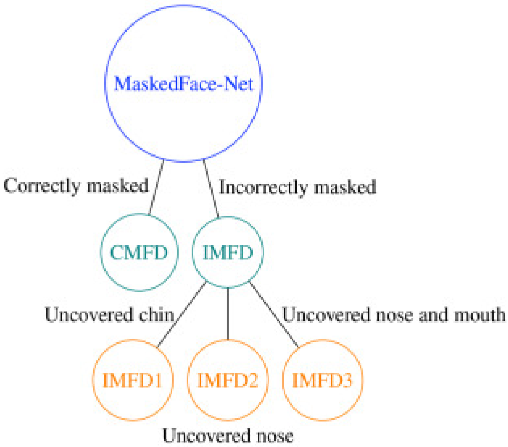


Figure 2.6: An example of an SVM model for 2-dimensional feature vectors (Adapted from [17])

2.5 Literature review

Adnane Cabani, Karim Hammoudi, Halim Benhabiles, and Mahmoud Melkemi (2020) illustrated in their paper [18] how the face masks dataset was constructed. The face images Flickr-Faces-HQ3 (FFHQ) dataset has been selected as a base for creating an enhanced dataset Masked Face-Net composed of correctly and incorrectly masked face images. They applied a mask-to-face deformable model by using an image of a widespread face protection mask (single-use blue face protection mask), where faces are detected using Haar feature-based cascade classifiers to get the region of interest which is the face rectangle. A specific KeyPoint detector "shape predictor 68 face landmarks (model derived from Sagonas et al. (2016) [19]) is applied to the detected region of interest and permits to detect 68 landmarks of the facial structure automatically. Fig (2.7a) illustrates the view of the masked face data set tree. Fig (2.7b) illustrates the pseudo-code of the mask to face model. So, a dataset set of about 137,016 images is constructed containing both correctly and incorrectly worn face masks.



(a) View of the MaskedFace-Net data tree.

```
function GENERATING MASKEDFACE-NET(FFHQ, mask)
    for (each face image ∈ FFHQ) do
        FaceDetection()
        FacialLandmarksDetection()
        if (CMFD targeted) then
            Match(landmards)
            Map(mask, face)
        end if
        if (IMFD1 targeted) then
            Match(landmards)
            Map(mask, face)
        end if
        if (IMFD2 targeted) then
            Match(landmards)
            Map(mask, face)
        end if
        if (IMFD3 targeted) then
            Match(landmards)
            Map(mask, face)
        end if
    end for
    return MaskedFace-Net
end function
```

(b) Pseudo-code for the generation of MaskedFace-Net.

Figure 2.7: Fig 2.7a illustrates the layout of the dataset. Fig 2.7b represents the pseudo-code of the mask to face mode in paper [18]

Mingjie Jiang, Xinqi Fan, and Hong Yan, (2020) [20] proposed RetinaFaceMask, which is a one-stage, high accuracy, and efficient face mask detector. The architecture of RetinaFaceMask as shown in Fig(2.8) consists of ResNet or MobileNet as the backbone, feature pyramid network (FPN) as the neck, and context attention modules as the heads. The strong backbone ResNet and light backbone MobileNet can be used for high and low computation scenarios, respectively. They utilized transfer learning to adopt weights from a similar task face detection, which is trained on a large dataset, to extract more robust features. Furthermore, they developed a unique context attention head module that focuses on face and mask characteristics, as well as a novel object removal cross-class method, ORCC, that removes objects with lower confidence and greater IoU. On a public face mask dataset, the proposed technique delivers state-of-the-art results, where we are 2.3 percent and 1.5 percent higher than the baseline in face and mask detection precision, and 11.0 percent and 5.9 percent higher in recall and face recognition, respectively.

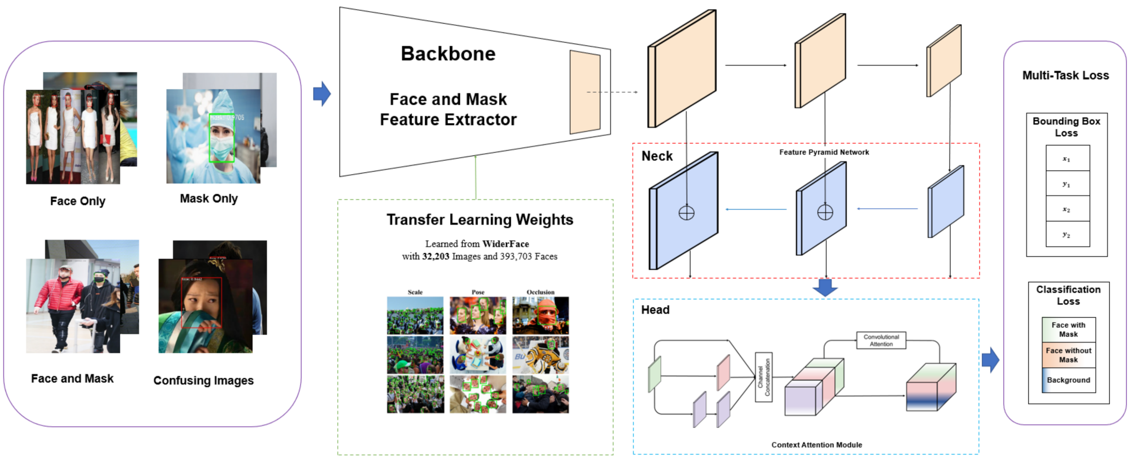


Figure 2.8: The architecture of RetinaFaceMask in paper [20]

Borut Batagelj, Peter Peer, Vitomir ˇStruc, and Simon Dobri ˇse, (2021) studied in their paper [21] the problem of face mask detection. The used dataset was constructed from MAFA (MAsked FAces) dataset and Wider Face-detection benchmark using specific metrics, partitioning the selected images into a training and testing set that can be used in standardized experiments to evaluate and compare face-mask detection techniques, and to make this dataset useful they added additional labels like gender, age, pose and ethnicity. They conducted an experimental study that looked at: (i) the performance of existing face detectors with masked-face images, (ii) the feasibility of recognition techniques aiming at the detection of properly worn face-masks and (iii) the usefulness of existing face-mask detection models for monitoring applications in the fight against COVID-19. The results showed that all tested detection models significantly deteriorate in performance when trying to detect masked faces compared to the performance observed with faces without masks. The most stable here was the RetinaFace model that also includes a generative component in the detection procedure. Furthermore, they observed that it is possible to design efficient techniques for recognizing faces with properly placed masked and that the selection of model architecture plays only a limited role in the final recognition performance. Finally, they demonstrated that existing models for face-masked detection have only limited value for real-life applications, as they only detect the presence of facial masks in the images, but not how these masks are placed.

Chapter 3

Methodology

3.1 Overall Pipeline

A two-stage pipeline is constructed, which consists of face detection and face classification stages, as illustrated in Figure (3.1). It takes as an input an image or a video frame then, face detection is applied to extract the face. The detected face is fed to the face classification stage where it determines the presence of the mask or not. And if the Face mask is present, the type of the mask is determined. The second stage depends on voting from the classification outputs from the built models.

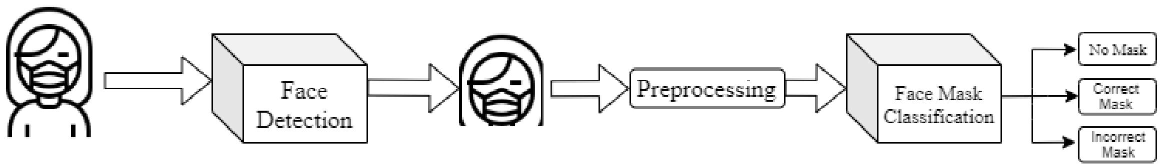


Figure 3.1: Overall Pipeline

3.2 Dataset

The MaskedFace-Net dataset (`https://github.com/cabani/MaskedFace-Net`) and FFHQ Faces Data Set (`https://kaggle.com/greatgamedota/ffhq-face-data-set`) are used. The MaskedFace-Net dataset consists of 133,783 human face images with correctly (67,049 images) and incorrectly worn face masks (66,734 images). FFHQ Faces dataSet consists of 70,000 high quality human face images.

3.2.1 Preprocessing

We separated images according to their labels into different folders. After detecting faces in the first stage, we stored them on the disk with the appropriate label and in the appropriate folder. So, we ended with five folders (Correctly worn face masks, no face masks, Incorrectly worn face masks covering only the nose and the mouth, Incorrectly worn face masks covering only the mouth and the chin, Incorrectly worn face masks covering only the chin).

For the second stage, for every machine learning model all the detected faces are resized to be of size 128x128, transformed to their grayscale version, and reshaped into a 1-D array.

3.3 Stages

3.3.1 Face Detection

At first, we tried the OpenCV library Haar for face detection . But due to the low accuracy of the HAAR cascade classifier in detecting faces, A Multi-Task Cascaded Convolutional Neural Network, or MTCNN for short is used instead. It is a 3-Stage neural network detector with a cascade structure. The 3 neural networks are Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net). Each of the three models in the cascade (P-Net, R-Net, and O-Net) are trained on three tasks, and make three types of predictions which are face classification, bounding box regression, and facial landmark localization.

3.3.2 Face Mask Classification

After detecting faces comes the Face Mask Classification phase. In this stage, different approaches are explored to detect the presence of the face mask on the face then determining the type of the face mask.

After trying the following approaches, we created a voting system using approach (2).

The Approaches :

1. Face mask detection.
   - (a) Face mask detection using Edge detection method.
   - (b) Face mask detection using Segmentation and Machine learning.
2. Face mask detection and classification.
   - (a) Face mask detection and classification depending on nose and mouth presence.
   - (b) Machine learning using Support Vector Machine classifiers

i. Training a model on all types of face masks (Correct, Incorrect) and faces without face masks.
ii. Training a model on faces with and without face masks, and training a model on faces with face masks correctly and incorrectly worn ones.

Edge detection for mask detection

In this approach, Canny edge detection algorithm is used to detect the presence of the face mask. Trying canny function from both OpenCV library and Skimage on a face, finding that one from Skimage library is more precise by adjusting sigma parameter to be equal 3. The idea is to depend on detection of eyes and getting their bounding boxes. Then getting midpoint of each eye box in x direction and getting maximum y of both eyes. And applying Canny edge detector producing edge image. Iterating on the Y dimension of the edge image starting from y calculated from the previous step and finding edge having x dimension as midpoint of each eye box as if we are moving with a pencil downwards on the edge image from each midpoint of the X dimension of the eyes bounding box until hitting an edge.

Segmentation and machine learning for mask detection

In this approach, Segmentation method and machine learning are used to detect the presence of the face mask or not. Applying a Histogram-based segmentation method as Triangle threshold on the detected faces. Then training a support vector machine model on the output thresholded images by defining two classes which are No face mask class and Face mask class.

Nose and mouth detection for mask detection and classification

In this model, nose and mouth cascade classifiers are used. The idea is to depend on nose and mouth detection from cascade classifier with the help of facial landmarks detection from MTCCN to know whether the detected face is with a face mask or not. And if the face mask is found, then the type is determined. In case of not detecting both nose and mouth, the detected face is then fed to a Support vector machine model. This SVM model will differentiate between the correctly worn face mask that covers the nose, mouth and chin, and the incorrectly worn face mask that covers only the nose and mouth.

Support vector machine classifiers

Applying supervised classification Machine learning models like Support Vector Machine or SVM for short on the detected faces. For the Support vector classifier (SVC), the used kernel is the Radial Basis Function (RBF), the Regularization parameter (C) equals 1.0, the Kernel coefficient (gamma) is scale.

- A 3-class SVM model

  We trained the SVM model on 3 classes which are No mask face images, correctly worn face mask face images, and incorrectly worn face mask face images.

- A pipeline of 2-class SVM models

  We trained a SVM model on 2 classes which are No mask face images, and all mask face images including the correctly worn face masks, and the incorrectly worn ones.

  Then we trained a SVM model on 2 classes which are correctly worn face mask face images, and incorrectly worn face mask face images.

  A pipeline of two models is constructed. The first model will be responsible for detecting whether the face mask is worn or not, then if it is a mask case, the second model will be responsible for classifying the mask type according to its wearing correctness.

3.4 Evaluation metrics

- **Face Detection stage**

  For this stage, We used the accuracy (3.3) and average accuracy (3.4) as metrics.

- **Face Classification stage**

  - **Edge detection for mask detection**

    For this approach, we employed accuracy (3.3) and average accuracy (3.4) as a metrics.

  - **Segmentation and machine learning for mask detection**

    For this approach, We used Confusion Matrix, Precision (3.1), Recall (3.1), F1-Score (3.2) and Accuracy (3.3).

  - **Nose and mouth detection for mask detection**

    For the part of using Cascade classifiers, we employed accuracy(3.4) as a metrics. For the part of machine learning we employed Confusion Matrix, Precision, Recall, F1-Score and Accuracy as metrics.

  - **Support vector machine classifiers**

    For this approach, We used Confusion Matrix, Precision, Recall, F1-Score and Accuracy.

- **Evaluation metrics are computed as :**

$$Recall = \frac{TP}{TP + FN} \quad , \quad Precision = \frac{TP}{TP + FP} \tag{3.1}$$

$$F1 - Score = \frac{1}{Recall + Precision} \tag{3.2}$$

$$Accuracy = \frac{Number\ of\ correct\ prediction}{Total\ number\ of\ predictions} \tag{3.3}$$

$$Average\ accuracy = \frac{The\ summation\ of\ accuracies\ across\ distinct\ classes}{Total\ number\ of\ claaes} \tag{3.4}$$

TP, FP and FN represent true positive, false positive and false negative respectively.

Chapter 4

Results

4.1 Overall results

4.1.1 Face Detection

Fig (4.1) shows the result faces from applying both MTCNN and Haar on different face classes such as face with and without face masks.

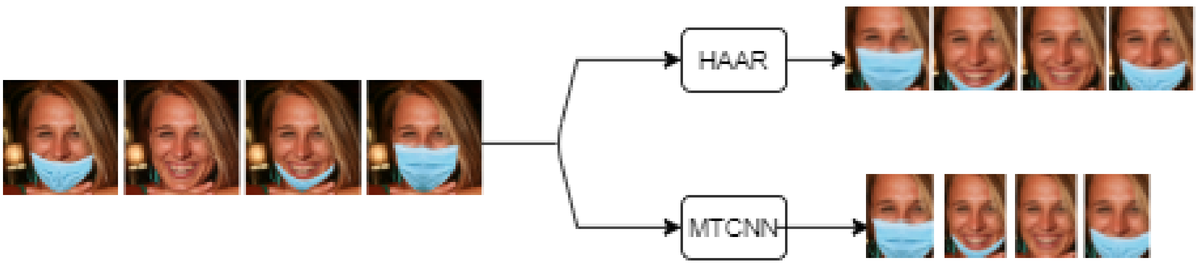


Figure 4.1: Face detection examples using both MTCNN and Haar

4.1.2 Face Mask Classification

Edge detection for mask detection

Fig (4.2) illustrates how Edge detection for mask detection approach works by taking a face with correct worn face mask as an example.

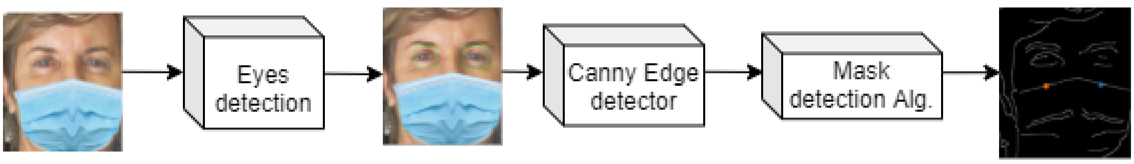


Figure 4.2: Edge detection for mask detection approach visualisation

Segmentation and machine learning for mask detection

Fig (4.3) illustrates how Segmentation and machine learning for mask detection approach works on different face classes such as face with face mask including the correct worn ones and incorrect worn ones and faces without face masks.

Figure 4.3: Segmentation and machine learning for mask detection

Nose and mouth detection for mask detection and classification

Fig (4.4) illustrates how Nose and mouth detection for mask detection and classification on different face classes such as face with face mask including the correct worn ones and incorrect worn ones and faces without face masks.

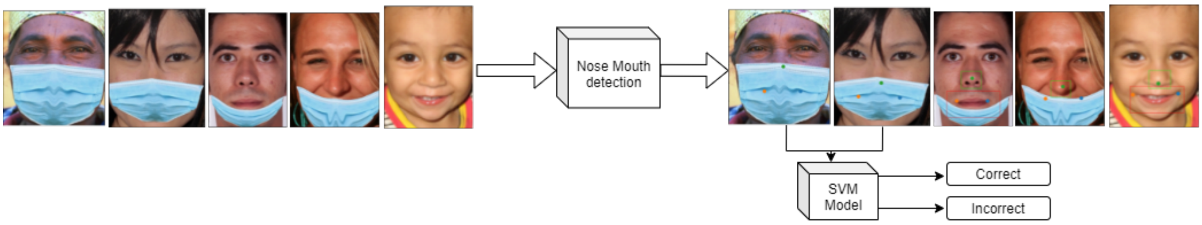


Figure 4.4: Nose and mouth detection for mask detection and classification approach visualisation

Support vector machine classifiers

A 3-class SVM model

Fig (4.5) illustrates how A 3-class SVM model on different face classes such as face with face mask including the correct worn ones and incorrect worn ones and faces without face masks.

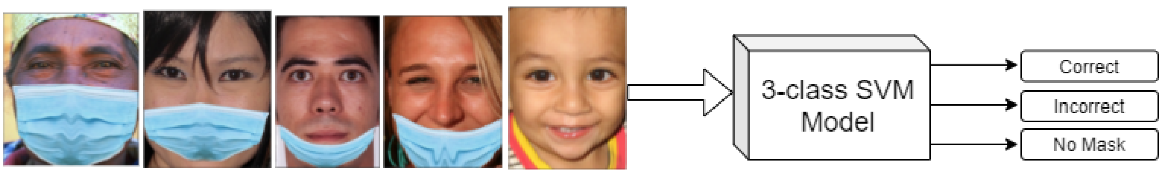


Figure 4.5: A 3-class SVM model for mask detection and classification approach visualisation

A pipeline of 2-class SVM models

Fig (4.6) illustrates how a pipeline of 2-class SVM models approach work by taking a face with correct worn face mask as an example.

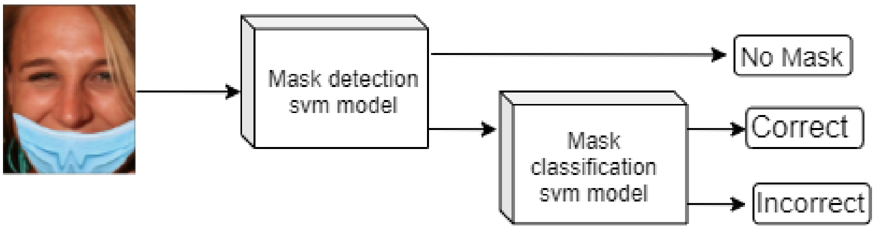


Figure 4.6: A pipeline of 2-class SVM models for mask detection and classification approach visualisation

4.2 Face Detection stage results

For face detection, we applied both MTCNN and Haar to different face image classes. Fig (4.7) illustrates the accuracy of MTCNN and Haar in detecting faces. The different face image classes are no face mask images, correctly worn face mask images, and incorrectly worn face mask images. The incorrectly worn face mask consists of three types: the exposed nose faces, the exposed nose, mouth faces, and the exposed chin faces. We found that MTCNN is more accurate than the Haar Cascade classifier in detecting faces with face masks either correctly or incorrectly worn with an average accuracy of 97.6%. In contrast, Haar can detect faces with masks with an average accuracy of 84.8%.

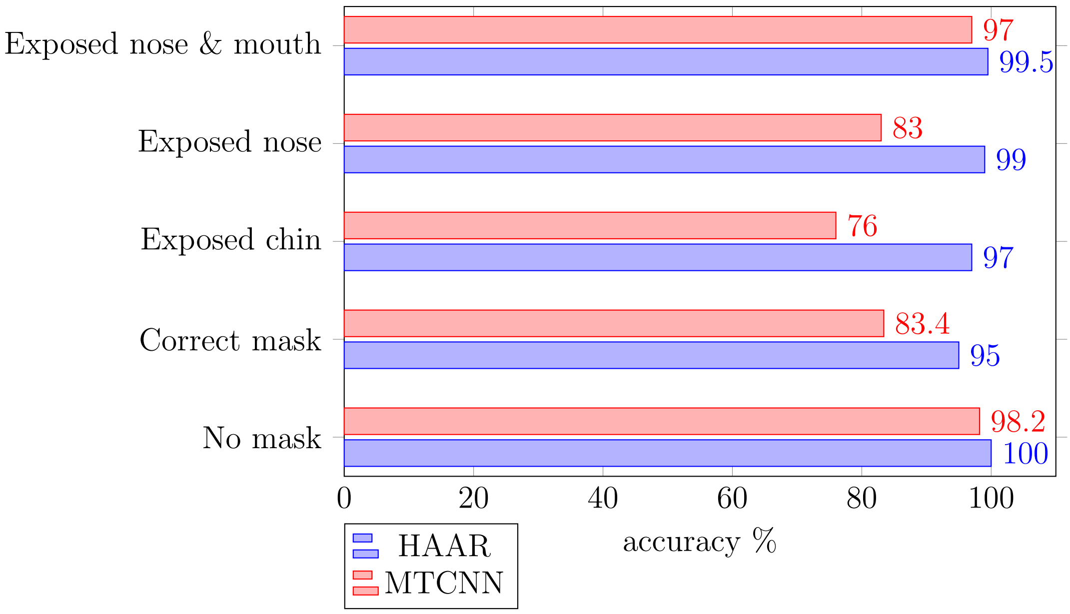


Figure 4.7: Visualisation of the accuracy of both Haar and MTCNN on different face classes

4.3 Face Mask Classification stage results

4.3.1 Edge detection for mask detection

To detect face masks, we depend on eye detection by getting the midpoint of both eyes bounding boxes and finding edges while moving down on the detected face as shown in Fig (4.8). Fig (4.9) illustrates the accuracy of applying this approach on different face images classes. We found that this approach has an accuracy of 66% in detecting no face mask. For the correctly worn face mask, the accuracy was 59.4%. For the incorrectly worn face mask where the nose is exposed, the accuracy was 55%. For the incorrectly worn face mask where the chin is exposed, the accuracy was 62.4%. Finally, for the incorrectly worn face mask where nose and mouth are exposed, the accuracy was 55%. So, this approach has higher accuracy in the no face mask class.

Figure 4.8: Sample of the results after applying Edge detection approach for mask detection

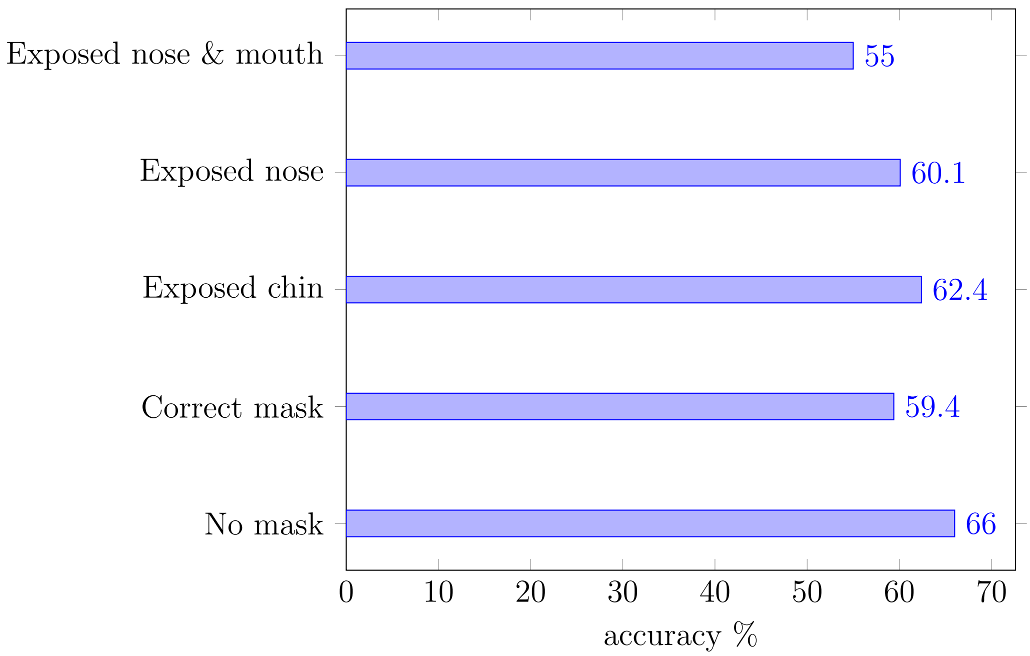


Figure 4.9: Visualisation of the accuracy of Edge detection approach for mask detection on different face classes

4.3.2 Segmentation and machine learning for mask detection

To know whether the mask is on the face or not, triangle threshold as a method of segmentation is applied. Then applying SVM model on the segmented face images by taking 70% of the data as training set and the remaining 30% as a testing and validation data. Table (4.1) illustrates the classification report of the applied SVM model . We found that this approach has an accuracy of 93% to differentiate between faces with mask and faces without mask.

class	precision	recall	f1-score	support
No face mask	0.90	0.98	0.94	21025
Face mask	0.97	0.89	0.93	19590
accuracy			0.93	40615
macro avg	0.94	0.93	0.93	40615
weighted avg	0.94	0.93	0.93	40615

Table 4.1: Classification report for the approach (Segmentation &ML for mask detection)

Fig (4.10) illustrates the SVM model confusion matrix, where rows represent the actual labels for data and columns represent the predicted label for the data. We found that out of 21025 no face mask images, 20515 images were truly classified as no face mask images while the rest were miss-classified as face mask images. And out of 19590 face mask images, 17428 images were truly classified as face mask images while the rest were miss-classified as no face mask images.

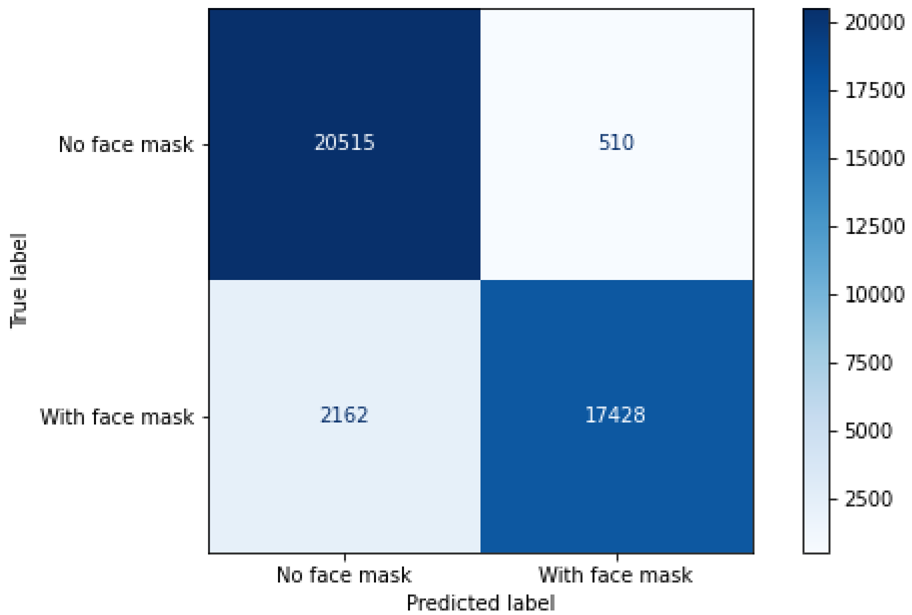


Figure 4.10: Confusion matrix visualization for Segmentation &ML approach.

4.3.3 Nose and mouth detection for mask detection and classification

Nose and mouth detection part

We applied nose and mouth detection approach to detect masks and classify them. Fig (4.11) illustrates the accuracy of the approach when applied on different face image classes. We found that this approach is more accurate in detecting Correctly worn face masks with an accuracy of 92%. Then we used a SVM model to differentiate between exposed chin mask and correct mask.

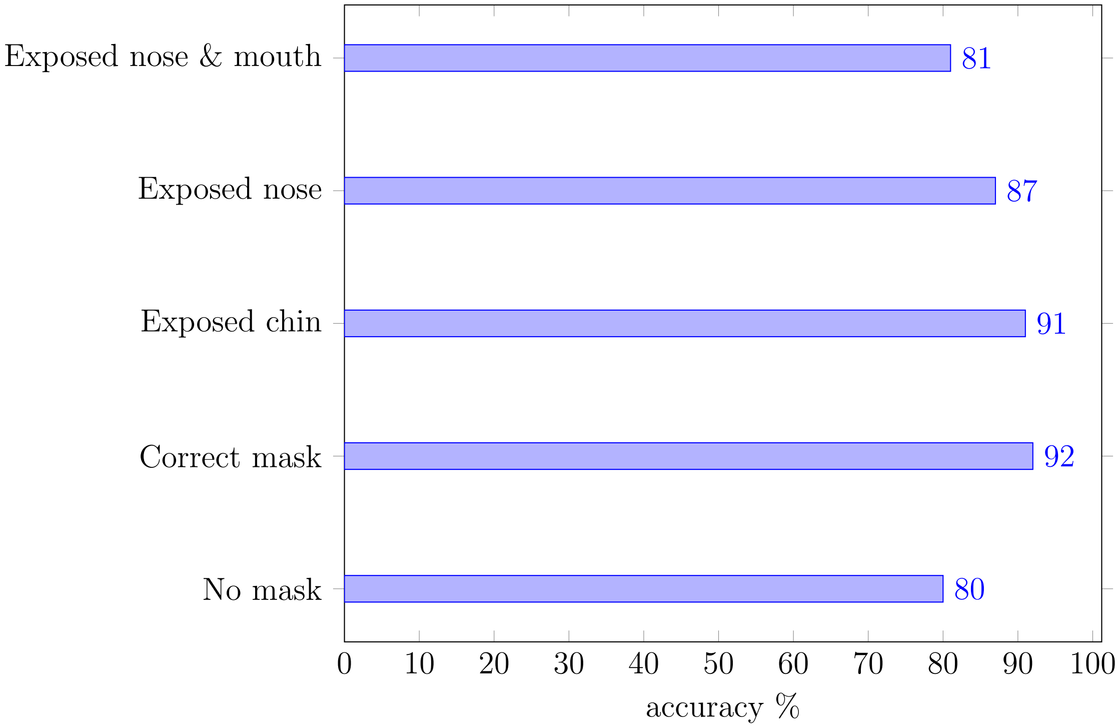


Figure 4.11: Visualisation of the accuracy of Nose and mouth detection approach for mask detection and classification on different face classes

SVM model part

In order to classify face mask type whether it correctly worn one that covers nose,mouth and chin or the incorrectly on that covers only nose and mouth, we applied SVM model on detected face images. Table (4.2) illustatrates the classification report of the applied SVM model. Taking 70% of the data as training set and the remaining 30% as a testing and validation data. We found that this approach has an accuracy of 96% to differentiate between Incorrectly worn face mask images and Correctly worn face mask .

class	precision	recall	f1-score	support
Incorrect Mask	0.98	0.94	0.96	1744
correct Mask	0.94	0.98	0.96	1744
accuracy			0.96	3488
macro avg	0.96	0.96	0.96	3488
weighted avg	0.96	0.96	0.96	3488

Table 4.2: Classification report of mask classification on two the types of the covered nose and mouth masks using SVM model.

Fig (4.12) illustrates the SVM mask classification model confusion matrix. We found that out of 1744 Incorrectly worn face mask (that covers nose and mouth only) images, 1633 images were truly classified as Incorrectly worn face mask images while the rest were miss-classified as Correctly worn face mask images. And out of 1744 Correctly worn face mask (that covers nose,mouth and chin) images, 1709 images were truly classified as face mask images while the rest were miss-classified as no face mask images. (4.12).

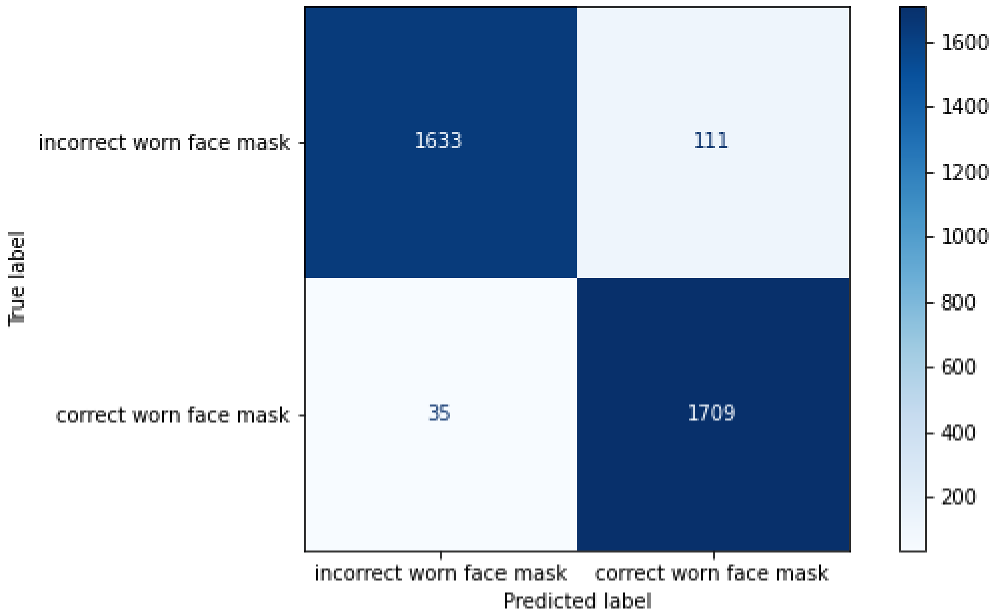


Figure 4.12: Confusion matrix visualization of mask classification on two the types of the covered nose and mouth masks using SVM model.

4.3.4 Support vector machine classifiers

A 3-class SVM model

For detection and classification of the face mask, we applied SVM model on the detected face images. Table (4.3) illustrates the classification report of the applied SVM model.

Taking 70% of the data as training set and the remaining 30% as a testing and validation data. We found that this approach has an accuracy of 97% to classify faces with correct worn face mask, faces with incorrect worn face mask and faces without face mask.

class	precision	recall	f1-score	support
Incorrect worn mask	0.97	0.94	0.96	19102
No face mask	0.97	0.98	0.97	19102
Correct worn mask	0.97	0.99	0.98	19101
accuracy			0.97	57305
macro avg	0.97	0.97	0.97	57305
weighted avg	0.97	0.97	0.97	57305

Table 4.3: Classification report of mask detection and classification model using SVM

Fig (4.13) illustrates the SVM mask detection and classification model confusion matrix. We found that out of 19102 incorrectly worn face mask images, 17985 images were truly classified as incorrectly worn face mask images, while 532 images were miss-classified as no face mask images and 585 images were miss-classified as correctly worn face mask. And out of 19102 no face mask images, 18691 images were truly classified as no face mask images, while 322 images were miss-classified as incorrectly worn face mask images and 89 images were miss-classified as correctly worn face mask. And out of 19101 correctly worn face mask images, 18863 images were truly classified as correctly worn face mask images, while 190 images were miss-classified as incorrectly worn face mask images and 48 images were miss-classified as no face mask.

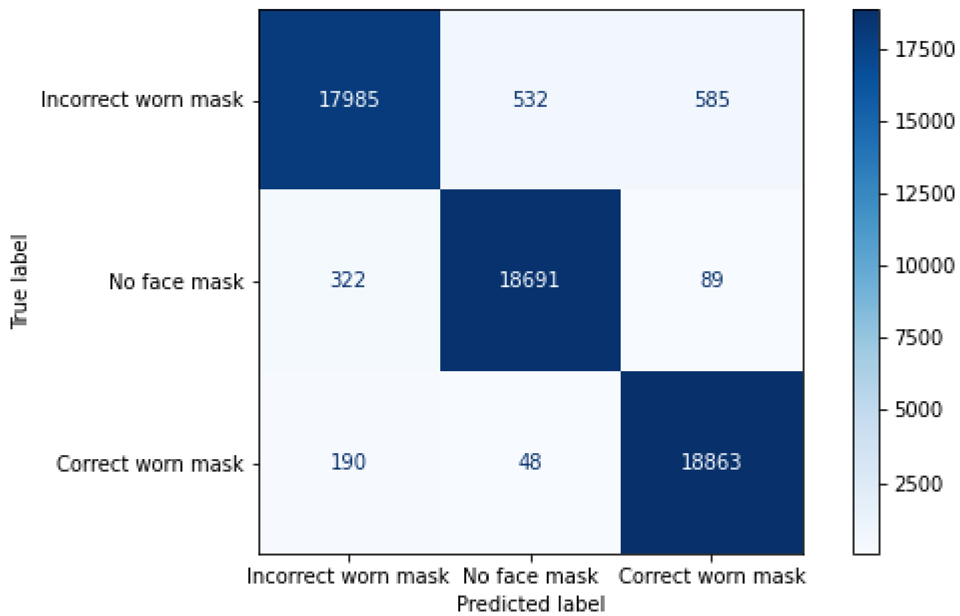


Figure 4.13: Confusion matrix visualization of mask detection and classification model using SVM.

A pipeline of 2-class SVM models

1. **Mask detection model**

   For detection of the face mask, we applied SVM model on the detected face images. Table (4.4) illustrates the classification report of the applied SVM model. Taking 70% of the data as training set and the remaining 30% as a testing and validation data. We found that this approach has an accuracy of 98% to differentiate between faces with mask and faces without mask.

class	precision	recall	f1-score	support
No face mask	0.98	0.98	0.98	19567
Face mask	0.98	0.98	0.98	19567
accuracy			0.98	39134
macro avg	0.98	0.98	0.98	39134
weighted avg	0.98	0.98	0.98	39134

Table 4.4: Classification report of mask detection model using SVM

Fig (4.14) illustrates the SVM mask detection model confusion matrix. We found that out of 19567 no face mask images, 19242 images were truly classified as no face mask images while the rest were miss-classified as face mask images. And out of 19567 face mask images, 19129 images were truly classified as face mask images while the rest were miss-classified as no face mask images.

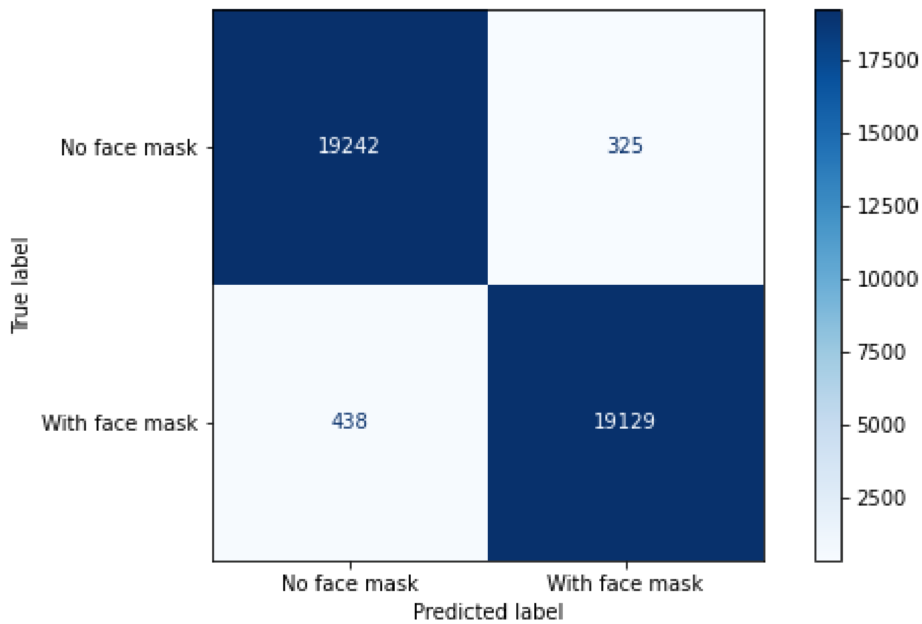


Figure 4.14: Confusion matrix visualization for mask detection model using SVM.

2. **Mask classification model**

To classify the incorrectly worn face masks and Correctly worn face masks, we applied SVM model on the detected face images. Table (4.5) illustrates the classification report of the applied SVM model. Taking 70% of the data as training set and the remaining 30% as a testing and validation data. We found that this approach has an accuracy of 98% to differentiate between correctly worn face masks and Incorrectly worn face masks.

class	precision	recall	f1-score	support
Incorrect worn mask	0.99	0.97	0.98	19102
Correct worn mask	0.97	0.99	0.98	19102
accuracy			0.98	38204
macro avg	0.98	0.98	0.98	38204
weighted avg	0.98	0.98	0.98	38204

Table 4.5: Classification report for mask classification model using SVM

Fig (4.15) illustrates the SVM mask classification model confusion matrix. We found that out of 19102 Incorrectly worn face mask images, 18444 images were truly classified as Incorrectly worn face mask images while the rest were miss-classified as Correctly worn face mask images. And out of 19102 Correctly worn face mask images, 18879 images were truly classified as face mask images while the rest were miss-classified as no face mask images.

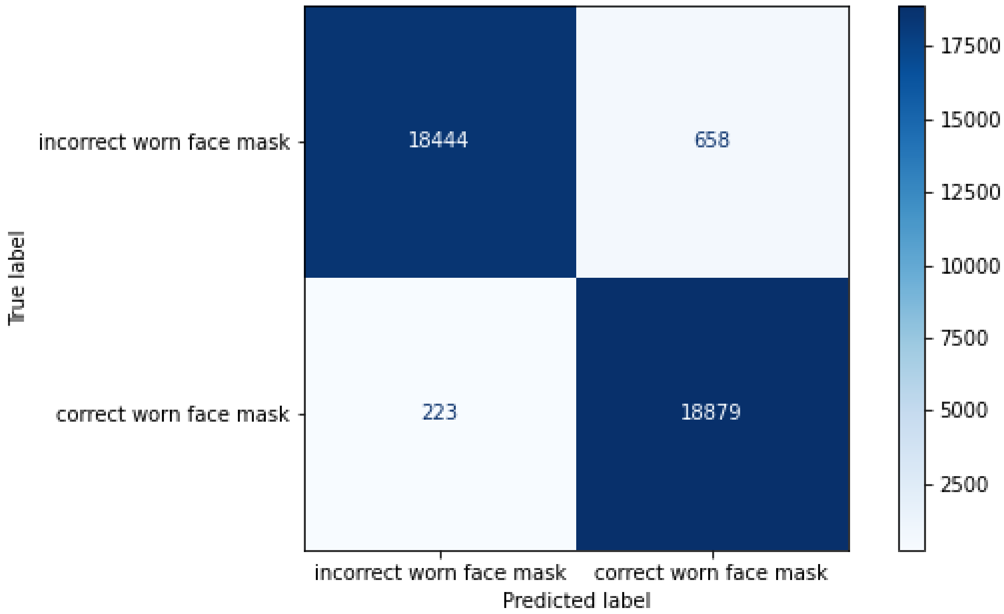


Figure 4.15: Confusion matrix visualization of mask classification model using SVM.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we studied the problem of face mask detection and classification to limit the spread of COVID-19. We tried different image processing and machine learning approaches. The problem of face mask detection and classification is solved in three stages which are (i) the Face detection stage and (ii) the Face mask detection and classification stage.

For the face detection stage, our results show that MTCNN is more accurate than Haar in detecting faces, especially the ones with face masks either correctly or incorrectly worn. Where MTCNN has an average accuracy of 97.6% in detecting masked faces, in contrast, Haar has an average accuracy of 84.8% in detecting masked faces.

For the face mask detection approaches, we tried edge detection methods and segmentation with machine learning using the SVM classifier approach. The edge detection method has an accuracy of 66% in the non-masked faces and an average accuracy of 59.2% in the masked faces. While The segmentation approach has an accuracy of 93% to differentiate between faces with masks and faces without masks.

For the face mask detection and classification approaches, we tried nose and mouth detection along with the SVM classifier and machine learning approach as SVM classifier. In the nose and mouth detection approach, the accuracy for detecting correctly worn face masks was 92%. Furthermore, the accuracy was 91% for incorrectly worn face masks where the chin is exposed. Furthermore, the accuracy was 87% for incorrectly worn face masks where the nose is exposed. Finally, for incorrectly worn face masks where nose and mouth are exposed, the accuracy was 81%, and for the no face masks, the accuracy was 80%. And for the SVM model, the accuracy was 96% to differentiate between Incorrectly worn face mask images (nose and mouth are covered) and Correctly worn face mask images (nose, mouth, and chin are covered). For the machine learning approach using SVM classifier, we had two models: A 3-class SVM model and A pipeline of 2-class SVM models. In the 3-class SVM model, the model had an accuracy of 97% to classify faces with correct worn face masks, faces with incorrectly worn face masks, and faces without face masks. The pipeline of 2-class SVM models consists of an SVM model to detect face masks and another SVM model to classify face masks if detected. The first model has an accuracy of 98% to differentiate between faces with mask and faces without mask. The second model has an accuracy of 98% to differentiate between correctly worn face masks and Incorrectly worn face masks.

We found that using SVM along with the nose and mouth detection approach for solving our problem has high premises in real-time.

5.2 Future Work

In the future, we can examine different existing face detection techniques. We can try applying deep learning models to solve the problem. Finally, We can also evaluate the current model performance on face masks other than the blue ones in the dataset.

Appendix

Appendix A

Lists

CNN	Convolutional Neural Network
COVID-19	Coronavirus disease 2019
Haar	Haar Cascade Classifiers
ML	Machine Learning
MTCNN	Multi-Task Cascaded Convolutional Neural Network
SVM	Support vector machine
WHO	World Health Organization

List of Figures

List of Tables

Bibliography

[1] World Health Organization. Coronavirus. https://www.who.int/westernpacific/health-topics/coronavirus. Accessed 20 July 2021.

[2] Here's How to Correctly and Safely Wear a Face Mask, Plus How to Not Fog up Your Glasses. https://people.com/health/heres-how-to-correctly-and-safely-wear-a-face-mask-plus-how-not-to-fog-up-your-glasses/. Accessed 21 July 2021.

[3] Kostiantyn Khabarlak and Larysa Koriashkina. Fast facial landmark detection and applications: A survey. *arXiv preprint arXiv:2101.10808*, 2021.

[4] What is Face Detection and How Does It Work?SearchEnterpriseAI. `https://searchenterpriseai.techtarget.com/definition/face-detection`. Accessed 30 July 2021.

[5] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001.

[6] H. Santoso and R. Pulungan. A parallel architecture for multiple-face detection technique using adaboost algorithm and haar cascade. 2013.

[7] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee, 2017.

[8] What are Convolutional Neural Networks? `https://www.ibm.com/cloud/learn/convolutional-neural-networks`. Accessed 30 July 2021.

[9] Sumit Saha. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.Medium, December 2018.

[10] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[11] Alem Fitwi, Meng Yuan, seyed yahya Nikouei, and Yu Chen. Minor privacy protection by real-time children identification and face scrambling at the edge. *Security and Safety*, page 164560, 05 2020.

[12] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[13] Gregory W Zack, William E Rogers, and Samuel A Latt. Automatic measurement of sister chromatid exchange frequency. *Journal of Histochemistry & Cytochemistry*, 25(7):741–753, 1977.

[14] Dilpreet Kaur and Yadwinder Kaur. Various image segmentation techniques: a review. *International Journal of Computer Science and Mobile Computing*, 3(5):809–814, 2014.

[15] Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

[16] Derek A Pisner and David M Schnyer. Support vector machine. In *Machine Learning*, pages 101–121. Elsevier, 2020.

[17] Andriy Burkov. *The hundred-page machine learning book*, volume 1. Andriy Burkov Canada, 2019.

[18] Adnane Cabani, Karim Hammoudi, Halim Benhabiles, and Mahmoud Melkemi. Maskedface-net–a dataset of correctly/incorrectly masked face images in the context of covid-19. *Smart Health*, 19:100144, 2021.

[19] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: database and results. *Image and Vision Computing*, 47:3–18, 2016. 300-W, the First Automatic Facial Landmark Detection in-the-Wild Challenge.

[20] Mingjie Jiang, Xinqi Fan, and Hong Yan. Retinamask: A face mask detector. *arXiv preprint arXiv:2005.03950*, 2020.

[21] Borut Batagelj, Peter Peer, Vitomir Štruc, and Simon Dobrišek. How to correctly detect face-masks for covid-19 from visual information? *Applied Sciences*, 11(5):2070, 2021.