Senior Project

Face Mask Detection with Voronoi Diagram

Chen Fan, Yichun Cao, Yupei Sun
Advisor: Dianna Xu

Submitted in Partial Fulfillment of the Requirements of the BA in Computer Science

Bryn Mawr College Spring 2021

Abstract

Under the influence of the Covid-19, wearing a face mask has become very critical. We propose two face mask detection algorithms based on the Voronoi Diagram with a mixture of machine learning techniques. Our algorithms are based on the detection of the existence of mouth. One is to evaluate the density of points of mouth region in the processed input image, and the other is to detect clusters in the mouth region. The general steps include face extraction, edge detection, plotting the Delaunay triangle, and passing the classifier. The final accuracy achieves 87.8% on 1000 images. The analysis and comparison between the two algorithms are also included.

Acknowledgments

We would like to thank Professor Xu for her academic and mental support and our classmates for their thoughtful advice during peer review. Also, a lot of thanks to our families and friends for their love, care, and support.

Contents

1 Introduction

Globally, by 6th March, 2021, there have been more than 115 million confirmed cases of COVID-19, including 2,571,823 deaths, reported to the World Health Organization(WHO) [STA]. Several Coronavirus disease advice was given to the public by the WHO. Among those, wearing a mask in public places and crowded areas is considered the most effective preventive measure to suppress transmission and save lives. Wearing a mask is even mandatory in many countries [STA]. The importance of wearing a mask is emphasized, however, there are still some scenarios when forcing the mask mandate is difficult, such as crowded areas like metro stations, airports, and movie theaters. Thus, the auto-identification of passengers who are not wearing a mask is necessary for situations of a lot of people with few inspectors.

Recognizing Voronoi Diagram's applications on pattern recognition, especially with people's faces, we decided to perform face mask detection via such a powerful and efficient data structure. As mainstream approaches of mask detection are through machine learning and deep learning, our proposal through VD has advantages on small dependency on training data sets and absence of the training process which is time and space consuming. With all advantages aside, one question arises that if VD has advantages of accuracy, efficiency, and flexibility on such application when compared with typical approaches in machine learning.

The main purpose of this work is to develop an algorithm to detect images of people who are not wearing face masks via the Voronoi diagram. In our proposed detection system, we use only colored images dominated by one person's face to reduce complexity. The algorithm is divided into two stages: face segmentation and mouth or face mask detection. The face segmentation is done by existing machine learning models. The mouth detection is done in two methods, one with density and the other with clustering of triangles using Delaunay Triangulation (DT). The density method includes cropping extracted face images, edge detection, and plotting DT with edge points as sites. Then, the classifying process is to calculate and compare the density of different regions and determine if a mouth is present. The DT clustering method is to cluster centroids of triangles obtained by the DT on cropped extracted face images. Using K-means clustering and DBSCAN (Density-Based Spatial Clustering), we can divide and select the triangles that are dense and close enough into different clusters to be considered as facial features such as eyes, nose, and mouth. Then we use the facial landmark detector to estimate the coordinates of the mouth on the faces and examine if there is a cluster formed in those coordinates to determine if the mouth is detected. From both mouth detection methods, if the mouth is detected, people are classified as unmasked, and masked otherwise.

2 Background

2.1 The Voronoi Diagram

To define Voronoi diagrams, we first define the points of our given finite set S as *sites*. For any *Voronoi region*, $\mathrm{Vor}(p)$ is the set of all the points x that are at least as close to site p as to any other site q in S. The points that lie on the boundary between regions do not have a unique nearest site, as they are equally close to two sites whose regions are separated by the boundary. These boundaries are also called *Voronoi edges* and its vertices are named *Voronoi vertices*. The *Voronoi diagram* $\mathrm{Vor}(S)$ is the collection of these boundaries: the set of all points in the plane that have more than one nearest neighbor [DO11]. The Voronoi diagram is a widely used technique in various applications. With such a technique, we can find the closest site to a point by just looking for its region, or get the middle point of two sites on the boundaries of the two regions they belong to. The application of Voronoi diagram includes Knuth's Post Office Problem, Autopilot, image processing, etc [ERI03].

As Figure 1 shows, the hollow points are the sites, where the black points are Voronoi vertices, and lines connecting them are Voronoi edges. The polygon regions (two are colored) are Voronoi regions, and points on this region are closest to its site.

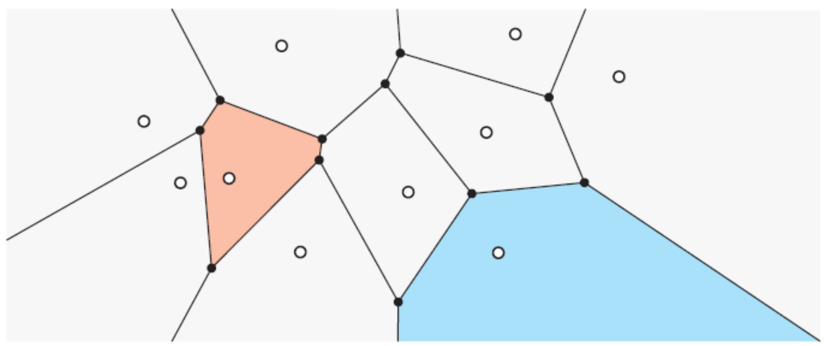


Figure 1: The Voronoi Diagram [DO11]

2.2 Delaunay Triangulation using the Voronoi Diagram

The dual of Voronoi diagram is defined by Delaunay, where any two sites whose regions share the same edge are connected. This dual concept has been denoted Delaunay tessellation or Delaunay triangulation [AK00]. To create Delaunay triangulation, we first construct the dual graph to the Voronoi diagram $\mathrm{Vor}(S)$. As shown in Figure 2, the nodes of the dual graph are the sites of $S$and if two sites share a Voronoi edge, they are connected by an arc crossing the corresponding edge. Figure 2 is the dual graph to Figure 1. Then, as shown in Figure 3, if we straighten the arcs, every three sites connected to form a Delaunay triangle.

Delaunay also proved that the straight-line dual graph of the Voronoi diagram is planar since the Voronoi diagram is drawn on the plane. This dual graph captures the proximity between regions of the Voronoi diagram, or adjacency from site to site [DO11]. We will further explain how to employ the Voronoi diagram and Delaunay triangulation in the Methodology section.

2.3 Overview of Algorithms Computing Delaunay Triangulation

There are a few well-known algorithms in computing Delaunay triangulation. One is the randomized incremental algorithm, which adds each site and updates the Delaunay triangulation after each addition. The update consists of discovering all Delaunay triangles whose

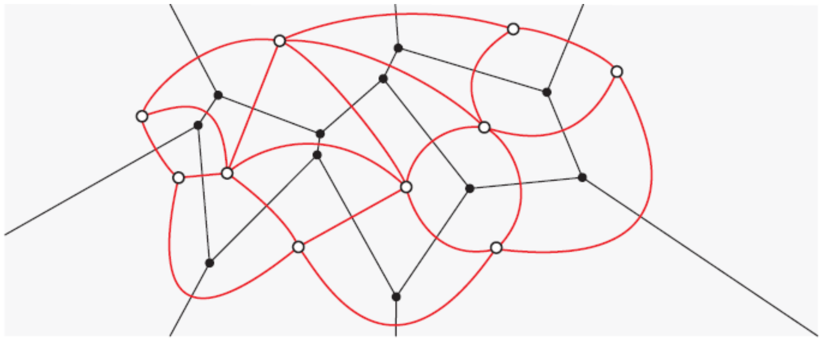


Figure 2: The dual graph of the Voronoi Diagram [DO11]

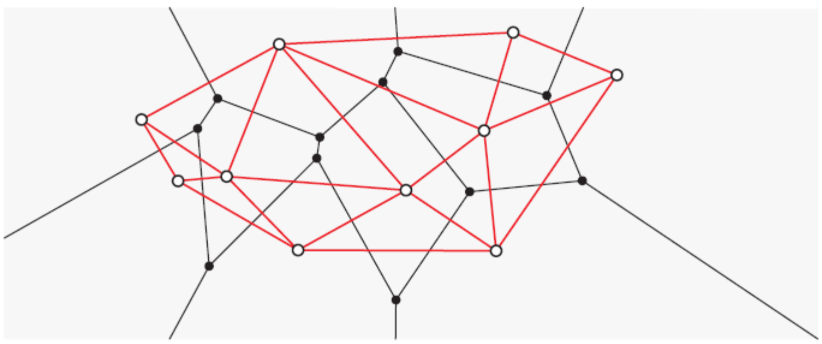


Figure 3: The straight-line dual graph of the Voronoi Diagram [DO11]

circumspheres contain the new site. These triangles are deleted and the empty the region is partitioned into new faces, each of which has the new site as a vertex. Another one is the plane sweep algorithm which computes a planar Delaunay triangulation using a horizontal line that sweeps upward across the plane. Other algorithms include the planar divide-and-conquer algorithm, flipping algorithm, and gift-wrapping algorithm which we will not go into further detail [GT17]. Our implementation will be using Delaunay Triangulation in both density and clustering methods to detect face masks.

3 Related Work

To our knowledge, we are the first to use VD on face mask detection, while most current research relies on machine learning techniques. Nevertheless, we have found related works on face detection by Voronoi diagram and by other approaches along with face mask detection using other algorithms to provide insight to our approach.

3.1 Face Detection by Voronoi Diagram

As face detection becomes widely used, we can see the application of it when entering an office building, unlocking our phone, or even going back home. As a well-known computational geometric data structure, the Voronoi Diagram can also be applied to face detection by triangulating to the facial region (Delaunay Triangular). The problem of face detection usually presents two difficulties: the complexity of modeling a vast amount of visual data in an image and intrinsic ambiguity in image perception [CMM08]. Because of these properties, several algorithms have been developed.

The one that inspired us is Abbas et al's research [CMM08] on face segmentation and feature extraction. For the image pre-processing part, Abbas used Histogram Equalization to reduce lighting ill effect and VD on feature points that results from gray intensity frequencies first. This is very similar to dynamic thresholding method for segmentation except that this is more based on divide and merge decision making. This method of pre-processing the data allows it to handle the segmentation of other types of images and can be extended to RGB images. Also, comparing to K-means clustering, this method is fully automated and unsupervised which does not require any parameter tuning. After processing the image, the author then used Distance Transformation, Ellipse Fitting, and Euler Number to extract the face region. These processes segmented the face region from a complete image, and for the face feature extraction, the author then used Eye Binary Template Model to locate eyes and Euclidean Distance to calculate the positions of nose and mouth.

Yi Xiao and Hong Yan's work [XY01] mainly focuses on identifying facial features. They first used intensity illustration of a negative grayscale facial image with a threshold to extract eyes and mouth from facial skin. And then, they applied DT on the extracted part to search for clustering of sites and calculated the area of the clustering to examine if it's eye or nose. Comparing to Abbas's work, Yi Xiao and Hong Yan's work are limited to binarized images and are not so good at reducing enough noise in the image as Abbas did.

Later work by Khitikun et al [MDR+13] also used a combination of similar algorithms, namely Distance Transformation, Ellipse Fitting, and the Euler number to segment face from images, but the main purpose of their work was face recognition. Therefore their work contained other works of face feature classification, which are not related to our work.

A most recent work from Moulay Ismail University by Safa Jida et al [JAO17] followed similar approach as Abbas did but rather than using 1D histogram for Histogram Equilization, Safa used 2D to keep the spatial distribution of pixels in the image. The advantage of their work comparing to Abbas is that their algorithm does not depend on rotations or facial expressions.

Abbas's work on face segmentation and feature extraction is most similar to our project except the detection of face mask. Based on Safa's work, we can improve the face segmentation part by using 2D histograms, and modify the feature extraction part.

3.2 Face Detection by Other Algorithms

Facial recognition can be roughly divided into two parts: "face detection" and "face localization". Where "face detection" refers to the rough estimation of the face region, "face localization" refers to the precise localization of facial features [HKK+05] and is also called feature recognition [CMM08]. We will discuss related papers in the two categories and compare their methods.

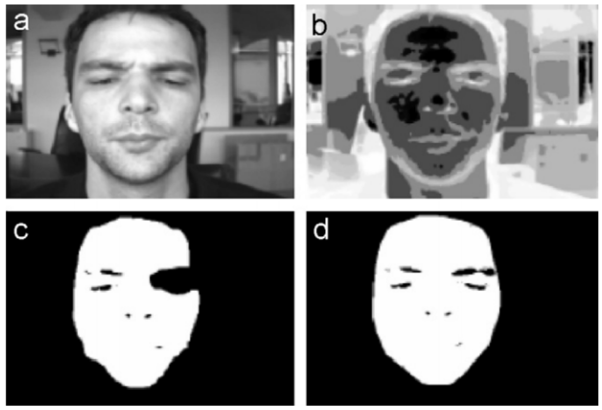


Figure 4: Faces repairing[CMM08]

3.2.1 Face Region Detection

While there is much more research on face localization, detection of face region (face detection) is less studied. However, face detection is not less helpful than face localization in our goal of detecting mask since we focus on the region of the face as well as facial features.

Viola and Jones Algorithm

Viola and Jones algorithm is an object detection method and it provides competitive advantage for face detection. This algorithm consists of three stages: cascading classifiers and Haar feature selection, creating an integral image, and adaboost training in machine learning [AI16]. As the result, Viola and Jones algorithm generates an image with only the face region. We use this algorithm since it is more effective and widely used than other methods that we will discuss below.

Chromatic Histogram

Another method is to analyze the chromatic distribution of the face region and the result is represented by a histogram in chromatic space [YO96], [ST98]. In the paper by Yoo and Oh, the authors first cut the image to roughly the face region and convert the RGB representation into HSI representation, which stands for Hue, Saturation, and Intensity respectively. They then compare the similarity of this histogram with other parts of the image to generate the more specific face region [YO96]. The color histogram and related parameters are dynamically updated over time and compared to previous ones [Kwo03]. Other research used a similar technique such as the hue and luminance components of the images in the YIQ color space [BAMU03] and their own modified models [ZRF10]. Research by Gürel et al also employs this method and can be used to detect multiple faces [GE13]. This method is more detailed in parameter tuning than Viola and Jones algorithm and thus less efficient.

Eigenfaces

The idea of eigenvector decomposition and clustering is widely used in face detection and face localization, and is one of the early approaches to these problems. In this method, a simple neural network is constructed to perform face recognition for aligned and normalized face images. The neural network computes a face description by approximating the eigenvectors of the image's auto-correlation matrix. These eigenvectors are known as Eigenfaces [YKA02]. In Wong et al's paper, the authors employ the genetic algorithm and the Eigenface technique to detect face region. The genetic algorithm is applied to search for possible face regions in an image, while the eigenface technique is used to determine the fitness of the regions [WLS01]. This method is more complicated than Viola and Jones algorithm and also less advanced.

3.2.2 Face localization

Active Shape Model and Active Appearance Model
Cootes proposed Multi-Resolution Active Shape Models (ASM) to extract shape and Active Appearance Models (AAM) to extract both shape and texture, and his development of these two critical algorithms [CTCG95] has contributed to the improvement of face recognition. Chen-Chiung Hsieh et al [HL15] also made use of ASM combined with the Voronoi Diagram, Laplacian of Gaussian filter, and Thin-plate Spline Analysis to do face moles detection and classification. Also, in the chapter written by Ting Shan et al [SBLC08], they proposed a technique to identify real-time human faces, both frontal and non-frontal, with or without the light effects, etc. They proposed a pose variability compensation technique, which is based on modeling the face via AAM, estimating the pose through a correlation model, and coupled with adaptive principal component analysis (APCA). Besides the applications of these two algorithms, Iqtait et al's paper[IMM18] compared the two algorithms and found out that ASM is faster and more accurate than AAM. Although AAM and ASM are foundational algorithms in face localization, they are hard to implement and thus we decided to use clustering techniques.

Shape and Color Information
Sobottka et al[SP96] performed face localization based the shape and color of human faces and the light conditions, and evaluated the shape and color information to detect facial features. Based on this, Javad et al [HAF02] then made some modification and added the definition of distance measure to do face localization and segmentation. Similar work has been done by other scientists as well [AP09], [TZ02], [TMS05], [CJSW01], [RAJ02]. Such methods are accurate but too detailed in parameter tuning, similar to chromatic histogram method in face region detection.

Variance Projection Function and Eye Detection
Guo Can Feng et al [FY01] proposed a novel eye detection process in their paper using multi-cues from face images. By considering the cues: the intensity of eyes, the direction of the lines joining the eyes, and the response of convolving the proposed eye variance filter. With these cues, they then used variance projection function to detect and verify. Combining Harris's response function with variance projection function, Haiying et al's work [XY09] found an approach to detect eye corners. This combination turned out to be very robust and accurate as demonstrated in their experiments. Other works on eye detection include methods such as combined binary edge and intensity information [SCL06] and linear and

nonlinear filters [AR01]. These methods are very advanced but are also hard to implement.

We have also looked into additional work in the field of face recognition. Examples are face recognition using near infrared [FFUS16], face tracking for surveillance videos [SAR12], and face recognition using various approaches [XL08], [PKB16], which are not directly related to our work.

3.3 Face Mask Detection by Other Algorithms

As the necessity of wearing face mask during the period of Covid-19 is stressed in Section 1, face mask detection has become research topic of interest and importance. Previous work on this topic are mostly done using deep learning. Comparing with most previous work found on this topic, the proposed algorithm with Voronoi Diagram has advantages that the learning process is skipped such that there much less dependency on train data and much lower demand on the time and memory space needed for the learnt model. Another advantage of VD is the flexibility and adaptability that the code can be easily adapted for glass, head set or hat detection and removal without additional data to train the machine learning. However, the shortcomings are that once the learning process is done, algorithms using deep learning may perform faster and also can integrate steps of locating facial features.

3.3.1 CNN

In [CPSA20] and [LMTK21b], a transferring learning model using InceptionV3 is proposed, which is trained and tested on the same simulated masked face dataset. Although the training and testing accuracy is about 99%, the model is trained and tested on images with only one person and the mask is simulated onto the picture. Thus, the high accuracy may not be obtainable in reality.

While [SPB21] uses the same simulated dataset for training, they made improvement on making real-time detection and notifying people who are not wearing mask via text. The mask is extracted from real-time faces in public and fed as input into CNN.

Similar work of real-time detection can be found in [BXZ+17] with a cascade CNN of three layers, [EI19] with multi-task cascaded CNN and SVM for classification, and [CDBD20] with a dual-stage CNN architecture. The three researches all achieved good accuracy.

The problem of using simulated data is addressed in [GLYL17], since it uses Medical Masks Dataset which consists of 3835 images including no mask wearing, incorrect mask wearing, and correct mask wearing. The article proposed LLE-CNNs that consists of three major modules. The first one is combining two pre-trained CNNs to extract facial regions, using locally linear embedding algorithm to turn CNNs into similarity-based descriptor, train and evaluate on the real dataset. The accuracy is 98.5%.

In [SS21] which directly uses PyTorch and OpenCV for image prepossessing and deep learning model with framework of MobileNetV2. The accuracy dropped to 79.24% since it uses video stream as data source.

The work by Jiang, Fan and Yan [JFY20] provides RetinaFaceMask as a one-stage detector, which consists of a feature pyramid network to fuse high-level semantic information with multiple feature maps, and a novel context attention module to focus on detecting face masks.

[MBV19] by Meenpal's group depends on training the Fully Convolutional Networks to semantically segment out the faces present in image with Binomial Cross Entropy used as a loss function. The algorithm can detect multiple facial masks in a single frame which achieves accuracy of 93.884%.

In [MDMM21], the author aims to perform mask detection and social distance measurement with different speed rates. The approach utilized in the article is RCNN, Fast RCNN, and Faster RCNN algorithm based on balanced face restriction, color changes, brightness changes, and contrast changes.

In some scenarios with resource constraints, the mask detection needs to be done on portable and light-weighted devices. A model named SSDMNV2 is proposed in [NJM+21] for face mask detection using OpenCV Deep Neural Network (DNN), TensorFlow, Keras, and MobileNetV2 architecture which is used as an image classifier. The proposed algorithm gives an accuracy score of 0.9264 with the advantage of lightweight which can be used in embedded devices for real-time mask detection. Similarly, [MPC20] proposes an architecture for detecting face masks for deployment on resource-constrained endpoints having extremely low memory footprints with deep learning models.

While the detection of existence of face mask is covered in various research papers, the detection of proper wearing is presented in [QL20].

Qin and Li designed a three-categories-classification that classifies images into "correct mask wearing", "incorrect mask wearing" and "no mask wearing". In their proposal, image pre-processing, face detection and crop adopts multitask cascaded convolutional neural network(CNN), and then perform image super-resolution and facemask-wearing conditions identification by the combination of CNNs with SR network with classification network(SRCNet) to improve accuracy.

The proposed system in [RSN20] uses fog nodes to process the video streams captured and Haar-cascade-classifiers to detect face portions. Each fog node deploys two MobileNet models, where the first deals with presence of face mask and the second deals with the proper wearing of the mask.

3.3.2 Combination of YOLO and CNN

One of the most popular object detection algorithms which has a robust performance is the You Only Look Once (YOLO). Many work of mask detection uses machine learning model, mostly CNN, first to extract face area and then detect face masks by different versions of YOLO. YOLO method was improved to YOLO V2 later which was able to detect over 9000 object categories and a novel, multi-scale training method was developed.

Work in [LMTK21a] gives a resolution that consists of two components that are feature extraction based on the ResNet-50 deep transfer learning model and the detection of medical face masks based on YOLO v2. This resolution proposed gives accuracy at about 80%, while it requires high-end machine to perform this model. One improvement is proposed in [WWL+21], the authors propose an in-browser serverless edge-computing based face mask detection solution. The solution integrates YOLO v2, high-performance neural network inference computing framework (NCNN), and a stack-based virtual machine(WebAssembly). The advantage is that the resolution requires minimal device limitation and privacy risk while scarifies some accuracy. Another upgraded version is YOLO v3 which is a bit more memory intensive but it is much faster and just as accurate. In [SAK+21], two state-of-the-art object detection models consisting YOLO v3 and faster R-CNN are used to achieve this task. In [SPAS20], a YOLO V4 deep learning has been chosen as the mask detection algorithm. The experimental results have been done in real-time application and the device has been installed at Politeknik Negeri Batam.

3.3.3 Other Methods besides deep learning

In [NRMB15], the research group proposed a system for the automatic detection of the mandatory surgical mask in operating rooms which will trigger alarm and thus the priority of this research is to lower the false-positive rate as much as possible. The work first utilizes Viola and Jones faces detector, and then applies classification on colored images by conditions of skin pixels ratio on different parts of the face.

In [DBUM16], the authors work on the masked face detection from the video. The masked person is detected through mainly 4 steps that are estimation of distance between camera and person, detection of eye line, detection of part of face and detection of eye.

4 Data

The datasets chosen contain colored images dominated by one face with or without face mask, which are listed as following:

- Dataset from Github contains around 1000 with-mask images which include both real and simulated ones, and 1000 without-mask images.[Bc20]
- Dataset from github contains 5000 with-mask real images from 525 people, 5000 real without-mask images. [Xz20]

5 Methodology

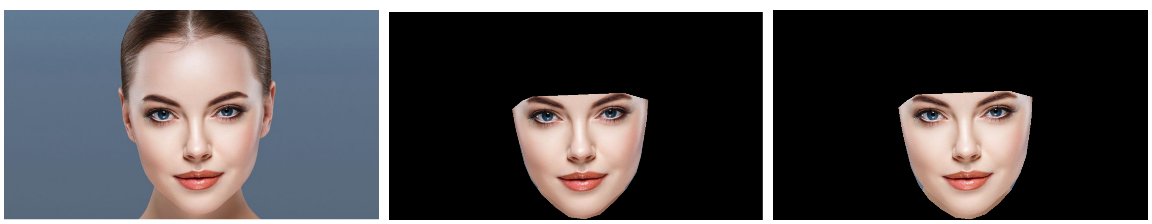
The two main parts of detecting face masks are face extraction and mask detection. The main principle of mask detection is to check the presence of mouth. Both methods require pre-processing of face extraction which segments out only the face region, image cropping which refines the segmentation of the face, and edge detection which captures major facial features.

5.1 Face Extraction

We use existing libraries to implement the face extraction part to maximize efficiency. From Modesto Mas' analysis on Python image processing libraries performance [Mas15], the OpenCV library outperforms other libraries such as Scipy and Scikit-Image. Therefore, we choose OpenCV to do the image processing. We use different methods and models such as the shape predictor method and "Facemark" models in OpenCV to extract faces from the image. Facemark is an API in OpenCV used for detecting facial landmarks, and two of its widely used models are Local Binary Features (LBF) and Kazemi, which is named after its implementer.

5.1.1 Model Selection

Since half of our data set are images of masked faces, the performance of face segmentation was as not as satisfying at first because it was difficult to extract facial features from a masked face. Therefore, we experiment on all three of the shape predictor, LBF, and Kazemi methods to see their performance. The following images are the original image and outputs of the shape predictor method and Kazemi method. Since the LBF model produces a result similar to the Kazemi model, we do not show it here for simplicity. We can see that the shape predictor method results in a more precisely extracted face, while the Kazemi method extracts a larger face area and the result contains more area of the chin and neck. Later, we can see the difference in the run time and performances of detecting faces between these methods. Note that the accuracy here is computed by the percentage of faces detected, which is different from the overall accuracy that we will discuss in the mask detection and later sections.



(a) Original Image (b) Extracted Face using Shape Predictor (c) Extracted Face using Kazemi

Figure 5: Face extraction with different methods

All our segmentation methods detect landmarks, or feature points, of the face, and then extract the face area based on these landmarks. The table below is the summary of the performance of these methods. In our experiment on 1000 images, the Kazemi model performed best with the accuracy of 79.5% in detecting faces and runtime of 376 seconds (about 6 minutes), and the shape predictor method is the runner-up for accuracy of 72.1%, but its runtime is the longest (about 23 minutes). The LBF model results in the lowest accuracy of 55.2%, and its runtime is not significantly better than the shape predictor method (about 17 minutes). Therefore, we have decided to rule out the LBF model and apply the other two methods in the following steps.

	Shape Predictor	LBF	Kazemi
Accuracy	72.1%	55.2%	79.5%
Runtime (in seconds)	1389	1055	376

Table 1: Accuracy and runtime of face segmentation methods, computed by the percentage of faces detected

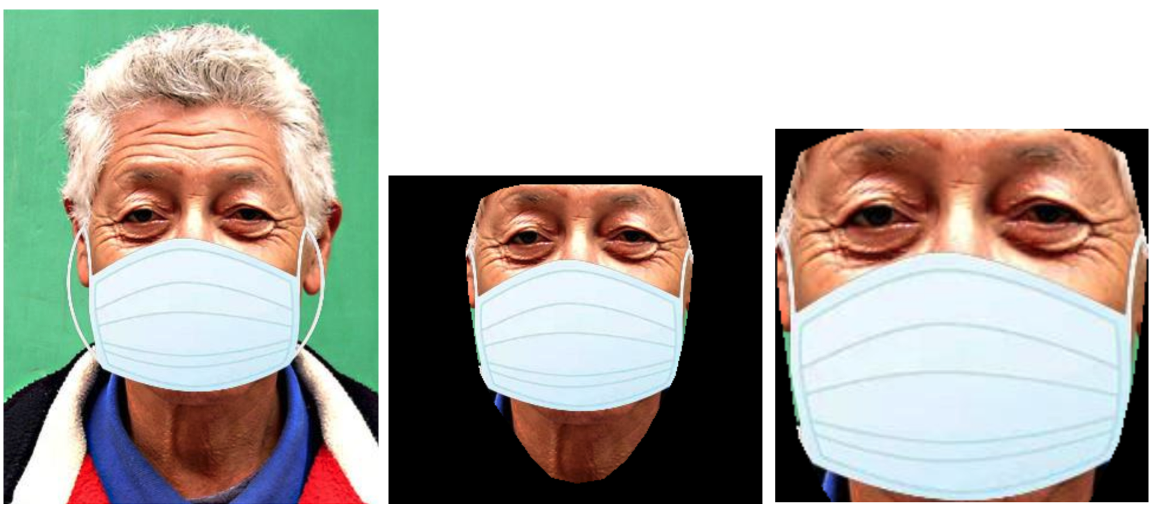
To further improve the overall accuracy of face mask detection, we choose to categorize images with undetectable faces into masked faces by default, since 75.2% of them are masked faces out of 5000 images (2500 masked and 2500 unmasked) combining the two of our datasets.

5.2 Image Cropping and Edge Detection

5.2.1 Image Cropping

Since the face extraction is performed half on masked face images, the extraction region is not as precise. In most masked images, the shaded chin and neck regions are included in the result which leads to clustering of points when performed edge detection and impacts the accuracy of the method.

The original cropping method is to find the top, bottom, leftmost, and rightmost colored pixels which determine the coordinates of the corners of the cropped square. The improved version of image cropping is simply to crop a square with side length as max width of the colored region. An example is shown below.



(a) Extracted Face (b) Face with Edge Extraction (c) Face with Region Division

Figure 6: Demonstration of different methods of image cropping in step A

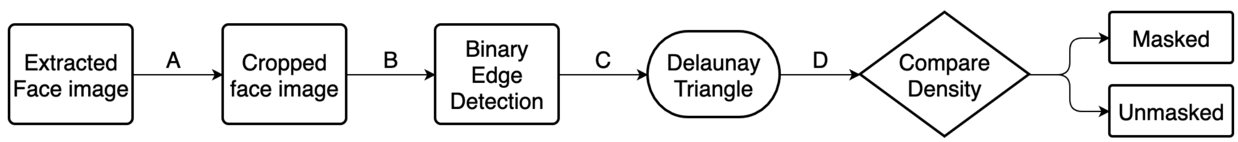
5.2.2 Edge Detection

Two methods are taken into consideration which are Canny and Sobel edge detection. Since edges are points with relatively great change of color, the norm of gradient vector is greater than non-edge points. Both methods compute an approximation of the gradient of the image intensity and filter out points with greater gradients along with other classifying functions. With a fixed threshold to classify edge points, the impact of edge detection is not sufficient to affect the performance of the mask detection accuracy. However, one major improvement from edge detection is with the auto-threshold of Canny edge detection with OpenCV implementation. The threshold for classifying the points as an edge is determined through the Gaussian blur function. The improved method is more sensitive to color changes so that the blurry mask region on unmasked images is detected which leads to great improvement in accuracy.

5.3 Mask Detection by Density

5.3.1 General Steps

The general approach is in the following graph with corresponding pictures below:

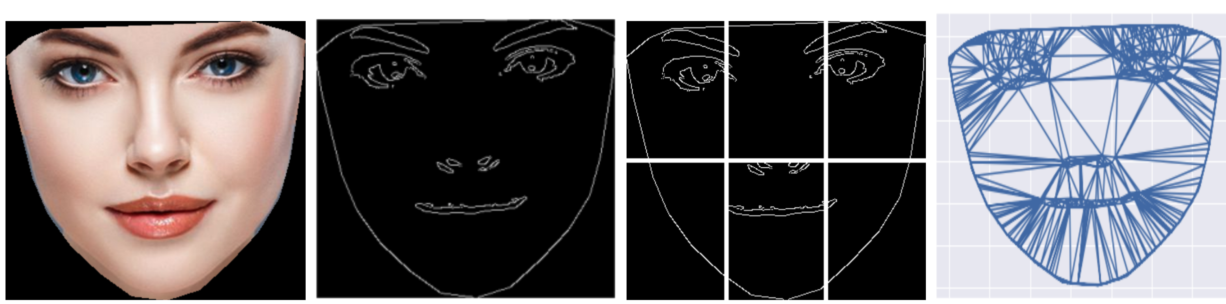


(A) Crop out only the face region from extracted face image.

(B) Edge detection by built-in functions in Open-CV.

(C) Delaunay triangle is plotted with points extracted from edge detection with probability of 0.2 for better visualization and runtime.

(D) Criterion for detecting mouth include density of the mouth region is above some threshold and if the density of eye regions is larger than that of mouth region.



(a) Extracted Face (b) Face with edge detection (c) Face with Region Division (d) Face with Delaunay Triangle

Figure 7: Processed images corresponding to the mentioned steps

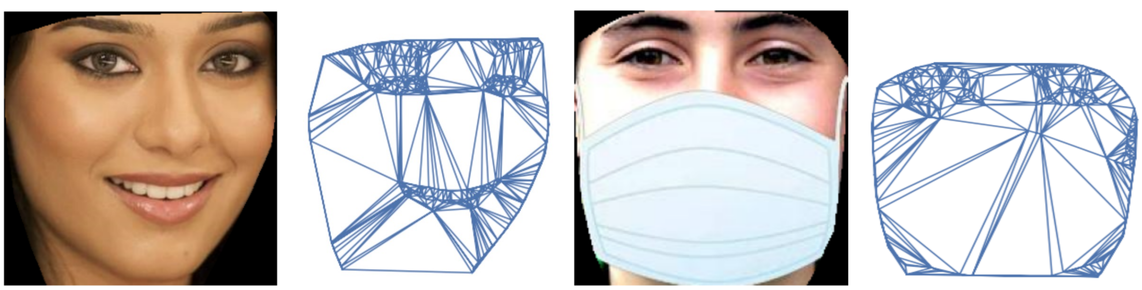
5.3.2 Modifying Methods

In each step, there are challenges and improvement conducted. Each is elaborated and the accuracy improvement is summarized.

- **Step A: Image Cropping**
  Image cropping in Section 5.2.1 improves the accuracy from 65.1% to 76.8% on 800 images that faces can be extracted.

- **Step B: Edge Detection**
  Edge detection mentioned in Section 5.2.2 improved the accuracy from 76.8% to 86.5% and false positive decreased from 116 cases to 70 cases.

- **Step C: Delaunay Triangulation**
  20% of extracted edge points are randomly selected as sites such that a better visualization is guaranteed without filled with lines. With the Delaunay Triangle (DT) plotted, the point density in all region is magnified. However, the DT also magnifies the noises such as folds on the face mask. Also, the edges of mouth is not so noticeable compared to that in eye regions and may not be detected due to poor lighting or different angles of face. Also, the densities of points in detection is confused by the DT lines between eyes and face contours. Thus, the result is not so satisfying, which is 77.4%. Thus, this step is skipped for the consideration of runtime and accuracy.



(a) Original image without mask (b) Unmasked image with Delaunay Triangle (c) Original Image with mask (d) Masked image with Delaunay Triangle

Figure 8: Comparison of DT on masked and unmasked images

- **Step D: Classification**
  On the extracted and cropped face image, the mouth locates in the lower middle area and the eyes locate at the top first and third area. There are two criterion for determining existence of mouth:
  (1) if the point density of mouth region is lower than some threshold, then the image is categorized as masked.
  (2) if the density of mouth regions is larger than that of horizontal surrounding area, meaning that there is a cluster in the mouth area.
  Since the point density of each area is fixed for each iteration, the best threshold in (1) is found by binary search. The highest accuracy achieves 86.5% but the test accuracy varies from 76.7% to 88.1%. The second criterion is chosen for stable and better performance. Other criterion taken into consideration is comparison of density between eyes and mouth region which is not satisfying since eye density is higher than mouth density for both masked and unmasked images.

5.3.3 Summary of Accuracy Improvements

Accuracy	Image Cropping	Edge detection	DT	Final
Original	65.1%	76.8%	77.4%	/
Applied	76.8%	86.5%	77.4%	86.5%

Table 2: Accuracy of mask detection, all unrecognizable images by face segmentation are not included in calculation

5.4 Mask Detection by Clustering

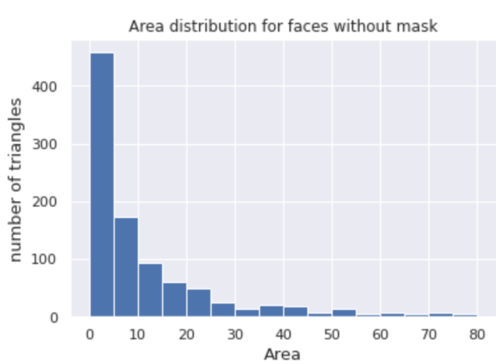
5.4.1 General Steps

With the Edge detection of images from Section 5.2.2, we now will use the following steps:

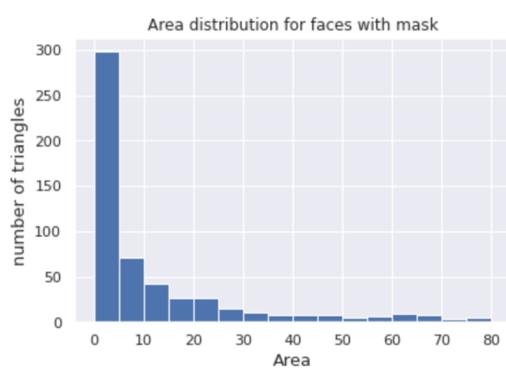
- **Step A: Apply Delaunay Triangles**

  From images with detected edges, we extract 20% edge points and apply DT on them. The result of DT is shown in Figure 8.

- **Step B: Data Pre-processing**
  We start by analyzing the triangles from the DT. Due to the density of facial feature regions, we can filter large triangles out for efficiency. From Figure 9, we can see that most triangles have sizes less than 30, so we use 30 as a threshold to filter out large triangles. After cleaning the data, we store the rest triangles into a data structure called GeoDataFrame. This data structure is good at processing variables with coordinates and calculating centroids. With GeoDataFrame, we then apply clustering on those triangles using the centroid of them. Figure 10 is the stored data in GeoDataFrame form.



(a) With mask



(b) Without mask

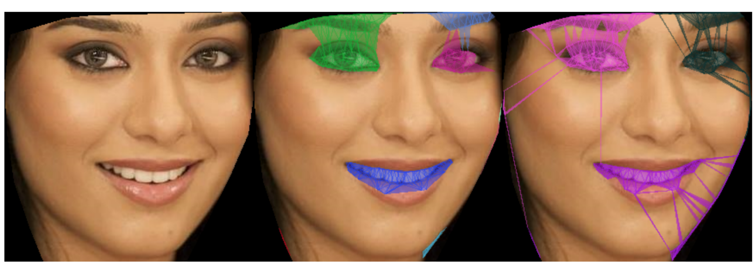
Figure 9: The two are the triangle area distributions of faces with a mask and without a mask

	geometry	x	y
0	POLYGON ((142.000 191.000, 145.000 188.000, 14...	143.666667	189.666667

Figure 10: An example row from the Data Frame, where geometry is the triangles, and (x,y) is the coordinate of the centroid of each triangle [DO11]

- **Step C: Clustering**
  With those triangles, we then apply clustering to them. We mainly use two clustering algorithms: K-means clustering and DBSCAN clustering.
  - K-means clustering is an unsupervised machine learning technique[Sha20]. K-means clustering starts with k random centroids and assigns other points to the nearest centroid to form a cluster. Therefore, K-means clustering is computationally faster than other clustering algorithms. It's very sensitive to the number of clustering K we choose and does not work well with outliers.
  - DBSCAN clustering is a Density-Based Spatial Clustering method, which forms clusters of dense regions of data points and ignores the low-density regions [Sha20]. This algorithm is good at dealing with noisy datasets and identifying outliers. However, it does not work well in sparse datasets and is sensitive to many parameters such as epsilon.

  The following two are the results of clustering on faces with masks and without a mask. From the results, we can see that there are lots of outliers in K-means clustering, which will influence the result when detecting clusters in the mouth region. Therefore, we choose DBSCAN as our clustering algorithm.



(a) With mask result

(b) Without mask result

Figure 11: Figures from left to right: Cropped Face, DBSCAN clustering, and K-means clustering

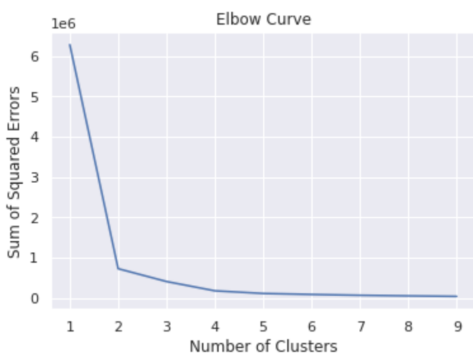
- **Step D: Classification**
  With the clusters and mouth coordinates predicted by shape predictor landmarks, we check if there is overlap between them. If so, there is dense region at the location of the mouth, which means we have found the mouth, and we will classify this face as unmasked, and vice versa.

5.4.2 Tuning Parameters

Within the DT method, there are four main parameters in total, so we have done analysis on these parameters.

- Number of clusters(k) in K-means clustering
  In K-means Clustering, one of the hyper-parameters is the number of clustering k. We use Elbow Method to identify the optimal k value, which goal is to find a small value of k that still has a low SSE(Sum of Squared Errors). From Figure 12(a), we can see that 3 is the optimal value of k. This make sense since from our cropped faces, there are mainly three dense areas which are two eyes and mouth. Therefore, we use k=3.

- Epsilon and Min-samples in DBSCAN
  Epsilon defines the radius of the neighborhood around a point x, and Min-samples is the minimum number of points to form a cluster. In our algorithm, we need a clustering that can drop an appropriate number of outliers, which means the number of outliers is neither too many nor too few. Due to the cost of the algorithm, we can only test the accuracy for a small range of parameters. From Figure 12(b), we can see that eps=5, min_samples=2 or eps=10, min_samples=10 is a good choice. We use eps=5, min_samples=2 in our algorithm.



(a) Tuning for k in K-means

eps	min_samples	n_clusters	outliers
1	2	61	791
5	2	40	37
5	10	17	485
10	2	14	9
10	10	9	83

(b) Tuning for DBSCAN

Figure 12: Tuning parameters and accuracy on 100 images

6 Results and Analysis

We applied both methods above on 1000 images from [Bc20], and compared the results using confusion matrices as listed in Table 3 and Table 4. For the clarity of analysis, we are labeling all unrecognizable images by face extraction simply as masked, as mentioned in Section 5.1.1.

		True diagnosis		
		With mask	Without mask	Total
Prediction	With mask	463	108	571
	Without mask	37	392	429
	Total	500	500	1000
	Accuracy: 85.5%			

Table 3: Confusion matrix of mask detection with density

		Actual		
		With mask	Without mask	Total
Prediction	With mask	500	122	622
	Without mask	0	378	378
	Total	500	500	1000
	Accuracy: 87.8%			

Table 4: Confusion matrix of mask detection with DT Clustering

As is seen in the confusion matrix, the detection by clustering method achieves higher overall accuracy. The DT clustering method is more adaptable on noisy images in that it is not affected by the overall point density of the mouth region in the image, while the density method has a fixed threshold. However, the density method has a much shorter run-time and is more flexible for the detection of other facial decorations, such as glasses or headsets.

There are some common shortcomings of both methods. They do not work well on images with noises on the mouth region, such as patterns or folds on masks and different colors of masks. Another noticeable drawback is that the false positive rate is relatively high with both methods. There are several reasons for such mistakes, including quality of images and wrong face segmentation. The quality of images directly affects the performance of edge detection. On images with offset angle, the mouth is not in the expected area which highly impacts the performance of both methods. On images with bad lighting, the mouth region is too blurry to be detected. Also, The performance of current face segmentation is not satisfying on images with high noises that it detects wrong areas with a similar pattern as a face, such as a poster in the background.

7 Summary and Future Direction

In summary, the implementation of face mask detection is conducted by first using machine learning models in face segmentation, then applying edge detection on cropped images, finally applying face mask detection with DT to detect whether the mouth is present. The face segmentation method gives an accuracy of 79.5% on 1000 images in which 75% of the images with unrecognized faces are masked. The face mask detection methods are the density method and the DT clustering method. The density method classifies images by comparing the density of feature points in different regions of the face and determine if the mouth is present. This method achieves an accuracy of 85.5% on 1000 images. The DT clustering method takes into account two clustering algorithms: K-means clustering and DBSCAN. In both algorithms, we use cluster centroids of triangles obtained by the DT on images of the extracted face and cluster them using these algorithms to generate feature clusters, and then use a facial landmark detector to detect the presence of mouth.

Our approach has achieved satisfying results, with the highest accuracy of 87.8%, which is within the range of accuracy of face mask detection using machine learning methods, although the datasets are different. The accuracy of our method is affected by noisy images such as low lighting quality, different colors, and patterns of masks. This provides future directions such as noise reduction techniques in pre-processing or extensive training on noisy images. Another possible improvement is detecting more than one face in an image.

References

[AI16] Mohamed Alhmdani and Zainab Ibrahim. A prototype hybrid algorithm to detect face oval based on facial expression recognition. *International Journal of Computing and Information Sciences*, 12:253–258, 2016.

[AK00] Franz Aurenhammer and Rolf Klein. Chapter 5 - voronoi diagrams. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 201–290. North-Holland, Amsterdam, 2000.

[AP09] Stylianos Asteriadis and Ioannis Pitas. Facial feature detection using distance vector fields. *Pattern Recognition*, 42:1388–1398, 2009.

[AR01] Saad A. Sirohey and Azriel Rosenfeld. Eye detection in a face image using linear and nonlinear filters. *Pattern Recognition*, 34(7):1367–1391, 2001.

[BAMU03] Md Bhuiyan, V. Ampornaramveth, Shin-yo Muto, and H. Ueno. Face detection and facial feature localization for human-machine interface. *NII Journal*, pages 25–39, 2003.

[Bc20] P. Bhandary and clone2code. Mask classifier. `https://github.com/prajnasb/observations`, 2020.

[BXZ+17] W. Bu, J. Xiao, C. Zhou, M. Yang, and C. Peng. A cascade framework for masked face detection. In *2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 458–462, 2017.

[CDBD20] Amit Chavda, Jason Dsouza, Sumeet Badgujar, and Ankit Damani. Multi-Stage CNN Architecture for Face Mask Detection. *arXiv:2009.07627 [cs, eess]*, 2020. arXiv: 2009.07627.

[CJSW01] H.D. Cheng, X.H. Jiang, Y. Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259–2281, 2001.

[CMM08] Abbas Cheddad, Dzulkifli Mohamad, and Azizah Abd Manaf. Exploiting voronoi diagram properties in face segmentation and feature extraction. *Pattern Recognition*, 41(12):3842–3859, 2008.

[CPSA20] G. Jignesh Chowdary, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. Face Mask Detection using Transfer Learning of InceptionV3. *arXiv:2009.08369 [cs, eess]*, October 2020. arXiv: 2009.08369.

[CTCG95] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[DBUM16] G. Deore, R. Bodhula, V. Udpikar, and V. More. Study of masked face detection approach in video analytics. In *2016 Conference on Advances in Signal Processing (CASP)*, pages 196–200, 2016.

[DO11] Satyan Devadoss and Joseph O'Rourke. *Discrete and Computational Geometry.* Princeton University Press, 2011.

[EI19] M. S. Ejaz and M. R. Islam. Masked face recognition using convolutional neural network. In *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pages 1–6, 2019.

[ERI03] Voronoi diagrams: Applications from archaology to zoology, 2003.

[FFUS16] Sajad Farokhi, Jan Flusser, and Usman Ullah Sheikh. Near infrared face recognition. 21(C):1–17, 2016.

[FY01] Guo Can Feng and Pong C. Yuen. Multi-cues eye detection on gray intensity image. *Pattern Recognition*, 34(5):1033–1046, 2001.

[GE13] Cahit Gürel and Abdulkadir Erden. Face detection algorithm with facial feature extraction for face recognition system. 20th Annual International Conference on Mechatronics and Machine Vision in Practice, M2VIP, 2013.

[GLYL17] S. Ge, J. Li, Q. Ye, and Z. Luo. Detecting masked faces in the wild with lle-cnns. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 426–434, 2017.

[GT17] O'Rourke J. Goodman, J.E. and C.D. Tóth. *Handbook of Discrete and Computational Geometry.* 2017.

[HAF02] J. Haddadnia, M. Ahmadi, and K. Faez. An efficient method for recognition of human faces using higher orders pseudo zernike moment invariant. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pages 330–335, 2002.

[HKK+05] M. Hamouz, J. Kittler, J. K. Kamarainen, P. Paalanen, H. Kalviainen, and J. Matas. Feature-based affine-invariant localization of faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1490–1495, 2005.

[HL15] Chen-Chiung Hsieh and Jun-An Lai. Face mole detection, classification and application. *J. Comput.*, 10:12–23, 2015.

[IMM18] M Iqtait, F S Mohamad, and M Mamat. Feature extraction for face recognition via active shape model (ASM) and active appearance model (AAM). *IOP Conference Series: Materials Science and Engineering*, 332:012032, 2018.

[JAO17] Safa Jida, B. Aksasse, and Mohammed Ouanan. Face segmentation and detection using voronoi diagram and 2d histogram. pages 1–5, 2017.

[JFY20] Mingjie Jiang, Xinqi Fan, and Hong Yan. RetinaMask: A Face Mask detector. *arXiv:2005.03950 [cs]*, 2020. arXiv: 2005.03950.

[Kwo03] B. Kwolek. Face tracking system based on color, stereovision and elliptical shape features. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, pages 21–26, 2003.

[LMTK21a] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, and Nour Eldeen M. Khalifa. Fighting against covid-19: A novel deep learning model based on yolo-v2 with resnet-50 for medical face mask detection. *Sustainable Cities and Society*, 65:102600, 2021.

[LMTK21b] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, and Nour Eldeen M. Khalifa. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement*, 167:108288, 2021.

[Mas15] Modesto Mas. Python image processing libraries performance: Opencv vs scipy vs scikit-image. `https://mmas.github.io/python-image-processing-libraries-performance-opencv-scipy-scikit-image`, 2015.

[MBV19] T. Meenpal, A. Balakrishnan, and A. Verma. Facial mask detection using semantic segmentation. In *2019 4th International Conference on Computing, Communications and Security (ICCCS)*, pages 1–5, 2019.

[MDMM21] S. Meivel, K. Indira Devi, S. Uma Maheswari, and J. Vijaya Menaka. Real time data analysis of face mask detection and social distance measurement using Matlab. *Materials Today: Proceedings*, 2021.

[MDR+13] Khitikun Meethongjan, Mohamad Dzulkifli, Amjad Rehman, Ayman Altameem, and Tanzila Saba. An intelligent fused approach for face recognition. *Journal of Intelligent Systems*, 22(2):197–212, 2013.

[MPC20] Puranjay Mohan, Aditya Jyoti Paul, and Abhay Chirania. A Tiny CNN Architecture for Medical Face Mask Detection for Resource-Constrained Endpoints. *arXiv:2011.14858 [cs, eess]*, 2020. arXiv: 2011.14858.

[NJM+21] Preeti Nagrath, Rachna Jain, Agam Madan, Rohan Arora, Piyush Kataria, and Jude Hemanth. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustainable Cities and Society*, 66:102692, 2021.

[NRMB15] A. Nieto-Rodríguez, M. Mucientes, and V. M. Brea. System for Medical Mask Detection in the Operating Room Through Facial Attributes. In Roberto Paredes, Jaime S. Cardoso, and Xosé M. Pardo, editors, *Pattern Recognition and Image Analysis*, Lecture Notes in Computer Science, pages 138–145, Cham, 2015. Springer International Publishing.

[PKB16] Hemprasad Patil, Ashwin Kothari, and Kishor Bhurchandi. Expression invariant face recognition using semidecimated dwt, patch-ldsmt, feature and score level fusion. *Applied Intelligence*, 44(4):913–930, 2016.

[QL20] Bosheng Qin and Dongxiao Li. Identifying Facemask-wearing Condition Using Image Super-Resolution with Classification Network to Prevent COVID-19. preprint, In Review, 2020.

[RAJ02] Rein-Lien Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.

[RSN20] S. R. Rudraraju, N. K. Suryadevara, and A. Negi. Face mask detection at the fog computing gateway. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 521–524, 2020.

[SAK+21] Sunil Singh, Umang Ahuja, Munish Kumar, Krishan Kumar, and Monika Sachdeva. Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. *Multimedia Tools and Applications*, 2021.

[SAR12] A real-time model for multiple human face tracking from low-resolution surveillance videos. *Procedia Technology*, 6:1004–1010, 2012.

[SBLC08] Ting Shan, Abbas Bigdeli, Brian Lovell, and Shaokang Chen. Robust face recognition technique for a real-time embedded face recognition system. *Pattern Recognition Technologies and Applications: Recent Advances*, 2008.

[SCL06] Jiatao Song, Zheru Chi, and Jilin Liu. A robust eye detection method using combined binary edge and intensity information. *Pattern Recognition*, 39(6):1110–1125, 2006.

[Sha20] Ekta Sharma. K-means vs. dbscan clustering — for beginners, 2020.

[SP96] K. Sobottka and I. Pitas. Face localization and facial feature extraction based on shape and color information. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 483–486 vol.3, 1996.

[SPAS20] S. Susanto, F. A. Putra, R. Analia, and I. K. L. N. Suciningtyas. The face mask detection for preventing the spread of covid-19 at politeknik negeri batam. In *2020 3rd International Conference on Applied Engineering (ICAE)*, pages 1–5, 2020.

[SPB21] K. Suresh, M. Palangappa, and S. Bhuvan. Face mask detection by using optimistic convolutional neural network. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pages 1084–1089, 2021.

[SS21] Sneha Sen and Khushboo Sawant. Face mask detection for covid_19 pandemic using pytorch in deep learning. *IOP Conference Series: Materials Science and Engineering*, 1070(1):012061, 2021.

[ST98] E. Saber and A. Murat Tekalp. Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions. *Pattern Recognit. Lett.*, 19:669–680, 1998.

[STA] WHO coronavirus (COVID-19) dashboard.

[TMS05] F. Tsalakanidou, S. Malassiotis, and M. G. Strintzis. Face localization and authentication using color and depth images. *IEEE Transactions on Image Processing*, 14(2):152–168, 2005.

[TZ02] Z. Tu and Song Zhu. Image segmentation by data-driven markov chain monte carlo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24:657–673, 2002.

[WLS01] Kwok-Wai Wong, Kin-Man Lam, and Wan-Chi Siu. An efficient algorithm for human face detection and facial feature extraction under different conditions. *Pattern Recognition*, 34(10):1993–2004, 2001.

[WWL+21] Zekun Wang, Pengwei Wang, Peter C. Louis, Lee E. Wheless, and Yuankai Huo. WearMask: Fast In-browser Face Mask Detection with Serverless Edge Computing for COVID-19. *arXiv:2101.00784 [cs, eess]*, 2021. arXiv: 2101.00784.

[XL08] Xudong Xie and Kin-Man Lam. Face recognition using elastic local reconstruction based on a single face image. *Pattern Recognition*, 41(1):406–417, 2008.

[XY01] Yi Xiao and Hong Yan. Facial feature location with delaunay triangulation/voronoi diagram calculation. In *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing - Volume 11*, VIP '01, page 103–108, AUS, 2001. Australian Computer Society, Inc.

[XY09] H. Xia and G. Yan. A novel method for eye corner detection based on weighted variance projection function. In *2009 2nd International Congress on Image and Signal Processing*, pages 1–4, 2009.

[Xz20] X-zhangyang. Maskedfacerecognition. `https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset#readme`, 2020.

[YKA02] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24:34 – 58, 2002.

[YO96] Tae-Woong Yoo and Il-Seok Oh. Extraction of face region and features based on chromatic properties of human faces. In Norman Foo and Randy Goebel, editors, *PRICAI'96: Topics in Artificial Intelligence*, pages 637–645. Springer Berlin Heidelberg, 1996.

[ZRF10] Jun Zheng, Geovany Ramirez, and Olac Fuentes. Face detection in low-resolution color images. volume 6111, pages 454–463, 2010.