ACL Digital
an ALTEN group company

Detecting Face Masks

With Machine Learning

WHITEPAPER

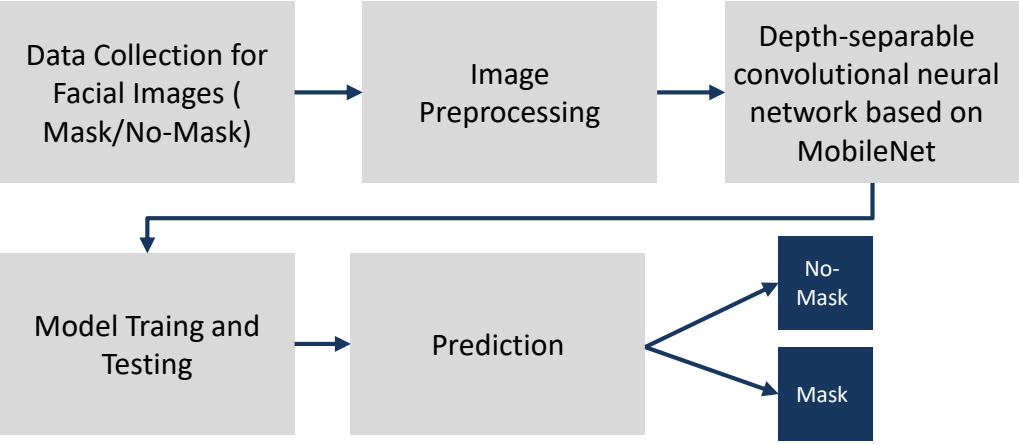
Table of Contents

Introduction

Before the Covid-19 pandemic, there was no concrete proof or evidence of using network masks to lessen respiration infections. Masks are supposed to save the wearer from spreading the viral droplets (supply control). Covid-19 and different respiration infections unfold via inhalation of respiration aerosols produced through coughing, sneezing, talking, or breathing. The virus propagates and migrates down the respiration tract and can result in pneumonia, acute respiration misery syndrome (ARDS), or even death. The ongoing pandemic and the unexpectedly rising variations have made this respiration illness a routine problem. Also, it would be great if people use a face mask as a part of their non-public protection tools and as a public fitness measure. Because of this, the improvement of a gadget that can become aware of humans carrying masks is crucial in the contemporary world.

Scientists have tried to build automated facial mask recognition systems to ensure that using face masks is not unusual in public areas. Following the COVID-19 epidemic, different researchers developed strategies for tracking face masks in not notable place areas. Photo processing algorithms require surveillance structure applications to monitor public areas if you want to ensure that no one's face is visible in crowded places. Deep learning primarily depends on end-to-end procedures for item identification and imagery analytics that have become more prominent. The bulk of the studies has used the convolutional neural community models. There are times wherein cutting-edge face mask detection algorithms cannot reliably become aware of the mask. When there are large gatherings of individuals in a marriage photo or video frame, it's intricate/complicated to become aware of all the faces "with masks and without masks." Women put on half-confronted veils that serve the same motive as a face mask in our nation. However, the cutting-edge techniques are now no longer becoming aware of them as face masks.

Building a more efficient and accurate category method is crucial in implementing facial mask detection strategies in mobile environments. However, numerous deep mastering techniques are expensive and time-consuming in their assessment steps, making them improper for mask detection within the facial photograph paradigm in a mobile environment. The recommended approach uses Depth Wise Separable Convolutions with



MobileNet for mask detection in facial images (3) to overcome the challenges of the present methodology. overDepthwise Separable Convolution (DSC) was first proposed and is now broadly utilized in photograph processing for category tasks.

Core Machine Learning applies an ML algorithm to training data to generate a model. We can use an ML model to make predictions based on new input data. For example, train a set of models to classify photos or discover specific objects within a photograph directly from its pixels.

After creating the model, you need to integrate it into your app and deploy it on the user's device. Our application uses Core Machine Learning APIs and user input data to make predictions and train the model.

Healthcare Detection

Automated skin lesion boundary detection has become a common issue in healthcare. Image processing algorithms already exist and are power-consuming on mobile devices. On the other hand, machine learning algorithms are on the rise, and new SDK frameworks have been developed to use these techniques with improved on-device performance. Since iOS 11.0 OS, the Apple platform provides a Core Machine Learning Interface to use machine learning models. Moreover, conversion tools allow the integration of 3rd party models into iOS applications. In this paper, we would like to introduce an overview of available frameworks for iOS devices and their limitations. We will evaluate the performance and maturity level of Neural Network frameworks for skin lesion boundary detection using only freely available pictures.

Before identifying faces, it is first essential to specify what features of the human face we should use to train a model. Once someone's face detection runs, the part of the face of the image is used for feature extraction. It is critical to choose characteristics

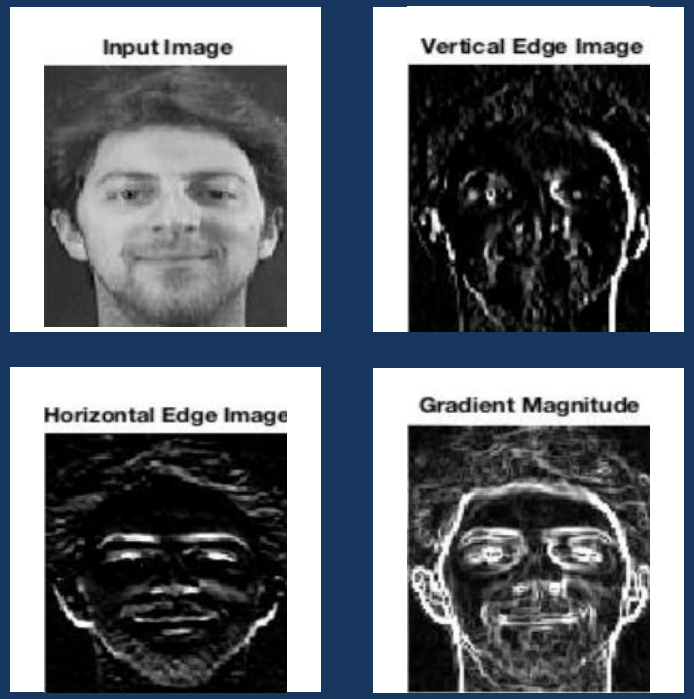
Unique to each face that is utilized to accumulate discriminant data in compact feature vectors. These feature graphics are an essential part of the training allocation of the facial recognition system, and in our work, we recommend using HOG features. As cited earlier, HOG features function well because they store edges and edge direction. High-quality local contrast normalization, coarse spatial binning, and top orientation binning are crucial to good HOG results. Extracting HOG features can be outlined with the following steps:

1. Calculate the histogram of gradients
2. Calculate the gradient of the image
3. Normalize histograms
4. Form the HOG feature vector

-1	0	1

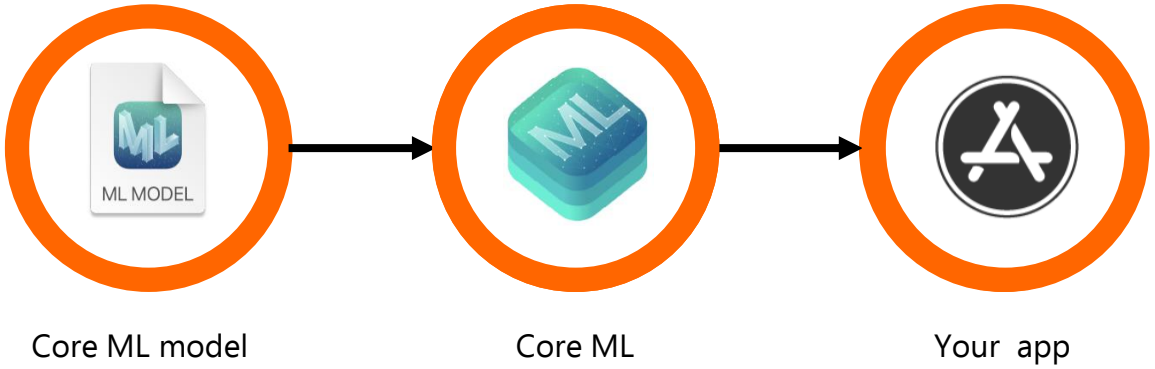
-1
0
1

Figure 3. (Left) Vertical edge detection kernel. (Right) Horizontal edge detector kernel. The first step needs to compute the gradient of the input face image in both the x and y directions using 1x3 and 3x1 edge detector kernels shown in Figure 3.



The horizontal kernel is applied to the input image to create a flat gradient image, while the vertical kernel has a gradient image. The order in which the kernels are used does not matter, and the vertical kernel could be applied first to get the same result.

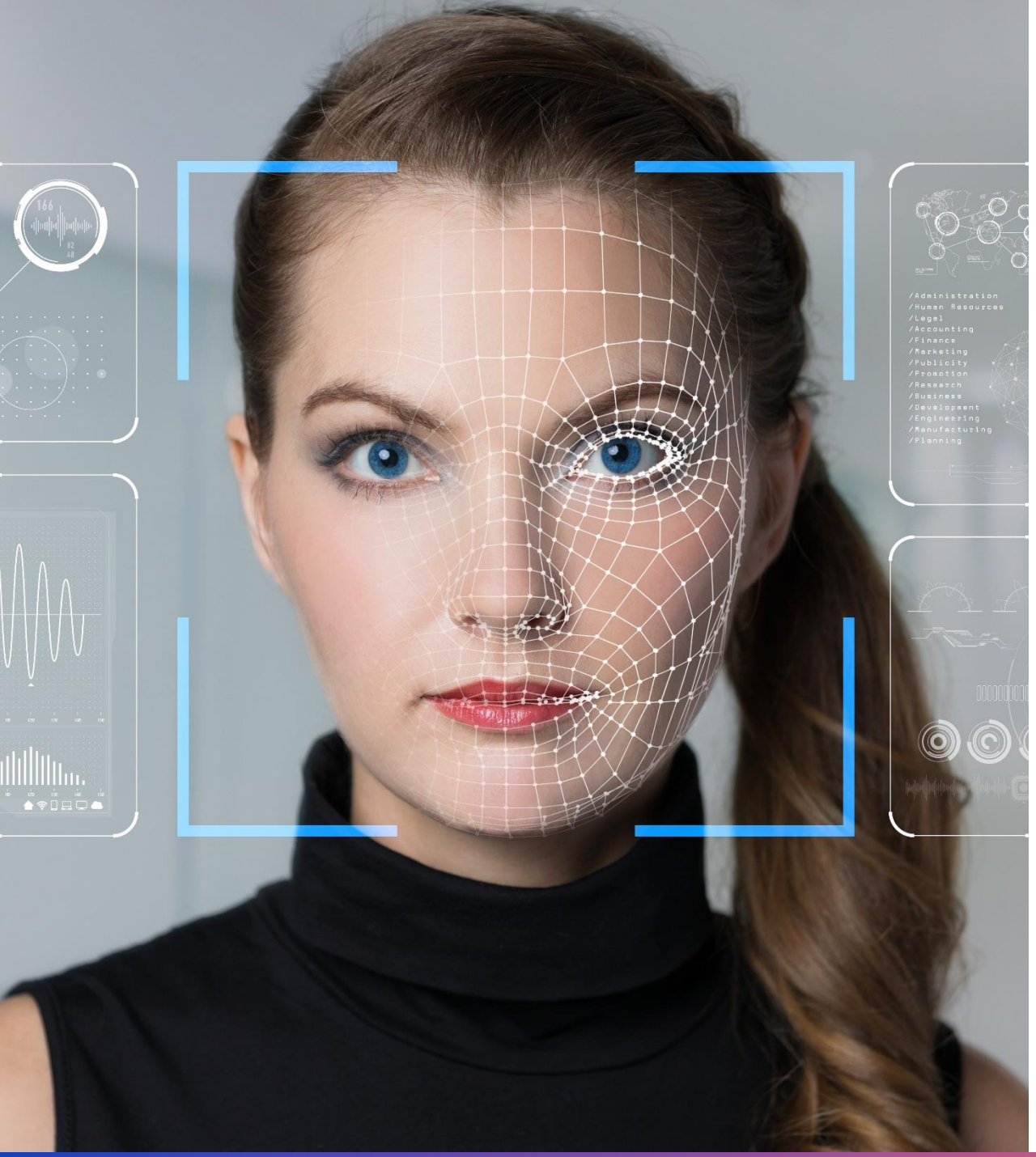
Experiment: Using the Core ML model



Core ML helps support machine learning models, including tree ensembles, neural networks, support vector machines, and generalized linear models. Core ML needs the Core ML model format (models with a .ml model file extension).

The Vision API contains features like:

1. Image Registration
2. Face tracking with ARkit
3. Text and Barcode recognition
4. Native Face detection API
5. Vision authorizes the use of custom Core ML models for all kinds of imaging assignments

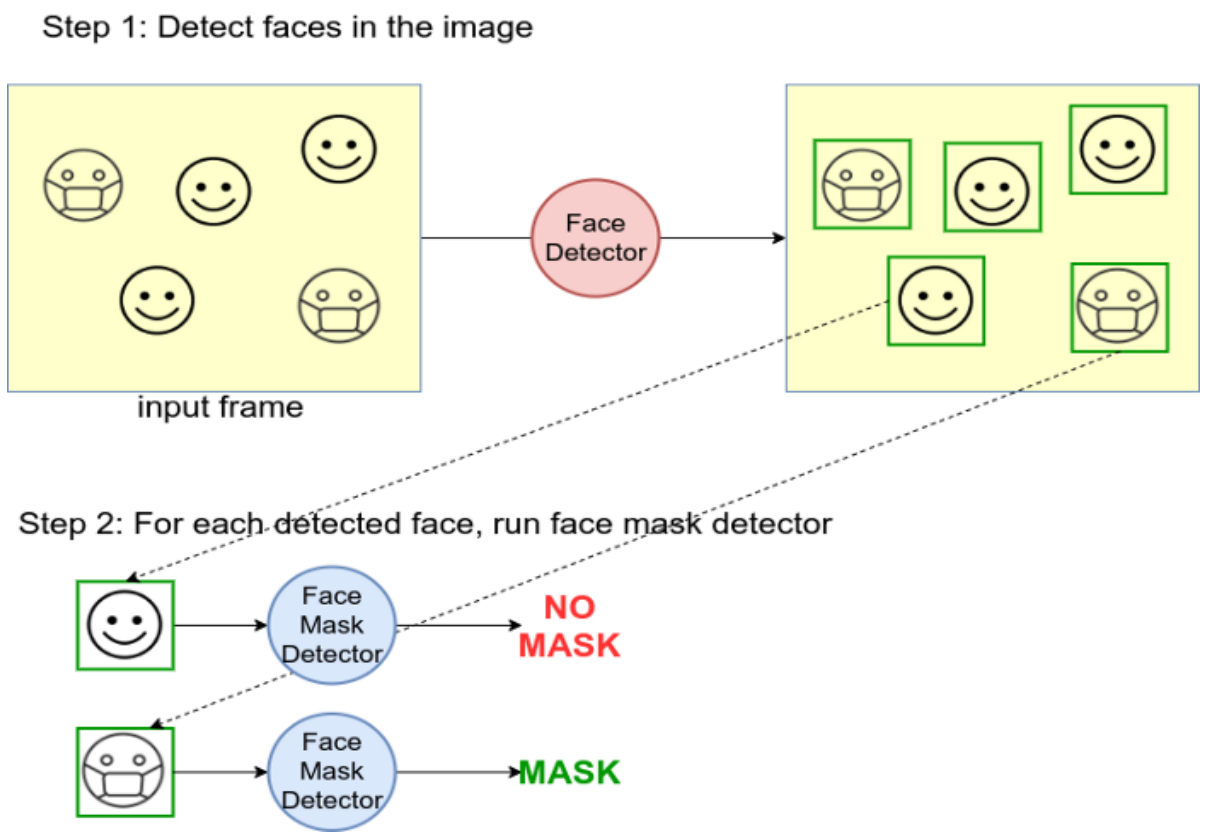


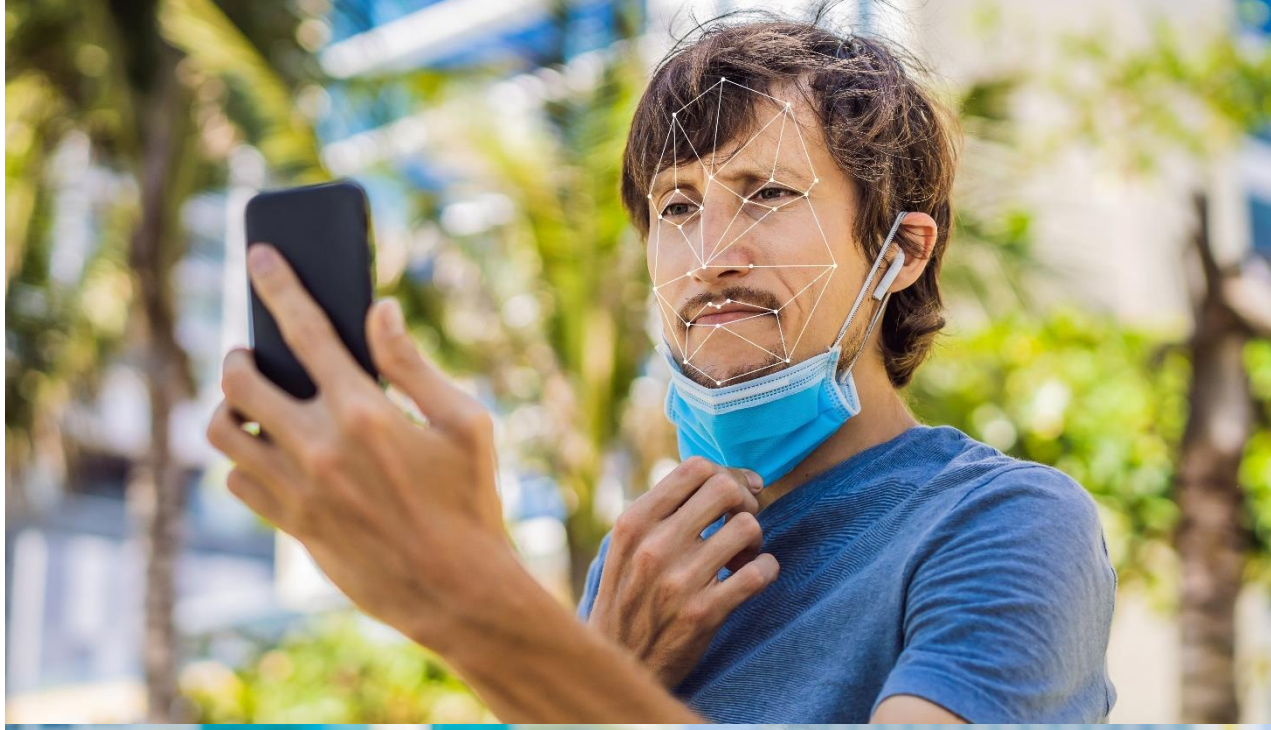
The AT&T faces dataset was the first dataset we used to train and test the facial recognition algorithm was the AT&T faces dataset. The AT&T set comprises 40 unique faces, each with ten different images. The composition of the photographs is a primarily frontal view, with consistent lighting conditions and some varying expressions. When training the SVM model, eight images were necessary for every individual to create a class for each ace. In comparison, the remaining two images were reserved for testing the model. Of the eighty tested images, 73/80 faces were successfully recognized, resulting in an accuracy of 91.25%, which is desirable. When analyzing the failed cases, we noticed that if the face was not facing forward and had a slight rotation to either the left or right, it was more susceptible to incorrect identifications.

ii(x, y) = X x0≤x,y0≤y i(x 0 , y0)

When the query faces the left input into the model, they are usually incorrect matches of the individual from the gallery. The subject is not looking directly for both input photos and requires their face and gaze to the left. There is a clear demonstration of this observation, where the first input face is matched correctly because the subject’s eyes and face are pointing forward. Nonetheless, when the second input face is a similar subject, but they look into the camera sideways, the model fails to identify the face again.

Experiment: Using tflite model





1. Training

Here, the focus is on loading our face mask detection dataset from disk, training a model (using Keras/TensorFlow) on this dataset, and then serializing the face mask detector to disk.

2. Deployment

Once the training for the face mask detector is done, we can then move on to loading the mask detector, performing face detection, and then classifying each face as with_mask or without_mask.

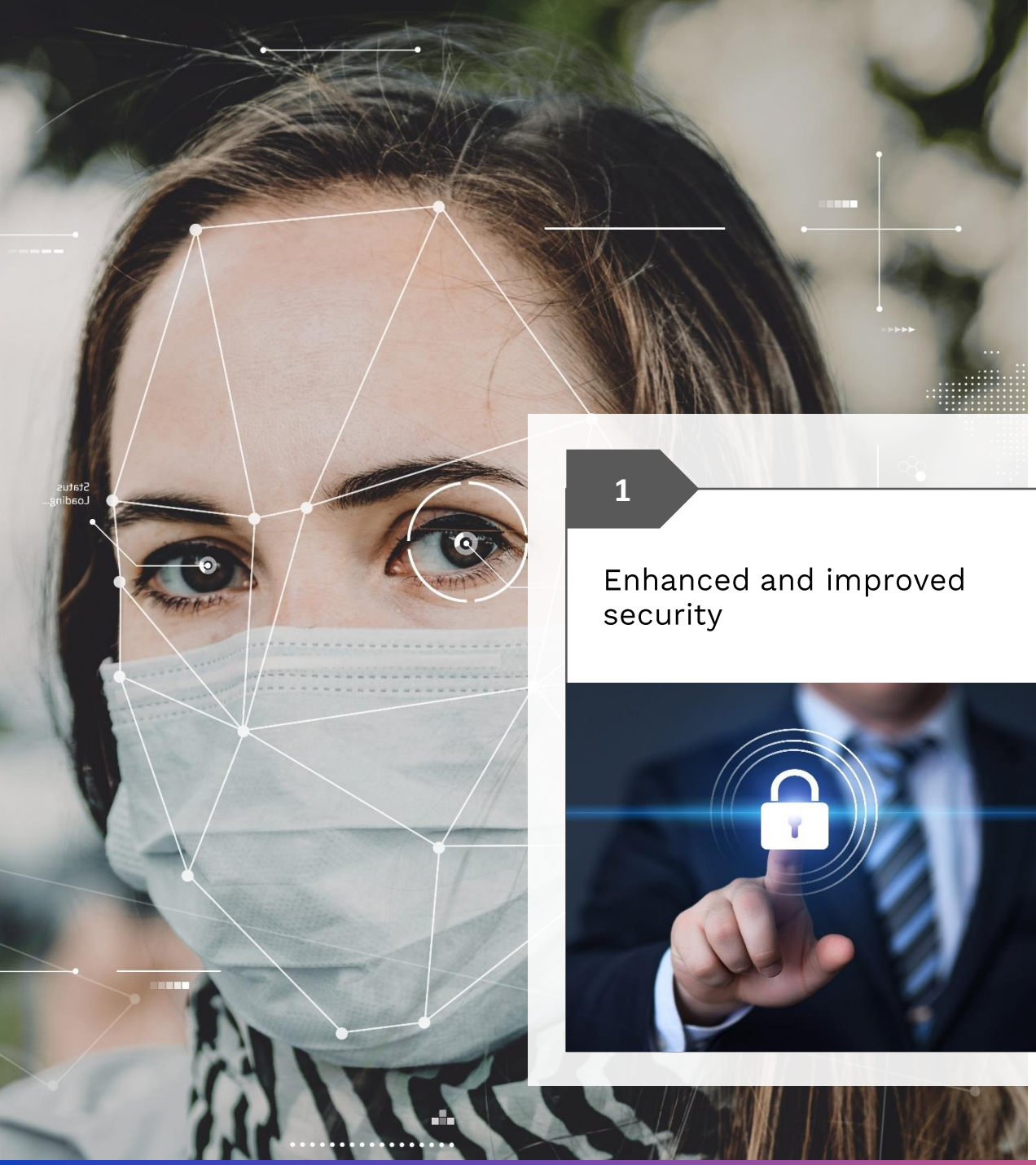
Phase #1 : Train Face Mask Detector

Load face mask dataset → Train face mask with Keras/TensorFlow → Serialize face mask classifier to disk

Phase #2: Apply Face Mask Detector

Leas face mask classifier from disk → Detect faces in image/video stream → Extract each face ROI → Show Results → Apply face mask classifer to each face ROI to determine “mask” of “ no Mask’

It would be ideal to use the MobileNetV2-based model. After the dataset preparation, we will train our model with the dataset. Then we will evaluate the model to check its accuracy of the model. Finally, we will be importing the model as the tflite model. So, it can be unified with our end-user applications (Android, IOS, and Web).



Advantages

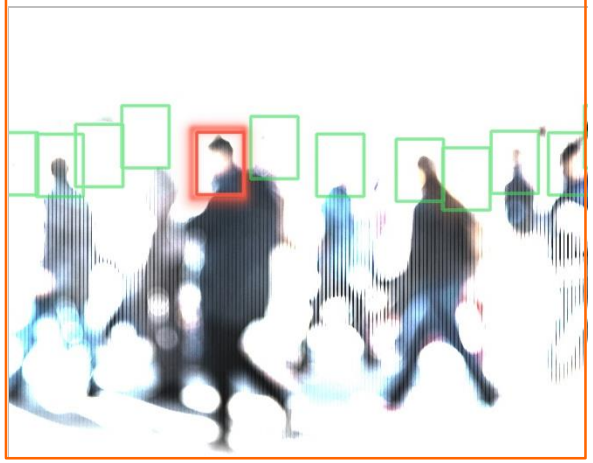
Machine Learning can either be unsupervised learning, hybrid learning, or supervised learning to discover patterns in the dataset. The technique is efficient, supports achieving high accuracy, and considerably improves detection speed. Take a look at some of the advantages below:

1

Enhanced and improved security

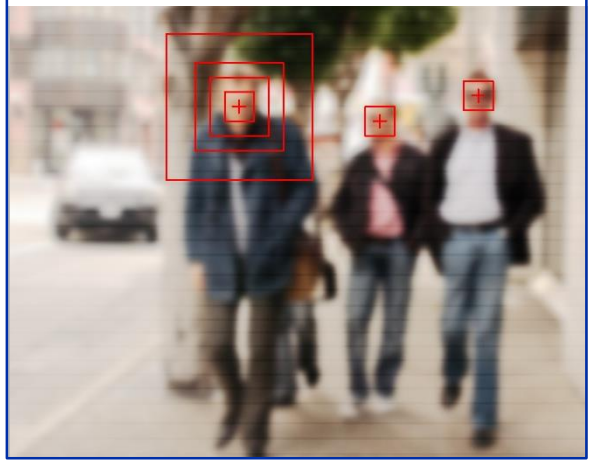
2

Face detection enhances surveillance efforts



3

Helps track down criminals and terrorists



Conclusion

This paper has explained the execution of a facial recognition system using a global method to feature extraction based on a histogram-oriented gradient. Feature vectors for multiple faces were extracted from the Yale and AT&T databases and used to train a binary-tree structure SVM learning model. Running the model on both databases resulted in over 90% accuracy in matching the input face to the valid person from the gallery. We also documented one of the drawbacks of using a global approach to feature extraction of a model trained using a feature vector of the entire face in place of its geometrical makes it less potent to orientation and angle changes. Nevertheless, when the deviation in facial orientation is not substantial, the global approach is still exact and more specific to execute than component-based approaches.

This paper also proposes the MobileNet-based Depthwise Separable Convolution Neural Network (DS-CNN) for mask detection in facial images. We evaluate our findings on the unique convolutional filters on particular datasets. The cautioned machine outperformed cutting-edge classical convolutions in experiments. The cautioned method is likewise contrasted with preceding paintings on a prompted baseline technique. Our findings show that the proposed approach produces the best overall performance throughout several evaluation metrics. The method calls for more significant processing to generate visualizations and, thanks to dataset constraints, cannot discriminate between proper and faulty mask usage. We aim to create face masks reputation datasets with specific masks carrying states or hire zero-shot examination to make the layout pick out defective masks having states.

References

1. Face Recognition using Machine Learning by Arun Alvappillai UCSD

2. Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV

3. Tensor Flow lite : https://www.tensorflow.org/lite

ACL Digital
an ALTEN group company

Digital
Product
Engineering

ACL Digital, an ALTEN Group Company, is a digital product innovation and engineering leader. We help our clients design and build innovative products (AI, Cloud, and Mobile ready), content and commerce-driven platforms, and connected, converged digital experiences for the modern world through a design-led Digital Transformation framework. By integrating our strategic design, engineering, and industry capabilities, we help our clients decode the digital world and accelerate their growth journey.

EMAIL US
business@acldigital.com

TALK TO US
+1 (408) 755 3000

www.acldigital.com