Mask Detection using OpenCV

Saurav Jayakumar
Computing & Software Systems
University of Washington Bothell
WA, USA
sauravsj@uw.edu

Utkarsh Darbari
Computing & Software Systems
University of Washington Bothell
WA, USA
ud21@uw.edu

Harpreet Kour
Computing & Software Systems
University of Washington Bothell
WA, USA
hkour@uw.edu

***Abstract*—During the COVID-19 pandemic, the significance of wearing masks to prevent the spread of the virus has become increasingly apparent. This paper introduces a novel approach to detecting faces with masks using OpenCV and Haar/LBP classifiers. The aim of this study is to develop a reliable system that can automatically identify and classify faces as either wearing masks or not, thereby aiding in the implementation of mask-wearing protocols in public settings. To evaluate the proposed mask detection system, a diverse dataset comprising images captured in various scenarios and lighting conditions is utilized. Performance metrics, including accuracy, precision, recall, and F1-score, are employed to assess the system's effectiveness in accurately identifying individuals who are wearing masks. Our findings indicate that when the face and eyes are correctly detected, the system can effectively determine whether or not a person in an image is wearing a mask.**

***Index Terms*—Mask wearing, face detection, OpenCV, Haar classifiers, LBP classifiers, machine learning, COVID-19, surveillance systems.**

I. Introduction

The outbreak of COVID-19 had a profound impact on global healthcare systems, causing widespread disruption. With the virus primarily affecting the respiratory system, wearing masks emerged as a crucial measure to contain its spread. Numerous studies [4] have demonstrated the effectiveness of mask-wearing in preventing the transmission of the virus. Consequently, mask detection has garnered significant attention in light of the COVID-19 pandemic and the urgent need to enforce mask-wearing protocols across diverse settings. Computer vision techniques, particularly utilizing the OpenCV library, have proven to be a valuable approach for mask detection.

OpenCV is a programming library that offers a range of functions designed for real-time computer vision, machine learning, and image processing tasks. It empowers developers to manipulate and extract information from real-time sources effectively. In the context of mask detection, OpenCV plays a pivotal role in identifying faces, objects, and handwriting, serving as a vital tool for distinguishing between individuals wearing masks and those without.

This paper presents a mask detection system utilizing the HAAR cascade classifier. Building upon existing work that utilizes Local Binary Patterns (LBP) features for training classifiers, we seek to enhance the accuracy of our system by training it on non-standard methods of mask-wearing. Additionally, we endeavor to minimize the influence of lighting and other environmental factors on the classification process.

II. Dataset

For our study, we utilized the Face Mask Detection Dataset [3] available on Kaggle. This dataset comprises 7,553 RGB images organized into two folders: "with_mask" and "without_mask." The images are appropriately labeled to indicate whether a mask is present or absent. Specifically, there are 3,725 images depicting faces wearing masks and 3,828 images featuring faces without masks.

However, since the dataset primarily consists of "in-the-wild" images, it is not directly compatible with the haar and LBP cascade classifiers. To address this issue, we handpicked a subset of 86 images from both the "with_mask" and "without_mask" categories to evaluate our program. These selected images encompass various alignment scenarios, including instances where multiple faces are present in a single image. To facilitate access to ground truth data, we modified the image labels to follow the format [with/without]_mask_[image id]_count_[number of faces].jpg.

III. Methods

A. Image pre-processing

The quality of an image can have a profound impact on the effectiveness of face and eye detection methods. Factors such as lighting, contrast, and shadows significantly influence the detection process. When images are captured under diverse conditions, it becomes challenging for any detection method to provide accurate results without appropriate image pre-processing techniques. To address this issue, Zhang et al. [1] proposed a set of pre-processing steps, including grayscale image conversion, histogram equalization, and Gaussian filtering, to enhance the performance of face and eye detection cascade classifiers. In our study, we have incorporated these pre-processing steps, as illustrated in Figure 1, prior to applying the face and eye detection algorithms on the images.

B. Face detection

After pre-processing the image, our algorithm employs both Haar and LBP cascade classifiers to detect faces and draw bounding boxes around them. We prioritize the Haar cascade classifier as our primary detector due to its higher reliability, as confirmed by our experiments in most cases. However, in

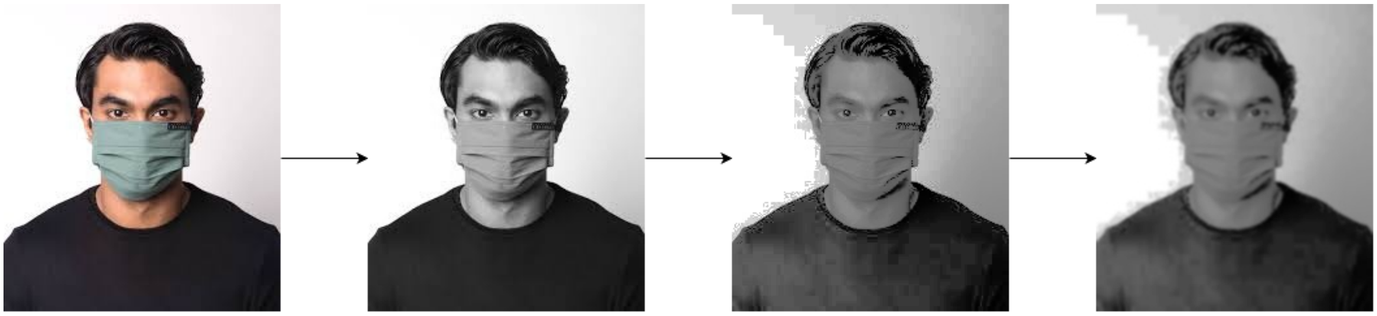

Fig. 1. Pre-processing steps: The input image is first converted to grayscale, then its histogram is equalized, and finally the image is smoothened using Gaussian filtering

situations where the Haar classifier fails to detect a face, we resort to the LBP cascade classifier, which has demonstrated favorable results in specific scenarios. Once the bounding boxes are obtained, we extract the detected faces by cropping them out from the original image, as depicted in Figure 2. It is important to note that if our classifiers are unable to detect a face, the program terminates as it cannot proceed further.

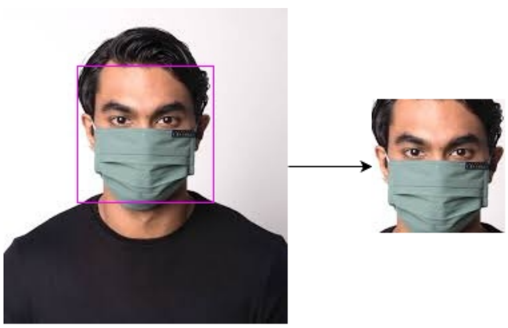

Fig. 2. Face detection steps: The pre-processed image is used for face detection using haar and LBP cascade classifiers (shown in the original image for clarity)

C. Eye detection

After obtaining the cropped faces, our subsequent objective is to detect the eyes within them. To accomplish this, we once again employ Haar cascade classifiers. We utilize various types of eye detectors, including general 2-eye detectors, right-eye detectors, left-eye detectors, and eyewear detectors. Our intention is to generate bounding boxes for the eyes, as illustrated in Figure 4, regardless of challenging conditions such as non-aligned faces or faces wearing glasses. Even if only a single eye is detected, we can leverage this information for partial mask detection.

D. Skin area segmentation

To accurately calculate the skin area on different regions of a person's face, it is essential to segment the skin from the background, hair, and other elements. In order to achieve this, we leverage the Cr component of the image after converting it to the YCrCb color space, following the approach proposed by Zhang et al. [1]. For the segmentation process, we employ Otsu thresholding, which is an adaptive thresholding method capable of automatically determining the optimal threshold to separate the foreground from the background. By utilizing the Cr component, we mitigate the impact of illumination and contrast in the image, as the Y component is separated. This segmentation process, depicted in Figure 3, is conducted on the cropped face images and serves as a precursor to further processing steps.

E. Oronasal region selection

By leveraging the bounding boxes for the eye region, Junjie Xiao et al. [2] proposed a method to calculate the bounding box for the mouth region as well.

Consider the bounding box around the left eye, with coordinates (x_1, y_1) for the top-left corner and (x_1, y_1+h_1) for the bottom-left corner. It is observed that the coordinates (x_1, y_1+h_1) and (x_1, y_1+3*h_1) represent the top-left and bottom-left corners of the bounding box around the oronasal region, respectively. This process can be repeated for the right eye, yielding the top-right and bottom-right coordinates. Combining these coordinates provides us with the bounding box for the oronasal region, as illustrated in Figure 4.

Fig. 4. Oronasal region selection: Using the eye bounding box in pink, the oronasal region area is calculated

F. Comparison of the skin area of the eyes with the oronasal region

Drawing on the proposal by Zhang et al. [1], we establish a criterion where the skin area surrounding the eyes should exceed the skin area within the oronasal region by a factor of 1.2. This guideline enables us to make insightful inferences regarding the adherence to proper mask usage.

Fig. 3. Skin area segmentation: The image is first converted to the YCrCb color space and then the Cr component is used for Otsu thresholding

When the skin area ratio exceeds the threshold, indicating a substantial difference between the areas, it strongly suggests that the individual is wearing a mask correctly. This outcome aligns with the expected behavior when a mask adequately covers the nose and mouth region. On the contrary, if the calculated ratio falls short of the prescribed threshold, it indicates a potential issue with mask-wearing compliance. This may indicate either the absence of a mask or an incorrect positioning of the mask on the face.

By leveraging this comparative analysis of skin areas, our approach provides a valuable metric for determining the effectiveness of mask usage based on visual cues derived from the eyes and oronasal region.

IV. Results

We conducted rigorous testing of our program using the carefully selected subset from the complete dataset. For each image, we manually assessed whether the program successfully detected the correct faces and eyes prior to implementing the actual mask detection algorithm. By analyzing the results of each individual image, we computed summary statistics presented in Table I and Table II. These tables provide a comprehensive overview of the program's performance and its ability to accurately detect faces, eyes, and masks.

The evaluation of mask and non-mask detection results relies on the successful detection of faces and eyes, rather than being dependent on ground truth labels. This is due to the inherent dependency of the mask detection algorithm on the accurate detection of faces and eyes. Without the correct identification of these facial features, the algorithm is unable to perform the subsequent mask detection process, resulting in a failure to produce reliable results. Hence, the effectiveness of the mask detection algorithm is contingent upon the accurate detection of faces and eyes in the input images.

TABLE II
Metrics

Metric	Value
Accuracy	92.7%
Precision	89.8%
Recall	96.3%
F1-Score	92.9%

Due to the utilization of off-the-shelf cascade classifiers for face and eye detection, the performance of these classifiers is compromised when confronted with non-aligned faces in in-the-wild images. Consequently, these classifiers often yield suboptimal results. As illustrated in Figure 5, several sample images exemplify the challenges encountered during face and eye detection, including incorrect or missing detections.

However, once the face and eye regions have been successfully detected, our model excels in accurately determining whether individuals in the image are wearing a mask or not. Despite the limitations in initial detection, our subsequent mask detection algorithm demonstrates robust performance, providing reliable outcomes.

Fig. 5. Images that failed during the face and eye detection stages

V. Conclusion

This research paper presents the development of a mask detection algorithm that incorporates face and eye detection, skin thresholding, and oronasal region selection. Experimental evaluation using a dataset revealed that the model achieved an accuracy of 92.7% and an F1-score of 92.9%. The performance of the algorithm is influenced by the use of off-the-shelf cascade classifiers for face and eye detection, which demonstrate satisfactory performance in certain scenarios but exhibit limitations in others. Additionally, image pre-processing techniques such as grayscale conversion, histogram equalization, and Gaussian filtering are essential for normalizing the images and enhancing overall performance.

Based on the findings, it is evident that further enhancements can be made to improve the algorithm's accuracy. Introducing more advanced face and eye detectors, as well as employing techniques for precise skin thresholding and oronasal region selection, are potential avenues for refinement.

TABLE I
DETECTION RESULTS

Images	**Face Detection**	**Additional Incorrect Face Detection**	**Eye Detection**	**Additional Incorrect Eye Detection**	**Mask Detection**	**Non-Mask Detection**
Masked Faces	98.8%	0%	97.7%	2.3%	96.3%	3.7%
Non-Masked Faces	98.9%	5.8%	88.5%	6.9%	11%	89%
Combined	98.85%	2.9%	90.1%	4.6%	N/A	N/A

These improvements hold the potential to yield more precise and reliable results.

REFERENCES

[1] Zhang, Shuxiang, et al. "Detection of Face Wearing Mask Based on AdaBoost and YCrCb." 2022 4th International Conference on Electrical Engineering and Control Technologies (CEECT), IEEE, 2022, pp. 349–53. DOI.org (Crossref), https://doi.org/10.1109/CEECT55960.2022.10030579.

[2] JJ. Xiao, Masked Face Detection and Standard Wearing Mask Recognition Based on YOLOv3 and YCrCb[J] Software, 2020, 41(7):164-169.

[3] Face Mask Detection Dataset. https://www.kaggle.com/datasets/omkargurav/face-mask-dataset. Accessed 31 May 2023.

[4] Upendra, H.S., Suman, S., Vishnu, S.S. and Dharani, J., 2021. Real-Time Face Mask Detection using OpenCV and Deep Learning. In ICET (Vol. 2021).