International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Face Mask Recognition-based Door Lock Mechanism Using HaarCascade, Mediapipe, and Arduino

Yash Verma Aryan Gupta Akshit Rathore Abhijeet Mohan Mishra
Vanshikha Gupta
(Project Mentor)
Inderprastha Engineering College
Ghaziabad 201010

***Abstract* — On 12 January 2020 Chinese authorities shared the sequence of a novel coronavirus termed severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). Since then COVID-19 pandemic had rapidly affected our day-to-day life disrupting world trade and movements. Wearing face masks is one of the preventive measures recommended by health authorities to reduce the spread of respiratory droplets containing viruses, including the SARS-CoV-2 virus responsible for COVID-19. Therefore, Many regions and organizations have implemented regulations or guidelines mandating the use of face masks in specific settings. This Python-based Machine Learning project provides an automated way to monitor whether individuals in a given environment are wearing masks or not. Pre-trained Haar Cascade files are used to detect the facial features and provide a signal to the Arduino controller to lock/unlock the door.**

***Keywords* — Python, Face Mask, Door Lock, Haar Cascade, Covid-19**

I. Introduction

The first cases of a novel coronavirus were reported in Wuhan, Hubei province, China, in December 2019. The virus was initially linked to a seafood market in the city. On January 7, 2020, Chinese authorities identified the new virus as a novel coronavirus and named it SARS-CoV-2. According to the World Health Organization (WHO)'s Weekly operational update on COVID-19 - 7 August 2020, COVID had globally infected over 18 million people causing over 0.7 million deaths [1]. Symptoms of COVID-19 can vary, but mild cases often experience fever, cough, and fatigue. Moderate cases may have difficulty breathing or mild pneumonia. While severe cases may have severe pneumonia, other organ failure & possible death [3]. Currently circulating variants of concern (VOCs) as of 15 March 2023 are XBB.1.5, XBB.1.16, EG.5, BA.2.86, and JN.1 [4].

To prevent respiratory viral diseases, including COVID-19, wearing a clinical mask is very necessary. Masks act as a barrier that helps prevent respiratory droplets, which may contain viruses or bacteria, from entering the air and reaching others. This is particularly important in situations where individuals may be asymptomatic or presymptomatic and unknowingly carrying a contagious virus. Wearing masks aligns with public health guidelines and measures, reinforcing collective efforts to control and prevent the spread of infectious diseases. It is important to make sure your mask covers your nose, mouth, and chin [5]. During pandemics or outbreaks, widespread mask usage can help control the spread of infectious diseases, protecting individuals and communities. Therefore, accurately detecting a face mask is a crucial task in present global society.

The process of detecting a face mask involves, detecting the face and then checking if the nose and chin are covered properly by the mask or not. Detecting faces in images or video frames can be challenging due to various factors. Faces can have a wide range of appearances, including different hairstyles, facial hair, accessories, and expressions. Moreover, variations in lighting conditions, such as shadows, highlights, and changes in ambient light, can affect the appearance of faces. This paper presents a simplified solution to the above problem by using basic Machine Learning (ML) packages pre-trained haarcascade files, and a mediapipe library to detect major facial features and use Arduino microcontroller and servo motors to implement door lock functionality.

II. Related Work

The face detection is done by detecting some of the facial attributes in the frame. According to [6], the neural network-based and view-based approaches require a large number of images containing faces and a large number of images having no faces in them and these algorithms are primarily designed to locate frontal faces in gray-scale images. Face detection is a challenging task because faces are not rigid and they change in size, shape, color, etc. Face detection becomes a more challenging task when a given image is not clear and occluded by any other thing and not proper lightning, not facing the camera, etc. Authors in [7] The absence of large datasets of masked faces and the lack of facial cues from the masked regions can hinder the performance of face detection algorithms.

According to the authors in [2], the utilization of the locally linear embedding (LLE) algorithm and trained dictionaries for addressing the challenges associated with masked faces and recovering facial expressions is a comprehensive strategy for handling occlusions and recovering facial expressions.

The main challenge of the project is to detect the face from the image correctly and then identify if it has a mask on it or not. To perform door lock and unlock using Arduino and servo motors.

III. INCORPORATED PACKAGES

A. OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products.[8]. This library was used to get the video feed frame by frame from an external camera source.

B. Haar Cascade

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning-based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.[9]

C. MediaPipe

MediaPipe is an open-source framework developed by Google that provides a comprehensive and flexible platform for building machine learning (ML) pipelines for perceptual computing applications. It is used in the project for the detection of face and hands.

D. Arduino Uno microcontroller

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header, and a reset button.[10]

IV. THE PROPOSED METHOD

The proposed method combines multiple detection techniques (face, hand, and nose) along with hardware control to create a system that ensures individuals are wearing masks before granting them access through the door.

A. Imports and Setup:

- The project starts by importing necessary libraries such as OpenCV, MediaPipe, and the controller module. These libraries provide tools for image processing, face detection, hand detection, and controlling hardware components like door lock and LED lights.
- Parameters such as font styles, colors, and durations are set up for displaying messages and controlling the behavior of the system.

B. Face Detection Initialization:

- The script initializes the MediaPipe Face Detection module with a minimum detection confidence threshold. This module is responsible for identifying and localizing faces within the frame.

C. Hand Detection Initialization:

- Similarly, the script initializes the MediaPipe Hands module for hand detection. This module detects and tracks the landmarks of the user's hands, which can be used to infer gestures or actions.

D. Video Capture:

- The script captures video from the default camera feed using OpenCV. This video feed will be processed frame by frame to detect faces, hands, and noses.

E. Detection Loop:

- Within a continuous loop:
  - ○ Frame Processing: Each frame captured from the video feed is processed.
  - ○ Gray Scaling: The frame is converted to grayscale for nose detection since the cascade classifier operates on grayscale images.
  - ○ Face Detection: The MediaPipe Face Detection module is applied to the frame to detect faces. It identifies the bounding boxes of detected faces and their confidence scores.
  - ○ Hand Detection: The MediaPipe Hands module is applied to detect hands. It identifies landmarks representing different parts of the hand (e.g., fingertips, palm) and draws connections between them to form a hand skeleton.
  - ○ Nose Detection: The script uses a cascade classifier to detect noses in the grayscale frame. This step aims to identify whether the person's face is wearing a mask or not.
  - ○ Mask Detection: Based on the presence of faces, hands, and noses, the script determines whether a mask is worn or not. If no face is detected, the door is assumed to be open, and the user is allowed entry. If a face is detected but no nose is found (implying no mask), the door is kept closed, and a

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

message prompts the user to wear a mask. If a face and nose are detected, it's assumed the person is wearing a mask, and the door is opened.

○ Visualization: Detected landmarks, bounding boxes, and text messages indicating mask status are overlaid onto the frame to provide visual feedback to the user.

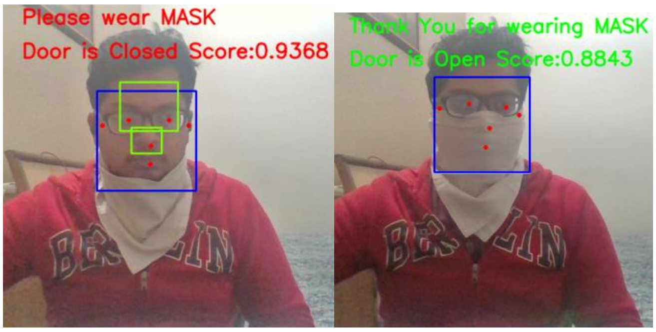
○ Control Signals: Depending on the detection results, control signals are sent to automate the door and change LED colors accordingly.

F. Termination:

- The loop continues until the 'Esc' key is pressed, at which point the video feed is released, and all windows are closed.

V. RESULT AND ANALYSIS

The pre-trained MediaPipe reaches up to an accuracy of 93% when the face is not covered and reaches up to 88% when the face is covered.



The face detection is stopped if the hand is detected in the frame. This is because it ensures that the person has not covered his/her mouth with the hand and also it helps in maintaining the required distance from the camera.

VI. Conclusions

The developed system represents a valuable tool for organizations and establishments seeking to enforce mask-wearing policies in various settings, including public spaces, workplaces, and healthcare facilities. By automating the monitoring process, the system reduces the need for manual intervention and ensures consistent adherence to mask guidelines, thereby contributing to efforts to control the spread of infectious diseases and safeguard public health. Further research and development in this area could focus on refining the detection algorithms, optimizing hardware integration, and exploring additional features to enhance the functionality and usability of the system.

References

[1] WHO Weekly Operational Report on COVID-19 7 August 2020. Available: https://www.who.int/publications/m/item/weekly-update-on-covid-19---7-august-2020.

[2] A. Das, M. Wasif Ansari, and R. Basak, "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV," 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, 2020, pp. 1-5, doi: 10.1109/INDICON49873.2020.9342585.

[3] WHO Report on COVID-19 Symptoms 2 August 2023. Available: https://www.who.int/westernpacific/emergencies/covid-19/information/asymptomatic-covid-19.

[4] WHO Report on Active Varients of COVID-19 15 March 2023. >>>>Available: https://www.who.int/activities/tracking-SARS-CoV-2-variants.

[5] WHO report on Advice for the public: Coronavirus disease (COVID-19) 18, March 2023. Avilable: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public

[6] Rein-Lien Hsu, M. Abdel-Mottaleb and A. K. Jain, "Face detection in color images," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 696-706, May 2002, doi: 10.1109/34.1000242.

[7] S. Ge, J. Li, Q. Ye and Z. Luo, “Detecting Masked Faces in the Wild with LLE-CNNs,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 426-434, doi: 10.1109/CVPR.2017.53.

[8] OpenCV documentation Available: https://opencv.org/about/

[9] OpenCV Haar cascade Classifier Documentation Available: https://docs.opencv.org/4.x/db/d28/tutorial_cascade_classifier.html

[10] Arduino Uno microcontroller store Available: https://store.arduino.cc/products/arduino-uno-rev3.