Face Mask Detection

Group Members:

Yazeed Marwan Mwafi: 202020186.

Laith Issam Jaber: 202010517.

Alaa Kamal Al-Hmoud: 202020305.

Hassan Mohammad Abzakh: 202220565.

Contents

1. Overview:

The Face Mask Detection system, now with IoT, uses a camera to check if people wear masks. The computer looks at a video and decides if someone has a mask on. This information is sent over the Internet to a small computer chip (Arduino). The chip controls lights and sounds. If someone has a mask, it turns on a green light and plays a nice sound. If not, it turns on a red light and makes a warning sound. It's like a smart system that helps make sure people wear masks in public places.

2. Components:

- Hardware Components:
  1. Camera.
  2. Computer: The main machine that processes the pictures and videos.
  3. Arduino Board
  4. LEDs.
  5. Buzzer.
  6. Breadboard.
- Software Components:
  1. Python Script: A set of instructions for the computer to look at pictures and decide if someone is wearing a mask or not.
  2. Face Detection Model: An algorithm that helps the computer recognize faces in pictures.
  3. Face Mask Detection Model: An algorithm that helps the computer decide if a person in a picture is wearing a mask.
  4. Arduino Code: Instructions for Arduino to turn on lights and sounds.
- Communication Components:
  1. Serial Communication: A way for the big computer to talk to the small computer.

3. System Architecture:

- Setup and Environment Preparation:

  Install necessary libraries and dependencies for Python and Arduino development.
  Set up the Python environment for computer vision, including TensorFlow, Keras, OpenCV, and imutils.
  Install the Arduino IDE.

- Hardware Setup:

  Connect the Arduino board to the computer via USB.
  Connect the Arduino components.

- Face Detection Model Integration:

  Test the face detection model separately with sample images.
  Integrate the face detection model into the Python script and ensure it works with real-time video input.

- Face Mask Detection Model Integration:

  Test the face mask detection model separately with sample images.
  Integrate the face mask detection model into the Python script and ensure it works with detected faces.

- Arduino Hardware Testing:

  Upload the provided Arduino code to the Arduino board.
  Test the Arduino board's response to different serial commands (e.g., "mask," "no_mask") using the Arduino IDE's serial monitor.

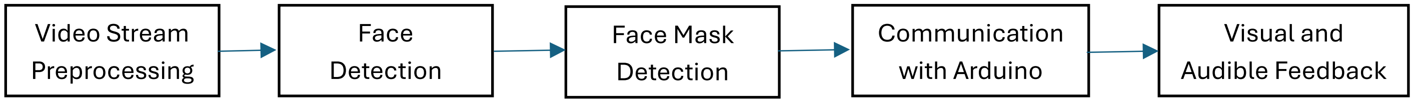
- Integration of Python and Arduino:

  Establish serial communication between the Python script and Arduino board.
  Modify the Python script to send appropriate commands to the Arduino based on the face mask detection results.

- Real-time Testing and Calibration:

  Run the integrated system in real-time using the video stream.
  Calibrate the system parameters, such as confidence thresholds for face and mask detection.

4. Methodology:


Video Stream Preprocessing
Face Detection
Face Mask Detection
Communication with Arduino
Visual and Audible Feedback

5. Problem Statement:

The challenge at hand is to develop an effective and automated Face Mask Detection system utilizing computer vision and hardware integration. In the context of a public health imperative, the goal is to create a system that can accurately identify individuals in real-time video streams and determine whether they are wearing face masks or not. The absence of widespread mask adherence in public places poses a risk to community health, and this system aims to address the issue by providing immediate visual and audible feedback. The system should seamlessly integrate with an Arduino board to control LEDs and a buzzer, enhancing the user interface and encouraging mask compliance. Additionally, the system should be designed for practical deployment, ensuring reliability and ease of use in various environments.

6. Troubleshooting Guide:

Installation and Environment

Issue: Libraries not installed or configured.

Solution: Double-check installation steps. Ensure required libraries like OpenCV, TensorFlow, and imutils are installed. Verify Python and Arduino IDE setups.

Model Loading:

Issue: Face detection or mask detection model not loading.

Solution: Confirm model paths in the script. Check file existence and permissions. Ensure the correct models for face detection and mask detection are used.

Hardware Connectivity:

Issue: LEDs and buzzer not responding.

Solution: Verify Arduino connections. Check if LEDs and buzzer are functional. Ensure the Arduino board is recognized by the system.

Serial Communication:

Issue: No communication between Python and Arduino.

Solution: Confirm correct COM port in the Python script. Ensure the Arduino sketch is uploaded to the board. Check for proper baud rates (e.g., 9600).

Model Accuracy:

Issue: Incorrect face or mask detection results.

Solution: Check the model's training data. Re-train models with diverse datasets for better generalization. Adjust confidence thresholds in the Python script.

7. Limitations:

- Single-Face Focus:
  The system is tailored for single-face detection, potentially encountering challenges when multiple faces are present in a frame.

- Standard Mask Assumption:
  The system is designed for conventional mask types and may encounter difficulties with non-standard or unconventional masks.

- Computational Speed Variability:
  Processing speed is contingent on the host computer's capabilities, with potential delays on less powerful machines.

- Limited User Interaction:
  User interaction is currently minimal, predominantly consisting of visual and audible feedback. Enhancements in this aspect are constrained in the current implementation.

- Arduino Hardware Dependency:
  The system is dependent on the proper functioning of an Arduino board; any issues with this hardware may impact visual and audible feedback provision.

8. Conclusion:

And there you have it – our Face Mask Detection project! We've built a smart system that watches out for people wearing masks in public places. With the help of special computer vision tricks and a small computer called Arduino, we can see faces, check for masks, and give instant feedback with lights and sounds.

Of course, our system isn't perfect. It might make mistakes, especially in tricky lighting or with unusual masks. But we've done our best to make it helpful in encouraging everyone to wear masks for public safety.

We're excited about the possibilities and potential improvements. If you have questions or ideas, feel free to get in touch! Thank you.