

# Face Recognition Door Access System

## Project Overview

This system provides secure door access control using facial recognition technology. It consists of a Python-based face recognition application that communicates with an Arduino microcontroller to control a door lock mechanism when authorized faces are detected.

## System Architecture

[USB Camera] → [Python Face Recognition] → [Serial Communication] → [Arduino Nano] → [Relay Module] → [Door Lock]

## Hardware Requirements

### Essential Components

- 1. Arduino Nano microcontroller
- 2. USB Webcam (minimum 720p resolution recommended)
- 3. 5V Relay Module (Active Low type)
- 4. Door Lock Mechanism (12V DC electric strike recommended)
- 5. Jumper Wires
- 6. USB Cable for Arduino
- 7. Power Supply for Arduino (7-12V DC)

### Optional Components

- 1. Status LED
- 2. 220Ω Resistor (for status LED)
- 3. External Power Supply for door lock

## Hardware Connections

### Arduino Nano Pinout

Arduino Pin	Connection	Notes
Digital 6	Relay IN Pin	Controls relay activation

Arduino Pin	Connection	Notes
Digital 13	Status LED (+)	Optional system status
GND	Relay GND, LED GND	Common ground
5V	Relay VCC	Relay power

#### Relay Wiring

Relay Terminal	Connection
COM	Door Lock Positive
NO (Normally Open)	Door Lock Negative
NC (Normally Closed)	Not Connected

**Important:** The door lock mechanism may require a separate power supply. Connect the relay between the power supply and the lock.

#### Software Requirements

Python Libraries

`opencv-python==4.5.5.64`

`numpy==1.21.5`

`pyserial==3.5`

Install required libraries using:

`pip install opencv-python numpy pyserial`

Arduino IDE

- Version 1.8.x or newer
- Required board: Arduino Nano

- Required library: None (uses standard libraries)

## Installation Guide

### Step 1: Hardware Setup

1. Connect Arduino Nano to computer via USB
2. Wire the relay module to Arduino Digital Pin 6
3. Connect door lock mechanism to relay terminals
4. Connect USB webcam to computer

### Step 2: Software Setup

1. Clone or download the project files
2. Upload the Arduino sketch to your Nano
3. Install required Python libraries
4. Create the directory for face storage: C:\Users\highes\Downloads\NewMove\faces

### Step 3: Face Registration

1. Run `face_capture.py` to register authorized users:

`python face_capture.py`

2. Position user in front of camera
3. Press 'c' to capture face (ensure face is clearly visible)
4. Press 'q' to quit when done

### Step 4: System Operation

1. Run the face recognition system:

`python face_recognition_door.py`

2. The system will:
  - Initialize camera and load known faces
  - Establish connection with Arduino
  - Begin monitoring for faces
  - Unlock door for 5 seconds when authorized face is detected

## File Structure

project-root/

|

├── face\_capture.py      # Script to register new faces

├── face\_recognition\_door.py # Main recognition application

├── door\_control.ino      # Arduino sketch for door control

├── known\_faces/          # Directory for stored faces (auto-created)

└── README.md            # This document

## Arduino Sketch Details

### Key Features

- Non-blocking design using millis() for timing
- 5-second relay activation period
- Debouncing to prevent rapid activation
- Active low relay control
- Serial communication at 9600 baud

## Uploading to Arduino

1. Open Arduino IDE
2. Load door\_control.ino sketch
3. Select Board: Arduino Nano
4. Select correct COM port
5. Click Upload

### Configuration Options

#### Python Script Configuration

Modify these variables in face\_recognition\_door.py:

python

# Serial port configuration

**arduino\_port = 'COM5' # Change to your Arduino's COM port**

# Recognition parameters

confidence\_threshold = 0.7 # Adjust for stricter/looser recognition

recognition\_interval = 1 # Seconds between recognition attempts

cooldown\_period = 5 # Seconds after detection before next attempt

**# Face storage location**

**faces\_dir = r'C:\Users\high\Downloads\NewMove\faces'**

Arduino Sketch Configuration

Modify these constants in the Arduino sketch:

cpp

const unsigned long relayActiveDuration = 5000; // Unlock duration in milliseconds

const unsigned long debounceDelay = 5000; // Minimum time between commands

Operation Instructions

Normal Operation

1. Power on the system
2. Run face\_recognition\_door.py
3. authorized users approach the camera
4. System unlocks door for 5 seconds upon recognition
5. Door automatically locks after timeout

**Adding New Users**

1. Run face\_capture.py
2. Position new user in good lighting
3. Press 'c' to capture face
4. Ensure multiple angles for better recognition

5. Restart recognition script to load new faces

## Troubleshooting

1. **Camera not detected:** Check USB connection and webcam drivers
2. **Arduino not connecting:** Verify COM port and check serial connection
3. **Poor recognition:** Ensure good lighting and retrain with better images
4. **Relay not activating:** Check wiring and power supply

## Safety Considerations

1. **Emergency Override:** Always maintain a physical override option
2. **Power Failure:** Implement battery backup for critical applications
3. **Fire Safety:** Ensure system doesn't prevent emergency egress
4. **Physical Security:** Protect hardware from tampering

## Maintenance

### Regular Checks

1. Verify camera focus and cleanliness
2. Test relay operation monthly
3. Update face database as needed
4. Check wiring connections periodically

### Software Updates

1. Monitor for OpenCV and library updates
2. Backup face database before updates
3. Test system thoroughly after updates

## Performance Notes

1. **Recognition Accuracy:** Depends on lighting conditions and image quality
2. **Processing Requirements:** Requires minimum 4GB RAM for smooth operation
3. **Response Time:** Typically 1-2 seconds from detection to unlock
4. **Multiple Faces:** System processes one face at a time

## Troubleshooting Guide

### Common Issues and Solutions

Issue	Possible Cause	Solution
No face detection	Poor lighting	Improve lighting conditions
False recognitions	Low threshold	Increase confidence_threshold
Serial communication error	Wrong COM port	Check Device Manager for correct port
Relay not activating	Wiring issue	Check relay connections and power
System lag	High CPU usage	Close other applications
Debug Mode		