Lock System with Facial Recognition and Fingerprint Verification

- Final Year Project – BSc (Hons) in Computing
- Presented by: Fiachra Mc Eniff

Ollscoil Teicneolaíochta an Atlantaigh

Atlantic Technological University

Introduction & Motivation

TRADITIONAL ACCESS METHODS (KEYCARDS, PASSWORDS) ARE NO LONGER RELIABLE.

BIOMETRIC SYSTEMS OFFER HIGHER ACCURACY AND ARE CONTACTLESS.

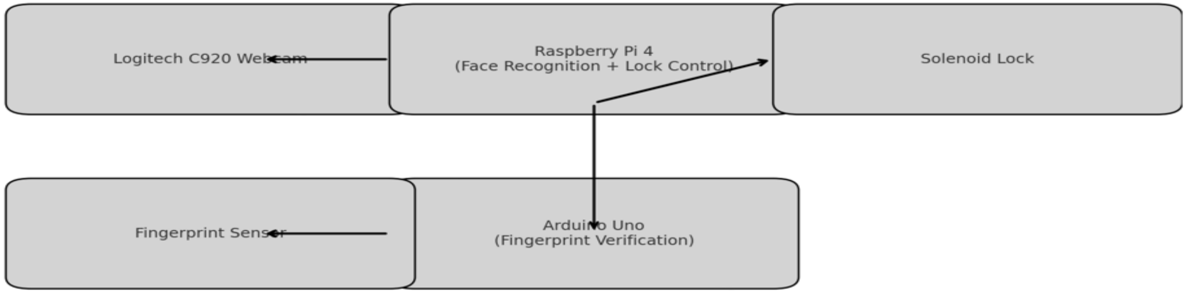
COVID-19 EMPHASIZED THE NEED FOR TOUCHLESS SECURITY.

IOT ENABLES REAL-TIME MONITORING AND SMARTER SYSTEMS.

Project Aim & Research Question

Aim: Build a secure locking system using facial and fingerprint verification.

Research Question: Can a locking system be effectively implemented using facial recognition and fingerprint verification to ensure only authorized individuals can access a premises?



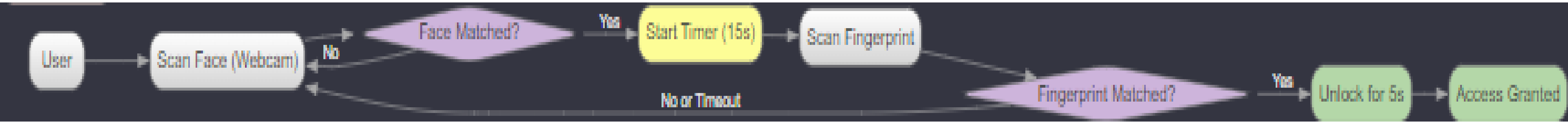
System Overview

RASPBERRY PI PROCESSES FACIAL RECOGNITION AND CONTROLS THE LOCK.

ARDUINO UNO MANAGES FINGERPRINT SCANNING AND SIGNALS PI.

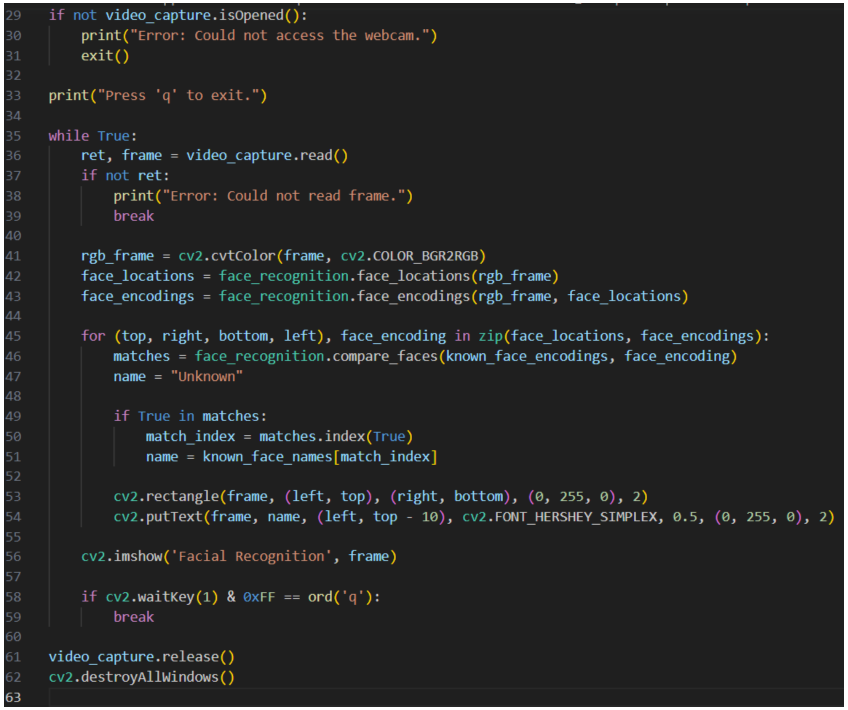
SOLENOID LOCK ACTIVATED ONLY AFTER BOTH BIOMETRICS SUCCEED.

USES OPENCV, FACE_RECOGNITION, AND GPIO FOR CONTROL.



Facial Recognition

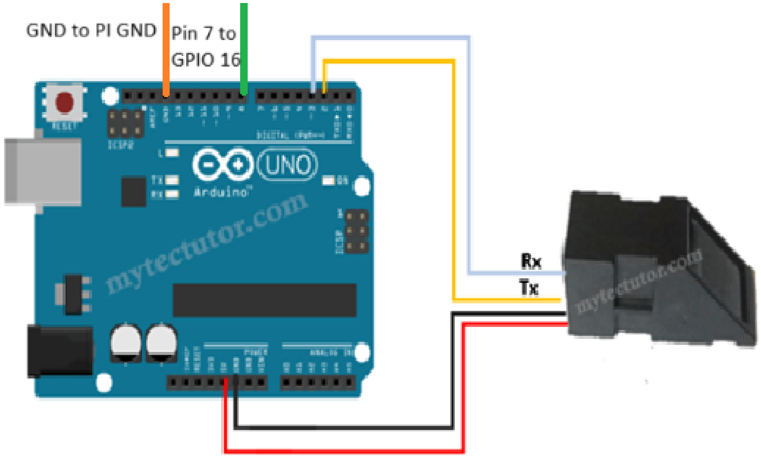
- Built using OpenCV and face_recognition.
- Processes live camera feed using CNN embeddings.
- Tested for lighting, angles, and spoofing resistance.
- Performs well in under 2 seconds on Raspberry Pi.



```python
if not video_capture.isOpened():
    print("Error: Could not access the webcam.")
    exit()

print("Press 'q' to exit.")

while True:
    ret, frame = video_capture.read()
    if not ret:
        print("Error: Could not read frame.")
        break

    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    face_locations = face_recognition.face_locations(rgb_frame)
    face_encodings = face_recognition.face_encodings(rgb_frame, face_locations)

    for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
        matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
        name = "Unknown"

        if True in matches:
            match_index = matches.index(True)
            name = known_face_names[match_index]

        cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
        cv2.putText(frame, name, (left, top - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

    cv2.imshow('Facial Recognition', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

video_capture.release()
cv2.destroyAllWindows()
```

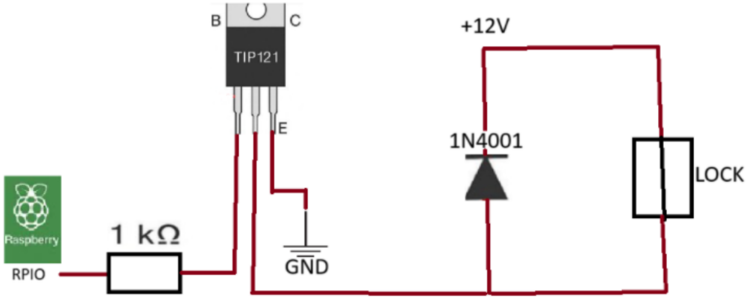
Fingerprint Verification

- Optical fingerprint scanner connected to Arduino.
- Fingerprint ID is stored, and success triggers GPIO signal to Pi.
- Added timeout and error handling for reliability.



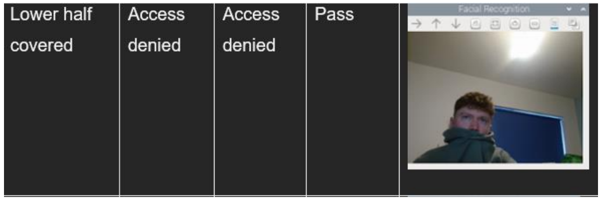
Hardware Implementation

- Raspberry Pi 4 with Raspbian OS (SSD-booted for stability).
- Logitech C920 webcam for facial input.
- TIP121 transistor circuit used to control solenoid.
- Safety features: 5-second unlock duration.

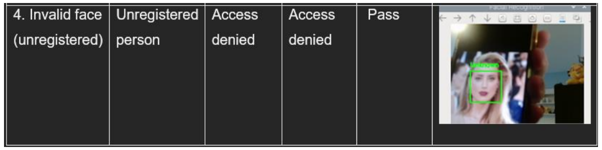


Testing & Results

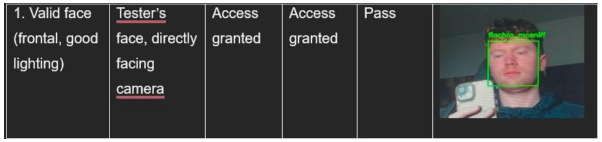
- Facial Recognition: Passed all test cases.
- Fingerprint Verification: Worked reliably before hardware failure.
- Combined system unlocks in <4 seconds total.



Lower half covered	Access denied	Access denied	Pass	



4. Invalid face (unregistered)	Unregistered person	Access denied	Access denied	Pass



1. Valid face (frontal, good lighting)	Tester's face, directly facing camera	Access granted	Access granted	Pass

Challenges & Limitations

- Hardware Issues: Fingerprint module failed late in project.
- Power Stability: Lock heating required unlock time adjustments.
- Environmental Sensitivity: Lighting affects face detection.

😢

R.I.P 2025-2025

Conclusion

Conclusion & Reflection

- Successfully set up a multi-factor biometric access using facial recognition and fingerprint verification (before the fingerprint module broke near the end) .
- Gained valuable experience learning Python, which I wasn’t very comfortable with before this project.
- Learned how to set up and work a Raspberry Pi and Arduino.
- Built a strong understanding of electronics, circuit wiring, and GPIO communication.

Thanks for listening!

☺