Facial Recognition

Baby sowjanya Damarla -C00314717
Ayomide Otinwa-C00313536

EXECUTIVE SUMMARY

Content:

Brief overview of the project objectives and outcomes:

- Developed a facial recognition system using Raspberry Pi.
- Real-time recognition with a user-friendly interface.
- Addressed challenges in deploying AI on low-powered devices.
- **Techniques Used:** AI models, edge computing.
- **Applications:** Personalized security, attendance systems.

INTRODUCTION

•**Background**:

•History of facial recognition:

•From PCA (1990) to modern deep learning models like FaceNet and DeepFace.

•Growth in practical applications: security, automation, personalization.

•**Problem Statement**:

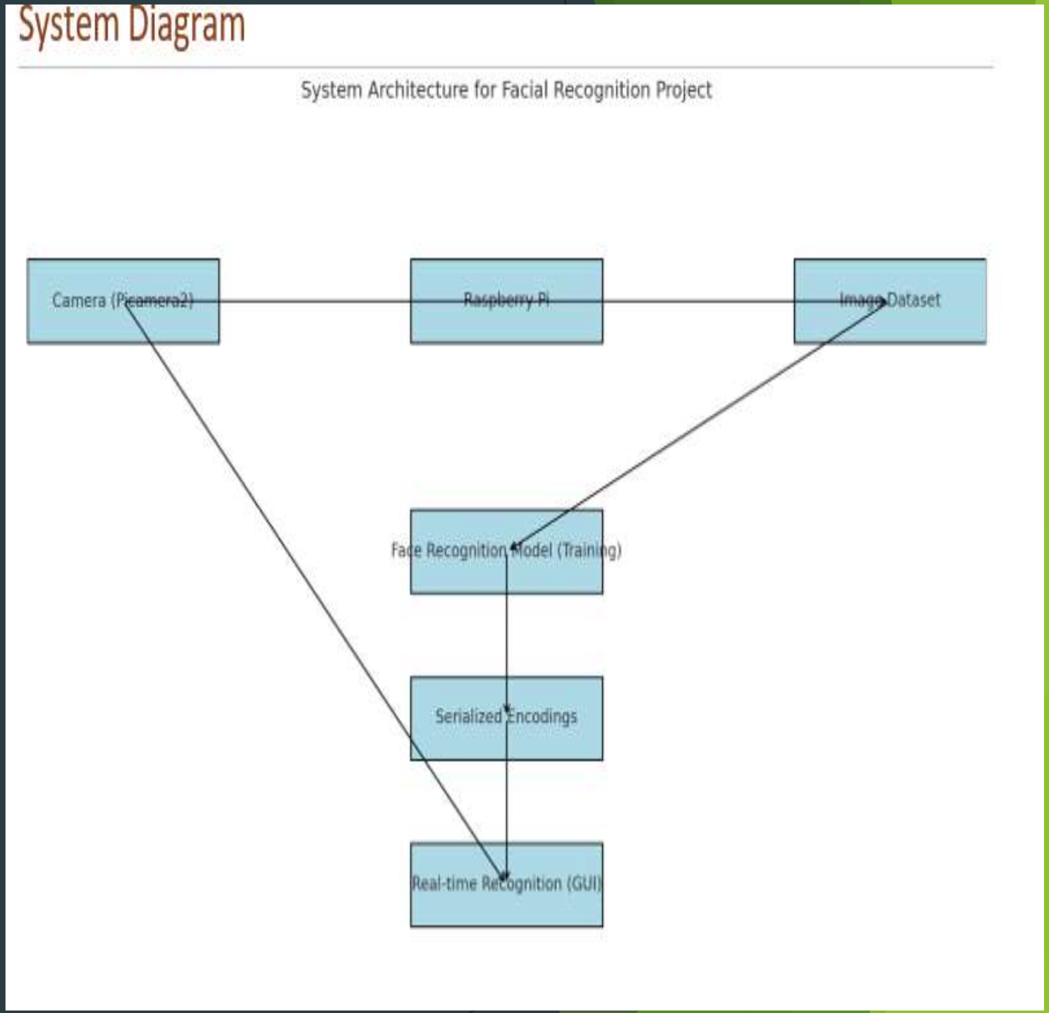
•Challenge of implementing real-time facial recognition on low-power devices like Raspberry Pi.

•**Objectives**:

•Develop a cost-effective system with real-time capabilities.

•Address compatibility and environmental challenges.

SYSTEM ARCHITECTURE



SYSTEM ARCHITECTURE

- **Key Components**
- Raspberry Pi 4 Model B: Processes data locally.
- Camera (Raspberry Pi Noir V2): Captures input images.
- AI Libraries: face_recognition and OpenCV.
- GUI: Tkinter for user interaction.

- **Workflow**
- Steps: Capture → Process → Recognize → Display.

AI and Edge Computing Techniques

- **AI Models**

- Algorithm: Histogram of Oriented Gradients (HOG) for face detection.
- Embedding: Pre-trained DNN generates facial encodings.

- **Edge Benefits**

- **Latency**: Faster recognition by processing on-device.
- **Privacy**: No data sent to external servers.

- **Optimization Strategies**

- Downscaling images to 320x240 for faster processing.
- Using lightweight libraries like OpenCV Headless.

Implementation

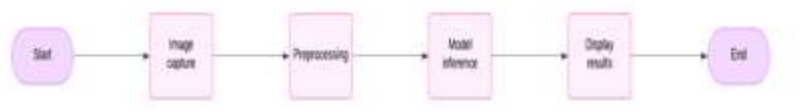


Fig. 2. Flowchart Diagram of the project

Development Phases

Planning and Research: Selecting tools and hardware.

Data Collection: Creating single-user datasets.

Model Training: Generating facial encodings.

Integration: Combining recognition with Tkinter GUI.

Optimization: Reducing lag and improving usability.

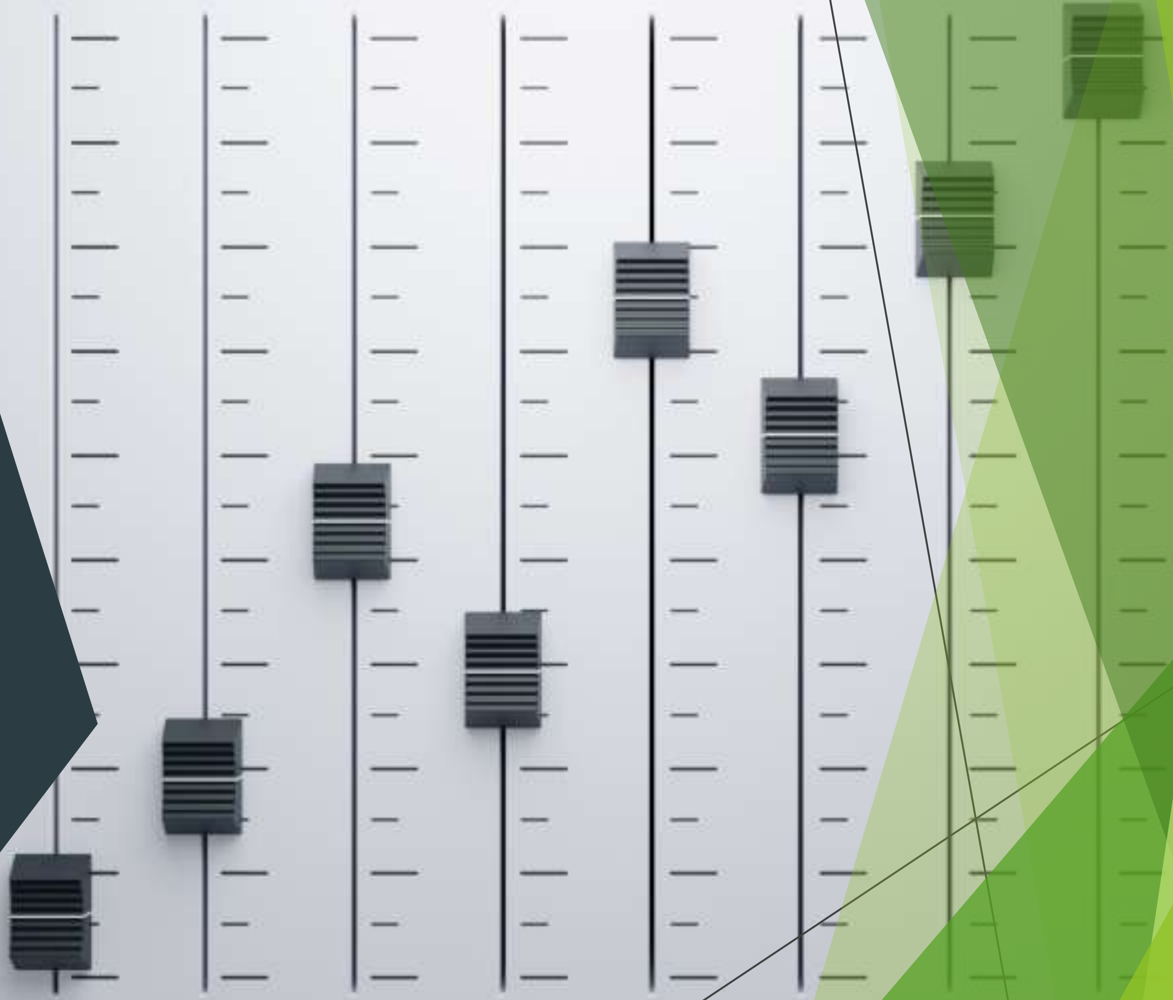
Tools

Python, OpenCV, face_recognition, Picamera2.

User Interaction and UI/UX

•**Interface Features**:
•Live camera feed.
•Real-time feedback: "Welcome, [Name]" or "Unknown.“

•**Accessibility**
•Large fonts, high-contrast colors, minimalistic layout.



Challenges and Solutions

•**Key Challenges**

•Compatibility issues between libraries.
•Managing Python environments and dependencies.

•**Solutions**

•Manual installation of compatible packages.
•Carefully tracking changes to system configurations.

•**Lessons Learned**

•Importance of dependency management and environment isolation.
•Consideration of tools like Docker for future projects.

Ethical and Legal Considerations

- **Ethical Issues**

- Bias in AI models: Gender and demographic disparities.
- Privacy concerns in facial recognition.

- **Legal Compliance**

- GDPR alignment: Local data processing, no storage of PII.

- **Future Enhancements**

- Data anonymization and encryption.

Results

- Achievements:
- Successfully recognized faces in real-time.
- Created a lightweight, deployable system on Raspberry Pi.

- Limitations
- Focused on single-user datasets.
- No integration of advanced deep learning models.

Future Work

ENHANCEMENTS

IMPLEMENTING ADVANCED AI MODELS LIKE TENSORFLOW LITE.

EXPANDING TO MULTI-USER DATASETS.

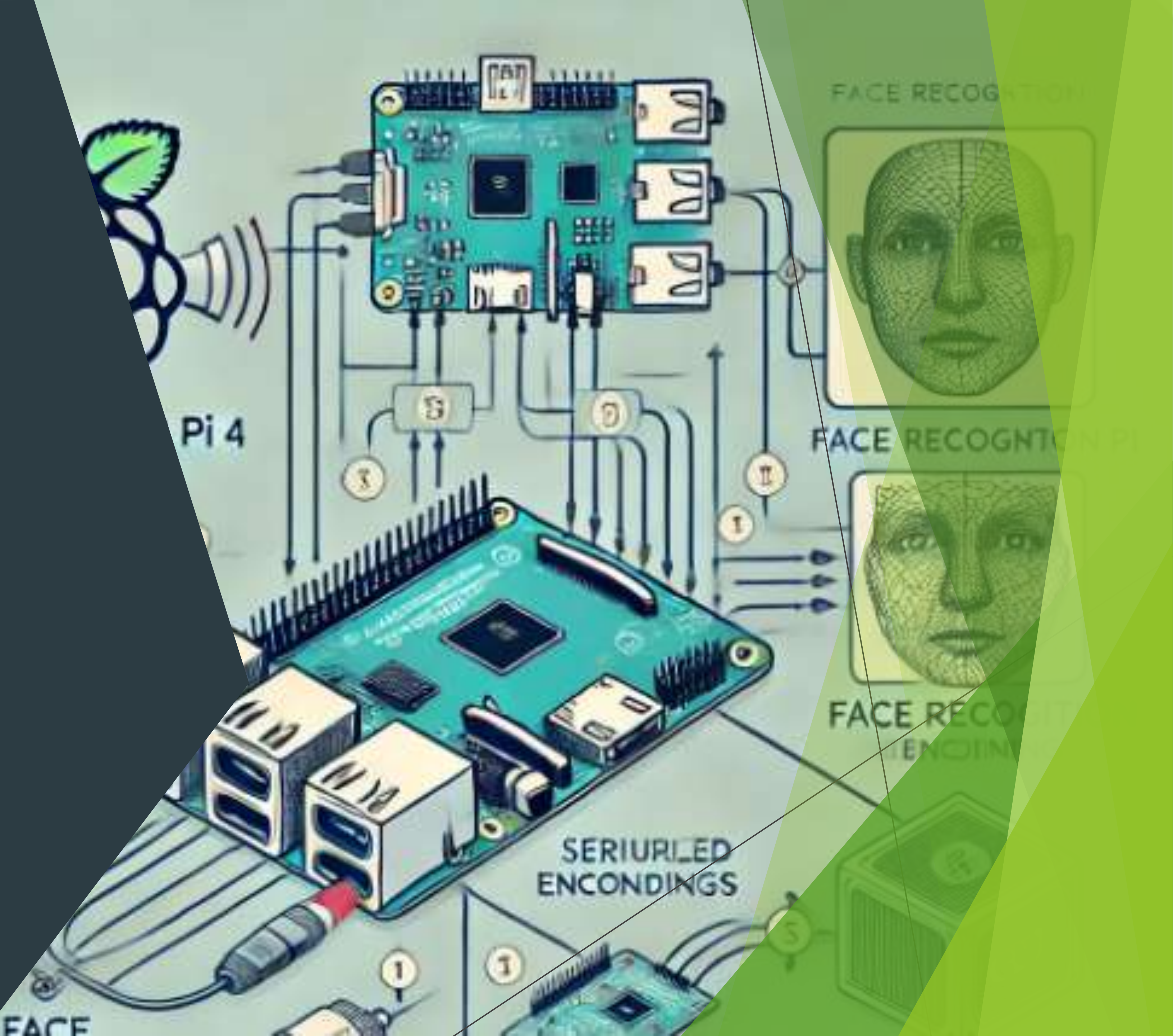
ADDING VOICE OR FINGERPRINT INTEGRATION FOR MULTI-MODAL RECOGNITION.

APPLICATIONS

SMART HOME SYSTEMS, ATTENDANCE TRACKING, PERSONALIZED SECURITY.

CONCLUSION

- Achieved real-time, on-device recognition using affordable hardware.
- Learned critical lessons in system optimization and user design.
- Demonstrated feasibility of deploying AI at the edge for practical applications.



REFERENCES

- Joy Buolamwini and Timnit Gebru, *"Gender Shades: Intersectional accuracy disparities in commercial gender classification,"* Proceedings of the ACM FAT, 2018.
- 2. Michael Kirby and Lawrence Sirovich, *"Application of the Karhunen-Loeve procedure for the characterization of human faces,"* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp. 103–108, 1990.
- 3. Florian Schroff, Dmitry Kalenichenko, and James Philbin, *"FaceNet: A unified embedding for face recognition and clustering,"* Proceedings of the IEEE CVPR, 2015.
- 4. Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf, *"DeepFace: Closing the gap to human-level performance in face verification,"* Proceedings of the IEEE CVPR, 2014.
- 5. Matthew Turk and Alex Pentland, *"Eigenfaces for recognition,"* Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86, 1991