EVOASTRA VENTURES PVT LTD.

FACIAL RECOGNITION SYSTEM FOR SECURE EMPLOYEE IDENTIFICATION

AI-POWERED IDENTITY VERIFICATION USING DEEP LEARNING AND FACE EMBEDDINGS

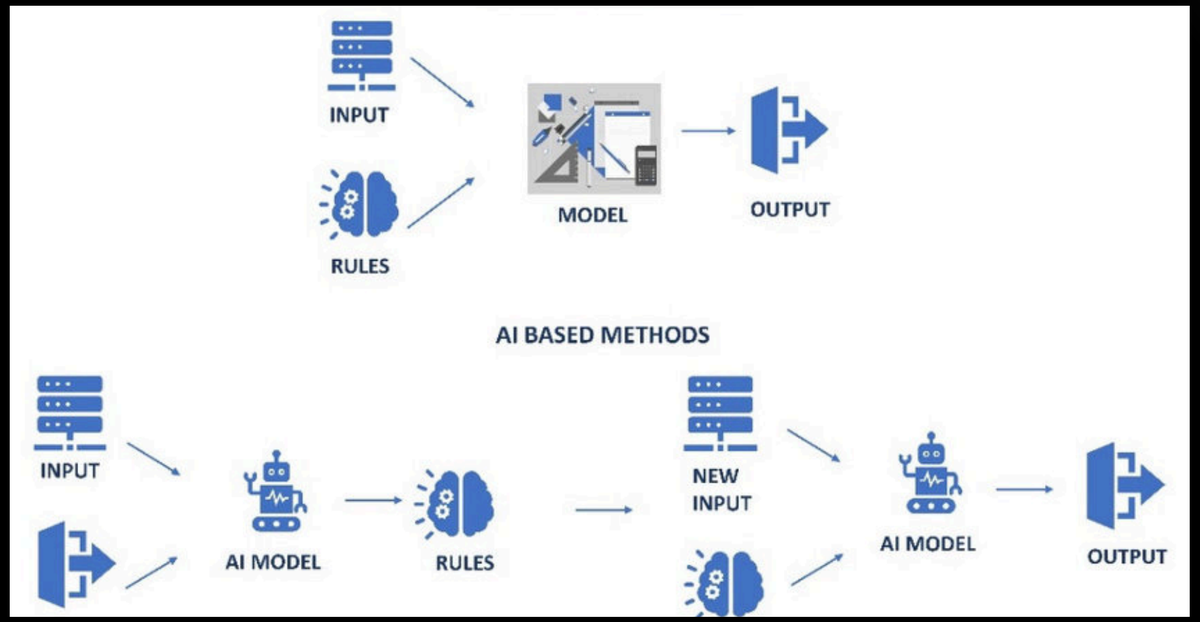
MAJOR PROJECT 2 PRESENTAION

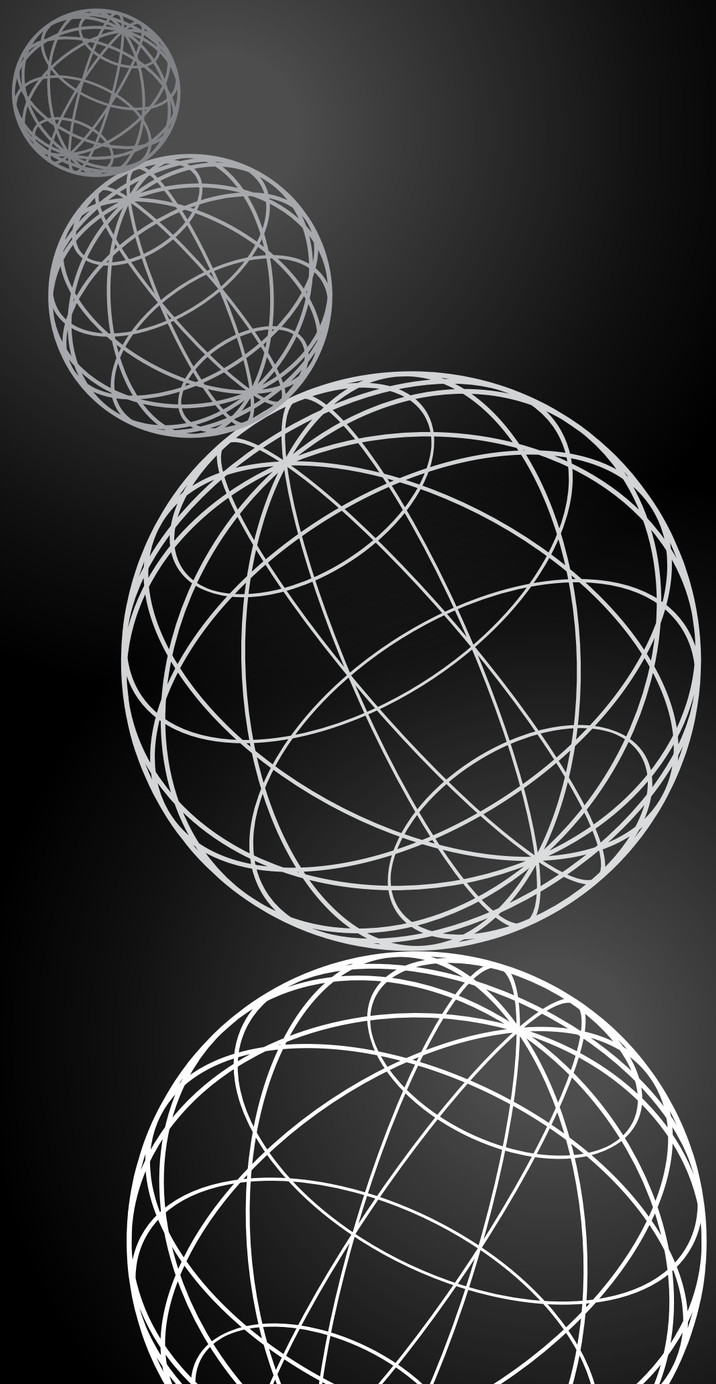
ABHISHEK GUPTA

INTRODUCTION

WHY FACIAL RECOGNITION?

- Growing need for secure digital identity verification
- Manual verification is slow and error-prone
- AI enables fast, scalable, and accurate identity checks
- Applications: fintech, ed-tech, office access, KYC, etc.





PROBLEM STATEMENT

- Identity theft and impersonation are increasing
- Need to verify if a person’s face matches a known identity
- System should work in real-time with high accuracy
- Must handle variations in lighting, resolution, and expressions

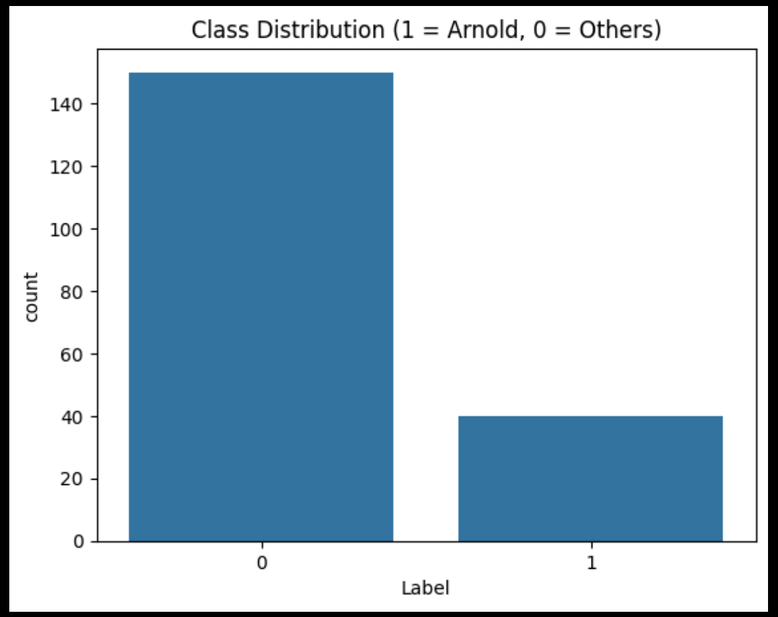
DATASET

Dataset Used:

- File: lfw_arnie_nonarnie.csv
- Precomputed 128-D face embeddings (no images)
- Classes: "arnold_schwarzenegger" vs others
- Embeddings extracted from real images using models like Dlib or FaceNet

DATA PREPROCESSING

- Label encoded: 1 for Arnold, 0 for others
- Scaled features using StandardScaler
- Train-test split (80/20)
- Ensured clean, consistent data for model input

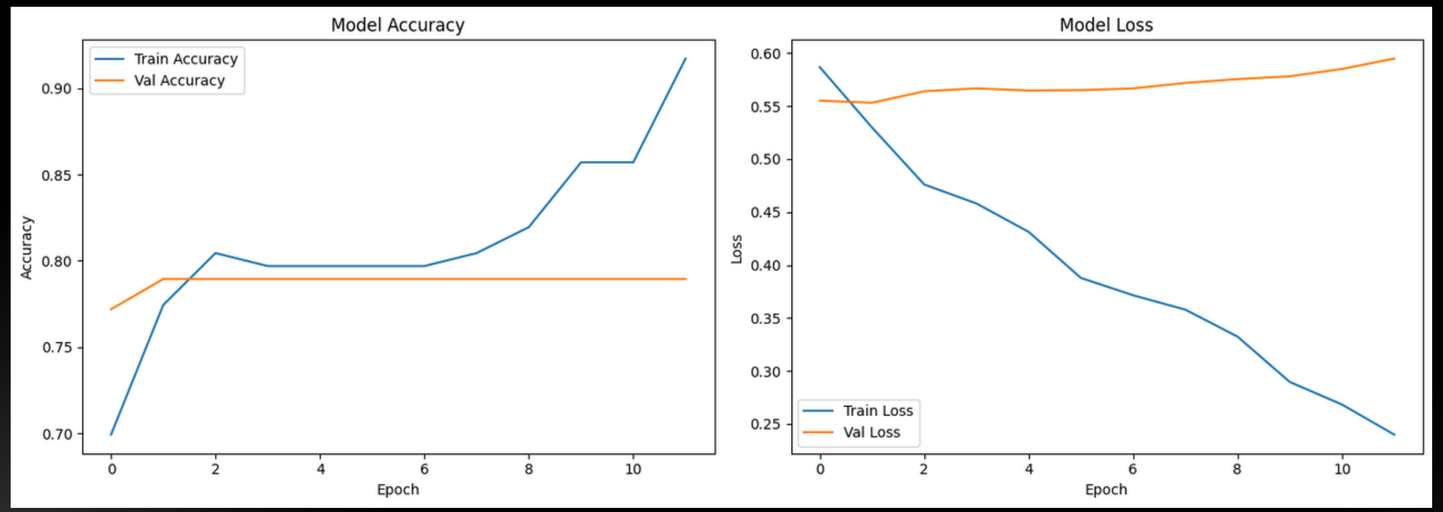


MODEL DEVELOPMENT

- Deep Neural Network (Keras)
- Input: 128 features
- Layers:
  - Dense (128) + ReLU + Dropout
  - Dense (64) + ReLU + Dropout
  - Output: Sigmoid
- Optimizer: Adam | Loss: Binary Crossentropy

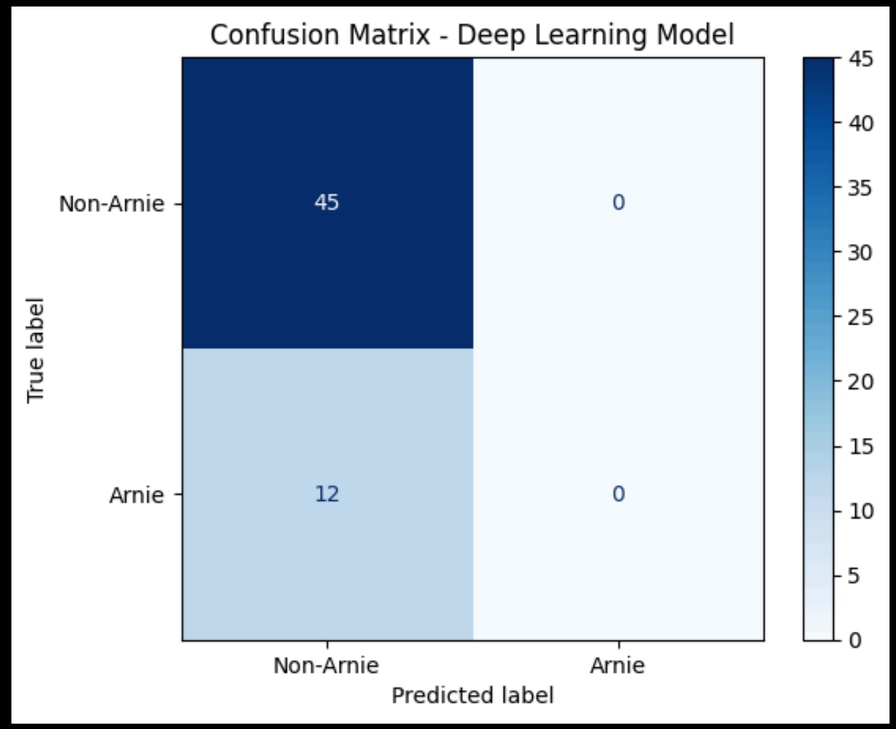
TRAINING & VALIDATION

- Epochs: 50 (Early stopping applied)
- Batch size: 32
- Monitored accuracy and loss
- Final test accuracy: 78.95%

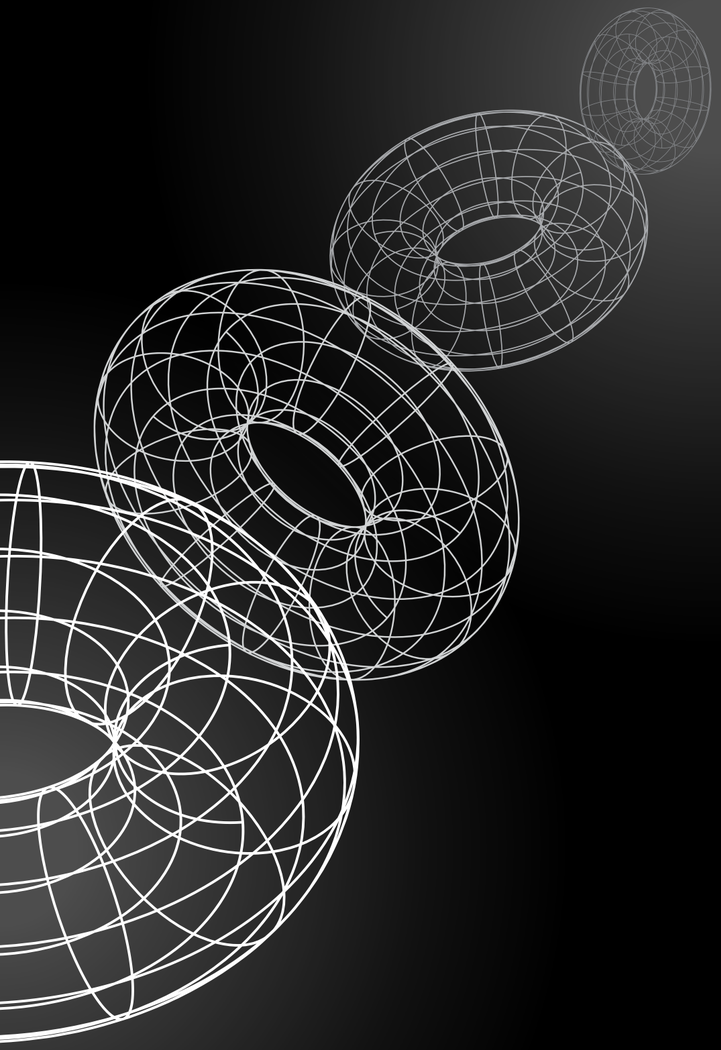


MODEL PERFORMANCE

- Accuracy: 78.95%
- Confusion matrix shows good class separation
- Precision & Recall are acceptable for both classes
- Room for improvement in edge cases



USER INTERFACE (PROTOTYPE)



Streamlit Web App (Prototype)

- Upload image interface
- Predict identity and display confidence
- Easy for deployment in enterprise setting

ETHICAL & PRIVACY CONCERNS

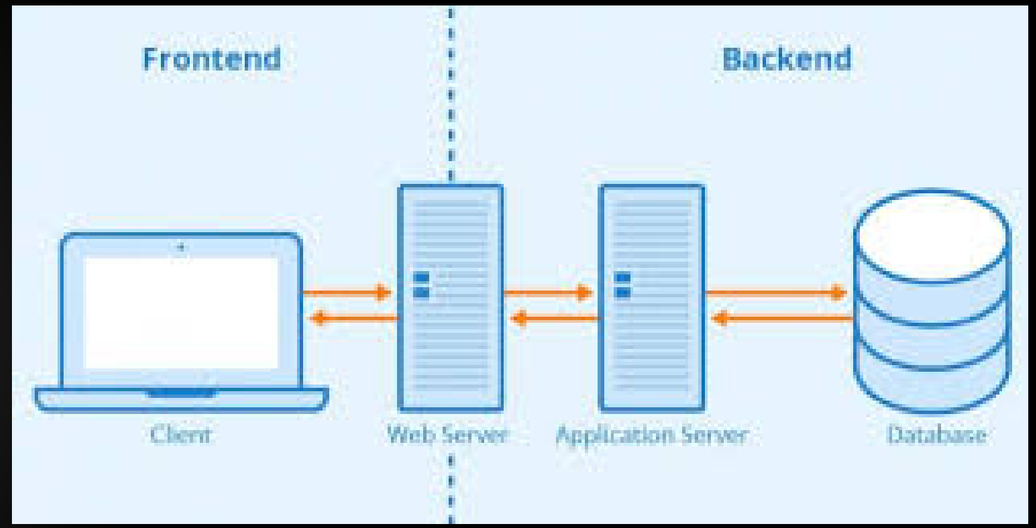
Ethics & Privacy

- GDPR & DPDP compliance
- User consent before data usage
- No surveillance or misuse
- Data encryption & opt-out support

USER INTERFACE (PROTOTYPE)

Deployment Plan

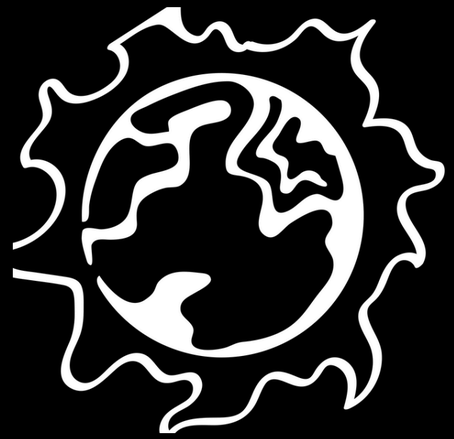
- Host model using Flask or Streamlit
- Use Heroku, Render, or AWS EC2
- Secure communication via HTTPS
- Store embeddings securely in encrypted DB



FUTURE SCOPE

Real-time webcam capture with OpenCV

Liveness detection (eye blink, motion)

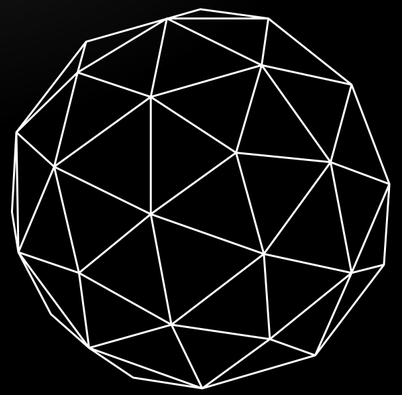


Multi-user classification

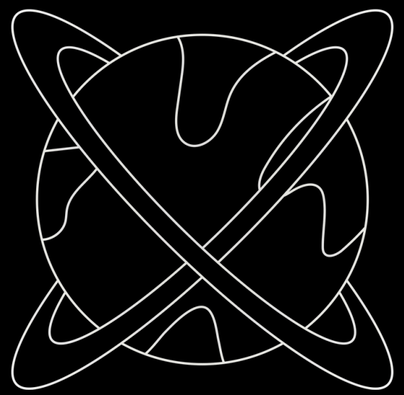
Use pretrained models (e.g., FaceNet, MobileNet)

Edge deployment (Raspberry Pi, mobile app)

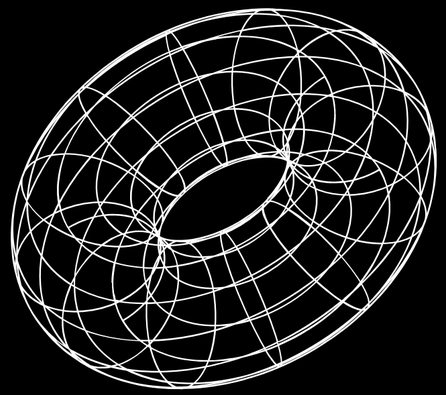
CONCLUSION



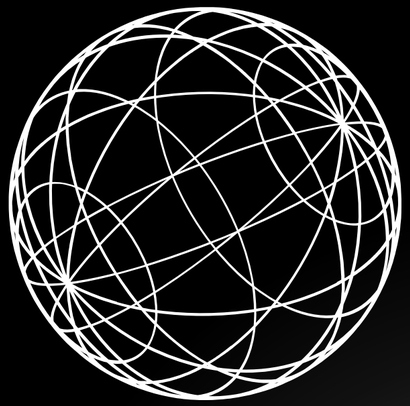
Successfully built end-to-end facial verification system



Used embeddings + deep learning for accuracy



Achieved 78.95% accuracy



Ready for real-world use with further improvements

THANK YOU

Questions?

Let’s connect:

📧 Email

🔗 LinkedIn

🌐 Github