Facial Recognition App

Mia Brown

AGENDA

- Introduction
- Application Development
- Classifying & Server connection
- Facial Recognition Implementation
- Video Demo
- Future Goals

Project plan

The plan for this project was to take Facial Recognition and implement that into a security system.

An Application – Android Studio

- User interface
- Adding of known faces to a database
- Viewing live feed from camera

Database

- My sql, php, apache, phpMyAdmin

Motion Detection Camera

- WYZE home security System

Facial Recognition

- Take the image from the Wyze live feed
- Recognize face
- Send back who it is

Key features

- **Real-time Image Capture and Upload:**
  - Capture or upload images directly from the app.
- **Flask Server for Facial Recognition:**
  - Flask receives images, processes them using a trained model, and returns the result.
- **Facial Recognition:**
  - Matches faces from the database and sends back the identified name (or "unknown").
- **Notifications:**
  - The app notifies users when a face is recognized.
- **Database Integration:**
  - Facial encodings are stored and compared with the uploaded images.

Technologies Used

Frontend (Android):

Java for app development.

Retrofit for making HTTP requests to the Flask server.

Backend (Flask):

Flask handles image requests and facial recognition logic.

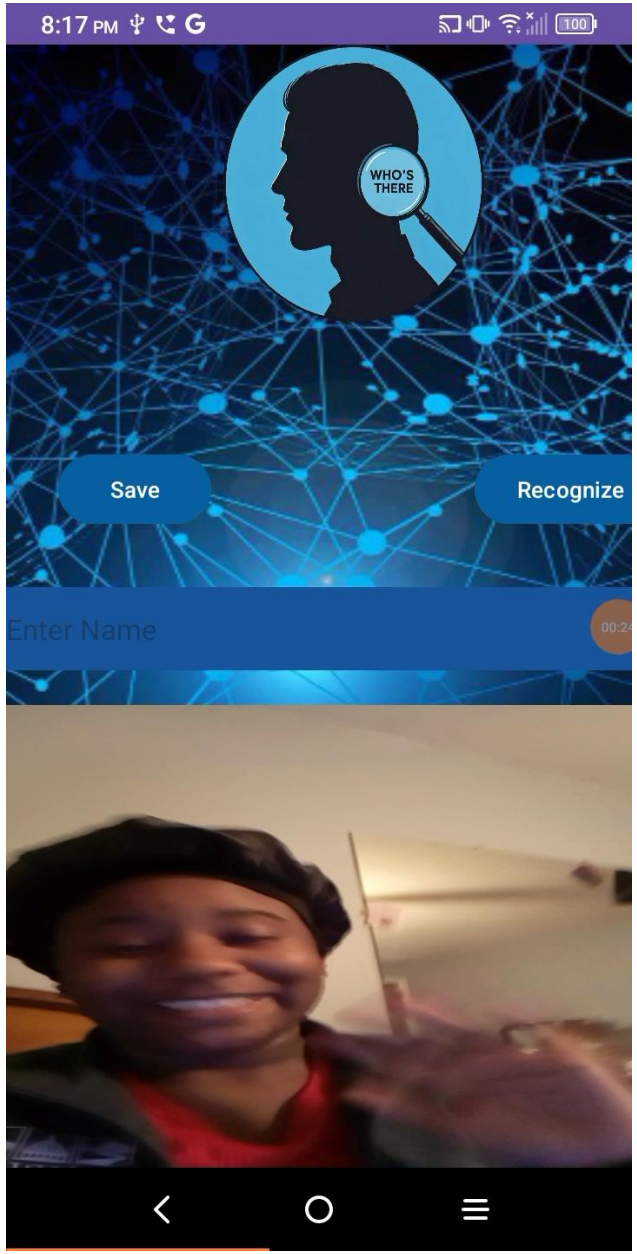
Libraries Used:

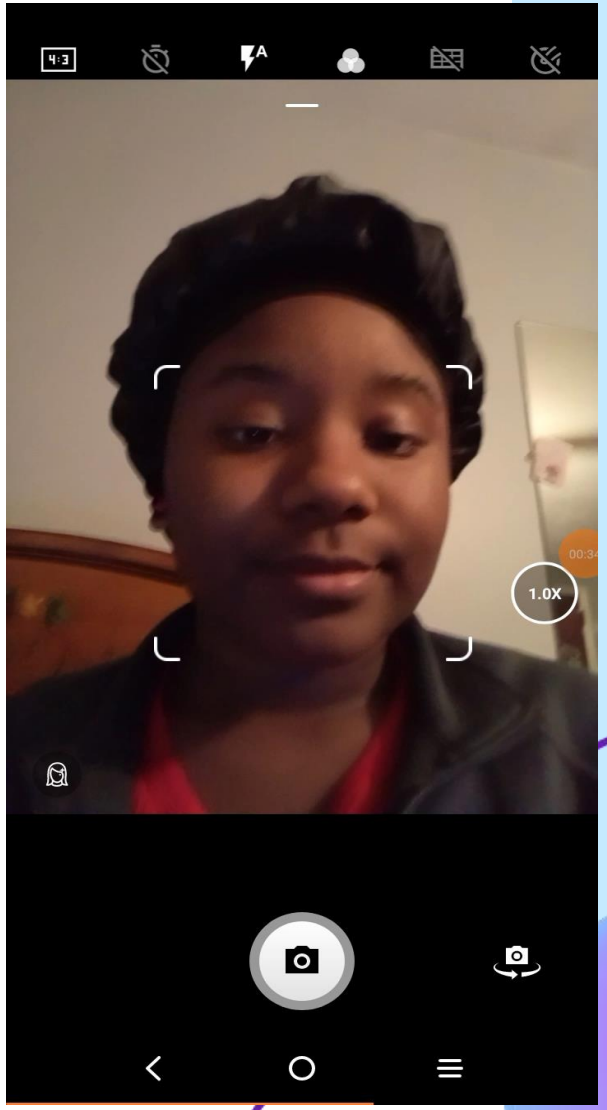
- `face_recognition` for facial detection and encoding.
- OpenCV for image processing.

Machine Learning:

KNN or other classifiers for face recognition.

Application – UI side


8:17 PM
WHO'S THERE
Save
Recognize
Enter Name


1.0X


WHO'S THERE

Dataset Handling

&

Preparation

Kaggle Dataset

- 2500 images
- Avg. 75 pp
- Name_0.jpg
- 160x160

QuickPic.py

Face Capture and ZIP Handling Overview

- **Camera Initialization**:

Opens the default camera (`cv2.VideoCapture(0)`), errors out if unavailable.

- **ZIP File**:

Images are saved to `archive.zip` located in `C:\Users\honey\Downloads`.

- **Capture & Process**:
  - Captures 76 frames.
  - Converts each frame to grayscale.
  - Detects faces using a Haar Cascade model.
- **Face Cropping & Resizing**:
  - Crops detected faces.
  - Resizes each face to 160x160 pixels.
- **Image Encoding & Saving**:
  - Encodes faces to JPEG.
  - Names (e.g., `Mia_Brown_0.jpg`) and saves to ZIP.
- **Completion**:

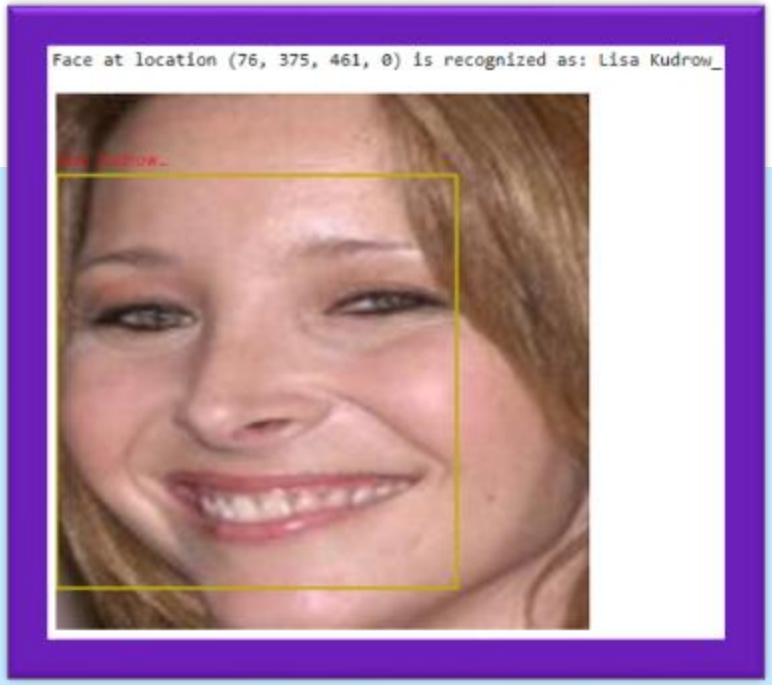
Prints success message after all frames are processed.

KNN Classifier - Kaggle

The **K-Nearest Neighbors (KNN)** is a **classifier** model, but it also has an aspect related to **training**

If the input image contains one or more faces:

- It detects their locations and encodes them.
- Compares the encodings with the trained model.
- Adds bounding boxes and the predicted names to the image.
- Displays the annotated image for review.

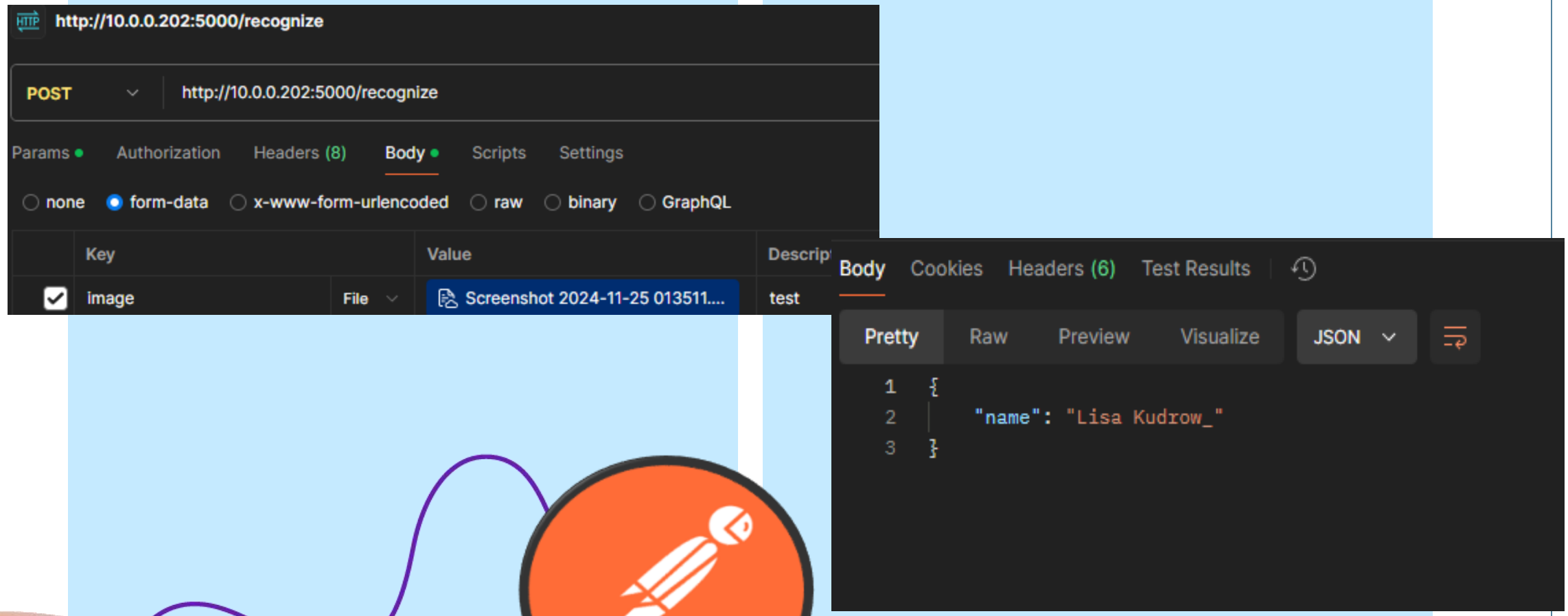


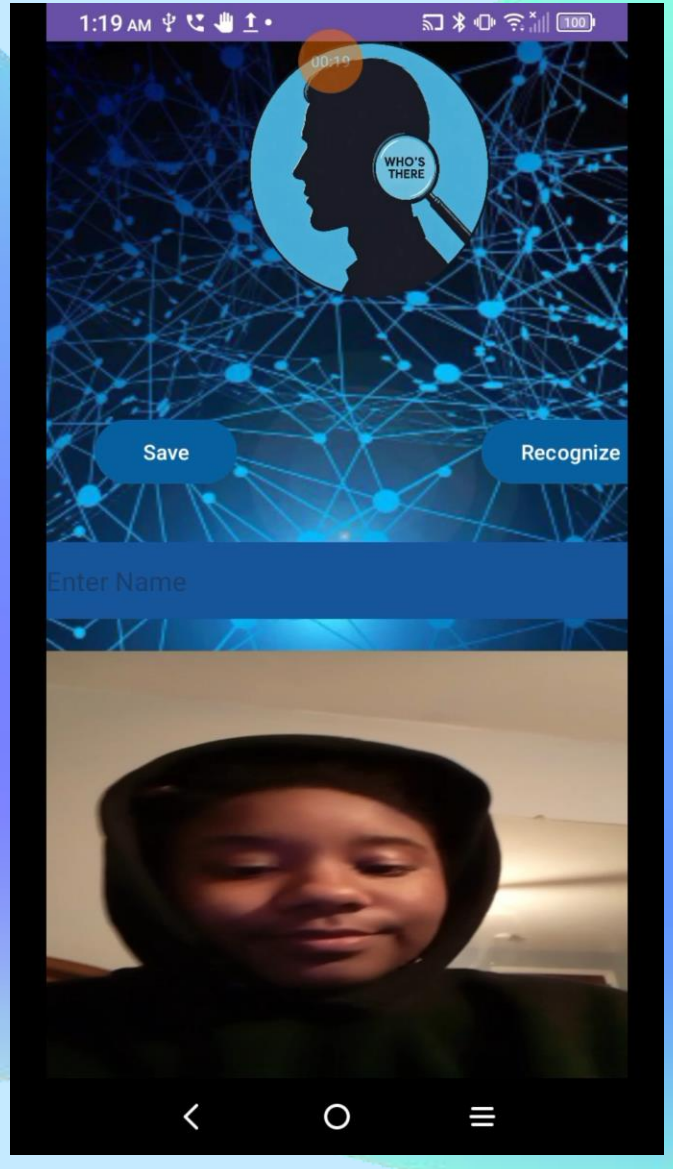
Flask

Flask Facial Recognition API

- **Setup**:
  - Imports Flask, CORS, `face_recognition`, and `pickle`.
  - Loads KNN model and label files.
- **Endpoint**:
  - `/recognize` handles POST requests with an image.
- **Processing**:
  - Validates input image contains one face.
  - Extracts face encodings for recognition.
- **Recognition**:
  - Predicts name using KNN model.
  - Matches label if confidence > 0.6; else returns "unknown".
- **Error Handling**:
  - Handles invalid input or processing errors.
- **Deployment**:
  - Runs on `0.0.0.0:5000`.

Testing with Postman





Demo

Future Goals

Handle "Unknown" faces

- Ask for name for image
- "save" and upload to zip file
- Duplicate image in range 75, increment label (ex- Bob_0.jpg, Bob_1.jpg)
- Import watchdog , Rerun KNN

Make a more Attractive app

Try with a different external camera

- Wyze had many safeguards to prevent third-party streaming

Implement MySql database, php, phpMyAdmin

Thank You

Live streaming | Android media | Android Developers
Android S - Capture Image
Youtube Video

Python: how to capture image from webcam on click using OpenCV - Stack Overflow
Zip in Python - Stack Overflow
How to crop and save the detected faces in OpenCV Python?

Kaggle - Dataset

KNN - Facial Rec - Kaggle