Assistive Mobile Application for Prosopagnosia Patients

1

The Team

Meet the team

Lamis Kamal

Mohamed Zakaria

Nada Elmaghraby

Suhaila Ahmed

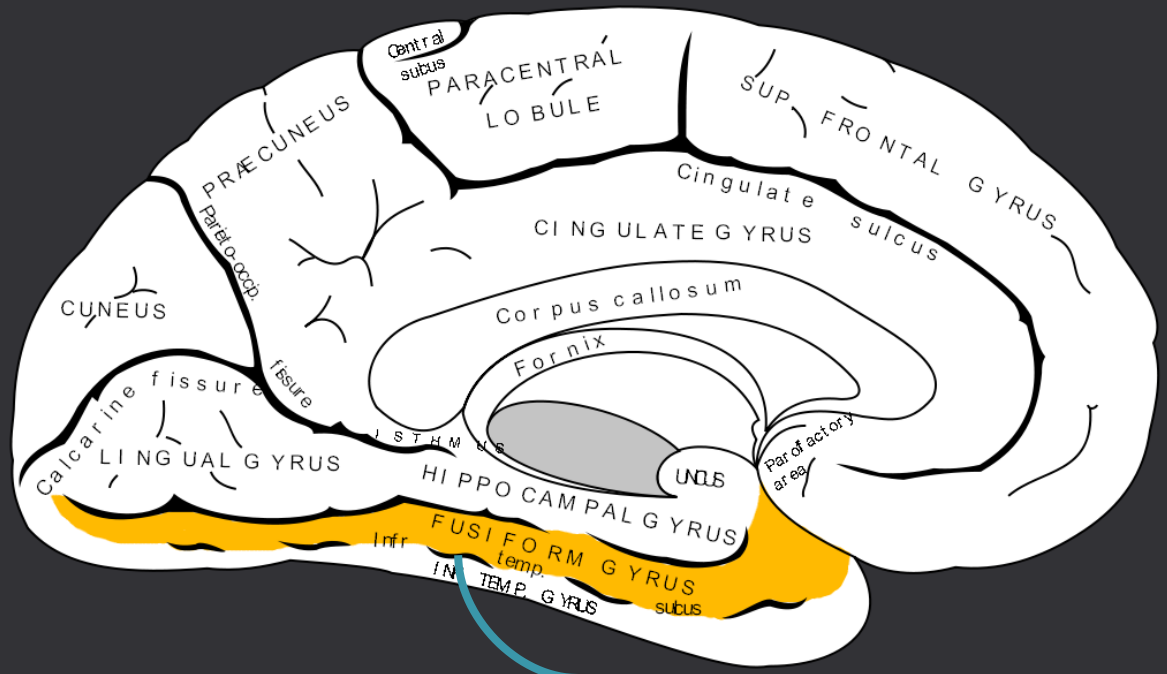
Shimaa Mamdouh

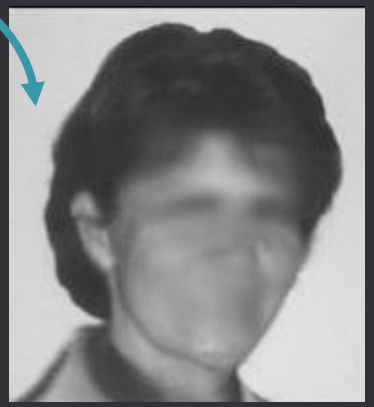
2

The Problem

Prosopagnosia (face blindness)

- Prosopagnosia is a result of abnormalities, damage, or impairment in the right fusiform gyrus.
- Which is a fold in the brain that coordinates the neural systems that control facial perception and memory.



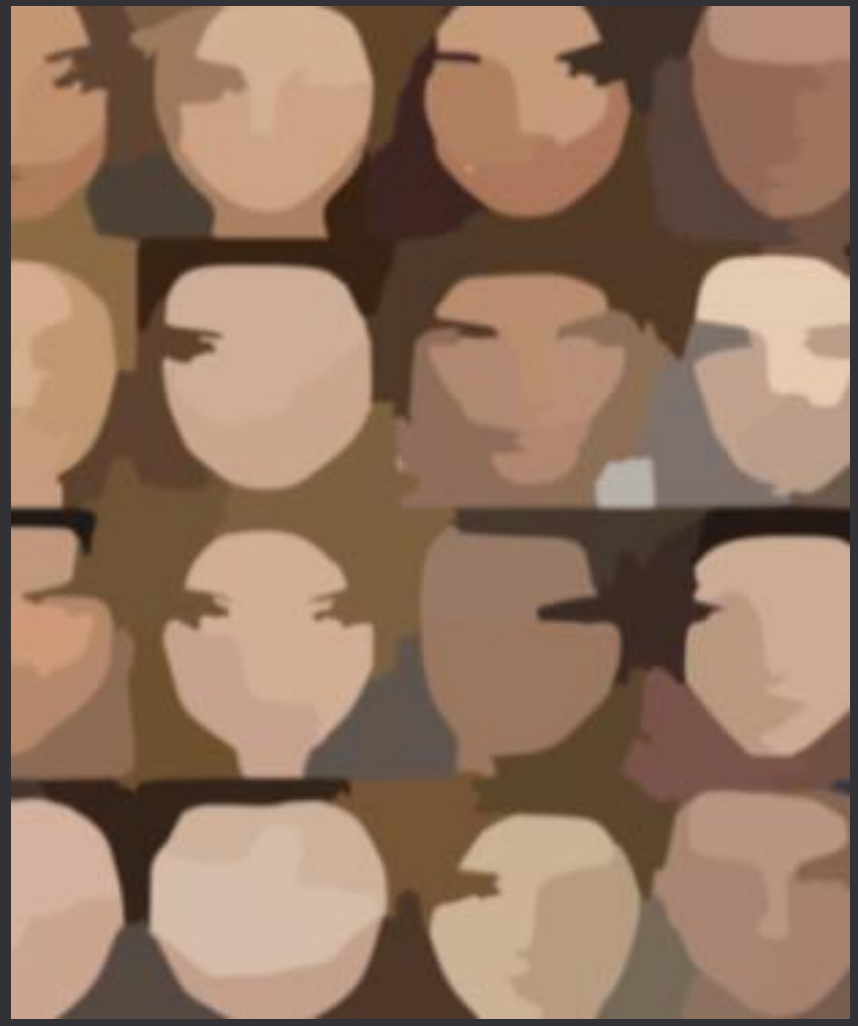


Prosopagnosia (face blindness)

- There are two types of prosopagnosia:
  1. Developmental
  2. Acquired

- Studies has indicated that 1 in 50 people may have developmental prosopagnosia.

- There is no treatment for prosopagnosia but patients adopt strategies for identifying people.



3

The Opportunity

The Opportunity

- Statistics show there are over **six billion** smartphone worldwide, and expected to grow.

- Artificial intelligence models in the field of facial recognition have become so advanced.

4

Proposed Solution

Proposed Solution

- A mobile app “FaceReminder” perform the face recognition task for the user.
- It notifies the user whether they know the person or not.
- User builds his own network and can modify it.

5

Similar Products

Similar Products

Orcam MyEye

1. For blind people
2. Activated by voice or click
3. Costs around $4250 USD

Social Recall

1. A mobile application for events
2. The information of attendees used during the event
3. Currently being expanded to include Prosopagnosia

6

How we differ

How we differ

Traget Prosopagnosia Patients

Tools and platforms used

Can be connected to an external camera via Bluetooth, that is optional to each user

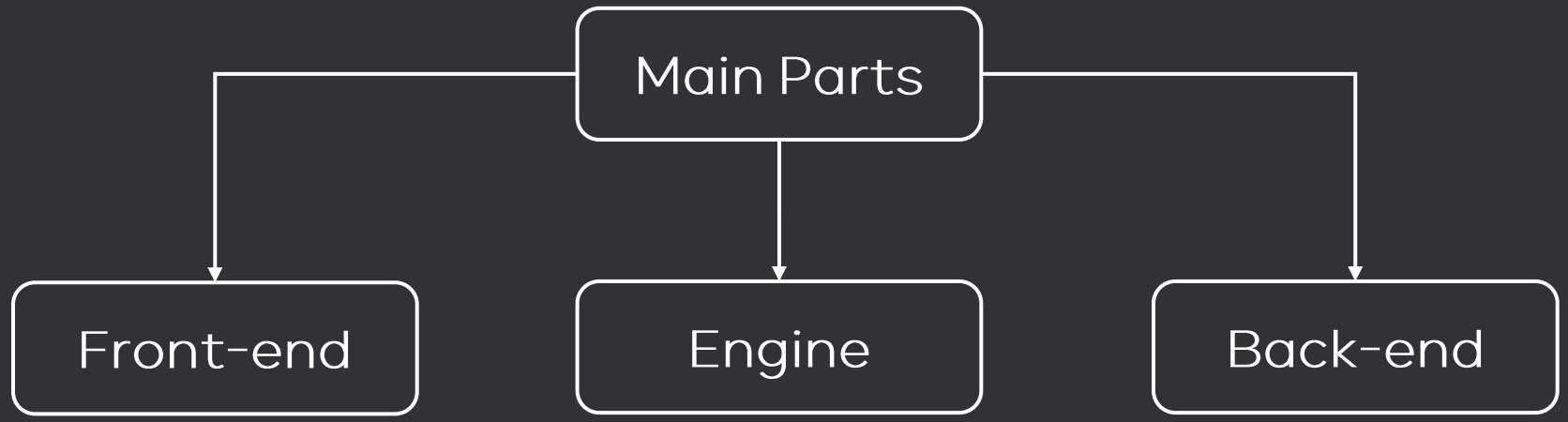
Cost effective compared to similar products

7

Methodology

Methodology

In order to achieve the goal of the application, we divided it into

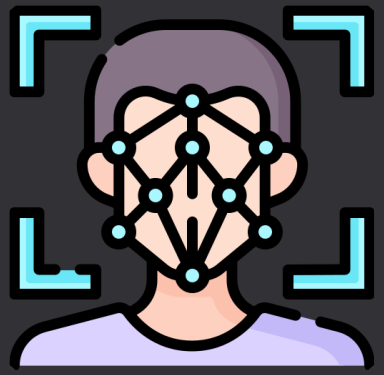


1. Engine

Detection

- The face/faces are bounded by boxes
- Annotating facial landmarks

Alignment & Analysis



- Producing face mesh
- Orienting the face in a certain direction
- Cropping tight around the face
- Extracting the values of interest

Recognition

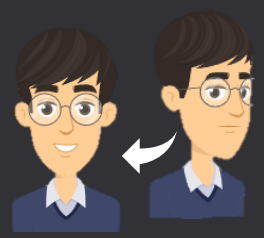
- Comparing the result to known faces

A & B. Detection &Alignment

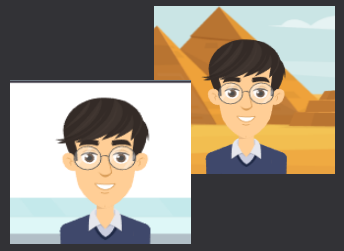
Why RetinaFace detector?

Challenges

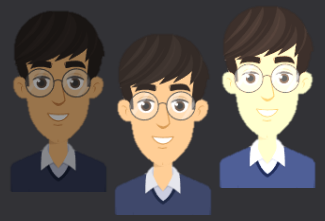
Uncontrollable environment



orientations & expressions



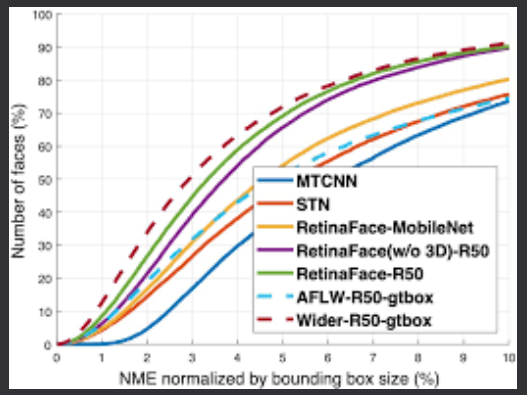
Outdoors vs indoors



Dim or bright lighting

Options

RetinaFace among other detectors:

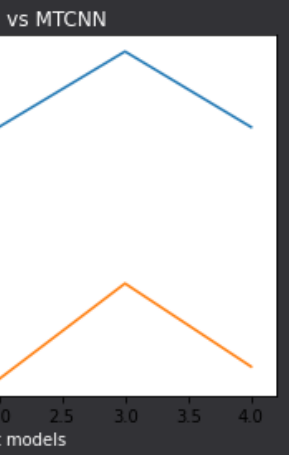


RetinaFace works in:

- Real-time
- Single CPU core
- Low-resolution image

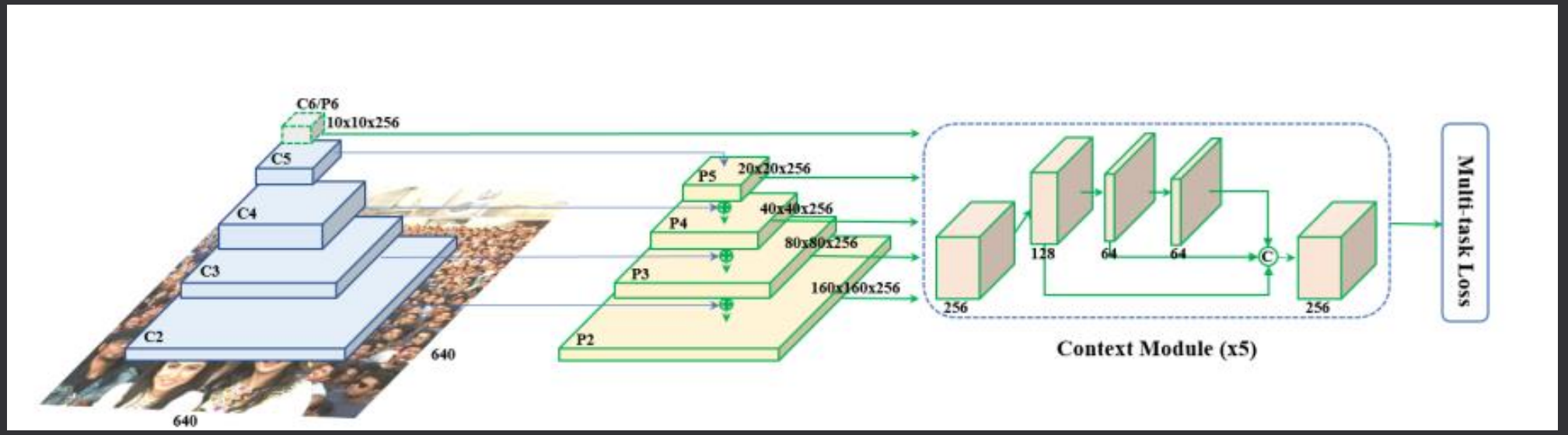
Decision

RetinaFace vs MTCNN



RetinaFace is 2x faster

RetinaFace implementation flow



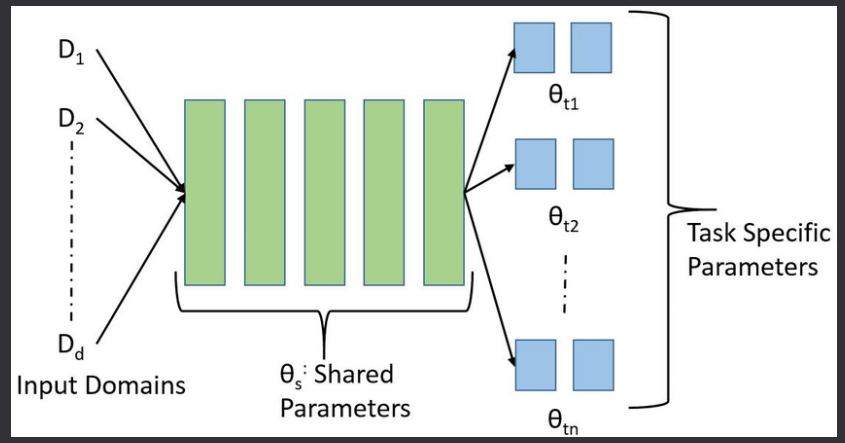
Feature Pyramid: to notice smaller faces

Context Module: for enhanced performance

Multi-task loss

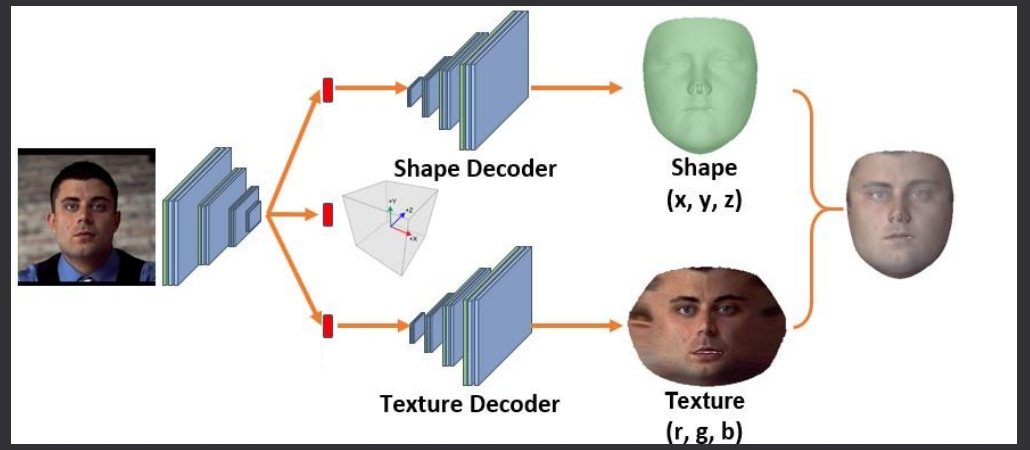
How RetinaFace could achieve faster and better performance?

By employing:



Multi-task loss:

- Saves computation time
- Emphases on Face box and then facial landmarks

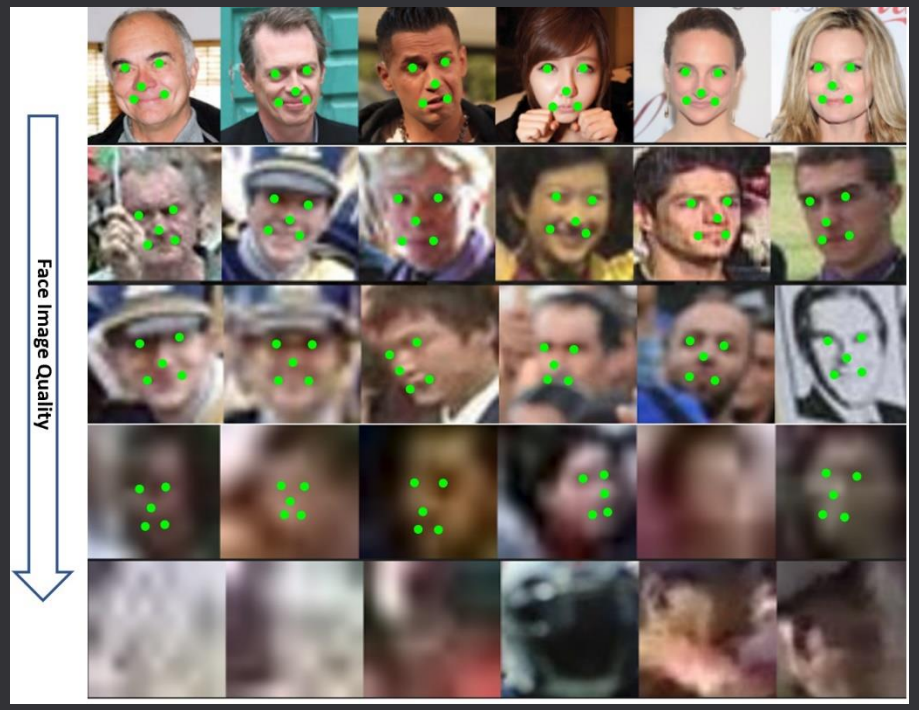


Mesh decoder

- Enhances performance
- Extracts shapes and texture

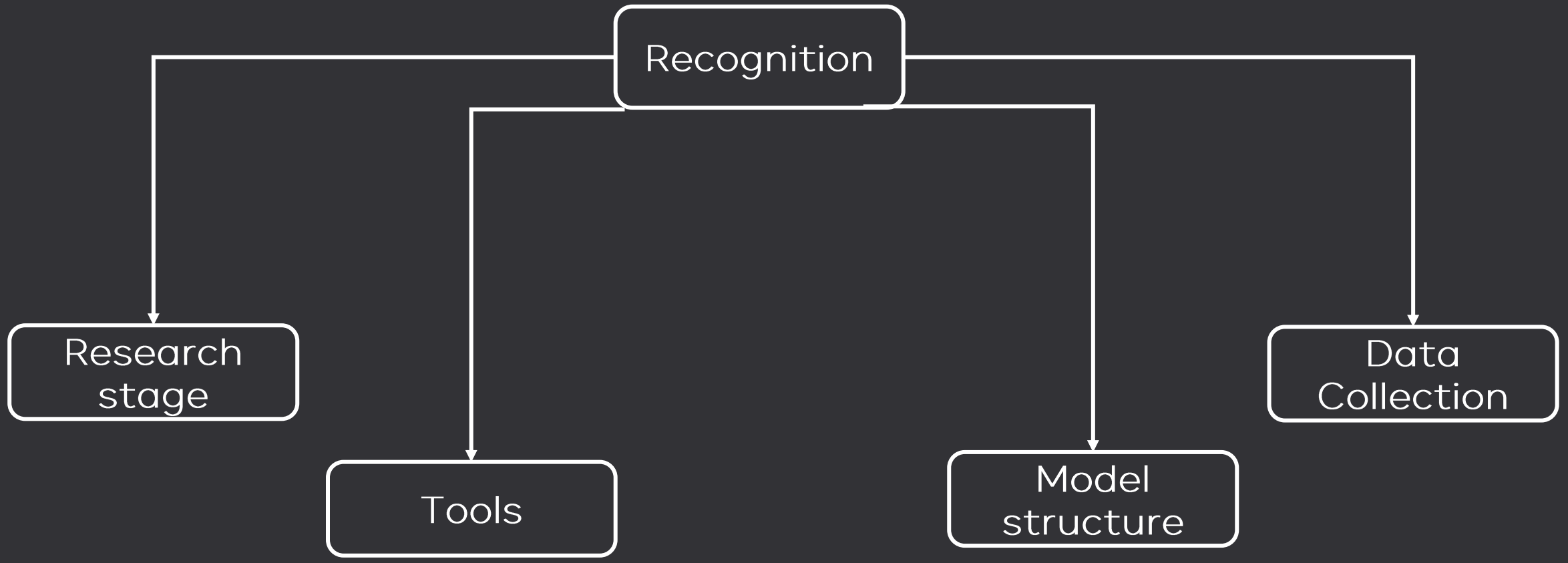
RetinaFace Dataset

- RetinaFace is trained on WIDERFACE hard dataset
- Its performance is enhanced by manually annotating facial landmarks
- Augmenting data by random cropping and horizontal flipping.



WIDERFACE hard dataset

C. Recognition



1. Research stage
How big names use face recognition?

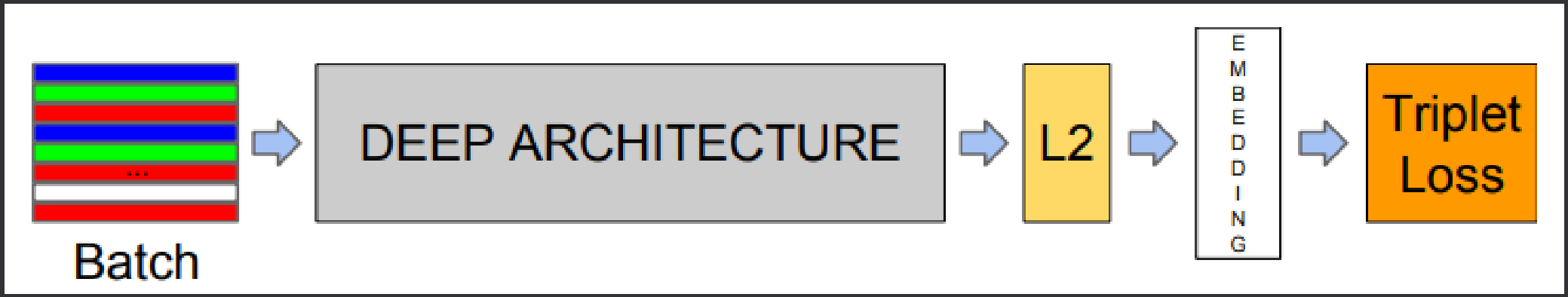
				
Used Technique	DeepFace	FaceNet	Vision Framework	Iris Recognition
Disadvantage	5x memory consumption	_____	Require unavailable facilities	Different route
Size	511 M	92 M	_____	_____

2. Tools

- DeepFace framework
  - Lightweight framework for python
  - Licensed under MIT License; allowed in commercial use
  - Provides different models including **FaceNet** for the face recognition task
  - Provides different face detectors

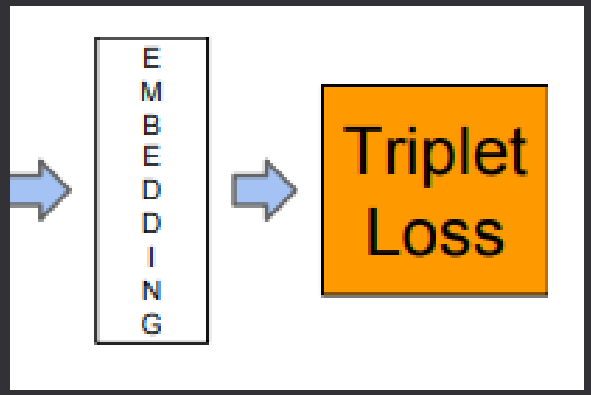
3. Recognition task (Model structure)

- We settled on *FaceNet* system developed by researchers at Google
- Based on Google, The system accuracy on LFW Dataset is nearly 99%



3.1. FaceNet Embeddings and Loss Function

- FaceNet differ from other techniques that it eliminates bottlenecks through embeddings.
- It extracts 128 byte per face.
- Triplet loss function which briefly uses 3 images:
  - anchor: reference input
  - positive: matching image
  - negative: non matching image



3.2. FaceNet loss function

- The loss function is calculated as follows using squared distance:

$$\left\| f(x_i^a) - f(x_i^p) \right\|^2 + \alpha < \| f(x_i^a) - f(x_i^n) \|^2$$

$$L = max(\left\| f(x_i^a) - f(x_i^p) \right\|^2 - \| f(x_i^a) - f(x_i^n) \|^2 + \alpha, 0)$$

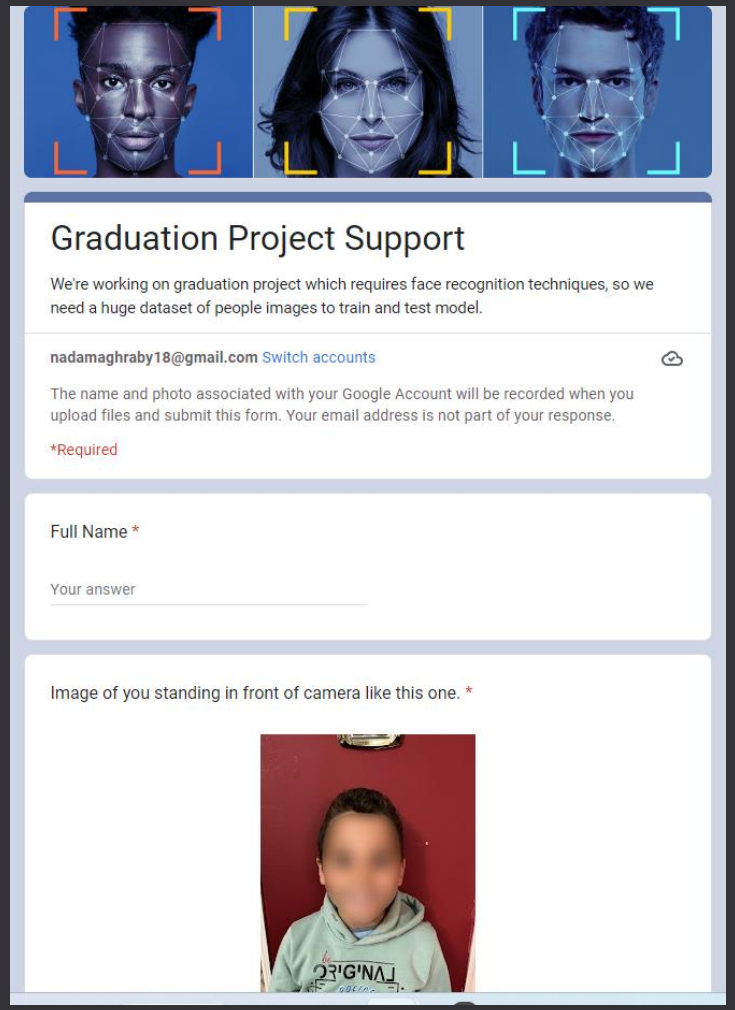
- The loss value is minimzed through iterations.
- As long as the value approaches zero, the accuracy of recognition increases.

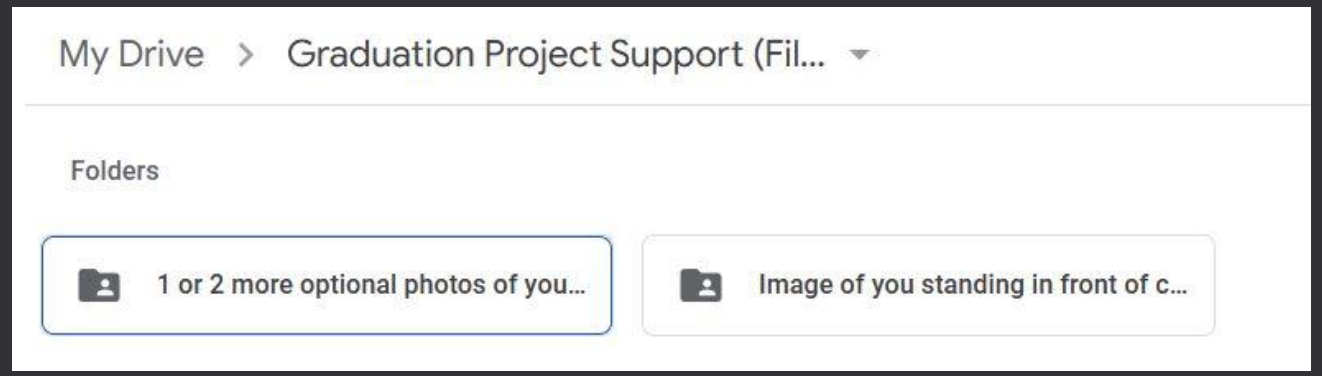
4. Data Collection

- Celebrity dataset which is made up of professional images that have no relevance to our app.

- Images are collected using a google form of images taken by normal cell phones which are quite relevant to our application.

Collection using a google form





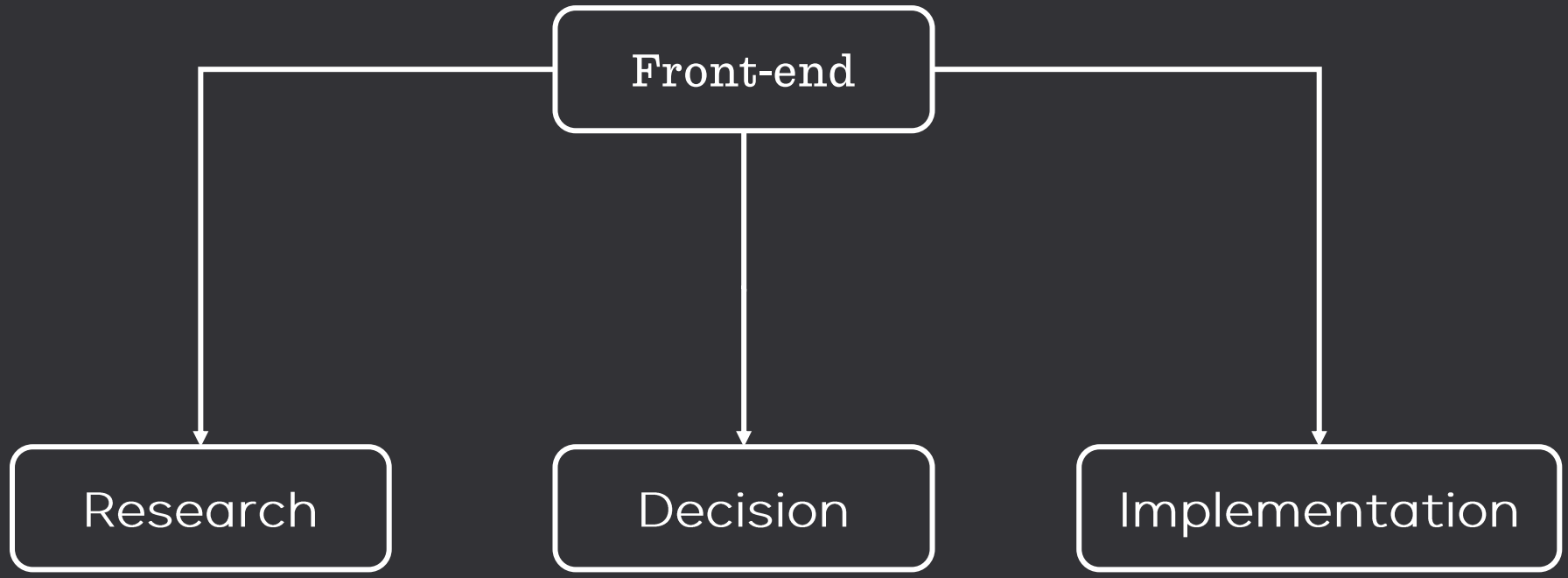
Data preparation

All gathered images are scaled and converted to gray scale before introducing them to the model.

Images are:

- Scaled to (512x512); as not to miss any important detail.
- Converted to gray scale to reduce processing time

2. Front-end



A. Research Stage

- Front-end Platforms

	React native 	Flutter 
Language	Java script	Dart
Community	Numerours open source libraries and resources	Less resources
Development time	Recompiled every time changes made	Changes can be seen in real time

B. Decision

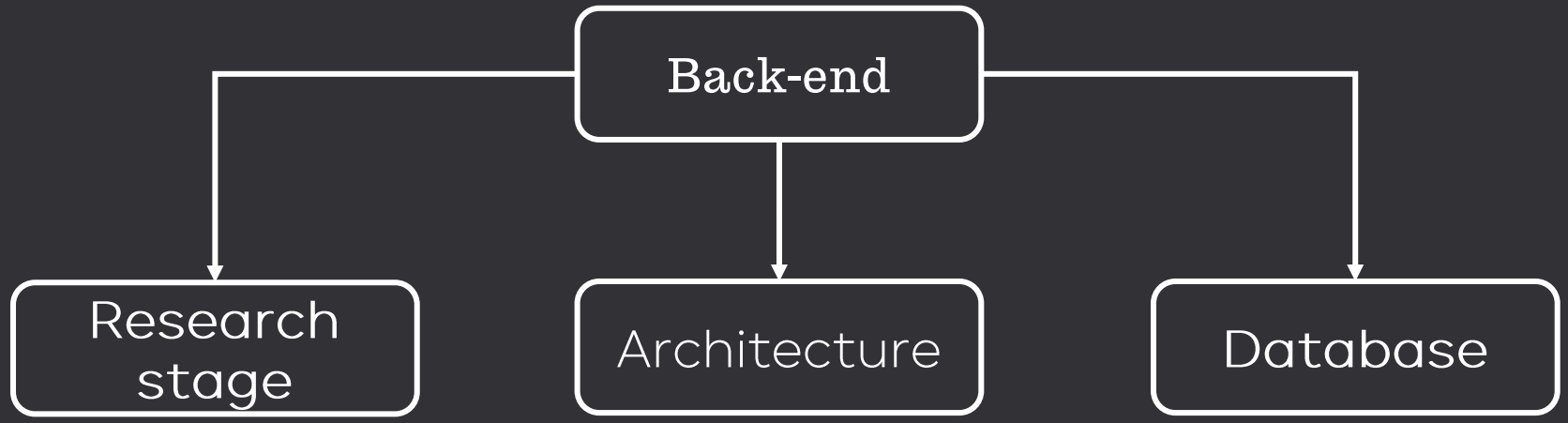
- To setup the store for user data, redux toolkit is used.
- To access mobile camera we used react-native-vision-camera library.
- To access mobile local storage we used react-native-image-picker library.

Redux toolkit

Vision-camera library

3. Back-end

In order to build the back-end system of application, three steps was taken



A. Research stage

- Back-end platforms:

a) Based on comparisons , we settled on Django and Flask because :

1. A Python language which we are familiar with.
2. To easily interact with AI models.

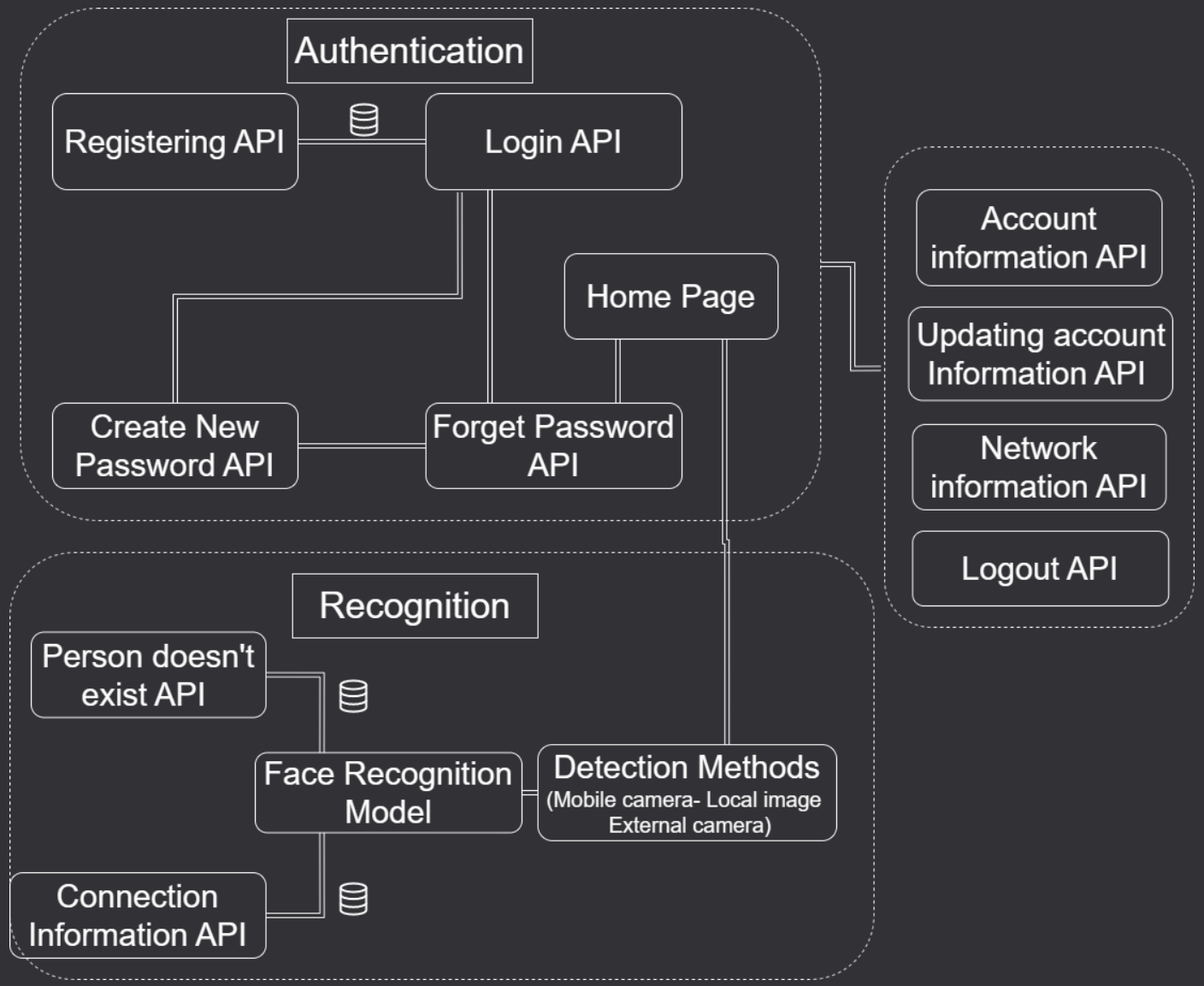
Flask

b) Then, we select Django because:

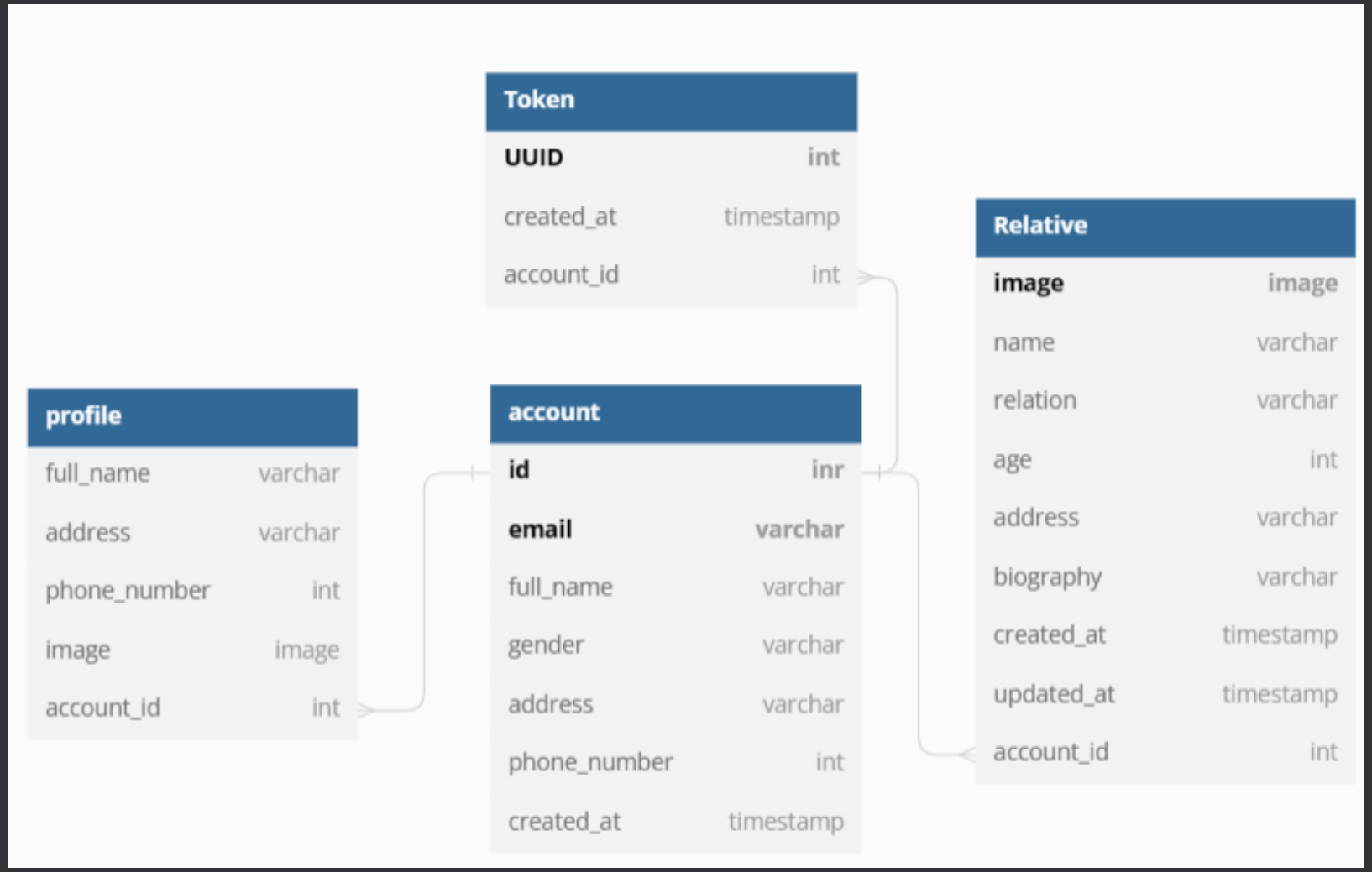
It has Django rest framework which is the best option to interact and send APIs to React Native.

Django

B. Architecture



C. Database


Token
UUID int
created_at timestamp
account_id int
Relative
image image
name varchar
relation varchar
age int
address varchar
biography varchar
created_at timestamp
updated_at timestamp
account_id int
profile
full_name varchar
address varchar
phone_number int
image image
account_id int
account
id inr
email varchar
full_name varchar
gender varchar
address varchar
phone_number int
created_at timestamp

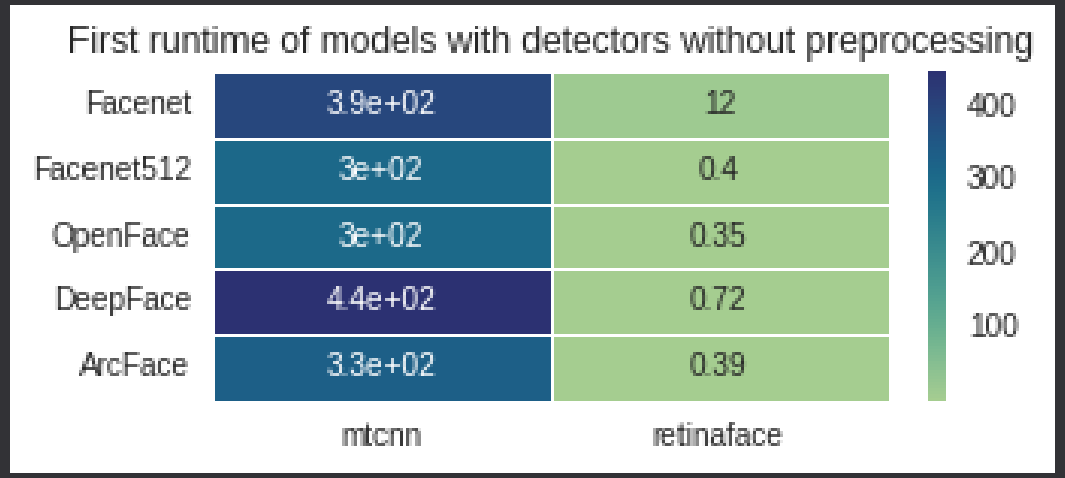
8

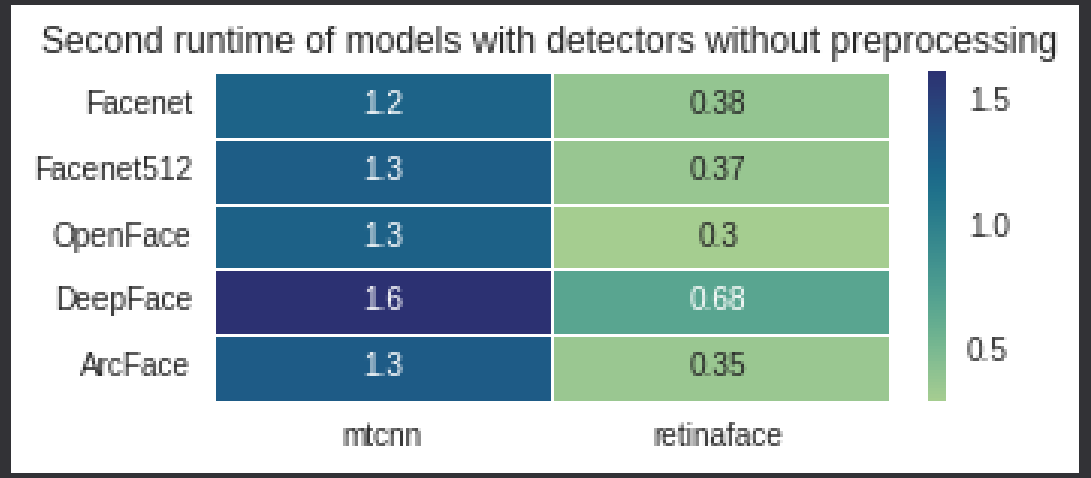
Results

Results

A. Face recognition model:

- Computation time was 7 minutes which is un-acceptable in our case.

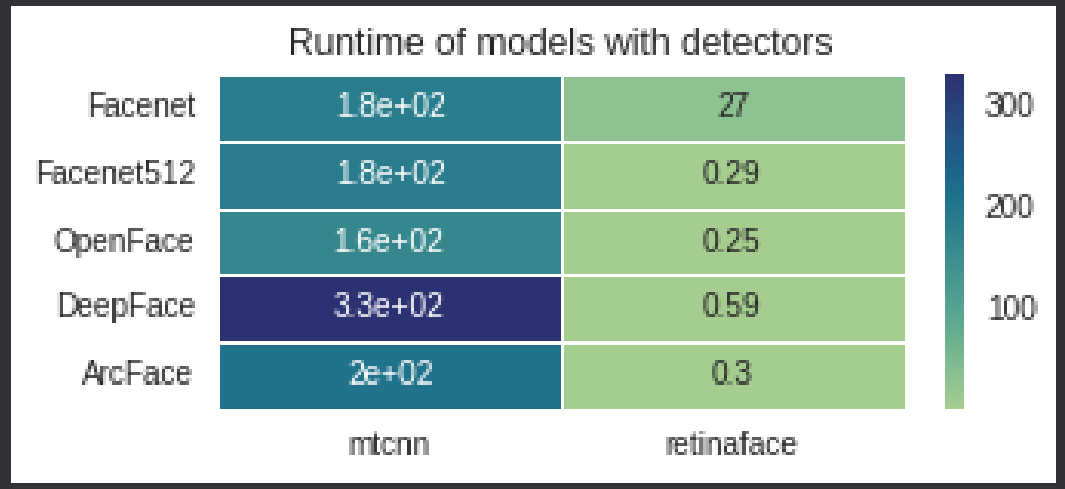


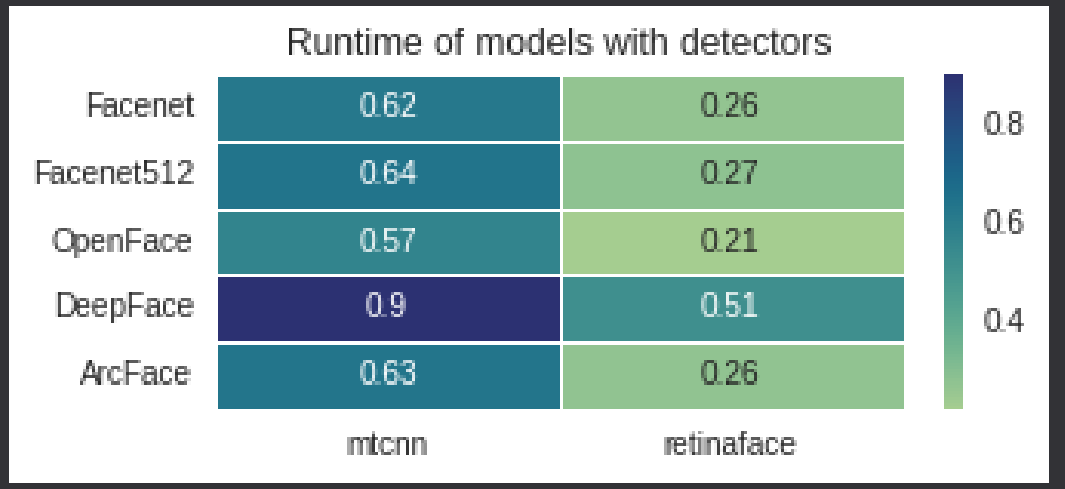


Results

A. Face recognition model (Modified):

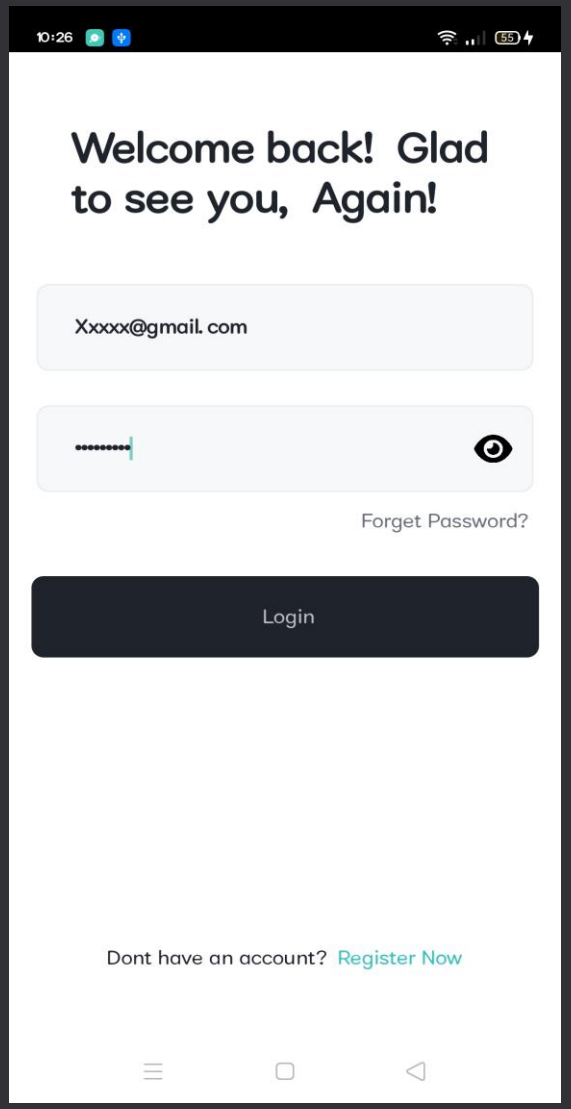
- Resizing and changing the depth of images significantly reduced the computation time to 3 minutes

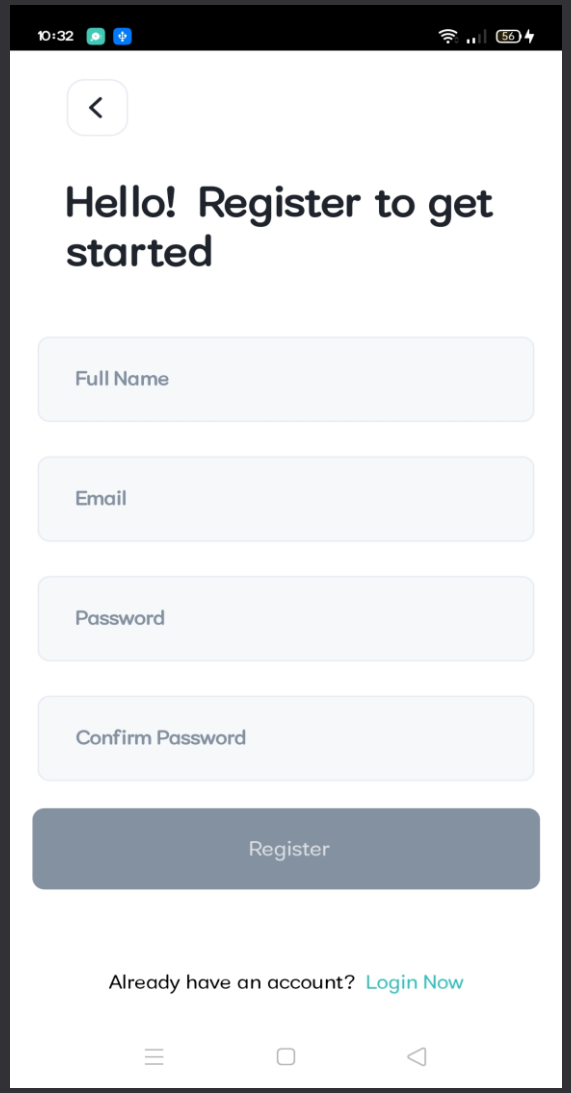


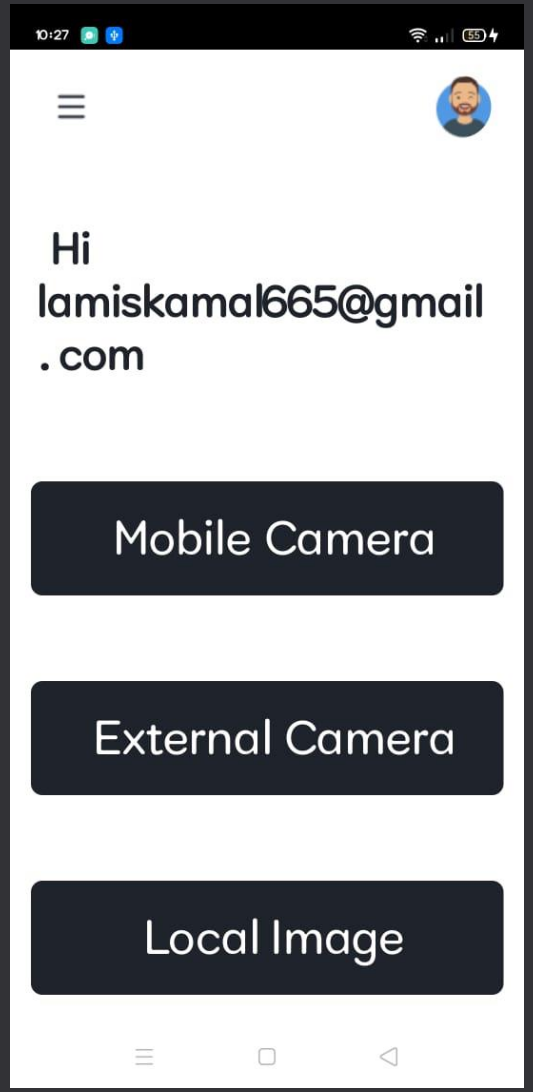


Results

B. Samples from FaceReminder application:

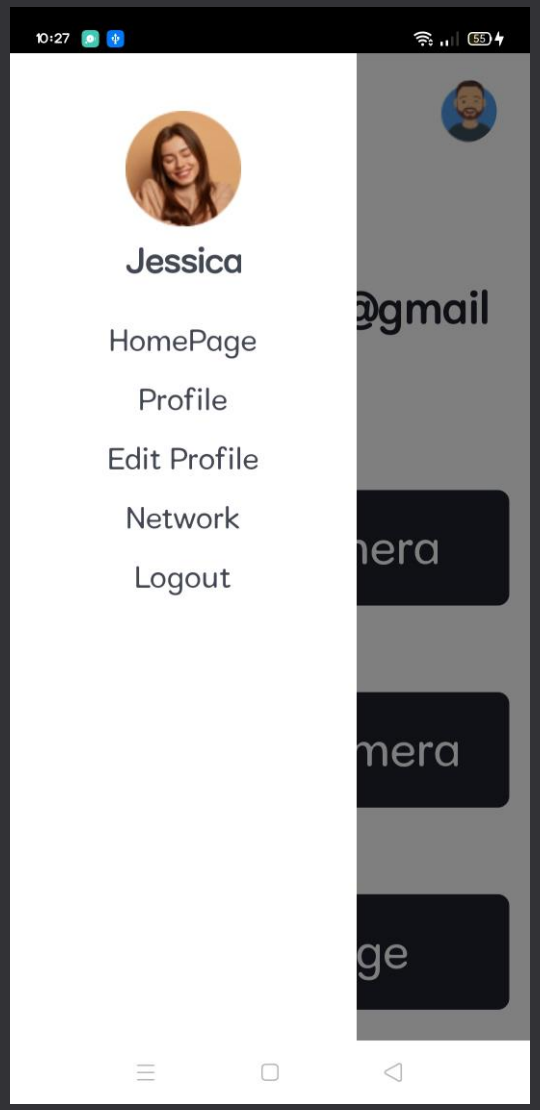


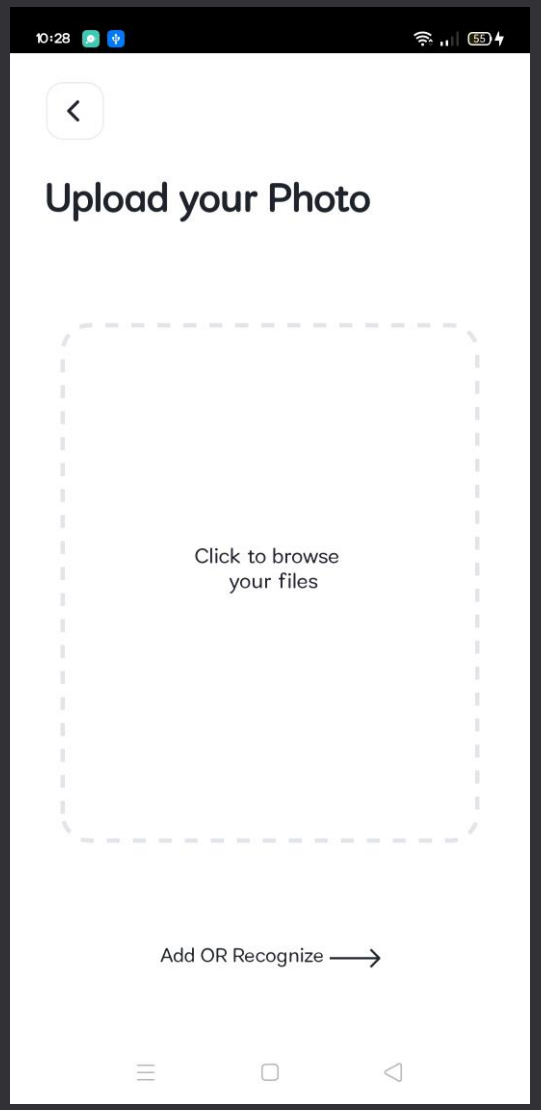




Results

B. Samples from FaceReminder application:





8

Future Work

Future Work

- Linking the recognition techniques with the mobile interface
- Increasing the used database.
- Making the user interface meet the human factors.
- The mobile application will be connected to a Bluetooth camera that will be optional for each user.

Questions?

Thank you