Implementation of algorithms for face detection on single board computers

Agron Sinani
Ardian Loku
Thang Tran

Structure

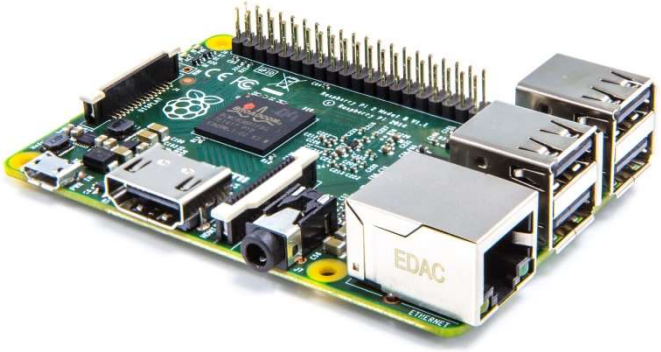
- Introduction
- Theoretical basics
- Face detection
- Face preprocessing
- Face recognition
- Program
- Conclusion
- Demonstration

Introduction

- Tasks
  - Implementing face detection and recognition algorithms
  - Using OpenCV & Raspberry Pi
  - Running on linux-based-OS
  - Write in C++
  - Documentation and schedule

Theoretical Basic
Raspberry Pi

- Single board computer
  - One single circuit board
  - Common and famous example is the Pi
  - 700 MHz single-core/900MHz quad-core CPU
  - 512MB/1GB RAM
  - Several in- and output interfaces
  - Wide range of use



Theoretical Basic
OpenCV

- Open source library “OpenCV“
  - Open source computer vision
  - Started 1999 at Intel in Russia
  - BSD-license
  - Enabled fast/real-time image processing

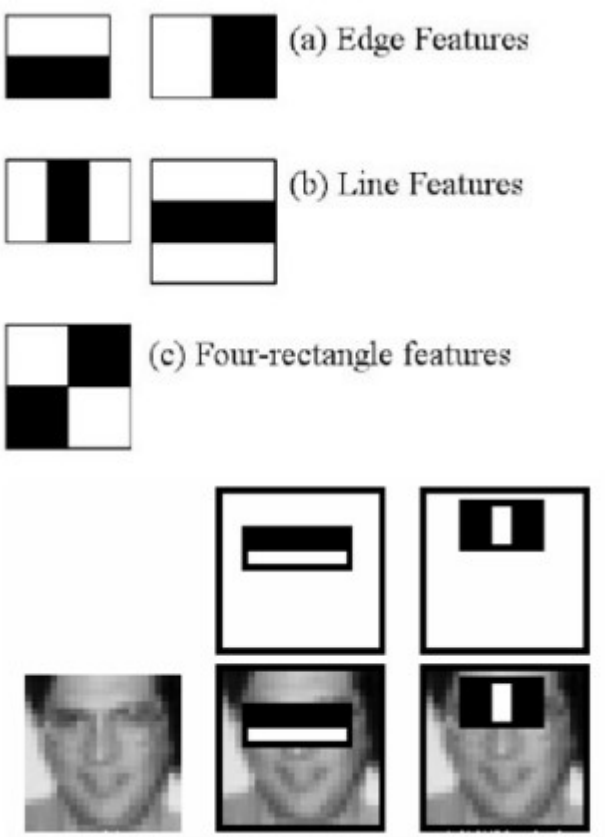
- Programming examples
  - Face detection & recognition
  - Red eyes remover
  - Identification of objects

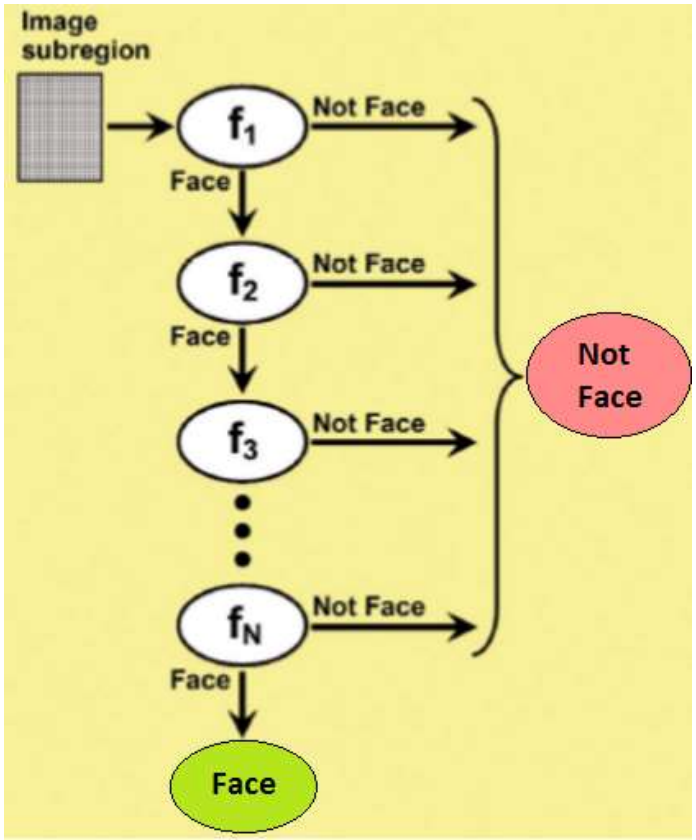
Face detection

- Cascade classifiers
  - Haar cascades
  - Local Binary Pattern (LBP) cascades
- Basic steps in face detection
  - Grayscale color conversion
  - Shrinking the camera image
  - Histogram equalization
  - Detecting the face

Face detection

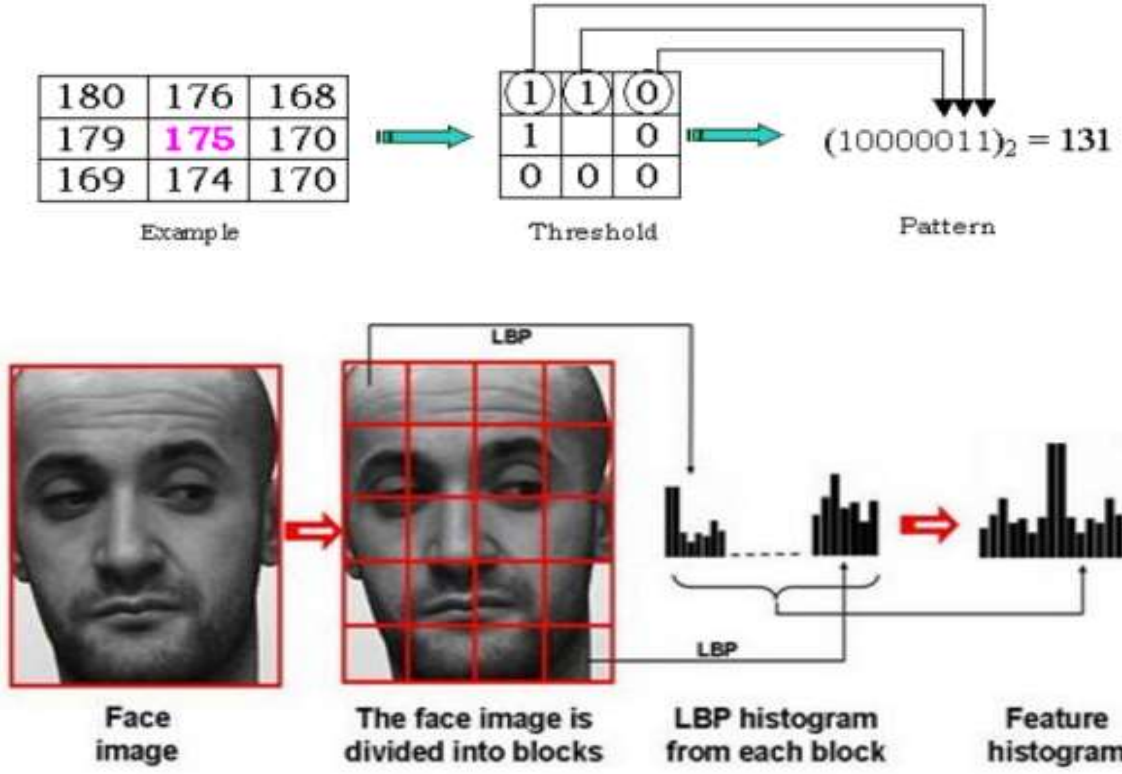
- Haar cascades





Face detection

- Local Binary Pattern (LBP) cascades



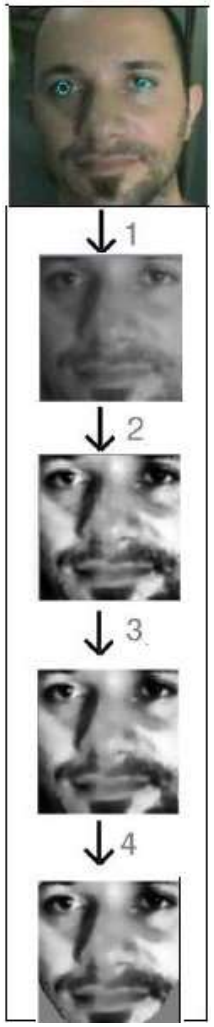
Face detection

- Basic steps in face detection
  - Grayscale color conversion

  - Shrinking the camera image
  - Histogram equalization

Face preprocessing

- Factors which hamper the face recognition
  - Picture quality
  - Lighting
  - Facial expression/orientation
- Preprocessing
  - Additionally to face, program needs eyes
  - Geometrical transformation (Figure 1)
  - Histogram equalization (Figure 2)
  - Bilateral filter to smooth image (Figure 3)
  - Elliptical mask (Figure 4)

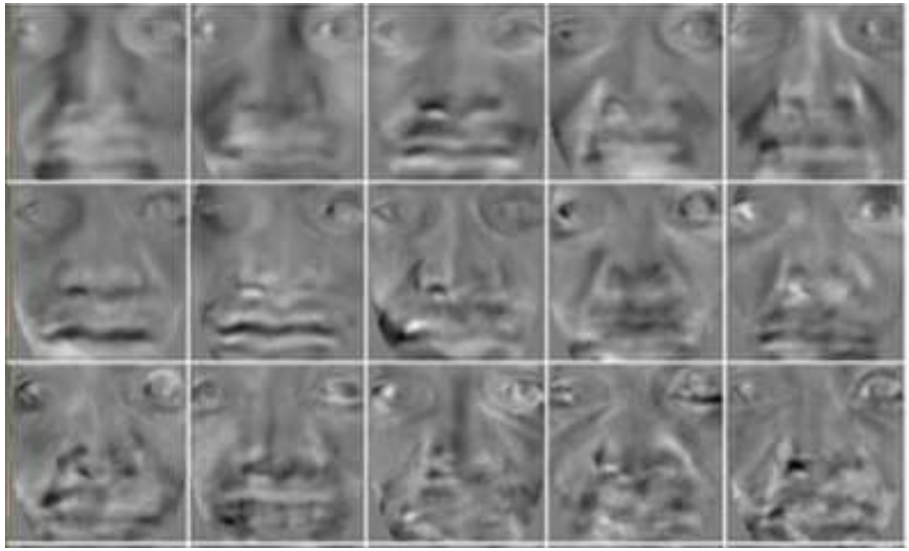


Face recognition

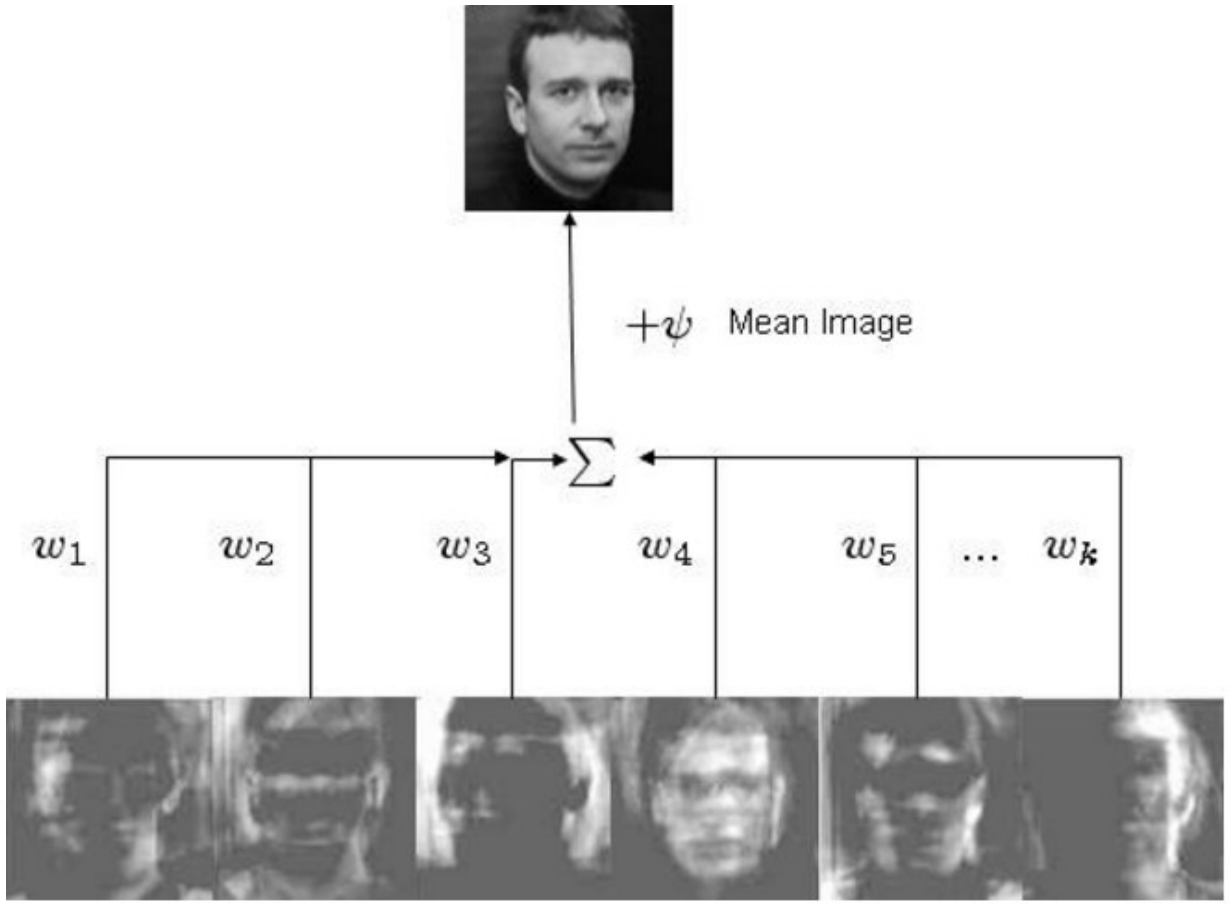
- Face recognition algorithms
  - Eigenface algorithm
  - Fisherface algorithm
- Basic steps in face recognition
  - Face identification
  - Face verification

Face recognition

- Eigenface vs Fisherface

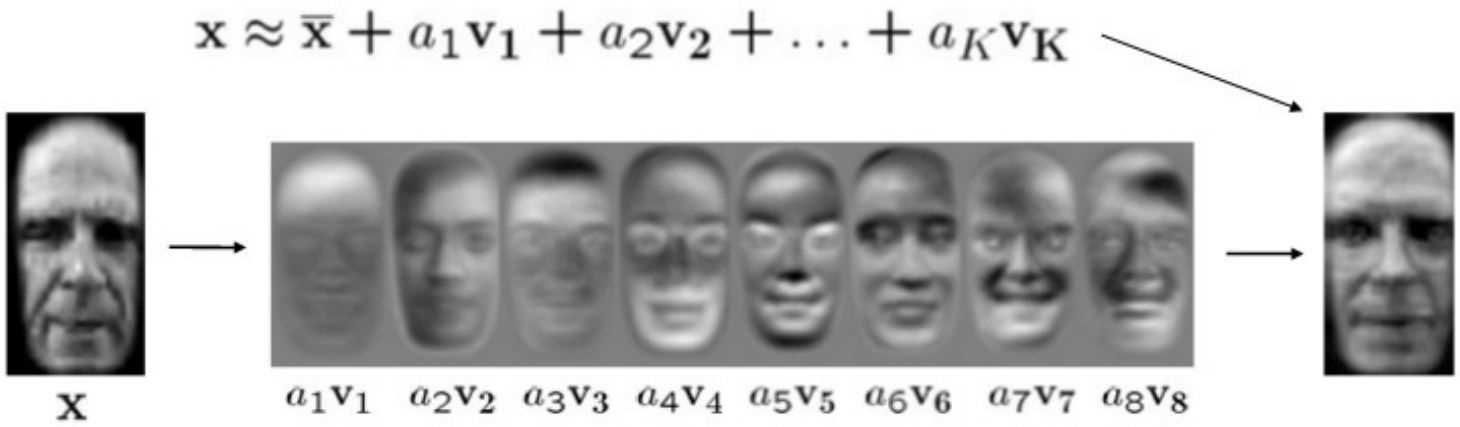


Face recognition

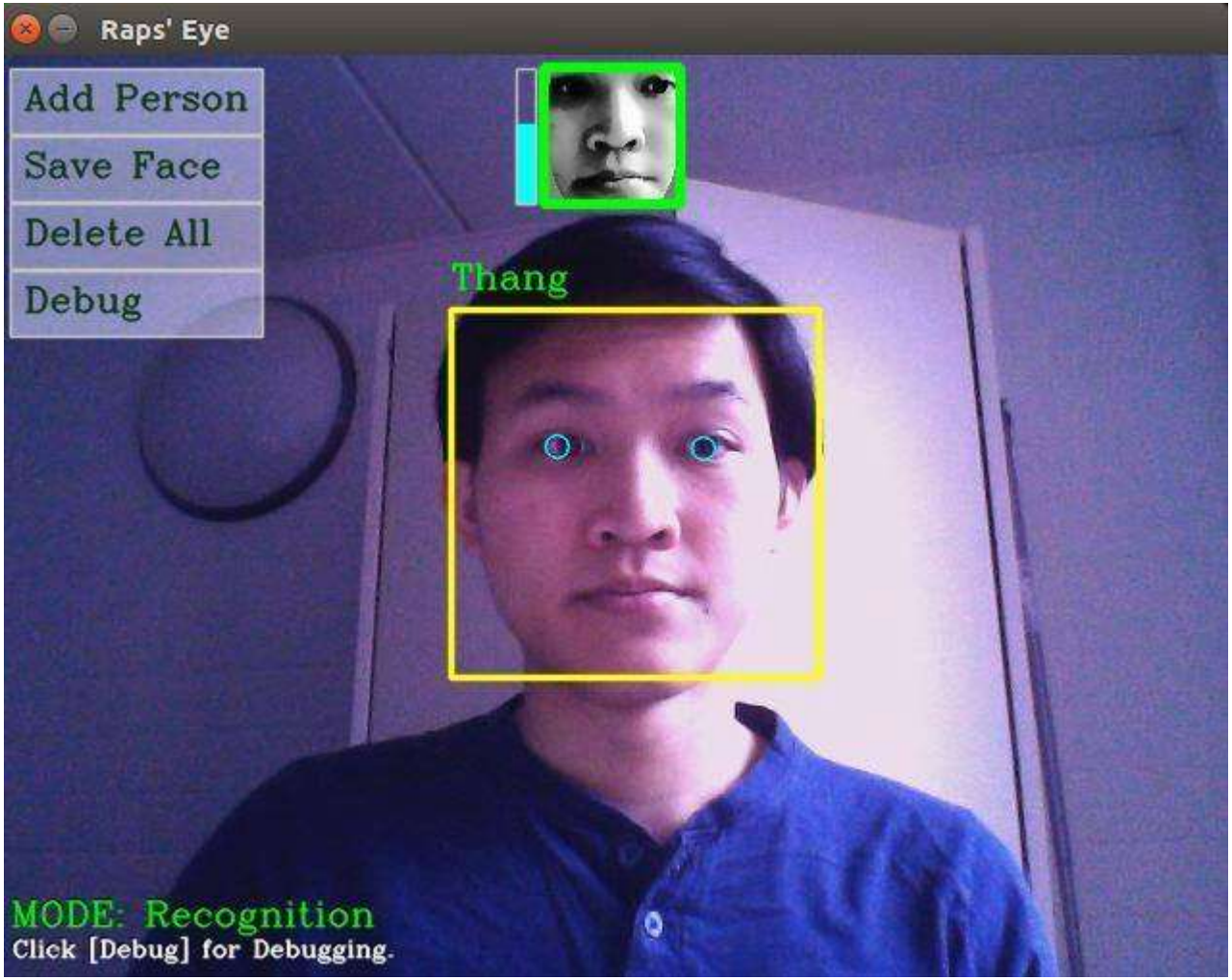


Face recognition

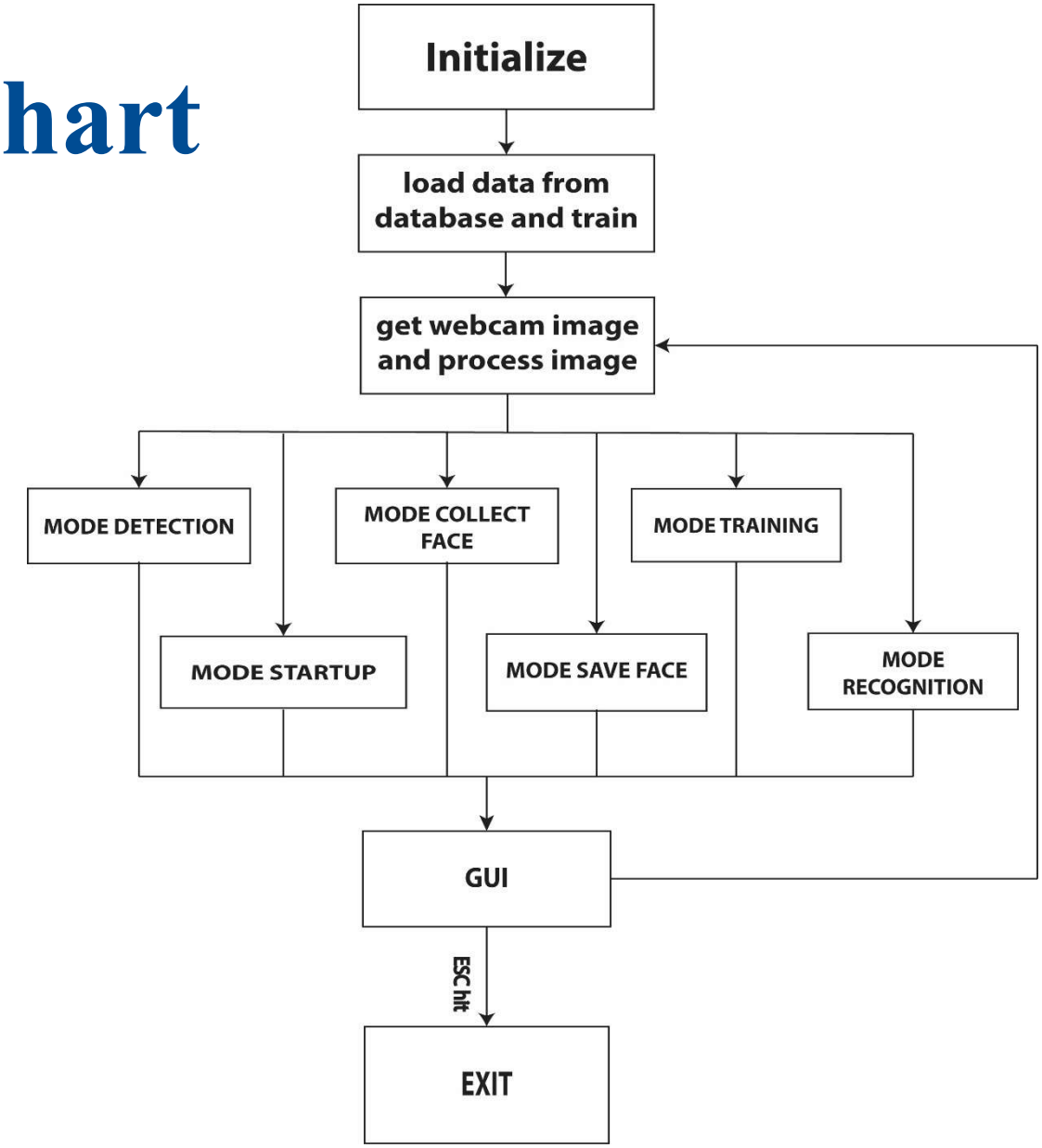
- Face identification
- Face verification



GUI


Raps' Eye
Add Person
Save Face
Delete All
Debug
Thang
MODE: Recognition
Click [Debug] for Debugging.

Flow chart


Initialize
load data from database and train
get webcam image and process image
MODE DETECTION
MODE COLLECT FACE
MODE TRAINING
MODE STARTUP
MODE SAVE FACE
MODE RECOGNITION
GUI
ESC hit
EXIT

Conclusion

- Aim of this project
  - Implement face recognition algorithms to Rapsberry Pi
  - Retrieve faces‘ data from a host PC
- Problem
  - Delay between streaming frames due to lack of computing power
- Solutions
  - Deactivate the streaming window
  - Move to a better system

Thank you for your interest!