Technological University ThanLyin (TTU)

Department of Information Technology Engineering

2022-2023 Academic Year

IT-51028 Integrated Design Project

Face Recognition System (Group 1 Project)

Presented by : Group 1

Supervisor : Daw Sabai Oo

Seminar : Defense Seminar

Date : 9.10.2023

Group Members List

Han Min Oo	VIT- 4R	[Leader]
Khin Ei Ei Htun	VIT- 7R	[Searching information & Help Coding]
Khin Phu Pwint Kyaw	VIT- 8R	[Searching information & Help Coding]
Khin Thuzar Moe	VIT- 14R	[Preparing book & User Testing]
Thazin Tun	VIT- 15R	[Preparing book & User Testing]
Shwe Yi Win	VIT- Ext3	[Checking Grammar & Printing]

Outline

- Introduction of face recognition project
- Abstract
- Aim & Objectives
- Python Programming Language
- Block diagram
- Flow chart
- Explanation of codes
- Formula
- LBPH Algorithm
- Improving LBPH Face Recognition Accuracy with Euclidean Distance
- Output screenshots of Face Recognition Projects
- Log file of Face Recognition System
- Benefit of face recognition system
- Conclusion

Introduction of Face Recognition System

- Face recognition systems are part of facial image processing applications and their significance as a research area are increasing recently
- It is a combination of face detection and recognition techniques in image analyzes
- Detection application is used to find position of the faces in a given image.
- Recognition algorithm is used to classify given images with known structured properties, which are used commonly in most of the computer vision applications.

Abstract

- This research project involves discovering how a Face Recognition based on AI .
- This has been done by examining situations such as training and testing on a small self-accumulated database, clutter, variations in background, noise, occlusion, computing Requirements, etc.
- In order to test all we assembled a database of approximately 30 Images per person.
- It can also be deployed in police station to identify and verify the criminals. Face detection has gained a lot of importance especially in fields like photography, security, and marketing

Aim & Objectives

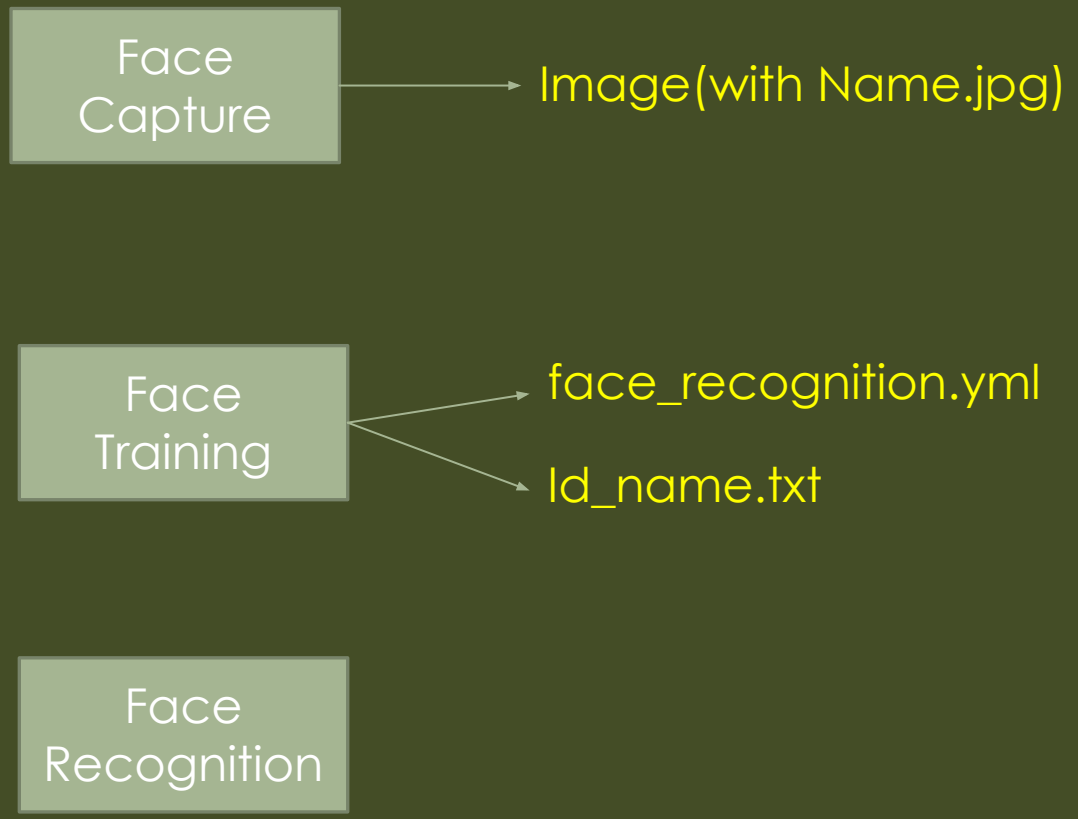
- To prevent the use of digital identities for criminal purposes
- To unlocking features in new phones requiring face ID
- To design software that can detect human faces from an image.
- In educational institutions or workplaces, face recognition can be used to automate attendance tracking
- Face recognition can be employed in medical fields for patient identification, monitoring, and disease diagnosis.

Python Programming Language

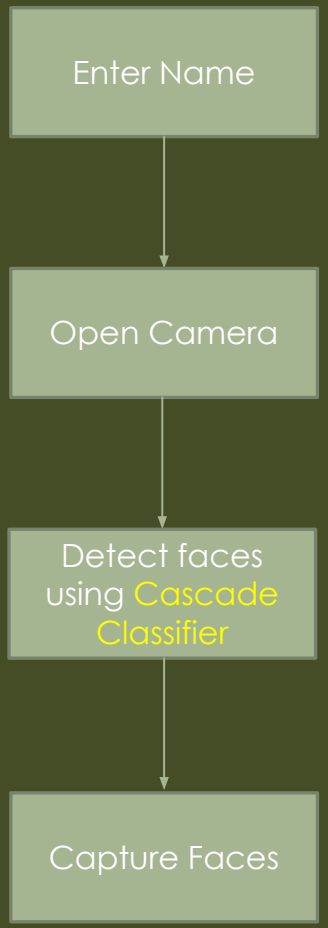
Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation via the off-side rule. Python is dynamically typed and garbage-collected.

Faces are made of thousands of fine lines and features that must be matched. The face recognition using Python, **break the task of identifying the face into thousands of smaller, bite-sized tasks**, each of which is easy to face Recognition Python is the latest trend in Machine Learning techniques.

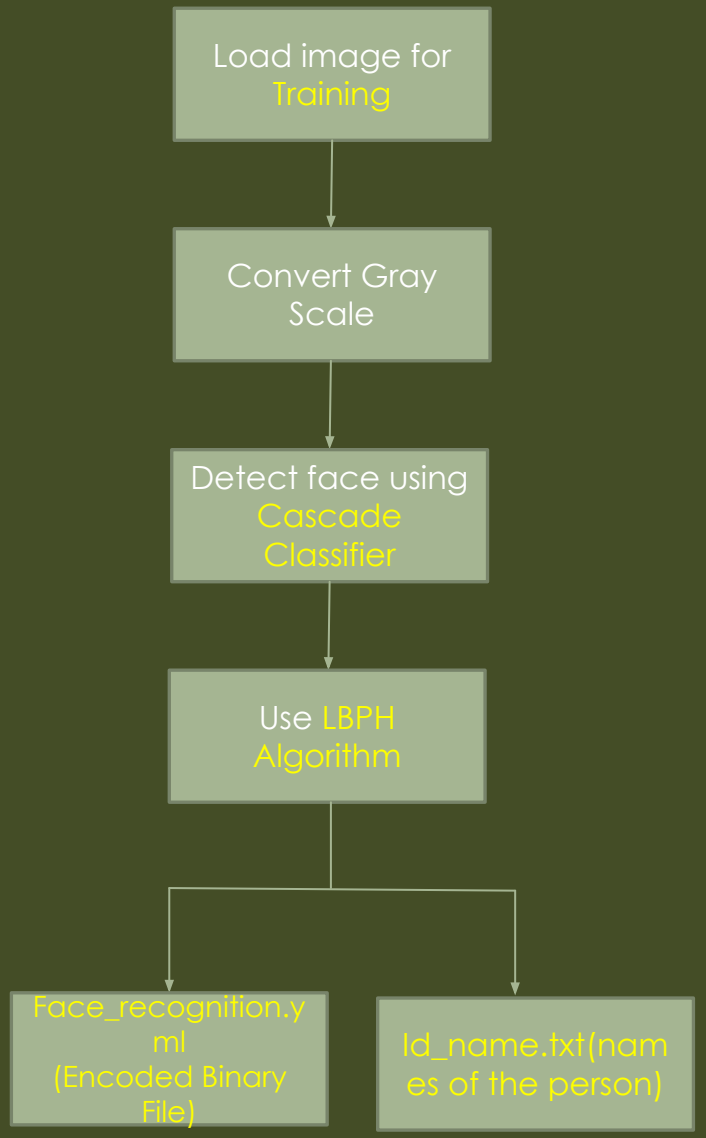
Block diagram of face recognition system



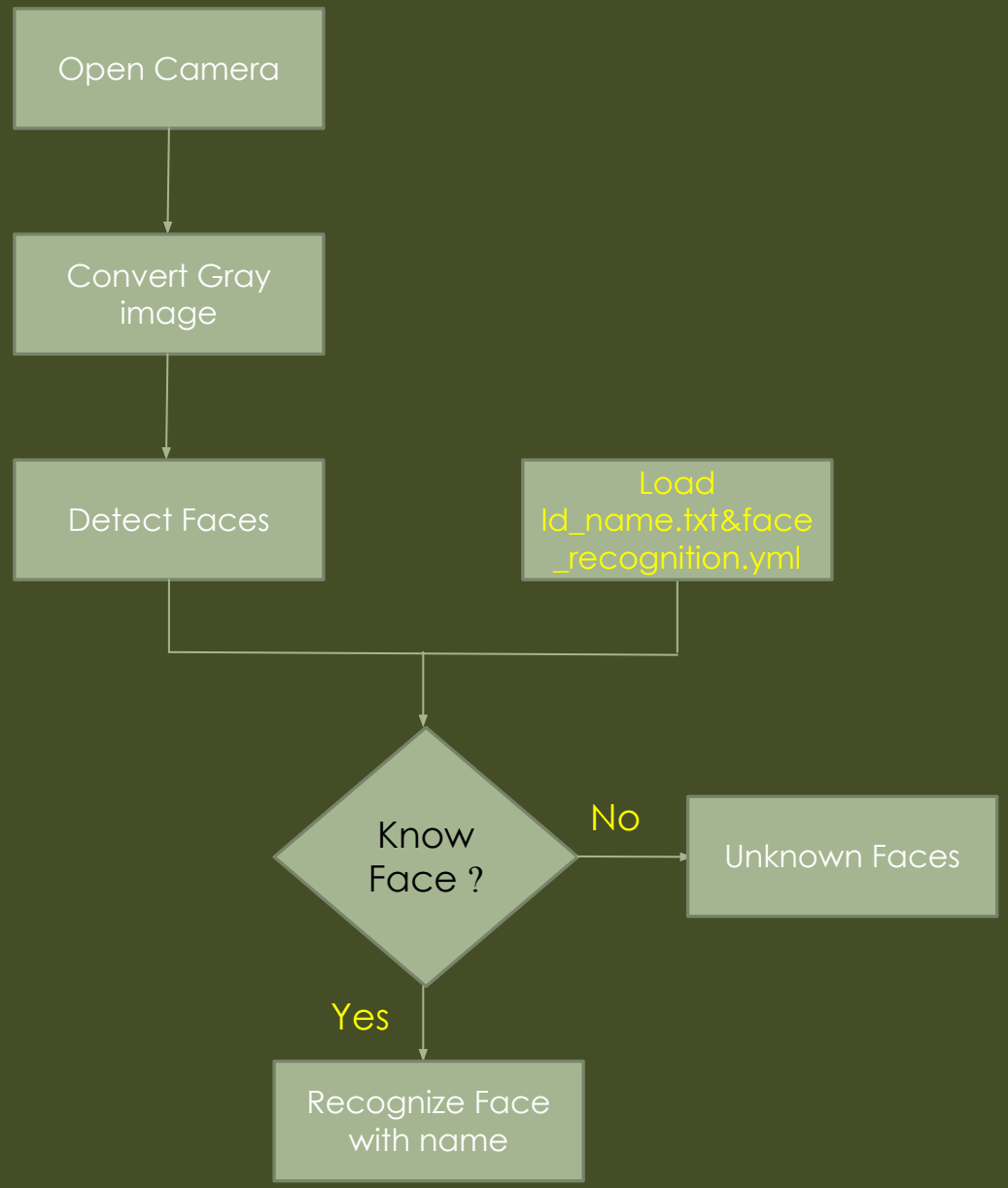
(1)
Face Capture



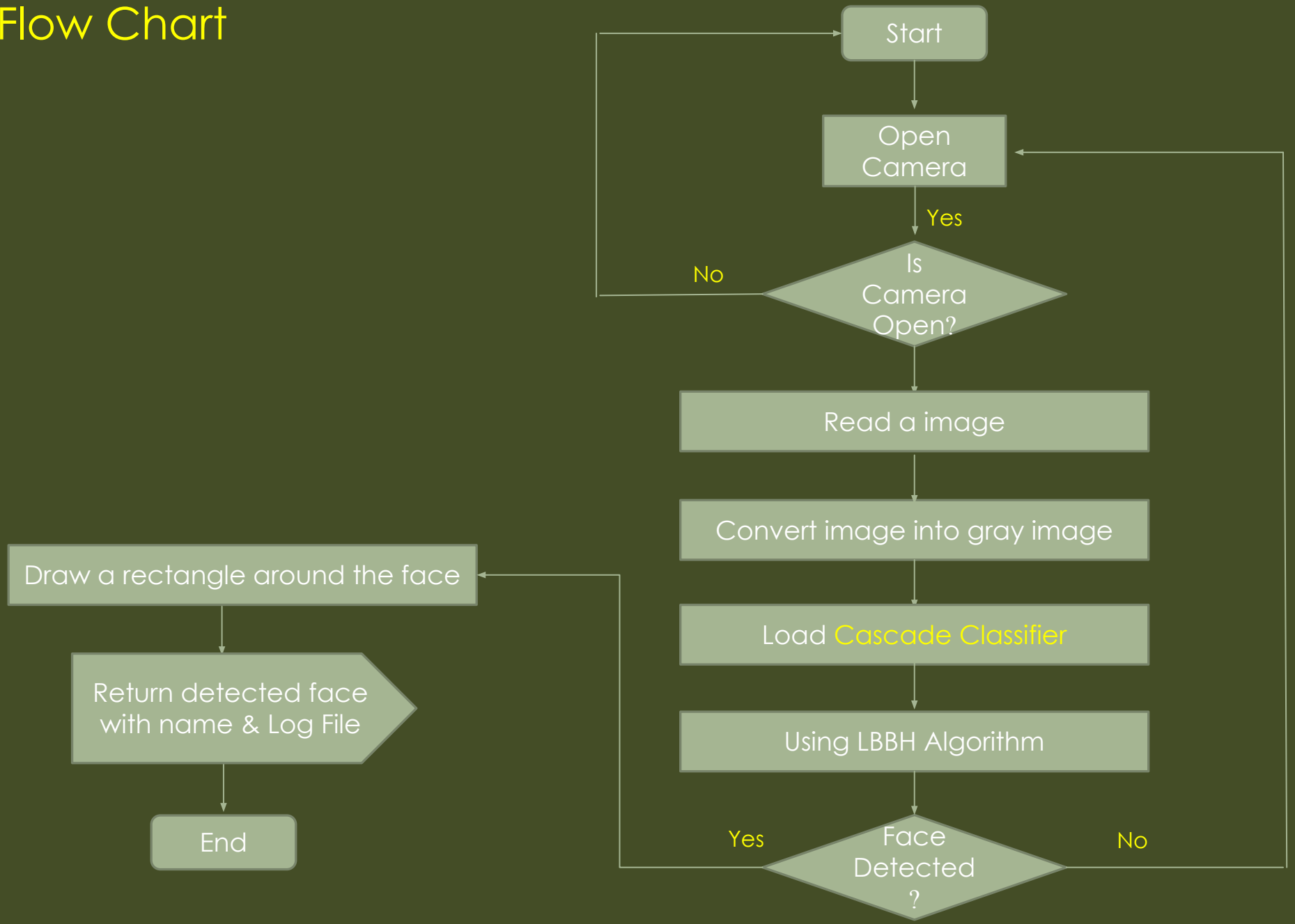
(2)
Face Training



(3)
Face Recognition



Flow Chart


Start
Open Camera
Yes
No
Is Camera Open?
Read a image
Convert image into gray image
Load Cascade Classifier
Using LBBH Algorithm
Yes
Face Detected ?
No
Draw a rectangle around the face
Return detected face with name & Log File
End

Detection

Face Recognition is different from face detection. In face detection, we had only detected the location of human faces, and we recognized the identity of faces in the face recognition task. Generally, face recognizers that are based on landmarks take face images and try to find essential feature points such as eyebrows, corners of the mouth, eyes, nose, lips, etc. There are more than 60 points.

```
Enter the user name : Han Min Oo
Han Min Oo is already taken!
Enter the user name : khin
Creating Image...../Image/Khin /Khin 1.jpg
Creating Image...../Image/Khin /Khin 2.jpg
Creating Image...../Image/Khin /Khin 3.jpg
Creating Image...../Image/Khin /Khin 4.jpg
Creating Image...../Image/Khin /Khin 5.jpg
Creating Image...../Image/Khin /Khin 6.jpg
Creating Image...../Image/Khin /Khin 7.jpg
```

Capture image from OpenCV & store in related folder

```python
# To capture video from webcam.
cap = cv2.VideoCapture(0)
# To use a video file as input
# cap = cv2.VideoCapture('filename.mp4')

while True:
    # Read the frame
    _, img = cap.read()

    # Convert to grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Detect the faces
    faces = face_cascade.detectMultiScale(gray, 1.1, 4)

    # Draw the rectangle around each face
    for (x, y, w, h) in faces:
        cv2.rectangle(gray, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

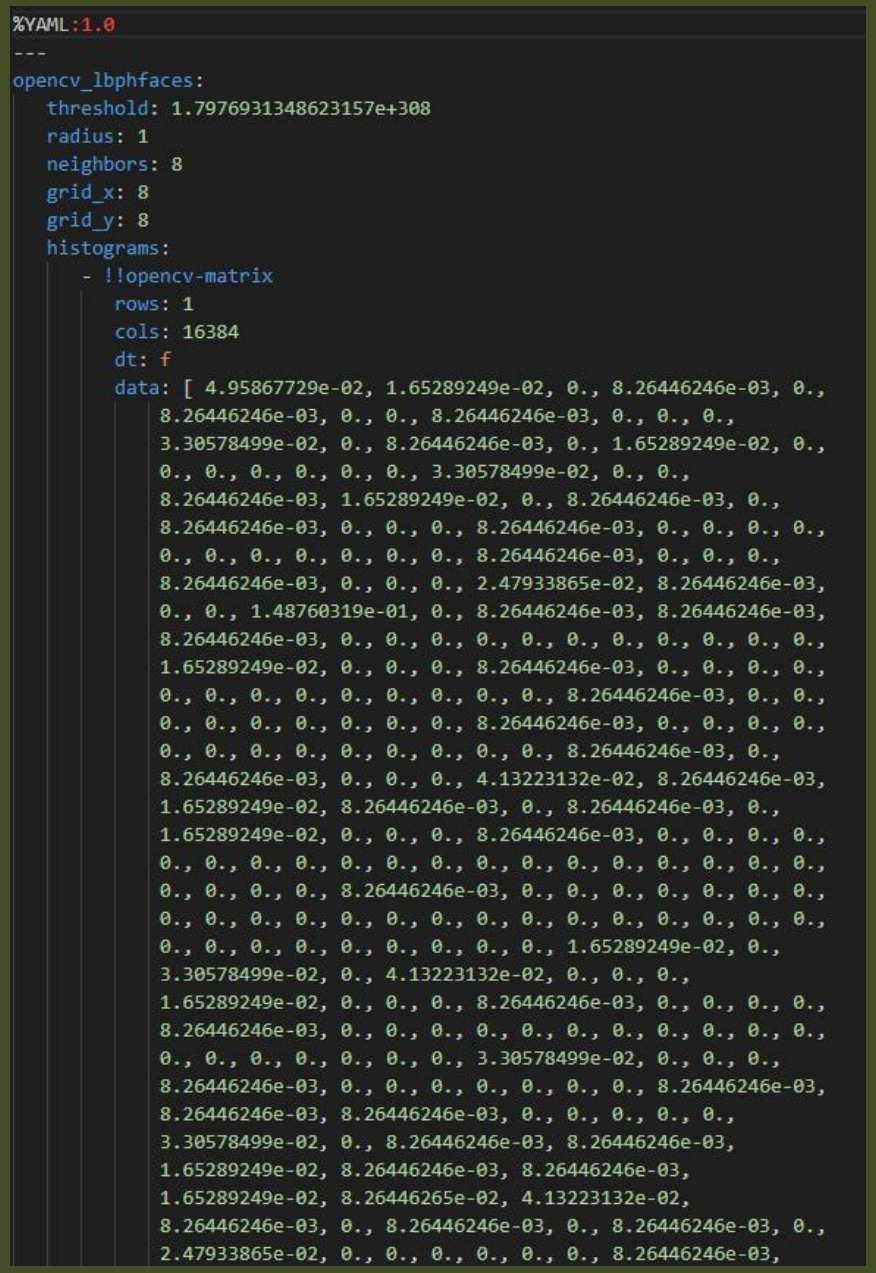
Convert image to gray scale image

```python
import cv2
import numpy as np
from PIL import Image
import os

# Parent path for face image database
parent_path = 'Image'

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
```

LBPH Algorithm using in our project

Face_recognition.yml File in our project

Face recognition system that needs to store information about individuals' facial encodings
(unique numerical representations of their faces).



```
%YAML:1.0
---
opencv_lbphfaces:
   threshold: 1.7976931348623157e+308
   radius: 1
   neighbors: 8
   grid_x: 8
   grid_y: 8
   histograms:
      - !!opencv-matrix
         rows: 1
         cols: 16384
         dt: f
         data: [ 4.95867729e-02, 1.65289249e-02, 0., 8.26446246e-03, 0.,
             8.26446246e-03, 0., 0., 8.26446246e-03, 0., 0., 0.,
             3.30578499e-02, 0., 8.26446246e-03, 0., 1.65289249e-02, 0.,
             0., 0., 0., 0., 0., 0., 3.30578499e-02, 0., 0.,
             8.26446246e-03, 1.65289249e-02, 0., 8.26446246e-03, 0.,
             8.26446246e-03, 0., 0., 0., 8.26446246e-03, 0., 0., 0., 0.,
             0., 0., 0., 0., 0., 0., 0., 8.26446246e-03, 0., 0., 0.,
             8.26446246e-03, 0., 0., 0., 2.47933865e-02, 8.26446246e-03,
             0., 0., 1.48760319e-01, 0., 8.26446246e-03, 8.26446246e-03,
             8.26446246e-03, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
             1.65289249e-02, 0., 0., 0., 8.26446246e-03, 0., 0., 0., 0.,
             0., 0., 0., 0., 0., 0., 0., 0., 0., 8.26446246e-03, 0., 0.,
             0., 0., 0., 0., 0., 0., 0., 8.26446246e-03, 0., 0., 0., 0.,
             0., 0., 0., 0., 0., 0., 0., 0., 8.26446246e-03, 0.,
             8.26446246e-03, 0., 0., 0., 4.13223132e-02, 8.26446246e-03,
             1.65289249e-02, 8.26446246e-03, 0., 8.26446246e-03, 0.,
             1.65289249e-02, 0., 0., 0., 8.26446246e-03, 0., 0., 0., 0.,
             0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
             0., 0., 0., 0., 8.26446246e-03, 0., 0., 0., 0., 0., 0., 0.,
             0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
             0., 0., 0., 0., 0., 0., 0., 0., 1.65289249e-02, 0.,
             3.30578499e-02, 0., 4.13223132e-02, 0., 0., 0.,
             1.65289249e-02, 0., 0., 0., 8.26446246e-03, 0., 0., 0., 0.,
             8.26446246e-03, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
             0., 0., 0., 0., 0., 0., 0., 3.30578499e-02, 0., 0., 0.,
             8.26446246e-03, 0., 0., 0., 0., 0., 0., 0., 8.26446246e-03,
             8.26446246e-03, 0., 8.26446246e-03, 0., 0., 0., 0., 0.,
             3.30578499e-02, 0., 8.26446246e-03, 8.26446246e-03,
             1.65289249e-02, 8.26446246e-03, 8.26446246e-03,
             1.65289249e-02, 8.26446265e-02, 4.13223132e-02,
             8.26446246e-03, 0., 8.26446246e-03, 0., 8.26446246e-03, 0.,
             2.47933865e-02, 0., 0., 0., 0., 0., 0., 8.26446246e-03,
```

Testing Code

```python
print("\n [INFO] Loading and splitting the data...")
faces, ids, id_dict = getImagesAndLabels(parent_path)

# Split the data into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(faces, ids, test_size=0.2, random_state=42)

print("\n [INFO] Training faces. It will take a few seconds. Wait ...")
recognizer.train(X_train, np.array(y_train))

# Save the model into trainer/ directory
trainer_dir = 'trainer/'
os.makedirs(trainer_dir, exist_ok=True)
recognizer.save(os.path.join(trainer_dir, 'face_recognition.yml'))

# Print the number of faces trained
print("\n [INFO] {0} faces trained.".format(len(np.unique(ids))))

# Save ID names to a text file
with open('trainer/id_names.txt', 'w') as file:
    for key, value in id_dict.items():
        file.write(f"{value}: {key}\n")



# Testing phase
print("\n [INFO] Testing the trained model.")
correct_predictions = 0
total_predictions = len(X_test)

for i in range(len(X_test)):
    label, confidence = recognizer.predict(X_test[i])
    predicted_name = [name for name, id in id_dict.items() if id == label][0]
    true_name = [name for name, id in id_dict.items() if id == y_test[i]][0]

    if label == y_test[i]:
        correct_predictions += 1

    confidence_percentage = round(100 - confidence, 2)
    print(f"Test {i + 1}: Predicted Name: {predicted_name}, True Name: {true_name}, Confidence: {confidence_percentage:.2f}%")

accuracy = (correct_predictions / total_predictions) * 100
print("\n [INFO] Testing completed.")
print(f"Accuracy: {accuracy:.2f}%")
```

Testing Data Result

```
Test 79: Predicted Name: Susandi, True Name: Susandi, Confidence: 38.89%
Test 80: Predicted Name: Testing User, True Name: Testing User, Confidence: 74.38%
Test 81: Predicted Name: Testing User, True Name: Testing User, Confidence: 75.02%
Test 82: Predicted Name: Thuzar, True Name: Thuzar, Confidence: 59.48%
Test 83: Predicted Name: Testing User, True Name: Testing User, Confidence: 74.72%
Test 84: Predicted Name: Thuzar, True Name: Thuzar, Confidence: 50.95%
Test 85: Predicted Name: Susandi, True Name: Susandi, Confidence: 38.40%
Test 86: Predicted Name: Susandi, True Name: Susandi, Confidence: 40.31%
Test 87: Predicted Name: Susandi, True Name: Susandi, Confidence: 45.25%
Test 88: Predicted Name: Thuzar, True Name: Thuzar, Confidence: 54.60%
Test 89: Predicted Name: Susandi, True Name: Susandi, Confidence: 40.77%
Test 90: Predicted Name: Testing User, True Name: Testing User, Confidence: 77.09%
Test 91: Predicted Name: Susandi, True Name: Susandi, Confidence: 23.07%
Test 92: Predicted Name: Thuzar, True Name: Thuzar, Confidence: 57.58%
Test 93: Predicted Name: Testing User, True Name: Testing User, Confidence: 72.05%
Test 94: Predicted Name: Testing User, True Name: Testing User, Confidence: 75.38%
Test 95: Predicted Name: Thuzar, True Name: Thuzar, Confidence: 57.94%
Test 96: Predicted Name: Susandi, True Name: Susandi, Confidence: 22.69%

 [INFO] Testing completed.
Accuracy: 98.96%

DELL@DELLv3578-8thgen-win10-8013 MINGW64 /d/MyPython/facedetection (master)
```

Formula used in class to calculate Confidence is

Distance (A, B) = √((A1 - B1)^2 + (A2 - B2)^2 + ... + (An - Bn)^2)

Where:

- A1, A2, ..., An are the elements of the feature vector A.
- B1, B2, ..., Bn are the elements of the feature vector B.
- n is the number of elements in the feature vectors (dimensionality).

Improving LBPH Face Recognition Accuracy with Euclidean Distance

A higher accuracy value indicates that the model is making more accurate predictions, while a lower accuracy value indicates that the model is making more errors.

Formula used in class to calculate Confidence is

Euclidean Distance (A, B) = √((A1 - B1)^2 + (A2 - B2)^2 + ... + (An - Bn)^2)

Where:

- A1, A2, ..., An are the elements of the feature vector A.
- B1, B2, ..., Bn are the elements of the feature vector B.
- n is the number of elements in the feature vectors (dimensionality).

The formula for Accuracy is:

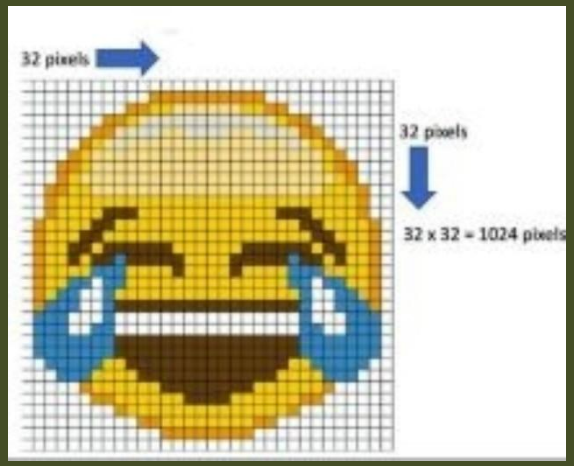
Accuracy = (Number of Correct Predictions) / (Total Number of Predictions) * 100

LBPH Algorithm

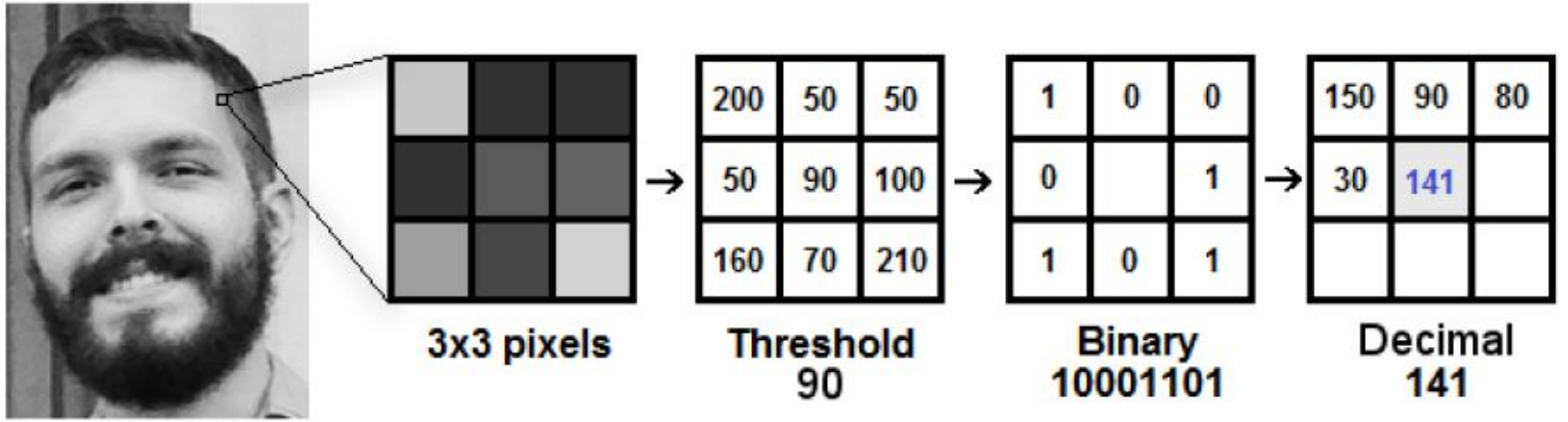
- LBPH (Local Binary Pattern Histogram) is a Face-Recognition algorithm it is used to recognize the face of a person. It is known for its performance and how it is able to recognize the face of a person from front face.

Images & Pixel

- **For every image, the value of pixels ranges between 0 to 255.**
- **This image here is 32 pixels wide and 32 pixels high.**
- **And when we multiply 32 by 32, the result is 1024, which is the total number of pixels in the image.**

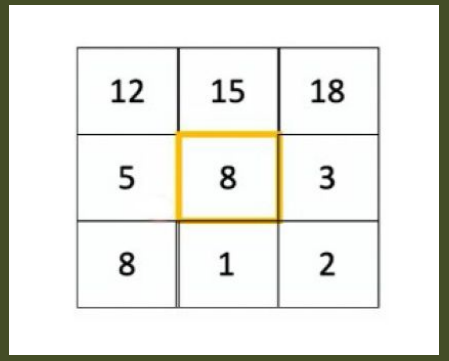


Understanding Face Recognition using LBPH Algorithm



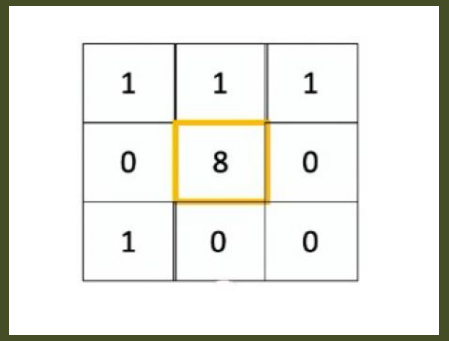
LBPH(Local Binary Patterns Histograms)

Step(1)

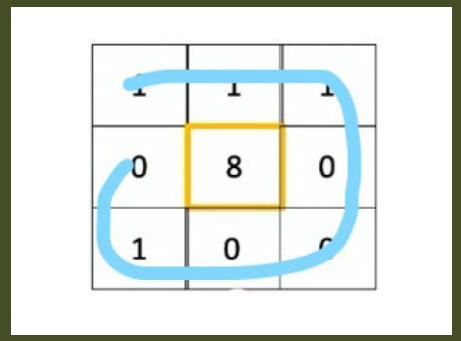


- In this example, we have three rows and three columns and the total number of pixels is nine. Let's select the central pixel here, value eight, and apply a condition.
- If the value is greater or equal to 8, the result is '1' otherwise, if the value is less than eight, the result is zero.

Step(2)



- The center element of the matrix.

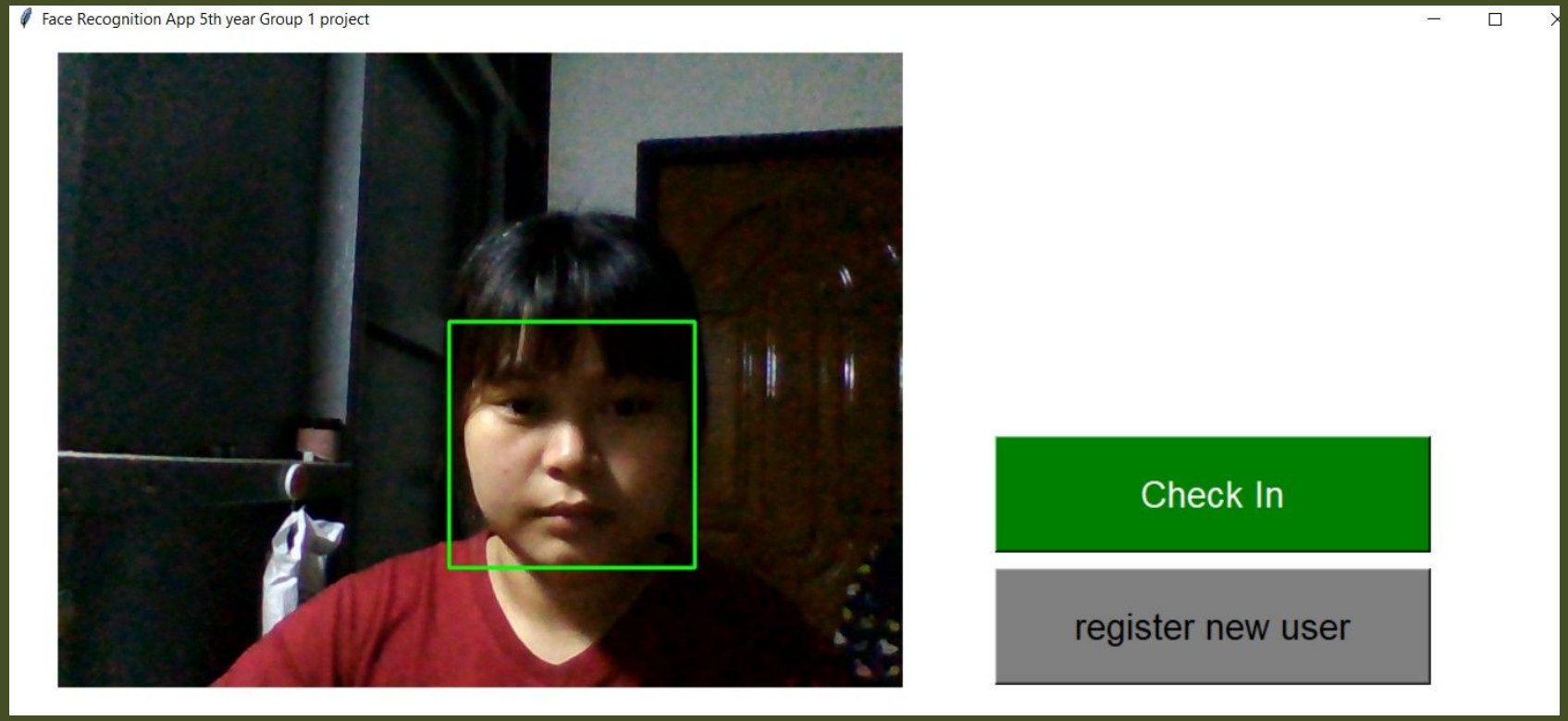


Binary value = 11100010 .
Decimal Value = 226

Higher the values the brighter the image and when values are lower darker the image will be. For this reason, this algorithm has good results in light and dark image because when the image becomes lighter or darker, all the pixels in the neightbourhood here will be changed.

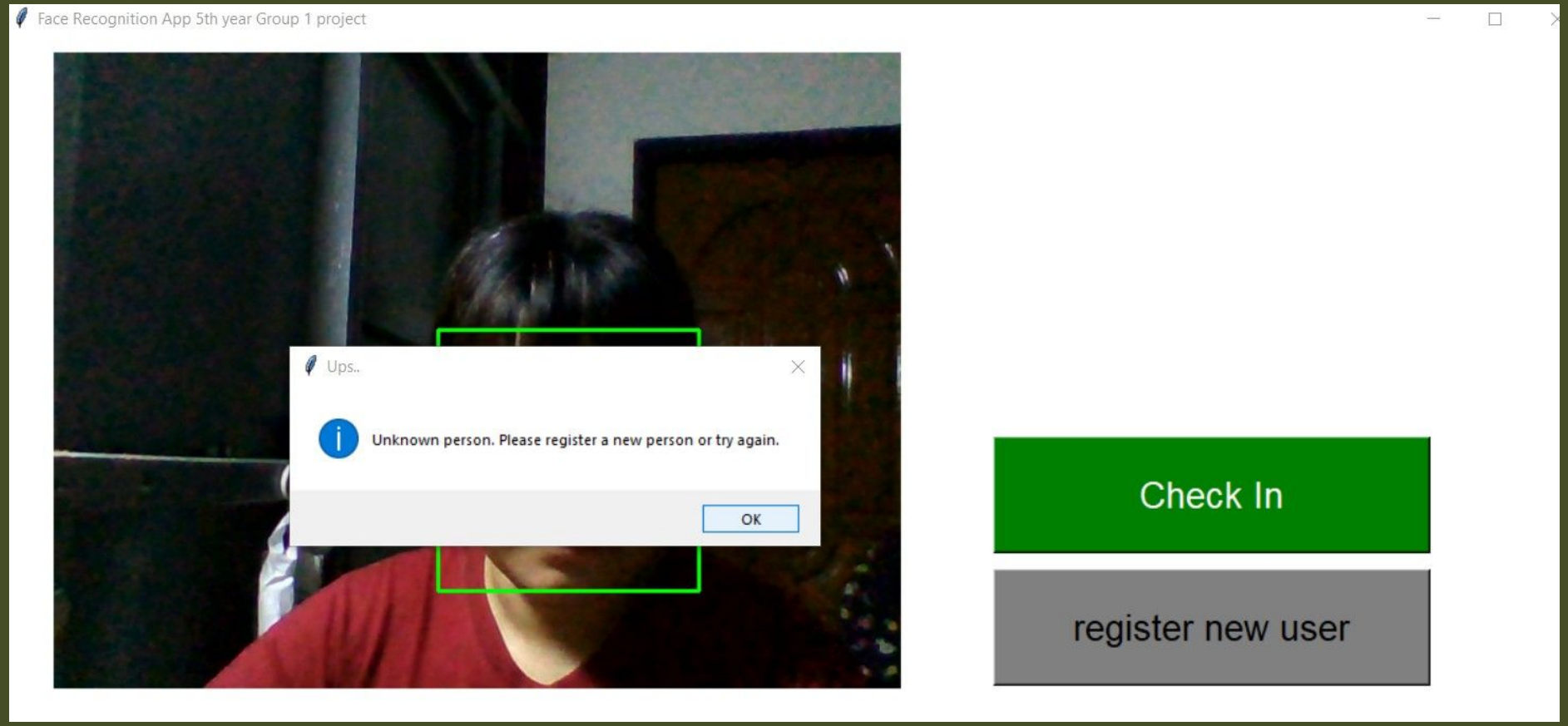
Output Screenshots of Face Recognition System

(1) Open Camera



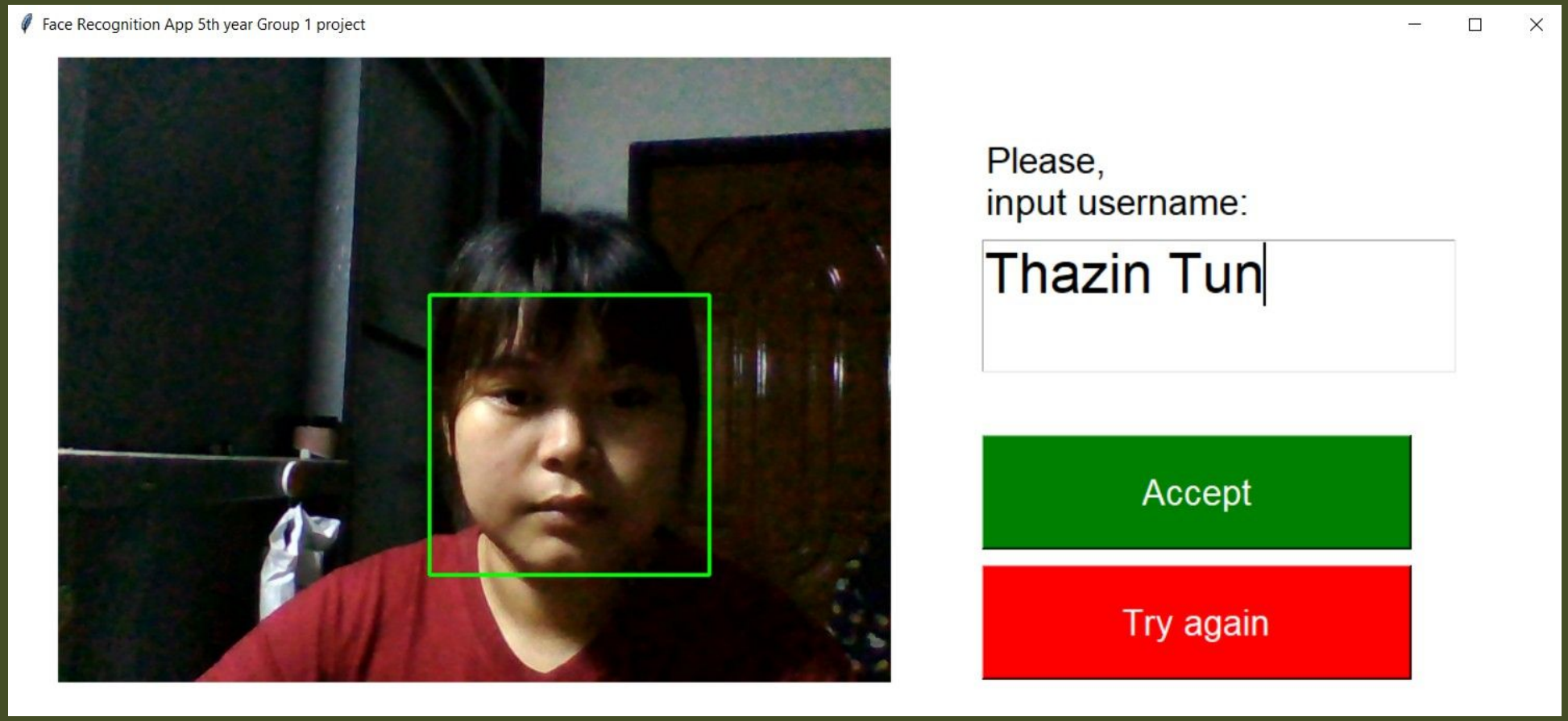
Output Screenshots of Face Recognition System

(2) If you're not register and you click check in button, then show you alert box.



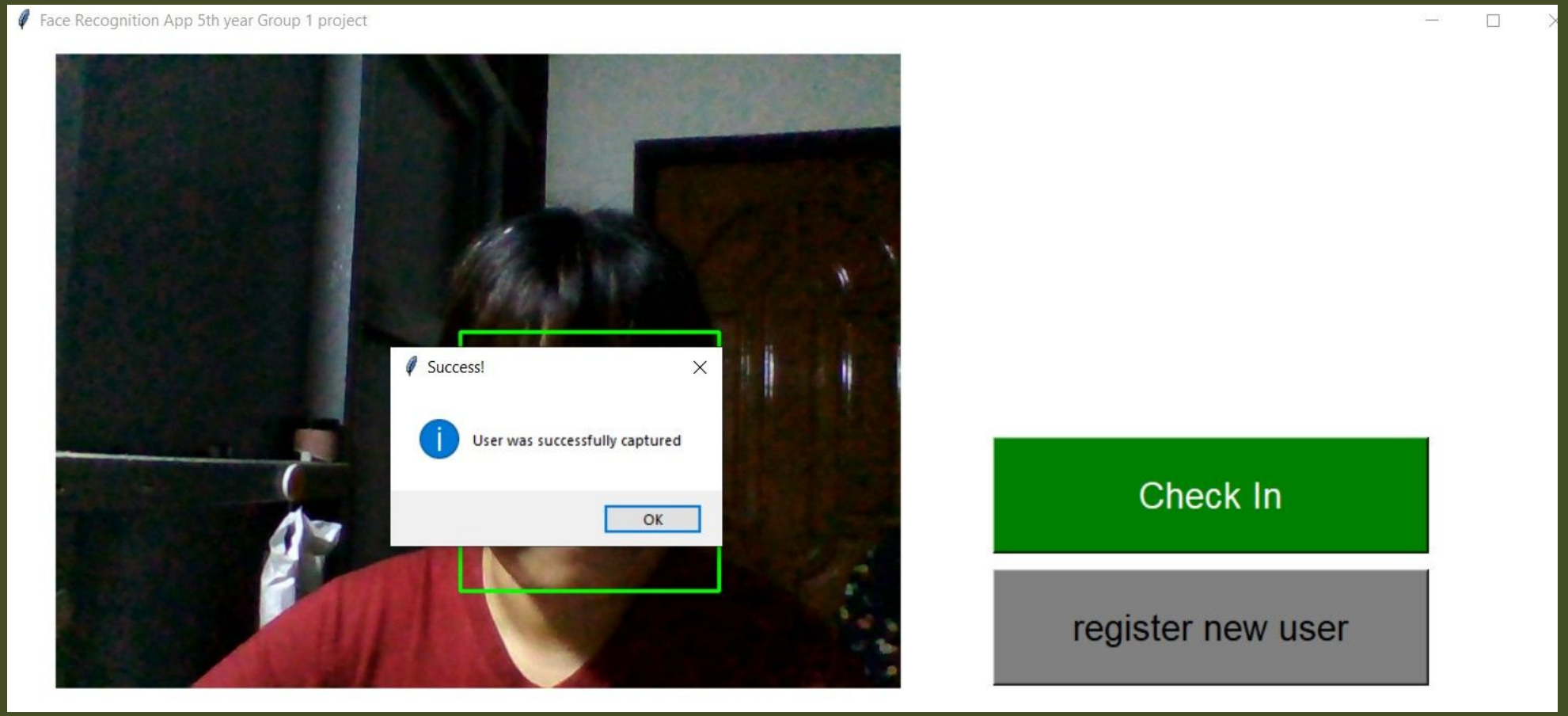
Output Screenshots of Face Recognition System

(3) After click register new user button, you need to enter username and click Accept button. Then you arrive the first stage .



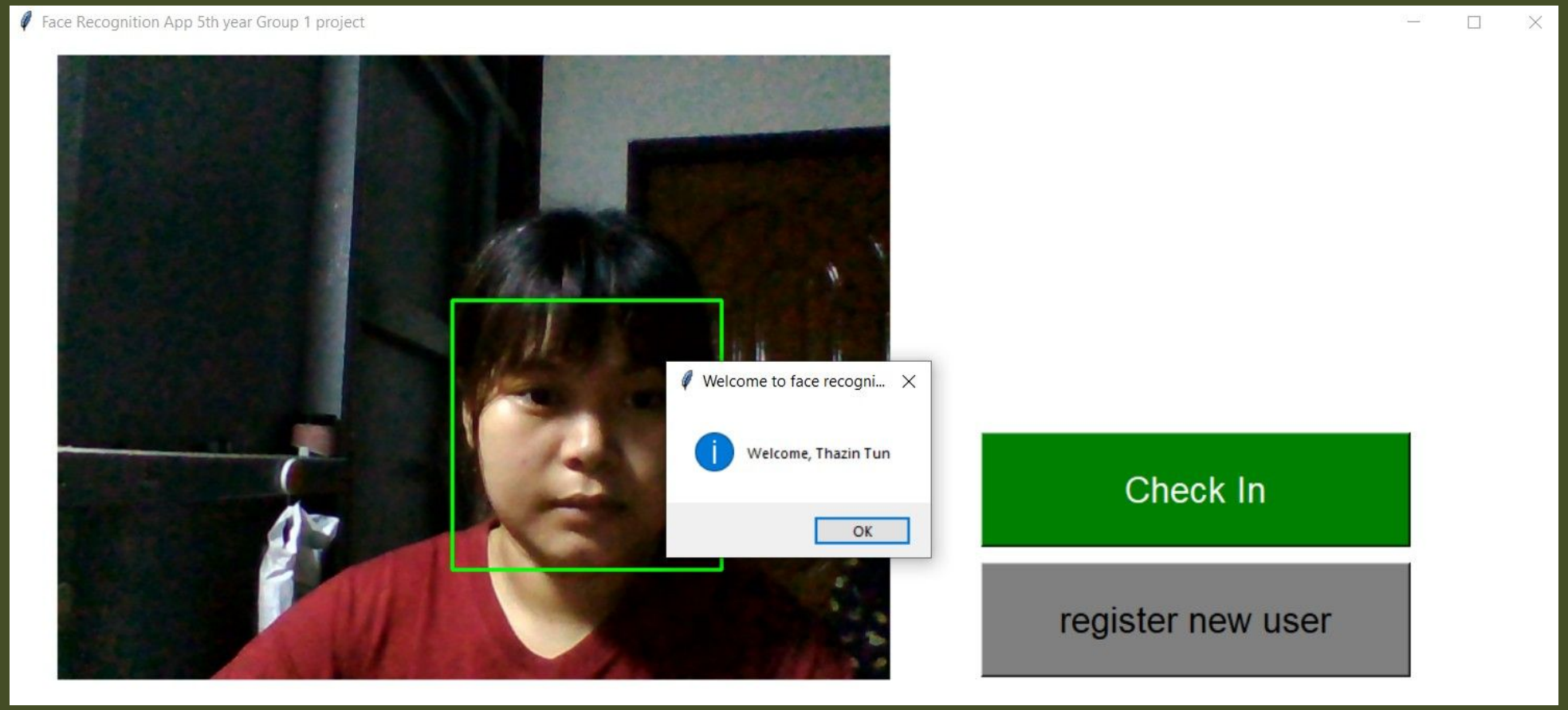
Output Screenshots of Face Recognition System

(4) After you click Accept button, "user was successfully captured" alert box is appear in the first stage.



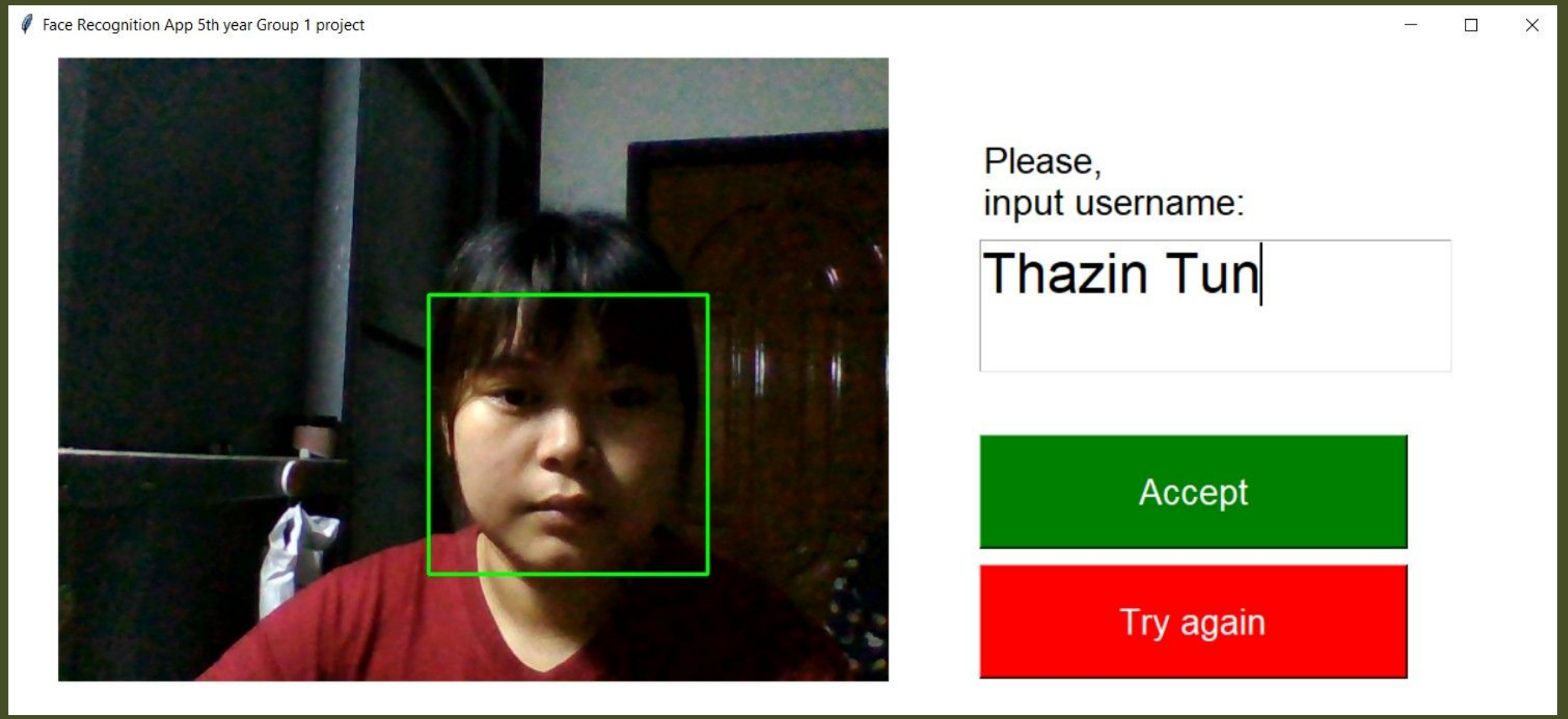
Output Screenshots of Face Recognition System

(5) If you click check in button, system is recognize your face and welcome with your name successfully.

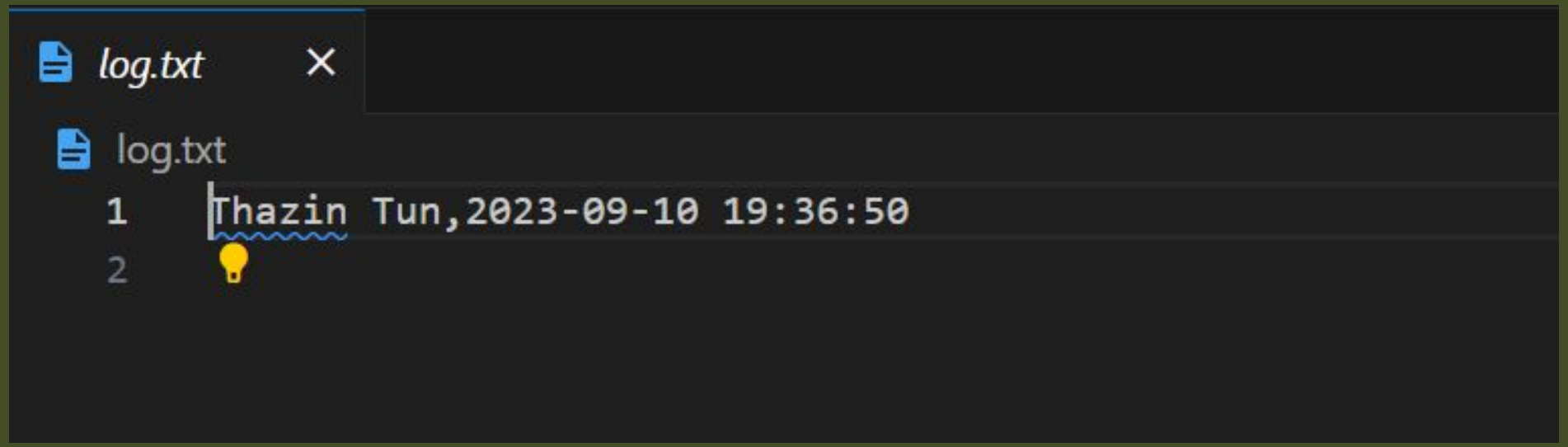


Output Screenshots of Face Recognition System

(6) If you click try again button in stage (3) , you arrive the first stage. It's work like back button.



Log file of Face Recognition system



- Log file can note with your user name & your face capture date & time. You can use this file to create another project of face recognition system such as security, attendance calculation or other recording purposes etc.

Benefits of face recognition system

There are various benefits of facial recognition, depending on the industry and application. It can be a convenient, safe, and hassle-free method to identify a person at a distance without any physical contact.

- Helps find missing people
- Protects businesses against theft
- Strengthens security measures
- It is also used for entertainment purposes. Snapchat, Instagram and other social media platforms recognize faces and make fun and creative filters for people to enjoy.

Conclusion

- Face Recognition system is very important in our daily life.
- It processes a really great advantage.
- It was very costly but now new technologies have evolved and the cost of equipment is going down dramatically due to the integration and the increasing processing power.
- Certain applications of face recognition technology are now cost effective, reliable and highly accurate.

The End