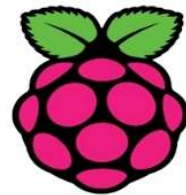




หุ่นยนต์ ตรวจจับใบหน้า อัจฉริยะ

INTELLIGENT FACIAL
RECOGNITION ROBOT



RaspberryPi



โรงเรียนบ้านตาอุด สพป.ศรีสะเกษเขต 3

www.ta-ud.ac.th

หุ่นยนต์ตรวจจับใบหน้าอัจฉริยะ

1. แนวคิด ความเป็นมา

การทำหุ่นยนต์สแกนหน้าหรือหุ่นยนต์ที่ใช้ในการสแกนใบหน้า (Facial Recognition Robot) เป็นเทคโนโลยีที่พัฒนาขึ้นเพื่อใช้ในหลาย ๆ ด้าน ตั้งแต่การรักษาความปลอดภัยจนถึงการเพิ่มประสิทธิภาพในธุรกิจต่าง ๆ แนวคิดในการทำหุ่นยนต์สแกนหน้ามีพื้นฐานมาจากความต้องการในการใช้เทคโนโลยีสแกนใบหน้าที่สามารถประมวลผลและวิเคราะห์ข้อมูลใบหน้าได้โดยอัตโนมัติผ่านอุปกรณ์อิเล็กทรอนิกส์และโปรแกรมประมวลผลที่ทันสมัย

โครงการนี้ได้พัฒนาเครื่องมือการระบุตัวตนจากการรู้จำใบหน้า (Face Recognition) ซึ่งการรู้จำใบหน้าจะทำการเปรียบเทียบใบหน้าบุคคลกับใบหน้าที่อยู่ในฐานข้อมูล โดยการรู้จำใบหน้าจะประกอบไปด้วย 2 ขั้นตอน คือ (1) ขั้นตอนการตรวจจับใบหน้าที่ (Face Detection) คือกระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอ หลังจากนั้นก็ทำการประมวลผลภาพใบหน้าที่ได้สำหรับนำไปใช้ในขั้นตอนถัดไป (2) ขั้นตอนการรู้จำใบหน้า (Face Recognition) คือกระบวนการนำภาพที่ได้จากขั้นตอนการตรวจจับใบหน้าไปเปรียบเทียบกับฐานข้อมูลของใบหน้าเพื่อระบุตัวตนของใบหน้านั้นซึ่งวิธีการระบุใบหน้าที่มีหลากหลายวิธีเช่น Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Elastic Bunch Graph Matching (EBGM) ในงานวิจัยนี้เลือกใช้วิธี Principal Component Analysis (PCA) โดยระบบระบุตัวตนนี้จะทำงานบนบอร์ด Raspberry Pi 4 Model B ในส่วนของโปรแกรมภาษาที่เลือกใช้ในการพัฒนาระบบก็คือ ภาษา Python และ PHP เนื่องจากเป็นภาษาที่เรียนรู้ได้ง่ายและมีไลบรารีให้ใช้งานหลากหลาย เช่น Library ที่เกี่ยวข้องกับการคำนวณทางคณิตศาสตร์ขั้นสูง เช่น อัลกอริธึม, Machine Learning, Artificial Intelligence รวมทั้งยังมี OpenCV (Open source Computer Vision) ที่เกี่ยวกับ Face Detection และ Face Recognition ให้เลือกใช้อีกด้วย และที่สำคัญยังเป็นภาษาที่เป็น Opensource ไม่ต้องเสียเงินซื้อเพื่อนำมาเป็นเครื่องมือในการพัฒนาระบบ

2. วัตถุประสงค์





1. เพื่อศึกษาเทคนิคการตรวจจับใบหน้าที่ใช้ Web Camera และ Raspberry Pi
2. เพื่อศึกษาการเขียนโปรแกรมด้วยภาษา Python กับ Raspberry Pi
3. เพื่อพัฒนาเครื่องมือเช็คชื่อเข้าเรียนด้วยเทคนิคการตรวจจับใบหน้าที่ใช้ Raspberry Pi
4. เพื่อประเมินประสิทธิภาพของระบบตรวจจับใบหน้าที่ใช้การเช็คชื่อเข้าเรียน

3. ขอบเขตของโครงการ

1. โครงการนี้จะใช้เทคนิคการตรวจจับใบหน้า (Face Recognition) เท่านั้น และระยะกล้องในการตรวจจับใบหน้าไม่เกิน 1.5 เมตร
2. โครงการนี้จะตรวจจับใบหน้าในมุมตรงเท่านั้น ความถูกต้องเมื่อใบหน้าที่้ม ใบหน้ายิ้ม การเปลี่ยนแปลงลักษณะท่ายากภาพของใบหน้า และความไวในการตรวจจับ สามารถปรับปรุงอัลกอริทึมของโปรแกรมได้
3. โครงการนี้จะใช้โปรแกรมภาษา Python และ PHP ในการพัฒนาระบบ
4. โครงการนี้จะใช้ข้อมูลของนักเรียนชั้นมัธยมศึกษาโรงเรียนบ้านตาอุดในการพัฒนาระบบ

4. อุปกรณ์ที่ใช้

4.1 Hardware

4.1.1 Raspberry Pi 4 Model B	
4.1.2 Raspberry Pi Camera Module 3	
4.1.3 Raspberry Pi 32GB Class A2 microSD	
4.1.4 Official RPi 14W USB C Power Adaptor	

4.1.4 หน้าจอโทรทัศน์ 32 นิ้ว



4.2 ภาษาโค้ด

Python ใช้ในการ Facial Recognition โลบาร์รี่ที่ใช้

face_recognition – ใช้สำหรับการรู้จำใบหน้า

cv2 (OpenCV) - ใช้สำหรับการจัดการภาพ (ภาพจากกล้องและการประมวลผลภาพ)

numpy – ใช้สำหรับการคำนวณทางคณิตศาสตร์และการจัดการข้อมูลอาเรย์

picamera2 - ใช้สำหรับการใช้งานกล้อง Pi Camera ใน Raspberry Pi

time – ใช้สำหรับการจัดการเวลา

pickle – ใช้สำหรับการจัดการการบันทึกและโหลดข้อมูลจากไฟล์

mysql.connector - ใช้สำหรับการเชื่อมต่อและทำงานกับฐานข้อมูล MySQL

datetime – ใช้สำหรับการจัดการวันที่และเวลา

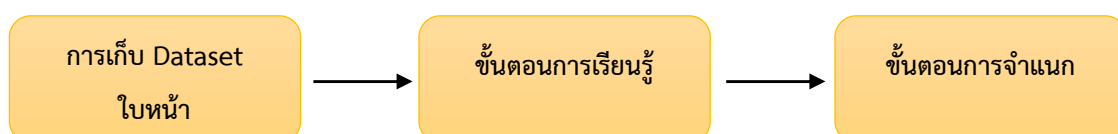
io - ใช้สำหรับการจัดการกับข้อมูลที่เป็น binary stream (เช่น การอ่านภาพจากกล้อง)

os - ใช้สำหรับการทำงานกับระบบปฏิบัติการ เช่น การตรวจสอบหรือสร้างไฟล์และโฟลเดอร์

PHP , MYSQL, ใช้ในการทำ user interface ให้ผู้ใช้ ผ่านเว็บไซต์ <https://face.ta-ud.ac.th>

5. ขั้นตอนการดำเนินงาน

โครงการนี้เป็นการศึกษาและพัฒนากระบวนการระบุตัวตนจากการรู้จำใบหน้า (Face Recognition) ซึ่งจะพัฒนาด้วยภาษา Python และการใช้ไลบรารี OpenCV โดยมีขั้นตอนการทำงานของระบบทั้งหมด ดังแสดงใน Flowchart ดังรูปที่ 1

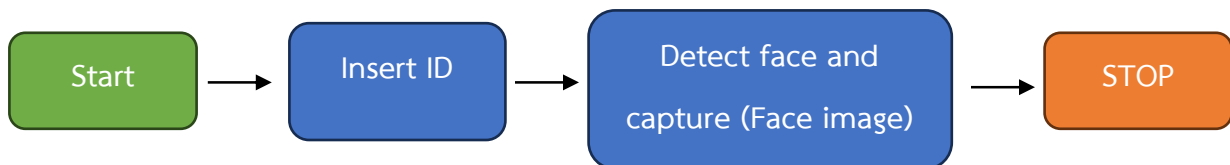


รูปที่ 1 ผังขั้นตอนการทำงานของระบบทั้งหมด

ระบบจะเริ่มด้วยการเก็บ dataset ใบหน้าของนักเรียนชั้นมัธยมศึกษา และนำไปใช้ในขั้นตอนของ การเรียนรู้ใบหน้าเพื่อให้ระบบได้รู้จำใบหน้าเหล่านั้น จากนั้นจึงทำการทดสอบระบบด้วยขั้นตอนการ จำแนกใบหน้า โดยแต่ละขั้นตอนสามารถอธิบายได้ดังนี้

5.1 การเก็บรวบรวมข้อมูลรูปภาพ (Dataset)

เป็นช่วงการเก็บข้อมูลรูปภาพเป็นดาต้าเบส สามารถเรียกเก็บได้จากภาพถ่ายและการถ่ายภาพจาก กล้องโดยตรง ภาพที่เก็บจะเป็นเฉพาะส่วนที่ทำการ detect ได้เท่านั้น ทั้งนี้เพื่อการจำแนกที่แม่นยำและรวดเร็ว การใช้เทค Haar-like แปลงข้อมูลรูปภาพเป็นค่าเวกเตอร์เฉพาะ ต้องทำการแปลงภาพจากภาพสีเป็นภาพขาวดำ ก่อน โดยมีผังขั้นตอนการทำงานของการทำงานการเก็บข้อมูลดังแสดงในรูปที่ 2



รูปที่ 2 ผังขั้นตอนการทำงานของการทำงานการเก็บ dataset

จากผังแสดงการทำงาน ระบบจะให้ใส่ค่า Input ซึ่งคือรหัสนักเรียน (ID) เมื่อใส่รหัสนักเรียนเรียบร้อยแล้ว กล้องจะเริ่มทำงานแล้วเริ่มทำการ detect ใบหน้าและทำการบันทึกภาพเฉพาะส่วนที่ detect ได้ซึ่งเทคนิคที่ใช้ในการตรวจจับใบหน้าในงานวิจัยนี้เลือกใช้เทคนิค Haar-like และการสร้างไฟล์ไลบรารีของค่าใบหน้าบุคคลจาก ขั้นตอน Adaboost และ Cascaded classifier เพื่อใช้เป็นค่าบรรทัดฐานของการจำแนกใบหน้าออกจากภาพพื้นหลังหรือ background ในส่วนของขั้นตอนการพัฒนานั้นใช้ อัลกอริทึมภาษา python โดยช่วง dataset จะรับ ภาพจากเฟรมกล้องโดยตรง ดังภาพ 3





รูปที่ 4 ตัวอย่างภาพที่บันทึกใน dataset

5.2 ขั้นตอนการเรียนรู้(Training)

ขั้นตอนนี้เป็นขั้นตอนการแปลงรูปภาพที่ได้จากขั้นตอนการเก็บ dataset เป็นข้อมูลเวกเตอร์เฉพาะของแต่ละใบหน้า (eigenfaces) โดยมีการทำงานของขั้นตอนนี้คือระบบจะเปิดไฟล์รูปภาพจาก dataset หลังจากนั้นรูปภาพจะถูกแปลงเป็นค่า subspace และระบบจะทำงานแบบนี้ไปจนครบทุก id ใน dataset ซึ่งในขั้นตอนนี้จะใช้เทคนิค PCA โดยนำภาพที่ได้จากการบันทึกทุกภาพใน dataset มาผ่านกระบวนการ normalization เพื่อแปลงเป็นเวกเตอร์ซึ่งจะใช้ชุดเวกเตอร์ภาพเรียนรู้ แต่ละเวกเตอร์มีขนาด 8 bits (LBPH) โดยใช้ไลบรารีการแปลงของ OpenCV และระบบจัดการ encodings.pickle ดังรูปที่ 5

```

✓ import os
  from imutils import paths
  import face_recognition
  import pickle
  import cv2

  print("[INFO] start processing faces...")
  imagePath = list(paths.list_images("dataset"))
  knownEncodings = []
  knownNames = []

  ✓ for (i, imagePath) in enumerate(imagePaths):
      print(f"[INFO] processing image {i + 1}/{len(imagePaths)}")
      name = imagePath.split(os.path.sep)[-2]

      image = cv2.imread(imagePath)
      rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

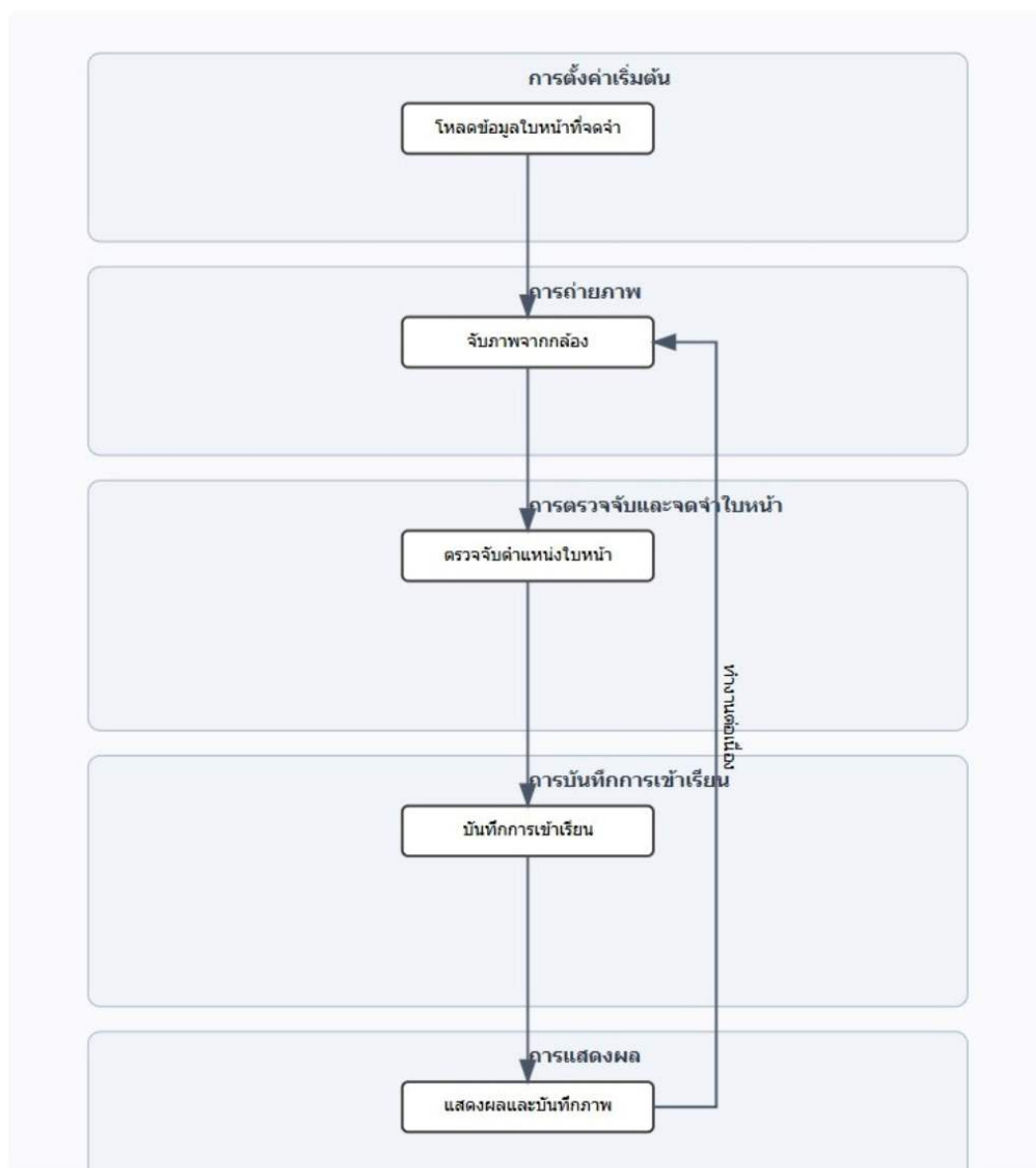
      boxes = face_recognition.face_locations(rgb, model="hog")
      encodings = face_recognition.face_encodings(rgb, boxes)

      ✓ for encoding in encodings:
          knownEncodings.append(encoding)
          knownNames.append(name)
  
```

รูปที่ 5 ตัวอย่างคำสั่งขั้นตอนการเรียนรู้(Training)

4.3 ขั้นตอนการจำแนก (Classification)

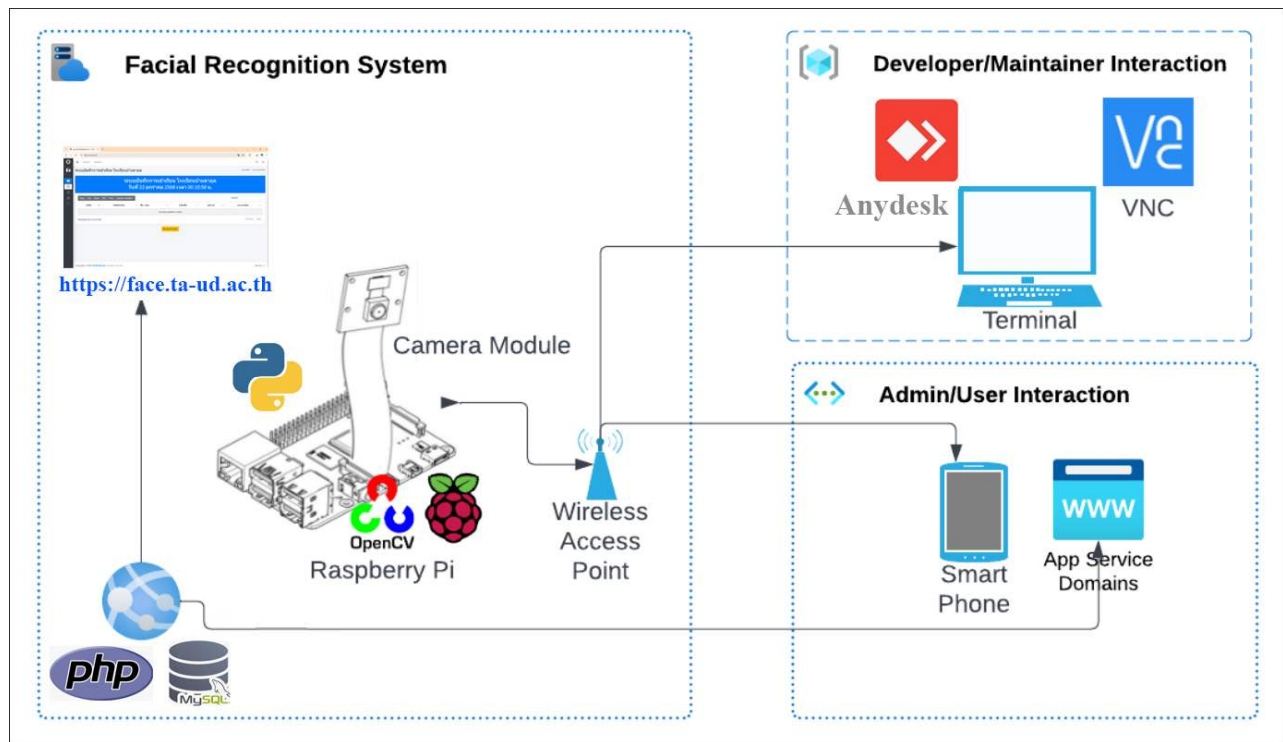
เป็นขั้นตอนนี้เป็นขั้นตอนที่จะนำข้อมูลที่ได้จากกล้องแบบเรลไทม์ ไปเปรียบเทียบกับข้อมูลที่ผ่านมาขั้นตอนการ training แล้ว และทำการเปรียบเทียบจำแนกใบหน้า โดยระบบจะรับภาพจากกล้อง Raspberry Pi Camera Module 3 เข้ามาในเฟรมของระบบเพื่อทำการเตรียมค้นหาใบหน้าเพื่อระบุตัวตน จากนั้นจะทำการเปรียบเทียบใบหน้าภายในเฟรมและใบหน้าที่เก็บไว้เป็นต้นแบบทั้งหมด โดยใช้วิธีทางสถิติจากค่าเวกเตอร์เฉพาะ ระหว่างใบหน้าภายในเฟรมกับใบหน้าที่เก็บไว้ในต้นแบบ จะทำการเปรียบเทียบกับทุกๆ ใบหน้าต้นแบบ จากนั้นจะนำใบหน้าที่มีความคล้ายคลึงมากที่สุดมาระบุตัวตน โดยมีผังการทำงานดัง รูปที่ 6



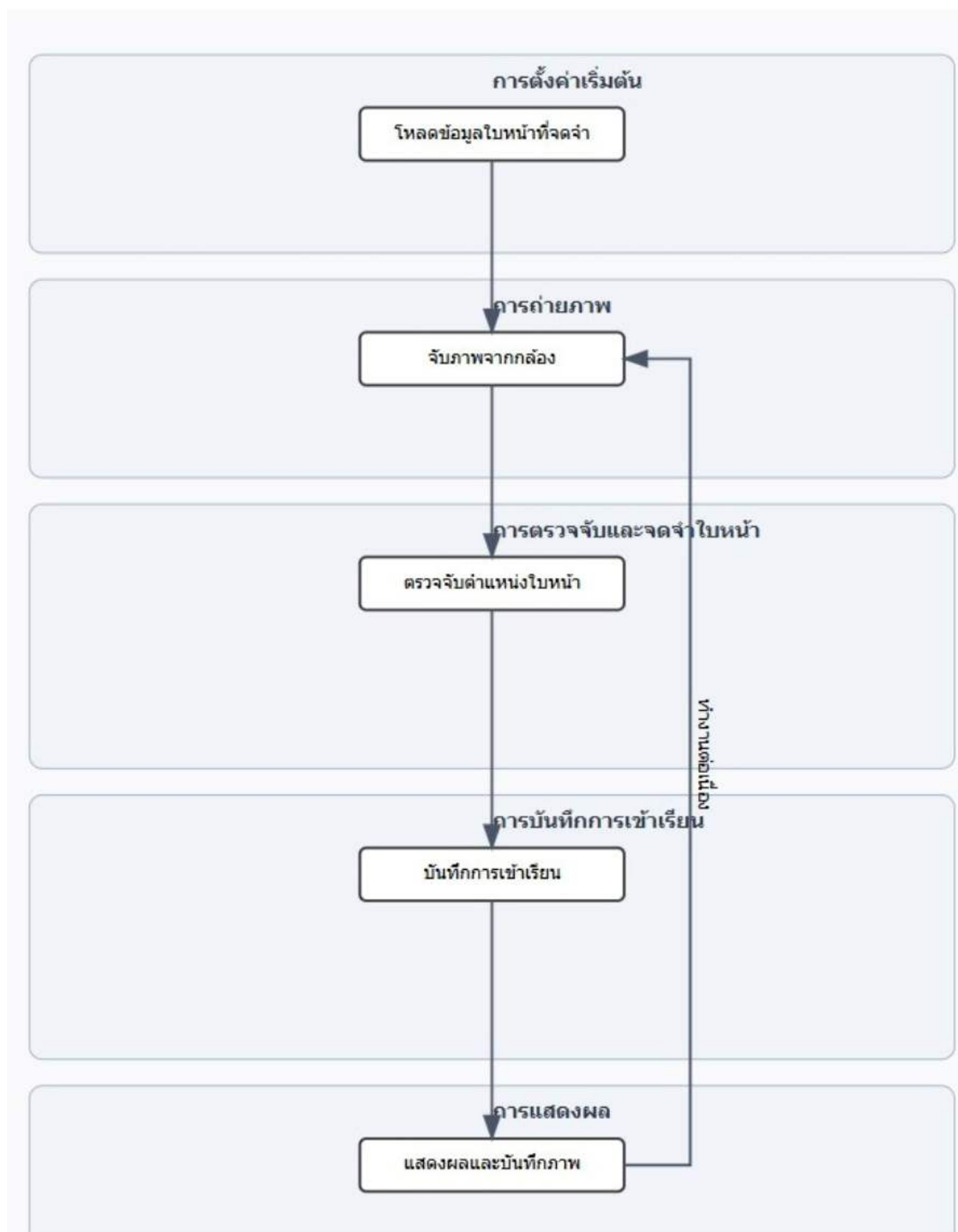
รูปที่ 6 ผังการทำงานของขั้นตอน Classification

เมื่อตรวจพบใบหน้าที่คล้ายคลึงได้แล้ว ระบบจะส่งค่าออกมาเป็นรหัสนักเรียนของใบหน้านั้นๆ และแสดงรหัสนักเรียนบนจอแสดงผล และบันทึกรหัสที่ตรวจจับได้ และเวลาตอนที่ตรวจจับ ภาพจากเฟรมกล้องจะนำมาผ่านขั้นตอนการตรวจจับใบหน้า (face detection) และขั้นตอนการnormalization แล้วแปลงเป็นเวกเตอร์โดยใช้เทคนิคเดียวกันกับขั้นตอนการเก็บ dataset และtraining เพื่อใช้ในการเปรียบเทียบกับเวกเตอร์ที่ได้จากการ training จะถูกส่งข้อมูลไปยัง server และจัดเก็บในฐานข้อมูล Mysql ผ่านเว็บไซต์ <https://face.ta-ud.ac.th>

6. หลักการทำงาน Block Diagram และ Flowchart



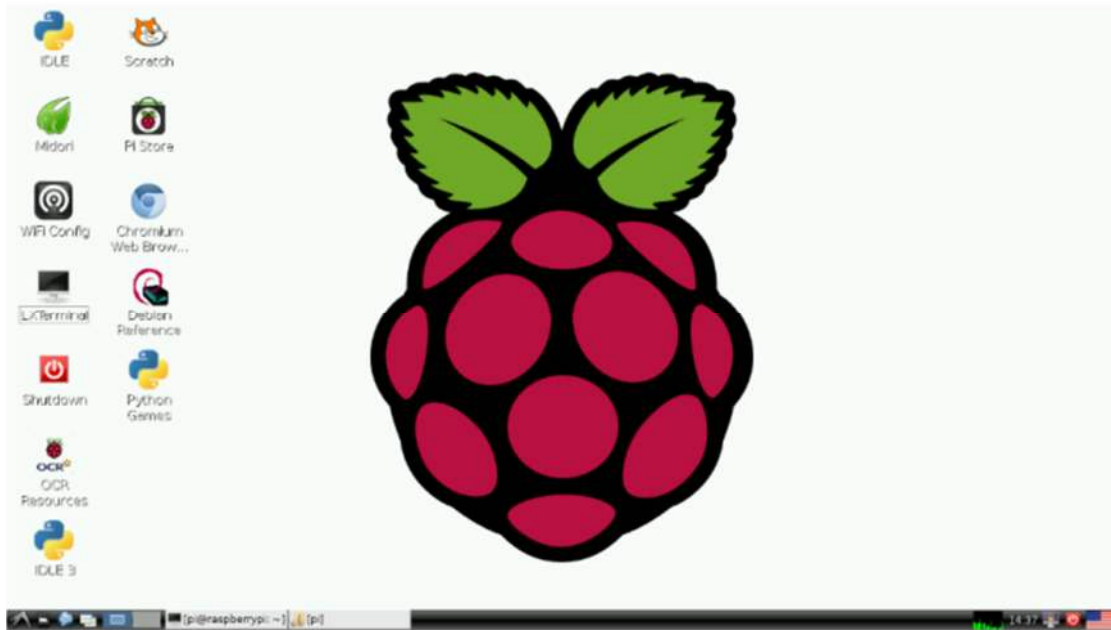
รูปที่ 7 หลักการทำงานของระบบ (Diagram)



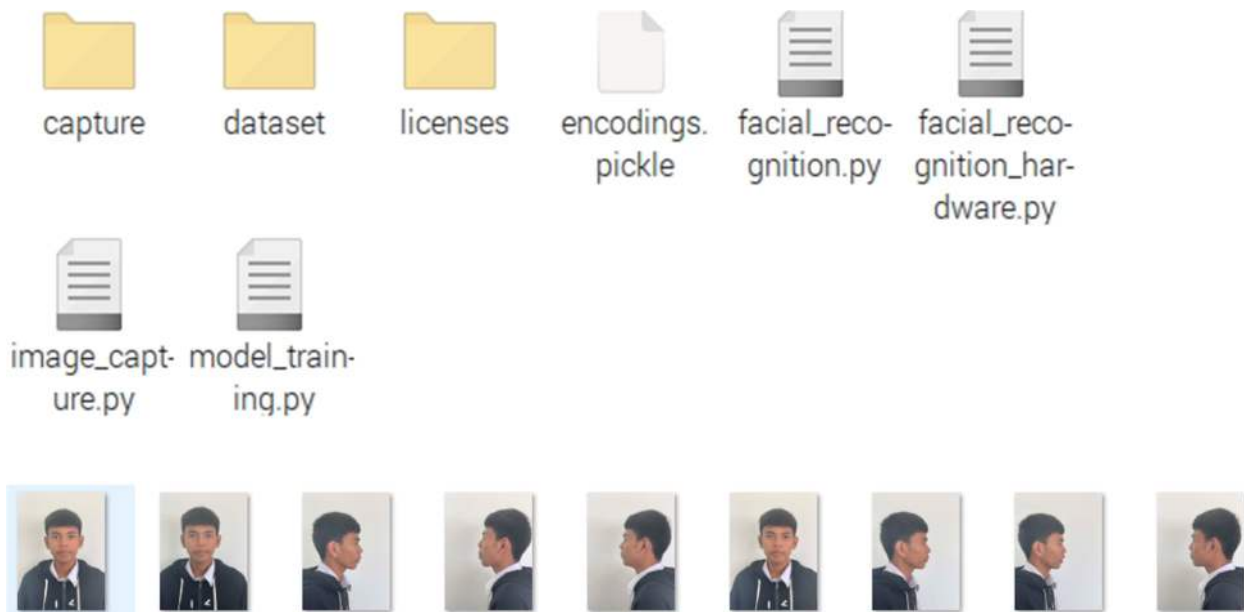
รูปที่ 8 แผนผังการทำงานระบบบันทึกการเข้าเรียนด้วยการจดจำใบหน้า

วิธีการใช้งานระบบ

1. ทำการเข้าสู่ระบบ raspberry pi 4 ผ่าน Terminal ด้วยโปรแกรม Anydesk หรือ VNC ดังภาพ



2. การเก็บรวบรวมข้อมูลรูปภาพ (Dataset) จากการถ่ายภาพจากกล้องโดยตรง โดยใช้รหัสนักเรียนเป็นจัดเก็บ ดังภาพ





3. หลังจากนั้นให้ Training รูปภาพ ด้วยภาษา Python ขั้นตอนนี้เป็นขั้นตอนการแปลงรูปภาพที่ได้จากขั้นตอนการเก็บ dataset เป็นข้อมูลเวกเตอร์เฉพาะของแต่ละใบหน้า (eigenfaces) โดยมีการทำงานของขั้นตอนนี้คือระบบจะเปิดไฟล์รูปภาพจาก dataset และระบบจะทำงานเข้ารหัส จะถูกส่งไปยังไฟล์ encodings.pickle ดังภาพ

```

1 import os
2 from imutils import paths
3 import face_recognition
4 import pickle
5 import cv2
6
7 print("[INFO] start processing faces...")
8 imagePath = list(paths.list_images("dataset"))
9 knownEncodings = []
10 knownNames = []
11
12 for (i, imagePath) in enumerate(imagePaths):
13     print(f"[INFO] processing image {i + 1}/{len(imagePaths)}")
14     name = imagePath.split(os.path.sep)[-2]
15
16     image = cv2.imread(imagePath)
17     rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
18

```

4. เมื่อ Training รูปภาพ ด้วยภาษา Python เสร็จแล้วให้ Run โปรแกรมสแกนใบหน้า ดังภาพ

```

facial_recognition.py ✕
1 import face_recognition
2 import cv2
3 import numpy as np
4 from picamera2 import Picamera2
5 import time
6 import pickle
7 import mysql.connector
8 from datetime import datetime
9 import io
10 import os
11
12
13
14
15 # Load pre-trained face encodings
16 print("[INFO] loading encodings...")
17 with open("encodings.pickle", "rb") as f:
18     data = pickle.loads(f.read())

```

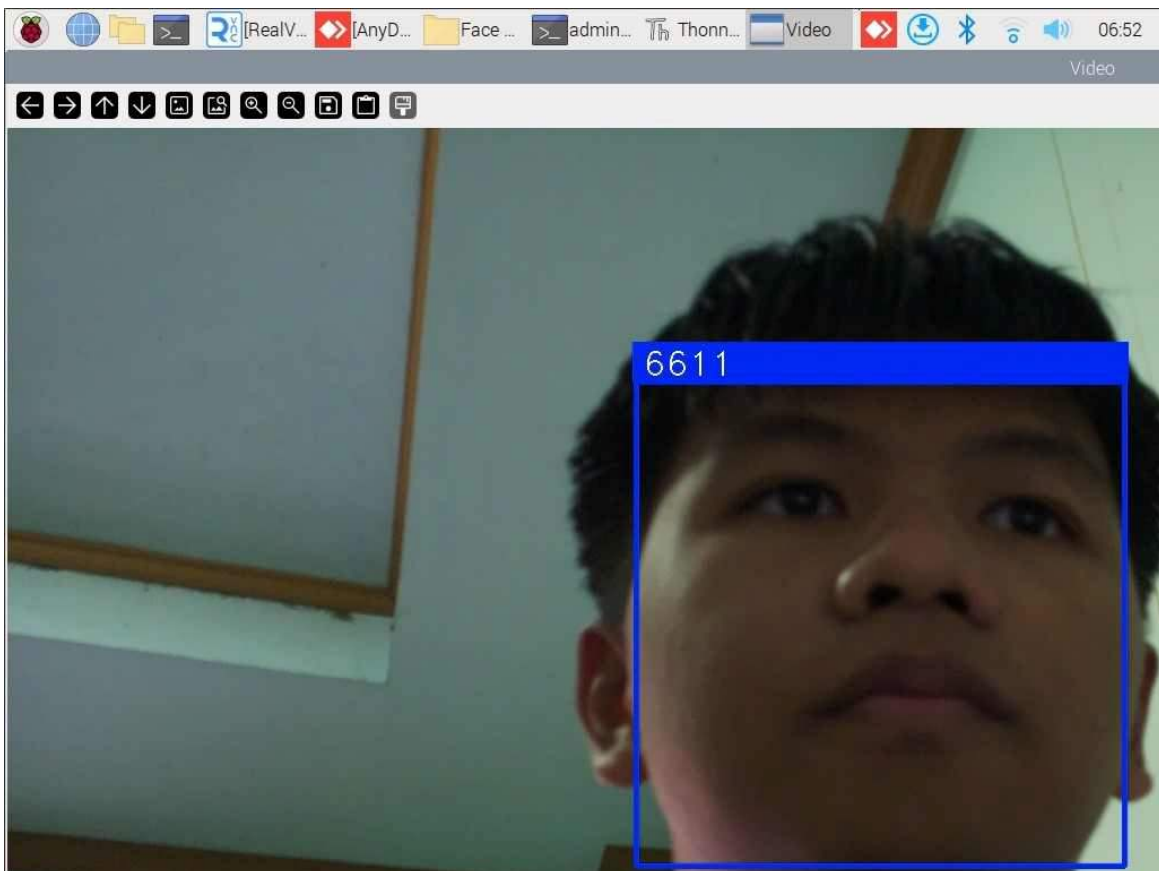


5. ระบบจะทำการจำแนก (Classification) ใบหน้าว่าตรงกับ Dataset ไหม ถ้าตรงระบบจะแจ้งเตือนด้วยกรอบสีน้ำเงิน **แสดงด้วยรหัสประจำตัวนักเรียน** และจะบันทึกข้อมูลการมาเรียนลงใน Database Server (<https://face.ta-ud.ac.th>) ถ้าตรวจสอบใบหน้าแล้วไม่พบระบบจะแสดงกรอบสีแดงและขึ้นข้อความ **unknown** ดังภาพ

```


1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 from tkcalendar import DateEntry
4 import mysql.connector
5 from datetime import datetime
6 import pandas as pd
7 from PIL import Image, ImageTk
8 import io
9
10 class AttendanceSystem:
11     def __init__(self, root):
12         self.root = root
13         self.root.title("ระบบเช็คชื่อนักเรียน")
14         self.root.geometry("1200x700")
15
16         # Database connection
17         self.db_config = {
18             'host': '203.146.252.149',
19             'user': 'bantaud_face',
20             'password': 'cw&0v139Kp91h^8wP1',
21             'database': 'bantaud_face'
22         }
23
24         # Create main frames
25         self.create_filter_frame()
26         self.create_table_frame()
27         self.create_stats_frame()
28
29     def create_filter_frame(self):
30         filter_frame = ttk.LabelFrame(self.root, text="ตัวกรอง", padding="10")
31         filter_frame.pack(fill="x", padx=5, pady=5)
32
33         # Date filter
34         ttk.Label(filter_frame, text="วันที่:").grid(row=0, column=0, padx=5)
35         self.date_filter = DateEntry(filter_frame, width=12, background='darkblue',

```



6. ระบบจะส่งข้อมูลการมาเรียนยัง เว็บไซต์ <https://face.ta-ud.ac.th> ดังภาพ

The screenshot shows a web browser displaying the website face.ta-ud.ac.th. The page title is "ระบบบันทึกการเข้าเรียน โรงเรียนบ้านตาอุด" (Student Attendance System, Ban Ta Oud School). The date and time displayed are "วันที่ 22 มกราคม 2568 เวลา 06:58:32 น." (January 22, 2568, 06:58:32 AM). The page shows a table with one entry for a student named นายสหภาพ สิริเชตร (Mr. Sompap Sirichet) with ID 6611, enrolled in ม.5/1 (Grade 5/1). The student's photo is shown, and the attendance time is 2025-01-22 06:50:03. The page includes a sidebar with navigation icons, a search bar, and a footer with copyright information: "Copyright © 2025 TA-UD Service. All rights reserved. Version 1.0".

ลำดับ	รหัสนักเรียน	ชื่อ - สกุล	ระดับชั้น	รูปภาพ	เวลาเข้าเรียน
1	6611	นายสหภาพ สิริเชตร	ม.5/1		2025-01-22 06:50:03

Showing 1 to 1 of 1 entries

Previous 1 Next

← กลับหน้าหลัก

Copyright © 2025 TA-UD Service. All rights reserved. Version 1.0

บรรณานุกรม

<https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

https://www.w3schools.com/python/python_mysql_getstarted.asp

<https://raspberrytips.com/install-opencv-on-raspberry-pi/>