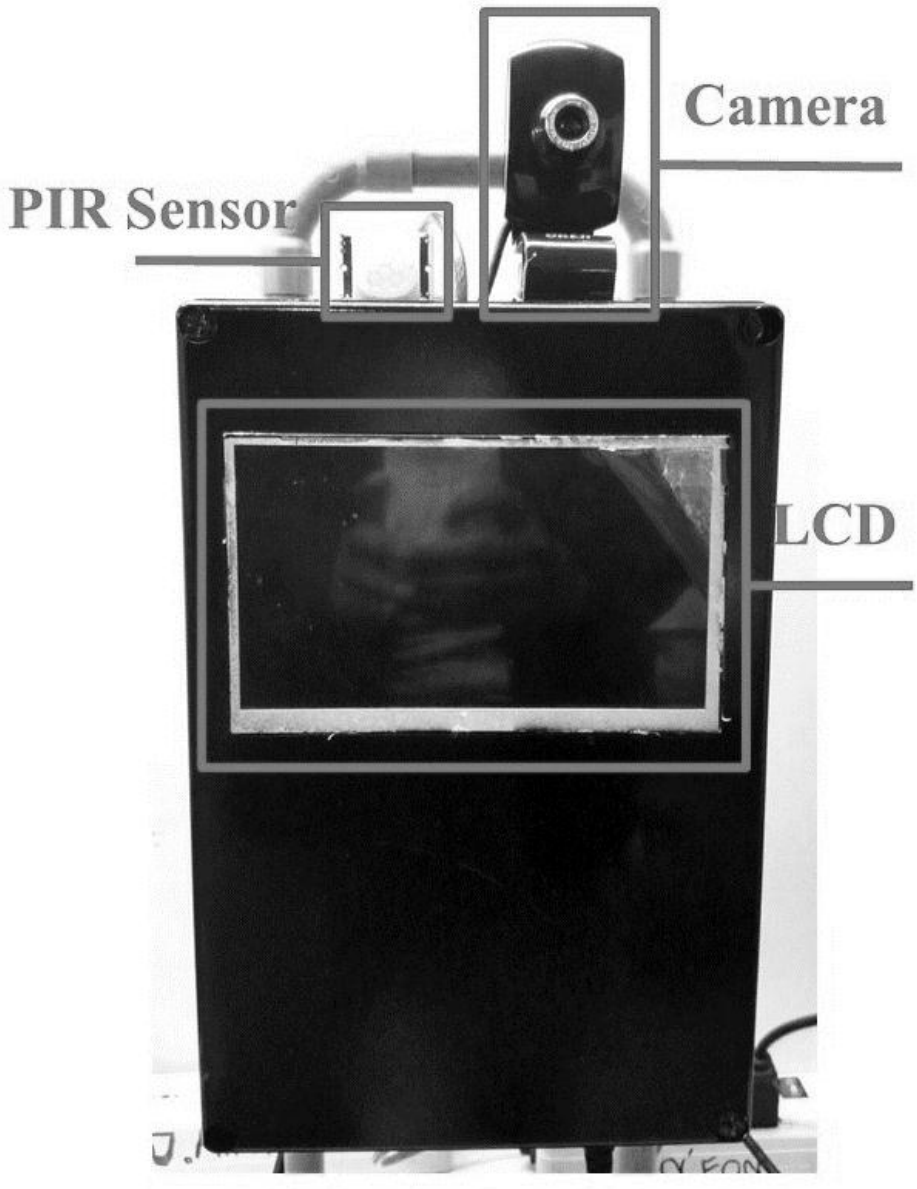
ภาคผนวก

ภาคผนวก ก

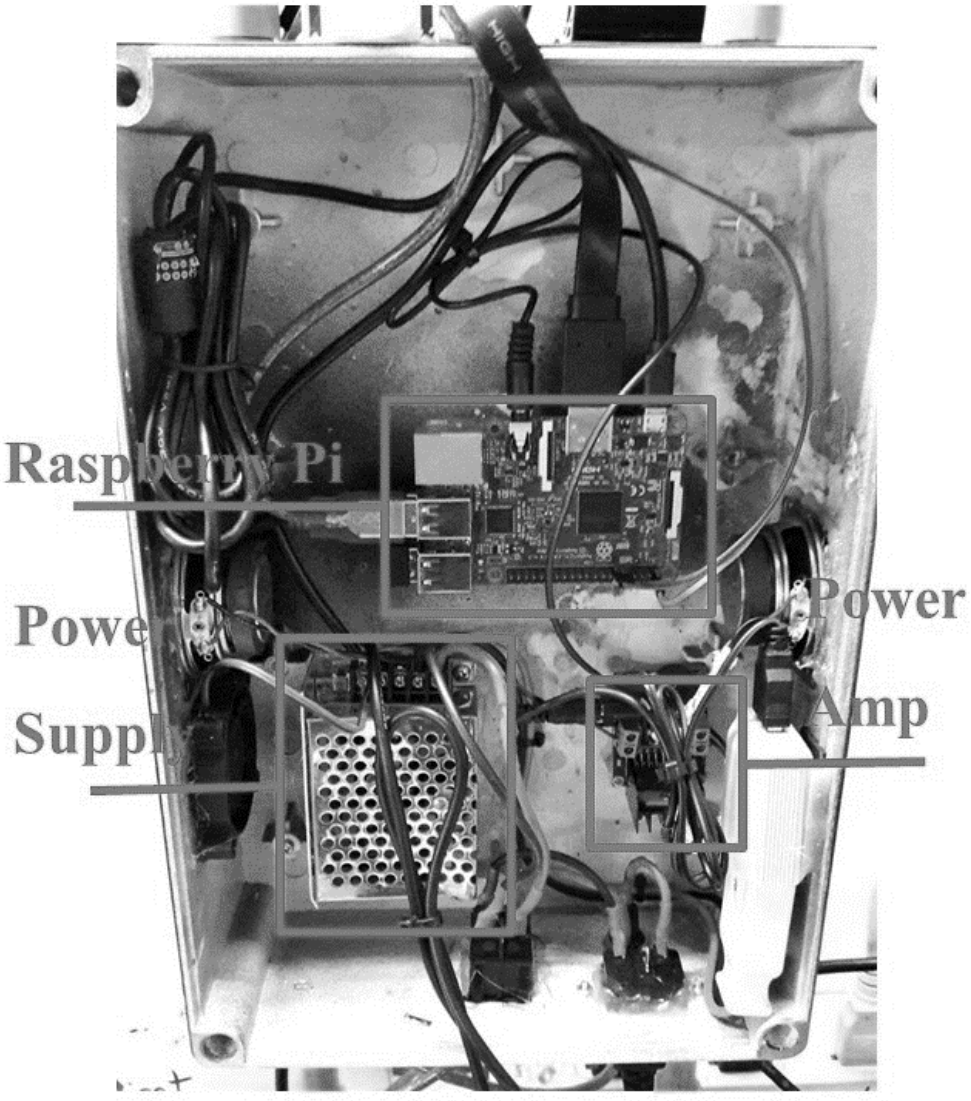
คู่มือการใช้งานระบบสืบค้นประวัติทางการแพทย์โดยเทคนิครู้จำใบหน้าสำหรับหุ่นยนต์ดูแลผู้ป่วย

ระบบสืบค้นประวัติทางการแพทย์โดยวิธีรู้จำใบหน้าสำหรับหุ่นยนต์ดูแลผู้ป่วย เป็นการนำเทคโนโลยีการรู้จำใบหน้า (Face Recognition) มาประยุกต์ใช้ในการทางการแพทย์ โดยนำมาประยุกต์ใช้ในการค้นหาประวัติทางการแพทย์ของผู้ป่วยหรือผู้สูงอายุ จากระบบฐานข้อมูลที่ได้จัดเก็บข้อมูลของผู้ป่วยหรือผู้สูงอายุไว้



ภาพที่ ก.1 องค์ประกอบภายนอกหุ่นยนต์ดูแลผู้สูงอายุ

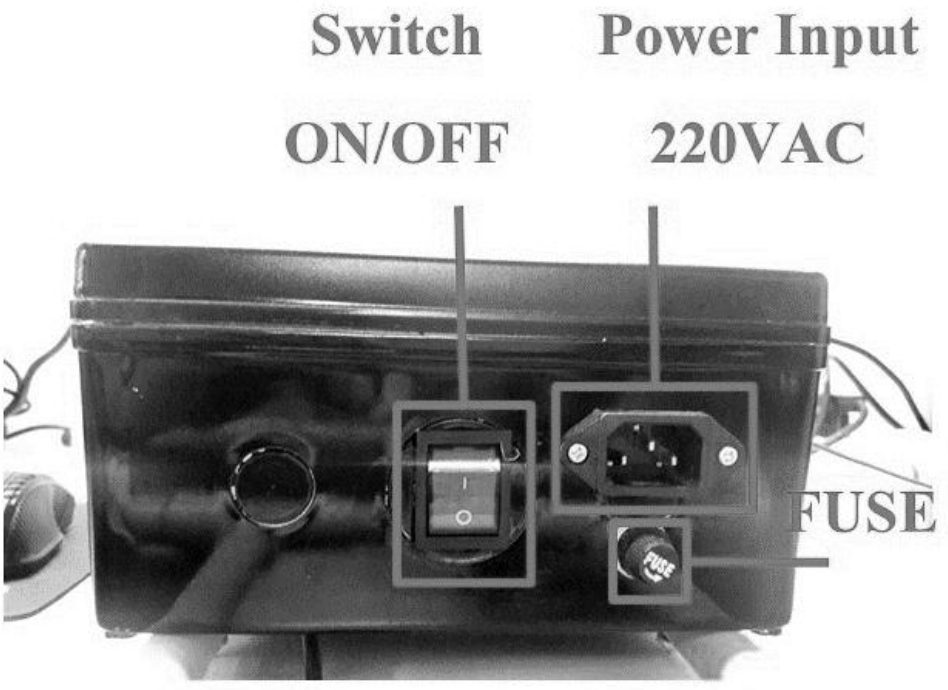
จากภาพที่ ก.1 เป็นองค์ประกอบภาพนอกของหุ่นยนต์ดูแลผู้สูงอายุ โดยจะประกอบด้วย พีไออาเซ็นเซอร์ (PIR Sensor) ใช้ในการตรวจจับการเคลื่อนไหวของสิ่งมีชีวิตโดยอ่านค่าจากคลื่นอินฟาเรด (Infrared) กล้องเว็บแคม (Webcam Camera) ใช้ในการถ่ายภาพของบุคคลที่พีไออา (PIR Sensor) ตรวจจับได้ จอแอดซีดี (LCD) จอแอลซีดีใช้ในการแสดงข้อมูลประวัติทางการแพทย์ของผู้ป่วยหรือผู้สูงอายุ



ภาพที่ ก.2 องค์ประกอบภายในของหุ่นยนต์ดูแลผู้สูงอายุ

จากภาพที่ ก.2 เป็นองค์ประกอบภายในของหุ่นยนต์จะมีองค์ประกอบคือ บอร์ดราดเบอร์รี่พาย (Raspberry Pi) เป็นหน่วยประมวลผลหลักของตัวหุ่นยนต์ใช้ในการอ่านค่าของ พีไออา

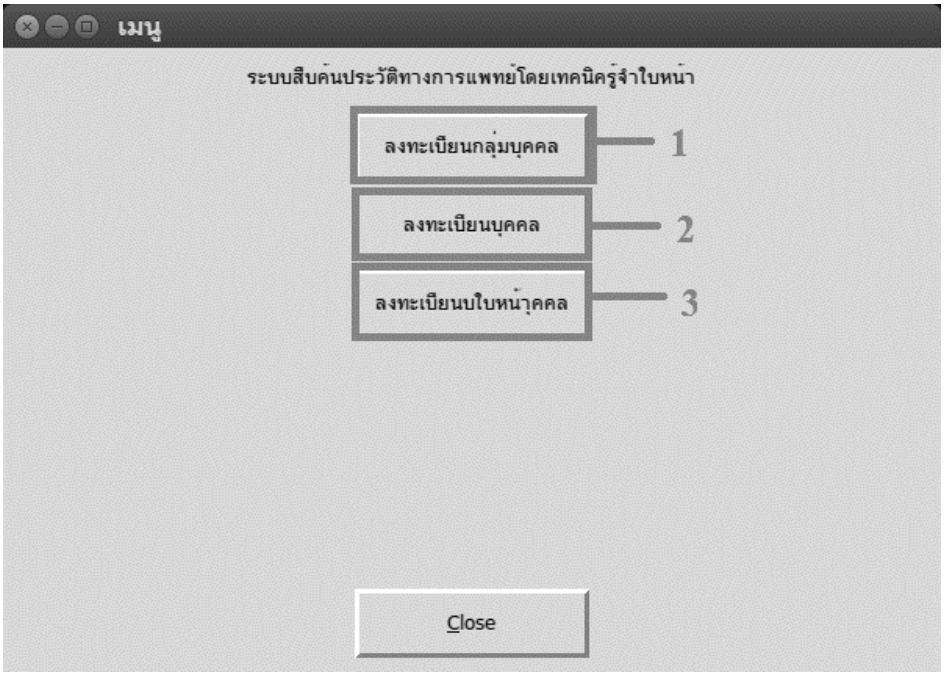
เซ็นเซอร์ (PIR Sensor) และกล้องเว็บแคม (Webcam Camera) เครื่องขยายเสียง (Power Amp) ใช้ขยายเสียงให้กับบอร์ดราดเบอร์รี่พาย (Raspberry Pi) แหล่งจ่ายไฟพาวเวอร์ซับพาย (Power Supply) ใช้จ่ายไฟให้กับอุปกรณ์ที่อยู่ภาพในของตัวหุ่นยนต์



ภาพที่ ก.3 ส่วนของภาคจ่ายไฟกับหุ่นยนต์ดูแลผู้สูงอายุ

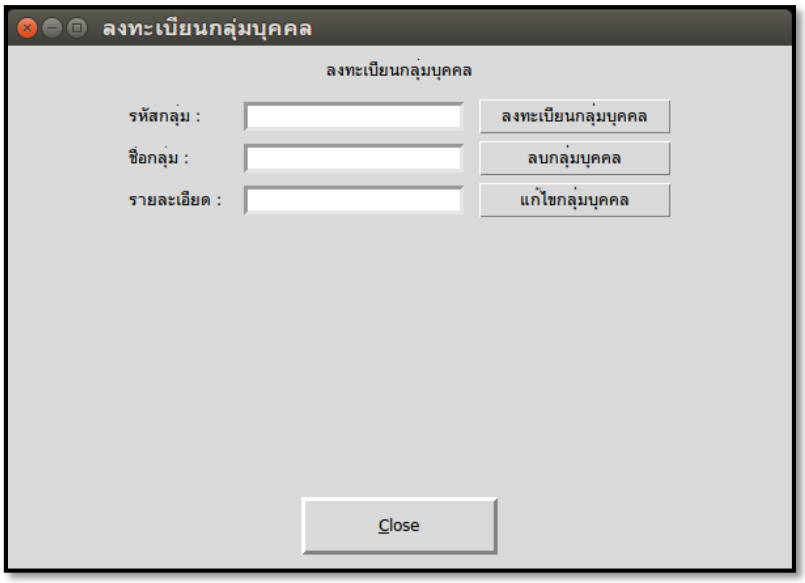
จากภาพที่ ก.3 เป็นส่วนภาคจ่ายไฟให้กับหุ่นยนต์ดูแลผู้ป่วย โดยประกอบด้วย แหล่งจ่ายไฟเข้า (Power Input 220AVC) ใช้จ่ายไฟไปที่ พาวเวอร์ซับพาย (Power Supply) สวิตซ์เปิด/ปิด (Switch ON/OFF) ใช้ในการเปิด/ปิด การทำงานของหุ่นยนต์ดูแลผู้สูงอายุ ฟิวส์ (FUSE) ใช้ตัดไฟเมื่อมีกระแสไฟสูงเกินป้องกันไม่ได้อุปกรณ์พังเสียหาย

การบันทึกข้อมูลของระบบสืบค้นประวัติทางการแพทย์โดยวิธีรู้จำใบหน้าสำหรับหุ่นยนต์ดูแลผู้ป่วย โดยทางเจ้าหน้าที่ทางโรงพยาบาล เป็นคนป้อนข้อมูลเข้าระบบฐานข้อมูล ผ่านทางโปรแกรมที่ได้จัดทำไว้



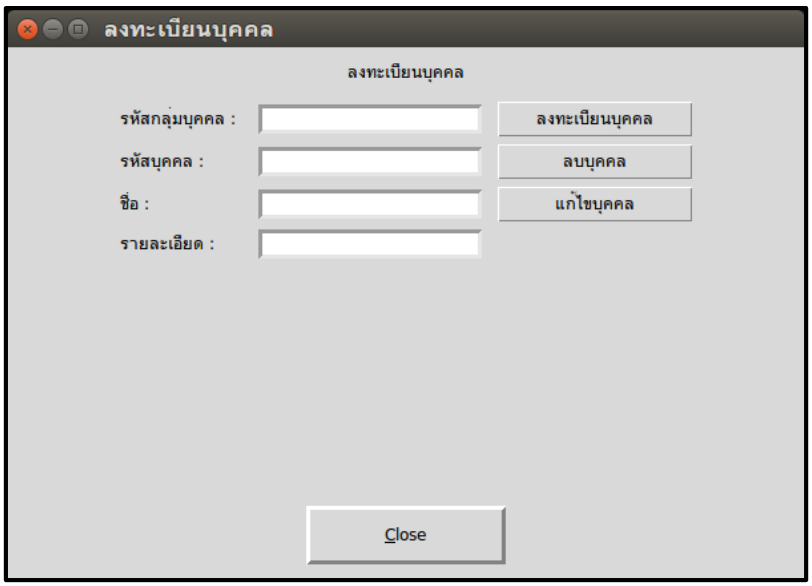
ภาพที่ ก.4 หน้าเมนูหลัก

จากภาพที่ ก.4 เป็นหน้าเมนูหลักของโปรแกรมที่ใช้บันทึกข้อมูลผู้ป่วยหรือผู้สูงอายุ ตัวโปรแกรมประกอบด้วย (1) การลงทะเบียนกลุ่มบุคคล (2) การลงทะเบียนบุคคล (3) การลงทะเบียนใบหน้าบุคคล



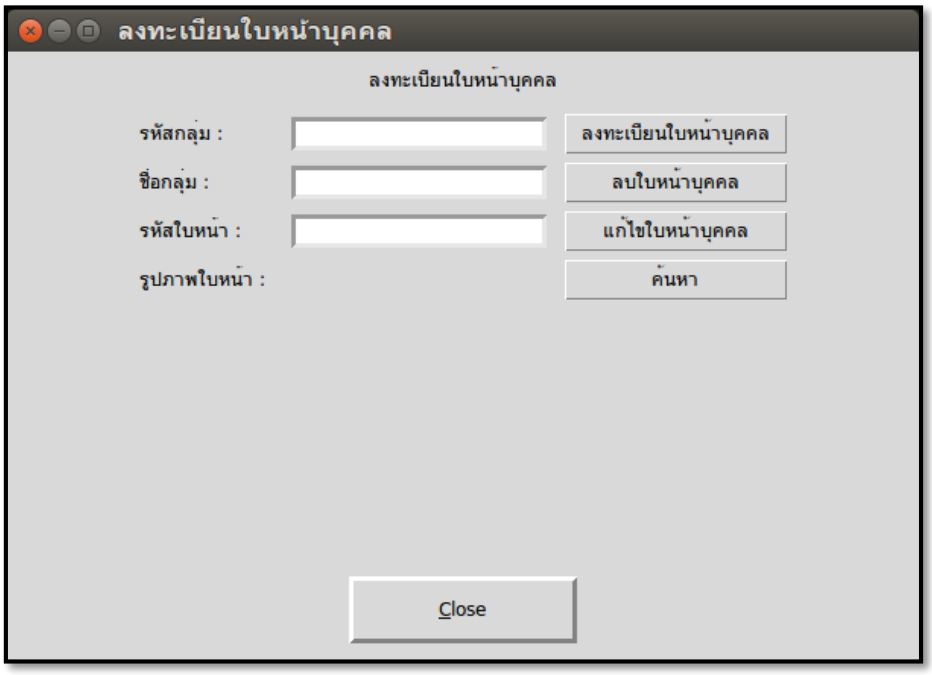
ภาพที่ ก.5 หน้าโปรแกรมลงทะเบียนกลุ่มบุคคล

จากภาพที่ ก.5 เป็นโปรแกรมที่ใช้ในการลงทะเบียนกลุ่มบุคคล เพื่อใช้ในการสร้าง ลบ หรือแก้ไขกลุ่มบุคคล โดยการลงทะเบียนกลุ่มบุคคลใช้ข้อมูล รหัสกลุ่มบุคคล ชื่อกลุ่มบุคคล และรายละเอียดของกลุ่มบุคคล การลบกลุ่มบุคคลใช้แค่รหัสกลุ่มในการลบข้อมูล การแก้ไขข้อมูลของกลุ่มบุคคลใช้รหัสกลุ่มบุคคลในการอ้างอิงกลุ่มที่ต้องการแก้ไขข้อมูลของกลุ่มบุคคล



ภาพที่ ก.6 หน้าโปรแกรมลงทะเบียนบุคคล

จากภาพที่ ก.6 เป็นโปรแกรมที่ใช้ในการลงทะเบียนบุคคล โดยการลงทะเบียนบุคคลใช้ข้อมูลรหัสกลุ่มบุคคล ชื่อบุคคล และรายละเอียดหรือหมายเหตุ ในการลงทะเบียนบุคคล เมื่อลงทะเบียนเสร็จได้รหัสกลุ่มบุคคล (PersonId) การลบข้อมูลบุคคล ใช้ข้อมูลรหัสกลุ่มบุคคล รหัสบุคคล ในการลบข้อมูลบุคคลที่อยู่ในระบบฐานข้อมูล การแก้ไขข้อมูลกลุ่มบุคคล ใช้ข้อมูลรหัสกลุ่มบุคคล รหัสบุคคล เพื่อใช้อ้างอิงบุคคลที่ต้องการแก้ไขข้อมูลบุคคล



ภาพที่ ก.7 หน้าโปรแกรมลงทะเบียนใบหน้าบุคคล

จากภาพที่ ก.7 เป็นโปรแกรมที่ใช้ในการลงทะเบียนใบหน้าบุคคลเพื่อให้ระบบฐานข้อมูลได้รู้จักลักษณะใบหน้าของบุคคล การลงทะเบียนใบหน้าบุคคลใช้ข้อมูลรหัสกลุ่มบุคคล รหัสบุคคล และรูปภาพใบหน้าที่ต้องการลงทะเบียน ระบบจะตอบรหัสใบหน้าบุคคลกลับมา การลบใบหน้าบุคคลให้ข้อมูลกลุ่มบุคคล รหัสบุคคล และรหัสใบหน้าบุคคลในการอ้างอิงข้อมูลในการลบ การแก้ไขใบหน้าบุคคล ใช้ข้อมูลรหัสกลุ่มบุคคล รหัสบุคคล และรูปภาพใบหน้าบุคคล ที่ต้องการแก้ไขลงในระบบฐานข้อมูล

ภาคผนวก ข

โปรแกรมอ่านค่า PIR Sensor โดยใช้ภาษา Python บอร์ด Raspberry Pi

โปรแกรมอ่านค่า PIR Sensor โดยใช้ภาษา Python บอร์ด Raspberry Pi

```
import RPi.GPIO as GPIO
import time
import test
import T_Camera

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.IN)

while True:
    i = GPIO.input(11)
    if i == 0:
        print "No intruders",i
        time.sleep(0.3)

    elif i == 1:
        print "Intruder detected",i
        T_Camera.TCamera()
        test.Testdd()
        time.sleep(0.5)
```

ภาคผนวก ค

โปรแกรมสั่งเปิดกล้อง Web camera และบันทึกรูปภาพ โดยใช้ภาษา Python บอร์ด Raspberry Pi

โปรแกรมสั่งเปิดกล้อง Web camera และบันทึกรูปภาพ โดยใช้ภาษา Python บอร์ด Raspberry Pi

```
#!/usr/bin/python
#-*-coding: utf-8 -*-
#--------------------------------------------------------------------------#
import cv2
import time
#--------------------------------------------------------------------------#
#cascPath = sys.argv[0]
class TCamera():
    def __init__(self):
        cascPath = '/home/pi/Haa/haarcascade_frontalface_alt_tree.xml'
        faceCascade = cv2.CascadeClassifier(cascPath)
        video_capture = cv2.VideoCapture(0)
#--------------------------------------------------------------------------#
        while True:
    # Capture frame-by-frame
            ret, frame = video_capture.read()
            cv2.imwrite('image1.jpg', frame)
            print "Face Detect !!"
            break
#--------------------------------------------------------------------------#
# When everything is done, release the capture
        video_capture.release()
        cv2.destroyAllWindows()

if __name__ == '__main__':
    TCamera()
```

ภาคผนวก ง

โปรแกรมวิเคราะห์แยกแยะและระบุตัวบุคคล โดยใช้ภาษา Python บอร์ด Raspberry Pi

โปรแกรมวิเคราะแยกแยะและระบุตัวบุคคล โดยใช้ภาษา Python บอร์ด Raspberry Pi

```
#!/usr/bin/python
#-*-coding: utf-8 -*-
#--------------------------------------------------------------------------#
import sys
import cv2
import cognitive_face as CF
import json
import MySQLdb
import config
from gtts import gTTS
import os
import Display
#--------------------------------------------------------------------------#
class Testdd():

   def __init__(self):
#--------------------------------------------------------------------------#
# connect Database
      conn = MySQLdb.connect("localhost","root","123456","robotic")
      cursor = conn.cursor()
      print ("connect to Database")
      print ("เชื่อมต่อฐานข้อมูล")
#--------------------------------------------------------------------------#
# Connect Cognitive Service Microsoft
      CF.Key.set(config.KEY)
```

```
        img_url = '/home/robotic/Project/Cognitive-Face/cognitive_face/tests/test_detect.jpg'
        res = CF.face.detect(img_url)
        result = json.dumps(res)
        # print result["faceId"]
        FaceId = result[13:49]
        print (result)
#-------------------------------------------------------------------------#
# Training And Face Recognition
        PGI = "005"
        FI = [FaceId]
        Num = "1"
        Con = "0.3"
        try:
            res = CF.face.identify(FI, PGI, Num, Con)
            jj = json.dumps(res)
            PersonId = jj[81:117]
            Confidence = jj[134:138]
            print jj
            print ("-----------------------------------------------------------")
            print "Sub String Cognitive Face Api"
            print "Face ID : ", jj[13:49]
            print "Person ID : ", PersonId
            print "Confidence : ", Confidence
            print ("-----------------------------------------------------------")
            Display.Display_Main(PersonId)
#-------------------------------------------------------------------------#
        except:
            print("Error: unable to fecth data")
```

```
            tts = gTTS(text = 'สวัสดี ค่ะ', lang = 'th')
            tts.save("Hello_Error.mp3")
            os.system("mpg321 Hello_Error.mp3")

        conn.close()
#-------------------------------------------------------------------------------#
if __name__ == '__main__':
    Testdd()
```

ภาคผนวก จ

โปรแกรมส่วนของการแสดงผลข้อมูลออกผ่านทางหน้าจอ

โปรแกรมส่วนของการแสดงผลข้อมูลออกผ่านทางหน้าจอ

```
#!/usr/bin/python
#-*-coding: utf-8 -*-
#--------------------------------------------------------------------------#
from Tkinter import *
from sys import argv
from gtts import gTTS
import tkMessageBox
import sys
import cv2
import cognitive_face as CF
import config
import json
import MySQLdb
import os
import time


class Person():


    def __init__(self, Window, personid):
        self.conn = MySQLdb.connect("localhost","root","123456","robotic")
        print ("connect to Database")
        self.PersonId = personid
        self.Window = Window
        # self.Window.geometry('600x400')
        Window.title('Create Person')


        self.Data_Show()
```

```
        self.Data_Qurey()

        BClose = Button(self.Window, text = "Close", command = self.Window.quit, relief =
RAISED)
        BClose.pack(side = BOTTOM, anchor = CENTER, pady = 10)
        BClose.config(width = 15, height = 1)

            self.Text2speech()

    def Data_Show(self):
        self.group = Frame(self.Window)
        self.group.pack(side = TOP, anchor = CENTER)

        L1 = Label(self.group, text = "FaceId : ").grid(row = 0, column = 0, sticky = W, pady = 2,
padx = 5)
        L2 = Label(self.group, text = "Name : ").grid(row = 1, column = 0, sticky = W, pady = 2,
padx = 5)
        L3 = Label(self.group, text = "Brithday : ").grid(row = 2, column = 0, sticky = W, pady = 2,
padx = 5)
        L4 = Label(self.group, text = "Address : ").grid(row = 3, column = 0, sticky = W, pady = 2,
padx = 5)
        L5 = Label(self.group, text = "Tel : ").grid(row = 4, column = 0, sticky = W, pady = 2, padx
= 5)

        self.L6 = Label(self.group).grid(row = 0, column = 1, sticky = W, pady = 2, padx = 5)
        self.L7 = Label(self.group).grid(row = 1, column = 1, sticky = W, pady = 2, padx = 5)
        self.L8 = Label(self.group).grid(row = 2, column = 1, sticky = W, pady = 2, padx = 5)
        self.L9 = Label(self.group).grid(row = 3, column = 1, sticky = W, pady = 2, padx = 5)
```

```
    self.L10 = Label(self.group).grid(row = 4, column = 1, sticky = W, pady = 2, padx = 5)

def Data_Qurey(self):
    cursor = self.conn.cursor()
    print("รอรับข้อมูลจากฐานข้อมูล")
    cursor.execute('SET NAMES utf8;')

    sql = "SELECT faceid FROM resume WHERE faceid = '"+self.PersonId+"'"
    cursor.execute(sql)
    results = cursor.fetchall()

    for row in results:
        txt = "".join(row)
        D1 = txt.decode("utf-8")
        self.D11 = D1.encode("utf-8")

    sql = "SELECT name FROM resume WHERE faceid = '"+self.PersonId+"'"
    cursor.execute(sql)
    results = cursor.fetchall()

    for row in results:
        txt = "".join(row)
        D2 = txt.decode("utf-8")
        self.D22 = D2.encode("utf-8")

    sql = "SELECT brithday FROM resume WHERE faceid = '"+self.PersonId+"'"
    cursor.execute(sql)
    results = cursor.fetchall()
```

```
        for row in results:
            txt = "".join(row)
            D3 = txt.decode("utf-8")
            self.D33 = D3.encode("utf-8")

        sql = "SELECT address FROM resume WHERE faceid = '"+self.PersonId+"'"
        cursor.execute(sql)
        results = cursor.fetchall()

        for row in results:
            txt = "".join(row)
            D4 = txt.decode("utf-8")
            self.D44 = D4.encode("utf-8")

        sql = "SELECT tel FROM resume WHERE faceid = '"+self.PersonId+"'"
        cursor.execute(sql)
        results = cursor.fetchall()

        for row in results:
            txt = "".join(row)
            D5 = txt.decode("utf-8")
            self.D55 = D5.encode("utf-8")

        self.L6 = Label(self.group, text = self.D11).grid(row = 0, column = 1, sticky = W, pady = 2,
padx = 5)
        self.L7 = Label(self.group, text = self.D22).grid(row = 1, column = 1, sticky = W, pady = 2,
padx = 5)
```

```
        self.L8 = Label(self.group, text = self.D33).grid(row = 2, column = 1, sticky = W, pady = 2,
padx = 5)
        self.L9 = Label(self.group, text = self.D44).grid(row = 3, column = 1, sticky = W, pady = 2,
padx = 5)
        self.L10 = Label(self.group, text = self.D55).grid(row = 4, column = 1, sticky = W, pady =
2, padx = 5)


    def Text2speech(self):
        tts = gTTS(text = 'สวัสดี', lang = 'th')
        tts.save("1.mp3")
        tts = gTTS(text = 'คุณ', lang = 'th')
        tts.save("2.mp3")
        tts = gTTS(text = self.D22, lang = 'th')
        tts.save("3.mp3")


        os.system("mpg321 1.mp3")
        os.system("mpg321 2.mp3")
        os.system("mpg321 3.mp3")


def Display_Main(personid):
    root = Tk()
    MyGUI = Person(root,personid)
    root.mainloop()


if __name__ == '__main__':
            Display_Main()
```