



ภาคผนวก ก
เครื่องมือวิจัย

ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏรำไพพรรณี

เครื่องมือวิจัย

1. หน้าจอพัฒนาโปรแกรมการอ่านภาพขึ้นมาตรวจสอบตามกำหนด

```

1 class SimpleFacerec:
2     def __init__(self):
3         self.known_face_encodings = []
4         self.known_face_names = []
5
6         # Resize frame for a faster speed
7         self.frame_resizing = 0.25
8
9     def load_encoding_images(self, images_path):
10        """
11        Load encoding images from path
12        :param images_path:
13        :return:
14        """
15        # Load Images
16        images_path = glob.glob(os.path.join(images_path, "*.jpg"))
17
18        print("{} encoding images found.".format(len(images_path)))
19
20        # Store image encoding and names
21        for img_path in images_path:
22            img = cv2.imread(img_path)
23            rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
24
25            # Get the filename only from the initial file path.
26            basename = os.path.basename(img_path)
27            (filename, ext) = os.path.splitext(basename)
28            # Get encoding
29            img_encoding = face_recognition.face_encodings(rgb_img)[0]
30
31            # Store file name and file encoding
32            self.known_face_encodings.append(img_encoding)
33            self.known_face_names.append(filename)
34        print("Encoding images loaded")
35
36    def detect_known_faces(self, frame):
37        small_frame = cv2.resize(frame, (0, 0), fx=self.frame_resizing, fy=self.frame_resizing)
38        # Find all the faces and face encodings in the current frame of video
39        # Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition uses)
40        rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
41        face_locations = face_recognition.face_locations(rgb_small_frame)
42        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
43
44        face_names = []
45        for face_encoding in face_encodings:
46            # See if the face is a match for the known face(s)
47            matches = face_recognition.compare_faces(self.known_face_encodings, face_encoding)
48            name = "Unknown"
49
50            # If a match was found in known_face_encodings, just use the first one.

```

ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏรำไพพรรณี

2. หน้าจอแสดงการส่งค่าแจ้งเตือนภาพระบบไลน์

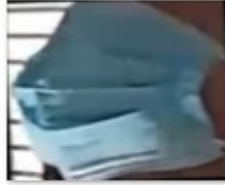
```

1 import cv2
2 from simple_facerec
3 import SimpleFacerec
4 import time
5 import requests
6
7 #encode faces from folder
8 sfr = SimpleFacerec()
9 sfr.load_encoding_images("images")
10
11 url = 'https://notify-api.line.me/api/notify'
12 token = '0uNjVFMV2mat0sVtILd5WBQAYeh1I19dtF62LRnPRYzC'
13
14 #initialize
15 cap = cv2.VideoCapture(0)
16
17 LINE_HEADERS = {"Authorization": "Bearer "+token}
18 while True:
19     ret, frame = cap.read()
20
21     # detect Faces
22     face_location, face_names = sfr.detect_known_faces(frame)
23     for face_loc, name in zip(face_location, face_names):
24         y1, x2, y2, x1 = face_loc[0], face_loc[1], face_loc[2], face_loc[3]
25         if name != None and name != "Unknown":
26             cv2.putText(frame, name, (x1, y1 - 10), cv2.FONT_HERSHEY_DUPLEX, 1, (0, 0, 500), 2)
27             cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 200), 4)
28             return_value, image = cap.read()
29             cv2.imwrite("face_{}.jpg".format(name), image)
30             ##print(name)
31             file = {'imageFile': open('Preechaya.jpg', 'rb')}
32             data = {
33                 'message': "Detected "+name
34             }
35             msg = ("Detect "+name)
36             session = requests.Session()
37             #r=session.post(url, headers=LINE_HEADERS, files=file, data=data)
38             ##print(r.text)
39             time.sleep(0.5)
40             cv2.imshow("Face Mask Detection", frame)
41
42             key = cv2.waitKey(1)
43             if key ==27:
44                 break
45
46 cap.release()
47 cv2.destroyAllWindows()
48

```

ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏรำไพพรรณี

3. หน้าจอการเรนภาพหน้ากากอนามัย



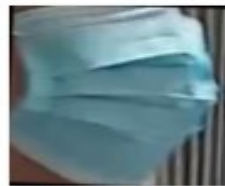
p2_PNG.rf.96407c
7bb86de5a3defe4
6e7aa9b48bb



p2_PNG.rf.f04fa6
39e29fc729b7a1b
bbc11f6a5cc



p5_PNG.rf.09707
d4593da3ac640ac
32df21fd8e2f



p5_PNG.rf.bb01b
7d95c3cc2163b35
ba5ddf6a2fa9

ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏรำไพพรรณี

แบบสอบถามความพึงพอใจ

.....

คำชี้แจง แบบสอบถามนี้จัดทำขึ้นเพื่อสำรวจความพึงพอใจต่อการระบบปัญญาประดิษฐ์สำหรับการตรวจจับผู้ไม่สวมใส่หน้ากากอนามัยในจังหวัดจันทบุรี ประจำปีงบประมาณ พ.ศ. 2565 (กรุณาตอบแบบสอบถามทุกข้อเพื่อให้ได้ข้อมูลที่สมบูรณ์)

ส่วนที่ 1 ข้อมูลทั่วไปของผู้ตอบแบบสอบถาม โปรดทำเครื่องหมาย ลงในช่อง

1.1 สถานที่.....

1.2 อาชีพ ข้าราชการ ลูกจ้างประจำ พนักงานราชการ ค้าขาย

1.3 เพศ ชาย หญิง

ส่วนที่ 2 ความพึงพอใจต่อการรับบริการของท่านโดยแบ่งความหมายของระดับความพึงพอใจ ดังนี้
5= มากที่สุด 4= มาก 3= ปานกลาง 2= น้อย 1= น้อยที่สุด 0= ไม่แสดงความคิดเห็น

| ข้อคำถาม | ระดับความพึงพอใจ | | | | | |
|---------------------------------------------------------------------|------------------|---|---|---|---|---|
| | 5 | 4 | 3 | 2 | 1 | 0 |
| 1. ด้านเจ้าหน้าที่ผู้ปฏิบัติงาน | | | | | | |
| 1.1 มีมนุษยสัมพันธ์ที่ดีสุภาพและเป็นมิตร | | | | | | |
| 1.2 มีความเอาใจใส่ กระตือรือร้น และเต็มใจในการให้บริการ | | | | | | |
| 1.3 ให้บริการด้วยความสะดวก รวดเร็ว | | | | | | |
| 1.4 มีความรู้ ความเข้าใจให้คำแนะนำและตอบข้อซักถาม ตรงกับความต้องการ | | | | | | |
| 2. ด้านกระบวนการ / ขั้นตอน | | | | | | |
| 2.1 มีขั้นตอนและระยะเวลาในการปฏิบัติงาน | | | | | | |
| 2.2 มีความชัดเจนและโปร่งใส | | | | | | |
| 2.3 มีการประชาสัมพันธ์และเผยแพร่ข้อมูลอย่างต่อเนื่อง | | | | | | |

| ข้อคำถาม | ระดับความพึงพอใจ | | | | | |
|-------------------------------------------------------------|------------------|---|---|---|---|---|
| | 5 | 4 | 3 | 2 | 1 | 0 |
| 3. ด้านสิ่งอำนวยความสะดวก | | | | | | |
| 3.1 มีอุปกรณ์ เครื่องมือที่ทันสมัยและรวดเร็วในการปฏิบัติงาน | | | | | | |
| 3.2 สถานที่ทำงาน สะอาด เป็นระเบียบ | | | | | | |
| 3.3 ความชัดเจนของป้ายแสดง | | | | | | |
| 3.4 โดยสภาพรวมทั้งหมดท่านมีความพึงพอใจระบบอยู่ในระดับใด | | | | | | |

ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏรำไพพรรณี