ระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพี

IP Security Camera Monitoring System

ปริญญานิพนธ์

ของ

สาวิตรี คันที

พิชาดา สายเชื้อ

โครงงานปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

พฤศจิกายน 2562

ชื่อโครงงานปริญญานิพนธ์ : ระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพี

ชื่อผู้ทำปริญญานิพนธ์ : นางสาวสาวิตรี คันที

นางสาวพิชาดา สายเชื้อ

อาจารย์ที่ปรึกษาปริญญานิพนธ์ : ผู้ช่วยศาสตราจารย์ ดร.โอฬาริก สุรินต๊ะ

บทคัดย่อ

ปริญญานิพนธ์นี้จัดทำขึ้นเพื่อพัฒนาระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพี โดยนำวิธีการตรวจสอบใบหน้ามาประยุกต์ใช้กับการรักษาความปลอดภัยภายในอาคารบ้านเรือน สำหรับระบบเฝ้าระวังฯ นั้นใช้วิธีการ Haar-Cascade Classifier ในการตรวจจับใบหน้า และใช้วิธีการ ResNet-50 ในการตรวจสอบใบหน้า โดยทีมผู้วิจัยได้ประเมินประสิทธิภาพของระบบตรวจสอบใบหน้าด้วยจำนวน 4 ชุดข้อมูล (The BioID Face, FERET, ColorFERET และภาพจากกล้องไอพี) และผลลัพธ์ของการทดลองแบ่งออกเป็นสองส่วน โดยในส่วนแรกคือการตรวจจับใบหน้า ผลลัพธ์แสดงให้เห็นว่าวิธีการ Haar-Cascade ทำงานได้ดีที่สุดเนื่องจากสามารถตรวจจับใบหน้าจากภาพที่มีความละเอียดต่ำได้ พร้อมทั้งมีความแม่นยำมากถึง 93.79% และส่วนที่สองคือการตรวจสอบใบหน้า ผลลัพธ์แสดงให้เห็นว่าวิธีการ ResNet-50 ทำงานได้ดีที่สุดเนื่องจากมีความแม่นยำถึง 100% ในชุดข้อมูล The BioID Face และ FERET พร้อมทั้งยังมีความแม่นยำสูงสุดเมื่อทดลองกับชุดข้อมูล ColorFERET และภาพจากกล้องไอพี

คำสำคัญ : การตรวจจับใบหน้า, การตรวจสอบใบหน้า, กล้องไอพี, (HOG+SVM), CNN, Faced, Haar-Cascade Classifier, ResNet-50, FaceNet, VGG16

Thesis Title : IP Security Camera Monitoring System

Author : Sawitri Khanthi and Pichada Saichua

Advisor : Asst. Prof. Dr. Olarik Surinta

Abstract

The purpose of this research was to develop ip security camera monitoring system face verification method applies to use in security within dwellings as for this the system. The applied to Haar-Cascade Classifier in face detection and a deep learning method called ResNet-50 architecture in face verification. At evaluate the performance of the face verification on four face datasets (The BioID Face, FERET, ColorFERET and image from IP Camera), The experimental results are divided into two parts; face detection and face verification. First, the result shows that the Haar-Cascade Classifier performs very well on the face detection part as can able to face detection from low resolution images with high accuracy up to 93.79%. Second, The ResNet-50 architectures perform best and obtain 100% accuracy on The BioID Face and FERET dataset. They also, achieved very high accuracy on ColorFERET dataset, image from IP Camera.

Keyword : Face Detection, Face Verification, IP Camera, Histograms of Oriented Gradients and Support Vector Machine (HOG+SVM), Convolutional Neural Network (CNN), Faced, Haar-Cascade Classifier, ResNet-50, FaceNet, VGG16

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีเนื่องจาก ผศ.ดร. โอฬาริก สุรินต๊ะ ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญานิพนธ์ที่ได้ให้ความรู้ ความช่วยเหลือ คำปรึกษา และคำแนะนำสำหรับแนวทางการแก้ไขข้อบกพร่องต่าง ๆ ระหว่างการทำงานวิจัยจนกระทั่งปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ และขอขอบพระคุณ ผศ.อนิรุทธ์ โชติถนอม ที่ให้เกียรติมาเป็นกรรมการในการสอบปริญญานิพนธ์ พร้อมทั้งให้คำแนะนำเพื่อพัฒนาระบบเพิ่มเติมในอนาคต

ขอขอบพระคุณบิดามารดาที่ให้การดูพร้อมทั้งความช่วยเหลือมาโดยตลอด และขอขอบคุณผู้เกี่ยวข้องในปริญญานิพนธ์ฉบับนี้ทุกท่านที่ไม่ได้เอ่ยนามในที่นี้ ประโยชน์อันใดที่เกิดจากการทำปริญญานิพนธ์ฉบับนี้ย่อมเป็นผลมาจากความกรุณาของทุกท่านดังที่กล่าวข้างต้น ทีมผู้วิจัยขอขอบพระคุณมา ณ โอกาสนี้

สาวิตรี คันที

พิชาดา สายเชื้อ

สารบัญ

เรื่อง **หน้า**

สารบัญรูปภาพ

ภาพประกอบ **หน้า**

สารบัญตาราง

ตาราง **หน้า**

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

เทคโนโลยีในปัจจุบันมีความก้าวหน้าและพัฒนาอย่างรวดเร็ว ทำให้ผู้คนนำเอาเทคโนโลยีมาประยุกต์ร่วมกับชีวิตประจำวัน ตัวอย่างเช่น การประยุกต์เข้ากับการรักษาความปลอดภัย โดยการติดตั้งกล้องวงจรปิด (Closed-Circuit Television : CCTV) การติดตั้งกล้องวงจรปิดนั้นเป็นที่นิยมอย่างมากไม่ว่าจะเป็นในองค์กรหรือแม้กระทั่งการติดตั้งตามบ้านเรือน เพราะกล้องวงจรปิดมีขนาดเล็ก สามารถติดตั้งไว้ได้ในทุกที่ พร้อมทั้งสามารถใช้เพื่อเฝ้าสังเกตการณ์ด้วยตนเองได้จากระยะไกล ภาพจากกล้องวงจรปิดยังสามารถบันทึกลงในเครื่อง DVR (Digital Video Recorder) เพื่อเก็บไว้เป็นหลักฐานเมื่อเกิดเหตุร้ายได้อีกด้วย การติดตั้งกล้องวงจรปิดส่งผลให้การรักษาความปลอดภัยมีประสิทธิภาพมากยิ่งขึ้น

ปัจจุบันกล้องวงจรปิดมีหลายรูปแบบ เช่น กล้องที่บันทึกรูปภาพหรือวิดีโอลงเก็บไว้ในเครื่อง DVR โดยเครื่อง DVR จะถูกนำไปใช้เป็นตัวกลางในการเชื่อมต่ออินเตอร์เน็ต ทำให้สามารถเรียกดูข้อมูลได้จากโทรศัพท์มือถือ และกล้องไอพี (IP Camera) โดยกล้องลักษณะนี้ตัวกล้องจะสามารถเชื่อมต่ออินเตอร์เน็ตได้โดยตรง และตัวกล้องยังสามารถบรรจุการ์ดหน่วยความจำ (Memory Card) จึงทำให้ไม่ต้องใช้เครื่อง DVR ในการช่วยบันทึก ผู้ใช้งานสามารถเรียกดูรูปภาพหรือวิดีโอผ่านแอพพลิเคชั่นบนโทรศัพท์มือถือ อีกทั้งยังสามารถดูวิดีโอได้แบบทันท่วงที (Real-time)

ด้วยเหตุผลที่กล่าวมาข้างต้น ทางคณะผู้วิจัยจึงได้พัฒนาระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพี โดยระบบจะเชื่อมต่อกับกล้องไอพี ทำให้สามารถใช้เป็นเครื่องมือในการเฝ้าระวังความปลอดภัยภายในอาคารบ้านเรือนได้ ทั้งนี้ คณะวิจัยจะนำรูปภาพที่เก็บบันทึกไว้จากระบบเฝ้าระวังฯ ไปทำการประมวลผลเพื่อค้นหาบุคคล โดยอาศัยหลักการของการค้นหาใบหน้า (Face Detection) และนำใบหน้าไปเปรียบเทียบกับภาพใบหน้าที่ต้องการค้นหา (Query Image) เพื่อตรวจสอบ (Face Verification) หาค่าความคล้ายคลึง (Similarity) และแสดงผลลัพธ์ทางเว็บบราวเซอร์

1.2 วัตถุประสงค์

เพื่อพัฒนาระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพี

1.3 ขอบเขตโครงงาน

ระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพีประกอบด้วยขอบเขตดังต่อไปนี้

1.3.1 การตรวจสอบใบหน้า (Face Verification)

1.3.1.1 เปรียบเทียบความแม่นยำของการตรวจจับใบหน้า (Face Detection)

1.3.1.2 เปรียบเทียบความถูกต้องของการตรวจสอบใบหน้า

1.3.2 ส่วนของการรับภาพ และตรวจจับใบหน้า

1.3.2.1 ตรวจจับใบหน้าจาก IP Camera

1.3.2.2 แสดงรูปภาพจาก IP Camera

1.3.2.3 บันทึกรูปภาพ และวิดีโอที่ได้จาก IP Camera

1.3.2.4 ตรวจสอบรูปภาพย้อนหลัง

1.3.2.5 ดูภาพจาก IP Camera แบบ Real-time

1.3.3 ส่วนของ Web Application สำหรับใช้ในการตรวจสอบใบหน้า

1.3.3.1 ตรวจจับใบหน้าจากรูปภาพ

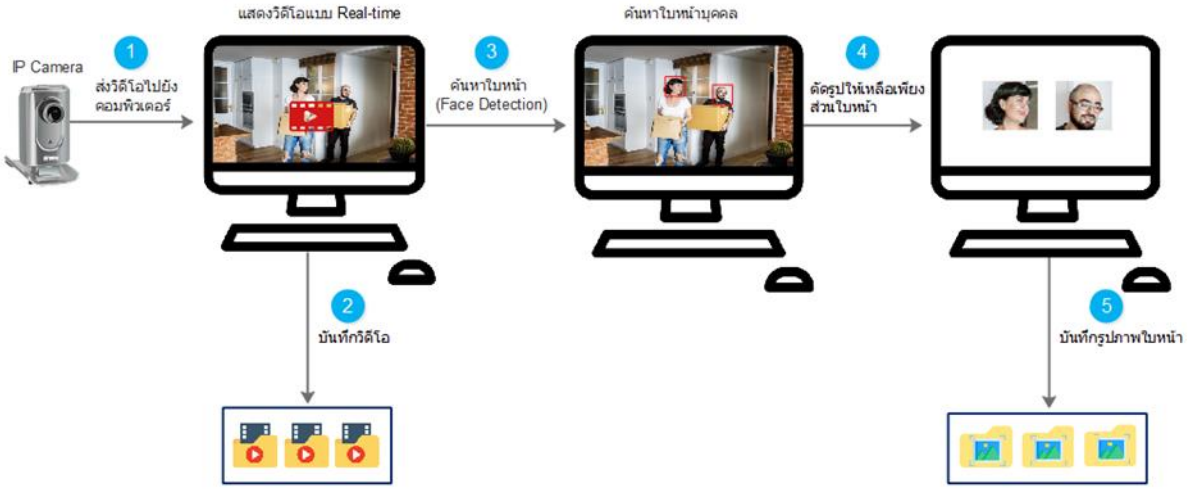
1.3.3.2 ตรวจสอบใบหน้าที่คล้ายคลึงกับภาพที่จัดเก็บอยู่ในไดเรกทอรี่ (query image)

1.3.3.3 แสดงรูปภาพของใบหน้าที่มีความคล้ายคลึงกัน

1.3.3.4 ระบุตำแหน่งของรูปภาพโดยอ้างอิงจาก IP Camera

รูปแบบการทำงานของระบบ

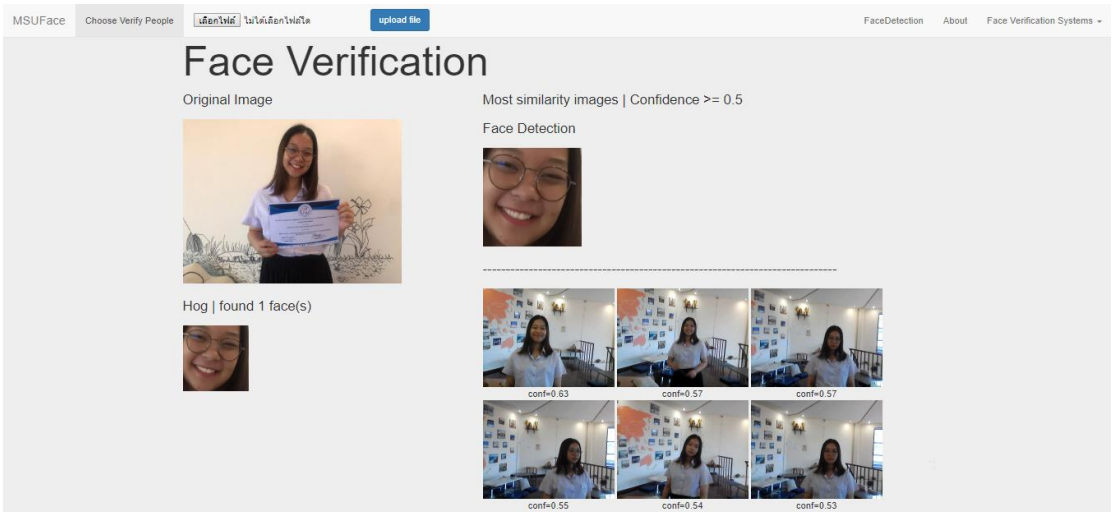
- ส่วนของการรับภาพ และตรวจจับใบหน้า



ภาพประกอบที่ 1-1 รูปแบบการทำงานของระบบในส่วนของการรับภาพ และตรวจจับใบหน้า

จากภาพประกอบที่ 1-1 แสดงให้เห็นถึงรูปแบบการทำงานของระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพีในส่วนของการรับภาพ และตรวจจับใบหน้าโดยในการรับและส่งข้อมูลภาพนั้นจะทำได้โดยการส่งข้อมูลภาพจากกล้องไอพีไปยังคอมพิวเตอร์ และข้อมูลภาพที่รับมานั้นจะมาแสดงผลได้แบบ Real-time โดยทางทีมผู้วิจัยได้เลือกใช้วิธี Histogram of oriented gradients [1] เพื่อใช้ในการตรวจจับใบหน้า จากนั้นระบบจะนำข้อมูลรูปภาพที่ได้จากการตรวจจับใบหน้า และวิดีโอที่บันทึกไว้ได้ส่งไปเก็บไว้ที่โฟลเดอร์ในเครื่องคอมพิวเตอร์เพื่อทำการประมวลผลต่อไป

- ส่วนของ Web Application



ภาพประกอบที่ 1-2 รูปแบบการทำงานของระบบส่วนของ Web Application

จากภาพประกอบที่ 1-2 แสดงให้เห็นถึงรูปแบบการทำงานของระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพีในส่วนในส่วนของ Web Application ซึ่งจะสามารถตรวจสอบและค้นหาความคล้ายคลึงของใบหน้าที่ต้องการจะตรวจสอบได้ โดยในขั้นตอนแรกผู้ใช้งานจะต้องเลือกรูปภาพที่ต้องการจะตรวจสอบและค้นหาจากนั้นระบบจะทำการตรวจจับใบหน้าและนำไปเปรียบเทียบกับรูปภาพที่มีอยู่ในฐานข้อมูล โดยทางทีมผู้วิจัยได้เลือกใช้วิธี ResNet-50 [2] เพื่อใช้ในการตรวจสอบคล้ายคลึงของใบหน้า และจะแสดงผลใบหน้าที่มีความคล้ายคลึงกันออกมาโดยจะเรียงลำดับความคล้ายคลึงจากมากไปน้อย

1.4 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน

1.4.1 IP Camera 1 เครื่อง

1.4.2 คอมพิวเตอร์ 2 เครื่อง

1.5 ภาษาที่ใช้ในการพัฒนา

1.5.1 Face Application

1) ภาษา Python

2) Library ที่ใช้สำหรับการพัฒนา

- OpenCV
- Scikit-learn
- TensorFlow
- Keras

3) วิธีการที่ใช้สำหรับการทดลองและพัฒนา

- Histogram of oriented gradients
- Haar-Cascade Classifier
- Resnet50

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ระบบที่สามารถตรวจสอบและดูภาพวิดีโอในบริเวณที่กำหนดได้แบบ Real-time

1.6.2 สืบค้นและตรวจสอบใบหน้า

1.7 แผนการดำเนินงานตลอดโครงการ

ระยะเวลาการจัดทำโครงงาน เริ่มตั้งแต่เดือนมกราคม พ.ศ. 2562 ถึง ธันวาคม พ.ศ. 2562 แสดงดัง

ตารางที่ 1-1 ตารางแสดงระยะเวลาในการจัดทำโครงงาน

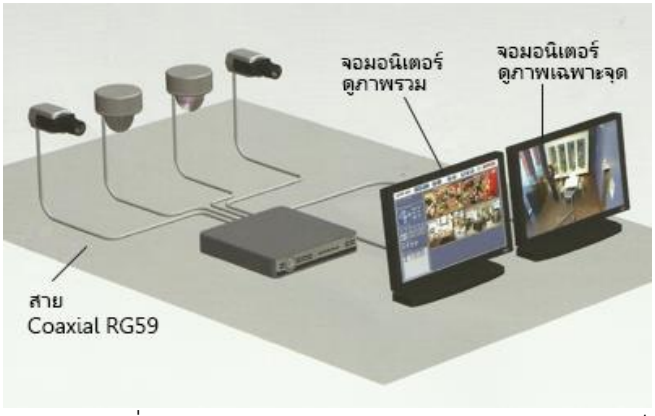
ขั้นตอนการดำเนินงาน	พุทธศักราช 2562											
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. กำหนดและเลือกหัวข้อ	◄——►											
2. ศึกษาข้อมูลที่เกี่ยวข้อง		◄——►										
3. กำหนดขอบเขตของเรื่องที่ศึกษา		◄——►										
4. วิเคราะห์และออกแบบระบบ		◄——►										
5. พัฒนาระบบ			◄———	———	———	———	———	———►				
6. ทดสอบการทำงานและปรับปรุงระบบ			◄———	———	———	———	———	———	———	———►		
7. สรุปผลการดำเนินงาน								◄———	———►			
8. จัดทำเอกสาร		◄———	———	———	———	———	———	———	———	———	———	———►

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ระบบกล้องวงจรปิด

ระบบกล้องวงจรปิด หรือ CCTV (Closed Circuit Television) [3] คือ ระบบการบันทึกภาพเคลื่อนไหวด้วยกล้องเพื่อใช้ในการรักษาความปลอดภัย หรือใช้เพื่อเฝ้าสังเกตการณ์ที่นอกเหนือจากความปลอดภัย การใช้งานระบบกล้องวงจรปิดครั้งแรกนั้นเป็นการใช้งานเพื่อเฝ้าสังเกตการณ์การทำงานของ V2-Rockets เป็นขีปนาวุธในเยอรมัน ซึ่งมีการออกแบบระบบกล้องวงจรปิดที่ซับซ้อนและใช้ค่าใช้จ่ายค่อนข้างสูงจึงไม่เหมาะที่จะมาใช้ในเชิงอุตสาหกรรม แต่ในปัจจุบันระบบกล้องวงจรปิดกลายเป็นสิ่งที่จำเป็นมากสำหรับธนาคาร สถานที่ราชการ ที่สาธารณะ และห้างสรรพสินค้า เพราะระบบกล้องวงจรปิดเป็นระบบรักษาความปลอดภัย และสอดส่องดูแลสถานการณ์ต่าง ๆ ได้อย่างมีประสิทธิภาพ [4] โดยกล้องวงจรปิดมีทั้งหมด 2 ระบบ ได้แก่ ระบบอนาล็อก และระบบไอพี



ภาพประกอบที่ 2-1 การทำงานของกล้องวงจรปิดระบบอนาล็อก

1) ระบบอนาล็อก

จากภาพประกอบที่ 2-1 แสดงให้เห็นถึงการทำงานของกล้องวงจรปิดระบบอนาล็อกโดยแบ่งเป็น 3 เทคโนโลยีหลักคือ AHD HD-CVI และ D-TVI ระบบจะใช้การเชื่อมต่อผ่านสาย Coaxial Cable เช่น RG6 หรือ RG59 เป็นต้น สายสัญญาณเหล่านี้ต่างกันที่ระยะการเดินสาย ยกตัวอย่างเช่น สาย RG6 สามารถใช้ในการเดินสายกล้องวงจรปิดในระยะสูงสุด 700 เมตร แต่ RG59 ใช้ในระยะสูงสุดที่ 200 เมตร

1.1) ข้อดีของกล้องระบบอนาล็อก [5]

1.1.1) ระบบอนาล็อกมีต้นทุนที่ถูกกว่าระบบไอพี

1.1.2) ยืดหยุ่นกว่าระบบไอพี เนื่องจากว่ามีกล้องหลากหลายประเภทให้เลือกใช้ตั้งแต่ระบบเล็กไปถึงระบบใหญ่ ทำให้มีตัวเลือกสำหรับการใช้งานในประเภทต่าง ๆ ได้อย่างเหมาะสม

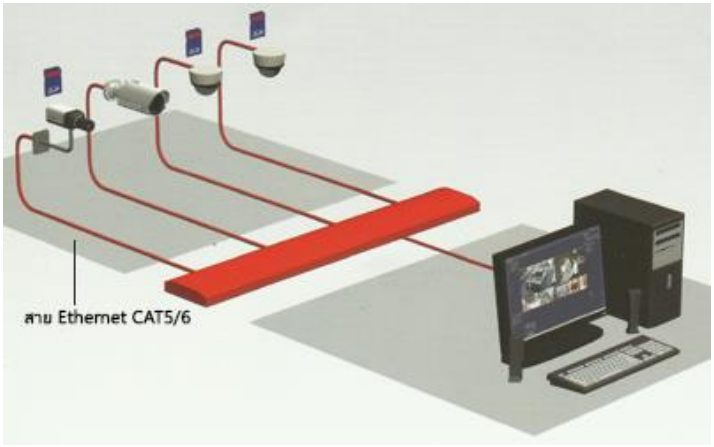
1.1.3) ในระบบอนาล็อกมีเพียงระบบ PAL และ NTSC เท่านั้น ทำให้สามารถเลือกกล้องต่างยี่ห้อมารวมในระบบเดียวกันได้

1.1.4) ปัญหาน้อยกว่าระบบไอพี เนื่องจากว่าระบบอนาล็อกถูกพัฒนามามาก จนแทบจะเรียกได้ว่าอยู่ในช่วงสุดท้ายของเทคโนโลยีของระบบอนาล็อกแล้ว ทำให้ปัญหาต่าง ๆ ถูกแก้ไขไปจนหมด ทำให้ในระบบอนาล็อกนี้เกิดปัญหาขึ้นน้อยมาก

1.2) ข้อเสียของกล้องระบบอนาล็อก

1.2.1) ระบบอนาล็อกมีความปลอดภัยน้อย เนื่องจากไม่มีการเข้ารหัสของข้อมูล ไม่ว่าใครก็สามารถดูภาพจากกล้องวงจรปิดได้

1.2.2) ไม่สามารถรองรับการส่งสัญญาณในระยะไกลได้



ภาพประกอบที่ 2-2 การทำงานของกล้องวงจรปิดระบบไอพี

2) ระบบไอพี

จากภาพประกอบที่ 2-2 แสดงให้เห็นถึงการกล้องวงจรปิดระบบไอพีโดยเป็นกล้องที่ต้องตั้งค่า IP ผ่านระบบเครือข่าย เพื่อกำหนดตัวตนในการแสดงภาพ และต้องอาศัยสายชนิดแลน (LAN) หรือ CAT5 มาเป็นตัวต่อเชื่อมต่อ โดยกล้องระบบไอพีหรือ IP Camera (Internet Protocol Camera) เป็นกล้องวงจรปิดที่รวมเอา คุณสมบัติของ Web Server ไว้ในตัวกล้อง เพื่อให้สามารถดูภาพสดบนระบบ internet หรือ ระบบเครือข่ายได้ โดยผู้ใช้งานสามารถดูภาพจากระยะไกลเพื่อใช้ในการรักษาความปลอดภัย และเฝ้าระวัง ภายในบ้าน สำนักงาน โรงงาน ห้างสรรพสินค้า และในพื้นที่

อื่น ๆ ได้ IP Camera มีทั้งแบบใช้สาย (Wiring) และแบบไร้สาย (Wireless) ตัวอย่างของ IP Camera นั้น ดังภาพประกอบที่ 2-3

ภาพประกอบที่ 2-3 ตัวอย่างของ IP Camera

2.1) ข้อดีของกล้องระบบไอพี

2.1.1) กล้องไอพีสามารถใช้ร่วมกับระบบแลน (LAN) ที่มีอยู่แล้วได้โดยไม่ต้องเดินสายใหม่

2.1.2) หากต้องการเพิ่มกล้องสามารถทำได้ง่ายโดยไม่ติดข้อจำกัดของ Channel ที่จำกัดของเครื่อง DVR

2.1.3) กล้องแต่ละตัวมี IP ของตัวเองทำให้การตั้งค่ากล้องแต่ละตัวทำได้ง่าย

2.1.4) เนื่องทำงานบนระบบ digital สามารถที่จะ backup ข้อมูลได้ตลอดเวลาบน server และ hacker ไม่สามารถดักเอาข้อมูลระหว่างทางได้

2.2) ข้อเสียของกล้องระบบไอพี

2.2.1) ค่าใช้จ่ายจะสูงขึ้นกว่าระบบอนาล็อก ไม่ว่าจะเป็นค่าอุปกรณ์ การดูแลรักษา รวมไปถึงความรู้ของผู้ที่บริหารจัดการข้อมูล

2.1.2 Face Detection

การตรวจจับใบหน้า (Face Detection) [6] คือ ระบบวิเคราะห์ใบหน้าถือว่าเป็นหนึ่งในระบบที่ใช้ในการพิสูจน์ยืนยันตัวตนบุคคลโดยใช้คุณลักษณะจำเพาะทางสรีระ (BIOMETRIC) โดยระบบรู้จำใบหน้าจะทำงานโดยการเปรียบเทียบใบหน้าจากภาพถ่ายดิจิตอลหรือภาพจากกล้องวีดีโอของบุคคลที่เราสนใจกับฐานข้อมูลใบหน้าที่มีอยู่ และเมื่อเปรียบเทียบเสร็จก็จะแสดงผลใบหน้าที่อยู่ในฐานข้อมูลที่มีใบหน้าเหมือนกับภาพที่นำมาเปรียบเทียบออกมา ระบบรู้จำใบหน้านั้นได้ถูกพัฒนามาอย่างต่อเนื่องเป็นเวลามากกว่าสิบปีเนื่องจากเป็นระบบที่ได้รับความสนใจมากจากนักวิชาการหลายสาขาวิชาจึงทำให้ระบบรู้จำใบหน้ามีผู้คนสนใจศึกษาและพัฒนากันเป็นอย่างมาก จนทำให้มีการพัฒนาวิธีในการทำงานของระบบออกมาหลายรูปแบบแตกต่างกันไป ซึ่งการพัฒนาวิธีจะ

แตกต่างกันไปตามยุคสมัยอันเนื่องมาจากปัจจัยด้านองค์ความรู้ และเทคโนโลยีของอุปกรณ์ต่าง ๆ ที่พัฒนาขึ้นเพื่อให้มีความเหมาะสมที่จะนำมาใช้ในระบบจึงทำให้ต้องออกแบบวิธีใหม่ให้เหมาะสมกับอุปกรณ์ใหม่ ๆ ในปัจจุบันระบบรู้จำใบหน้าได้มีการพัฒนาไปอย่างรวดเร็ว ทำให้ระบบรู้จำใบหน้ามีความน่าเชื่อถือมากขึ้น

ภาพประกอบที่ 2-4 ภาพตัวอย่างการตรวจจับใบหน้า (Face Detection)

จากภาพประกอบที่ 2-4 นั้นได้มีการนำระบบรู้จำใบหน้าไปใช้ประโยชน์โดยการติดตั้งในสนามบินเพื่อป้องกันคนร้ายหนีเข้าออกนอกประเทศ และมีระบบรู้จำใบหน้าสำหรับการยืนยันตัวคนร้ายในคดีต่าง ๆ ได้อีกด้วย

2.1.3 ภาษาที่ใช้สำหรับการพัฒนา

1) ภาษา Python

ภาษาไพทอน (Python Programming Language) [7] คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง โดยที่การทำงานของ Python เริ่มแรกนั้นถูกสร้างโดย กีโด ฟาน รอสซัม (Guido van Rossum) และในต่อมาถูกพัฒนาโดย มูลนิธิซอฟต์แวร์ไพทอน ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบปฏิบัติการ Unix, Linux, Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัวนี้เป็น Open Source เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ ทั้งสิ้นเพราะตัวแปรภาษา Python อยู่ภายใต้ลิขสิทธิ์ GNU และภาษา Python ยังเป็นภาษาประเภท Server side Script คือการทำงานของภาษา Python จะทำงานด้านฝั่ง Server แล้วส่งผลลัพธ์กลับมายัง Client ทำให้มีความปลอดภัยสูง อีกทั้งยังเป็นซอฟต์แวร์ที่เปิดเผย ซอร์สโค้ด (Open Source) ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครอบคลุมกับทุกลักษณะงาน

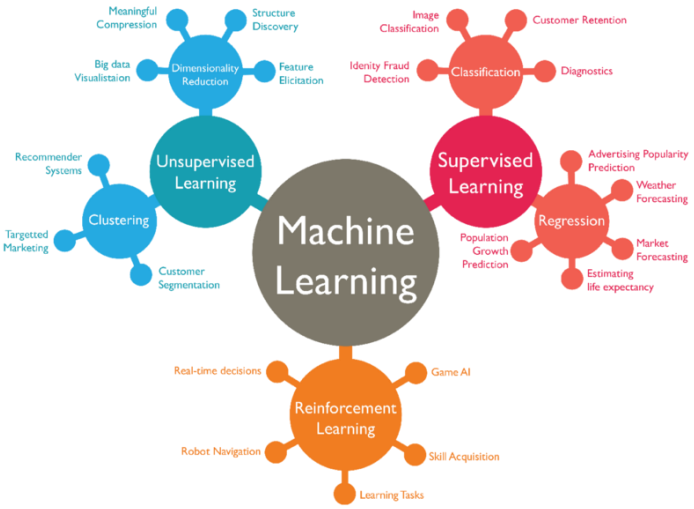
2.1.4 Library ที่ใช้สำหรับการพัฒนา

1) OpenCV

OpenCV (Open source Computer Vision) [8] เป็นไลบรารีฟังก์ชันการเขียนโปรแกรม (Library of Programming Functions) สำหรับจับ, ประมวลผล และ แสดงผลภาพ จากกล้อง Camera ในคอมพิวเตอร์ หรือ โทรศัพท์มือถือ ซึ่งสามารถแสดงผลได้แบบเรียลไทม์ (Real-Time Computer Vision) โดย OpenCV นั้นถูกเขียนขึ้นด้วยหลายภาษา เช่น C/C++, Python, Java และสำหรับ Interface เหล่านี้สามารถพบได้ในเอกสารออนไลน์ ซึ่งมีการรวมภาษาเอาไว้หลากหลาย ตัวอย่างเช่น C#, Perl, Ch, Haskell และ Ruby ได้รับการพัฒนาเพื่อส่งเสริมการนำมาใช้งานโดยผู้ใช้ที่เพิ่มขึ้น และ OpenCV ยังสามารถรันได้ทั้งบน Window, Linux, Android และ Mac

2) Scikit-Learn

Scikit–Learn [9] เป็นไลบรารีฟังก์ชันสำหรับการทำ Machine Learning Library โดย Scikit–Learn นี้ เป็นตัวเสริมต่อจาก Library SciPy ได้รับความนิยมอย่างแพร่หลายในเวลาอันรวดเร็ว เนื่องจากมี Algorithm ต่าง ๆ ของ Machine Learning ให้ใช้งานอย่างครบครัน



ภาพประกอบที่ 2-5 ประเภทของ Machine Learning

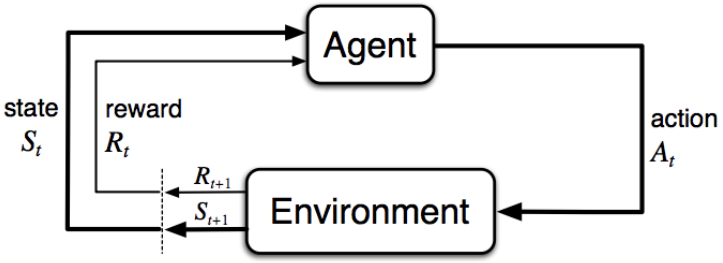
จากภาพประกอบที่ 2-5 แสดงให้เห็นประเภทของ Machine Learning นั้นสามารถแบ่งออกได้ 3 ประเภท ได้แก่

2.1) การเรียนรู้แบบมีผู้สอน (Supervised Learning) [10] เป็นกลุ่มของ Algorithm ที่เน้นสอน Computer โดยการศึกษาจากชุดข้อมูลตัวอย่าง ตัวอย่างเช่น หากต้องการให้

computer แยกภาพสุนัขออกจากภาพสัตว์ชนิดอื่น ๆ เราก็ต้องมีชุดข้อมูลภาพตัวอย่างของสุนัขป้อนให้ Computer เพื่อให้ว่ารูปภาพที่มีลักษณะแบบนั้นคือภาพสุนัข

2.2) การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) การเรียนรู้แบบไม่มีผู้สอนนี้จะตรงกันข้ามกับ Supervised Learning กล่าวคือ Computer สามารถเรียนรู้ได้โดยไม่มีการสอนหรือไม่มีชุดข้อมูลตัวอย่างมีเพียงชุดข้อมูลที่ประกอบไปด้วยคุณลักษณะเท่านั้น

2.3) การเรียนรู้ตามสภาพแวดล้อม (Reinforcement Learning) [11] หรือปัญญาประดิษฐ์ Artificial Intelligence (AI) ที่ใช้สำหรับพัฒนา robot (หรือ Agent) ให้สามารถตัดสินใจภายใต้แต่ละสถานการณ์เพื่อนำมาซึ่งผลลัพธ์ที่ดีที่สุด โดยที่ robot นั้นจะไม่ได้ถูกบอกให้รู้ถึงกฎเกณฑ์ในการเลือกกระทำสิ่งใดภายใต้สถานการณ์ใดโดยตรง แต่ robot จะพยายามพัฒนาระบบความคิด การตัดสินใจด้วยตนเองจากการลองผิดลองถูก และเรียนรู้ไปเรื่อย ๆ



ภาพประกอบที่ 2-6 ส่วนประกอบและความสัมพันธ์กันของ Reinforcement Learning

จากภาพประกอบที่ 2-6 แสดงให้เห็นส่วนประกอบและความสัมพันธ์กันของ Reinforcement Learning สามารถอธิบายในแต่ละส่วนได้ดังต่อไปนี้

2.4) Agent หรือ robot คือระบบที่ถูกสร้างขึ้นเพื่อให้ระบบสามารถตัดสินใจภายใต้สถานการณ์ที่แตกต่างกันได้ เพื่อให้ได้ผลลัพธ์มากที่สุด เปรียบเทียบ Agent คือหุ่นยนต์ที่รับคำสั่งเพื่อทำตามในสิ่งที่ต้องการ หรือการลองผิดลองถูก (trial-and-error)

2.5) Action (A) คือสิ่งที่ Agent สามารถทำได้ หากเปรียบเทียบเป็นเกมแล้ว Action คือปุ่มกดของเกม เป็นทางเลือกในการเล่น ผู้เล่นสามารถเลือกได้ว่าจะกดปุ่มใด

2.6) Environment & State (S) นั้นจะเปรียบเสมือนเป็นสถานะปัจจุบันของ Environment หรือสถานะก่อน ๆ หน้ารวมกัน Information บางอย่างที่ทำให้เรารู้ว่าปัจจุบันนี้เกมเป็นอย่างไรบ้าง หน้าตาเป็นอย่างไร เลือดของตัวละครเหลือเท่าไหร่ จำนวนชีวิตเหลือเท่าไหร่ ซึ่งจะเป็นข้อมูลที่เป็นประโยชน์ต่อ agent ในการเลือก Action ต่อไป

2.7) Reward (R) คือสิ่งที่ agent ต้องการที่จะ maximize หรือทำให้ได้รับมากที่สุด หากเปรียบเทียบเป็นเกม reward คือคะแนนในเกม เช่น ถ้าเราสร้าง agent สำหรับเล่น

เกม pong แล้ว agent จะได้รับ reward +1 ถ้าสามารถตีลูกข้ามผู้ต่อสู้ได้ หรือจะได้รับ reward -1 ถ้าไม่สามารถรับลูกที่ตีจากคู่ต่อสู้ได้

3) TensorFlow

TensorFlow เป็นไลบรารีสำหรับใช้พัฒนา Machine Learning เป็น Open source โดยเขียนด้วยภาษาไพทอน (Python) พัฒนาโดย Google [12] ซึ่งใช้ Machine Learning มาเพิ่มประสิทธิภาพกับผลิตภัณฑ์มากมาย ไม่ว่าจะเป็น เครื่องมือค้นหา (Search Engine), การแปลภาษา (Translation), คำบรรยายภาพ (Image Captioning) และ เครื่องมือช่วยการเสนอแนะ (Recommendations) Google นำ AI มาเอื้ออำนวยความสะดวกของผู้ใช้ ทั้งในแง่ความรวดเร็วของผลลัพธ์ และในแง่ผลลัพธ์ที่ถูกต้องแม่นยำมากขึ้น อย่างเช่น หากลองพิมพ์คำบางอย่างลงไปในช่องค้นหา Google สามารถแนะนำคำต่อไป หรือคำที่สมบูรณ์ให้เราได้ทันที อีกทั้งยังสามารถใช้ TensorFlow เพื่อสร้าง Deep Learning ได้ ตัวอย่างเช่น CNN และ RNN

4) Keras

Keras เป็น Deep Learning Library ในภาษาไพทอน (Python) [13] โดยสามารถรันบน CPU และเพิ่มประสิทธิภาพในการรันบน GPU ได้โดยการทำงานของ Keras จะทำงานบน TensorFlow หรือ Theano ซึ่งจัดเป็น Deep Learning Library ที่สมรรถนะสูงทั้งคู่ และข้อได้เปรียบของ Keras ได้แก่

4.1) Keras อนุญาตให้ใช้ Code เดียวกันในการรันทั้ง CPU และ GPU

4.2) API ง่ายต่อการใช้งาน และทำให้ตัวต้นแบบในการสร้าง Deep Learning Model ทำงานเสร็จได้อย่างรวดเร็ว

4.3) มีฟังก์ชัน Build-in ในการใช้งาน Convolution Network สำหรับงาน Computer Vision และงาน Recurrent Network สำหรับงาน Sequence Processing

4.4) สนับสนุนการทำงานของ Network หลากหลายรูปแบบ เช่น MIMO Models, Layer Sharing, Model Sharing และ อื่น ๆ สิ่งนี้หมายความว่า Keras เหมาะสำหรับการสร้าง Deep Learning Model ในทุก ๆ แบบ จาก Generative Adversarial Network ไปจนถึง Neural Turing Machine

2.2 สูตรการคำนวณหาความถูกต้องในการตรวจจับใบหน้า

$$Accuracy = Acc - Err$$

$$Acc = \frac{c * 100}{N}$$

$$Err = \frac{e*100}{N}$$

c = จำนวนใบหน้าที่ตรวจจับได้

e = จำนวนใบหน้าที่ผิดพลาด

N = จำนวนใบหน้าทั้งหมด

2.3 สูตรการคำนวณหาค่าความคล้ายคลึง (Cosine Similarity)

สูตรการคำนวณหา Cosine similarity เป็นการหาหาความคล้ายคลึงของชุดข้อมูล ซึ่งจะแสดงสูตรดังนี้

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

จากสมการหา Cosine Similarity ซึ่งค่า Cosine ที่ได้จะอยู่ระหว่าง 0 ถึง 1 โดยหากค่าที่ได้มีค่าเป็น 1 จะหมายถึงชุดข้อมูล A และชุดข้อมูล B มีความคล้ายคลึงกัน แต่หากค่าที่ได้มีค่าเป็น 0 จะหมายถึงชุดข้อมูล A และชุดข้อมูล B ไม่มีความคล้ายคลึงกัน

2.4 งานวิจัยที่เกี่ยวข้อง

2.4.1 การตรวจจับภาพใบหน้าด้วยเทคนิควิธีพิจารณาพื้นที่สีผิวร่วมกับตรวจสอบองค์ประกอบบนใบหน้า Face Detection using Hybrid detection Characteristic facial with Skin color based on Viola-Jones

การตรวจจับภาพใบหน้าด้วยเทคนิควิธีพิจารณาพื้นที่สีผิวร่วมกับตรวจสอบองค์ประกอบบนใบหน้าได้พัฒนาโดยสมประสงค์ กาบบัวลอย, อภิราม จึงมั่นคง, ธีรศักดิ์ เรืองจรัส, สิรภัทร เชี่ยวชาญ

วัฒนา และคำรณ สุนัติ [14] โดยบทความนี้ได้นำเสนอการตรวจจับใบหน้า (Face Detection) โดยใช้เทคนิคพื้นฐานของวิโอล่า-โจนส์ (Viola-Jones Algorithm) ซึ่งใช้วิธีการตรวจจับสีผิวร่วมกับการวิเคราะห์องค์ประกอบบนใบหน้า เช่น ตา จมูก และปาก ซึ่งใช้ข้อมูลทดลองเป็นภาพ 4 ประเภทคือ ประเภทภาพหน้าตรง ประเภทของหน้าเอียง ประเภทภาพกลุ่มหลายคน และประเภทภาพจากสื่อสิ่งพิมพ์ โดยเปรียบเทียบค่าความแม่นยำ และเวลาในการประมวลผลกับสองเทคนิควิธีวิธีคือ เทคนิควิธีการตรวจจับใบหน้าของวิโอล่า-โจนส์ และการตรวจจับจากพื้นที่สีผิวเพียงอย่างเดียว จากการทดลองพบว่า เทคนิควิธีวิโอล่า-โจนส์ให้ความแม่นยำ 91% และใช้เวลา 5.6 วินาที ในขณะที่เทคนิควิธีการตรวจจับจากพื้นที่สีผิวให้ความแม่นยำ 87.63% และใช้เวลา 6.5 วินาทีต่อภาพ ซึ่งเทคนิควิธีที่เสนอได้ผลลัพธ์ที่ดีกว่าวิธีวิโอล่า-โจนส์ และวิธีการตรวจจับเฉพาะสีผิวเพียงอย่างเดียว โดยมีความแม่นยำสูงถึง 98.5% และมีความเร็วเฉลี่ยเพียง 1.72 วินาทีต่อภาพ

2.4.2 การตรวจจับใบหน้าด้วยวิธีการพื้นฐานของการจำลองรูปแบบ Haar-like Face Detection based-on Haar-like Features

การตรวจจับใบหนาด้วยวิธีการพื้นฐานของการจำลองรูปแบบ Haar-like ได้พัฒนาโดย Viola-Jones ในปี 2001 [15] วิธีการตรวจจับใบหน้าของ Viola-Jones ประกอบด้วย 3 ขั้นตอนคือ การคำนวณการจำลองรูปแบบ Haar-like ด้วย Integral Image การคนหาการจำลองรูปแบบ Haar-like ด้วย Adaboost และการรวมตัวจำแนกกลุมแบบตอเรียง (Cascade Classifier) ซึ่งในการใชการจำลองรูปแบบ Haar-like นั้นถือวามีความสำคัญต่อความแม่นยำที่สุด เพราะเป็นเครื่องมือที่ใชในการดึงลักษณะเดนจากใบหนา จึงมีงานวิจัยเป็นจำนวนมากที่มุ่งเน้นในการพัฒนาการจำลองรูปแบบ Haar-like ดังนั้น บทความฉบับนี้จึงทำการรวบรวมและสรุปงานวิจัยที่มุ่งเน้นในการเพิ่มหรือปรับปรุงรูปร่างของการจำลองรูปแบบ Haar-like ซึ่งเป็นผลใหเพิ่มความเร็ว และความแม่นยำในการตรวจจับใบหนาหรือเพิ่มความสามารถในการตรวจจับใบหนาในมุมอื่น ๆ นอกเหนือจากใบหน้าตรง

2.4.3 ระบบตรวจจับใบหน้าและติดตามบุคคลผ่านกล้องวงจรปิด (CCTV Face Detection and Tracking System)

ระบบตรวจจับใบหน้า และติดตามบุคคลจากกล้องวงจรปิดได้พัฒนาโดย รศ.ดร. อรฉัตร จิตต์โสภักตร์, จตุพล เบญจประกายรัตน์ และชัยพิทักษ์ พัฒนกิตติคุณ [16] ในการทำโครงการเรื่องนี้คือ ระบบตรวจจับใบหน้า และติดตามบุคคลผ่านกล้องวงจรปิดจะติดตั้งภายในห้องว่างไม่มีสิ่งรบกวน โดยนำกล้องวงจรปิดมาเสริมระบบ Image processing เพื่อเพิ่มประสิทธิภาพ โดยมีสามส่วนที่ใช้ในระบบ คือ Human detection, Face detection และ Face recognition โดยใช้การตรวจจับใบหน้า และวิเคราะห์ในการระบุตัวบุคคล และใช้การตรวจจับใบหน้าบุคคลเพื่อระบุตำแหน่ง ซึ่งการ

ตรวจจับใบหน้าจะวิเคราะห์ได้ทีละคน เมื่อเข้าใช้ระบบจะสามารถแบ่งแยกบุคคลว่าเป็นใครได้ จะมีการบันทึกภาพ และตำแหน่งที่ตรวจจับได้ ซึ่งจะสามารถดูย้อนหลังได้พัฒนาบนภาษา C# และสามารถมอนิเตอร์ได้บน Android โดยโครงงานนี้เหมาะสำหรับใช้ในอาคารปิด ที่ต้องการระบบรักษาความปลอดภัยแบบระบุตัวตน รวมถึงสามารถติดตั้งกล้องวงจรปิดได้

บทที่ 3
ขั้นตอนการดำเนินงาน

3.1 ชุดข้อมูลที่เลือกใช้สำหรับการทดลอง

ชุดข้อมูลที่เลือกใช้ในการทดสอบการตรวจจับใบหน้า และการตรวจสอบใบหน้านั้น ประกอบด้วย 3 ชุด ได้แก่ The BioID Face, FERET และ ColorFERET ทางทีมผู้วิจัยมีการคัดเลือกรูปภาพที่มีความเหมาะสมมาใช้งาน ซึ่งในการทดสอบนั้นทางทีมผู้วิจัยได้แบ่งชุดข้อมูลออกเป็นใบหน้าบุคล ย่อย โดยมีรายละเอียดดังต่อไปนี้

3.1.1 ชุดข้อมูล The BioID Face

จากภาพประกอบที่ 3-1 จะเป็นการแสดงตัวอย่างของชุดข้อมูล The BioID Face โดยประกอบด้วยใบหน้าของคนจำนวน 21 คนที่แตกต่างกัน และจะแบ่งออกเป็น 21 ใบหน้าบุคคล รวมทั้งสิ้น 1,513 ใบหน้า (384 x 288 pixel, grayscale) ชุดข้อมูลนี้ถูกสร้างขึ้นโดยบริษัท BioID และได้บันทึกภาพใบหน้าบุคคลเมื่อมีการประชุมของบริษัท BioID โดยชุดข้อมูลนี้ถูกสร้างขึ้นเพื่อใช้ในการเปรียบเทียบประสิทธิภาพในการตรวจจับใบหน้าที่มีความรวดเร็ว และแม่นยำ [17]

ภาพประกอบที่ 3-1 ตัวอย่างของชุดข้อมูล The BioID Face

3.1.2 ชุดข้อมูล FERET และ ColorFERET

ชุดข้อมูลที่ใช้สำหรับการจดจำใบหน้า FERET และ ColorFERET ได้เผยแพร่เมื่อปี 1993 โดย J. Phillips และ P. Rauss [18, 19] ซึ่งชุดข้อมูลเหล่านี้ประกอบด้วย 1,199 ใบหน้าบุคคล และจำนวนภาพใบหน้าทั้งหมดคือ 14,126 ขนาดของรูปภาพ 384x 256 พิกเซล ในการทดลองของนี้ทีมผู้วิจัยได้ใช้ FERET และ ColorFERET สำหรับตรวจสอบใบหน้า และได้ทำการคัดเลือกรูปภาพใบหน้าจากชุดข้อมูล FERET จำนวน 1,372 ภาพ จาก 196 ใบหน้าบุคล โดยตัวอย่างของชุดข้อมูล FERET จะแสดงดังภาพประกอบที่ 3-2 สำหรับชุดข้อมูล ColorFERET ได้ทำการคัดเลือกรูปภาพใบหน้าจำนวน 3,553 ภาพ จาก 474 ใบหน้าบุคล และจะแสดงตัวอย่างของชุดข้อมูล ColorFERET ดังภาพประกอบที่ 3-3

ภาพประกอบที่ 3-2 ตัวอย่างของชุดข้อมูล FERET

ภาพประกอบที่ 3-3 ตัวอย่างของชุดข้อมูล ColorFERET

3.2 ขั้นตอนการตรวจจับใบหน้า

3.2.1 วิธีการที่ใช้สำหรับการทดลองการตรวจจับใบหน้า

สำหรับการตรวจจับใบหน้านั้นทางทีมผู้วิจัยได้เลือกใช้ 4 วิธี ดังต่อไปนี้

1) การตรวจจับใบหน้าด้วยวิธี Convolutional Neural Networks (CNNs)

Convolutional Neural Network (CNN) [20] หรือ โครงข่ายประสาทแบบคอนโวลูชัน เป็นโครงข่ายประสาทเทียมหนึ่งในกลุ่ม bio-inspired หลายเลเยอร์ที่มีโครงสร้างเฉพาะตัว โดยถูกออกแบบมาเพื่อเพิ่มความสามารถในการสกัดเอา feature ที่มีความซับซ้อนจากชุดข้อมูล โดย CNN นั้น มักจะถูกใช้เพื่อการสกัด feature จากข้อมูลประเภทที่ไม่ค่อยเป็นระเบียบหรือไม่ได้มีโครงสร้างเป็นรูปแบบเฉพาะตัว (unstructured data) อย่างเช่น รูปภาพ (image) กระบวนการทำงานมีดังนี้ รับรูปภาพ input เข้ามา ซึ่งรูปภาพเหล่านี้จะถูกจัดเก็บในรูปแบบของ pixel โดยทั่วไปแล้วจะใช้ 1 layer ในการเก็บข้อมูลในรูปแบบของเชดสีขาวดำ (greyscale) และจะใช้อีก 3 layer ในการเก็บข้อมูลในรูปแบบเชดสีต่าง ๆ ในระหว่างที่ model ทำการเรียนรู้ (feature learning) หรือสกัด feature ที่ hidden layer, model จะทำการหา feature ที่มีลักษณะสำคัญต่อชุดข้อมูล input ที่รับเข้ามา เมื่อ model ทำการเรียนรู้ (feature learning) เสร็จสมบูรณ์แล้วจะแสดงผลลัพธ์ของแต่ละรูปเป็นความน่าจะเป็น และหากรูปภาพนั้น ๆ มีความน่าจะเป็นแบบใดสูงที่สุด model จะตอบเป็นสิ่งนั้น

1.1) ตัวอย่างคำสั่งในการตรวจจับใบหน้าโดยวิธี Convolutional Neural Networks (CNNs) โดยจะแสดงในภาพประกอบที่ 3-4 โดยจะมีรายละเอียดดังต่อไปนี้

1.1.1) ต้องทำการนำเข้าโมดูล mmod_human_face_detector.dat เพื่อใช้ในการตรวจสอบด้วยวิธี CNN

1.1.2) ในกรณีที่บางรูปภาพนั้นตำแหน่งของใบหน้านั้นไม่ได้อยู่ในตำแหน่งที่มีค่ามากกว่า 0 หรือเป็นค่าติดลบจึงจำเป็นต้องทำให้มีค่าเป็น 0 ก่อนจึงจะนำเข้าไปทำการตรวจจับใบหน้า และครอบใบหน้าต่อไป

```
cnn_face_detector =
dlib.cnn_face_detection_model_v1('mmod_human_face_detector.dat')
dets = cnn_face_detector(img, 1)
for i, d in enumerate(dets):
    top = d.rect.top()
    left = d.rect.left()
    right = d.rect.right()
    bottom = d.rect.bottom()
    if(top < 0):
      top = 0
    if(left < 0):
      left = 0
    if(right < 0):
       right = 0
    if(bottom < 0):
       bottom = 0
    crop_img_cnn = img[top:bottom,left:right]
```

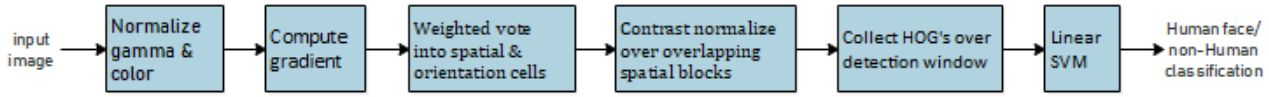
ภาพประกอบที่ 3-4 ตัวอย่างโค้ดในการตรวจจับใบหน้าโดยวิธี Convolutional Neural Networks (CNNs)

2) การตรวจจับใบหน้าด้วยวิธี Histograms of Oriented Gradients (HOG)

Histograms of Oriented Gradients (HOG) หรือค่าความถี่ของทิศทางตามค่าเกรเดียนท์ เป็นวิธีการดึงคุณลักษณะเฉพาะของวัตถุซึ่งสามารถดึงรูปร่างภายในภาพโดยใช้การกระจายตัวของความเข้มเกรเดียนท์หรือทิศทางของเส้นขอบ การดึงคุณลักษณะของ HOG จะทำได้โดยการแบ่งภาพออกเป็นส่วนย่อยขนาดเล็กหรือเซลล์ (Cells) โดยแต่ละเซลล์จะรวบรวมฮิสโตรแกรมของทิศทางเกรเดียนท์หรือทิศทางของขอบภายในเซลล์ที่มีขนาดหนึ่งมิติ (1-D) โดยที่จะมีการรวมฮิสโตรแกรมนั้นเข้าด้วยกัน เพื่อแสดงถึงคุณลักษณะ
เฉพาะของวัตถุที่สนใจ เพื่อให้มีประสิทธิภาพของความถูกต้องเพิ่มมากขึ้น สามารถนำฮิสโตรแกรมมาทำนอร์มอลไลซ์ด้วยการคำนวณตัวชี้วัดของค่าความเข้มทั่วทั้งพื้นที่ขนาดใหญ่ของภาพหรือบล็อค (Block) การทำนอร์มอลไลซ์ ต้องทำทุกเซลล์ภายในบล็อค ผลลัพธ์จากการทำนอร์มอไลซ์จะทำให้

ผลกระทบจากการเปลี่ยนแปลงของแสงสว่าง และเงาน้อยลง สามารถหาคุณลักษณะเฉพาะของวัตถุได้ดีมากยิ่งขึ้น

ภาพรวมของการหาค่าความถี่ของทิศทางตามค่าเกรเดียนท์หรือ Histograms of Oriented Gradients (HOG) มีด้วยกันทั้งหมด 6 ขั้นตอน ดังภาพประกอบที่ 3-5



ภาพประกอบที่ 3-5 ภาพรวมของการหาค่าความถี่ของทิศทางตามค่าเกรเดียนท์หรือ Histograms of Oriented Gradients (HOG) [1]

หน้าต่างในการตรวจจับจะเรียงต่อกันเป็นแผ่นที่ทับซ้อนกัน ซึ่งใช้ค่าความถี่ตามทิศทางแกรเดียนท์มาเป็นคุณสมบัติในการจำแนกคุณลักษณะเฉพาะ การรวมตัวกันของเวกเตอร์จะใช้ Support Vector Machine (SVM) จำแนกประเภทที่เป็นวัตถุและไม่ใช่วัตถุ หน้าต่างในการตรวจจับจะถูกแสกนทุกตำแหน่งของและขนาดของภาพ ขั้นตอนในการสกัดคุณลักษณะเฉพาะของภาพเมื่อรับภาพเข้ามาในระบบ มีดังนี้

ขั้นตอนที่ 1 ทำการนอร์มอไลซ์จากค่าความสว่าง (Gamma) และสี (Color)

ขั้นตอนที่ 2 ทำการคำนวณหาค่าเกรเดียนท์

ขั้นตอนที่ 3 ทำการโหวตหาค่าถ่วงน้ำหนักของระยะห่างที่สอดคล้องกันของเซลล์ (Cells)

ขั้นตอนที่ 4 ทำการนอร์มอไลซ์พื้นที่ที่ซับซ้อนกันของบล๊อค (Block)

ขั้นตอนที่ 5 รวบรวมหน้าต่างของ HOG สำหรับการตรวจจับทั้งหมด

ขั้นตอนที่ 6 ทำการจำแนกคุณลักษณะเฉพาะด้วย Support Vector Machine (SVM) แบบเชิงเส้น (Linear)

2.1) ตัวอย่างคำสั่งในการตรวจจับใบหน้าโดยวิธี Histograms of Oriented Gradients (HOG) โดยจะแสดงในภาพประกอบที่ 3-6 โดยจะมีรายละเอียดดังต่อไปนี้

2.1.1) ในการตรวจจับใบหน้าด้วยวิธี HOG จำเป็นจะต้องใช้โมดูล dlib เข้ามาช่วยดังนั้นจึงต้องใช้คำสั่ง import dlib

2.1.2) ในกรณีที่บางรูปภาพตำแหน่งของใบหน้านั้นไม่ได้อยู่ในตำแหน่งที่มีค่ามากกว่า 0 หรือเป็นค่าติดลบนั้น จึงจำเป็นต้องทำให้มีค่าเป็น 0 ก่อนที่จะนำเข้าไปทำการตรวจจับ และ
ครอปใบหน้าต่อไป

```
import dlib
      detector = dlib.get_frontal_face_detector()
      dets = detector
      for i,d in enumerate(dets):
         top = d.top()
              left = d.left()
              right = d.right()
              bottom = d.bottom()
               if(top < 0):
                 top = 0
               if(left < 0):
                 left = 0
               if(right < 0):
                 right = 0
               if(bottom < 0):
                 bottom = 0
           crop_img_hog =[top:bottom,left:right]
```

ภาพประกอบที่ 3-6 ตัวอย่าง code ที่ใช้ในการตรวจจับใบหน้าด้วยวิธี Histograms of Oriented Gradients (HOG)

3) การตรวจจับใบหน้าด้วยวิธี Haar-Cascade Classifier

ในการใช้งานจำเป็นต้องมีไฟล์ที่ได้รับการเรียนรู้ และแยกประเภทหรือมีข้อมูลการเรียนรู้แล้วว่า วัตถุนั้นหรือสิ่งเหล่านั้นคืออะไร โดยไฟล์จะมีนามสกุล .XML โดยทาง OpenCV จะมีไฟล์ XML มาให้อยู่แล้วใน Library มีทั้งหมด 2 ไฟล์ดังนี้ haarcascade_frontalface_default.xml และ haarcascade_
eye.xml โดยเทคนิคการตรวจจับใบหน้าของ Haar-Cascade Classifier นี้สามารถแบงออกเป็น 3 ขั้นตอน ดังต่อไปนี้

- การคำนวณรูปแบบการจำลองด้วย Integral Image
- การคนหารูปแบบการจำลองด้วย Adaboost
- การรวมตัวจำแนกกลุมแบบตอเรียง (Cascaded Classifier)

3.1) ตัวอย่างคำสั่งในการตรวจจับใบหน้าโดยวิธี Haar-Cascade Classifier โดยจะแสดงในภาพประกอบที่ 3-7 โดยจะมีรายละเอียดดังต่อไปนี้

3.1.1) โดยในวิธีนี้จะทำการแปลงรูปสี (RGB) ให้เป็นรูปขาวดำ (Gray) จากนั้นจะนำรูปภาพที่เป็นสีขาวดำไปประมวลผลการตรวจจับใบหน้า และในตัวอย่างนี้เมื่อตรวจจับเสร็จสิ้นกระบวนการแล้วจะทำการเก็บลงตัวแปรที่ชื่อว่า crop_img_haar

```
gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
face_cascade =
cv.CascadeClassifier('haarcascade_frontalface_default.x
ml')
faces = face_cascade.detectMultiScale(gray, 1.2, 4)
for (x,y,w,h) in faces:
```

ภาพประกอบที่ 3-7 ตัวอย่างคำสั่งที่ใช้ในการตรวจจับใบหน้าด้วยวิธี Haar-Cascade Classifier

4) การตรวจจับใบหน้าด้วยวิธี Faced

Faced เป็นชุดของ 2 โครงข่ายประสามเทียม (ดำเนินการโดยใช้เมตริกซ์) ออกแบบมาเพื่อการทำงานที่รวดเร็วขึ้นตามเวลาจริงในหน่วยประมวลผลกลางของเครื่องคอมพิวเตอร์ Central Processing Unit (CPU) แบ่งเป็นขั้นตอนได้ทั้งหมด 2 ขั้นตอน ดังต่อไปนี้

ขั้นตอนที่ 1 : การปรับใช้เครือข่ายประสาทเทียม fully convolutional neural network (FCNN) แบบกำหนดเองตาม YOLO คือ การนำรูปภาพสี (RGB) ขนาด 288x288 และผลตาราง 9x9 ซึ่งแต่ละเซลล์สามารถคาดการณ์ตำแหน่ง และความน่าจะเป็นของ 1 ใบหน้า

ขั้นตอนที่ 2 : มาตรฐานที่กำหนดของโครงข่ายประสาทแบบคอนโวลูชัน Convolutional Neural Network (CNN) (convolutions + Fully Connected layers) คือ ใช้เพื่อถ่ายภาพสี่เหลี่ยมบนใบหน้า และทำนายตำแหน่งของใบหน้า ขั้นตอนนี้คือ การปรับแต่ง (ผลลัพธ์ของขั้นตอนที่ 1 นั้นอาจไม่ถูกต้องมากนัก ขั้นตอนนี้จึงเป็นการแก้ไขตำแหน่งแต่ละส่วนที่คาดการณ์ไว้จากขั้นตอนก่อนหน้านี้เพื่อปรับปรุงคุณภาพของตำแหน่งให้มีความถูกต้องมากยิ่งขึ้น)

4.1) ตัวอย่างคำสั่งในการตรวจจับใบหน้าโดยวิธี Faced โดยจะแสดงในภาพประกอบที่ 3–8 และ 3-9 โดยจะมีรายละเอียดดังต่อไปนี้

4.1.1) ภาพประกอบที่ 3-8 จะแสดงให้เห็นถึงคำสั่งการติดตั้งโปรแกรม faced ก่อนที่จะทำการประมวลผล

```
pip install git+https://github.com/iitzco/faced.git
```

ภาพประกอบที่ 3-8 คำสั่งการติดตั้งโปรแกรม faced

4.1.2) ในการตรวจจับใบหน้าด้วยวิธี faced นั้นจะทำการแปลงค่าของสีรูปภาพจาก BGR ไปเป็น RGB ก่อน (ในการใช้ cv2 อ่านรูปภาพเข้ามานั้นค่าของรูปภาพจะเป็น BGR) จึงจะเข้าสู่ขั้นตอนตรวจจับตำแหน่งของใบหน้า

```
from faced import FaceDetector
from faced.utils import annotate_image
face_detector_faced = FaceDetector()
rgb_img = cv.cvtColor(img.copy(), cv.COLOR_BGR2RGB)
bboxes = face_detector_faced.predict(rgb_img)
img_h, img_w, _ = img.shape
for x,y,w,h,p in bboxes:
     faces_img_faced = img[int(y - w/2):int(y + h/2),int(x -
                          w/2):int(x + h/2)]
```

ภาพประกอบที่ 3-9 ตัวอย่างคำสั่งที่ใช้ในการตรวจจับใบหน้าด้วยวิธี Faced

โดยคำสั่งการทำงานของโปรแกรมตรวจจับใบหน้านั้นจะแสดงดังภาพประกอบที่ 3–10 ถึงภาพประกอบที่ 3–13 และจะมีรายละเอียดดังต่อไปนี้

1) นำที่อยู่ของรูปภาพเก็บไว้ในตัวแปร dirName

2) ส่ง dirName เข้าไปประมวลผล ในการดึงรูปภาพแต่ละรูปออกมาโดยใช้ function dir2filename และจะส่งเข้าไปในรูปของตัวแปร list_filenames การทำงานของ dir2filename นั้นมีดังต่อไปนี้ function dir2filename

```
dirName = "your destination"
list_filenames = list()
list_filenames = utils.dir2filename(dirName)
face_img = face_detectors.### (list_filenames)
```

ภาพประกอบที่ 3- 10 คำสั่งการทำงานของโปรแกรมตรวจจับใบหน้า (1)

2.1) รับค่า list_filenames เข้ามาในฟังก์ชัน

2.2) อ่านไฟล์จาก list_filenames ด้วยคำสั่ง os.walk ซึ่งเป็นคำสั่งในการสแกนไฟล์ในภาษาไพธอน

2.3) ทำการอ่านค่าจาก filenames ด้วยคำสั่ง fnmatch.filter โดยจะเลือกอ่านเฉพาะไฟล์ที่เป็น jpg หรือ pgm เท่านั้น

3) เมื่อทำการอ่านไฟล์รูปภาพมาเก็บไว้ที่ list_filenames แล้ว ขั้นตอนต่อไปจะทำการส่งรูปเข้าไปเพื่อตรวจสอบหาใบหน้าโดยจะส่งไฟล์รูปเข้าไปในฟังก์ชันของแต่ละวิธีที่ใช้ในการทดสอบ และตรวจสอบใบหน้า

```
def dir2filename(self, list_filenames):
     matches = []
     for root, dirnames, filenames in
os.walk(list_filenames):
            for filename in fnmatch.filter(filenames,
'*.[jpg]*[pgm]'):
                matches.append(os.path.join(root,filename))
     return matches
```

ภาพประกอบที่ 3-11 คำสั่งการทำงานของโปรแกรมตรวจจับใบหน้า (2)

4) สามารถอ่านรูปภาพได้โดยใช้คำสั่ง io.imread ออกมากจาก list ที่เก็บรูปภาพไว้ หรือในภาพประกอบที่ 3–12 นี้คือตัวแปร fn

```
for fn in face_img :
     img = io.imread(fn)
```

ภาพประกอบที่ 3-12 คำสั่งการทำงานของโปรแกรมตรวจจับใบหน้า (3)

5) ในภาพประกอบที่ 3–13 นั้นการตั้งชื่อไฟล์นั้นจะตั้งชื่อไฟล์โดยการใช้ timestamp และตัวแปร cnt เข้ามาเพื่อช่วยไม่ให้ชื่อไฟล์นั้นซ้ำกัน

6) ในการบันทึกรูปภาพนั้นในภาพประกอบที่ 3–13 นี้จะใช้วิธีการจาก skimage โดยใช้คำสั่ง io.imsave

```
cnt = 1
target = os.path.join("your destination ")
tmp_prefix_name = "your destination"
timestamp = time.strftime("%Y%m%d-%H%M%S")

save_dir_path = target + '/' + tmp_prefix_name + timestamp
               + '-'+ str(cnt) + '.png'
               io.imsave
               (save_dir_path, 'your variable of image')

cnt = cnt+1
```

ภาพประกอบที่ 3-13 คำสั่งการทำงานของโปรแกรมตรวจจับใบหน้า (4)

3.3 ขั้นตอนในการหาคุณลักษณะพิเศษของใบหน้า

3.3.1 วิธีการที่ใช้สำหรับการทดลองในการหาคุณลักษณะพิเศษของใบหน้า

สำหรับการตรวจสอบ หรือการหาคุณลักษณะพิเศษของใบหน้าบุคคล (Feature Extraction) นั้นทางทีมผู้วิจัยได้เลือกใช้ 3 วิธี ดังต่อไปนี้

1) การหาคุณลักษณะพิเศษด้วยวิธี ResNet-50

ResNet-50 เป็นเครือข่ายประสาทเทียมที่ได้รับการเทรนภาพมากกว่าล้านภาพจากฐานข้อมูล ImageNet โดย ResNet-50 นั้นมีความลึก 50 ชั้น และยังสามารถจัดประเภทรูปภาพได้ถึง 1,000 ประเภทเช่น แป้นพิมพ์, เมาส์, ดินสอ และสัตว์อีกหลายชนิด ด้วยเหตุนี้เอง ResNet-50 จึงสามารถเรียนรู้คุณสมบัติของรูปภาพที่หลากหลายได้ ซึ่ง ResNet-50 นี้มีขนาดรูปภาพ 224x224 [2]

1.1) ตัวอย่างคำสั่งในการตรวจสอบใบหน้าโดยวิธี ResNet-50 จะมีรายละเอียดดังต่อไปนี้

1.1.1) จากภาพประกอบที่ 3–14 จะแสดงให้เห็นถึงวิธีการกำหนดขนาดของรูปภาพ และการนำ Model ของ ResNet-50 เข้ามาใช้ในโปรแกรม โดยจำเป็นต้องกำหนดขนาดของรูปภาพที่จะนำไปทำการหาคุณลักษณะพิเศษของใบหน้าบุคคลโดยในโมเดลนี้นั้นกำหนดให้มีขนาด 224, 224, 3 จากนั้นทำการ import model มาเก็บไว้ที่ตัวแปร resnet50_features เพื่อใช้ในขั้นตอนต่อไป

```
resnet50_features = VGGFace(model='resnet50',
                    include_top=False, input_shape=(224,
                    224, 3),pooling='avg')
```

ภาพประกอบที่ 3-14 วิธีการกำหนดขนาดของรูปภาพ และการนำ Model ของ ResNet-50 เข้ามาใช้ในโปรแกรม

1.1.2) จากภาพประกอบที่ 3–15 จะแสดงให้เห็นถึงวิธีการอ่านและแปลงขนาดของรูปภาพรูปภาพจากตัวแปร image_path โดยในขั้นแรกจะใช้คำสั่งจาก opencv นั่นคือ cv.imread ในการอ่านรูปภาพ และก่อนที่จะนำไปหาคุณลักษณะพิเศษของใบหน้าบุคคลนั้นจะต้องนำไปทำการแปลงขนาดของรูปภาพ โดยใช้คำสั่ง cv.resize ให้มีขนาดเท่ากับที่กำหนดไว้ข้างต้นนั่นคือ 224, 224 จากนั้นจึงนำไปเก็บไว้ที่ตัวแปร detected_face

```
img = cv.imread(image_path)
detected_face = cv.resize(img, (224, 224))
```

ภาพประกอบที่ 3-15 การอ่าน และแปลงขนาดของรูปภาพ

1.1.3) จากภาพประกอบที่ 3–16 จะแสดงให้เห็นถึงขั้นตอนการนำรูปภาพเข้าไปประมวลผลเพื่อหาคุณลักษณะพิเศษใบหน้าบุคคลของวิธี ResNet-50 ในตัวอย่างนี้จะประกาศตัวแปรที่ชื่อว่า feature_list เป็นประเภท list หรือ array เพื่อใช้ในการเก็บ feature ของแต่ละภาพ จากนั้นนำตัวแปร detected_face ที่เก็บรูปภาพที่แปลงขนาดแล้วมาทำการเปลี่ยน type เป็น array ด้วยคำสั่ง img_to_array และนำเข้าไปประมวลผลเพื่อหาค่า feature ด้วยคำสั่ง resnet50_features.predict เมื่อได้ค่าออกมาแล้วจะนำไปเก็บไว้ที่ตัวแปร feature_list ด้วยคำสั่ง append

```
features_list = []

x = image.img_to_array(detected_face)
x = np.expand_dims(x, axis=0)
x = utils.preprocess_input(x, version=1)
features = resnet50_features.predict(x)
features_list.append(features_np.flatten())
```

ภาพประกอบที่ 3-16 ขั้นตอนการนำรูปภาพไปประมวลผลหาคุณลักษณะพิเศษใบหน้าบุคคลของวิธี ResNet-50

2) การหาคุณลักษณะพิเศษด้วยวิธี FaceNet

FaceNet เป็นระบบจดจำใบหน้าที่อธิบายโดย Florian Schroff พร้อมคณะที่บริษัท Google ในปี 2015 ในบทความงานวิจัยเรื่อง "FaceNet: Unified Embeded for Face Recognition and Clustering" โดยเป็นระบบที่จะแยก feature ออกจากภาพใบหน้า จากนั้นจะทำการพยากรณ์องค์ประกอบทั้ง 128 จุดบนในหน้าซึ่งองค์ประกอบเหล่านี้เรียกว่าการฝังใบหน้า และใน FaceNet นี้เป็นโครงข่ายประสาทเทียมเชิงลึกที่ได้รับการเทรนผ่านฟังก์ชัน triplet ที่ส่งเสริมเวกเตอร์สำหรับตัวตนเดียวกันให้มีความคล้ายกันมากขึ้น ในขณะที่เวกเตอร์สำหรับตัวตนแตกต่างกันให้มีความคล้ายกันน้อย ซึ่งจะมุ่งเน้นไปที่การเทรนเพื่อสร้าง embeddings หรือการฝันใบหน้าโดยตรง [21]

2.1) ตัวอย่างคำสั่งในการตรวจสอบใบหน้าโดยวิธี FaceNet จะมีรายละเอียดดังต่อไปนี้

2.1.1) จากภาพประกอบที่ 3–17 จะแสดงให้เห็นถึงวิธีการนำ Model ของ FaceNet เข้ามาใช้ในโปรแกรม โดยจำเป็นต้อง import FaceNet ซึ่งเป็นโมดูลที่จะเลือกใช้ จากนั้นจะเรียกใช้โดยการใช้คำสั่ง FaceNet() และจัดเก็บไว้ในตัวแปรที่ชื่อว่า embedder เพื่อใช้ในขั้นตอนต่อไป

```
from keras_facenet import FaceNet
embedder = FaceNet()
```

ภาพประกอบที่ 3-17 วิธีการนำ Model ของ FaceNet เข้ามาใช้ในโปรแกรม

2.1.2) จากภาพประกอบที่ 3–18 จะแสดงให้เห็นถึงวิธีการอ่านและแปลงขนาดของรูปภาพรูปภาพจากตัวแปร image_path โดยในขั้นแรกจะใช้คำสั่งจาก opencv นั่นคือ cv.imread ในการอ่านรูปภาพ และก่อนที่จะนำไปหาคุณลักษณะพิเศษของใบหน้าบุคคลนั้นจะต้องนำไปทำการแปลงขนาดของรูปภาพ โดยใช้คำสั่ง cv.resize ให้มีขนาดเท่ากับที่ที่เหมาะในการนำไปประมวลผลนั่นคือ 224, 224 จากนั้นจึงนำไปเก็บไว้ที่ตัวแปร detected_face

```
img = cv.imread(image_path)
detected_face = cv.resize(img, (224, 224))
```

ภาพประกอบที่ 3-18 การอ่าน และแปลงขนาดของรูปภาพ

2.1.3) จากภาพประกอบที่ 3–19 จะแสดงให้เห็นถึงขั้นตอนการนำรูปภาพเข้าไปประมวลผลเพื่อหาคุณลักษณะพิเศษใบหน้าบุคคลของวิธี FaceNet ในตัวอย่างนี้จะประกาศตัวแปรที่ชื่อว่า feature_list เป็นประเภท list หรือ array เพื่อใช้ในการเก็บ feature ของแต่ละภาพ จากนั้นนำตัวแปร detected_face ที่เก็บรูปภาพที่แปลงขนาดแล้วมาทำการเปลี่ยน type เป็น array ด้วยคำสั่ง img_to_array และนำเข้าไปประมวลผลเพื่อหาค่า feature ด้วยคำสั่ง embedder.embeddings เมื่อได้ค่าออกมาแล้วจะนำไปเก็บไว้ที่ตัวแปร feature_list ด้วยคำสั่ง append

```
features_list = []

x = image.img_to_array(detected_face)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)
features = embedder.embeddings(x)
features_list.append(features_np.flatten())
```

ภาพประกอบที่ 3-19 ขั้นตอนการนำรูปภาพไปประมวลผลหาคุณลักษณะพิเศษใบหน้าบุคคลของวิธี FaceNet

3) การหาคุณลักษณะพิเศษด้วยวิธี VGG16

VGG16 เป็นรูปแบบโครงข่ายประสาทเทียมที่เสนอโดย K. Simonyan และ A. Zisserman จาก University of Oxford ในบทความงานวิจัยเรื่อง "Very Deep Convolutional Networks for Large-Scale Image Recognition" แบบจำลองนี้ได้รับการทดสอบความแม่นยำโดยค่าความแม่นยำสูงสุดที่ 92.7% จาก 5 อันดับแรกจากฐานข้อมูล ImageNet ซึ่งเป็นชุดข้อมูลของรูปภาพมากกว่า 14 ล้านภาพสามารถจัดประเภทรูปภาพได้ถึง 1,000 ประเภท เป็นหนึ่งในรูปแบบที่มีชื่อเสียงได้นำเสนอที่งาน ILSVRC-2014 เป็นการพัฒนาให้ดีกว่า AlexNet โดยการเปลี่ยนฟิลเตอร์เคอร์เนลให้มีขนาดใหญ่ (11 และ 5 ในชั้นแรกและชั้นที่สองตามลำดับ) ด้วยตัวกรองขนาดเคอร์เนล 3×3 ซึ่ง VGG16 ได้รับการฝึกฝนเป็นเวลาหลายสัปดาห์และใช้ NVIDIA Titan Black GPU ในการช่วยพัฒนา [22]

3.1) ตัวอย่างคำสั่งในการตรวจสอบใบหน้าโดยวิธี VGG16 จะมีรายละเอียดดังต่อไปนี้

3.1.1) จากภาพประกอบที่ 3–20 จะแสดงให้เห็นถึงวิธีการกำหนดขนาดของรูปภาพ และการนำ Model ของ VGG16 เข้ามาใช้ในโปรแกรม โดยจำเป็นต้องกำหนดขนาดของรูปภาพที่จะนำไปทำการหาคุณลักษณะพิเศษของใบหน้าบุคคลโดยในโมเดลนี้นั้นกำหนดให้มีขนาด 224, 224, 3 จากนั้นทำการ import model มาเก็บไว้ที่ตัวแปร model เพื่อใช้ในขั้นตอนต่อไป

```
image_input = Input(shape=(224,224,3))
model =
VGG16(include_top=False,weights="imagenet",input_tensor=image_input
)
```

ภาพประกอบที่ 3-20 วิธีการกำหนดขนาดของรูปภาพ และการนำ Model ของ VGG16 เข้ามาใช้ในโปรแกรม

3.1.2) จากภาพประกอบที่ 3–21 จะแสดงให้เห็นถึงวิธีการอ่านและแปลงขนาดของรูปภาพรูปภาพจากตัวแปร image_path โดยในขั้นแรกจะใช้คำสั่งจาก opencv นั่นคือ cv.imread ในการอ่านรูปภาพ และก่อนที่จะนำไปหาคุณลักษณะพิเศษของใบหน้าบุคคลนั้นจะต้องนำไปทำการแปลงขนาดของรูปภาพ โดยใช้คำสั่ง cv.resize ให้มีขนาดเท่ากับที่กำหนดไว้ข้างต้นนั่นคือ 224, 224 จากนั้นจึงนำไปเก็บไว้ที่ตัวแปร detected_face

```
img = cv.imread(image_path)
detected_face = cv.resize(img, (224, 224))
```

ภาพประกอบที่ 3-21 การอ่าน และแปลงขนาดของรูปภาพ

3.1.3) จากภาพประกอบที่ 3–22 จะแสดงให้เห็นถึงขั้นตอนการนำรูปภาพเข้าไปประมวลผลเพื่อหาคุณลักษณะพิเศษใบหน้าบุคคลของวิธี VGG16 ในตัวอย่างนี้จะประกาศตัวแปรที่ชื่อว่า feature_list เป็นประเภท list หรือ array เพื่อใช้ในการเก็บ feature ของแต่ละภาพ จากนั้นนำตัวแปร detected_face ที่เก็บรูปภาพที่แปลงขนาดแล้วมาทำการเปลี่ยน type เป็น array ด้วยคำสั่ง img_to_array และนำเข้าไปประมวลผลเพื่อหาค่า feature ด้วยคำสั่ง model.predict เมื่อได้ค่าออกมาแล้วจะนำไปเก็บไว้ที่ตัวแปร feature_list ด้วยคำสั่ง append

```
features_list = []

x = image.img_to_array(detected_face)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)
features = model.predict(x)
features_list.append(features_np.flatten())
```

ภาพประกอบที่ 3-22 ขั้นตอนการนำรูปภาพไปประมวลผลหาคุณลักษณะพิเศษใบหน้าบุคคลของวิธี VGG16

3.4 ขั้นตอนในการตรวจสอบใบหน้า

จากภาพประกอบที่ 3-1 ในขั้นแรกจะนำตัวแปรที่เก็บค่าคุณลักษณะพิเศษของใบหน้าบุคคล (features_list) มาเก็บไว้ที่ตัวแปร X จากนั้นนำตัวแปร fn_label ที่ใช้ในการเก็บชื่อไดเรกทอรี่ของแต่ละบุคคลมาทำการแปลงให้เป็นชนิด int พร้อมทั้งจัดเก็บไว้ในตัวแปร y_strs และนำไปเก็บไว้ที่ตัวแปร y เพื่อใช้ในการประมวลผลในข้อตอนถัดไป

```
X = features_list
y_strs = []
acc_item = []
for i in range(0, len(fn_label)):
    fn_label[i] = int(fn_label[i])
    y_strs.append(fn_label[i])
y = y_strs
```

ภาพประกอบที่ 3-23 ขั้นตอนในการหาคุณลักษณะพิเศษของใบหน้าบุคคล (1)

จากภาพประกอบที่ 3-24 จะแสดงให้เห็นว่าการทดลองการตรวจสอบใบหน้านั้นจะทำ 5 รอบด้วยกัน โดยเริ่มจากการนำตัวแปร X และ y ที่ได้จัดเก็บข้อมูลดังที่กล่าวไปข้างต้นส่งเข้าไปในฟังก์ชั่น train_test_split โดยจะกำหนด test size ให้มีขนาด 0.2 และจะประกาศตัวแปร X_train, X_test, y_train และ y_test เพื่อรับค่าที่รีเทิร์นกลับมา โดยในฟังก์ชันนี้จะทำการ random ชุดข้อมูล X , y ให้เป็นข้อมูลเพื่อที่จะทำการ test และ train ในขั้นตอนถัดไป

```
for epoch in range(0,5):
    X_train, X_test, y_train, y_test =
     train_test_split(X, y, test_size=0.2)
```

ภาพประกอบที่ 3-24 ขั้นตอนในการหาคุณลักษณะพิเศษของใบหน้าบุคคล (2)

จากภาพประกอบที่ 3-25 จะนำตัวแปร X_test และ X_train ซึ่งเป็นค่าคุณลักษณะพิเศษของใบหน้านำไปคำนวณในฟังก์ชัน distance cosine เพื่อให้ได้ค่า similarity จากนั้นนำค่าที่ได้มาไปหาค่าสูงสุด และจัดเก็บลงที่ตัวแปร tmp_max_simm_index

```
max_simm = []
max_simm_y = []
ans_true_false = []
for i in range(len(X_test)):
     for j in range(len(X_train)):
       simm = 1-spatial.distance.cosine(X_test[i],X_train[j])
       max_simm.append(simm)
     tmp_max = max(max_simm)
     tmp_max_simm_index = max_simm.index(tmp_max)
```

ภาพประกอบที่ 3-25 ขั้นตอนในการหาคุณลักษณะพิเศษของใบหน้าบุคคล (3)

จากภาพประกอบที่ 3-26 จะทำการนับจำนวนในตัวแปร X_test และเก็บไว้ในตัวแปร num เพื่อจะนำไปคำนวณหาค่าความถูกต้องแม่นยำ และนำไปเก็บไว้ที่ตัวแปร acc_item

```
num = len(X_test)
    acc = (float(ans_true_false.count(True)) / num) * 100.0
    acc_item.append(acc)
```

ภาพประกอบที่ 3-26 ขั้นตอนในการหาคุณลักษณะพิเศษของใบหน้าบุคคล (4)

จากภาพประกอบที่ 3-27 จะทำการเปรียบเทียบตัวแปร y_test กับ y_train หากมีค่าตรงกันจะนับว่า true แต่ถ้าไม่ตรงกันจะเป็น false

```
if(y_test[i] == y_train[tmp_max_simm_index]):
     ans_true_false.append(True)
else:
     ans_true_false.append(False)
```

ภาพประกอบที่ 3-27 ขั้นตอนในการหาคุณลักษณะพิเศษของใบหน้าบุคคล (5)

3.5 อุปกรณ์ที่ใช้ในการรับภาพ

สำหรับอุปกรณ์ที่ใช้ในการรับภาพของการทดลองนี้ ทางทีมผู้วิจัยได้เลือกใช้กล้องรุ่น DCS-942L ซึ่งแสดงดังภาพประกอบที่ 3–28

ภาพประกอบที่ 3-28 กล้องรุ่น DCS-942L

โดยคุณสมบัติของกล้องรุ่น DCS-942L นั้นสามารถเชื่อมต่ออินเตอร์ได้แบบไร้สาย และสามารถส่งข้อมูลภาพได้แบบเรียลไทม์ อีกทั้งกล้องนั้นยังมีโหมด Infrared ทำให้สามารถใช้งานได้ในที่

ที่มีแสงน้อยตัว

อย่างของรูปภาพที่ถ่ายโดยกล้อง DCS-942L นั้นจะแสดงดังภาพประกอบที่ 3-29 และภาพประกอบที่ 3–30

ภาพประกอบที่ 3-29 ภาพสว่าง

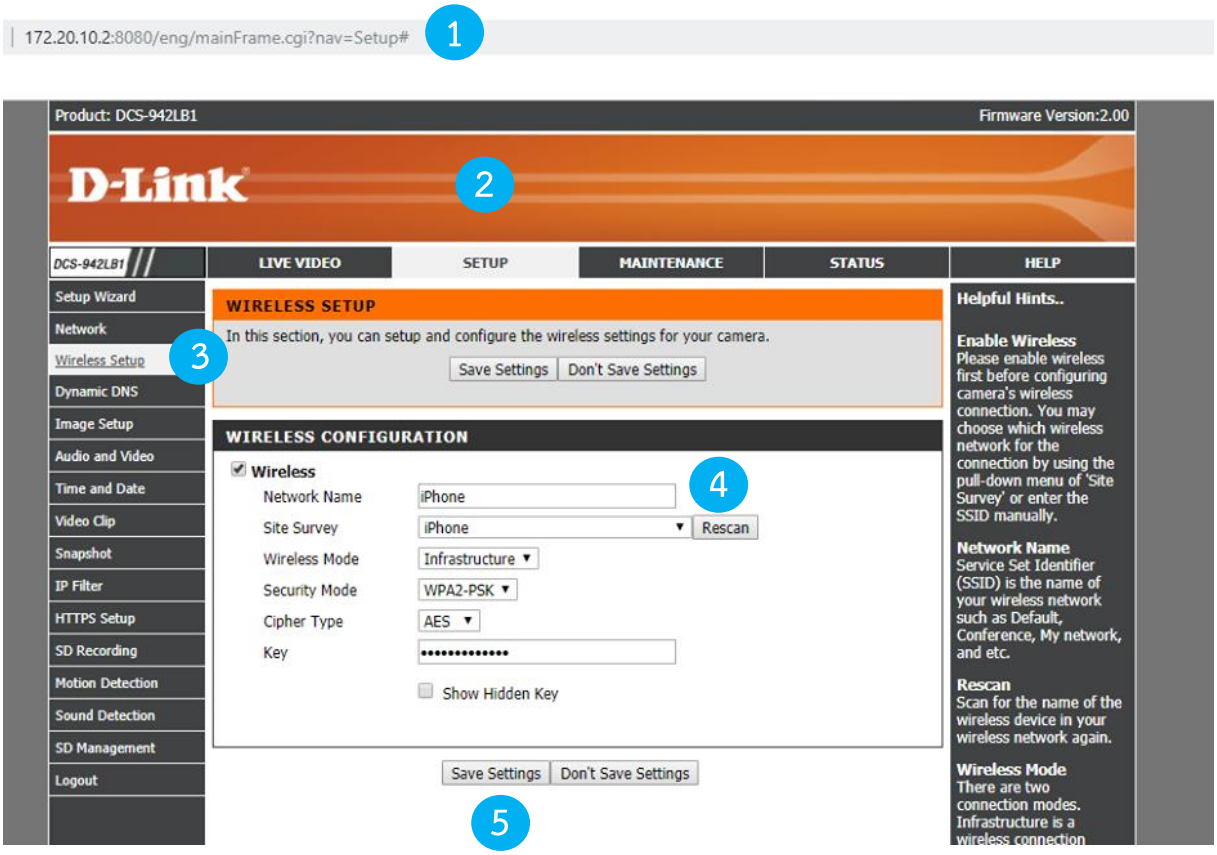
ภาพประกอบที่ 3-30 ภาพแสงน้อย

3.6 วิธีการติดตั้งกล้องไอพี

จากภาพประกอบที่ 3-31 จะแสดงให้เห็นถึงวิธีการติดตั้งก่อนนำไปใช้งานจริง โดยในขั้นแรกจะต้องนำกล้องไอพีเชื่อมต่อกับสายแลน และเชื่อมอินเตอร์เน็ตให้กับคอมพิวเตอร์โดยจะต้องอยู่ในวงแลน (Lan) เดียวกับอินเตอร์เน็ตที่ต่อกับกล้องไอพี

ภาพประกอบที่ 3-31 ตัวอย่างการติดตั้งกล้อง IP (1)

เมื่อเชื่อมต่ออินเตอร์เน็ตของกล้องไอพี และคอมพิวเตอร์แล้วให้ทำการเข้าไปที่ URL ของกล้องไอพีซึ่งเป็น IP Address ของกล้อง จากนั้นทำการตั้งค่า WIFI ให้กับกล้องเพื่อการใช้งานแบบไร้สายอินเตอร์เน็ตดังภาพประกอบที่ 3-32 และ 3-33



ภาพประกอบที่ 3-32 ตัวอย่างการติดตั้งกล้องไอพี (2)

ภาพประกอบที่ 3-33 ตัวอย่างการติดตั้งกล้องไอพี (3)

3.7 คำสั่งที่ใช้ในการรับภาพ

การรับภาพมาจากกล้องเข้ามาประมวลผลในโปรแกรมนั้นสามารถทำได้ดังภาพประกอบที่ 3-34 และมีรายละเอียดดังต่อไปนี้

1) นำ IP ของกล้องมาเพื่อประมวลผลในการรับภาพโดยใช้คำสั่ง cv2.VideoCapture ในการอ่านภาพจากกล้อง และเก็บไว้ที่ตัวแปร stream

2) ใช้คำสั่ง .read() เพื่ออ่านค่าจากตัวแปร stream มาเก็บไว้ที่ตัวแปร r และ f

3) แปลงค่าจากตัว f ให้เป็น ชนิด numpy array เพื่อที่จะสามารถแสดงภาพแบบ real-time และเก็บไว้ที่ตัวแปร color_live

4) การที่จะแสดงภาพแบบ real-time นั้นจำเป็นต้องใช้คำสั่ง cv2.imshow()

```
stream = cv2.VideoCapture('your destination')
while True:
     r, f = stream.read()
     color_live = np.array(f, dtype='uint8')
     cv2.imshow('My IP', color_live)
```

ภาพประกอบที่ 3-34 คำสั่งในการรับภาพจากกล้องไอพี

5) เมื่อทำการรันคำสั่งในการรับภาพในภาพประกอบที่ 3-34 แล้วจะได้ผลลัพธ์ดังภาพประกอบที่ 3-35

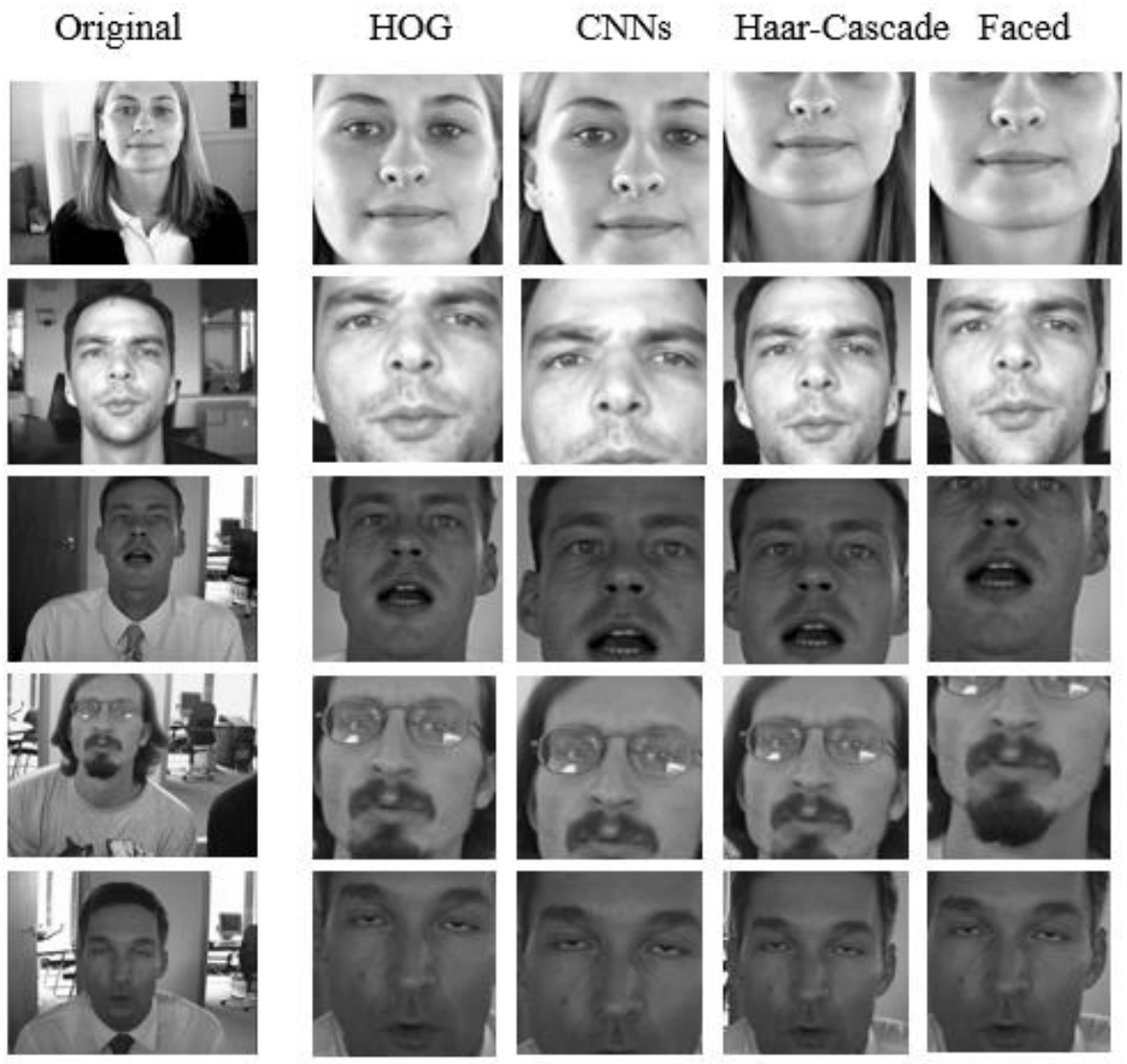
ภาพประกอบที่ 3-35 ตัวอย่างการแสดงภาพจากกล้องไอพี

บทที่ 4

ผลการทดลอง

4.1 การตรวจจับใบหน้า

จากภาพประกอบที่ 4–1 จะแสดงตัวอย่างของผลการตรวจจับใบหน้าของทั้ง 4 วิธี ได้แก่ Convolutional Neural Networks (CNNs), Histograms of Oriented Gradients (HOG), Haar-Cascade Classifier และ Faced ตามลำดับ



ภาพประกอบที่ 4-1 ตัวอย่างของผลการตรวจจับใบหน้าของทั้ง 4 วิธี

4.2 ผลการทดลองการตรวจจับใบหน้าจากชุดข้อมูล The BioID Face

จากตารางที่ 4–1 จะแสดงตารางสรุปผลการตรวจจับใบหน้าของทั้ง 4 วิธี ได้แก่ Convolutional Neural Networks (CNNs), Histograms of Oriented Gradients (HOG), Haar-Cascade Classifier และ Faced โดยในการทดลองนี้วิธี Histograms of Oriented Gradients นั้นได้รวมกับวิธี Support Vector Machine (SVM) โดยจะเป็นวิธีที่ดีที่สุดในการตรวจจับใบหน้าได้โดยไม่มีข้อผิดพลาด ประสิทธิภาพของเทคนิค HOG+SVM ที่ได้จากชุดข้อมูล BioID face คือ 99.60%

ตารางที่ 4-1 ตารางสรุปผลการตรวจจับใบหน้าของทั้ง 4 วิธี

วิธีที่ใช้ในการทดสอบ	จำนวนภาพที่ตรวจจับได้	จำนวนภาพ Error	ความถูกต้องแม่นยำ (%)
HOG+SVM	1,507	0	99.60
CNNs	1,513	40	97.36
Haar-Cascade	1,459	40	93.79
Faced	1,449	107	88.70

4.3 ผลการทดลองการตรวจสอบใบหน้าจากชุดข้อมูล The BioID Face, FERET และ ColorFERET

สำหรับเทคนิค face encoding ที่ได้นำมาทดลองประสิทธิภาพนั้น ได้แก่ VGG16, FaceNet และ ResNet-50 โดยความละเอียดของภาพที่ใช้ในการทดลองคือ 224x224 พิกเซล ในการทดสอบ VGG16 จะมีจำนวนของฟีเจอร์มากที่สุดคือ 25,088 ตามด้วย ResNet-50 และ FaceNet ดังตารางที่ 4-2

ตารางที่ 4-2 ตารางแสดงรายละเอียดของ face encoding ทั้ง 3 วิธี

พารามิเตอร์	วิธีที่ใช้ในการตรวจจับใบหน้า		
	VGG16	FaceNet	ResNet-50
ความละเอียดของภาพ	224x224	224x224	224x224
จำนวนของ feature	25,088	512	2,048

ในข้อมูลของตารางที่ 4-1 ทางทีมผู้วิจัยพบว่า HOG+SVM เป็นวิธีการตรวจจับใบหน้าที่ดีที่สุดตามการทดลองของเราในชุดข้อมูล The BioID face จากนั้นเราจึงเลือกวิธี HOG+SVM ในการตรวจจับใบหน้าจากชุดข้อมูลใบหน้า 3 ชุด ได้แก่ The BioID face, FERET และ ColorFERET เป็น

ผลให้จำนวนภาพใบหน้าตรวจพบจาก ชุดข้อมูลที่กล่าวไปข้างต้นได้ผลลัพธ์เป็น 1,507, 1,372 และ 3,553 ภาพใบหน้าตามลำดับโดยแสดงดังตารางที่ 4-3

ตารางที่ 4-3 ตารางสรุปผลค่าความถูกต้องของทั้ง 3 วิธี

ชุดข้อมูล	จำนวนภาพ	จำนวนบุคคล	ค่าความถูกต้อง (%) ของวิธีที่ใช้ในการตรวจสอบใบหน้า		
			VGG16	ResNet-50	FaceNet
The BioID Face	1,507	21	99.74±0.38	100	100
FERET	1,372	196	83.93±0.77	100	100
Color FERET	3,553	474	74.96±1.26	99.60±0.46	99.32±0.32

ในการทดลองนี้มีการตรวจสอบการแบ่งชุดข้อมูลแบบการสุ่มจำนวน 5 ครั้งเพื่อประเมินประสิทธิภาพของเทคนิค face encoding ที่แตกต่างกัน ในการทดลองของเราเทคนิคที่ดีที่สุดสำหรับ face encoding คือวิธี ResNet-50 และ FaceNet เนื่องจากวิธีของทั้ง 2 นี้ได้มีความแม่นยำ 100% สำหรับชุดข้อมูล The BioID face และ FERET แต่เมื่อทำการทดลองกับชุดข้อมูล ColorFERET ซึ่งประกอบด้วยภาพใบหน้า 3,553 ภาพที่มีอาสาสมัคร 474 คน พบว่าวิธี ResNet-50 และ FaceNet มีความแม่นยำสูงถึง 99.60% และ 99.32% ตามลำดับ

4.4 ผลการทดลองการตรวจจับใบหน้าจากกล้องไอพี

เมื่อได้ผลการทดลองวิธีการในการตรวจจับใบหน้าจากชุดข้อมูล The BioID Face แล้วทางทีมผู้วิจัยจึงได้ทำการทดลองกับชุดข้อมูลจริงซึ่งทำการรับภาพจากกล้องไอพี โดยตัวอย่างภาพที่รับเข้ามาจะแสดงดังตารางประกอบที่ 4-4 และ 4-5

ตารางที่ 4-4 ตัวอย่างการตรวจจับใบหน้าจากกล้องไอพี (1)

วิธีที่ใช้ในการทดสอบ	ภาพต้นฉบับ	ภาพใบหน้า
HOG+SVM		
		

ตารางที่ 4-5 ตัวอย่างการตรวจจับใบหน้าจากกล้องไอพี (2)

วิธีที่ใช้ในการทดสอบ	ภาพต้นฉบับ	ภาพใบหน้า
Haar-Cascade		
		

ตารางที่ 4-6 จะแสดงตารางสรุปผลการตรวจจับใบหน้าของทั้ง 2 วิธี ได้แก่ Histograms of Oriented Gradients (HOG) และ Haar Cascade Classifier โดยวิดีโอที่ถ่ายด้วยกล้องไอพี เพื่อนำมาทดสอบนั้นมีทั้งหมด 5 วิดีโอ ซึ่งแบ่งเป็นวิดีโอละ 15 นาที รวมทั้งสิ้น 75 นาทีด้วยกัน และสถานที่ที่ใช้ในการทดลองนั้นได้แก่ ห้อง IT-401 และ SC1-101

จากตารางจะแสดงให้เห็นว่าในการใช้งานกับกล้องไอพี นั้นวิธี Haar-Cascade จะมีประสิทธิภาพมากกว่าเนื่องจากคุณภาพจากกล้องไอพี นั้นมีคุณภาพค่อนข้างต่ำ และวิธี HOG+SVM นั้นจำเป็นต้องใช้ภาพที่มีความคมชัดเป็นเหตุให้ตรวจพบใบหน้าได้น้อยกว่าวิธี Haar-Cascade

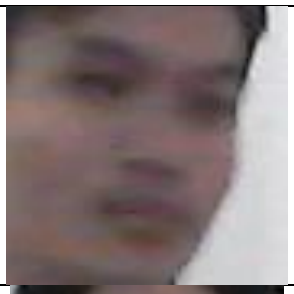
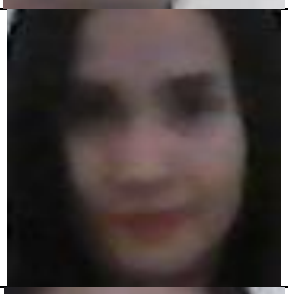
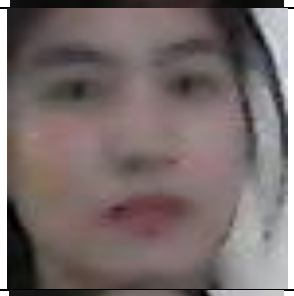
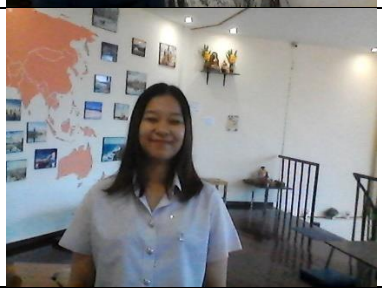
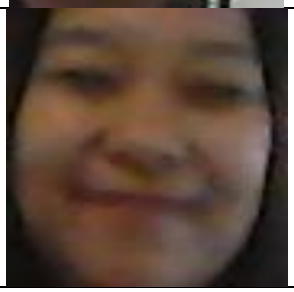
ตารางที่ 4-6 ตารางสรุปผลการตรวจจับใบหน้าของทั้ง 2 วิธี

วิธีที่ใช้ในการทดสอบ	จำนวนภาพที่ตรวจจับได้	จำนวนภาพ Error	จำนวนภาพที่ถูกต้อง
HOG+SVM	68	7	61
Haar-Cascade	591	96	495

4.5 ผลการทดลองการตรวจสอบใบหน้าจากกล้องไอพี

ในข้อมูลของตารางที่ 4-6 ทางทีมผู้วิจัยพบว่า Haar-Cascade เป็นวิธีการตรวจจับใบหน้าที่ดีที่สุดตามการทดลองจากชุดข้อมูลจากกล้อง IP ดังนั้นเราจึงเลือกวิธี Haar-Cascade ในการตรวจจับใบหน้าเพื่อนำมาทำการตรวจสอบใบหน้า โดยเราได้ทำการเลือกใบหน้าบุคคลจำนวน 5 คน โดยจะทำการแบ่งออกเป็นใบหน้าบุคคลจำนวนละ 16, 20, 11, 12 และ 26 ตามลำดับ ตัวอย่างภาพของแต่ละใบหน้าบุคคลจะแสดงดังตารางที่ 4-7 และ 4-8

ตารางที่ 4-7 ตัวอย่างการตรวจจับใบหน้าจากกล้องไอพี (3)

ใบหน้าบุคคล	ภาพต้นฉบับ	ภาพใบหน้า
01		
02		
03		
04		
05		

จากจากตางที่ 4-8 นี้ทำการการทดลองโดยใช้ข้อมูลจากตารางที่ 4-6 มาทำการการตรวจสอบการแบ่งชุดข้อมูลแบบการสุ่มจำนวน 5 ครั้งเพื่อประเมินประสิทธิภาพของเทคนิค face encoding ที่แตกต่างกัน ในการทดลองของเราเทคนิคที่ดีที่สุดสำหรับ face encoding คือวิธี ResNet-50 เนื่องจากมีความแม่นยำถึง 97.65%

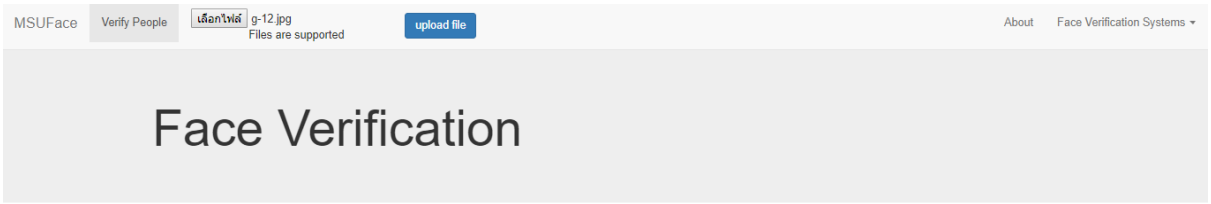
ตารางที่ 4-8 ตารางสรุปผลค่าความถูกต้องจากชุดข้อมูลที่รับจาก IP Camera ของทั้ง 3 เทคนิค

ชุดข้อมูล	จำนวนภาพ	จำนวนบุคคล	ค่าความถูกต้อง (%) ของวิธีที่ใช้ในการตรวจสอบใบหน้า		
			VGG16	FaceNet	ResNet-50
IP Camera	85	5	91.76±4.71	96.47±2.88	97.65±2.88

4.4 ตัวอย่างการทำงานของ Web Application

ในส่วนของ Web application นั้นใช้สำหรับตรวจสอบใบหน้าโดยมีทั้งหมด 3 ขั้นตอนดังต่อไปนี้

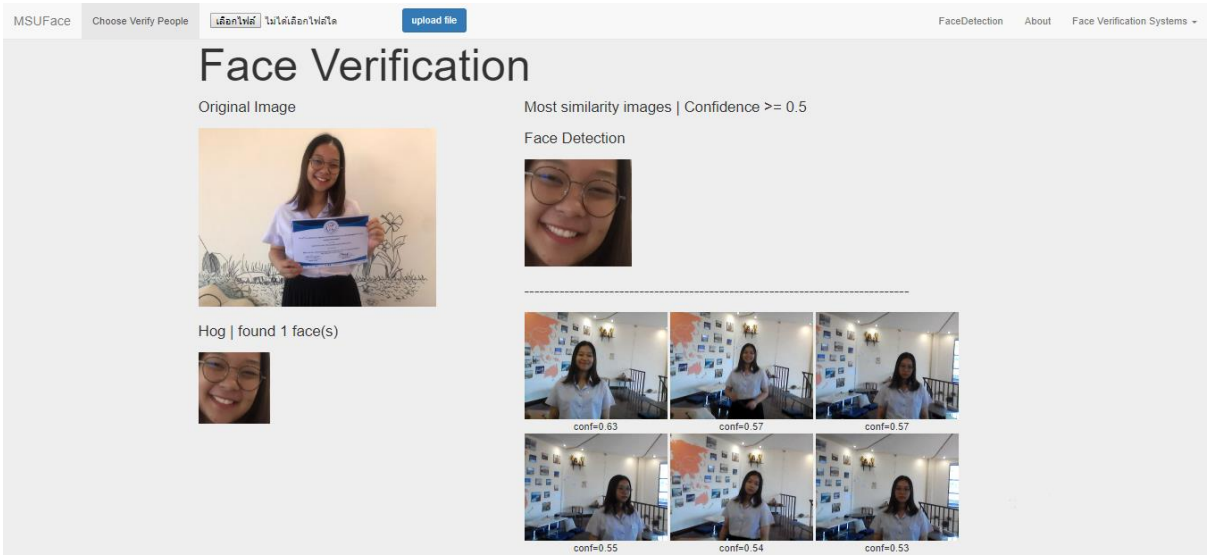
4.4.1 จากภาพประกอบที่ 4-2 จะแสดงให้เห็นถึงส่วนของหน้าหลัก Web Application ซึ่งจะเป็นส่วนของการอัพโหลดรูปภาพที่ผู้ใช้งานต้องการตรวจสอบใบหน้า



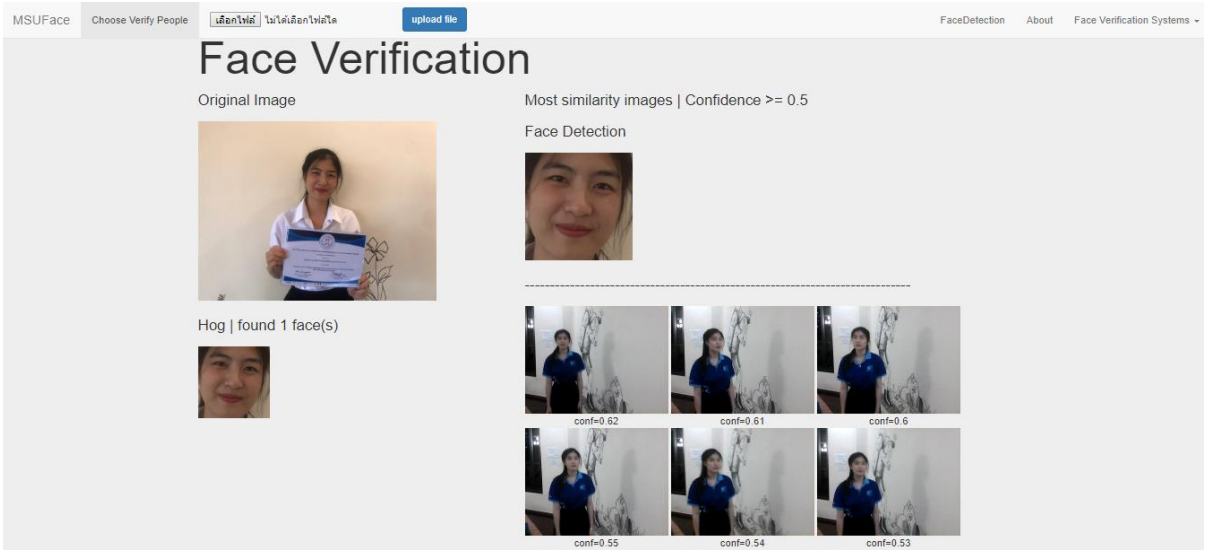
ภาพประกอบที่ 4-2 ตัวอย่างหน้าหลักของ Web Application

4.4.2 ระบบจะนำรูปภาพที่อัพโหลดเข้ามาทำการตรวจจับใบหน้า (Face Detection) และแสดงผลลัพธ์ใบหน้าที่ตรวจจับได้ในฝั่งซ้ายมือของหน้าเว็บไซต์ดังภาพประกอบที่ 4-3

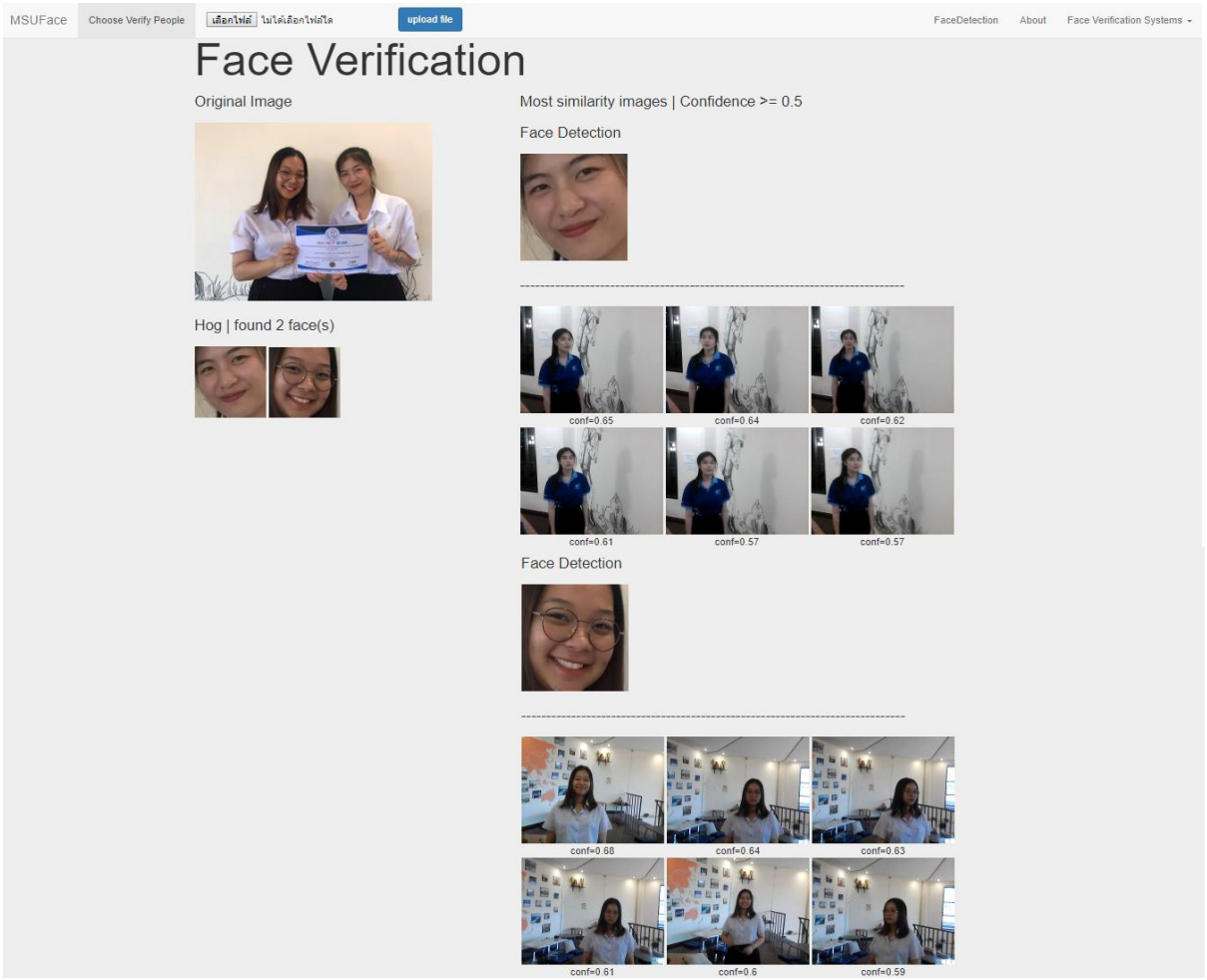
4.4.3 จากนั้นระบบจะนำใบหน้าที่ตรวจจับได้มาเปรียบเทียบความล้ายคลึงของใบหน้ากับรูปภาพที่มีอยู่ในระบบ และจะแสดงรูปภาพที่มีความคล้ายคลึงกัน โดยในตัวอย่างนี้กำหนดค่าความเชื่อมั่นเท่ากับ 0.5 และจัดอันดับค่าความเชื่อมั่นจากมากไปหาน้อย และตัวอย่างการแสดงผลลัพธ์จะแสดงดังภาพประกอบที่ 4-3 ถึง ภาพประกอบที่ 4-5



ภาพประกอบที่ 4-3 ตัวอย่างหน้าแสดงผลลัพธ์ของการตรวจสอบใบหน้าที่ 1



ภาพประกอบที่ 4-4 ตัวอย่างหน้าแสดงผลลัพธ์ของการตรวจสอบใบหน้าที่ 2



ภาพประกอบที่ 4-5 ตัวอย่างหน้าแสดงผลลัพธ์ของการตรวจสอบใบหน้า 2 ใบหน้า

บทที่ 5
สรุปและอภิปรายผล

งานวิจัยครั้งนี้มีวัตถุประสงค์เพื่อการพัฒนาระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพี โดยระบบนั้นสามารถตรวจจับใบหน้าบุคคลจากกล้องไอพี และสามารถตรวจสอบใบหน้าบุคคลได้ ซึ่งระบบที่ทีมผู้วิจัยได้พัฒนาขึ้นมานั้นสามารถใช้เป็นเครื่องมือในการเฝ้าระวังความปลอดภัยภายในอาคารบ้านเรือนได้ ทั้งนี้คณะผู้วิจัยได้นำรูปภาพที่เก็บบันทึกไว้จากระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพีไปทำการประมวลผลเพื่อค้นหาบุคคล โดยอาศัยหลักการของการตรวจจับใบหน้า (Face Detection) และนำใบหน้าไปเปรียบเทียบกับภาพใบหน้าที่ต้องการค้นหา (Query Image) เพื่อตรวจสอบ (Face Verification) หาค่าความคล้ายคลึง (Similarity) และแสดงผลลัพธ์ทางเว็บบราวเซอร์ โดยการพัฒนานี้ทีมผู้วิจัยได้ทำการทดลองกับชุดข้อมูลที่เป็นภาพนิ่ง และชุดข้อมูลที่ได้จากกล้องไอพี ซึ่งสามารถสรุปผลงานวิจัย และอภิปรายผลของการดำเนินการพัฒนาได้ดังต่อไปนี้

5.1 สรุปผลงานวิจัย

ระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพีนั้นประกอบไปด้วย 2 ส่วนหลัก ได้แก่ การตรวจจับใบหน้า และการตรวจสอบใบหน้า เพื่อให้ได้ผลลัพธ์ที่ถูกต้องแม่นยำมากที่สุดนั้นทางทีมผู้วิจัยจึงได้ทำการทดลองเพื่อหาวิธีการตรวจจับ และตรวจสอบใบหน้าที่ดีที่สุด สำหรับการทดลองการตรวจจับใบหน้านั้นเราได้เลือกใช้ 4 วิธีการ ได้แก่ HOG+SVM, CNN, Faced และ Haar-Cascade โดยในขั้นแรกนำมาทดลองกับชุดข้อมูล The BioID Face และผลลัพธ์จากการทดลองแสดงให้เห็นว่าวิธีการ HOG+SVM เป็นวิธีที่มีประสิทธิภาพมากที่สุด จากนั้นทางทีมผู้วิจัยจึงได้ทำการทดลองการตรวจสอบใบหน้าด้วย 3 วิธีการ ได้แก่ ResNet-50, FaceNet และ VGG16 โดยทดลองกับ 3 ชุดข้อมูล ได้แก่ The BioID face, FERET และ Color FERET สำหรับวิธีการที่เลือกใช้ในการตรวจจับใบหน้าก่อนที่จะนำไปตรวจสอบใบหน้านั้นทีมผู้วิจัยได้เลือกใช้วิธีการ HOG+SVM และผลลัพธ์จากการทดลองการตรวจสอบใบหน้านั้นแสดงให้เห็นว่าวิธีการ ResNet-50 เมื่อทดลองกับชุดข้อมูล Color FERET มีความแม่นยำ 99.60% และเมื่อได้ทดลองกับชุดข้อมูล The BioID face และ FERET พบว่ามีความแม่นยำมากถึง 100%

จากการทดลองกับชุดข้อมูล The BioID face, FERET และ Color FERET พบว่าการตรวจจับใบหน้าที่มีประสิทธิภาพมากที่สุดคือ HOG+SVM และการตรวจสอบใบหน้าที่มีประสิทธิภาพมากที่สุดคือ ResNet-50 ทางทีมผู้วิจัยจึงได้นำวิธีการดังกล่าวมาทดลองใช้กับชุดข้อมูลภาพที่รับมาจากกล้องไอพีทำให้พบว่าในการตรวจจับใบหน้าโดยวิธี HOG+SVM ได้ผลลัพธ์ที่มีประสิทธิภาพต่ำ

เนื่องจากเป็นวิธีที่ต้องการภาพที่มีความละเอียดสูงแต่ภาพที่รับมาจากกล้องไอพีนั้นมีคุณภาพ และความละเอียดต่ำเป็นผลให้สามารถตรวจจับใบหน้าได้เพียง 68 ภาพ แต่ในขณะเดียวกันวิธี Haar-Cascade สามารถตรวจจับใบหน้าได้ถึง 591 ภาพ จากผลลัพธ์จากการทดลองแสดงให้เห็นว่าวิธี Haar-Cascade เป็นวิธีที่เหมาะสมที่สุดในการตรวจจับใบหน้าเมื่อใช้งานกับกล้องไอพี จากนั้นทีมผู้วิจัยได้เลือกใบหน้าบุคคลที่ตรวจจับได้โดยวิธี Haar-Cascade จำนวน 5 คนมาทดลองการตรวจสอบใบหน้า และจากผลการทดลองนั้นจะแสดงให้เห็นว่าวิธีการที่มีประสิทธิภาพมากที่สุดยังคงเป็น ResNet-50 ซึ่งมีความแม่นยำมากถึง 97.65%

ดังนั้นในการพัฒนาระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพีทางทีมผู้วิจัยจึงได้เลือกใช้วิธี Haar-Cascade ในการตรวจจับใบหน้า และเลือกใช้วิธี ResNet-50 ในการตรวจสอบใบหน้า

5.2 ข้อเสนอแนะ

5.2.1 ข้อเสนอแนะจากการวิจัย

1) สำหรับการทดลองวิธีการในการตรวจจับใบหน้า และตรวจสอบใบหน้านั้นจำเป็นต้องใช้เครื่องมือในการประมวลผลที่มีประสิทธิภาพสูง เพื่อให้ได้ผลลัพธ์ที่ถูกต้องและรวดเร็วมากยิ่งขึ้น

2) ในส่วนของการรับภาพจากกล้องไอพีนั้นควรเลือกกล้องที่ให้คุณภาพของภาพสูงเพื่อความแม่นยำในการตรวจจับและตรวจสอบใบหน้า

5.2.2 ข้อเสนอแนะสำหรับการวิจัยครั้งต่อไป

1) ระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพีในขณะนี้ในส่วนของการตรวจสอบใบหน้าไม่สามารถทำงานพร้อมกันกับการตรวจจับใบหน้าได้ สำหรับการทดลองและพัฒนาในครั้งต่อไปควรทำให้เป็นระบบที่สามารถตรวจจับและตรวจสอบใบหน้าได้แบบ Real-time

2) ควรทำการแจ้งเตือน (Notification) ไปยังผู้ใช้งานเมื่อพบบุคคลไม่ต้องประสงค์

เอกสารอ้างอิง

[1] VOCAL Technologies, “Histogram of Oriented Gradients (HOG) for Object Detection,” [ออนไลน์]. Available: https://www.vocal.com/video/histogram-of-oriented-gradients-hog-for-object-detection/. [วันที่เข้าถึง15 มีนาคม 2562].

[2] K. He, X. Zhang, S. Ren และ J. Sun, “Deep Residual Learning for Image Recognition,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 770-778, 2559.

[3] วีพี อินเทลลิเจนท์ ดีไวซ์ จำกัด, “vpintelligent,” VP Intelligent Devices Co, 2554. [ออนไลน์]. Available: http://www.vpintelligent.com/A7. [วันที่เข้าถึง5 กุมภาพันธ์ 2562].

[4] ซีเวิลด์ ซัพพลาย แอนด์ เทรดดิ้ง จำกัด, “seaworldcctvsecurity,” ซีเวิลด์ ซัพพลาย แอนด์ เทรดดิ้ง จำกัด, 2560. [ออนไลน์]. Available: http://www.seaworldcctvsecurity.com/94-analog-9A-ip/. [วันที่เข้าถึง 5 กุมภาพันธ์ 2562].

[5] บริษัท มีเดีย เสิร์ช จำกัด, “cctvbangkok,” MEDIA SEARCH CO.,LTD., 12 มิถุนายน 2556. [ออนไลน์]. Available: http://www.cctvbangkok.com/article/41_. [วันที่เข้าถึง5 กุมภาพันธ์ 2562].

[6] เอเอสดี ดิสทริบิวชั่น จำกัด, “asd,” เอเอสดี ดิสทริบิวชั่น จำกัด, 24 มกราคม 2562. [ออนไลน์]. Available: http://asd.co.th/-face-detection-camera-/. [วันที่เข้าถึง5 กุมภาพันธ์ 2562].

[7] aosoft.co.th, “aosoft,” 23 กรกฎาคม 2561. [ออนไลน์]. Available: https://www.aosoft.co.th/article/322/Python-A3.html. [วันที่เข้าถึง7 กุมภาพันธ์ 2562].

[8] N. Chuntra, “OpenCV คืออะไร?,” 14 ธันวาคม 2561. [ออนไลน์]. Available: https://medium.com/@nut.ch40/opencv-A3-8771e2a4c414. [วันที่เข้าถึง7 กุมภาพันธ์ 2562].

[9] INRIA, [ออนไลน์]. Available: https://scikit-learn.org/stable/. [วันที่เข้าถึง7 กุมภาพันธ์ 2562].

[10] V. Minaphinant, “Machine Learning คืออะไร?,” 28 กุมภาพันธ์ 2561. [ออนไลน์]. Available: https://blog.finnomena.com/machine-learning-A3-fa8bf6663c07. [วันที่เข้าถึง7 กุมภาพันธ์ 2562].

[11] P. Tongpradit, “มาทำความรู้จัก Tensorflow,” 16 ธันวาคม 2561. [ออนไลน์]. Available: https://www.thaiprogrammer.org/2018/12/มาทำความรู้จัก-tensorflow/. [วันที่เข้าถึง7 กุมภาพันธ์ 2562].

[12] P. Wannaphong, “พัฒนา Machine learning ด้วย TensorFlow,” 30 มกราคม 2559. [ออนไลน์]. Available: https://python3.wannaphong.com/2016/01/machine-learning-tensorflow.html. [วันที่เข้าถึง7 กุมภาพันธ์ 2562].

[13] T. Wonghong, “keras คืออะไร,” กันยายน 28 2561. [ออนไลน์]. Available: http://drtanet.blogspot.com/2018/09/keras.html. [วันที่เข้าถึง8 กุมภาพันธ์ 2562].

[14] A. J. S. Gabbualoy, T. Rueangcharat, S. Chiewchanwattana และ K. Sunat, “การตรวจจับภาพใบหน้าด้วยเทคนิควิธีพิจารณาพื้นที่สีผิวร่วมกับตรวจสอบองค์ประกอบบนใบหน้า Face Detection using Hybrid detection Characteristic facial with Skin color based on Viola-Jones,” *The Twelfth National Conference on Computing and Information Technology,* pp. 575-580, 2559.

[15] R. Sutthaweekul และ W. Lee, “การตรวจจับใบหน้าด้วยวิธีการพื้นฐานของการจำลองรูปแบบ Haar-like,” *SWU Engineering Journal,* เล่มที่ 6, %12, pp. 34-43, 2 ธันวาคม 2554.

[16] J. Benjaparkairat, “cekmitl,” [ออนไลน์]. Available: http://www.ce.kmitl.ac.th/project.php?action=view&PJ_ID=308. [วันที่เข้าถึง9 กุมภาพันธ์ 2562].

[17] O. Jesorsky, K. J. Kirchberg และ R. W. Frischholz, “Robust Face Detection Using the Hausdorff Distance,” *Lecture Notes in Computer Science book series (LNCS),* p. 90–95, 2554.

[18] Y. Taigman, M. Yang, M. Ranzato และ L. Wolf, “DeepFace: Closing the gap to human-level performance in face verification,” *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 1701-1708, 2557.

[19] F. Schroff, D. Kalenichenko และ J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 815-823, 2558.

[20] H. Shu, D. Chen, Y. Li และ S. Wang, “A highly accurate facial region network for unconstrained face detection,” *IEEE International Conference on Image Processing (ICIP),* pp. 665-669, 2560.

[21] F. Schroff, D. Kalenichenko และ J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR),* 2558.

[22] X. Zhang, ,. J. Zou, K. He และ J. Sun, “Accelerating Very Deep Convolutional Networks for Classification and Detection,” *The IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 1-14, 2559.

[23] softmelt, “softmelt,” Softmelt Co.,Ltd, 2554. [ออนไลน์]. Available: http://www.eduthaieasyelec.com/16623242/. [วันที่เข้าถึง5 กุมภาพันธ์ 2562].

[24] mindphp, “mindphp,” 2560 มีนาคา 2560. [ออนไลน์]. Available: https://www.mindphp.com/68-php-e-commerce/2055-database. [วันที่เข้าถึง9 กุมภาพันธ์ 2562].

[25] mindphp.com, “JavaScript คืออะไร,” mindphp.com, 14 มีนาคม 2560. [ออนไลน์]. Available: https://www.mindphp.com/73-A3/2187-java-javascript-A3.html. [วันที่เข้าถึง9 กุมภาพันธ์ 2562].

[26] mindphp.com, "SQL คืออะไร," .mindphp.com, 14 มีนาคม 2560. [Online]. Available: https://www.mindphp.com/73-A3.html. [วันที่เข้าถึง9 กุมภาพันธ์ 2562].

[27] N. Rakthong, “Supervised Vs Unsupervised Learning,” 29 พฤษภาคม 2561. [ออนไลน์]. Available: https://medium.com/@nattaponra/ml2-supervised-vs-unsupervised-learning-A7-aae9aa6f142b. [วันที่เข้าถึง6 กุมภาพันธ์ 2562].

[28] K. Preechakul, “Reinforcement Learning,” 23 สิงหาคม 2558. [ออนไลน์]. Available: https://medium.com/o-v-e-r-f-i-t-t-e-d-/b6a9a1167820. [วันที่เข้าถึง7 กุมภาพันธ์ 2562].

[29] P. Pornchaloempong และ N. Nunak, “Accuracy / ความถูกต้อง ความแม่นยำ,” [ออนไลน์]. Available: http://www.foodnetworksolution.com/wiki/word/4289/accuracy-?fbclid=IwAR0aCOHDwXV_8kkJSBfwET4snhRZpKE3_zcIc9zKK4Uzcnk8Z1_7DoijYbo%87-%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A. [วันที่เข้าถึง5 เมษายน 2562].

[30] N. Dalal และ B. Triggs, “Histograms of Oriented Gradients for Human Detection,” *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* 2548.

ภาคผนวก

ภาคผนวก ก

งานวิจัยเพิ่มเติม

ภาคผนวก ก

งานวิจัยเพิ่มเติม

The 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP 2019) October 30 - November 1, 2019, Chiang Mai, Thailand

Effective Face Verification Systems Based on the Histogram of Oriented Gradients and Deep Learning Techniques

Sawitree Khunthi
Department of Information Technology
Faculty of Informatics
Mahasarakham University
Maha Sarakham, Thailand
sawitri0212@gmail.com

Pichada Saichua
Department of Information Technology
Faculty of Informatics
Mahasarakham University
Maha Sarakham, Thailand
pichadakt@gmail.com

Olarik Surinta
Multi-agent Intelligent Simulation Laboratory (MISL), Department of Information Technology, Faculty of Informatics, Mahasarakham University
Maha Sarakham, Thailand
olarik.s@msu.ac.th

***Abstract*—**In this paper, we proposed a face verification method. We experiment with a histogram of oriented gradients description combined with the linear support vector machine (HOG+SVM) as for the face detection. Subsequently, we applied a deep learning method called ResNet-50 architecture in face verification. We evaluate the performance of the face verification system on three well-known face datasets (BioID, FERET, and ColorFERET). The experimental results are divided into two parts; face detection and face verification. First, the result shows that the HOG+SVM performs very well on the face detection part and without errors being detected. Second, The ResNet-50 and FaceNet architectures perform best and obtain 100% accuracy on the BioID and FERET dataset. They also, achieved very high accuracy on ColorFERET dataset.

Keywords— face verification systems, face detection, face verification, ResNet-50, FaceNet

I. Introduction

Face verification is part of the face recognition system that focuses on the one-to-one matching problem [2] to compare whether it is the same person or not the same person. For this reason, face verification is much used in security, surveillance, and immigration, for example, to search for people from closed circuit television (CCTV) or to check if the person is a criminal by comparison of a face captured on camera with faces from a database. Many problems, such as images, low-light images, blurred image, and flare on an image resulting from stray light entering the camera lens, will occur depending on the quality and location of the camera. These effects are of concern for the researchers working on face recognition.

Face verification systems perform two main tasks. The first task is face detection and is essential to any face verification system because the system cannot process if the face is not detected. Many researchers focus on developing algorithms for face detection such as edge detection [3], Haar-cascade classifier [3][4], and histogram of oriented gradients (HOG) [5–7]. These algorithms allow us to find faces even in low-light and blurred images. Moreover, convolutional neural networks (CNNs) that have been proposed [8][9] provide a robust method to detect a face in many conditions such as a small faces, occlusion, or images that do not show the entire face.

The second task of face verification, is the extraction of information from the face (called face encoding) which is sent to the similarity function to calculate and compare the unknown face and detected face. A high similarity value shows that the two faces are the most similar face. Many algorithms have been proposed for the face encoding such as local directional number pattern [11], local binary patterns [12], common encoding feature discriminant [13] and supervised feature encoding [14] are proposed. Nowadays, deep learning approaches are successful in encoding the face, including VGGNet [15], DeepFace [1], FaceNet [16] and ResNet [17].

Contribution: In this paper, we evaluate the performance of face verification systems on three well-known face datasets (BioID, FERET, and ColorFERET). It is quite challenging to verify faces from the ColorFERET because this dataset consists of 3,553 face images of 474 subjects. We divided the experiment into two parts; face detection and face

verification. In the face detection part, four different face techniques, including the histogram of oriented gradients combined with the linear support vector machine (HOG+SVM), max-margin object detection with convolutional neural network (MMOD-CNN) [18][19], Haar-Cascade Classifier [20][21] and Faced techniques were evaluated on the BioID dataset. The experiments showed that the HOG+SVM performs very well and without errors of face detection. Moreover, in the face verification part, three robust deep CNN architectures called VGG16, FaceNet, and ResNet-50 architectures were used as the face encoding. The experimental results showed that the ResNet-50 and FaceNet performed best and obtained 100% accuracy on the BioID and FERET dataset. Additionally, both architectures achieved very high accuracy on the ColorFERET dataset.

Paper outline: This paper is organized as follows: In Section II, the face verification systems are described in detail. In Section III, three well-known face image datasets are explained. The experimental results of face detection and verification are presented in Section IV. The last section is the conclusion and suggestions for future work.

II. Face Verification Systems

In the following, we describe the face verification systems used in the experiments; the histogram of oriented gradients and linear support vector machine aimed for face detection. Two face encoding methods; FaceNet and ResNet-50, are computed.

A. Face Detection

For face detection, the Viola-Jones face detector [20][21] is a well-known method that was first proposed for object and then for pedestrian detection. Nowadays, this technique, called Haar-cascade classifier, has become a standard technique for face detection. The Viola-Jones face detector computes feature vector based on the Haar feature. It calculates from the rectangle detector or sub-window. The detector scans through the image. Then, the set of the feature vector is given to the AdaBoost classifier, which is the weak classifier. This approach can process in real-time and get high precision. However, this approach performs not very well on the BioID dataset.

We proposed to use the histogram of oriented gradients and the linear support vector machine, called HOG+SVM, in face detection experiments.

First, the well-known HOG [22] is proposed to compute a feature vector from sub-images that scans over the whole image. With this method, the oriented gradients are computed using a gradient detector. Then the oriented gradients of each sub-image are weight to the orientation bins and used as a feature vector [23]. The gradient detector is calculated as follows:

$$G_x = I(x+1, y) - I(x-1, y) \quad (1)$$

$$G_y = I(x, y+1) - I(x, y-1) \quad (2)$$

where G_x is the horizontal and G_y is the vertical components of the gradients.

The gradient magnitude (M) and the oriented gradients (θ) are computed as:

$$M(x, y) = \sqrt{\left(G_x^2 + G_y^2\right)} \quad (3)$$

$$\theta(x, y) = tan^{-1} \frac{G_y}{G_x} d \quad (4)$$

where $M(x, y)$ is the gradient magnitude and $\theta(x, y)$ is the orientation of the gradients at the location (x, y).

Consequently, orientation bins are selected based on oriented gradients. The gradient magnitudes for each oriented gradient are weight and summed up to each orientation bin. Then, the orientation bins for each sub-image are normalized using the L2 normalization.

Second, the support vector machine (SVM) [24] algorithm with a linear kernel is proposed in this paper due to the two-class classification. With the SVM algorithm, the

hyperplane, which is the maximum distance to the training points, is used to separate training data. The training points that are closest to the calculated separating hyperplane are called support vectors. So, the best hyperplane is the distance between the closest data points of both classes and the hyperplane [25]. The optimal hyperplane is calculated as;

$$g(x) = W^T X + b \quad (5)$$

where W is the weight vector and b is the bias. The decision rule is

$$y = \begin{cases} 1 \; if \; g(x) > 0 \\ 0 \; otherwise \end{cases} \quad (6)$$

B. Face Encoding

In this research, two deep learning architectures for face encoding; ResNet-50 and FaceNet are proposed as the face encoding.

1) ResNet-50

The residual network architecture, which is a very deep network, was invented by He et al. [27], called ResNet architecture. The deep residual network creates simple stack layers, therefore the network can be set up as 18, 34, 50, 101, and 152-layer. This architecture is quite different from the original convolutional neural network (CNN) that each layer feedforward to the next layer. A deep residual learning block is implemented in the ResNet architecture (see Fig. 1). Hence, each layer allows to feed the output to feed into the next layer and directly into the next 2-3 forward blocks. This architecture known as shortcut connections.

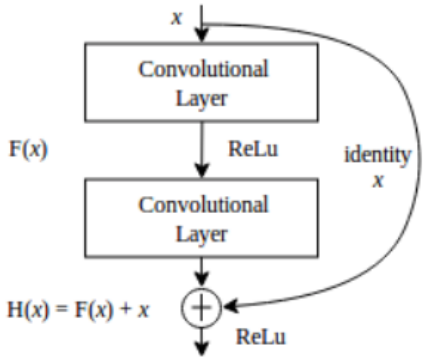


Fig. *1*. The residual network [26].

In this paper, we applied ResNet architecture with 50 layers for the face encoding (called ResNet-50). The lower-level features, which are more specific to the training data, are extracted from the face image. To encode a feature vector; we applied the flatten after the average pooling layer, which is the last layer of the ResNet-50. This architecture encodes 2,048 features and uses them as a feature vector.

1) FaceNet

FaceNet architecture was invented by Schroff et al. [16] to solve the problem of face recognition and clustering. This architecture is invariant to illumination and pose. Firstly, in this technique, the deep CNN architecture, which is inspired by Inception network, is used as a black box. The size of the parameters in FaceNet architecture is 7.5M. The small mini-batch size of around 40 faces per identity (in total, around 1,800 examples) are fed to the deep CNN. These direct to increase convergence while optimizing the network with Stochastic Gradient Descent (SGD).

Secondly, the output from the deep CNN architecture is normalized using L2 normalization and sent to the face embedding process. The embedding process is embeds in a face image into a dimensional space using the Euclidean function. This method guarantees the identity that the face image of person A is closer to other face images of the person A than closer to other face images of other persons.

Finally, the triplet selection is the last process of FaceNet. This process is given the face image of person A

to compare other face images from the mini-batch to avoid poor training. From this process, two parameters are selected, argmax and argmin, which are the hardest positive image of the same person and the hardest negative image of a different person, are selected.

In this paper, we applied FaceNet architecture using Inception network as the core network. This architecture encodes 512 features and used as a feature vector.

III. Face Image Datasets

Many face image datasets were invented for face verification systems. In this paper, we select three face image datasets; the BioID, FERET, and ColorFERET dataset for evaluating the face detection and face verification.

A. BioID Face Dataset

The BioID face dataset used in the face detection experiment includes 1,513 frontal view images [27]. In this dataset, the image resolution is 384x286 pixels and stored on the grey level. Additionally, the number of people (subject) used in the face verification experiment is 21 subjects from 1,507 face images. The BioID dataset is shown in Fig. 2(a).

B. FERET and ColorFERET Datasets

The face recognition technology (FERET) dataset and ColorFERET were published in 1993 by J. Phillips and P. Rauss [15-16]. These datasets consist of 1,199 subjects, and the total number of the face images is 14,126 images with an image resolution of 384x256 pixels. In our experiments, we have used the FERET and ColorFERET for face verification. As for the FERET dataset. We selected 1,372 images from 196 subjects from the FERET dataset (See Fig. 2(b)). and 3,553 images from 474 subjects from the ColorFERET dataset (Fig. 2(c)).

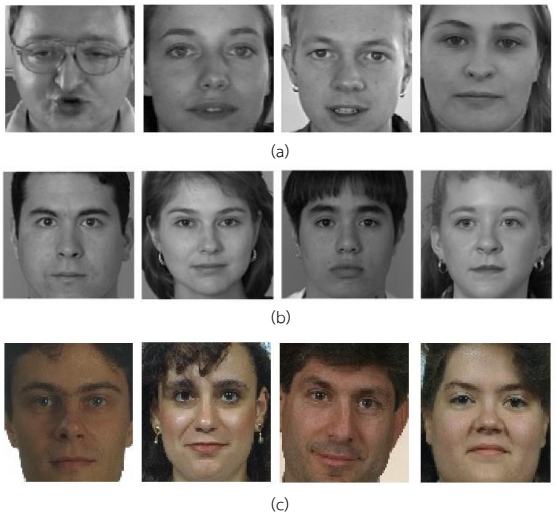


Fig. 2. Sample of face images in the (a) BioID, (b) FERET, and (c) ColorFERET datasets

IV. Experimental Results

A. Evaluation Methods

We have used two methods to evaluate the face verification system. The first evaluation method is face detection accuracy which is given by:

$$Accuracy = Acc - Err \quad (1)$$

where

$$Acc = \frac{c*100}{N} \quad (2)$$

$$Err = \frac{e*100}{N} \quad (3)$$

where c is the number of the face images after applying face detection method, and e is the number of the error face images N is the total number of the face images of the face dataset.

The second method is the accuracy of face verification.

1) We used the cosine similarity function to compare a feature vector extracted from the face image. The most similarity face is given the highest value. Then the correct prediction is that if the label of the highest value is the same as the test image. The cosine similarity function is computed as follows:

$$cos(\theta) = \frac{A.B}{\|A\|\|B\|} \quad (4)$$

where $A.B$ is the dot product of feature vector A and B.

2) To calculate the accuracy, the total number of correct predictions is multiplied by 100 and then divided by the total number of faces in the dataset.

B. Results

In this section, we show the experimental results of face detection techniques and face verification accuracies of CNN face encoding architectures.

1) Face Detection Results

To illustrate the results of face detection, Fig. 3(a) shows face images cropped so as to leave the entire face visible and Fig. 3(b) shows error due to poor cropping that results in the face being only partly visible. In this paper, when calculating the accuracy of the face detection method, we carefully reject the error face images by calculating the error (Err), as shown in Equation 3.

Table I show the experimental results of four different face detection techniques; HOG+SVM, MMOD-CNN, Haar-Cascade, and Faced techniques. Here, the histogram of oriented gradient combined with the linear support vector machine (HOG+SVM) is the only one face detection method that detects face without any error. The performance of HOG+SVM technique obtained on the BioID face dataset is 99.60%. The accuracy obtained from all face detection techniques was over 90%, except for the Faced technique. The face detection results are shown in Fig. 4.

Fig. 3. Sample results of the face images after applying face detection method. (a) entire faces and (b) error faces.

2) Face Verification Results

For the face encoding techniques, we evaluated the performance of three deep convolutional neural networks, including VGG16, FaceNet, and ResNet-50. The image resolution used in the experiments was 224x224 pixels. In the experiments, the VGG16 extracts the highest feature dimension with 25,088 features, followed by ResNet-50 and FaceNet architectures. The image resolution and size of the feature vector are shown in Table II.

(a) (b) (c) (d)

Fig. 4. Face detection results after applying face detection techniques. (a) BioID images, (b) HOG+SVM, (c) MMOD-CNN, (d) Haar-Cascade, and (e) Faced techniques.

TABLE I. PERFORMANCE OF FACE DETECTION TECHNIQUES ON BIOID DATASET

Methods	Number of face detected	Number of error detected	Accuracy (%)
HOG+SVM	1,507	0	**99.60**
MMOD-CNN	1,513	40	97.36
Haar-Cascade	1,459	40	93.79
Faced	1,449	107	88.70

2) Face Verification Results

For the face encoding techniques, we evaluated the performance of three deep convolutional neural networks, including VGG16, FaceNet, and ResNet-50. The image resolution used in the experiments was 224x224 pixels. In the experiments, the VGG16 extracts the highest feature dimension with 25,088 features, followed by ResNet-50 and

FaceNet architectures. The image resolution and size of the feature vector are shown in Table II.

In this paper we found that HOG+SVM was the best face detection method based on our experiments on the BioID dataset. We then chose the HOG+SVM method for detecting faces from three face datasets; BioID, FERET, and ColorFERET. As a result, the number of face images detects from the BioID, FERET, and ColorFERET were 1,507, 1,372, 3,553 face images, respectively. This was quite challenging because of the number of subjects in the ColorFERET (474 subjects) was 20 times higher than in the BioID dataset (only 21 subjects). The number of face images and the number of subjects are shown in Table III.

TABLE II. THE RESOLUTION OF FACE IMAGES REQUIRES FOR CNN METHODS AND THE NUMBER OF FEATURES EXTRACTS FROM THREE CNN FACE ENCODING TECHNIQUES

Parameters	Method		
	VGG16	FaceNet	ResNet-50
Image resolution	224x224	224x224	224x224
Feature vector	25,088	512	2,048

TABLE III. FACE VERIFICATION ACCURACIES (%) AND STANDARD DEVIATIONS OF THREE CNN FEATURE EXTRACTION METHODS. THE EXPERIMENTAL RESULTS ARE COMPUTED USING THREE FACE DATASETS

Dataset	Number of image	Number of subjects	Accuracy (%)		
			VGG16	FaceNet	ResNet-50
BioID	1,507	21	99.74 ± 0.38	100	100

FERET	1,372	196	83.93±0.77	100	100
Color FERET	3,553	474	74.96±1.26	99.32±0.32	99.60±0.46

In this paper, five random fold cross-validations are applied to evaluate the performance of the different face encoding methods. In our experiments, the best deep convolutional neural network (CNN) architecture for face encoding was ResNet-50 and FaceNet architectures because these two architectures obtain an accuracy of 100% on BioID and FERET face datasets. We particularly note that ResNet-50 outperforms other deep CNN architectures when experimenting on the ColorFERET dataset which consists of 3,553 face images with 474 subjects. The ResNet-50 and FaceNet architectures had highly accuracies of 99.60% and 99.32%, respectively.

V. Conclusion

The key factor in achieving the highest accuracy in face verification systems consists of face detection and the face encoding process. In this paper, we have presented an effective face verification systems. First, the histogram of oriented gradients method combined with the linear support vector machine (HOG+SVM) was applied as the face detection process. The experimental results showed that the HOG+SVM method outperformed other face detection methods; CNN, Haar-Cascade, and Faced methods. There is no error while detecting faces in the BioID dataset with this method. Second, the FaceNet and the Resnet-50 architectures, which are the deep convolutional neural network (CNN), are proposed to use as the face encoding methods. Surprisingly, these two deep CNN architectures obtained an accuracy of 100% on the BioID and FERET datasets. Moreover, ResNet-50 architecture was slightly better than FaceNet architecture. The ResNet-50 and FaceNet architectures obtain very high verification accuracy on ColorFERET dataset, with accuracy of 99.60% and 99.32%, respectively.

References

[1] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1701–1708.

[2] D. Li, H. Zhou, and K. M. Lam, "High-Resolution face verification using pore-scale facial features," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2317–2327, 2015.

[3] A. Singh, M. Singh, and B. Singh, "Face detection and eyes extraction using Sobel edge detection and morphological operations," in *Conference on Advances in Signal Processing (CASP)*, 2016, pp. 295–300.

[4] C. Li, Z. Qi, N. Jia, and J. Wu, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers," in *IEEE 13th International Conference on Electronic Measurement and Instruments (ICEMI)*, 2017, pp. 483–487.

[5] E. K. Shimomoto, A. Kimura, and R. Belem, "A faster face detection method combining bayesian and Haar cascade classifiers," in *IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, 2015, pp. 7–12.

[6] A. Ade-ibijola and K. Aruleba, "Automatic attendance capturing using histogram of oriented gradients on facial images," in *IST-Africa Week Conference (IST-Africa)*, 2018, pp. 1–8.

[7] H. X. Jia and Y. J. Zhang, "Fast human detection by boosting histograms of oriented gradients," in *Proceedings of the Fourth International Conference on Image and Graphics Fast (ICIG)*, 2007, pp. 683–688.

[8] H. ChunYang and X. A. Wang, "Cascade face detection based on histograms of oriented gradients and support vector machine," in *10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, 2015, pp. 766–770.

[9] H. Shu, D. Chen, Y. Li, and Shengjin Wang State, "A highly accurate facial region

network for unconstrained face detection," in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 665–669.

[10] L. Pang, Y. Ming, and L. Chao, "F-DR Net:Face detection and recognition in one net," in *International Conference on Signal Processing (ICSP)*, 2018, pp. 332–337.

[11] A. R. Rivera, J. R. Castillo, and O. Chae, "Local directional number pattern for face analysis: face and expression recognition," *IEEE Trans. image Process.*, vol. 22, no. 5, pp. 1740–1752, 2013.

[12] F. Juefei-Xu and M. Savvides, "Encoding and decoding local binary patterns for harsh face illumination normalization," in *IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 3220–3224.

[13] D. Gong, Z. Li, W. Huang, X. Li, and D. Tao, "Heterogeneous face recognition: a common encoding feature discriminant approach," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2079–2089, 2017.

[14] A. Majumdar, R. Singh, and M. Vatsa, "Face verification via class sparsity based supervised encoding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1273–1280, 2017.

[15] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015, pp. 1–12.

[16] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 815–823, 2015.

[17] K. Cao, Y. Rong, C. Li, X. Tang, and C. C. Loy, "Pose-Robust Face Recognition via Deep Residual Equivariant Mapping," in *the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5187–5196.

[18] D. E. King, "Max-margin Object Detection," 2015.

[19] O. Surinta and S. Khruahong, "Tracking people and objects with an autonomous unmanned aerial vehicle using face and color detection," in *International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and*

Telecommunications Engineering (ECTI DAMT-NCON), 2019, pp. 206–210.

[20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[21] P. Viola and M. Jones, "Robust real-time object detection," *Vingtieme Siecle Rev. d'Histoire*, vol. 57, pp. 1–25, 2007.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.

[23] M. Karaaba, O. Surinta, L. Schomaker, and M. A. Wiering, "Robust face recognition by computing distances from multiple histograms of oriented gradients," in *IEEE Symposium Series on Computational Intelligence, (SSCI)*, 2015, pp. 203–209.

[24] V. N. Vapnik, *Statistical Learning Theory*. 1998.

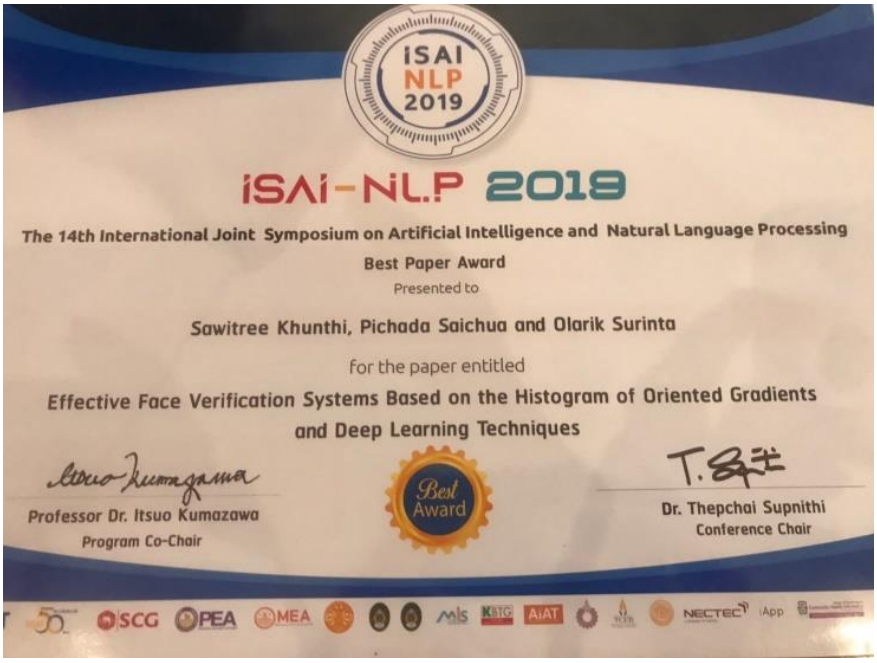
[25] O. Surinta, M. F. Karaaba, L. R. B. Schomaker, and M. A. Wiering, "Recognition of handwritten characters using local gradient feature descriptors," *Eng. Appl. Artif. Intell.*, vol. 45, pp. 405–414, 2015.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[27] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the hausdorff distance," in *Lecture Notes in Computer Science book series (LNCS)*, 2001, pp. 90–95.

[28] P. J. Phillips, P. J. Rauss, and S. a. Rizvi, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, 2000.

[29] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image Vis. Comput.*, pp. 295–306, 1998.



ภาพประกอบที่ ก-1 รางวัล The best Paper Award

ภาพประกอบที่ ก-2 รับรางวัล

ภาคผนวก ข

การแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย

ภาคผนวก ข

การแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย

ศูนย์ประสานงานและอุตสาหกรรมซอฟต์แวร์ ภาคตะวันออกเฉียงเหนือ (E-Saan Software Park)
อุทยานวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น เลขที่ 123 หมู่ที่ 16 อาคารอำนวยการอุทยานวิทยาศาสตร์ ภาคตะวันออกเฉียงเหนือ
มหาวิทยาลัยขอนแก่น อ.เมือง จ.ขอนแก่น 40002 โทรศัพท์ 0-4304-8048 โทรสาร 0-4320-2292

ที่ 22220371

ตุลาคม 2562

เรื่อง แจ้งผลการพิจารณาโครงการ "การแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย" รอบชิงชนะเลิศโครงการ
เรียน นาย ไอย์ซิน สุริเดีย

สิ่งที่ส่งมาด้วย ข้อตกลงการรับทุนอุดหนุน "การแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย" จำนวน 2 ฉบับ

ตามที่ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ ร่วมกับ
ศูนย์ประสานงานและอุตสาหกรรมซอฟต์แวร์ ภาคตะวันออกเฉียงเหนือ (E-Saan Software Park) ได้จัดให้มี
"การแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 22 (The Twenty-Two National Software Contest: NSC
2020)" โดยมีวัตถุประสงค์เพื่อสนับสนุน และ ส่งเสริมบุคลากรให้มีความรู้ความสามารถ ทางด้านการพัฒนาโปรแกรมคอมพิวเตอร์
ในระดับมัธยมศึกษา นิสิต นักศึกษา ตลอดจน พัฒนาบุคลากรในการเสริมโปรแกรม อันจะเป็นรากฐานที่สำคัญยิ่ง
ต่อการพัฒนาอุตสาหกรรมด้านซอฟต์แวร์ในอนาคตนั้น

ศูนย์ฯ ร่วมกับ ศูนย์ประสานงานและอุตสาหกรรมซอฟต์แวร์ ภาคตะวันออกเฉียงเหนือ (E-Saan Software Park)
ได้พิจารณาข้อเสนอโครงการและมีความยินดีขอแจ้งให้ท่านทราบว่า โครงการ "ระบบเตือนระวังความปลอดภัยด้วยเทคโนโลยี IP" รหัสโครงการ
22p1460037 โดยมี นางสาว สาวิตรี อันนี สังกัด สาขา เทคโนโลยีสารสนเทศ คณะวิทยาเทคโนโลยีสารสนเทศ และ วิทยาการสารสนเทศ
มหาวิทยาลัยมหาสารคาม เป็นผู้พัฒนาโครงการ และท่านเป็นอาจารย์ที่ปรึกษา ผ่านการพิจารณาเพื่อรับทุนสนับสนุนโครงการในวงเงินรวม
ทั้งสิ้น มีข้อเสนอแนะสำหรับโครงการของท่าน ซึ่งท่านสามารถตรวจสอบได้ในระบบออนไลน์ผ่านเว็บไซต์ http://nsc.nst.or.ac.th/GENA/
โดยท่านสามารถดำเนินการขอรับทุนจำนวน 3,000 บาท (สามพันบาทถ้วน) ได้ โดยดำเนินการดังนี้

1. พิมพ์ใบโครงการและรายชื่อที่ปรึกษาลงในข้อตกลงการรับทุนอุดหนุน "การแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย
   ครั้งที่ 22" ทั้ง 2 ฉบับ รายละเอียดตามสิ่งที่ส่งมาด้วย สำหรับนักพัฒนาที่อายุต่ำกว่า 15 ปี ต้องมีผู้ปกครองเซ็นชื่อร่วมลงนาม
   ในข้อตกลงการรับทุนด้วย
2. เตรียมสำเนาบัตรประชาชนหรือผู้ปกครองนิสิต นิสิต นักศึกษาของท่านสำหรับโครงการ พร้อมเตรียมนำมารับรองสำเนาถูกต้อง
3. นำส่งเอกสารข้อตกลงการรับทุนทั้ง 2 ฉบับ และ สำเนาบัตรนิสิตนิสิตของทุนโครงการการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย
   ครั้งที่ 22 ในวันที่ และเวลา ณ ตามที่ศูนย์ประสานงานระบุ โดยนักพัฒนาสามารถติดต่อขอรับทุนพัฒนาโครงการงวดที่ 1 จำนวน 3,000
   บาท ได้ที่ ศูนย์ประสานงานภูมิภาค ศูนย์ประสานงานและอุตสาหกรรมซอฟต์แวร์ ภาคตะวันออกเฉียงเหนือ (E-Saan Software Park)

หากท่านมีข้อสงสัยประการใด กรุณาติดต่อที่ ศูนย์ประสานงานและอุตสาหกรรมซอฟต์แวร์ ภาคตะวันออกเฉียงเหนือ (E-Saan Software Park)
โทรศัพท์ 0-4304-8048 หรือ ดูรายละเอียดเพิ่มเติมได้ที่ http://www.nectec.or.th/nsc

จึงเรียนมาเพื่อโปรดทราบและดำเนินการ จักขอบคุณยิ่ง

ขอแสดงความนับถือ

(นาย อภิรดี วงศ์ศิวิรพล)
ศูนย์ประสานงานและอุตสาหกรรมซอฟต์แวร์ ภาคตะวันออกเฉียงเหนือ (E-Saan Software Park)

ศูนย์ประสานงานและอุตสาหกรรมซอฟต์แวร์ ภาคตะวันออกเฉียงเหนือ (E-Saan Software Park)
โทรศัพท์ 0-4304-8048
โทรสาร 0-4320-2292
E-Mail: nsc.isan.kkusp@gmail.com
สำนักงาน นางสาว สาวิตรี อันนี

ภาพประกอบที่ ข- 1 ผลการพิจารณาโครงการ

แบบฟอร์มหน้าปกข้อเสนอโครงการ

รหัสโครงการ **22p14i0037**

ข้อเสนอโครงการ
การแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย

ชื่อโครงการ (ภาษาไทย) ระบบเฝ้าระวังความปลอดภัยด้วยกล้อง IP
(ภาษาอังกฤษ) IP Security Camera Monitoring System
ประเภทโปรแกรมที่เสนอ 14 โปรแกรมเพื่องานการพัฒนาด้านวิทยาศาสตร์และเทคโนโลยี

ทีมพัฒนา
หัวหน้าโครงการ

1. ชื่อ-นามสกุล	นางสาว สาวิตรี คันที	(หญิง)			
วัน/เดือน/ปีเกิด	02 ธันวาคม 2540 21 ปี 9 เดือน	ระดับการศึกษา	ปริญญาตรี		
สถานศึกษา	สาขา เทคโนโลยีสารสนเทศ ภาควิชา เทคโนโลยีสารสนเทศ คณะ วิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม				
ที่อยู่ตามทะเบียนบ้าน	เลขที่181 ถ.มณีโกมล ซ.- ม.1 ต.ท่าบ่อ อ.ท่าบ่อ จ.หนองคาย 43110				
สถานที่ติดต่อ	- เลขที่- ต.ขามเรียง อ.กันทรวิชัย จ.มหาสารคาม 44150				
โทรศัพท์	043754359	มือถือ	0918562899	e-mail	Sawitri0212@gmail.com

ลงชื่อ .....สาวิตรี.....

ผู้ร่วมโครงการ

2. ชื่อ-นามสกุล	นางสาว พิชาดา สายเสื้อ	(หญิง)			
วัน/เดือน/ปีเกิด	29 ตุลาคม 2540 21 ปี 10 เดือน	ระดับการศึกษา	ปริญญาตรี		
สถานศึกษา	สาขา เทคโนโลยีสารสนเทศ ภาควิชา เทคโนโลยีสารสนเทศ คณะ วิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม				
ที่อยู่ตามทะเบียนบ้าน	- เลขที่8 ถ.- ซ.- ม.10 ต.สระนกแก้ว อ.โพนทอง จ.ร้อยเอ็ด 45110				
สถานที่ติดต่อ	เลขที่- ต.ขามเรียง อ.กันทรวิชัย จ.มหาสารคาม 44150				
โทรศัพท์	043754359	มือถือ	0973350480	e-mail	pichadakt@gmail.com

ลงชื่อ .....พิชาดา.....

อาจารย์ที่ปรึกษาโครงการ

ชื่อ-นามสกุล	นาย โอฬาริก สุรินต๊ะ	(ชาย)			
ระดับการศึกษา	ปริญญาเอก	ตำแหน่งทางวิชาการ	ผู้ช่วยศาสตราจารย์		
สังกัด/สถาบัน	สาขา เทคโนโลยีสารสนเทศ ภาควิชา เทคโนโลยีสารสนเทศ คณะ วิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม				
สถานที่ติดต่อ	เลขที่- ต.ท่าขอนยาง อ.กันทรวิชัย จ.มหาสารคาม 44150				
โทรศัพท์	043754359	มือถือ	0959678342	e-mail	olarik.s@msu.ac.th

คำรับรอง "โครงการนี้เป็นความคิดริเริ่มของนักพัฒนาโครงการและไม่ได้ลอกเลียนแบบมาจากผู้อื่นผู้ใด
ข้าพเจ้ารอรับรองว่าจะให้คำแนะนำและสนับสนุนให้นักพัฒนาในความดูแลของข้าพเจ้าดำเนินการศึกษา/วิจัย/พัฒนา
ตามหัวข้อที่เสนอและจะทำหน้าที่ประเมินผลงานดังกล่าวให้กับโครงการฯ ด้วย"

ลงชื่อ .....

หัวหน้าสถาบัน (อธิการบดี/คณบดี/หัวหน้าภาควิชา/ผู้อำนวยการ/อาจารย์ใหญ่/หัวหน้าหมวด)

ชื่อ-นามสกุล	นาง ศศิธร แก้วมั่น	(หญิง)		
ตำแหน่งทางวิชาการ	ผู้ช่วยศาสตราจารย์	ตำแหน่งทางบริหาร		คณบดี/ ผ.อ.วิทยาเขตฯ
สังกัด/สถาบัน	มหาวิทยาลัยมหาสารคาม			
สถานที่ติดต่อ	เลขที่- ต.ขามเรียง อ.กันทรวิชัย จ.มหาสารคาม 44150			
โทรศัพท์	043754359	มือถือ		e-mail

คำรับรอง "ข้าพเจ้าขอรับรองว่าผู้พัฒนามีสิทธิ์ขอรับทุนสนับสนุนตามเงื่อนไขที่โครงการฯกำหนดและอนุญาต
ให้ดำเนินการศึกษา/วิจัย/พัฒนาตามหัวข้อที่ได้เสนอมานี้ในสถาบันได้ภายใต้การบังคับบัญชาของข้าพเจ้า"

ลงชื่อ .....

ภาพประกอบที่ ข-2 แบบฟอร์มหน้าปกข้อเสนอโครงการ

1. สาระสำคัญของโครงการ

ในปัจจุบันมีการนำเอาเทคโนโลยีมาประยุกต์ใช้ร่วมกับชีวิตประจำวันอย่างแพร่หลาย เช่น ใช้เพื่ออำนวยความสะดวกสบายให้กับมนุษย์มากยิ่งขึ้น ด้วยเหตุนี้จึงได้มีการนำเทคโนโลยีเข้ามาร่วมกับระบบรักษาความปลอดภัยเนื่องจากความปลอดภัยเป็นสิ่งสำคัญในการดำรงชีวิตของมนุษย์ ดังนั้นการติดตั้งกล้องวงจรปิดจึงเป็นทางเลือกหนึ่งของระบบรักษาความปลอดภัยที่จะทำให้การรักษาความปลอดภัยมีประสิทธิภาพมากยิ่งขึ้น และด้วยคุณสมบัติของกล้องไอพีที่ทางทีมผู้วิจัยได้เลือกใช้นั้นสามารถเชื่อมต่ออินเตอร์เน็ตได้โดยตรง ผู้ใช้งานจึงสามารถเรียกดูรูปภาพหรือวิดีโอได้แบบทันท่วงที (Real-time) และยังสามารถนำรูปภาพที่เก็บบันทึกไว้จากกล้องไอพีไปทำการประมวลผลเพื่อค้นหาบุคคล โดยอาศัยหลักการของการค้นหาใบหน้า (Face Detection) และนำใบหน้าไปเปรียบเทียบกับภาพใบหน้าที่ต้องการค้นหา (Query Image) เพื่อตรวจสอบ (Face Verification) หาค่าความคล้ายคลึง (Similarity) ได้อีกด้วย

คำสำคัญ (Keywords) : *Face Verification Systems, Face Detection, Face Verification, ResNet-50, Histogram of oriented gradients*

2. หลักการและเหตุผล

เทคโนโลยีในปัจจุบันมีความก้าวหน้าและพัฒนาอย่างรวดเร็ว ทำให้ผู้คนนำเอาเทคโนโลยีมาประยุกต์ร่วมกับชีวิตประจำวัน ตัวอย่างเช่น การประยุกต์เข้ากับการรักษาความปลอดภัย โดยการติดตั้งกล้องวงจรปิด (Closed-Circuit Television : CCTV) [1] การติดตั้งกล้องวงจรปิดนั้นเป็นที่นิยมอย่างมากไม่ว่าจะเป็นในองค์กรหรือแม้กระทั่งการติดตั้งตามบ้านเรือน เพราะกล้องวงจรปิดมีขนาดเล็ก สามารถติดตั้งไว้ได้ในทุกที่ พร้อมทั้งสามารถใช้เพื่อเฝ้าสังเกตการณ์ด้วยตนเองได้จากระยะไกล ภาพจากกล้องวงจรปิดยังสามารถบันทึกลงในเครื่อง DVR (Digital Video Recorder) เพื่อเก็บไว้เป็นหลักฐานเมื่อเกิดเหตุร้ายได้อีกด้วย การติดตั้งกล้องวงจรปิดส่งผลให้การรักษาความปลอดภัยมีประสิทธิภาพมากยิ่งขึ้น

ปัจจุบันกล้องวงจรปิดมีหลายรูปแบบ เช่น กล้องที่บันทึกรูปภาพหรือวิดีโอลงเก็บไว้ในเครื่อง DVR โดยเครื่อง DVR จะถูกนำไปใช้เป็นตัวกลางในการเชื่อมต่ออินเตอร์เน็ต ทำให้สามารถเรียกดูข้อมูลได้จากโทรศัพท์มือถือ และกล้องแบบไอพี (IP Camera) โดยกล้องลักษณะนี้ตัวกล้องจะสามารถเชื่อมต่ออินเตอร์เน็ตได้โดยตรง และตัวกล้องยังสามารถบรรจุการ์ดหน่วยความจำ (Memory

Card) จึงทำให้ไม่ต้องใช้เครื่อง DVR ในการช่วยบันทึก ผู้ใช้งานสามารถเรียกดูรูปภาพหรือวิดีโอผ่านแอพพลิเคชั่นบนโทรศัพท์มือถือ อีกทั้งยังสามารถดูวิดีโอได้แบบทันท่วงที (Real-time)

ด้วยเหตุผลที่กล่าวมาข้างต้น ทางคณะผู้วิจัยจึงได้พัฒนาระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพี โดยระบบจะเชื่อมต่อกับกล้องไอพี ทำให้สามารถใช้เป็นเครื่องมือในการเฝ้าระวังความปลอดภัยภายในอาคารบ้านเรือนได้ ทั้งนี้ คณะวิจัยจะนำรูปเอาภาพที่เก็บบันทึกไว้จากระบบเฝ้าระวัง ฯ ไปทำการประมวลผลเพื่อค้นหาบุคคล โดยอาศัยหลักการของการค้นหาใบหน้า (Face Detection) และนำใบหน้าไปเปรียบเทียบกับภาพใบหน้าที่ต้องการค้นหา (Query Image) เพื่อตรวจสอบ (Face Verification) หาค่าความคล้ายคลึง (Similarity) และแสดงผลลัพธ์ทางเว็บบราวเซอร์

3. วัตถุประสงค์

วัตถุประสงค์ในโครงการนี้คือเพื่อพัฒนาระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพี

4. ปัญหาหรือประโยชน์ที่เป็นเหตุผลให้ควรพัฒนาโปรแกรม

กล้องวงจรปิดในปัจจุบันนั้นส่วนใหญ่มักจะมีความสามารถเพียงบันทึกรูปภาพหรือวิดีโอได้เท่านั้นพร้อมทั้งทำการติดตั้งและใช้งานยากต้องให้ช่างหรือผู้เชี่ยวชาญทำการติดตั้งให้หรือหากต้องการกล้องวงจรปิดที่มีฟังก์ชันการทำงานที่นอกเหนือจากที่กล่าวไปข้างต้นจะต้องมาพร้อมกับราคาที่สูงจนทุกคนไม่สามารถเลือกซื้อหรือติดตั้งได้ ด้วยเหตุนี้เองคณะผู้วิจัยจึงได้ทำการนำกล้องไอพีซึ่งมีราคาที่ทุกคนสามารถเป็นเจ้าของได้มาพัฒนาร่วมกับเทคนิคในด้าน Image processing [2] เพื่อเพิ่มฟังก์ชันในการตรวจจับใบหน้าบุคคลได้อีกด้วย

5. เป้าหมายและขอบเขตของโครงการ

ระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพีประกอบด้วยขอบเขตดังต่อไปนี้

5.1 ส่วนของการรับภาพ และตรวจจับใบหน้า

- ตรวจจับใบหน้า (Face Detection) จาก IP Camera
- บันทึกรูปภาพ และวิดีโอที่ได้จาก IP Camera
- ตรวจสอบรูปภาพย้อนหลัง
- ดูภาพจาก IP Camera แบบ Real-time

5.2 ส่วนของ Web Application สำหรับใช้ในการตรวจสอบใบหน้า

- ตรวจจับใบหน้า (Face Detection) จากรูปภาพ

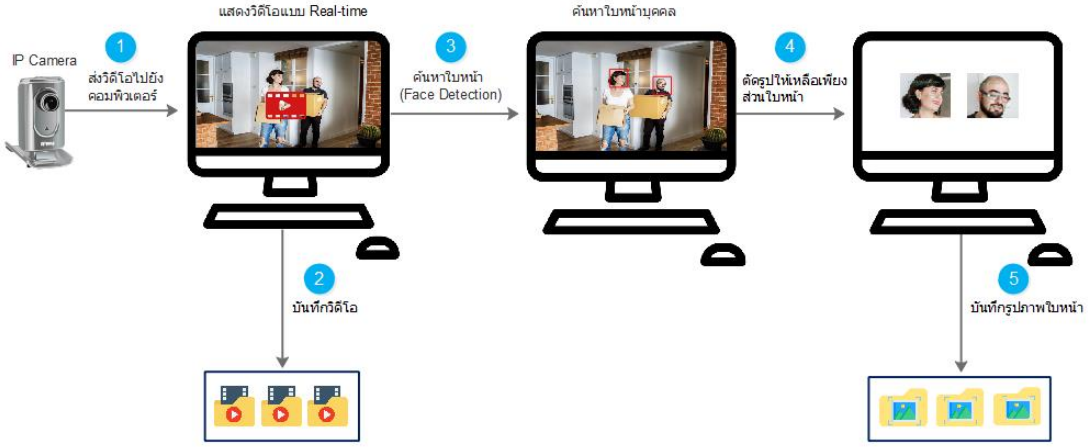
- ตรวจสอบใบหน้า (Face Verification) ที่คล้ายคลึงกับภาพที่มีฐานข้อมูล (Query image)
- แสดงรูปภาพของใบหน้าที่มีความคล้ายคลึงกัน
- ระบุตำแหน่งของรูปภาพ

6. รายละเอียดของการพัฒนา

6.1 เนื้อเรื่องย่อ (Story Board)

6.1.1 แบบจำลองของโปรแกรมที่ต้องการจะพัฒนาขึ้น

ส่วนของการรับภาพ และตรวจจับใบหน้า

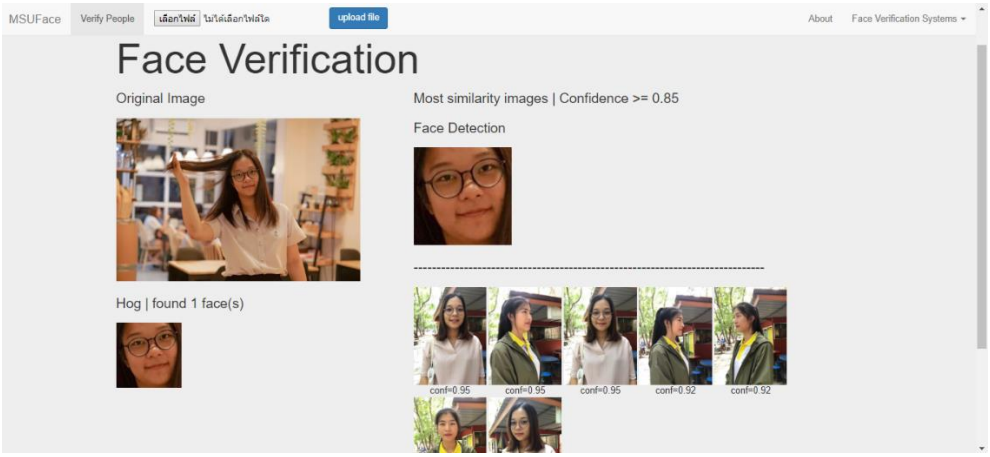


ภาพประกอบที่ ข-3 รูปแบบการทำงานของระบบในส่วนของการรับภาพ และตรวจจับใบหน้า

จากภาพประกอบที่ ข-2 แสดงให้เห็นถึงรูปแบบการทำงานของระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพีในส่วนของการรับภาพ และตรวจจับใบหน้าโดยในการรับและส่งข้อมูลภาพนั้นจะทำได้โดยการส่งข้อมูล

ภาพจากกล้องไอพีไปยังคอมพิวเตอร์ และข้อมูลภาพที่รับมานั้นจะมาแสดงผลได้แบบ Real time โดยทางทีมผู้วิจัยได้เลือกใช้ Algorithms HOG [25] เพื่อใช้ในการตรวจจับใบหน้า จากนั้นระบบจะนำข้อมูลรูปภาพที่ได้จากการตรวจจับใบหน้า และวิดีโอที่บันทึกไว้ได้ส่งไปเก็บไว้ที่โฟลเดอร์ในเครื่องคอมพิวเตอร์เพื่อทำการประมวลผลต่อไป

ส่วนของ Web Application



ภาพประกอบที่ ข-4 รูปแบบการทำงานของระบบส่วนของ Web Application

จากภาพประกอบที่ ข-3 แสดงให้เห็นถึงรูปแบบการทำงานของระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพีในส่วนในส่วนของ Web Application ซึ่งจะสามารถตรวจสอบและค้นหาความคล้ายคลึงของใบหน้าที่ต้องการจะตรวจสอบได้ โดยในขั้นตอนแรกผู้ใช้งานจะต้องเลือกรูปภาพที่ต้องการจะตรวจสอบและค้นหาจากนั้นระบบจะทำการตรวจจับใบหน้าและนำไปเปรียบเทียบกับรูปภาพที่มีอยู่ในฐานข้อมูล โดยทางทีมผู้วิจัยได้เลือกใช้ Algorithms Resnet-50 [4] เพื่อใช้ในการตรวจสอบคล้ายคลึงของใบหน้า และจะแสดงผลใบหน้าที่คล้ายคลึงกันออกมาโดยจะเรียงลำดับความคล้ายคลึงจากมากไปน้อย

6.2 เทคนิคหรือเทคโนโลยีที่ใช้

Library ที่ใช้ในการการพัฒนา : OpenCV, Scikit-learn, TensorFlow, Keras

Algorithms ที่ใช้ในการพัฒนา : Histogram of oriented gradients และ Resnet50 ใช้ในการเปรียบเทียบใบหน้าที่คล้ายคลึงกัน

6.3 เครื่องมือที่ใช้ในการพัฒนา

ระบบปฏิบัติการ : Microsoft Windows 10

ภาษาที่ใช้ในการพัฒนา : Python

เครื่องมือที่ใช้ในการพัฒนา : Visual Studio Code และ Atom

6.4 รายละเอียดโปรแกรมที่จะพัฒนา (Software Specification)

6.4.1 Input and Output Specification

ส่วนของการรับภาพ และตรวจจับใบหน้า

- Input : รูปภาพ และวิดีโอที่ได้จากจากกล้อง IP

- Output : รูปภาพที่ได้จากการตรวจเจอใบหน้าบุคคล และวิดีโอเพื่อนำไว้ใช้ตรวจสอบในภายหลัง

ส่วนของ Web Application

- Input : รูปภาพที่ได้จากกล้อง IP ที่ค้นหา และตรวจพบใบหน้าบุคคล

- Output : ผลการหาค่าความคล้ายคลึงของใบหน้าบุคคล

6.4.2 Functional Specification

- แสดงวิดีโอได้แบบ Real-time

- ตรวจจับใบหน้าบุคคลจากวิดีโอที่รับมาจากกล้อง IP

- บันทึกรูปภาพและวิดีโอลงในคอมพิวเตอร์เพื่อตรวจสอบในภายหลัง

- ตรวจสอบใบหน้าที่รับเข้ามากับใบหน้าในฐานข้อมูลเพื่อความปลอดภัย

6.5 ขอบเขตและข้อจำกัดของโปรแกรมที่พัฒนา

ข้อจำกัดทางด้านฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ และกล้องไอพีจะต้องอยู่ในเครือข่ายเดียวกัน

- จำเป็นต้องเชื่อมต่ออินเตอร์เน็ตตลอดเวลาเพื่อรับ และส่งภาพ

ข้อจำกัดทางด้านซอฟท์แวร์

- ต้องใช้ library ดังต่อไปนี้ OpenCV, Scikit-learn, TensorFlow, Keras เพื่อการรับภาพจากกล้องไอพีเพื่อที่จะนำมาแสดงผล และค้นหาพร้อมทั้งตรวจสอบใบหน้า

บรรณานุกรม

[1] บริษัท ซายเนค เทคโนโลยี จำกัด, “กล้องวงจรปิด CCTV (Closed Circuit Television System),” 30 เมษายน 2561. [ออนไลน์]. Available: https://www.zynek.com/content/8491/-cctv-closed-circuit-television-system. [%1 ที่เข้าถึง17 กันยายน 2562].

[2] Jarat, “Image processing เทคโนโลยีการประมวลผลภาพ,” 11 ตุลาคม 2552. [ออนไลน์]. Available: https://jaratcyberu.blogspot.com/2009/10/image-processing.html?fbclid=IwAR1M2eZvBjfJ8PaELqargS2i0_LXXTDSLRB4AErkppObZhmwkqklzeqVcjw. [%1 ที่เข้าถึง17 กันยายน 2562].

[3] VOCAL Technologies, “Histogram of Oriented Gradients (HOG) for Object Detection,” [ออนไลน์]. Available: https://www.vocal.com/video/histogram-of-oriented-gradients-hog-for-object-detection/. [%1 ที่เข้าถึง15 มีนาคม 2562].

[4] K. He, X. Zhang, S. Ren และ J. Sun, “Deep Residual Learning for Image Recognition,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 770-778, 2559.

ภาคผนวก ค

คู่มือการตั้งค่าใช้งานกล้องไอพี

ภาคผนวก ค

คู่มือการตั้งค่าใช้งานกล้องไอพี

1. การตั้งค่าเพื่อให้กล้องเชื่อมต่อกับอินเตอร์เน็ตแบบไร้สาย (WIFI)

1.1 เชื่อมต่อกล้องไอพีกับสายแลน (LAN)

1.2 เชื่อมต่ออินเตอร์เน็ตที่อยู่ในวงแลนเดียวกับที่เชื่อมกับกล้องไอพี

1.3 ค้นหาเลขไอพีของกล้อง

1.4 เมื่อทราบเลขไอพีของกล้องแล้วต้องทำขั้นตอนดังภาพประกอบที่ ค-1

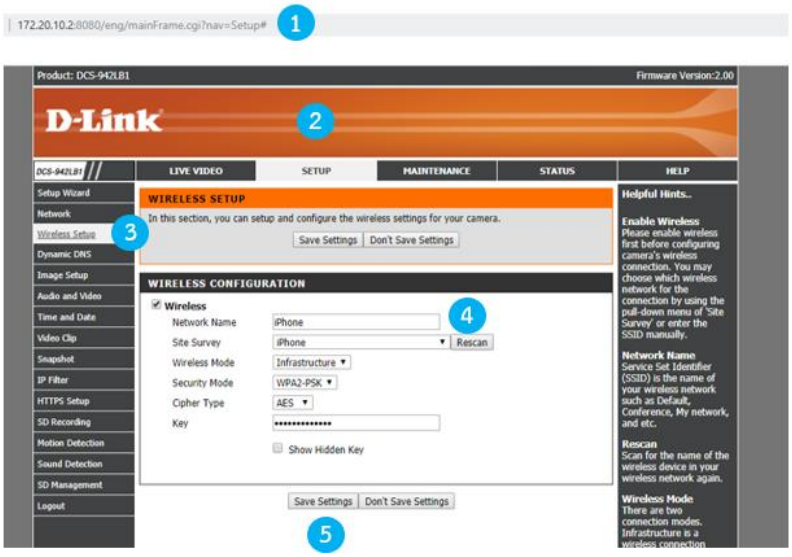
(1) เปิดบราวเซอร์แล้วพิมพ์เลขไอพีของกล้องเพื่อเข้าไปตั้งค่า

(2) คลิกที่ปุ่ม SETUP

(3) คลิกที่ปุ่ม Wireless Setup ในบริเวณแถบเมนูด้านซ้าย

(4) เลือกอินเตอร์เน็ตไร้สายที่ต้องการเชื่อมต่อในช่อง Site Survey โดยในการเลือกอินเตอร์เน็ตนั้นผู้ใช้งานสามารถเลือกเครือข่ายใดก็ได้ตามความสะดวก ในตัวอย่างนี้เลือกเครือข่ายของโทรศัพท์ผู้ใช้งาน (iPhone) จากนั้นใส่ password ของเครือข่ายให้เรียบร้อย

(5) ตรวจเช็คความถูกต้องและคลิกที่ปุ่ม Save Settings



ภาพประกอบที่ ค-1 ขั้นตอนการ setup กล้องไอพี

1.5 เมื่อระบบทำการประมวลผลเสร็จแล้วให้ผู้ใช้งานทำการถอดสายแลน และทำการปิดเปิดกล้องไอพีใหม่

1.6 หากตัวกล้องสามารถเชื่อมต่อกับอินเตอร์เน็ตแบบไร้สายได้สำเร็จ ไฟด้านหลังของกล้องจะปรากฏเป็นสีเหลือง

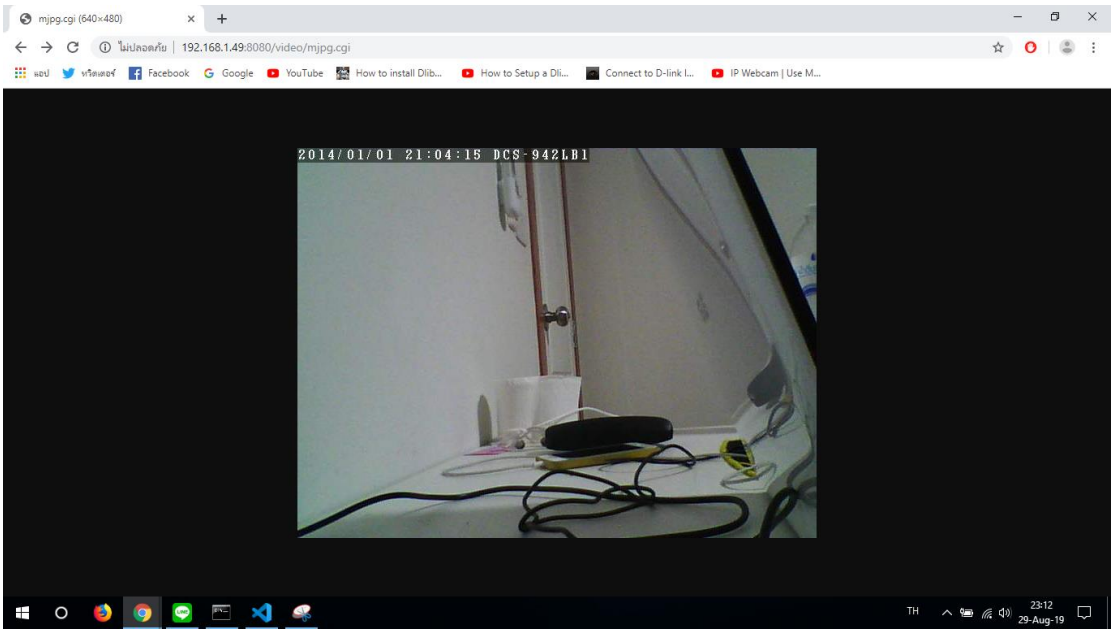
1.7 ให้ผู้ใช้งานทำการเชื่อมต่ออินเตอร์เน็ตเครือข่ายเดียวกับที่ตั้งค่าให้กับกล้องไอพีจากนั้นทำการค้นหาเลขไอพีของกล้องอีกครั้ง

1.8 ทดสอบการรับภาพจากกล้องไอพีได้โดยเข้าไปที่ URL ดังตัวอย่างต่อไปนี้

username:password@yourIP/video/mjpg.cgi

(http://admin:123456789@172.20.10.2/video/mjpg.cgi)

ผลลัพธ์จะแสดงดังภาพประกอบที่ ค-2



ภาพประกอบที่ ค-2 หน้าจอการรับภาพจากกล้องไอพี

ภาคผนวก ง

คู่มือการใช้งานโปรแกรม

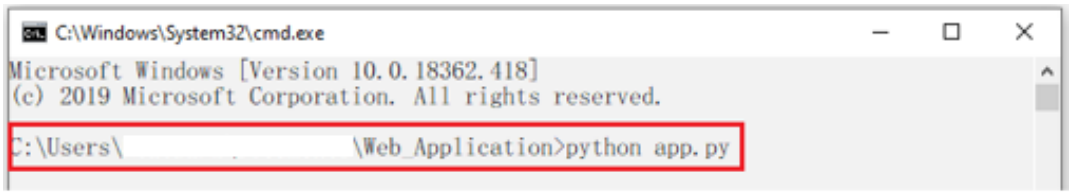
ภาคผนวก ง

คู่มือการใช้โปรแกรม

การทำงานของโปรแกรมจะแบ่งออกเป็น 2 ส่วน ได้แก่ 1. ส่วนของการนำภาพมาตรวจจับใบหน้า และ2. ส่วนของการนำภาพมาตรวจสอบใบหน้า โดยการใช้งานโปรแกรมมีดังต่อไปนี้

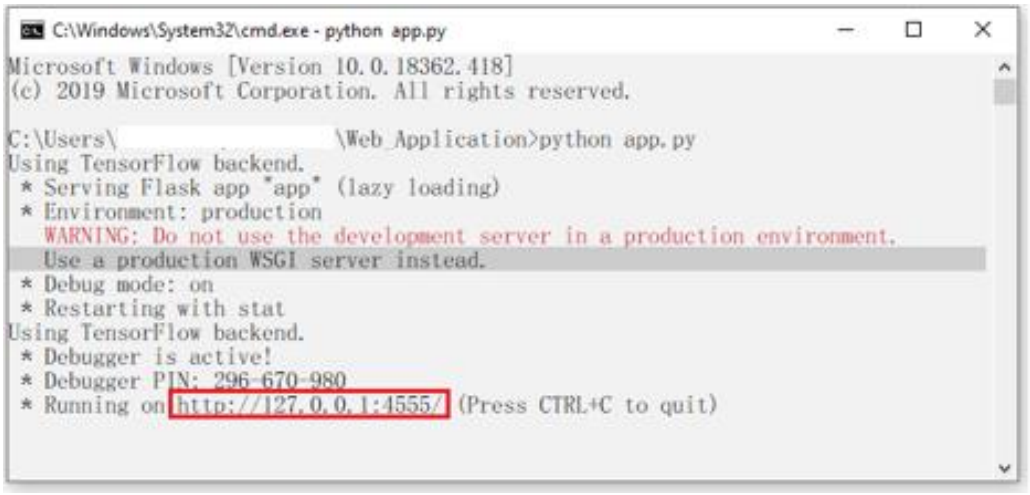
1. ส่วนของการนำภาพมาตรวจจับใบหน้า

1.1 เรียกใช้งานโปรแกรมด้วย Command Prompt (cmd) ตามด้วยคำสั่ง python app.py โดยใน app.py เป็นคำสั่งการทำงานของโปรแกรมทั้งหมด

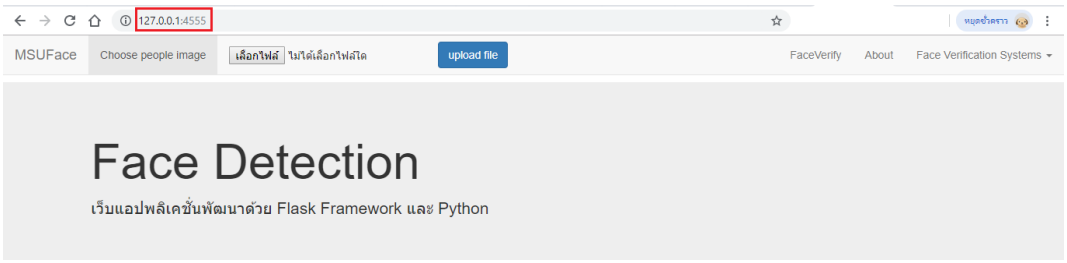


ภาพประกอบที่ ง-1 เรียกใช้งานโปรแกรมด้วย Command Prompt (cmd)

1.2 คัดลอกลิงก์เพื่อเปิดใช้งานโปรแกรมผ่านเว็บบราว์เซอร์

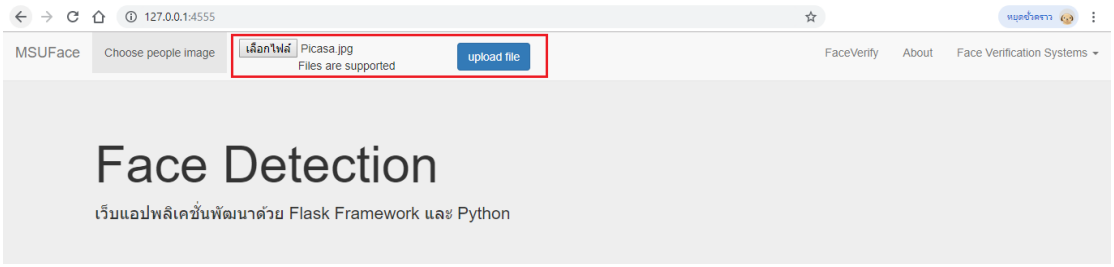


ภาพประกอบที่ ง-2 ลิงก์ในการเปิดใช้งานโปรแกรมผ่านเว็บบราว์เซอร์ (1)



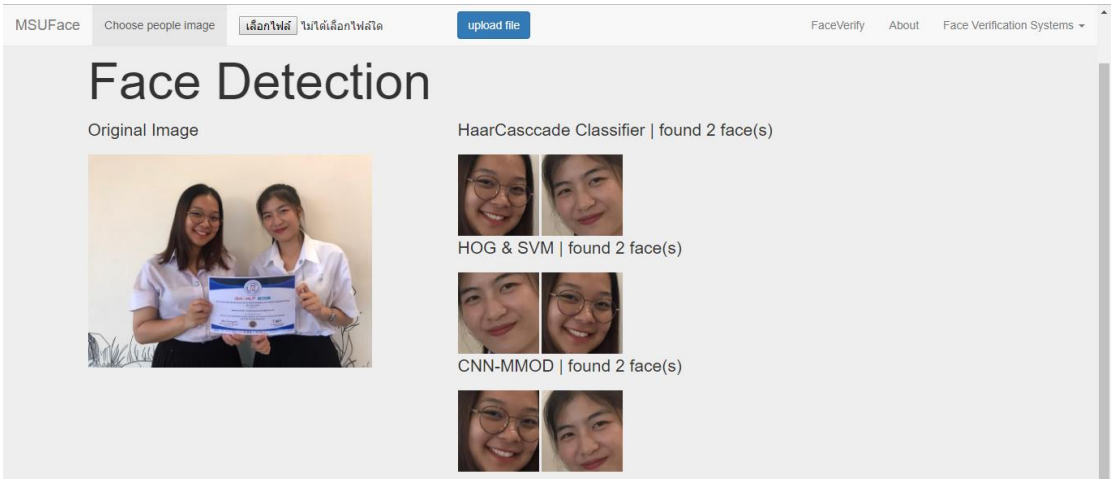
ภาพประกอบที่ ง-3 ลิงก์ในการเปิดใช้งานโปรแกรมผ่านเว็บบราว์เซอร์ (2)

1.3 หน้าเว็บบราว์เซอร์สำหรับการตรวจจับใบหน้า (Face Detection) เลือกภาพที่ต้องการและกด upload file



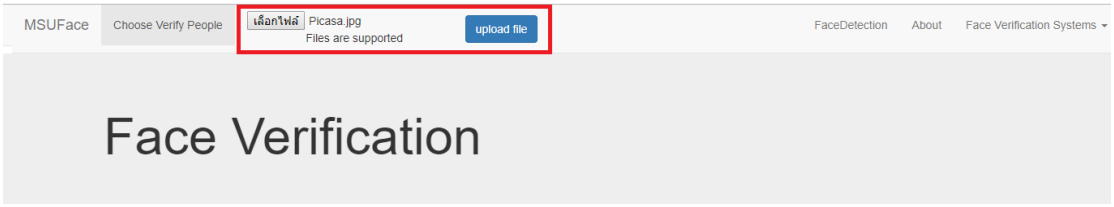
ภาพประกอบที่ ง-4 หน้าเว็บบราว์เซอร์สำหรับการตรวจจับใบหน้า

1.4 ผลลัพธ์ของโปรแกรมในส่วนของการตรวจจับใบหน้า



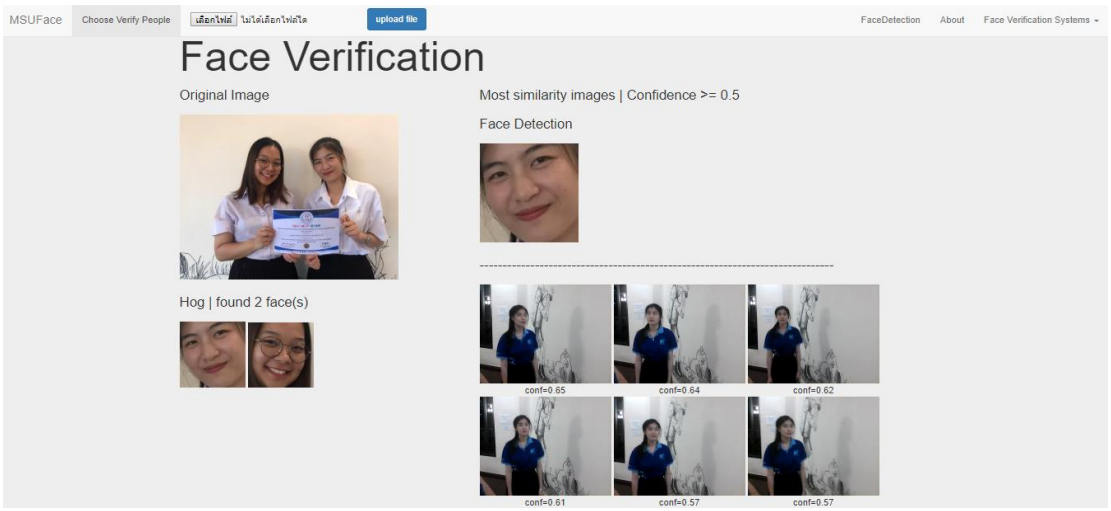
ภาพประกอบที่ ง-5 ผลลัพธ์ของโปรแกรมในส่วนของการตรวจจับใบหน้า

1.5 หน้าเว็บบราว์เซอร์สำหรับการตรวจสอบใบหน้า (Face Verification) เลือกภาพที่ต้องการตรวจสอบและกด upload file

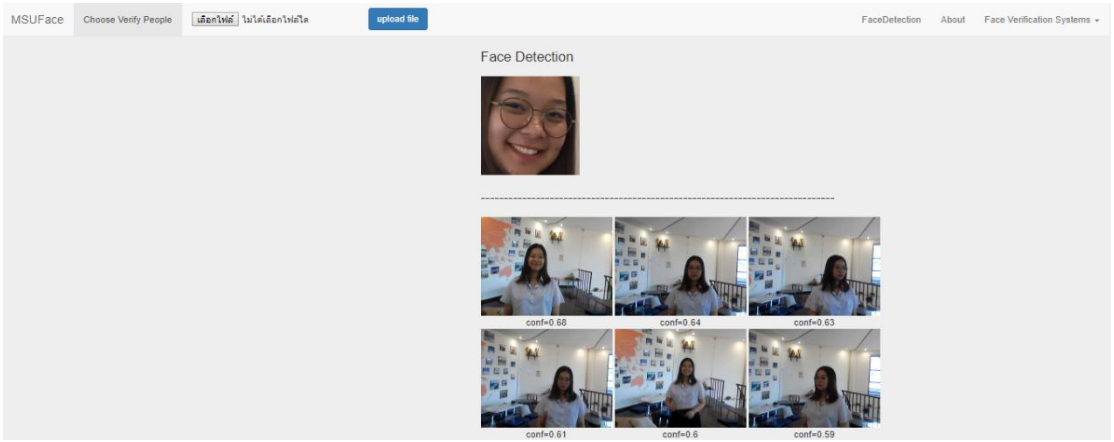


ภาพประกอบที่ ง-6 หน้าเว็บบราว์เซอร์สำหรับการตรวจสอบใบหน้า

1.6 ผลลัพธ์ของโปรแกรมในส่วนของการตรวจสอบใบหน้า



ภาพประกอบที่ ง-7 ผลลัพธ์ของโปรแกรมในส่วนของการตรวจสอบใบหน้า (1)



ภาพประกอบที่ ง-8 ผลลัพธ์ของโปรแกรมในส่วนของการตรวจสอบใบหน้า (2)

ประวัติผู้จัดทำโครงงานปริญญานิพนธ์

ชื่อ – นามสกุล : นางสาวสาวิตรี คันที
รหัส : 59011211130
ชื่อปริญญานิพนธ์ : ระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพี
: IP Security Camera Monitoring System
สาขาวิชา : เทคโนโลยีสารสนเทศ
คณะ : วิทยาการสารสนเทศ

ประวัติส่วนตัว

เกิดวันที่ : 02 มีนาคม พ.ศ. 2540
ที่อยู่ : 181 หมู่ 1 ตำบลท่าบ่อ อำเภอท่าบ่อ จังหวัดหนองคาย 43110
E-mail : Sawitri0212@gmail.com

ประวัติการศึกษา

ประถมศึกษา : โรงเรียนโกมลวิทยาคาร จังหวัดหนองคาย
มัธยมศึกษาตอนต้น : โรงเรียนท่าบ่อ จังหวัดหนองคาย
มัธยมศึกษาตอนปลาย : โรงเรียนท่าบ่อ จังหวัดหนองคาย
ปริญญาตรี : สาขาวิชาเทคโนโลยีสารสนเทศ
คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
จังหวัดมหาสารคาม

ประวัติผู้จัดทำโครงงานปริญญานิพนธ์

ชื่อ – นามสกุล : นางสาวพิชาดา สายเชื้อ
รหัส : 59011211167
ชื่อปริญญานิพนธ์ : ระบบเฝ้าระวังความปลอดภัยด้วยกล้องไอพี
: IP Security Camera Monitoring System
สาขาวิชา : เทคโนโลยีสารสนเทศ
คณะ : วิทยาการสารสนเทศ

ประวัติส่วนตัว

เกิดวันที่ : 29 ตุลาคม พ.ศ. 2540
ที่อยู่ : 8 หมู่ 10 ตำบลสระนกแก้ว อำเภอโพนทอง จังหวัดร้อยเอ็ด 45110
E-mail : pichadakt@gmail.com

ประวัติการศึกษา

ประถมศึกษา : โรงเรียนอนุบาลโพนทอง จังหวัดร้อยเอ็ด
มัธยมศึกษาตอนต้น : โรงเรียนสาธิตมหาวิทยาลัยมหาสารคาม (ฝ่ายมัธยม)
จังหวัดมหาสารคาม
มัธยมศึกษาตอนปลาย : โรงเรียนสาธิตมหาวิทยาลัยมหาสารคาม (ฝ่ายมัธยม)
จังหวัดมหาสารคาม
ปริญญาตรี : สาขาวิชาเทคโนโลยีสารสนเทศ
คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
จังหวัดมหาสารคาม