

ระบบตรวจนับจำนวนผู้โดยสารอัตโนมัติสำหรับการบริหารจัดการ รถไฟฟ้า

อุเทน วงศ์ธนาวิวัฒน์¹ และ เกรียงศักดิ์ เตมีย¹

¹ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

Emails: somchai@nu.ac.th, kreangsakt@nu.ac.th

บทคัดย่อ

การศึกษานี้มีวัตถุประสงค์เพื่อช่วยแก้ปัญหาบริการรถไฟฟ้าภายในมหาวิทยาลัยนเรศวร ประกอบด้วย ความแออัด ความล่าช้าของรถไฟฟ้า จำนวนรถไฟฟ้าที่ไม่เพียงพอต่อจำนวนผู้โดยสารที่จะใช้บริการภายในมหาวิทยาลัยนเรศวร ทำให้บริการรถไฟฟ้าไม่ได้รับความนิยม เพื่อแก้ปัญหาดังกล่าวผู้วิจัยพัฒนาระบบตรวจนับใบหน้าผู้โดยสาร ณ ป้ายรถรถไฟฟ้า เพื่อบันทึกจำนวนผู้โดยสาร ด้วยการใช้เทคโนโลยีการตรวจจับใบหน้า (Face Detection) ของ Python – OpenCV แสดงผลโดยใช้ Google Maps มาพัฒนานบนบอร์ด Raspberry PI3 ระบบที่พัฒนาขึ้นนั้น สามารถรายงานจำนวนผู้โดยสารที่กำลังรอรถไฟฟ้าได้อย่างถูกต้อง ซึ่งข้อมูลเหล่านี้สามารถนำไปประยุกต์ใช้ในการวางแผนการเดินทางให้สอดคล้องกับจำนวนคนได้ นับเป็นทางเลือกหนึ่งของการช่วยแก้ปัญหาบริการรถไฟฟ้า และช่วยเพิ่มประสิทธิภาพในการบริการของขนส่งมวลชนภายในมหาวิทยาลัยนเรศวร

ABSTRACT

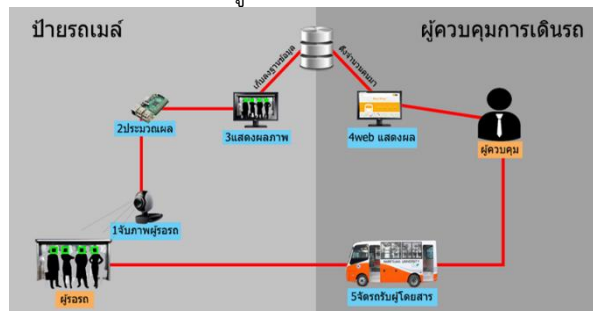
This study aims to relieve electric bus problems in Naresuan university include passenger crowding, bus delays and lack of buses to serve passengers cause the electric bus popularity decrease. We propose the automatic passenger counting system through face detection at bus stop in Naresuan university by Python-OpenCV and Google Maps API implement on Raspberry PI3. The system can correctly count a number of passenger at bus stop and report to bus operation manager for adjusting a number of buses correspond to passengers. As a results this system will be alternate way to help electric bus problems and increase service performance efficiency in Naresuan university mass transit.

คำสำคัญ– ตรวจนับใบหน้า; Raspberry PI; Python; OpenCV; รถไฟฟ้า

1. บทนำ

ปัจจุบันปัญหาเกี่ยวกับการใช้บริการรถไฟฟ้าขนส่งมวลชนภายในมหาวิทยาลัยนเรศวรนั้นมีมากมาย อาทิเช่น ปัญหาความแออัดบนรถไฟฟ้าทำให้เกิดความไม่สะดวกสบายในการเดินทาง ปัญหารถไฟฟ้าไม่เพียงพอต่อผู้โดยสาร รถไฟฟ้ามารับผู้โดยสารช้าทำให้ผู้โดยสารต้องรอรถเป็นเวลานานเป็นต้น ปัญหาเหล่านี้เกิดจากการไม่ทราบว่ามีผู้โดยสารในแต่ละป้ายเป็นจำนวนเท่าไร จึงทำให้การจัดการรถไฟฟ้าไปรับไม่เพียงพอต่อผู้โดยสาร และรถไฟฟ้าต้องเดินรถให้ครบทุกป้ายในหนึ่งรอบการเดินรถ แม้บางป้ายไม่มีผู้โดยสารเลยก็ตาม ทำให้ผู้รอรถในป้ายอื่นๆ ต้องรอรถไฟฟ้าเป็นเวลานาน และสิ้นเปลืองพลังงานจากการเดินรถอีกด้วย ปัญหานี้อาจทำให้มีผู้ใช้บริการรถไฟฟ้าเป็นจำนวนน้อยได้

ผู้ศึกษาจึงหาแนวทางแก้ปัญหาเหล่านี้โดยการพัฒนากระบวนตรวจนับใบหน้าผู้โดยสาร ณ ป้ายรถไฟฟ้า เพื่อนำไปช่วยในการเดินรถให้สอดคล้องระหว่างจำนวนรถไฟฟ้า กับจำนวนคน ด้วยเทคโนโลยีการตรวจจับใบหน้า(Face Detection) ที่พัฒนาด้วยภาษา Python และ OpenCV ลงบนบอร์ด Raspberry PI3 Model B และติดตั้งที่ป้ายรถ เพื่อนับจำนวนผู้โดยสารที่รอรถ แสดงผลเป็น Responsive Website ที่มีการดึง Google Maps API มาช่วยในการแสดงผล และแสดงจำนวนคนที่รอรถอยู่ในป้ายแบบ Real-time ดังรูปที่ 1



รูปที่ 1 หลักการทำงานของระบบ

รูปที่ 1 แสดงหลักการทำงานของระบบ อุปกรณ์จะติดตั้ง ณ ป้ายรถไฟฟ้าเพื่อนับจำนวนผู้โดยสาร จากนั้นแสดงจำนวนผู้โดยสารแต่ละป้ายรถไฟฟ้าเพื่อให้ผู้ควบคุมการเดินรถตัดสินใจปรับจำนวนรถไฟฟ้าให้เหมาะสมกับจำนวนผู้โดยสารที่กำลังรอรถไฟฟ้า

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1. การประมวลผลภาพ (Image Processing)

ที่ผู้วิจัยใช้ Raspberry Pi ประมวลผลภาพ เพื่อให้ทราบว่าภาพนั้นคือภาพอะไร หรือมีสิ่งที่สนใจอยู่ในภาพหรือไม่ โดยใช้เว็บแคมเป็นตัวเก็บภาพแล้วส่งให้ Raspberry Pi ประมวลผลตัดสินว่าในภาพมีสิ่งที่สนใจอยู่หรือไม่ การคิดคำนวณนั้นมีหลายวิธี ซึ่งแต่ละวิธีก็มีประโยชน์แตกต่างกันไป ไม่ว่าจะเป็นการนำเอาสีแต่ละจุด (Pixel) มาคิด, การคิดคำนวณเป็นบริเวณหลายๆ จุดรวมๆ กัน (Area) เช่น การดูพื้นผิว (Texture), การดูรูปแบบ (Pattern), การวิเคราะห์หารูปทรง (Shape) หรือการวิเคราะห์แบบอื่นๆ เพื่อหาค่าที่สามารถระบุได้ว่า ภาพนั้นมีลักษณะอย่างไร ซึ่งจะนำไปสู่การตัดสินใจได้ว่า สิ่งนั้นเป็นสิ่งที่กำลังค้นหาหรือสนใจอยู่หรือไม่ [1]

2.2. OpenCV (Open Source Computer Vision Library)

เป็นไลบรารี ที่ผู้วิจัยนำมาใช้ในการตรวจจับใบหน้า สามารถดาวน์โหลดได้ทั่วไปและง่ายต่อการเรียกใช้ใน Python ภายใต้ระบบปฏิบัติการ Linux OpenCV มีเพื่อให้เกิดความง่ายต่อการใช้งานและ สามารถสร้างการทำงานที่ซับซ้อนได้อย่างรวดเร็ว [2]

2.3. การตรวจจับใบหน้า (Face Detection)

คือกระบวนการค้นหาใบหน้าของบุคคลจากภาพหรือวิดีโอ หลังจากนั้นก็จะทำการประมวลผลภาพใบหน้าที่ได้สำหรับขั้นตอนถัดไปเพื่อให้ภาพใบหน้าที่ตรวจจับได้ง่ายต่อการจำแนก และ อัลกอริทึมที่ใช้ในการตรวจจับใบหน้าในปัจจุบันก็มีอยู่ด้วยกันหลายวิธีซึ่งอัลกอริทึมในการตรวจจับใบหน้าที่ดีนั้นมีส่วนช่วยในการจำแนกใบหน้าได้แม่นยำ และรวดเร็วขึ้นเป็นอย่างมาก หลักการพื้นฐานของอัลกอริทึมของ Viola-Jones คือการสแกน sub-window เพื่อตรวจหาใบหน้าจากรูปภาพอินพุต การประมวลผลภาพแบบทั่วไปจะใช้การปรับขนาดภาพเข้าแตกต่างกันหลายๆขนาด และใช้ตัวตรวจหา (Detector) ที่มีขนาดคงที่ค้นหาวัตถุ ซึ่งวิธีนี้กินเวลาในการคำนวณมาก เนื่องมาจากการคำนวณบนรูปภาพที่มีขนาดแตกต่างกัน

Viola-Jones ได้เสนอวิธีใหม่โดยการปรับขนาดตัวตรวจหาแทนที่จะปรับขนาดภาพเข้า และใช้ตัวตรวจหาค้นหาวัตถุหลายๆรอบ ซึ่งทั้งสองวิธีนี้จะใช้เวลาในการคำนวณไม่ต่างกันมากนัก แต่ Viola-Jones ได้คิดค้นตัวตรวจหาที่ใช้จำนวนครั้งใน

การคำนวณคงที่แม้จะมีขนาดของภาพแตกต่างกัน โดยตัวตรวจหาดังกล่าวนี้นี้สร้างขึ้นโดยใช้ features ของ Haar wavelets [3] ซึ่งผู้วิจัยได้ใช้หลักการนี้ในการตรวจจับใบหน้า

2.4. งานวิจัยที่เกี่ยวข้อง

ธนิสรา [4] วิธีฮิวริสติกสำหรับการจัดตารางเดินรถประจำทาง A HEURISTIC METHOD FOR BUS SCHEDULING งานวิจัยนี้จะเกี่ยวข้องกับนโยบายและการพัฒนาการจัดตารางเวลาการให้บริการรถประจำทาง เพื่อลดจำนวนของรถประจำทางที่ให้บริการ และมีค่าใช้จ่ายในการดำเนินงานน้อยที่สุด ซึ่งการวางแผนการให้บริการของรถประจำทางจะประกอบด้วย 5 ขั้นตอน คือ การกำหนดเส้นทางและความถี่ในการให้บริการ เป็นการกำหนดนโยบายการจัดตารางและความถี่ของการให้บริการการเดินรถประจำทาง

วิรัตน์ และ นพพน [5] ระบบควบคุมความปลอดภัยบน raspberry pi ด้วยเทคนิค Eigen face และการปรับเปลี่ยนสถานะแสง Security controlled system on raspberry pi with Eigen face technique and illumination change บทความชิ้นนี้จะเกี่ยวข้องกับการประมวลผลภาพ โดยค้นหาใบหน้าจากภาพที่ได้จากกล้อง และตัดเฉพาะส่วนของใบหน้านั้นเพื่อทำการวิเคราะห์วัตถุตัวบุคคลว่าเป็นใบหน้าของบุคคลใด โดยนำมาเปรียบเทียบกับใบหน้า ที่มีอยู่ในฐานข้อมูล พัฒนาเป็นโปรแกรมสำหรับเปิดปิดประตูอัตโนมัติ และการปรับปรุงสถานะแสงของใบหน้าก่อนการ train ด้วยอัลกอริทึม Eigenface

วุฒิชัย และ ชรศักดิ์ [6] การพัฒนาระบบรู้จำใบหน้าบุคคล โดยใช้โครงข่ายประสาทเทียม Development of the Human Face Recognition System using the Artificial Neural Network บทความนี้นำเสนอแนวทางการพัฒนาระบบที่สามารถรู้จำบุคคลด้วยเค้าโครงใบหน้าโดยใช้อัลกอริทึมแบบแพร่ กลับของโครงข่ายประสาทเทียม ระบบนี้ถูกพัฒนาขึ้นโดยใช้ภาษา C# และชุดคำสั่งของ EmguCV พร้อมติดตั้งกล้องเว็บแคมสำหรับการจับภาพใบหน้าบุคคลที่มาอยู่ตรงหน้าแล้วนำภาพที่ได้ขึ้นมาผ่านกระบวนการประมวลผลภาพดิจิทัลเพื่อเตรียมภาพให้เป็นตัวแปรอินพุตสำหรับโครงข่ายประสาทเทียมเพื่อตรวจสอบ ถ้าภาพใบหน้าบุคคลนั้นตรวจสอบแล้วไม่พบในฐานข้อมูลก็จะทำการสนทนาเพื่อต้องการทราบรายละเอียดของบุคคลนั้น โดยเฉพาะชื่อซึ่งจะใช้เป็นตัวแปรทาร์เก็ตเพื่อนำไปใช้ฝึกสอนให้ข่ายสมองจดจำชื่อและใบหน้าพร้อมทั้งรายละเอียดเก็บไว้ในฐานข้อมูลสำหรับบุคคลที่รู้จักแล้วต่อไป

3. วิธีการดำเนินการวิจัย

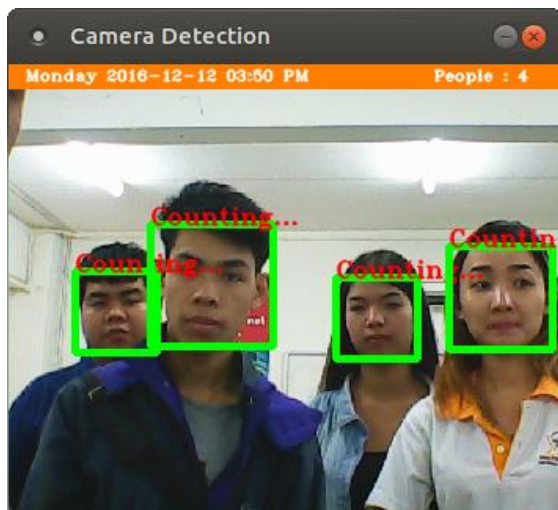
- 1) ลงพื้นที่เก็บรวบรวมข้อมูลตารางเดินรถไฟฟ้า, เส้นทาง, ป้ายรถไฟฟ้า
- 2) ศึกษาวิธีการเกี่ยวกับการตรวจจับใบหน้า (Face Detection)

- 3) ออกแบบและพัฒนาระบบ ซึ่งเขียนด้วยภาษา Python ลงบน Raspberry PI โดยเชื่อมต่อกับกล้อง Web Cam โดยระบบสามารถจับภาพใบหน้าบุคคล (Face detection) แล้วนำมาคำนวณว่าใบหน้าที่ถูกกล้องจับได้นั้นมีจำนวนกี่คน เพื่อนำไปวางแผนการเดินทาง
- 4) ออกแบบและพัฒนาเว็บไซต์(Responsive Website)
- 5) ทดสอบการทำงานของโปรแกรมและเว็บไซต์ เพื่อตรวจสอบความถูกต้อง และแก้ไขจุดบกพร่อง

4. ผลการวิจัย

4.1. ผลการออกแบบโปรแกรมตรวจจับใบหน้า

ระบบที่พัฒนาด้วย Python-OpenCV เพื่อตรวจจับใบหน้าบน Terminal ของเครื่อง Raspberry pi ที่ถูกติดตั้งไว้ที่ป้ายรถไฟฟ้า ก็จะแสดงภาพถ่ายวิดีโอที่ถูกถ่ายด้วยกล้องเว็บแคมตั้งรูปที่ 2 จำนวนคนที่นับได้นั้นจะส่งข้อมูลจำนวนคน ไปยัง Web Server

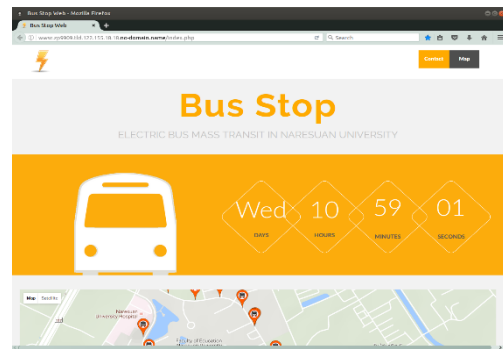


รูปที่ 2: หน้าจอ GUI ของโปรแกรมตรวจจับใบหน้า

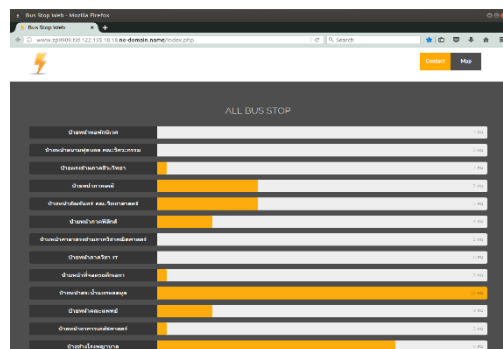
รูปที่ 2 เมื่อกล้องที่ติดตั้งที่ป้ายรถไฟฟ้า ตรวจพบใบหน้าผู้โดยสารที่กำลังรอลง จะทำการตีกรอบใบหน้าและแสดงข้อความ counting และจำนวนคนที่นับได้ด้านบนขวาของจอเพื่อให้ทราบว่าคุณนับแล้ว

4.2. ผลการออกแบบเว็บไซต์แสดงผล

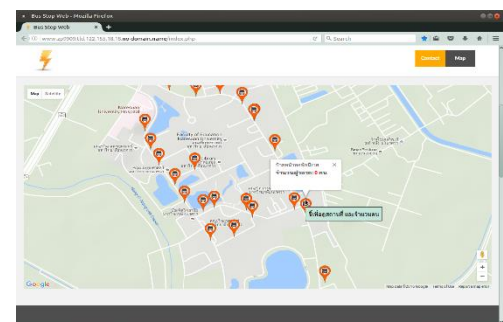
ส่วนแสดงผลจะแสดงผลในรูปแบบของเว็บไซต์ แบบ Responsive Website โดยดึงข้อมูลจำนวนคนมาจาก web server นำมาแสดงผลบน Marker ของ Google Maps เมื่อผู้ใช้เมาส์ไปยัง Marker จะแสดงชื่อป้าย และจำนวนคนที่รออยู่ในเวลานั้นๆ ดังแสดงในรูปที่ 3



(ก)



(ข)



(ค)



(ง)

รูปที่ 3 หน้าจอแสดงผลในส่วนเว็บไซต์

รูปที่ 3 ส่วนแสดงผลในเว็บไต์ (ก) ส่วนแสดงผลในเว็บไซด์เมื่อเข้าใช้งาน (ข) แสดงจำนวนผู้โดยสารแต่ละป้ายจอดรถไฟฟ้า (ค) แสดงแผนที่ตำแหน่งป้ายรถไฟฟ้าด้วย Google map API (ง) แสดงผลได้ในอุปกรณ์ที่หลากหลาย

4.3. อภิปรายผล

งานวิจัยนี้สามารถนำมาใช้งานได้จริง มีความแม่นยำในการตรวจจับใบหน้า (Face Detection) ประมาณ 80% จากการทดลอง อีก 20% จะเป็นข้อจำกัดในงานวิจัยนี้คือ ปัจจัยทางด้านแสงสว่าง, มุมอับของกล้อง, ความกว้างของเลนส์กล้อง, ความคมชัด และความละเอียดในการจับภาพของกล้อง รวมถึงคนที่ใส่เสื้อหน้ากาก หรือโปสเตอร์รูปหน้าคนที่มีความคล้ายคลึงกับหน้าของคนจริงๆ จะทำให้โปรแกรมสามารถตรวจจับได้ แต่ใบหน้าของสัตว์ต่างๆ ไม่สามารถตรวจจับได้ จึงทำให้เหมาะแก่การนำมาใช้งานตรวจจับใบหน้าคน เพื่อนับจำนวน

จากการที่ผู้ศึกษาได้หาข้อมูลปัญหาเกี่ยวกับการใช้บริการรถไฟฟ้าภายในมหาวิทยาลัยพบว่าปัญหาอันดับแรกๆ คือ รถไฟฟ้าไม่เพียงพอต่อผู้โดยสาร รัคนเป็นจำนวนมากทำให้มีคนต้องยืนเกาะประตูรถไฟฟ้าเพื่อเดินทาง และรถไฟฟ้าเดินทางมารับผู้โดยสารช้าทำให้ผู้โดยสารต้องรอนานเป็นต้น สาเหตุเกิดจากการที่ไม่ทราบจำนวนผู้โดยสารในแต่ละป้าย หรือไม่ทราบว่าป้ายไหนมีผู้โดยสาร ป้ายไหนไม่มีผู้โดยสารทำให้ต้องเดินรถให้ครบทุกป้าย จึงทำให้มารับผู้โดยสารล่าช้า ปัญหาเหล่านี้ทำให้ผู้ใช้บริการรถไฟฟ้าน้อยลง งานวิจัยนี้จึงสามารถช่วยลดปัญหาความแออัดและความไม่สะดวกสบายบนรถไฟฟ้าได้

5. สรุปผลงานวิจัย

งานวิจัยนี้เป็นการออกแบบและพัฒนาระบบการเดินรถไฟฟ้าให้สอดคล้องกับจำนวนคนที่รอในป้ายรถเมล์ โดยใช้เทคนิคการตรวจจับใบหน้าผ่านกล้องเว็บแคม พัฒนาโดยใช้ภาษา python ร่วมกับ library OpenCV ในการเขียนโปรแกรมตรวจจับใบหน้า (Face Detection) บนบอร์ด Raspberry PI3 และออกแบบการแสดงผลเป็นแบบ Responsive Website ที่สามารถแสดงได้หลาย Device เพื่อให้ผู้ที่ต้องการทราบจำนวนผู้โดยสารสามารถดูได้ ไม่ว่าจะใช้อุปกรณ์แบบใดก็ตาม ผู้ควบคุมการเดินรถไฟฟ้าสามารถเปิดเว็บไซต์ เพื่อดูจำนวนคนในแต่ละป้ายได้ว่ามีผู้รอรถเป็นจำนวนเท่าไร ซึ่งการแสดงผลนั้นจะเป็นแบบ Google Map เมื่อผู้ใช้ชี้ที่ Marker จะแสดงสถานที่ป้าย, ชื่อป้าย และจำนวนผู้รอรถไฟฟ้าในป้ายแบบ Real-time พร้อมแสดงภาพรวมเป็นกราฟแสดงให้เห็นว่า ป้ายหรือเส้นทางไหนมีจำนวนผู้รอรถมากที่สุด ทำให้สามารถจัดรถไปรับผู้รอรถในแต่ละป้ายได้เพียงพอช่วยลดปัญหาความแออัดและความไม่สะดวกสบายบนรถไฟฟ้าได้ อีกทั้งยังช่วยประหยัดพลังงานในการเดินรถของรถไฟฟ้าได้ โดยเดินรถในเส้นทางที่มีผู้รอรถในป้าย

เอกสารอ้างอิง

- [1] นายกรกริช ดิเรกสุนทร และ นางสาวศศินันท์ โคมเดือน. (2555). ระบบแจ้งเตือนผู้บุกรุก *Intruder warning System*. สืบค้นเมื่อ 30 มีนาคม 2559. จาก http://www.research.eng.ku.ac.th/index.php?option=com_phocadownload&view=category&download=71:intruder-warning-system&id=17:2553elec&Itemid=55&start=60
- [2] สนั่น ศรีสุข, คารณ, สุนติและวีระศักดิ์, ครูธัช. (2541). *Learning OpenCV Computer Vision with the OpenCV Library*, OREILLY, Gary Bradski & Adrian Kaebler *Face Detection using Eigen Face and Correlation Analysis*. สืบค้นเมื่อ 30 มีนาคม 2559. จาก <http://www.shervinemami.co.cc/faceRecognition.html>
- [3] mns-smartpro. (2558). สืบค้นเมื่อ 20 พฤษภาคม 2559. จาก <http://www.mns-smartpro.com/Blog/วิเคราะห์ใบหน้า-blog.aspx>
- [4] ว่าที่ร้อยตรีหญิง ธนิสรา บุตรสิงขรณ์. (2555). วิธีฮิวริสติกสำหรับการจัดตารางเดินรถประจำทาง *A HEURISTIC METHOD FOR BUS SCHEDULING*. สืบค้นเมื่อ 30 มีนาคม 2559. จาก <http://203.158.6.11:8080/sutir/bitstream/123456789/4149/2/fulltext.pdf>
- [5] นาย วิรัตน์ พรหมเมศ และ นายนพพล เลิศชูวงศา. (2558). ระบบควบคุมความปลอดภัยบน raspberry pi ด้วยเทคนิค Eigen face และการปรับเปลี่ยนสถานะแสง *Security controlled system on raspberry pi with Eigen face technique and illumination change*. สืบค้นเมื่อ 5 เมษายน 2559. จาก http://manage.nstru.ac.th/pinsorn/paper/ECTI_CARD_2015_opf_files/1081-ossbss-camera_ready-1.pdf
- [6] วุฒิชัย ปวงมณี และชจรศักดิ์ คันธพนิต. (2557). การพัฒนาระบบรู้จำใบหน้าบุคคลโดยใช้โครงข่ายประสาทเทียม *Development of the Human Face Recognition System using the Artificial Neural Network*. สืบค้นเมื่อ 5 เมษายน 2559 . จาก <http://nctechd.org/NCTechEd07/NCTechEd07SIT03.pdf>