

การตรวจจับตัวอักษรป้ายทะเบียนรถยนต์ด้วยการเรียนรู้เชิงลึก เพื่อความปลอดภัยของผู้อยู่อาศัย

CAR LICENCE PLATES' LETTER DETECTION WITH DEEP LEARNING FOR RESIDENTS'SAFETY

วรุฒ พลอยเลี้ยง¹

ผศ. ดร.ชัยพร เขมะภาคะพันธ์²

บทคัดย่อ

การวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาวิธีปรับปรุงกระบวนการตรวจจับตัวอักษรป้ายทะเบียนรถยนต์ด้วยการเรียนรู้เชิงลึกให้มีประสิทธิภาพมากขึ้น เพื่อความปลอดภัยของกลุ่มเป้าหมาย โดยใช้การเรียนรู้เพื่อตรวจสอบภาพป้ายทะเบียนรถยนต์ เพื่อนำมาประยุกต์ใช้ในการเข้า-ออกรถยนต์ในพื้นที่ ที่ได้จำกัดไว้ นำมาเป็นฐานในการศึกษาในการวิจัยครั้งนี้ การวิจัยนี้สามารถนำไปประยุกต์ ใช้ในงานวิจัยอื่นๆได้อีกด้วย

โดยงานวิจัยนี้ใช้อุปกรณ์ ใช้ Raspberry Pi ในการทำงานวิจัยครั้งนี้ เพื่อนำมาเป็นตัวคำนวณโปรแกรมและนำมาพัฒนาซอฟต์แวร์ Python ในการคำนวณ Image processing ป้ายทะเบียนรถยนต์ และนำอุปกรณ์ต่างๆ มาเชื่อมต่อกัน เช่น Barrier Gate (ที่กั้นรถยนต์) และ สัญญาณไฟต่างๆ การตรวจจับป้ายทะเบียน ใช้ Pytesseract เพื่อแปลงสตริงเป็นข้อความ(ตัวเลข) และนำมาบันทึกข้อมูลลงหน่วยความจำ โดยการเก็บข้อมูลในฐานข้อมูล เพื่อนำมาเป็นการประวัติการเข้า-ออก วัน-เวลา และมีการลงทะเบียนรถยนต์ ของผู้ที่มีสิทธิ์ต้องมาลงทะเบียนกับผู้ดูแลเท่านั้น ผลการทดลองการทำงานของการทำงานของป้ายทะเบียนรถยนต์ 20 คัน มีผลเฉลี่ย 90 เปอร์เซ็นต์ ที่มีความถูกต้อง อีก 10เปอร์เซ็นต์ ที่เหลือ ที่มีความผิดพลาดเกิด จาก 1) แสงสว่างไม่เพียงพอ 2) ป้ายทะเบียนรถยนต์ มีตัวอักษร ที่ขาดหาย 3) ระยะการโฟกัสไกลไปหรือใกล้ไป 4) ตัวอักษรมีความใกล้เคียงกัน ถ้าสามารถลดความผิดพลาดได้ ก็สามารถถูกต้อง ประมาณ 90 เปอร์เซ็นต์ จึงสรุปได้ว่ากระบวนการตรวจจับตัวอักษร มีความคาดเดาได้สูงมาก แล้วสามารถนำไปพัฒนาในด้านอื่นๆ ได้สามารถลดเวลาการเข้า-ออก ในการจอดรถ เพื่อความปลอดภัยและเก็บประวัติ

¹ นักศึกษาหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม

² ที่ปรึกษาสารนิพนธ์หลัก

ABSTRACT

The purpose of this research is to study how to enhance the process of detecting license plate letters using deep learning, to improve safety for the target group. The focus is on utilizing learned techniques to analyze license plate images for use in controlling vehicle entry and exit within confined spaces. This research has potential applications in other related studies.

This research utilizes the Raspberry Pi device as a programmable calculator to develop Python software for image processing, specifically for license plate recognition. It also facilitates the connection of various devices such as barrier gates (car barriers) and different light signals. License plate detection involves the use of Pytesseract to convert strings into text (numbers) and save the data into memory. Furthermore, the data is stored in a database to track entry-exit history, daytime activity, and vehicle registration. Only eligible individuals are required to register with the caretaker.

The results of the 20-car license plate detection test show an average effectiveness of 90 percent with an additional 10 percent accuracy. The remaining errors are attributed to: 1) insufficient lighting, 2) missing characters on the car license plate, 3) incorrect focus distance (either too far or too close), and 4) similarity among characters. By addressing these issues, it is possible to reduce the error rate by approximately 90 percent. Hence, it can be concluded that the character detection process is highly predictable and can be applied in various other domains. This advancement could potentially reduce entry-exit times in parking lots for improved safety and historical recordkeeping.

1. บทนำ

ในปัจจุบันยุคสมัยได้เปลี่ยนแปลงไปอย่างรวดเร็วจากความก้าวหน้าทางเทคโนโลยี ก่อให้เกิดการพัฒนานวัตกรรมและสิ่งประดิษฐ์ทางปัญญาใหม่ๆ ทั้งในด้านอุตสาหกรรม ด้านเกษตรกรรม และด้านธุรกิจ นอกจากนี้ยังมีการบูรณาการความรู้ในองค์ประกอบต่างๆ เพื่อเป็นแรงผลักดันให้ประเทศขับเคลื่อนเข้าสู่ยุค 4.0 โดยประเทศไทยได้มีนโยบายทั้งภาครัฐและเอกชนสร้างสรรค่นวัตกรรม เพื่อพัฒนาประเทศ โดยริเริ่มโครงการเมืองอัจฉริยะ ซึ่งถือเป็นโครงการที่กระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคมจัดขึ้น ด้วยเป้าหมายที่จะพัฒนาคนให้กลายเป็นพลังขับเคลื่อนที่สำคัญของประเทศ และเพื่อเพิ่มความสะดวกสบาย รวดเร็วในการดำเนินชีวิตให้ทันต่อการเปลี่ยนแปลงไปของโลกและสังคมเมืองในสภาพในปัจจุบันที่ต้องการด้านความ

รวดเร็ว และความถูกต้องแม่นยำสามารถตรวจสอบข้อมูลย้อนกลับได้ และจำเป็นต้องมีความปลอดภัยอีกด้วยเช่นกัน

จากคำกล่าวในข้างต้น ทางผู้จัดทำเห็นโอกาสในการทำระบบที่จอดรถอัจฉริยะในสถานที่ส่วนบุคคลต่างๆ เช่น หมู่บ้าน คอนโดมิเนียม ชุมชนที่พักอาศัยที่ต้องการความปลอดภัย โดยใช้เทคนิควิธีการตรวจจับป้ายทะเบียนและตรวจสอบกับระบบฐานข้อมูลว่าทะเบียนรถคันดังกล่าวได้ลงทะเบียนไว้หรือไม่ และสามารถตรวจสอบระยะเวลาในการเข้า-ออกของรถคันดังกล่าวได้อีกด้วย ช่วยทำให้ลดความผิดพลาดจากตัวบุคคล และให้ระบบมีความแม่นยำมากขึ้น มีความสะดวก รวดเร็ว และเพิ่มความปลอดภัยในพื้นที่อยู่อาศัย หรือพื้นที่ชุมชนที่ต้องการการเป็นส่วนตัวได้ดียิ่งขึ้น และจากการทดสอบระบบจากตัวอย่าง 10 ตัวอย่าง มีความถูกต้องอยู่ที่ร้อยละ 90

วัตถุประสงค์งานวิจัย

1. เพื่อพัฒนาการประยุกต์ใช้เทคนิคการตรวจจับป้ายทะเบียน สำหรับพัฒนาระบบตรวจจับและรู้จักป้ายทะเบียนรถยนต์
2. เพื่อทดสอบประสิทธิภาพของเทคนิคที่ใช้ในการพัฒนาระบบตรวจจับและรู้จักป้ายทะเบียนรถยนต์ ภายใต้สภาพแวดล้อมต่างๆ
3. เพื่อเป็นแนวทางระบบจอดรถอัตโนมัติ และ ระบบที่เกี่ยวข้องได้

ขอบเขตงานวิจัย

1. ระบบทำงานในการตรวจจับป้ายทะเบียนแบบ (Real Time)
2. ระบบสามารถตรวจจับป้ายทะเบียนภายใต้สภาวะที่มีความเข้มของแสงในเวลากลางวันและกลางคืน
3. การพัฒนาระบบจะแบ่งเป็น 2 ส่วนคือ การประมวลผลข้อมูลภาพจากกล้องวงจรปิด โดย Raspberry Pi และการเก็บฐานข้อมูล

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ระบบตรวจจับป้ายทะเบียนรถยนต์ และตรวจสอบประมวลผลกับฐานข้อมูลได้
2. สามารถนำระบบที่พัฒนา ไปต่อยอด เพื่อตอบรับโครงการ Smart City
3. ได้เครื่องต้นแบบที่สามารถนำไปพัฒนาต่อ เข้าระบบเก็บเงินค่าจอดรถอัตโนมัติได้

2. แนวคิดและทฤษฎีที่เกี่ยวข้อง

ราสเบอร์รี่พาย (Raspberry Pi) เป็นหนึ่งในความนิยมมากที่สุดในโลก บอร์ดขนาดเล็ก เครื่องจักรสามารถทำเกือบทุกอย่างที่คอมพิวเตอร์เดสก์ท็อปสามารถทำได้สำหรับทุกคนที่สนใจในการสำรวจคอมพิวเตอร์ ราสเบอร์รี่พาย (Raspberry Pi) มาพร้อมกับซอฟต์แวร์ ระบบปฏิบัติการเป็นโปรแกรมที่สำคัญที่สุดที่จะช่วยให้พัฒนาและรัน โปรแกรมช่วยให้ฮาร์ดแวร์เพื่อสื่อสารกับซอฟต์แวร์เพื่อสร้างปฏิสัมพันธ์ที่มีความหมาย นอกจากนี้ยังจัดการทรัพยากรหน่วยความจำใดก็ได้ที่เครื่องพิมพ์สร้างส่วนติดต่อผู้ใช้และให้บริการซอฟต์แวร์ประยุกต์ ระบบปฏิบัติการที่สามารถเรียกใช้ใน ราสเบอร์รี่พาย (Raspberry Pi) เดิมเรียกว่า ราสเบียน (Raspbian) เป็นระบบปฏิบัติการอย่างเป็นทางการสำหรับไมโครคอมพิวเตอร์เครื่องนี้ ลินุกซ์ Distro ขึ้นอยู่กับ Debian และนำทุกสิ่งที่จำเป็นเพื่อให้ได้รับประโยชน์สูงสุดจากไมโครพีซีเครื่องนี้ Distro นี้ได้รับการปรับให้เหมาะกับการทำงานบนคอมพิวเตอร์ ARM และมาพร้อมกับแพ็คเกจและโปรแกรมที่ติดตั้งไว้ก่อนหน้านี้จำนวนมาก ราสเบอร์รี่พาย โอเอส (Raspberry Pi OS) ใช้ เดสก์ท็อป PIXEL (สภาพแวดล้อม X-Window ที่ปรับปรุงแล้ว, น้ำหนักเบา) ที่ใช้ LXDE ซึ่งมีประโยชน์และมีน้ำหนักเบา

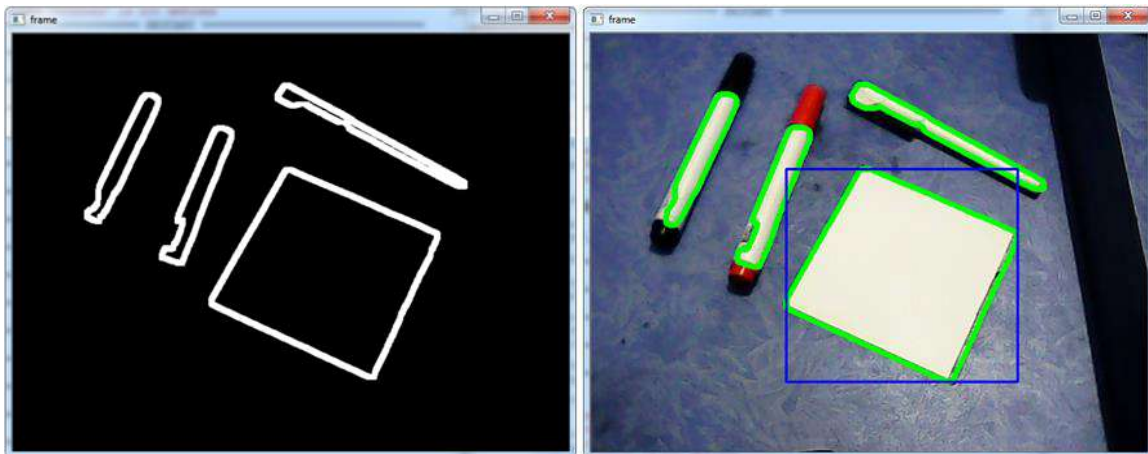
Opencv ในไพทอน จะเขียนขึ้นจากภาษาซีและทำงานในภาษาซีเป็นหลัก แต่สามารถเรียกใช้ผ่านภาษาโปรแกรมระดับสูงที่เขียนง่ายๆเช่นไพทอนได้ การเขียนในไพทอนนั้นข้อดีก็คือเขียนง่ายกว่าเขียนในภาษาซีโดยตรง แม้จะมีข้อเสียคือทำงานช้าไปบ้าง แต่การทำงานของคำสั่งต่างๆใน Opencv นั้นโดยหลักๆแล้วจะไปเรียกใช้ฟังก์ชันซึ่งเขียนในภาษาซี ไม่ได้เขียนด้วยไพทอนโดยตรง เมื่อการทำงานเบื้องหลังเป็นภาษาซีจึงทำงานรวดเร็ว ไม่ต้องห่วงว่าใช้ในไพทอนแล้วจะช้าลงกว่าใช้ในภาษาซีโดยตรงมากนัก แต่ก็เพราะ Opencv ยื่นพื้นจากภาษาซีเป็นหลัก จึงทำให้มีลักษณะการใช้งานที่ออกจะเป็นสไลด์ภาษาซี แม้จะใช้ผ่านไพทอนก็ตาม ซึ่งคนที่เขียนไพทอนเป็นหลักอาจรู้สึกไม่คุ้นชินอยู่บ้าง อีกทั้งภาษาไพทอนมีโมดูลอื่นๆหลากหลายซึ่งสามารถนำมาใช้ร่วมกันเพื่อทำงานได้ยืดหยุ่นกว้างขวางขึ้น

Image Processing การประมวลผลภาพเพื่อที่จะได้ค่าวิเคราะห์นั้นประกอบด้วยหลายตัวแปรหรือหลายปัจจัย ด้วย โดยที่มีความสำคัญที่สุดก็คงหนีไม่พ้นเรื่องของสี ที่จะเป็นตัวบอกความแตกต่างระหว่างพื้นหลัง และตัววัตถุ หรือรูปทรงต่างๆ (Contour) และอันดับต่อมาคือ หน่วยของภาพ หรือ พิกเซล ที่มีผลอีกเช่นกัน ในเรื่องของความละเอียด ซึ่งยิ่งละเอียดมากก็จะสามารถแยกส่วนของข้อมูลภาพได้ถูกต้องแม่นยำยิ่งขึ้น และแสงก็มีความสำคัญไม่แพ้กัน ต่อให้ภาพมีความละเอียดสูงแต่ถ้าแสงน้อยเกินไปอาจจะทำให้วิเคราะห์ไม่ตรงกับความจริงเลย ดังนั้น ปัจจัยพื้นฐานที่จะได้ข้อมูลจากภาพเพื่อนำไปวิเคราะห์ได้อย่างแม่นยำนั้นได้แก่ ปริมาณของพิกเซล และแสง ซึ่งมีรายละเอียด ดังนี้

Median Blur เพื่อจำกัดแสงเงาและ สิ่งรบกวนต่างๆ ทำให้ภาพ Smooth ขึ้น จากนั้นแล้ว Thresholding จากนั้นนำภาพขาวดำไปทำ Dilation แล้วจึงนำภาพนั้นไปหาพิกัดของ นำไป Crop เพื่อเป็น Input ต่อไป

ภาพเบลอปรับภาพให้เรียบ การทำให้ภาพเบลอทำได้โดยการรวมภาพด้วยเคอร์เนลตัวกรองความถี่ต่ำ มีประโยชน์ในการขจัดเสียงรบกวน ที่จริงแล้วจะลบเนื้อหาที่มีความถี่สูง (เช่น สัญญาณรบกวน ขอบ) ออกจากภาพ ทำให้ขอบเบลอเมื่อใช้ฟิลเตอร์นี้ มีเทคนิคการเบลอขอบที่ไม่เบลอ OpenCV มีเทคนิคการเบลอประเภทหลักๆ ค่าเฉลี่ย สิ่งนี้ทำได้โดยการทำให้ภาพเป็นวงกลมด้วยฟิลเตอร์กล่องมาตรฐาน เพียงแค่ใช้ค่าเฉลี่ยของพิกเซลทั้งหมดภายใต้พื้นที่เคอร์เนลและแทนที่องค์ประกอบกลางด้วยค่าเฉลี่ยนี้

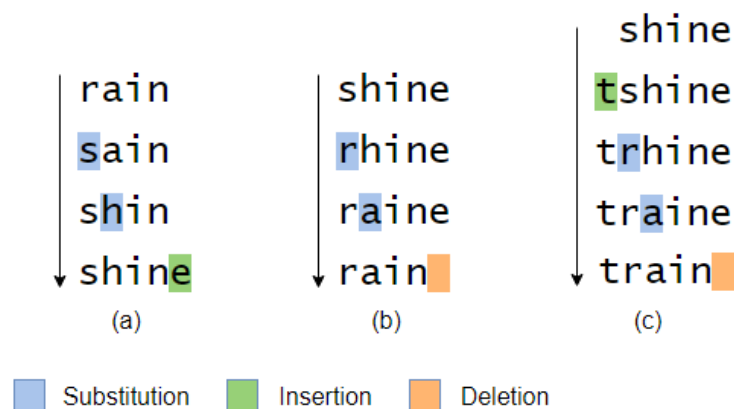
การตรวจจับรูปร่างสี่เหลี่ยม จะแยกวัตถุผ่านการวิเคราะห์รูปร่างหรือวิธีการตรวจสอบรูปร่าง ด้วยการ ใช้ OpenCV ฟังก์ชัน findcontour การหารูปร่างสี่เหลี่ยมโดยใช้ `=cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)` `cv2.findContours` ได้ส่งตัวแปรเอาต์พุต ที่สามารถหารูปร่างสี่เหลี่ยมได้



ภาพที่ 1 ทดสอบการ Crop ภาพ

Levenshtein Distance จากคำสองคำ สามารถถามได้ว่าคำสองคำนี้มีความคล้ายคลึงกันมากน้อยเพียงใด ยังสามารถถามคำถามสองประโยคหรือลำดับสตริง ในการหาปริมาณความคล้ายคลึงกัน จำเป็นต้องวัด ระยะทาง Levenshtein เป็นมาตรการดังกล่าว ด้วยวิธีการง่ายๆ (เช่น การแทรก การลบ และการแทนที่) สามารถกำหนดวิธีแปลงคำหรือลำดับคำหนึ่งเป็นคำหรือลำดับอื่นได้ อาจมีหลายวิธีในการบรรลุเป้าหมายนี้ ระยะทาง Levenshtein ถูกกำหนดเป็นจำนวนการดำเนินการขั้นต่ำที่จำเป็นเพื่อให้อินพุตทั้งสองมีค่าเท่ากัน ตัวเลขที่ลง อินพุตสองตัวที่กำลังเปรียบเทียบยังมีความคล้ายคลึงกันมากเท่านั้น มีอัลกอริทึมสองสามข้อในการแก้ปัญหาระยะทางนี้ Levenshtein Distance มีประโยชน์ในโดเมนแอปพลิเคชันมากมาย รวมถึงการประมวลผลสัญญาณ การประมวลผลภาษาธรรมชาติ และชีววิทยาการคำนวณ Levenshtein Distance เรียกอีกอย่างว่าระยะแก้ไข

Could you explain Levenshtein Distance with an example?



ภาพที่ 2 Levenshtein Distance จำลองการประมวลผล

Whitelisting and Blacklisting Characters with Tesseract and Python เป็นตัวเลขหรือตัวอักษร จากภาพนำเข้า เป็นกระบวนการควบคุมและกรองตัวอักษรดังกล่าว ตัวอย่างเช่น เมื่อนำภาพนำเข้า ต้องการแยกตัวเลขและตัวอักษรอื่น ๆ และตัวอักษรพิเศษ เช่นเครื่องหมายดอลลาร์เครื่องหมายทศนิยม เช่น จุด และเครื่องหมายจุลภาค ในการควบคุมรายละเอียด ของข้อมูลได้อย่างถูกต้อง



ภาพที่ 3 ภาพ Code ของ Levenshtein Distance

3. ระเบียบวิธีวิจัย

วิธีการดำเนินงานวิจัยระบบตรวจจับป้ายทะเบียนรถยนต์จะแบ่งหัวข้อวิธีการดำเนินงานวิจัยออกเป็น 5 กระบวนการดังนี้

- 1 การออกแบบระบบฮาร์ดแวร์
- 2 การออกแบบระบบซอฟต์แวร์
- 3 การเก็บภาพข้อมูลตัวอย่าง
- 4 การสร้างชุดข้อมูลเรียนรู้
- 5 การตรวจจับป้ายทะเบียนและเรียนรู้จากตัวอักษร

ระบบตรวจจับตัวอักษรป้ายทะเบียน เป็นระบบแอปพลิเคชัน (Pytesseract) ที่ติดตั้ง บนเครื่องแม่ข่าย โดยใช้ .NET Framework ในการพัฒนา โดยระบบจะทำการโหลดข้อมูลกล้อง จากฐานข้อมูล ซึ่งระบุถึงรหัส ชื่อสถานี่ ละจุดตรวจสอบของสถานี่นั้นๆ เพื่อทำการเชื่อมต่อ ด้วยโอเพนซีวี (Pytesseract) และดึงภาพจากกล้องในแต่ละสถานี่เพื่อนำมาประมวลผล ตัวอักษรป้ายทะเบียนรถยนต์

การจัดเก็บฐานข้อมูล ระบบตรวจจับป้ายทะเบียนที่ติดตั้งบนเครื่องแม่ข่าย จะทำการประมวลผลภาพที่ได้รับ จากกล้องที่ติดตั้งในแต่ละสถานี แล้วจับเก็บข้อมูลลงสู่ระบบฐานข้อมูลที่ติดตั้งบนเครื่องแม่ข่าย ซึ่งมีข้อมูลเกี่ยวกับ ภาพป้ายทะเบียน ข้อมูลป้ายทะเบียน วันที่และเวลาในการตรวจจับ ข้อมูลละติจูดลองจิจูด และรายละเอียดของกล้องที่ติดตั้งในแต่ละสถานีเข้าทำการร้องขอ

ระบบตรวจจับป้ายทะเบียน

ระบบตรวจจับป้ายทะเบียนเป็นระบบแอปพลิเคชัน (Pytesseract) ที่ติดตั้ง บนเครื่องแม่ข่าย โดยใช้ .NET Framework ในการพัฒนา โดยระบบจะทำการโหลดข้อมูลกล้อง จากฐานข้อมูล ซึ่งระบุถึง รหัส ชื่อสถานี ละติจูดลองจิจูด ของสถานีนั้นๆ เพื่อทำการเชื่อมต่อ ด้วยโอเพนซีวี (Pytesseract) และดึงภาพจากกล้องในแต่ละสถานีเพื่อนำมาประมวลผล โดยตัวระบบสามารถเลือกเฉพาะกล้องที่สนใจ หรือกล้องทั้งหมดที่ติดตั้งได้ในการเปิดใช้งาน

การจำแนกพื้นที่บริเวณสัญลักษณ์บนหน้ายา

เป็นขั้นตอนหาตำแหน่งที่แท้จริงของสัญลักษณ์บนหน้ายาเพื่อให้ได้ภาพเฉพาะส่วนที่เป็นพื้นที่สัญลักษณ์บนหน้ายา เพื่อลดปัญหาที่เกิดจากอิทธิพลของค่าสีพิกเซลที่ไม่เกี่ยวข้องสำหรับขั้นตอนการสร้างภาพขาวดำโดยเฉพาะ

เริ่มจากการหาขอบสัญลักษณ์บนหน้ายาโดยใช้เทคนิค Canny edge detection กับภาพเมดิคาที่ละช่องสี RGB หลังจากนั้นนำภาพผลลัพธ์การทำ Edge detection ของแต่ละช่องสีมารวมกันเป็นภาพผลลัพธ์ดังสมการ เนื่องจากขอบตัวอักษรที่ได้เป็นเส้น Morphology dilation เพื่อเพิ่มขนาดเส้นขอบตัวอักษร หลังจากเพิ่มขนาดขอบวัตถุพบว่ามีรูปภายในขอบวัตถุซึ่ง ก่อให้เกิดเส้นรอบวงของรูปที่ไม่พึงประสงค์ส่งผลให้ผลลัพธ์การหา Character contour ในขั้นตอนต่อไปแม่นยำน้อยลง ดังนั้นเพื่อกลบรูปเหล่านี้เลือกใช้เทคนิค Morphology closing ที่มี Kernel ทรงสี่เหลี่ยมจัตุรัสขนาดเล็กที่สุดที่เป็นไปได้ซึ่งก็คือขนาด 3*3 พิกเซลเพื่อก่อให้เกิดการประมวลผลน้อยที่สุด



ภาพที่ 4 ผลลัพธ์ Canny edge detection ในการหาตัวอักษรของยา

การหารูปสี่เหลี่ยมผืนผ้า

ลำดับต่อมาผู้วิจัยแปลงภาพโดยใช้การกระทำแบบขยาย (Dilation) แบ่งการขยายตามแนวแกน ขยายใน ระนาบแกน X กระทำ ขนาด 80×5 และขยายในระนาบแกน Y กระทำ ด้วย B ขนาด 5×80 จากนั้นทำการรวมแกน X, Y กระทำด้วย B ขนาด 5×30 ขยายพิกเซลไบนารีในลักษณะของรูปสี่เหลี่ยมผืนผ้า (Rectangle) เพื่อความเหมาะสมกับลักษณะของแผ่นป้ายทะเบียนรถยนต์

ขั้นตอนลำดับต่อมาผู้วิจัยแปลงภาพด้วยการกัดกร่อน (Erosion) ขั้นตอนดังกล่าวเป็นการลดสัญญาณรบกวนไบนารีวิเคราะห์ไบนารีโดยใช้ Library ของโปรแกรม Matlab พื้นที่ของภาพ (area) จุดกลางของวัตถุ (centroid) และตำแหน่งล้อมรอบวัตถุหรือเส้นอ้างอิงขอบภาพ (Bounding Box)

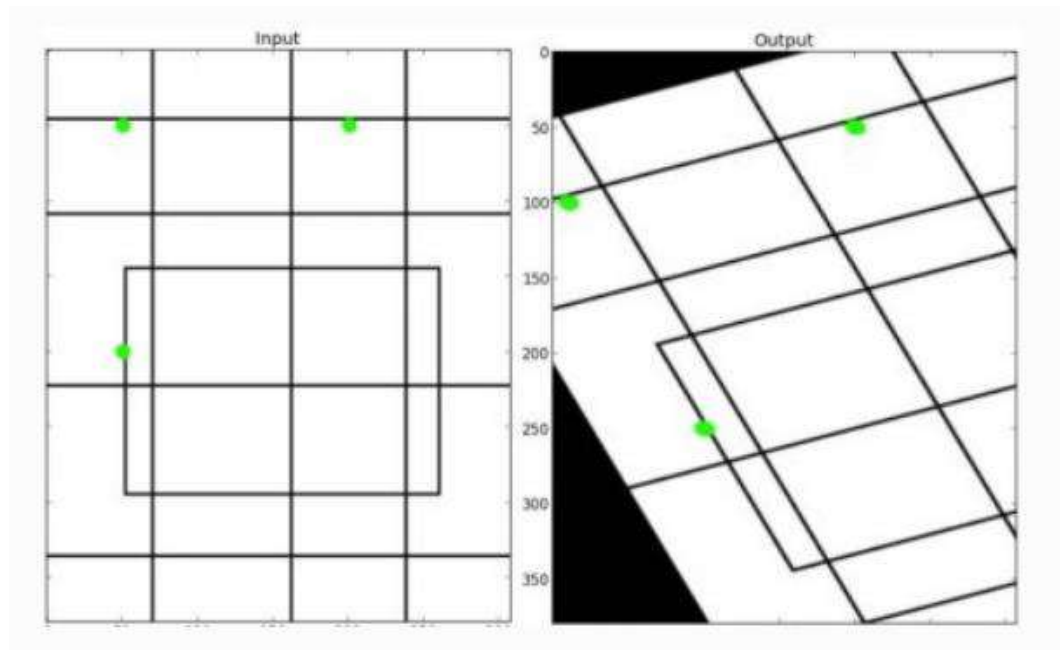


ภาพที่ 5 ผลลัพธ์ ของการหาป้ายทะเบียน เป็นรูปสี่เหลี่ยม

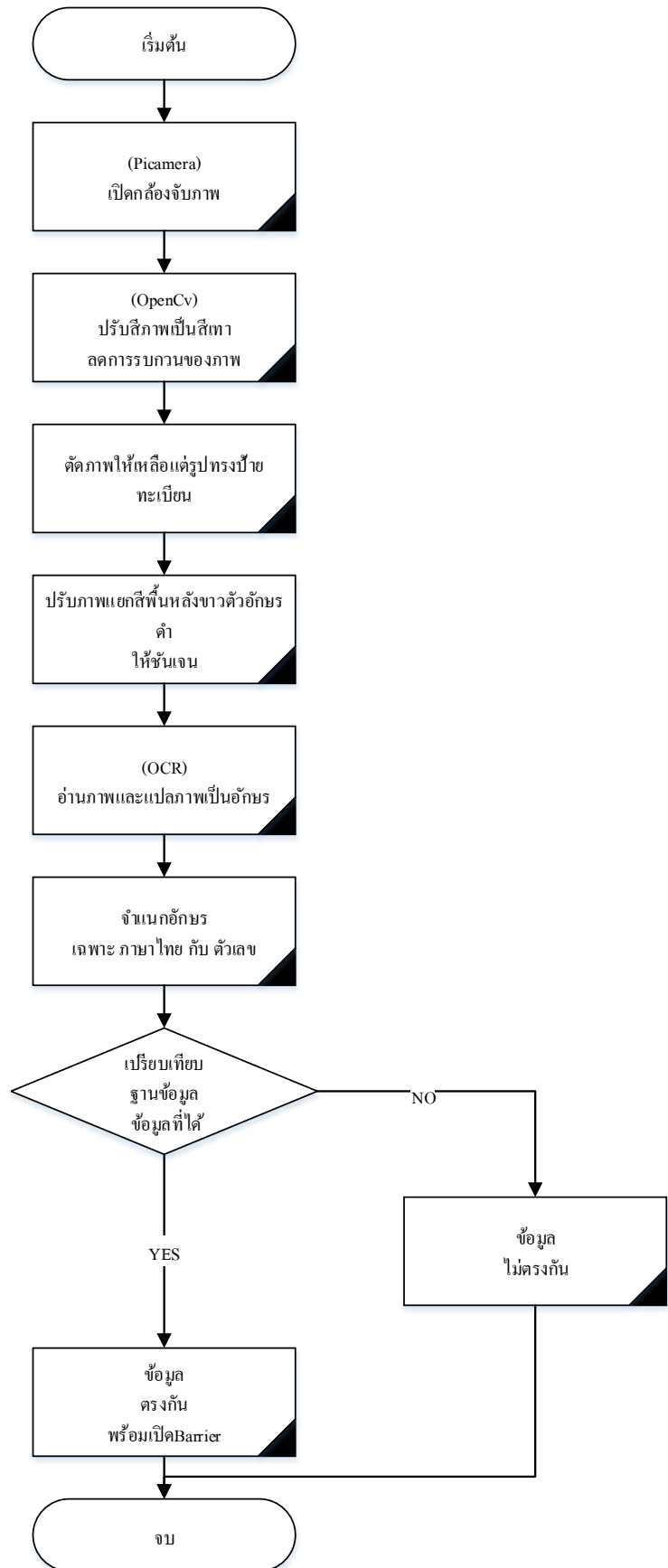
Geometric Transformations of Images

การใช้การหาเรขาคณิตสี่เหลี่ยมในภาพ กับรูปภาพ cv2.getPerspectiveTransform การเปลี่ยนแปลง OpenCV มีฟังก์ชันการแปลงสองฟังก์ชัน ได้แก่ cv2.warpAffine และ cv2.warpPerspective ซึ่งสามารถใช้การแปลงได้ทุกรูปแบบ cv2.warpAffine ใช้เมทริกซ์การแปลง 2x3 ในขณะที่ cv2.warpPerspective ใช้เมทริกซ์การแปลง 3x3 เป็นอินพุต Scaling เป็นเพียงการปรับขนาดภาพ OpenCV มาพร้อมกับฟังก์ชัน cv2.resize() เพื่อจุดประสงค์นี้ ขนาดของภาพสามารถระบุได้ หรือสามารถระบุตัวคูณมาตราส่วนได้ ใช้วิธีการแก้ไขที่แตกต่างกัน วิธีการแก้ไขที่แนะนำคือ cv2.INTER_AREA สำหรับการย่อและ cv2.INTER_CUBIC (ซ้ำ) & cv2.INTER_LINEAR สำหรับการซูม ตามค่าเริ่มต้น วิธีการแก้ไขที่ใช้คือ cv2.INTER_LINEAR สำหรับวัตถุประสงค์ในการปรับขนาดทั้งหมด สามารถปรับขนาดภาพที่ป้อนด้วย

Affine Transformation ในการแปลงเลียนแบบ เส้นคู่ขนานทั้งหมดในรูปภาพต้นฉบับจะยังคงขนานกันในรูปภาพที่เอาต์พุต ในการหาเมทริกซ์การแปลง ต้องการจุดสามจุดจากภาพอินพุตและตำแหน่งที่เกี่ยวข้องในภาพเอาต์พุต จากนั้น cv2.getAffineTransform จะสร้างเมทริกซ์ขนาด 2x3 ซึ่งจะส่งผ่านไปยัง cv2.warpAffine ตรวจสอบตัวอย่างด้านล่าง และดูจุดที่ฉันทเลือกด้วย ซึ่งมีเครื่องหมายเป็นสีเขียว



ภาพที่ 6 ผลลัพธ์ ของการหาป้ายทะเบียน เป็นรูปสี่เหลี่ยม



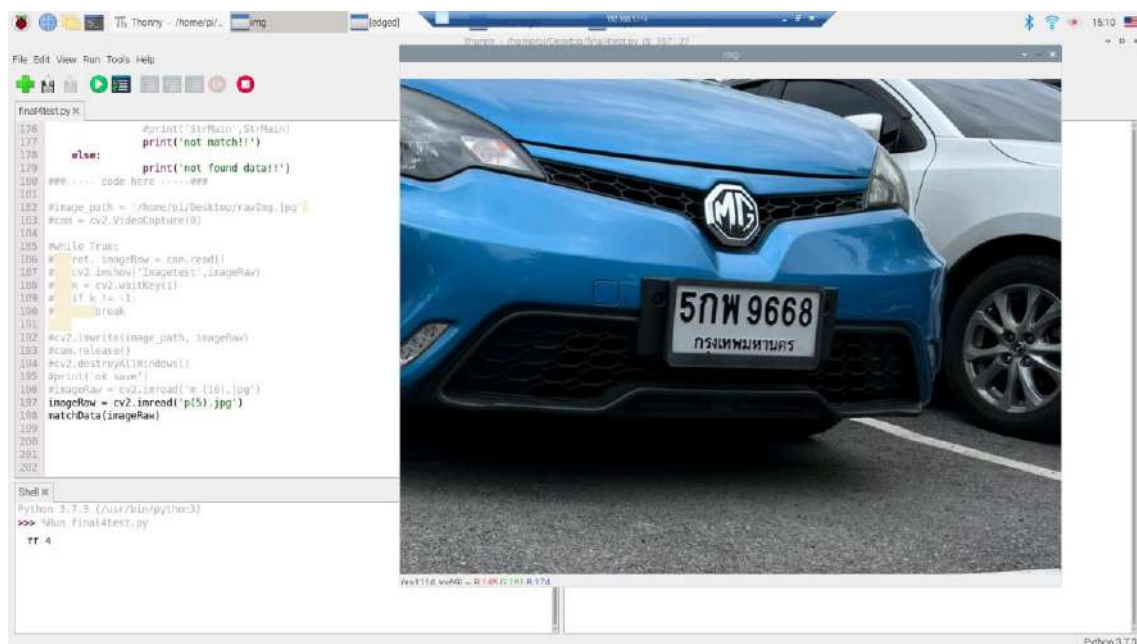
ภาพที่ 7 Flow การทำงานของระบบ

4. ผลการวิจัย

นำตัวอักษรและตัวเลขแต่ละตัวเปรียบเทียบกับแม่แบบ (Template Matching)
การนำรูปภาพที่ต้องการหาค่าไปเปรียบเทียบกับแม่แบบ (Template) ซึ่งรูปแบบตัวอักษรนี้จะเก็บค่า และกำหนดลักษณะสำคัญต่างๆ ที่สามารถแยกความแตกต่างของตัวอักษรต่างๆ ได้และเมื่อนำภาพตั้งต้นที่มีมาเปรียบเทียบกับรูปแบบที่มี เพื่อตรวจสอบความคล้ายคลึงกันของภาพทั้งสองซึ่งอาศัยค่าระดับที่ตั้งขึ้นมาเพื่อตรวจสอบหรือตัดสินใจ นั้นจำเป็นที่จะต้องมีการหาสองส่วนหลักๆ คือ แม่แบบและภาพต้นฉบับ โดยส่วนที่เรียกว่าแม่แบบนั้นเป็นส่วนจากรูปแบบของภาพ ซึ่งจะเก็บรูปแบบภาพทั้ง 54 ตัว คือเลข 0 ถึง 9 และ ตัวอักษร 44 ตัว ซึ่งอาจมีชุดของรูปแบบตัวเลขที่เพิ่มเข้ามาอีก ส่วนที่สองคือ ภาพต้นฉบับเป็นส่วนของชุดรูปภาพตัวเลขป้ายและตัวอักษรป้ายที่นำเข้าโปรแกรมเพื่อวิเคราะห์ผล

การนำเข้าภาพ (Input Image)

การนำเข้าภาพ จะแบ่งภาพออกเป็น 2 ประเภทคือ ภาพที่เป็นเป้าหมายในการตรวจจับ ซึ่งหมายถึงภาพป้ายทะเบียนรถยนต์ (Positive Image) และภาพที่ไม่ใช่เป้าหมายในการตรวจจับ (Negative Image) ซึ่งหมายถึงภาพอื่นๆที่ไม่ใช่ป้ายทะเบียนรถยนต์ เช่นภาพ สภาพแวดล้อม อาคาร ถนน เป็นต้น แปลงภาพสีให้เป็นขาวดำเพื่อลดพื้นหลังของภาพ



ภาพที่ 8 ภาพตัวอย่างการทำภาพขาวดำกับการทำ Canny Edge Detection

ทำการหาขอบภาพด้วยวิธีแคนนี่ (Canny Edge Detection)

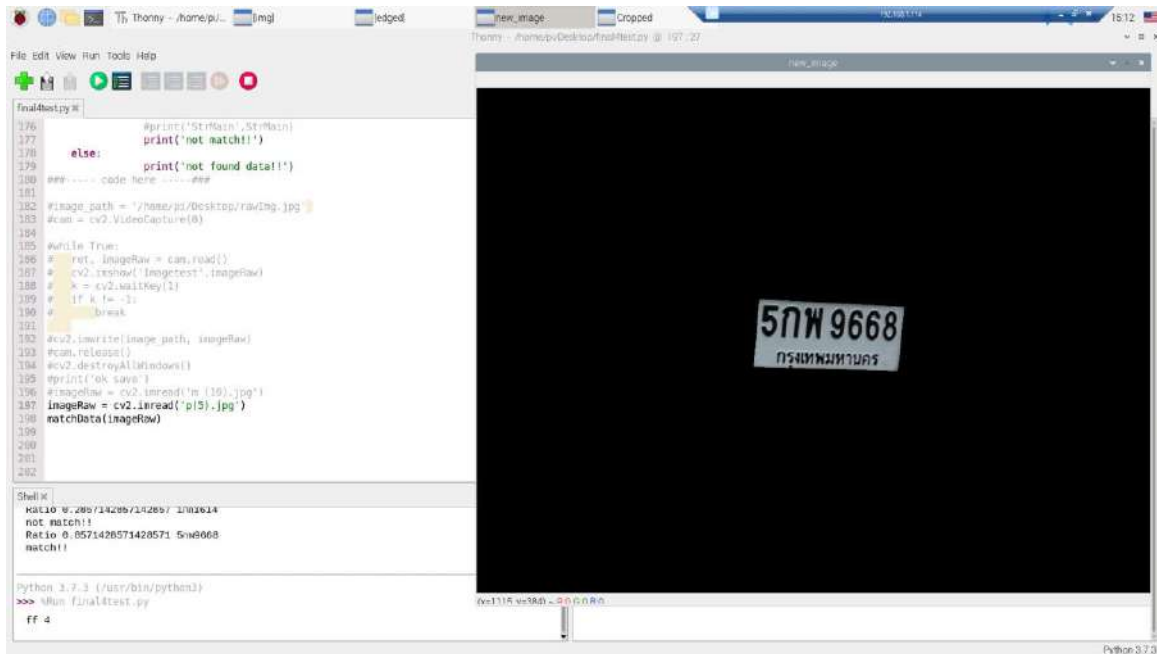
เป็นการหาเส้นรอบวัตถุที่อยู่ในภาพ เมื่อทราบเส้นรอบวัตถุจะสามารถคำนวณหาพื้นที่หรือรู้จำชนิดของวัตถุนั้นได้ การหาขอบภาพที่ถูกต้องสมบูรณ์ไม่ใช่เป็นเรื่องที่ง่าย โดยเฉพาะอย่างยิ่งการหาขอบของภาพที่มีภาพต่ำมีความแตกต่างระหว่างพื้นหน้ากับพื้นหลังน้อย หรือมีความสว่างไม่สม่ำเสมอทั่วภาพขอบภาพเกิดจากความแตกต่างของความเข้มแสงจากจุดหนึ่งไปยังอีกจุดหนึ่ง



ภาพที่ 9 ตัวอย่าง การทำ CannyEdge Detection

ตัดภาพป้ายทะเบียน

โดยใช้ Opencv การหาพื้นที่ป้ายทะเบียน เป็นรูปสี่เหลี่ยม จะต้องหา มุม 4 มุมของป้ายทะเบียน โดยใช้ approxPolyDP และได้ทำการCrop ภาพ



ภาพที่ 10 ภาพตัวอย่างการตัดภาพ โดยใช้ approxPolyDP

ระบบตรวจจับป้ายทะเบียน (Pytesseract)

ระบบตรวจจับป้ายทะเบียนเป็นระบบแอปพลิเคชัน (Pytesseract) ที่ติดตั้ง บนเครื่องแม่ข่าย โดยใช้ .NET Framework ในการพัฒนา โดยระบบจะทำการโหลดข้อมูลกล่อง จากฐานข้อมูล ซึ่งระบุถึง รหัส ชื่อ สถานี ละติจูดลองจิจูด ของสถานีนั้นๆ เพื่อทำการเชื่อมต่อ ด้วยโอเพนซีวี (Pytesseract) และดึงภาพจากกล้อง ในแต่ละสถานีเพื่อนำมาประมวลผล โดยตัวระบบสามารถเลือกเฉพาะกล่องที่สนใจ หรือกล่องทั้งหมดที่ติดตั้ง ได้ ในการเปิดใช้งาน เป็นเครื่องมือใช้ในการรู้จำตัวอักษรหรือที่รู้จักกัน คือ optical character recognition (OCR) Tesseract มีความสามารถในการเรียนรู้จดจำได้มากกว่า 100 ภาษา และยังเป็น Open Source ที่ได้ใช้ Tesseract ในงาน Text Detection ใช้ Code pytesseract.image_to_string

```
Shell x
Python 3.7.3 (/usr/bin/python3)
>>> %Run final4test.py
ff 4
data 6กต 2660
↓
strCar 6กต2660
strCar 6กต2660
Ratio 0.0 ฎง9999
not match!!
Ratio 0.3076923076923077 กท5628
```

ภาพที่ 11 ภาพระบบ ตรวจจับป้ายทะเบียน Pytesseract ข้อมูลที่ได้

การคัดตัวอักษรที่ไม่เกี่ยวกับป้ายทะเบียนรถยนต์

เป็นตัวเลขหรือตัวอักษรจากภาพนำเข้า เป็นกระบวนการควบคุมและกรองตัวอักษรดังกล่าว ตัวอย่างเช่น เมื่อนำภาพนำเข้า ต้องการแยกตัวเลขและตัวอักษรอื่น ๆ และตัวอักษรพิเศษ เช่นเครื่องหมายดอลลาร์เครื่องหมายทศนิยม เช่น ในการควบคุมรายละเอียด ของข้อมูลได้อย่างถูกต้อง โดยใช้ code =
tessedit_char_whitelist=123456789กขขคคฅงจจขชฌญญฎฐฏฒณคตถททณบปฝฝฟฟภมยรลวศษสฬหออฮ
ในการจำแนกตัวอักษรทะเบียนรถยนต์

<pre>ff 4 data [6กต2660 ↓ strCar [6กต2660 strCar [6กต2660 Ratio 0.0 ฎง9999 not match!! Ratio 0.2857142857142857 กท5628 not match!! Ratio 0.14285714285714285 ฎง5568</pre>	<pre>ff 4 data 6กต 2660 ↓ strCar 6กต2660 strCar 6กต2660 Ratio 0.0 ฎง9999 not match!! Ratio 0.3076923076923077 กท5628 not match!! Ratio 0.15384615384615385 ฎง5568</pre>
---	---

ภาพที่ 12 ตัวอย่าง การทำจำแนกตัวอักษรทะเบียนรถยนต์ โดยตัดอักษรพิเศษออก

การออกแบบระบบซอฟต์แวร์ในการเก็บข้อมูล

การออกแบบของระบบ จะแบ่งออกเป็น 2 ส่วน คือ ระบบตรวจจับป้ายทะเบียน ซึ่งเป็นระบบในการเชื่อมโยงข้อมูลระหว่างกล้องในแต่ละสถานีกับเครื่องแม่ข่ายแล้วส่งข้อมูลไปจัดเก็บ ที่ฐานข้อมูล ซึ่งเป็นระบบในการ

เชื่อมโยงระหว่างเครื่องแม่ข่ายและเครื่องลูกข่าย โดยดึงข้อมูลจากฐานข้อมูลแล้วนำไปแสดงผล ข้อมูลที่ได้จะมีดังนี้ 1.ทะเบียนรถยนต์ 2.วันที่ 3.เวลา มีการใช้ Code เพื่อนำข้อมูล ออกมาเป็นไฟล์

```

177
178 else:
179     print('not found d
180     ##----- code here -----##
181
182 #image_path = '/home/pi/Desktop/ra
183 #cam = cv2.VideoCapture(0)
184
185 #while True:
186 #     ret, imageRaw = cam.read()
187 #     cv2.imshow('Imagetest',imageRa
188 #     k = cv2.waitKey(1)
189 #     if k != -1:
190 #         break
191
192 #cv2.imwrite(image_path, imageRaw)

```

```

Shell M
not match!!
Ratio 0.42857142857142855 5mm9668
not match!!
Ratio 0.3333333333333333 กบ213
not match!!
Ratio 0.2857142857142857 2mm8256
not match!!
Ratio 1.0 6mm2660
match!!

```

ภาพที่ 13 ภาพฐานข้อมูล ป้ายทะเบียน

5. สรุปผลการวิจัย

การวิจัยในครั้งนี้ เป็นการวิจัยเพื่อการตรวจจับป้ายทะเบียนรถยนต์ การเรียนรู้เชิงลึกเพื่อนำไปพัฒนาหรือสิ่งประดิษฐ์ใหม่ๆ ในปัจจุบันมีรถยนต์เพิ่มมากขึ้นอย่างรวดเร็ว โดยมีวัตถุประสงค์เพื่อศึกษาการปรับเพิ่มกระบวนการตรวจจับป้ายทะเบียนรถยนต์ การเรียนรู้เชิงลึกเพื่อการสร้างความสะดวกในการดำเนินชีวิต และทดสอบประสิทธิภาพของกระบวนการตรวจจับป้ายทะเบียนรถยนต์ เก็บข้อมูลมาวิเคราะห์ และนำไปสร้างนวัตกรรมใหม่ มีความปลอดภัยในการดำเนินชีวิต

เป็นการนำผลลัพธ์หลักของป้ายทะเบียนรถยนต์ที่ได้ จากการตัดมาทำการหาตัวอักษรและตัวเลขของป้ายทะเบียนรถหรือวัตถุที่อยู่ในป้ายทะเบียนว่ามีวัตถุหรือไม่ โดยมีวิธี การทำเทรชโฮลด์ภาพ (Threshold image) ได้มีการกำหนดค่าเทรชโฮลด์ขึ้นมา 2 ค่า คือ ค่าเทรชโฮลด์สูง และค่าเทรชโฮลด์ต่ำ โดยจุดสีที่มีค่ามากกว่า จะถูกปรับเป็น 1 แต่ถ้าน้อยกว่า จะถูกปรับเป็น 0 ส่วนค่าที่อยู่ระหว่างค่า Threshold ทั้งสอง การปรับเป็นค่า 0 หรือ 1 นั้นขึ้นอยู่กับจุดสีที่อยู่รอบข้าง หากพบว่าจุดสีที่อยู่รอบข้างของจุดสีที่เป็นขอบ มีค่ามากกว่า จะปรับค่าจุดสีดังกล่าวให้มีค่าเป็น 1 และถือเป็นสมาชิกหนึ่งในภาพ

ตรวจสอบป้ายทะเบียนในพื้นที่กลางแจ้งแบบอัตโนมัติด้วยกระบวนการวิเคราะห์ภาพโดยระบบสามารถตรวจจับป้ายทะเบียนได้แสดง ผลลัพธ์ในการทดลองโปรแกรม เป็นผลลัพธ์โดยรวมภาพที่ใช้ทดสอบ

ทั้งหมด ตัวอย่างที่มีรหัสทั้งสิ้นจำนวน 10 ป้ายทะเบียนรถยนต์ ค้นพบว่าระบบสามารถตรวจจับป้ายทะเบียนทะเบียนได้ถูกต้องมากกว่า 76.93%

ตารางการทดลอง

ในการโปรแกรมใช้งานจริงในสิ่งแวดล้อม เดียวกัน ได้ทำการทดลองสแกนป้ายทะเบียนรถยนต์ ผ่านโปรแกรมที่ได้ทำการจัดเตรียมไว้เป็นจำนวน 10 ป้ายทะเบียนรถ เป็นมุ่มเดียวกัน ห่างจากตัวรถยนต์ ระยะ 1.5 เมตร มุมเฉียงออกมาจากข้างรถ ประมาณ 50-60 องศา ความสูงจากพื้น 50 เซนติเมตร ในที่ลานจอดรถที่มีสิ่งแวดล้อมเดียวกัน จะมีการเก็บตัวอักษรและตัวเลขของป้ายทะเบียนรถยนต์ เพอร์เซ็นความถูกต้องจะใช้ไลบรารี Levenshtein Distance ในการหาความถูกต้อง ของตัวอักษรและตัวเลข

ผลที่ได้ มาจากโปรแกรมที่ได้คำนวณมา จะมีตัวอักษรพิเศษหรือตัวอักษรที่ไม่จำเป็นของป้ายทะเบียนรถ จะทำการตัดตัวอักษร นั้นๆ ด้วยการ ใช้ Code Whitelisting and Blacklisting Characters with Tesseract and Python ในการตัวอักษรที่ไม่จำเป็นออกไป เพื่อ ให้ผลที่ออกมาสมบูรณ์มากขึ้น

ลำดับ	เลขป้ายทะเบียน	เปอร์เซ็นความถูกต้อง	ผลที่ได้	หมายเหตุ
1	5ก9668	100.00%	5ก9668	
2	5กน920	92.31%	5กน 9๑20	
3	6กต2660	42.86%	[6กต2660	
4	งค4716	0.00%	-	จับวัตถุมีรูปลี่เหลี่ยม
5	4ขก2394	49.56%	1ขอ ยี 2894	
6	1ขธ5942	100.00%	1ขธ5942	
7	6กต2660	100.00%	6กต2660	
8	กบ213	84.58%	กป213	
9	6กต2660	100.00%	6กต2660	
10	2ขษ8256	100.00%	2ขษ8256	
ผลรวม		76.93%		

ภาพที่ 14 การแสดงผลลัพธ์ในการทดลองโปรแกรม

6. อภิปรายผล

ผลการวิจัยเรื่อง ระบบการตรวจจับป้ายทะเบียนรถยนต์การเรียนรู้เพื่อลดขั้นตอนและความ สะดวกสบายในการใช้ชีวิต มีความเป็นไปได้ที่จะทำให้โครงการนี้เป็นจริงขึ้นมา เนื่องจากได้มี การศึกษา โปรแกรม สามารถอ่านป้ายทะเบียนรถยนต์ออกมาได้ แต่ยังมีข้อจำกัดอยู่คือ ไม่ สามารถอ่านค่าป้ายทะเบียนรถยนต์ได้หากภาพที่ได้จากกล้องวิดีโอหรือภาพไม่ชัดเจนพอหรือมีแสงสว่างไม่ มากพอ ซึ่งเป็นส่วนที่ควรนำไปปรับปรุงต่อไปโดยสรุปแล้วข้อดีของโปรแกรมการอ่านป้ายทะเบียนรถยนต์ อัตโนมัติ คือสามารถช่วยลดระยะเวลาในการใช้ชีวิตมากขึ้น ซึ่งในอนาคตหากเทคโนโลยีนี้ได้รับการพัฒนา ให้มีประสิทธิภาพและนำไปใช้

7. ข้อเสนอแนะ

- 1 สามารถแจ้งเตือน ผ่านแอปพลิเคชันได้
- 2 สามารถยืนยันป้ายทะเบียนที่สามารถใช้ทางพิเศษ (ทางด่วน)
- 3 สามารถใช้ระบบตรวจป้ายทะเบียนในการเปิดประตูบ้านได้

บรรณานุกรม

- กิตติพงษ์ เรืองหา, นิธิรัตน์ ยางงาม, และ วิสาข์ รังสีโยภาส. (2555) ระบบตรวจสอบทะเบียนและข้อมูล รถยนต์ภายใน มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี. <http://www.research.rmutt.ac.th/wp-content/uploads/2014/03/133121.pdf>
- วรรณพงษ์. (2558). สร้างระบบจดจำใบหน้าเฉพาะบุคคลด้วย OpenCV กับภาษาไพทอน. <https://python3.wannaphong.com/2015/09/opencv.html>
- มิสเตอร์เนย์. (2018). Detect License Plate with API. <https://rithikied.medium.com/detect-license-plate-with-api-8ddbe83be66f>
- ศิริรัตน์ สุนทรสุข.(2559). การวิเคราะห์ข้อความตัวอักษรหน้าจากภาพอย่างอัตโนมัติ. คณะวิทยาศาสตร์และเทคโนโลยี สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยธรรมศาสตร์. https://ethesisarchive.library.tu.ac.th/thesis/2016/TU_2016_5709035033_5280_3918.pdf
- อนุรักษ พรหมโคตร. (2561). การรู้จำตัวอักษรแผ่นป้ายทะเบียนด้วยเทคนิคการรู้จำตัวอักษรด้วยแสงร่วมกับ เทคนิคการหาค่าที่ใกล้เคียงโดยแสดงผลเป็นเสียงพูด. คณะวิศวกรรมศาสตร์, หลักสูตรวิศวกรรม ศาสตรมหาบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี. <http://www.repository.rmutt.ac.th/handle/123456789/3457>

Chayapol Moemeng (2018). วิเคราะห์ป้ายทะเบียนรถ ด้วย *OpenALPR* และ *Python*.

<https://mchayapol.medium.com>