

ระบบตรวจจับและอ่านป้ายทะเบียนยานพาหนะ

สำหรับระบบบริหารที่จอดรถ

Vehicle License Plate Detection and Recognition for Parking Management System

ดุลยณัฐ นิตยใหม่¹ ศราวุฒิ พ่อสาร¹ และ ธนพล ตั้งชูพงศ์^{1*}

¹สาขาวิชาวิทยาการคอมพิวเตอร์ วิทยาลัยการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น

Emails: dulyanut.nitmai@kkumail.com¹, sarwut.p@kkumail.com¹, thanaphon@kku.ac.th^{1*}

บทคัดย่อ

งานวิจัยนี้นำเสนอกระบวนการพัฒนาระบบตรวจจับและอ่านป้ายทะเบียนจากการประมวลผลภาพ โดยอาศัยระบบโครงข่ายประสาทเทียมร่วมกับอัลกอริทึมสำหรับการประมวลผลภาพ โดยระบบที่นำเสนอแบ่งออกเป็น 2 ส่วน ได้แก่ ส่วนแรกการพัฒนาแบบจำลองสำหรับตรวจจับป้ายทะเบียน ซึ่งได้คัดเลือกตัวแบบจากตัวแปร จำนวนของพารามิเตอร์ และการเปรียบเทียบประสิทธิภาพค่า mAP@0.5:0.95 ของตัวแบบ YOLOv5n YOLOv5s และ YOLOv5m ซึ่งวัดค่า mAP ได้ 0.647, 0.665 และ 0.685 ตามลำดับ สำหรับฝึกสอนชุดข้อมูลรูปภาพรถยนต์ที่มีป้ายทะเบียนติดอยู่ จำนวน 610 ภาพ ซึ่งผู้วิจัยได้เลือก YOLOv5s ซึ่งมีค่า mAP ที่ใกล้เคียงกับโมเดล YOLOv5m และ มีจำนวนพารามิเตอร์ที่น้อยกว่าอยู่ถึง 2.8 เท่า และส่วนที่สองเป็นการประมวลผลภาพเพื่ออ่านตัวอักษรจากป้ายทะเบียนโดยใช้เครื่องมือสำหรับการอ่านข้อมูลจากภาพ เปลี่ยนเป็นตัวอักษร หรือ Tesseract OCR ผลลัพธ์ที่ได้ เปอร์เซนต์ค่าเฉลี่ยของความถูกต้องในการอ่านตัวอักษรทุกตัวรวมกันได้เท่ากับ 78.34% และ ความถูกต้องในการอ่านชื่อจังหวัดทุกจังหวัดรวมกันได้เท่ากับ 79.35%

คำสำคัญ: ตรวจจับป้ายทะเบียน อ่านตัวอักษรป้ายทะเบียน Tesseract YOLO

* Corresponding author, e-mail: thanaphon@kku.ac.th

Abstract

This research presents a process for developing a license plate detection and reading system based on image processing. An artificial neural network system, in conjunction with an algorithm for image processing, is used to develop the system. The proposed system is divided into two parts. The first part involves the development of a model for detecting license plates. A number of models were considered, and their mean average precision (mAP) values were compared at a threshold of 0.5-0.95. The models selected for comparison were YOLOv5n, YOLOv5s, and YOLOv5m. These models were trained on a dataset of 601 license plate images of cars, and the resulting mAP values were 0.647, 0.665, and 0.685, respectively. The researcher chose YOLOv5s due to its similar mAP values to the YOLOv5m model and its 2.8 times fewer parameters. The second part of the proposed system involves the use of Tesseract OCR, a tool for reading data from images to characters, to read the characters from the license plates. The average percentage of reading accuracy of all letters was 78.34% and the combined accuracy of reading province names of all provinces was 79.35%.

Keywords: License Plate Detection, License Plate Recognition, Tesseract, YOLO

1. ที่มาและความสำคัญ

ปริมาณของรถที่มีจำนวนเพิ่มมากขึ้น นำมาซึ่งความท้าทายที่สำคัญในการจัดการและบริหารจัดการที่จอดรถ ดังนั้นระบบที่สามารถตรวจจับและอ่านป้ายทะเบียนรถยนต์ สามารถนำไปสู่ประโยชน์หลายอย่าง เช่น การบริหารจัดการการเข้า-ออกของรถยนต์อย่างมีประสิทธิภาพ การประหยัดเวลาในการบันทึกข้อมูลการเข้า-ออกของยานพาหนะ และการตรวจสอบรถที่ได้รับอนุญาตให้เข้า-ออก เป็นต้น ด้วยเหตุนี้ทางผู้คณะวิจัยจึงได้มีแนวคิดที่จะพัฒนาระบบตรวจจับและอ่านป้ายทะเบียนรถ สำหรับทะเบียนรถในประเทศไทย ให้มีความสามารถตรวจจับตำแหน่งและอ่านตัวอักษรจากป้ายทะเบียนของยานพาหนะแบบอัตโนมัติ โดยสิ่งที่ต้องการจากป้ายทะเบียน ได้แก่ หมวดหมู่ตัวอักษรไทย หมายเลขทะเบียน และจังหวัดที่จดทะเบียน ซึ่งสามารถใช้งานได้เฉพาะป้ายทะเบียนที่ใช้ในประเทศไทยและได้รับรองโดยกรมการขนส่งทางบกเท่านั้น

จากเหตุผลที่กล่าวมาข้างต้นทางคณะผู้วิจัยได้นำเสนอการจัดเตรียมชุดข้อมูลเพื่อที่จะนำไปใช้ในการพัฒนาแบบจำลองสำหรับตรวจจับตำแหน่งของป้ายทะเบียนจากยานพาหนะ รวมทั้งการพัฒนาวิธีการประมวลผลของภาพเพื่อให้ได้ข้อมูลที่พร้อมในการสกัดข้อมูลจากวิธีการอ่านอักขระด้วยแสงในการอ่านตัวอักษรจากภาพป้ายทะเบียน

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ระบบสี RGB

RGB ย่อมาจาก red, green และ blue [1] คือ กระบวนการผสมสีจากแม่สี 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน การใช้สัดส่วนของสี 3 สีนี้ต่างกัน จะทำให้เกิดสีต่างๆ ได้อีกมากมาย กล่าวคือระบบสี RGB นั้นเป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง สีเขียว และสีน้ำเงิน โดยมีการรวมกันแบบการบวก (Additive) และในการกำหนดค่าสีนั้นจะกำหนดได้ตั้งแต่ 0-255 เช่น $R = 0$, $G = 174$, $B = 239$ เป็นต้น

2.2. ทฤษฎีโทนสีเทา (Gray scale)

โทนสีเทา เป็นรูปที่เก็บโดยใช้รูปแบบของอาร์เรย์ 2 มิติ โดยค่าที่เก็บจะมีค่าอยู่ในช่วงหนึ่ง ซึ่งระดับของสีขึ้นอยู่กับขนาดของบิตที่ใช้เก็บค่าสี ทำได้โดยการนำค่าของแสงสีแต่ละสีมารวมกันแล้วหารด้วย 3 หรือการนำค่าของแสงสีแต่ละสีมาหาค่าเฉลี่ย แต่หากทำโดยวิธีนี้อาจจะทำให้เกิดปัญหาได้ คือ ถ้าค่าของแม่สีใดมีค่ามากกว่าสีอื่นมาก ๆ จะทำให้ค่าเฉลี่ยของสีนั้นไปทางค่านั้น มีผลทำให้ภาพมีโทนสีที่ผิดเพี้ยนไปได้ ดังนั้นจึงต้องคำนึงถึงความสว่างของแม่สีด้วย [1]

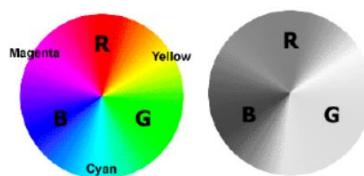
$$Gray = R*0.299 + G*0.587 + B*0.114 \quad (1)$$

โดยที่ Gray = โทนสีเทา

R = ระดับของแสงสีแดง

G = ระดับของแสงสีเขียว

B = ระดับของแสงสีน้ำเงิน



รูปที่ 1 แสดงการเปรียบเทียบความสว่างของแต่ละเฉดสี [1]

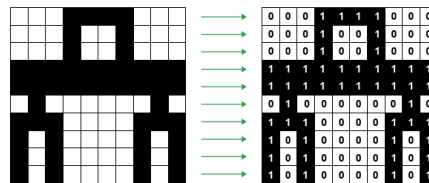
สมการข้างต้นเป็นวิธีการทั่วไปในการแปลงภาพจากพื้นที่สี RGB (แดง เขียว น้ำเงิน) เป็นโทนสีเทา กระบวนการแปลงภาพจากสีเป็นโทนสีเทาเกี่ยวข้องกับการหาค่าเฉลี่ยของค่าสีแดง เขียว และน้ำเงินสำหรับแต่ละพิกเซลในภาพ

ค่าสัมประสิทธิ์ 0.299, 0.587 และ 0.114 ในสมการจะใช้เพื่อถ่วงน้ำหนักค่าสีแดง สีเขียว และสีน้ำเงิน ตามลำดับ ก่อนที่จะหาค่าเฉลี่ย เหตุผลในการใช้ค่าสัมประสิทธิ์เฉพาะเหล่านี้คือ ค่าสัมประสิทธิ์ดังกล่าวขึ้นอยู่กับความไวสัมพัทธ์ของสายตามนุษย์ต่อแสงสีแดง เขียว และน้ำเงิน

2.3 ทฤษฎีภาพขาว-ดำ (Binary image)

ภาพขาว-ดำ เป็นรูปภาพที่ประกอบด้วยสีขาวและสีดำเท่านั้น โดยที่ภาพจะอยู่ในรูปแบบของช่องขนาดเล็กลงๆ ช่องรวมกัน ซึ่งแต่ละช่องจะนับเป็นส่วนหนึ่งของรูปภาพหรือตัวอักษร และเราจะเรียกแต่ละจุดหรือ

แต่ละช่องนั้นว่า พิกเซล โดยแต่ละพิกเซลจะถูกกำหนดให้มีระดับของความเข้มของสีขาวและสีดำ ซึ่งแสดงให้อยู่ในรูปของรหัสฐานสอง (0 กับ 1) แต่ละพิกเซลก็จะแทนด้วย บิต (bits) ซึ่งจะถูเก็บเป็นลำดับในคอมพิวเตอร์ โดยส่วนใหญ่แล้วคอมพิวเตอร์เองจะสามารถวิเคราะห์ข้อมูลจากรูปภาพขาวดำได้ง่ายกว่ารูปแบบอื่นๆ เนื่องจากรูปภาพขาวดำมีค่าข้อมูลของสีเพียง 2 ค่าซึ่งแทนด้วยค่า 0 กับ 1 โดยที่ค่า 0 จะแทนสีดำและ 1 จะแทนสีขาว [2]



รูปที่ 2 แสดงแต่ละ pixel ที่ถูกกำหนดให้มีค่า 0 (สีดำ) และ 1 (สีขาว) [2]

2.4 การทำเทรชโฮลด์ (Thresholding)

การทำเทรชโฮลด์ เป็นเทคนิคที่ถูกใช้อย่างมากในกระบวนการประมวลผลภาพ ซึ่งเป็นกระบวนการแปลงภาพสีให้มีการแสดงผลได้แค่ 2 สี คือ สีขาวและสีดำ ซึ่งจะแปลงข้อมูลภาพให้เป็นภาพแบบขาว-ดำ โดยจะทำการแปลงภาพที่มีความเข้มหลายระดับ (multilevel image) ให้เป็น ภาพขาว-ดำ โดยกำหนดค่าให้พิกเซลที่มีค่าต่ำกว่าค่าเทรชโฮลด์เป็นค่าๆ หนึ่ง และมีค่าสูงกว่าเป็นค่า ๆ อีกค่าหนึ่ง



รูปที่ 3 เปรียบเทียบภาพก่อนและหลังการทำเทรชโฮลด์ [3]

ในการพิจารณาจุดพิกเซลบนภาพว่าจุดใดควรแทนด้วยจุดขาว หรือจุดใดควรเป็นจุดที่มีค่าเท่ากับ 1 จะทำโดยการเปรียบเทียบค่าความเข้มของแสงของแต่ละพิกเซลกับค่าคงที่ที่เรียกว่า เทรชโฮลด์ โดยเทคนิคนี้ มักใช้กันในกรณีที่วัตถุและพื้นหลังที่มีความแตกต่างกันมากๆ

2.5 การหมุนหรือบิดแปลงภาพ (Perspective transformation)

การหมุนหรือบิดแปลงภาพ เป็น การแปลงภาพทางเรขาคณิต ประเภทหนึ่งที่แปลงพิกัดจุดในภาพอินพุต ไปยังตำแหน่งต่างๆ ในภาพเอาต์พุต [4] ซึ่งสามารถใช้เพื่อแก้ไขความผิดเพี้ยนของภาพหรือเปลี่ยนขอบเขตการมองเห็น

$$\begin{bmatrix} t_i x'_i \\ t_i y'_i \\ t_i \end{bmatrix} = \text{map_matrix} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (2)$$

โดยที่

$$src(i) = (x_i, y_i) \quad (3)$$

$$dst(i) = (t_i x'_i, t_i y'_i) \quad (4)$$

$$i = 0, 1, 2, 3 \quad (5)$$

src: คือภาพอินพุต ที่เป็นจุดตำแหน่งอาร์เรย์ขนาดสองมิติ ที่มีขนาด (x,y)

dst: คือภาพเอาต์พุต ที่มีขนาดและประเภทเดียวกับภาพอินพุต

map_matrix: เมทริกซ์สำหรับแปลงพิกัดจุด (Affine Transformation Matrix) [4] [5]

i: คือลำดับจุดแมพที่อยู่ในภาพ

2.6 การวัดผลด้วยการคำนวณหาอัตราส่วนระหว่างพื้นที่ที่ทับซ้อน (Intersection over Union)

อัตราส่วนระหว่างพื้นที่ที่ทับซ้อน (Intersection over Union: IoU) เป็นการวัดประสิทธิภาพซึ่งเป็นที่นิยมในแบบจำลองประเภทการตรวจจับวัตถุ ซึ่งสามารถหาได้จาก อัตราส่วนระหว่างพื้นที่ที่ทับซ้อนของ 2 bounding box หาด้วยพื้นที่รวมของกรอบทั้งสอง หรืออาจเรียกว่า ดัชนีแจ็กการ์ด (Jaccard Index) เป็นหลักวิธีการในการหาจำนวนเปอร์เซ็นต์ที่ทับซ้อนกันระหว่างผลเฉลย (Ground Truth) และผลจากการทำนาย (Predict) ซึ่งหากถ้าได้ค่า IoU ที่มากกว่า 0.5 ขึ้นไปก็จะสามารถยอมรับได้ [6]

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \quad (6)$$

เมื่อ A คือ พื้นที่ที่ถูกทำนาย และ

B คือ พื้นที่ที่ถูกลาเบลไว้

2.7 การหาความเฉลี่ยความแม่นยำ (mean Average Precision)

การหาความเฉลี่ยความแม่นยำ (mAP) [7] ใช้เพื่อวัดประสิทธิภาพของโมเดลการมองเห็นด้วยคอมพิวเตอร์ mAP เท่ากับค่าเฉลี่ยของตัวชี้วัดความแม่นยำเฉลี่ยในทุกคลาสในแบบจำลอง สามารถใช้ mAP เพื่อเปรียบเทียบเวอร์ชันที่แตกต่างกันในงานเดียวกัน โดยที่ mAP วัดได้ระหว่าง 0 ถึง 1

$$mAP = \frac{1}{|classes|} \sum_{c \in classes} \frac{|TP_c|}{|FP_c| + |TP_c|} \quad (7)$$

โดยที่

TP = True Positive คือ ข้อมูลที่ทำนายตรงกับข้อมูลจริงในคลาสที่กำลังพิจารณา

FP = False Positive คือ ข้อมูลที่ทำนายผิดเป็นคลาสที่กำลังพิจารณา

2.8 ระยะทางเลเวนชเตйнหรือระยะแก้ไข (Levenshtein Distance)

ระยะทางเลเวนชเตйн [8] หรือระยะแก้ไข เป็นการวัดความคล้ายคลึงกันระหว่างสตริงสองสาย เป็นจำนวนขั้นต่ำของการแก้ไขอักขระเดียว (การแทรก การลบ หรือการแทนที่) ที่จำเป็นสำหรับการเปลี่ยนสตริงหนึ่งไปยังอีกสตริงหนึ่ง มักใช้ในการประมวลผลภาษาธรรมชาติ การตรวจตัวสะกด และชีวสารสนเทศ ยิ่งระยะการแก้ไขต่ำเท่าไร สตริงทั้งสองก็ยิ่งคล้ายกันมากขึ้นเท่านั้น สามารถคำนวณได้โดยใช้โปรแกรมไดนามิก

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases} \quad (8)$$

โดยที่

lev = ระยะทางเลเวนชเตย์น

a = อักขระของตัวแปร a

b = อักขระของตัวแปร b

i = ตำแหน่งของอักขระที่ i

j = ตำแหน่งของอักขระที่ j

ai หมายถึงอักขระของสตริง a ที่ตำแหน่ง i

bj หมายถึงอักขระของสตริง b ที่ตำแหน่ง j

ทำตรวจสอบว่า a และ b เท่ากันหรือไม่ หากเท่ากัน ไม่จำเป็นต้องแก้ไข ไม่เพิ่ม 1 ในทางกลับกัน หากไม่เท่ากัน ต้องเพิ่ม 1 เพื่อนับการแก้ไข

2.9 ระบบจำแนกป้ายทะเบียนอัตโนมัติ

ชัยรัช มินเดินเรือ [9] ได้จัดทำระบบจำแนกป้ายทะเบียนอัตโนมัติ โดยเป็นระบบที่จะตรวจสอบและจดจำทะเบียนรถ ที่ผ่านเข้ามาในบริเวณที่ติดตั้งระบบเอาไว้ เพื่อบันทึกกิจกรรมทั้งหมดที่เกิดขึ้นในพื้นที่ ซึ่งใช้ระบบการแจ้งเตือนผ่าน Mobile Application ซึ่งจะส่งข้อความไปยังอุปกรณ์ที่ติดตั้งเอาไว้ ตัวอย่างการทำงานของระบบ เช่น หากมีการนำรถออกจากพื้นที่จอดรถโดยไม่ได้รับอนุญาต ระบบจะแจ้งเตือนทะเบียนของรถที่ต้องสงสัย และสามารถจัดทำสถิติเพื่อใช้ในการวิเคราะห์การเข้า-ออกของรถภายในพื้นที่จอดรถได้ ผลลัพธ์การดำเนินงานพบว่า ปัญหาที่ส่งผลต่อการอ่านป้ายทะเบียน คือ 1. ความคมชัดของป้ายทะเบียน เนื่องจากบริเวณที่ใช้งานมีแสงสว่างไม่เพียงพอและรถยนต์จอดไม่นิ่งพอ 2. ความละเอียดของภาพถ่าย ซึ่งขึ้นอยู่กับคุณภาพของกล้อง และสิ่งที่ตามมาก็คือหากภาพมีความละเอียดสูงก็จะใช้เวลาในการประมวลผลนานขึ้น 3. ความใกล้เคียงของตัวอักษรบนป้ายทะเบียนรถยนต์ เนื่องจากตัวอักษรภาษาไทยมีความคล้ายกันหลายตัวอักษร เช่น ‘ก’ กับ ‘ถ’ กับ ‘ภ’, ‘ด’ กับ ‘ต’ กับ ‘ค’ หรือ ‘ข’ กับ ‘ช’ กับ ‘ส’ เป็นต้น 4. ปัญหาป้ายทะเบียนรถยนต์ชำรุด ซึ่งปัญหานี้เป็นปัญหาที่อยู่นอกเหนือการควบคุม

2.10 เทคโนโลยีการอ่านป้ายทะเบียนรถยนต์อัตโนมัติ

ชมพูนุท ไชยประสิทธิ์ [6] ได้นำเทคโนโลยีการอ่านป้ายทะเบียนรถยนต์อัตโนมัติ มาช่วยติดตามรถยนต์ที่ถูกโจรกรรม โดยการนำภาพจากกล้องวงจรปิดที่ติดตั้งอยู่ตามบริเวณแยกต่างๆ มาใช้ประโยชน์ โดยการเอาภาพจากกล้องมาทำการเปลี่ยนแปลงจากภาพสีให้เป็นภาพขาว-ดำ ปรับขนาดและความชัดเจนของภาพให้เหมาะสม ลดสิ่งรบกวนหรือสิ่งที่ไม่ต้องการออกจากภาพ ก่อนจะนำมาทำการ matching และ template ตัวอักษรไทยและตัวเลข โดยทำการทดลองกับป้ายทะเบียน 3 ประเภท ได้แก่ ป้ายทะเบียนพื้นหลังสีขาว สะท้อนแสง ตัวหนังสือสีดำ ป้ายทะเบียนพื้นหลังสีเหลืองสะท้อนแสง ตัวหนังสือสีดำ และ ป้ายทะเบียนที่มีพื้น

หลังเป็นภาพกราฟิก ผลการวิจัยพบว่า ระบบอ่านป้ายทะเบียนรถยนต์อัตโนมัติสามารถอ่านค่าได้อย่างถูกต้องมากกว่า 70 เปอร์เซ็นต์ แต่ก็ยังมีปัจจัยที่ส่งผลให้ระบบไม่สามารถอ่านค่าป้ายทะเบียนได้ถูกต้องทั้งหมด หรือมีข้อผิดพลาดจากการอ่านป้ายทะเบียน คือ ช่วงเวลาและแสงสว่าง ทั้งจากแสงอาทิตย์และแสงไฟรถยนต์ โดยช่วงเวลาที่สามารถอ่านป้ายทะเบียนได้ดีที่สุด คือ ช่วง 11.00 น. และประเภทของป้ายทะเบียนที่สามารถอ่านค่าได้ถูกต้องที่สุด คือ ป้ายทะเบียนพื้นหลังสีขาวสะท้อนแสง ตัวหนังสือสีดำ ดังนั้น จึงจะต้องทำการปรับปรุงระบบให้สามารถทำงานได้ดียิ่งขึ้น

3. วิธีดำเนินงานวิจัย

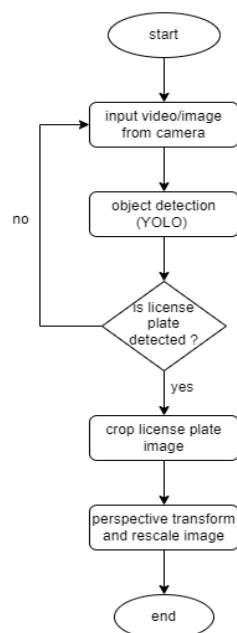
3.1 การเก็บรวบรวมข้อมูล

เก็บรวบรวมข้อมูลภาพรถยนต์จากอินเทอร์เน็ตจำนวน 610 ภาพ ซึ่งได้นำมาจากเว็บไซต์ โรโบฟลว์ (Roboflow) ซึ่งเป็นชุดข้อมูลที่มีการทำการวาดกรอบและติดป้ายกำกับส่วนที่เป็นป้ายทะเบียนเอาไว้แล้ว โดยจะเป็นภาพขนาด 3 ช่องสัญญาณ (Channel) หรือภาพ RGB

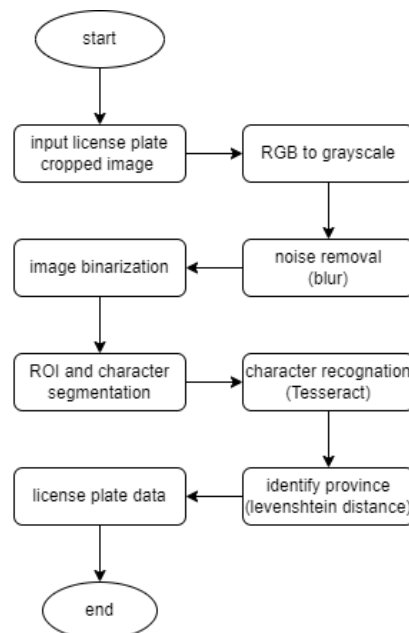
3.2 การเตรียมชุดข้อมูล

รูปภาพของรถยนต์ที่ติดแผ่นป้ายทะเบียนจำนวน 610 รูป ซึ่งเป็นชุดข้อมูลที่มีการทำการวาดกรอบและติดป้ายกำกับส่วนที่เป็นป้ายทะเบียนเอาไว้แล้ว และใช้วิธีการแบ่งข้อมูลแบบ split test โดยแบ่งชุดข้อมูลออกเป็น 2 ส่วน คือ 70 เปอร์เซ็นต์ ต่อ 30 เปอร์เซ็นต์ ดังนี้ ชุดข้อมูลฝึกสอน (Train data) เป็น 70 เปอร์เซ็นต์ จำนวน 426 รูป ชุดข้อมูลตรวจสอบ (Validate data) เป็น 15 เปอร์เซ็นต์ จำนวน 91 รูป และชุดข้อมูลทดสอบ (Test data) เป็น 15 เปอร์เซ็นต์ จำนวน 92 รูป

3.3 การพัฒนาแบบจำลองสำหรับตรวจจับป้ายทะเบียน



รูปที่ 4 ขั้นตอนและกระบวนการในการตรวจจับป้ายทะเบียน



รูปที่ 5 ขั้นตอนและกระบวนการในการอ่านตัวอักษรจากป้ายทะเบียน

การพัฒนาแบบจำลองจะฝึกสอนด้วยการใช้ YOLOv5 [10] โดยจะเลือกรูปแบบของแบบจำลอง โดยเป็นแบบ YOLOv5s เนื่องจากระบบตรวจจับและอ่านป้ายทะเบียนเป็นระบบที่ต้องใช้ความเร็วในการทำงาน จากนั้นนำชุดข้อมูลฝึกสอนในงานวิจัยนี้จะใช้ข้อมูลฝึกสอนที่เลเบลในรูปแบบของ YOLOv5s ซึ่งขนาดของภาพอินพุตจะถูกปรับให้เป็น 640×640 พิกเซล ภาพขนาด 3 ช่องสัญญาณ จากนั้นนำมาฝึกสอนกับชุดข้อมูลทดสอบ (Validation) เพื่อวัดประสิทธิภาพของตัวแบบในแต่ละอีพ็อกซ์ (epoch) โดยกำหนดเทรชโฮลด์ที่ 0.4 โดยมี จำนวนพารามิเตอร์ทั้งหมด 7,246,518 ตัว และฝึกสอนทั้งหมด 50 อีพ็อกซ์ โดยมีกระบวนการทำงานเพื่อตรวจจับป้ายทะเบียนดังรูปที่ 4 และเนื่องจากในความเป็นจริงภาพที่ได้จากกล้องถ่ายภาพอาจบิดเบือนหรือเอียงกว่าภาพที่ใช้ในการทดลอง ดังนั้นจึงต้องมีการปรับแก้ไขภาพก่อน ดังแสดงในขั้นตอนสุดท้ายของรูปที่ 4 โดยเป็นการฉายจุดลงบนฉากรับภาพตามแนวเส้นที่พุ่งออกมาจากจุดจุดหนึ่งโดยเรียกวิธีนี้ว่าการหมุนหรือบิดแปลงภาพ (Perspective transformation) ดังแสดงในสมการที่ (2)

3.4 การประมวลผลภาพเพื่ออ่านตัวอักษรจากป้ายทะเบียน

การใช้เทคนิคการอ่านอักขระด้วยแสง (OCR) ด้วย Tesseract ซึ่งจะสามารถรองรับการอ่านตัวอักษรภาษาไทยโดยการติดตั้งภาษาไทยเพิ่มเติม ซึ่งข้อมูลภาพที่ส่งเข้าไบนั้นจะต้องเป็นภาพแบบขาว-ดำ และเพื่อให้การอ่านข้อมูลมีความถูกต้องและแม่นยำที่สุด จึงจะต้องมีกระบวนการปรับปรุงคุณภาพของรูปภาพก่อน โดยมีขั้นตอนดังรูปที่ 5

จากรูปที่ 5 สามารถอธิบายเป็นขั้นตอนได้ดังนี้

- (1) นำเข้าข้อมูลภาพที่ผ่านกระบวนการตรวจจับป้ายทะเบียนด้วยแบบจำลองที่ฝึกสอนมาแล้ว
- (2) การแปลงรูปภาพที่มีระบบสี RGB ให้เป็นระบบสีเทา และลดสัญญาณรบกวนภายในภาพ
- (3) การแปลงรูปภาพที่มีระบบสีเทา ให้เป็นรูปภาพขาว-ดำ ด้วยการทำเทรชโฮลด์
- (4) การหาบริเวณที่สนใจในภาพหรือตัวอักษรบนป้ายทะเบียนด้วยวิธี การหาเส้นเค้าโครง (contour detection)
- (5) การอ่านตัวอักษรบนแผ่นป้ายทะเบียนที่ได้จากการหาเส้นเค้าโครงด้วย Tesseract จากนั้นใช้วิธีการวัดระยะทางเลขเวกเตอร์ กับชื่อจังหวัดที่อ่านได้โดยการเปรียบเทียบกับข้อมูลชื่อจังหวัดที่ถูกกำหนดไว้เป็นภาษาไทยทั้งหมด 77 ชื่อ จากนั้นทำการบันทึกข้อมูลที่ไดลงในฐานข้อมูล

3.5 การประเมินประสิทธิภาพของแบบจำลองตรวจจับป้ายทะเบียน

การประเมินประสิทธิภาพของแบบจำลองตรวจจับป้ายทะเบียน โดยใช้คะแนน IoU เป็นที่นิยมในการวัดประสิทธิภาพแบบจำลองสำหรับการตรวจจับวัตถุ และค่าผลลัพธ์ของ IoU ที่ได้จะมีค่าอยู่ระหว่าง 0 ถึง 1 ยิ่งค่าที่ได้เข้าใกล้ 1 มากเท่าไรแบบจำลองก็จะมีประสิทธิภาพมากเท่านั้น โดยที่ผลลัพธ์ของ IoU จะต้องมีความมากกว่า 0.5 ขึ้นไป ถึงจะเป็นที่ยอมรับได้

4. ผลและวิจารณ์

4.1 การศึกษาแบบจำลอง

ทดลองการฝึกสอนแบบจำลองสำหรับตรวจจับแผ่นป้ายทะเบียนจากยานพาหนะ โดยใช้แบบจำลอง YOLOv5 ทั้งหมด 3 แบบ ได้แก่ YOLOv5n YOLOv5s และ YOLOv5m [10] ซึ่งใช้ชุดข้อมูลชุดเดียวกัน กำหนดค่าอีพอคซ์ที่ 50 เท่ากันทั้งหมด และกำหนดขนาดภาพ 640 x 640 พิกเซล เท่ากันทั้ง 3 แบบ การทดลองทั้งหมดได้ฝึกสอนแบบจำลองบน Google Colab โดยใช้ GPU NVIDIA A100

จากตารางที่ 1 แสดงผลการฝึกสอนในทุกการทดลอง โดยแสดง จำนวนของพารามิเตอร์ เวลาที่ใช้ในการฝึกสอนโมเดล กับคะแนนเฉลี่ย precision, recall, mAP@.5 mAP@.5:.95 และ จำนวนโอเปอร์เรชันในการประมวลผลต่อเฟรม (FLOPs) [10] ของทุกโมเดลที่ได้มาจากการแบ่งข้อมูลฝึกสอนและทดสอบจำนวน 5 ส่วน จากเทคนิค K-Fold Cross Validation

ตารางที่ 1 ตารางแสดงลักษณะเด่นของแต่ละแบบจำลอง

models	parameters	Time (sec)	precision	recall	mAP @.5	mAP @.5:.95	FLOPs (Giga)
YOLOv5n	1,760,518	184.33	0.99798	0.99979	0.99500	0.647	4.5
YOLOv5s	7,246,518	202.48	0.99811	0.99895	0.99500	0.665	16.5
YOLOv5m	20,852,934	268.16	0.99884	0.99974	0.99500	0.685	49.0

จากการทดลองแบบจำลอง YOLOv5m มีประสิทธิภาพค่า mAP ที่มากที่สุด แต่ทางผู้วิจัยได้เลือกแบบจำลอง YOLOv5s เนื่องจากจำนวนพารามิเตอร์ที่น้อยกว่าถึง 2.8 เท่า ที่ 7.2 ล้านพารามิเตอร์ ซึ่งการประมวลผลภาพแต่ละเฟรมจะใช้คอมพิวเตอร์ชันนอลโอเปอเรชันที่ 16.5 GigaFLOPs ซึ่งส่งผลต่อความเร็วในการประมวลผลแบบเรียลไทม์ (realtime) ในการตรวจจับแผ่นป้ายทะเบียนที่รวดเร็วกว่า แต่มีประสิทธิภาพในการตรวจจับที่ใกล้เคียงกัน



รูปที่ 6 ตัวอย่างภาพที่สามารถตรวจจับป้ายทะเบียนได้และทำการตัดภาพส่วนที่ต้องการ

จากรูปที่ 6 แสดงให้เห็นว่าแบบจำลองสามารถตรวจจับป้ายทะเบียนที่มีพื้นหลังเป็นสีขาวได้ โดยที่มีความมั่นใจอยู่ที่ 0.92

เนื่องจากในความเป็นจริงภาพที่ได้จากกล้องถ่ายภาพอาจมีองค์ประกอบของการจับภาพที่ไม่สอดคล้องหรือเอียงกว่าภาพที่ใช้ในการพัฒนาโมเดลในการอ่านตัวอักษร ดังนั้น ถึงแม้ว่าโมเดลการตรวจจับแผ่นป้ายทะเบียนจะทำการตรวจจับและตัดแผ่นป้ายออกมาแล้ว แต่มุมมองของภาพที่ได้อาจจะไม่ได้อยู่ในรูปแบบที่ต้องการ ดังแสดงในภาพด้านซ้ายของ รูปที่ 7 จึงต้องมีการปรับแก้ไขมุมมองของภาพก่อนผ่านกระบวนการบิดแปลงภาพ โดยเป็นการฉายจุดลงบนฉากรับภาพตามแนวเส้นที่พุ่งออกมาจากจุดอ้างอิงจุดหนึ่งตามสมการที่ 2 [5]



รูปที่ 7 ตัวอย่างรูปก่อนและหลังจากการทำบิดแปลงภาพ

จากนั้นจึงนำภาพของแผ่นป้ายทะเบียน เข้าสู่กระบวนการในการอ่านตัวอักษร ดังแสดงในขั้นตอนวิธีของรูปที่ 5 ซึ่งได้แสดงตัวอย่างผลลัพธ์ในขั้นตอนต่าง ๆ ดังนี้



a) ตัวอย่างรูปที่แบบจำลองตรวจจับได้



b) ตัวอย่างรูปที่แปลงเป็นระบบสีเทา



c) ตัวอย่างรูปที่แปลงเป็นรูปภาพขาว-ดำ



d) ตัวอย่างรูปที่ได้หลังจากทำการทำการพองขยาย



e) ตัวอย่างรูปตัวอักษรบนป้ายทะเบียนที่ได้จากการทำการตัดเอาพื้นที่ ที่สนใจ



f) ตัวอย่างรูปตัวอักษรบนป้ายทะเบียนที่ได้จากการทำการแสดงเส้นรูปร่าง

รูปที่ 8 แสดงผลลัพธ์ของขั้นตอนการทำงานของอัลกอริทึมในการอ่านแผ่นป้ายทะเบียนหลังการบิดแปลงภาพ รูป a, b) แสดงตัวอย่างการแปลงรูปภาพระบบสี RGB ให้เป็นระบบสีเทา รูป c) แสดงการแปลงรูปภาพที่มีระบบสีเทา ให้เป็นรูปภาพขาว-ดำ ด้วยการทำการ threshold d) แสดงภาพที่ได้หลังจากทำการทำการพองขยาย (dilation) e, f) แสดงการหาตำแหน่งของตัวอักษรบนป้ายทะเบียนด้วยวิธีการแสดงเส้นรูปร่าง

4.2 ผลทดสอบจากการประมวลผลรูปภาพเพื่ออ่านตัวอักษรจากป้ายทะเบียน

จากภาพแผ่นป้ายทะเบียนในชุดข้อมูลทดสอบที่แบบจำลองตรวจจับได้และตัดภาพส่วนนั้นออกมาจำนวนทั้งหมด 92 ภาพจากการสุ่มจากข้อมูลภาพป้ายทะเบียน จะได้ทั้งหมด 420 ภาพ โดยจากภาพทั้งหมดมีจำนวนตัวอักษรรวมทั้งหมด 2591 ตัว โดยแบ่งเป็นตัวเลขอารบิกจำนวน 1671 ตัว และตัวอักษรภาษาไทยจำนวน 920 ตัว ซึ่งผลลัพธ์ในการอ่านตัวอักษรจากแผ่นป้ายทะเบียนสามารถสรุปได้ ดังตารางที่ 2

จากตารางที่ 2 จะเห็นว่าตัวอักษรภาษาไทยบางตัวไม่มีตัวอย่างของข้อมูล เนื่องจากกรรมการขนส่งทางบกได้ละเว้นตัวอักษรบางตัวไว้ในการนำมาสร้างหมวดหมู่ตัวอักษรบนป้ายทะเบียน เนื่องจากเป็นตัวอักษรที่มีความคล้ายคลึงกัน ได้แก่ ข ค ซ ฎ ฏ ป ฝ และ ฟ และจากการใช้วิธีการอ่านอักขระด้วยแสงอ่านตัวอักษรจากแผ่นป้ายทะเบียน พบว่ามีค่าความถูกต้องเฉลี่ยของการอ่านตัวเลขอารบิกรวมกันเท่ากับ 76.67% ส่วนค่าความถูกต้องเฉลี่ยของการอ่านตัวอักษรภาษาไทยรวมกันเท่ากับ 81.52% และค่าความถูกต้องเฉลี่ยของการอ่านตัวอักษรภาษาไทยและตัวเลขอารบิกรวมกันเท่ากับ 78.34%

ตารางที่ 2 ตารางแสดงผลการทดลองฝึกสอนของแบบจำลองโดยภาพรวม

ตัวอักษร	จำนวนทั้งหมด	จำนวนที่ถูก	ความถูกต้อง	ตัวอักษร	จำนวนทั้งหมด	จำนวนที่ถูก	ความถูกต้อง
0	24	24	100%	ฒ	1	0	0
1	42	37	88.10%	ณ	1	0	0
2	23	20	86.96%	ด	1	1	100%
3	31	15	48.39%	ต	5	1	20%
4	30	20	66.67%	ถ	1	1	100%
5	42	32	76.18%	ท	5	4	80%
6	30	22	73.32%	ธ	5	1	20%
7	37	32	86.49%	น	5	2	40%
8	38	29	76.32%	บ	19	15	78.95%
9	39	26	66.67%	ป	-	-	-
ก	51	39	76.46%	ผ	4	2	50%
ข	19	14	73.67%	ฝ	-	-	-
ฅ	-	-	-	พ	4	4	100%
ค	8	4	50%	ฟ	-	-	-
ค	-	-	-	ภ	1	0	0
ฌ	3	1	33.32%	ม	4	3	75%
ง	8	7	87.50%	ย	3	1	33.32%
จ	5	2	40%	ร	2	2	100%
ฉ	8	5	62.50%	ล	1	1	100%
ช	1	0	0	ว	3	1	33.32%
ช	-	-	-	ศ	1	1	100%
ณ	1	0	0	ษ	2	0	0
ญ	3	0	0	ส	2	2	100%
ฎ	1	1	100%	ห	2	1	50%
ฏ	-	-	-	ฬ	1	1	100%
ฐ	1	0	0	อ	1	0	0
ฑ	-	-	-	ฮ	1	0	0

จากภาพแผ่นป้ายทะเบียนในชุดข้อมูลทดสอบที่แบบจำลองตรวจจับได้และตัดภาพส่วนนั้นออกมาจำนวนทั้งหมด 92 ภาพ ในการอ่านชื่อจังหวัดจากแผ่นป้ายทะเบียนซึ่งเป็นชื่อจังหวัดในประเทศไทย 77 ชื่อ โดยที่มีชื่อจังหวัดรวมกันทั้งหมด 92 ตัวอย่าง ผลลัพธ์ในการอ่านชื่อจังหวัดจากป้ายทะเบียนสามารถสรุปได้ดังตารางที่ 3

ตารางที่ 3 ผลการทดลองจากการอ่านจังหวัดจากป้ายทะเบียน

ชื่อจังหวัด	จำนวนทั้งหมด	จำนวนที่อ่านได้	ความถูกต้อง	ชื่อจังหวัด	จำนวนทั้งหมด	จำนวนที่อ่านได้	ความถูกต้อง
กรุงเทพมหานคร	16	12	75%	เพชรบูรณ์	1	1	100%
กระบี่	1	0	0	แพร่	1	1	100%
กาญจนบุรี	1	1	100%	ภูเก็ต	1	1	100%
กาฬสินธุ์	1	1	100%	มหาสารคาม	1	1	100%
กำแพงเพชร	1	1	100%	มุกดาหาร	1	1	100%
ขอนแก่น	1	1	100%	แม่ฮ่องสอน	1	1	100%
จันทบุรี	1	1	100%	ยโสธร	1	1	100%
ฉะเชิงเทรา	1	1	100%	ยะลา	1	1	100%
ชลบุรี	1	0	0	ร้อยเอ็ด	1	1	100%
ชัยนาท	1	1	100%	ระนอง	1	1	100%
ชัยภูมิ	1	1	100%	ระยอง	1	1	100%
ชุมพร	1	0	0	ราชบุรี	1	1	100%
เชียงราย	1	1	100%	ลพบุรี	1	0	0
เชียงใหม่	1	1	100%	ลำปาง	1	1	100%
ตรัง	1	1	100%	ลำพูน	1	0	0
ตราด	1	1	100%	เลย	1	1	100%
ตาก	1	0	0	ศรีสะเกษ	1	1	100%
นครนายก	1	1	100%	สกลนคร	1	1	100%
นครปฐม	1	1	100%	สงขลา	1	1	100%
นครพนม	1	1	100%	สตูล	1	1	100%
นครราชสีมา	1	1	100%	สมุทรปราการ	1	1	100%
นครศรีธรรมราช	1	0	0	สมุทรสงคราม	1	1	100%
นครสวรรค์	1	1	100%	สมุทรสาคร	1	1	100%
นนทบุรี	1	1	100%	สระแก้ว	1	0	0
นราธิวาส	1	0	0	สระบุรี	1	0	0
น่าน	1	1	100%	สิงห์บุรี	1	1	100%
บึงกาฬ	1	1	100%	สุโขทัย	1	0	0

ชื่อจังหวัด	จำนวนทั้งหมด	จำนวนที่อ่านได้	ความถูกต้อง	ชื่อจังหวัด	จำนวนทั้งหมด	จำนวนที่อ่านได้	ความถูกต้อง
บุรีรัมย์	1	1	100%	สุพรรณบุรี	1	1	100%
ปทุมธานี	1	1	100%	สุราษฎร์ธานี	1	1	100%
ประจวบคีรีขันธ์	1	1	100%	สุรินทร์	1	1	100%
ปราจีนบุรี	1	1	100%	หนองคาย	1	0	0
ปัตตานี	1	1	100%	หนองบัวลำภู	1	0	0
พะเยา	1	1	100%	อ่างทอง	1	1	100%
พระนครศรีอยุธยา	1	1	100%	อำนาจเจริญ	1	0	0
พังงา	1	1	100%	อุดรธานี	1	1	100%
พัทลุง	1	1	100%	อุดรดิษฐ์	1	0	0
พิจิตร	1	1	100%	อุทัยธานี	1	1	100%
พิษณุโลก	1	1	100%	อุบลราชธานี	1	1	100%
เพชรบุรี	1	1	100%				

จากตารางที่ 3 จะเห็นว่าจังหวัดกรุงเทพมหานครเพียงจังหวัดเดียวนั้นมีจำนวนถึง 16 ตัวอย่าง เนื่องจากในการทดสอบการอ่านตัวอักษรภาษาไทยจากแผ่นป้ายทะเบียน ต้องการตัวอย่างข้อมูลของตัวอักษรภาษาไทยที่หลากหลาย ซึ่งมีเพียงจังหวัดกรุงเทพมหานครเท่านั้นที่ได้รับหมวดหมู่ตัวอักษรของป้ายทะเบียนจนครบตามที่กำหนด ด้วยข้อจำกัดของฐานข้อมูลรูปถ่ายป้ายทะเบียนของจังหวัดต่าง ๆ และมีการนำตัวเลขอารบิกมาใส่ไว้หน้าหมวดหมู่ตัวอักษรและเริ่มรันหมวดหมู่ตัวอักษรใหม่ โดยผลลัพธ์ของการอ่านชื่อจังหวัดด้วยวิธีการอ่านอักขระด้วยแสง และใช้วิธีการหาค่าระยะทางเลเวนชเตย์นเทียบระหว่างชื่อจังหวัดที่เป็นต้นแบบ มีค่าความถูกต้องเฉลี่ยเท่ากับ 79.35%

5. สรุปผลการวิจัย

งานวิจัยนี้ได้นำแบบจำลอง YOLOv5 มาฝึกสอนเพื่อใช้ในการตรวจจับป้ายทะเบียนจากยานพาหนะ โดยได้เลือกแบบจำลอง YOLOv5 มาฝึกสอนทั้งหมด 3 แบบด้วยกัน โดยพบว่าแบบจำลอง YOLOv5m นั้นมีประสิทธิภาพมากที่สุด แต่แบบจำลองที่เลือกนำมาใช้งานคือแบบจำลอง YOLOv5s เนื่องจากมีความรวดเร็วในการตรวจจับที่มากกว่าซึ่งแลกมาด้วยประสิทธิภาพที่ลดลงเพียงเล็กน้อย โดยที่ประสิทธิภาพของแบบจำลอง YOLOv5s ในการตรวจจับป้ายทะเบียนจากยานพาหนะโดยใช้ชุดข้อมูลทดสอบคิดเป็น 98.38 เปอร์เซ็นต์ จากนั้นจึงนำรูปภาพป้ายทะเบียนที่ตรวจจับได้มาทำการประมวลผลภาพด้วยวิธีการต่าง ๆ ก่อนจะใช้วิธีการอ่านอักขระด้วยแสงโดยใช้ Tesseract อ่านตัวอักษรจากแผ่นป้ายทะเบียน พร้อมทั้งได้มีการนำวิธีการวัดระยะทางเลเวนชเตย์น เข้ามาช่วยในการตรวจสอบและแก้ไขชื่อจังหวัดที่ผิดให้ถูกต้องด้วย

จากการคำนวณค่าความถูกต้องในการอ่านตัวอักษรจากป้ายทะเบียนโดยคิดเป็นเปอร์เซ็นต์พบว่าในส่วนของหมายเลขและตัวอักษรของป้ายทะเบียนมีค่าความถูกต้องเท่ากับ 71.93 เปอร์เซ็นต์ และในส่วนชื่อจังหวัดมีค่าความถูกต้องเท่ากับ 72.5 เปอร์เซ็นต์ ซึ่งสรุปได้ว่า Tesseract นั้นมีข้อจำกัดในการอ่านชื่อ

จังหวัด แม้ว่าจะใช้วิธีการวัดระยะทางเลเวนชเตย์น ของตัวอักษรที่อ่านได้ กับฐานข้อมูลรายชื่อจังหวัด ผลลัพธ์ความถูกต้องยังไม่เป็นที่น่าพอใจนัก นอกจากนี้ยังมีปัจจัยที่ส่งผลต่อประสิทธิภาพของการตรวจจับและอ่านป้ายทะเบียน ได้แก่ การใช้ภาพที่ถ่ายในช่วงเวลากลางคืน เนื่องจากแสงภายในภาพมีปริมาณน้อยจนเกินไป และแผ่นป้ายทะเบียนชำรุดหรือมีคราบต่าง ๆ ติดอยู่ ส่งผลให้ตัวหนังสือบนป้ายทะเบียนไม่ชัดเจน

6. เอกสารอ้างอิง

- [1] RGB color model. (2023, July 10). In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/RGB_color_model
- [2] Binary image. (2023, May 28). In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Binary_image
- [3] Thresholding (image processing). (2023, February 19). In *Wikipedia*. Retrieved from [https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))
- [4] Transformation matrix. (2023, April 21). In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Transformation_matrix
- [5] OpenCV: Geometric Transformations of Images. (n.d.). Docs.opencv.org. Retrieved from https://docs.opencv.org/3.4/da/d6e/tutorial_py_geometric_transformations.html
- [6] ชมพูนุท ไชยประสิทธิ์. (2561) การประยุกต์ใช้เทคโนโลยีการอ่านป้ายทะเบียนในการติดตามรถยนต์ที่ถูกโจรกรรม. (ปริญญาานิพนธ์ปริญญามหาบัณฑิต, สถาบันเทคโนโลยีไทย-ญี่ปุ่น, กรุงเทพฯ). สืบค้นจาก <http://library.tni.ac.th/thesis/upload/files/MET%202018/Chompunoot%20Chaiprasit%20Thesis%20MET%202018.pdf>
- [7] Sharma, A. (2022, July 14). Mean Average precision (MAP) using the COCO evaluator - PyImageSearch. PyImageSearch. Retrieved from <https://pyimagesearch.com/2022/05/02/mean-average-precision-map-using-the-coco-evaluator/>
- [8] Levenshtein, Vladimir I. (February 1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*. 10 (8): 707–710.
- [9] ชัยธวัช มินเดินเรือ. (2559) Automatic License Plate Recognition System. (ปริญญาานิพนธ์ปริญญามหาบัณฑิต, มหาวิทยาลัยเทคโนโลยีมหานคร, กรุงเทพฯ)
- [10] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, ... Prashant Rai. (2020, October 29). ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements (Version v3.1). Zenodo. Retrieved from <https://doi.org/10.5281/zenodo.4154370>