YOLO-BASED IMAGE ANALYSIS AND TRAFFIC FLOW ESTIMATION IN URBAN AREA USING BETWEENNESS CENTRALITY

JAKKRIT SUTTANURUK

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Civil, Transportation
and Geo-Resources Engineering
Suranaree University of Technology
Academic Year 2022

การวิเคราะห์ภาพด้วยโยโลและการประมาณกระแสจราจรในเขตเมือง โดยวิธีค่าความเป็นศูนย์กลางจากการคั่นกลาง

นายจักรกฤษณ์ สุทธานุรักษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมโยธา ขนส่ง และทรัพยากรธรณี
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2565

YOLO-BASED IMAGE ANALYSIS AND TRAFFIC FLOW ESTIMATION IN URBAN AREA USING BETWEENNESS CENTRALITY

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Degree of Doctor of Philosophy.

Thesis Examining Committee

....

(Prof. Dr. Thaned Satiennam)

Chairperson

....

(Prof. Dr. Vatanavongs Ratanavaraha)

Member (Thesis Advisor)

....

(Asst. Prof. Dr. Sajjakaj Jomnonkwao)

Member (Thesis Co-Advisor)

....

(Asst. Prof. Dr. Rattanaporn Kasemsri)

Member

....

(Dr. I-Soon Raungratanaamporn)

Member

....

(Asst. Prof. Dr. Sarunya Kanjanawattana)

Member

....

(Asst. Prof. Dr. Thanapong Champahom)

Member

....

(Assoc. Prof. Dr. Chatchai Jothityangkoon)

Vice Rector for Academic Affairs and Quality Assurance

....

(Assoc. Prof. Dr. Pornsiri Jongkol)

Dean of Institute of Engineering

จักรกฤษณ์ สุทธานุรักษ์ : การวิเคราะห์ภาพด้วยโยโลและการประมาณกระแสจราจรในเขตเมืองโดยวิธีค่าความเป็นศูนย์กลางจากการคั่นกลาง (YOLO-Based Image Analysis and Traffic Flow Estimation in Urban Area Using Betweenness Centrality)
อาจารย์ที่ปรึกษา: ศาสตราจารย์ ดร.วัฒนวงศ์ รัตนวราห, 72 หน้า

คำสำคัญ: คอมพิวเตอร์วิทัศน์, ทฤษฎีกราฟจากหลักการความเป็นจุดศูนย์กลาง

อุบัติเหตุทางถนนเป็นสาเหตุหลักของการบาดเจ็บและการเสียชีวิตที่เกิดขึ้นทั่วโลก เฉกเช่นเดียวกับประเทศไทย ซึ่งเป็นประเทศที่ติดการจัดอันดับอุบัติเหตุทางถนนที่ร้ายแรงที่สุดเป็นเวลาหลายปี โดยอยู่ในลำดับที่ 8 ในปี พ.ศ. 2561 รถโดยสารสองแถวที่พัฒนามาจากรถกระบะดัดแปลงนั้นเป็นหนึ่งในยานยนต์สาธารณะที่เมื่อเกิดอุบัติเหตุแล้ว มักจะก่อให้เหตุอุบัติเหตุร้ายแรงต่อชีวิตของผู้โดยสาร เนื่องจากรถสองแถวนั้นปราศจากอุปกรณ์นิรภัยต่าง ๆ เช่น เข็มขัดนิรภัย ถุงลมนิรภัย รวมไปถึงการบรรทุกผู้โดยสารเกินอัตราและปล่อยให้มีการห้อยโหนเป็นต้น การศึกษานี้มีวัตถุประสงค์ที่จะจัดการกับปัญหาดังกล่าว โดยใช้เทคนิคการเรียนรู้เชิงลึกร่วมกับภาพวิดีโอจากกล้องวงจรปิดจราจร หรือ CCTV (Closed-Circuit Television) ที่มีอยู่ทั่วไปตามท้องถนน มาเก็บข้อมูลจราจร จำแนกประเภทรถสี่ล้อ (Passenger Cars) และนำมาจำแนกรถสองแถวโดยสารปกติ (PTPT – Public Pickup Trucks) ออกจากรถสองแถวบรรทุกผู้โดยสารเกินอัตรา (Overcrowded PTPT) เพื่อช่วยลดอุบัติเหตุ

เนื่องจากลักษณะทางกายภาพของรถโดยสารสองแถวและการบรรทุกผู้โดยสารนั้น มีลักษณะจำเพาะเจาะจง จึงจำเป็นที่จะต้องพัฒนาโมเดลการเรียนรู้เชิงลึกใหม่ขึ้นมา เพื่อใช้ในการจำแนกรถกระบะโดยสารทั้งสองแบบออกจากกัน โดยในการศึกษานี้ ผู้วิจัยได้ใช้ YOLOv5 ซึ่งเป็นโมเดลสำหรับตรวจจับวัตถุที่สนใจที่มีประสิทธิภาพที่ดีที่สุดในเวลานั้น (State-of-the-art object detection) เป็นตัวต้นแบบในการพัฒนา ทั้งนี้ผู้วิจัยได้สร้างชุดข้อมูล (dataset) ขึ้นมาเพื่อพัฒนาโมเดลการเรียนรู้เชิงลึกที่สามารถตรวจจับการบรรทุกเกินอัตราของรถสองแถวได้ โดยโมเดลดังกล่าวนั้นให้ค่าเฉลี่ยของความแม่นยำ (mean average precision – mAP) ที่ 95.1% และใช้เวลาประมวลผลภาพ (inference time) ที่ 33 เฟรมต่อวินาที บนหน่วยประมวลผลกราฟิกส์ (Graphics Processing Unit – GPU) การศึกษานี้ได้สร้างประโยชน์ต่าง ๆ ได้แก่ การพัฒนาชุดข้อมูลภาพรถโดยสารสองแถวแบบปกติและแบบเกินอัตรา การพัฒนากรอบแนวทางการปรับค่าที่เหมาะสมที่สุดของโมเดลต่าง ๆ สำหรับงานจำแนกวัตถุ นอกจากนี้ โมเดลที่เหมาะสมที่สุดที่ได้รับการคัดเลือกนั้นมีแนวโน้มในการนำไปใช้บนคอมพิวเตอร์ที่เป็นส่วนหนึ่งของศูนย์ควบคุมจราจร เพื่อช่วยแจ้งเตือนตำรวจจราจรหาก

ตรวจพบการฝ่าฝืนบรรทุกผู้โดยสารห้อยโหนเกินอัตรา อันจะเป็นการช่วยลดการเกิดอุบัติเหตุที่ร้ายแรงถึงแก่ชีวิต

อย่างไรก็ดี ข้อมูลปริมาณจราจรจากกล้อง CCTV ที่จำแนกรถสี่ล้อและรถสองแถวโดยสารสาธารณะที่จะถูกนำมาใช้เป็นข้อมูลขาเข้า (input data) สำหรับการทำนายปริมาณจราจรในขั้นต่อไปนั้น ได้ถูกทดแทนด้วยข้อมูลปริมาณจราจรจากเซนเซอร์ที่ได้จากจุดสำรวจของกรมทางหลวง เนื่องจากข้อจำกัดบางประการ เช่น งานก่อสร้างและบำรุงทางทำให้มีการรื้อถอนกล้องหลายจุด โดยงานวิจัยนี้ได้เสนอวิธีทางเลือกใหม่ที่ง่ายและประหยัดในการทำนายปริมาณกระแสจราจร (traffic flow) โดยใช้ทฤษฎีกราฟและค่าระหว่างกลาง (Betweenness Centrality – BC) ซึ่งเป็นวิธีที่ใช้เวลาน้อยกว่าและสิ้นเปลืองน้อยกว่าวิธีต่าง ๆ ที่ใช้คาดคะเนปริมาณกระแสจราจรที่ใช้อยู่ในปัจจุบัน ซึ่งโดยมากมักเป็นกระบวนการที่ใช้เวลานานและมีราคาสูง

ผู้วิจัยได้เสนอการนำ BC มาประยุกต์ใช้ร่วมกับดัชนีการจราจรติดขัดหรือ TCI (Traffic Congestion Index) โดยอาศัยข้อมูลจราจรของ Google เพื่อใช้ทำนายกระแสจราจร งานวิจัยนี้ใช้โครงข่ายถนนของเขตเทศบาลนครนครราชสีมาเป็นกรณีศึกษาเพื่อประเมินประสิทธิภาพของวิธีที่ผู้วิจัยนำเสนอนี้ โดยผลการศึกษาชี้ให้เห็นว่า ค่าสหสัมพันธ์ระหว่างปริมาณกระแสจราจรจากการทำนายกับค่า TCI นั้นสูงกว่า 0.58 นั่นแสดงว่าวิธีที่ผู้วิจัยนำเสนอนั้นสามารถใช้อธิบายการเคลื่อนที่ของการจราจรในแต่ละช่วงถนนได้อย่างมีประสิทธิภาพ

โดยสรุปแล้ว การศึกษานี้มีส่วนสร้างประโยชน์ทั้งในด้านความปลอดภัยทางการจราจรและการทำนายกระแสจราจร และผู้วิจัยหวังว่าการศึกษานี้จะส่งผลกระทบเชิงบวกต่อการลดอุบัติเหตุทางถนนรวมถึงการพัฒนาการบริหารงานจราจรให้ดียิ่งขึ้น

สาขาวิชา วิศวกรรมขนส่ง
ปีการศึกษา 2565

ลายมือชื่อนักศึกษา..
ลายมือชื่ออาจารย์ที่ปรึกษา..
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม..

PANUWAT WISUTWATTANASAK : YOLO-BASED IMAGE ANALYSIS AND TRAFFIC FLOW ESTIMATION IN URBAN AREA USING BETWEENNESS CENTRALITY. THESIS ADVISOR: PROF. VATANAVONGS RATANAVARAHA, Ph.D., 72 PP.

Keyword: Convolutional Neural Networks, Centrality Graph Theory

Road accidents are a significant cause of death and injury worldwide, and Thailand is no exception. It has been on the World Health Organization's list of deadliest roads for several years, ranking eighth in 2018. Among all road fatalities, pickup trucks converted into public transportation vehicles are found to be the most problematic, as they often lack proper passenger safety measures and frequently carry excessive passengers. To address this issue, this study proposes the use of deep learning techniques and existing closed-circuit television (CCTV) traffic cameras to classify overloaded public transport pickup trucks (PTPT) and help reduce accidents.

As the characteristics of PTPT vehicles and their passenger occupancy are unique, it is necessary to develop a new model to accurately identify these vehicles. In this study, various state-of-the-art object detection YOLOv5 (You Only Look Once) models were investigated, and a custom dataset was created to obtain the optimal overcrowded detection model. The results showed that the YOLOv5L model yielded the best performance with a mean average precision (mAP) of 95.1% and an inference time of 33 frames per second (FPS) on a GPU. This study contributes to the field by making the custom dataset available and by providing insights into the optimization of the object detection models. The selected model is expected to be deployed on traffic control computers to alert the police of passenger-overloading violations and help reduce traffic-related fatalities.

In addition to the PTPT classification, passenger cars are also classified and counted for use in the traffic estimation task. However, the data gathering through CCTV has been replaced by the data obtained from sensors installed by the Department of Highways. This change was implemented due to certain limitations associated with the absence of multiple CCTVs during road construction. This study further presents an introduction to a straightforward and cost-effective method that

utilizes graph theory and betweenness centrality (BC) for estimating traffic flow. Current methods for measuring traffic volume are often time-consuming and expensive, so this study proposes a shorter and less costly approach. The proposed method integrates BC and the traffic congestion index (TCI) based on Google traffic data to estimate traffic flow. A case study was conducted in the road network of central Nakhon Ratchasima, Thailand, to evaluate the effectiveness of the proposed method. The results showed that the correlation between the estimated traffic volume and the TCI was higher than 0.58, indicating that the proposed method could effectively describe the movement of traffic through the road sections.

Overall, this study provides new and valuable contributions to both the field of traffic safety and traffic flow estimation, and we expect that our work will have a positive impact on reducing road accidents and improving traffic management in Thailand.

School of Transportation Engineering
Academic Year 2022

Student's Signature..............................
Advisor's Signature..............................
Co-Advisor's Signature..............................

ACKNOWLEDGEMENT

I would like to pay great respects to persons, group of people who give good recommendation and help me to contribute both in academic and research work as mentioned illustrations:

First and foremost, I would like to express my sincere appreciation to my thesis advisors, Prof. Dr. Vatanavongs Ratanavaraha, Assistant Professor Dr. Sajjakaj Jomnonkwao, and Assistant Professor Sarunya Kanjanawattana. Their guidance and support throughout my research journey have been invaluable. Their valuable insights and suggestions have greatly contributed to the improvement of my research and academic knowledge. I am grateful for their dedication and commitment to my thesis. Working under their mentorship has been an honor, and I sincerely appreciate their contributions to my academic growth.

I would like to extend my heartfelt gratitude to Ms. Wanpen Suebsai and Ms. Piyatida Punharach, the Secretaries of Transportation Engineering, for their invaluable coordination efforts throughout my study. I am sincerely thankful to Suranaree University of Technology for granting me a doctoral scholarship and for their generous support through the External Grants and Scholarships for Graduate Students (OROG) program, which significantly contributed to covering my tuition fees. Their support has been instrumental in facilitating my academic journey, and I am truly appreciative of their assistance. I would also like to express my gratitude to the people at the Department of Highways for their provision of the traffic data. Their cooperation has been essential to the success of my research.

Lastly, I would like to express my deep gratitude to my parents, wife, and beloved daughter for their unwavering love, support, and belief in my educational journey. Their encouragement and sacrifices have played a pivotal role in my academic accomplishments. I am genuinely thankful for their presence in my life and the invaluable support they have provided me. Their enduring love and support have consistently motivated me, and I am truly fortunate to have such a caring and supportive family by my side.

Jakkrit Suttanuruk

TABLE OF CONTENTS

Page

ABSTRACH (THAI) I
ABSTRACH (ENGLISH) III
ACKNOWLEDGEMENT V
TABLE OF CONTENTS VII
LIST OF TABLES IX
LIST OF FIGURES X
LIST OF ABBREVIATIONS XI
CHAPTER
1 INTRODUCTION 1
1.1 Rationale and Purpose of the Research 7
1.2 Purpose of the Research 8
1.3 Research Questions 8
1.4 Research Contribution 9
2 LITERATURE REVIEW 10
2.1 Computer Vision for Data Collection 10
2.2 Traffic estimation using Modified Betweenness Centrality 13
2.2.1 Traffic Flow Estimation 13
2.2.2 BC (Betweenness Centrality) 14
2.2.3 Google Traffic Data 15
3 MATERIALS AND METHODS 18
3.1 Vehicle Detection based on YOLO 19
3.1.1 Datasets 21
3.1.2 YOLOv5 Architecture 23
3.1.3 Finetuning 24
3.1.4 Performance Metrics 25

TABLE OF CONTENTS (Continued)

Page

3.2 Traffic Flow Estimation using the Proposed Enhanced Betweenness Centrality 26

3.2.1 Study Area 26

3.2.2 Traffic Distribution Based on The BC Method 29

3.2.3 Betweenness Centrality (BC) 29

3.2.4 Traffic Congestion Index (TCI) 30

3.2.5 Statistical Analysis 31

3.2.6 Traffic Volume Estimation 31

4 RESULTS AND DISCUSSION 33

4.1 CNN 33

4.2 BC 39

5 CONCLUSIONS 43

REFERENCES 46

APPENDIX

APPENDIX A List of Publications 55

BIOGRAPHY 71

LIST OF TABLES

Table Page

2.1 Summary of related studies. ... 13

3.1 Summary of previous studies on WTP for risk reduction and findings. ... 20

3.2 An Example of the TCI of Different Road Sections ... 31

4.1 Evaluation Metrics for Different YOLOv5 Models ... 33

4.2 Inference Speed Comparison. ... 35

LIST OF FIGURES

Figure **Page**

2.1 Theory of planned behavior model. (Ajzen, 1985) ... 16
2.2 The Relationship Between Vehicle Speed and Traffic Density ... 17
3.1 Overview of Framework ... 19
3.2 Process Flow Diagram ... 20
3.3 S × S Grid ... 21
3.4 Public Transport Pickup Trucks (PTPT) ... 22
3.5 Overcrowded Public Transport Pickup Trucks (Overcrowded PTPT) ... 22
3.6 Model Structure of YOLOv5L (You Only Look Once) ... 24
3.7 Methodological Framework ... 26
3.8 Study Area, City Center of Nakhon Ratchasima, Thailand ... 27
3.9 Nodes ... 28
3.10 Road Network ... 28
3.11 Nodes and Road Sections ... 29
3.12 Example Calculations of Route Section 10 ... 32
4.1 Precision, Recall, and Mean Average Precision on Five YOLOv5 Models ... 34
4.2 Train and Validation Loss ... 35
4.3 Precision-Recall Curve of YOLOv5L ... 36
4.4 F1 Plot of YOLOv5L ... 37
4.5 Confusion Matrix of YOLOv5L ... 37
4.6 Overcrowded Detection by YOLOv5L (Online, 2019) ... 38
4.7 Multiclass Detection by YOLOv5L (Online, 2016) ... 38
4.8 Selected Routes ... 39
4.9 Estimated Traffic Volume of Different Road Sections ... 40
4.10 Correlation of Estimated Traffic Volume Based on BC and TCI ... 41
4.11 Level of the Estimated Traffic Volume at 50% Detour Ratio ... 41

LIST OF ABBREVIATIONS

α	=	Statistically significant level
$\sigma_{mp}(n)$	=	Total number of shortest paths from node 'm' to node 'p' that pass through 'n'
AP	=	Average precision
API	=	Application programming interface
BC	=	Betweenness centrality
CCTV	=	Closed-circuit television
CNN	=	Convolutional neural network
CPU	=	Central processing unit
GPS	=	Global positioning system
GPU	=	Graphics processing unit
HOG	=	Histogram of oriented gradients
FP	=	False positive
FPS	=	Frame per second
FN	=	False negative
IOU	=	Intersection over union
ITS	=	Intelligent transportation systems
mAP	=	Mean average precision
NOx	=	Nitrogen oxides
O-D	=	Origin-destination
PTPT	=	Public transport pickup truck
PM	=	Particulate matter
R-CNN	=	Region-based convolutional neural network
ReLU	=	Rectified linear unit
RMSE	=	Root mean square error
SOA	=	State-of-the-art
SPP	=	Spatial pyramid pooling

LIST OF ABBREVIATIONS (Continued)

SVM	=	Support vector machine
TCI	=	Traffic congestion index
TN	=	True negative
TP	=	True positive
V_{al}	=	Total traffic volume on the main route ($_{l}$)
VOCs	=	Volatile organic compounds
WHO	=	World health organization
YOLO	=	You only look once
LL	=	Log-Likelihood
SD	=	Standard deviation

CHAPTER 1

INTRODUCTION

Thailand is again included in the World Health Organization (WHO)'s infamous list of the deadliest traffic fatalities worldwide. In a 2018 global status report on road safety, the WHO ranked Thailand 8th out of 175 other countries, an improvement from the previous two surveys in which Thailand ranked 2nd (World Health Organization, 2018). In 2018, the death toll from road accidents was determined to be 22,491 people, with a fatality rate of around 32.7 per 100,000 people. The number could be significantly worse than it appears to be due to data collection incompetency (Lurstwut, Phothiyod, & Ounban, 2018). Fatalities involving public transportation are typically high in terms of occupancy size (18,584 accidents in 2018, a 33.79% increase from the previous year) (Lurstwut et al., 2018). As of April 2022, motorcycles accounted for 52.7% (i.e., 21.8 million) of all vehicles registered in Thailand, followed by sedans (26.6%) and light pickup trucks (17%) (Transportation, 2021). One reason pickup trucks are in high demand is that the government state oil fund subsidizes 50% of every increase in diesel price. Moreover, biodiesel B20, which is a less expensive alternative to regular diesel, has become commercially viable and was endorsed by the Energy Ministry in 2019. This has resulted in the increase in terms of the market share of 1-ton trucks. As previously stated, pickup trucks are cost effective and therefore are modified to serve as a mode of public transportation.

Approximately 100,000 public transport pickup trucks (PTPT) are currently registered with the Department of Land Transport, which is equivalent to the US Department of Motor Vehicles. These trucks have taken over the streets of the country, displacing regular buses. Depending on the structure and model, PTPTs can accommodate eight seated passengers and four to six standing passengers. Most truck operators are private companies that must pay additional taxes and fees to operate on designated routes. Therefore, the operators often carry as many passengers as they can. Typically, this means that they wait for passengers to arrive and fill the vehicle.

This results in being behind schedule. Thus, they must exceed speed limits, breaking numerous traffic laws and increasing the risk of injury and death. The Thai traffic authority has a limited number of highway patrols to conduct traffic stops. Furthermore, they rely on roadblocks and closed-circuit television (CCTV) cameras to enforce traffic laws to minimize road accidents.

The proposed framework aims to address the increasing number of road accidents caused by overcrowded pickup trucks (Transportation, 2022) in urban areas and traffic congestion that leads to delays and increased air pollution. There is scientific evidence that traffic congestion leads to increased air pollution. Studies have shown that traffic-related emissions, including those from vehicles stuck in traffic, are a significant source of air pollution. These emissions include particulate matter (PM), nitrogen oxides (NOx), and volatile organic compounds (VOCs), which have been linked to a variety of negative health outcomes such as respiratory and cardiovascular disease (Alobaidi, Badri, & Salman, 2020; Li & Managi, 2021). Studies have also shown that traffic congestion causes increased idling, which leads to higher levels of emissions and increased exposure to pollutants for people living and working near congested roads. Additionally, traffic congestion can lead to increased traffic volume, which can result in more emissions and higher overall pollution levels. Overall, traffic congestion is a major contributor to air pollution, and reducing it can help to improve air quality and promote public health (Bernardo, Fageda, & Flores-Fillol, 2021).

There are two broad categories related to overloading detection: physical sensors and computer vision. Sensors are used in intelligent transportation systems (ITS), such as Bluetooth, laser, radio wave, infrared, etc., to sense the weight of the vehicles. The use of weight sensors combined with loop induction sensors necessitates physical installations, i.e., opening a road surface and blocking traffic. Computer vision can use existing cameras. In case of a new installation, tampering with the road surface and road blocking is minimal. Recently, the use of computer vision has been developed and made available. Compared with ITS solutions, they are less expensive and easier to maintain.

In this study, we focus on computer vision category. Classic machine learning methods, such as feature extraction using histogram of oriented gradients (HOG), classification using support vector machine (SVM), and many others have been used and achieved satisfactory results. In recent years, deep learning techniques have become widely popular, especially since the introduction of various convolution neural network (CNN) algorithms in 2012. Hardware, particularly graphic processing units (GPU), also plays a significant role in reducing the time required to train a model because they are much faster than central processing units (CPU) (Kayid, 2018). However, a more powerful CPU comes at a higher cost and consumes more power. One could opt for online services for alternative cost-efficient solutions, such as Google Colab (Carneiro et al., 2018). As a result, image classification has become more accurate. Specifically, vehicle classification in the modern era is accomplished using deep learning techniques, such as a CNN with the transfer learning method coupled with the power of modern GPUs, which allows for a significant reduction in training time.

The framework employs the use of convolutional neural networks to identify and classify public pickup trucks among other vehicle types, by analyzing image data extracted from video frames obtained from various traffic cameras across the targeted network. CNN has been widely used in image and video analysis tasks, including object detection and image classification. In the context of vehicle detection and categorization, CNNs can be used to analyze images from traffic cameras and identify different types of vehicles, such as cars, trucks, buses, and motorcycles. The CNN algorithm can also be trained to detect specific characteristics of vehicles, such as license plates, make and model, and number of occupants (Gholamhosseinian & Seitz, 2021). Utilizing the pretrained network is considered highly effective because it has previously seen a tremendous number of images with thousands of categories. The mentioned CNN model subsequently adds extra interested classes to the last few layers of the network to enable its classification of newly added targets. In this case, the images of Thai-modified pickups and passenger-overloaded trucks are inputted into the networks to achieve the overcrowded PTPT classifier. However, the inference speed is slow, and we need to find bounding boxes or the locations of the objects of interest, i.e., one regular PTPT and two overcrowded ones, in the vicinity of the image from a CCTV video feed.

Region-based CNNs, such as R-CNN, have been proposed to identify the bounding box of an object of interest by splitting an image into regions. However, this approach is time-consuming. Fast R-CNN and Faster R-CNN are subsequently released to overcome the inference time issues. These methods are classified as two-stage detectors because they comprise, first, the generation of region proposals by selective search as in the R-CNN or by a Region Proposal Network and, second, object classification. Faster R-CNN is over 100 times faster than the traditional R-CNN. However, single-stage detectors, such as the YOLO (You Only Look Once) and its variants, such as the You Only Learn One Representation (Wang, Yeh, & Liao, 2021) and the YOLOX (Ge, Liu, Wang, Li, & Sun, 2021), an anchor-free version of the YOLO, have recently emerged by directly applying object classification and bounding box regression (Carranza-García, Torres-Mateo, Lara-Benítez, & García-Gutiérrez, 2021). Generally, two-stage detectors are more accurate but slower than single-stage detectors, such as the YOLO.

Thus, in this study, we propose a way to accurately detect overcrowded PTPT in real time providing valuable information to be fed into other system in the pipeline and also for policy makers to improve public pickup truck safety and traffic flow in the area. Therefore, the algorithm should have an acceptable accuracy and inference speed in the said detection task when using existing traffic cameras. Currently, there is no model trained specifically for this task. Therefore, to bridge this gap, we propose training a model to detect the PTPT to ensure road safety and reduce risks associated with public transport. Many peer-reviewed papers recommend the YOLOv5 for real-time object detection tasks (J. a. Kim, Sung, & Park, 2020; Kwan et al., 2020; Nepal & Eslamiat, 2022). We have compared all the pretrained YOLOv5 (version 6) models, such as models N, S, M, L, and X, to find an acceptable one to classify a unique characteristic of Thai public vehicles and their usage concerning public safety. Precisely, the model aims to identify whether or not public pickup trucks used for transportation are overcrowded. We have used the transfer learning technique on our manually labeled 1,239 images in the YOLOv5 PyTorch format (Paszke et al., 2019) in order to recognize the overcrowded and the normal ones. The implementation of the model is envisioned to assist in the enforcement of traffic regulations, leading to a decrease in fatal traffic incidents.

By incorporating artificial intelligence automation, this framework also decreases the dependence on human resources in the traffic accident prevention task force (Eraqi et al., 2021). The YOLO model, as described above, facilitates the analysis of video footage captured by traffic cameras by enabling automatic detection and classification of vehicles, including the identification of overcrowded vehicles and license plates. This reduces the need for manual surveillance and intervention. To further enhance the traffic flow analysis, the results of the traffic count obtained using YOLO and the distinction between passenger cars and other vehicles can be utilized in graph theory calculations.

Having successfully automated the collection of traffic data, the next step is to utilize it in graph network calculations for estimating traffic flow patterns. The flow of traffic from one node to another in a road network is influenced by factors such as the network's topology and travel demand characteristics. To understand and analyze this flow, and to estimate the traffic volume for each road section, the concept of Betweenness Centrality (BC) was used. BC is a measure of the importance of a node in a graph based on the number of shortest paths that pass through it (Mahmoody, Tsourakakis, & Upfal). In the context of transportation network analysis, BC can be used to identify the most important nodes or links in the network in terms of their role in connecting other nodes. When applied to an origin-destination matrix, betweenness centrality can be used to identify the most important locations or transportation hubs in terms of their connectivity to other locations. For example, a high betweenness centrality value for a particular location may indicate that it is a major transportation hub or a key location for connecting different parts of the network (Altshuler, Puzis, Elovici, Bekhor, & Pentland, 2012). BC can also be a measure that quantifies the relationship between traffic flow and the roadway network structure using graph theory principles. It is based on the assumption that drivers prefer the shortest route to reach their destination (Ali & Abid, 2021). BC is expressed as a percentage of all shortest routes that pass through specific sections or nodes of interest, serving as an indicator of their centrality in the defined road network (Freeman, 1977). The theory behind BC and traffic flow estimation assumes that drivers will most likely choose the shortest route between their origin and destination (O-D). As a result, sections or nodes with higher BC indices experience higher traffic demand, resulting in higher traffic volume.

According to previous studies Hillier, Penn, Hanson, Grajewski, and Xu (1993), Jiang (2009), and Henry, Bonnetain, Furno, Faouzi, and Zimeo (2019), Betweenness Centrality (BC) has been found to be an effective tool for characterizing the flow of traffic from one node to another. The results of the study by (Henry et al., 2019) showed a strong correlation between the estimated traffic flow using BC and the actual observed traffic volume for each route section. However, other studies have highlighted that BC alone may not provide a complete picture of traffic movement (Gigerenzer, 2008). There have been instances where the estimated traffic flow using BC has been found to be significantly different from the actual traffic volume (Gao, Wang, Gao, & Liu, 2013). This highlights the fact that other factors, such as familiarity with a particular route or personal interests in certain activities, also play a role in drivers' route selection decisions, not just the shortest route. Therefore, the question remains whether BC can effectively explain the traffic flow in a road network. While BC has been widely used for traffic flow analysis, it has limitations in terms of considering other important factors such as traffic demand, speed, travel time, and road capacity (Kazerani & Winter, 2009). To overcome these limitations, researchers have proposed modifications to the traditional BC method. In a study by Ye, Wu, and Fan (2016), they proposed a BC-based method combined with geographic information to predict traffic flow. The results of the study showed that their modified BC method was effective in estimating the traffic flow in a road network. It can be concluded that for accurate traffic flow estimation, BC should be combined with actual traffic demand on each road section. The volume of pass-by trips on a road section can be estimated from the average traffic volume of origin-destination (O-D) trips. However, the challenge remains in determining the volume of O-D trips.

In this research, a modified BC approach is introduced to estimate traffic flow in the town center of Nakhon Ratchasima, Thailand, which incorporates real-time Google traffic data and the traffic congestion index (TCI). The aim is to provide a useful tool for traffic management and decision-making, as well as to improve road safety in urban areas by utilizing existing public data sources, such as CCTV traffic cameras and Google traffic data (Kumarage, Bandara, & De Silva, 2018) , especially in the situation where traffic data collection is not fully available. This approach is expected to be cost-effective and scalable, as it relies on existing resources rather than requiring new equipment. The implementation of this method will offer a comprehensive understanding of traffic conditions in the town center, enabling effective management of traffic congestion and reduction of hotspots.

1.1 Rationale and Purpose of the Research

This research is motivated by the need to improve traffic management in urban areas. Urbanization and population growth have led to significant traffic congestion in cities worldwide. The goal of this research is to address this problem by creating a framework for policy makers that uses advanced technology aiming to provide a valuable tool for decision makers to reduce traffic congestion and improve road safety in urban areas. The research aims to provide a comprehensive solution for traffic management by focusing on both public pickup truck management and traffic flow in urban areas. The purpose of the research is the following:

1.1.1 Address the increasing number of road accidents caused by overcrowded pickup trucks in urban areas. Overcrowded pickup trucks can lead to a higher risk of accidents due to reduced visibility, increased weight and reduced stability of the vehicle. Additionally, the increased number of passengers in the truck can lead to more severe injuries in the event of an accident. Moreover, the overcrowded pickup trucks can also cause traffic congestion, which leads to delays and increased air pollution.

1.1.2 Utilize existing open public data such as CCTV traffic cameras and Google traffic data in urban areas. This can be a cost-effective way to detect and classify overcrowded pickup trucks in real-time traffic situations. CCTV traffic cameras are widely used for traffic monitoring and can provide real-time footage of traffic on roads. Google traffic data, on the other hand, can provide information about the current traffic conditions on roads without the need for additional equipment or resources. This can help in reducing the cost of the proposed solution, and also it can be scalable and can be applied to multiple locations. Furthermore, by utilizing open public data, the proposed framework can be easily integrated into existing traffic management systems as well.

1.1.3 Support policy makers in enhancing public pickup truck safety and traffic flow in the region. It is crucial to improve the overall transportation system and reduce the number of road accidents caused by overcrowded pickup trucks.

One way to achieve this is by providing a cost-efficient solution that utilizes existing resources instead of the need for new equipment installation, including publicly available data such as Google traffic via the Google API. This approach not only saves costs, but also allows for a more efficient and scalable implementation.

1.1.4 Reduce the number of road accidents, traffic congestion, and improve overall traffic flow in urban areas.

1.2 Purpose of the Research

1.2.1 The Public Transport Pickup Trucks (PTPT) and Overcrowded Public Transport Pickup Truck (Overcrowded PTPT) dataset includes image and video footage collected from traffic cameras and dashboard cameras in the city center areas of Nakhon Ratchasima both daytime and nighttime conditions.

1.2.2 Traffic flow data collected on Mittraphap Road (Highway 2) passing through the southwesterly direction both before entering and after exiting the perimeter of Nakhon Ratchasima city for a week (Monday through Sunday) from 06:00 to 21:00 hours by extracting video footage.

1.2.3 Google Traffic flow data collected on Mittraphap Road (Highway 2) passing through the southwesterly direction both before entering and after exiting the perimeter of Nakhon Ratchasima city for a week (Monday through Sunday) from 06:00 to 21:00 hours by obtaining via Google Maps API at 15 minute-interval.

1.3 Research Questions

1.3.1 How can computer vision aid in reducing manpower for transportation engineering tasks, such as vehicle counting and distinguishing between overcrowded public transport pickup trucks and normal ones?

1.3.2 Can the use of Betweenness Centrality in graph theory be enhanced by combining it with Google Maps data to estimate traffic flow in a predefined network?

1.4 Research Contribution

1.4.1 The results of this study can provide a basis for developing a computer vision model that can extract and count vehicles from video footage obtained from traffic cameras, and also detect overcrowded passengers on public transport pickup trucks (PTPT). The following key points summarize the potential implications of this study:

1.4.1.1 The development of a real-time deep learning model for the detection of overcrowded PTPT using existing traffic cameras.

1.4.1.2 A comparison of the existing state-of-the-art deep learning networks for real-time Thai public pickup truck detection.

1.4.1.3 The open-sourcing of a manually labeled dataset of regular and overcrowded PTPT for future research purposes.

1.4.2 This research proposes an algorithm that automatically extract passenger cars from traffic cameras, feed into graph network, and calculate Betweenness Centrality (BC) to estimate traffic flow in a road network.

CHAPTER 2
LITERATURE REVIEW

In this research, the author divides the literature review into two sections: the computer vision for data collection and traffic estimation using graph theory. The first section focuses on the application of computer vision techniques for collecting traffic data, while the second section focuses on the utilization of graph theory for estimating traffic patterns and flow.

2.1 Computer Vision for Data Collection

Traditional methods in computer vision are established approaches that do not depend on machine learning but on mathematical algorithms. Despite their long-standing usage, they have often been surpassed in performance by deep learning methods (Ofir & Nebel, 2021). An example of these classic algorithms is the Canny edge detector (Canny, 1986), which is widely used for detecting curves in an image. Another study, conducted on Thai roadways in 2013, utilized various classical computer vision techniques, such as background subtraction, edge detection, and graph matching, to achieve an accuracy of 87.24% with an efficiency of five cars per second (Intachak & Wacharin, 2013). These studies demonstrate the usefulness of classical computer vision techniques, however, recent advancements in deep learning have shown to be more effective in various computer vision tasks.

In 2019, a study conducted in a similar environment used a Convolutional Neural Network (CNN) to classify vehicle color and type, including small, medium, large, and unknown vehicles. The models were trained on 917 images and achieved color and type classification accuracy rates of 70.09% and 81.62%, respectively (Maungmai & Nuthong, 2019).

In 2018, Roecker, Costa, Almeida, and Matsushita (2018a) utilized a Convolutional Neural Network (CNN) on the Beijing Institute of Technology-Vehicle dataset, which consisted of 9,850 high-resolution vehicle images. The results showed approximately 95% precision, recall, and f-measure for the truck classification task.

In the study by Cristiani et al. (2020), the authors aimed to address the issue of low-light environments that pose a challenge to traffic flow classification using image processing alone. To overcome this limitation, they introduced a luminosity sensor and integrated it with Adaboost Classifier image processing techniques to categorize road congestion. Through their approach, they achieved a remarkable overall accuracy of 92% in traffic flow classification tasks, which highlights the effectiveness of combining multiple techniques to overcome common challenges in computer vision-based traffic analysis.

Similarly, Piniarski, Pawlowski, and Dabrowski (2020) aimed to improve pedestrian detection in infrared images, which is an essential aspect of traffic analysis. They proposed a method that could speed up the pedestrian detection process, resulting in a 74% reduction in classification time. The study compared three different classifiers, including a HOG with an SVM, an aggregate channel feature, and a deep CNN, and found that the deep CNN model performed the best in terms of accuracy on various databases.

The proposed Tree-CNN by Roy, Panda, and Roy (2020) utilizes a unique architecture that addresses a common issue in machine learning models, known as "catastrophic forgetting." This occurs when adding new information to a model, such as through transfer learning techniques, leads to a complicated and less effective model. The tree-like architecture of the Tree-CNN allows the model to adapt and grow dynamically, accommodating new classes or data without sacrificing performance.

Recent developments in the field of object detection can be classified into two main categories: single-stage algorithms, such as YOLO (You Only Look Once), and two-stage algorithms, such as Faster R-CNN. Several studies have compared the performance of YOLO and Faster R-CNN and have come to the conclusion that YOLO outperforms Faster R-CNN in terms of accuracy and inference time. The single-stage nature of YOLO makes it a more efficient method for object detection, as it requires fewer computational resources and less time to make predictions compared to the two-stage Faster R-CNN. As a result, YOLO has become a popular choice for various applications, including traffic analysis and accident prevention.

Kwan et al. (2020) conducted a study that focused on real-time vehicle detection and classification using a method called pixel-wise coded exposure. In this study, the authors directly captured sample videos from pixel-wise coded aperture cameras, and the resulting images were resized to 448 × 448 pixels and then input into a YOLOv1 network. The authors reported that this approach resulted in decent performance for real-time tasks, demonstrating a good balance between speed and accuracy.

In 2020, J. Kim (2020) proposed techniques for the detection of vehicles driving in tunnel environments. These techniques combined object detection (using YOLOv2) with noise reduction and illuminance smoothing applied to image frames obtained from dashcams. The authors reported a significant improvement of 94.1% in vehicle detection accuracy. Additionally, Zhu, Wang, Wang, and Chen (2020) conducted a study on moving object detection (including airplanes, cars, and people) using YOLOv3 in combination with background subtraction on video frames. The results of this study demonstrated a mean average precision (mAP) of 91% and a processing speed of 27 frames per second (FPS).

In 2020, S. Tabassum (2020) conducted research in Bangladesh that aimed to detect native vehicles, such as trucks, using the YOLO object detection algorithm. The results of the study showed a 73% intersection over union (IOU) at a frame rate of 55 frames per second.

Table 2.1 presents a summary of the existing studies that are related to this research work. It is highlighted that there is currently no detection model available that can identify pickup trucks that are carrying an excessive number of passengers. This is the gap that this study aims to address and bridge, by developing a model that will be able to effectively identify and classify pickup trucks with overloaded passengers.

Table 2.1 Summary of related studies.

Author	Country (year)	Method	Camera Angle	mAP (%)	FPS	Pickup Truck	Overcrowded Pickup Truck
Maungmai and Nuthong (2019)	Thailand (2019)	CNN	Front	84.6*	NA	Classification	X
Roecker, Costa, Almeida, and Matsushita (2018b)	Brazil (2018)	CNN	Front	93.9*	NA	Classification*	X
J. Kim (2020)	Korea (2020)	YOLOv2	Back	94.1*	NA	Detection*	X
Zhu et al. (2020)	China (2020)	YOLOv3 -SOD	All	91.0	27	Detection*	X
S. Tabassum (2020)	Bangladesh (2020)	YOLO	All	73.0**	NA	Detection	X
This research	Thailand	YOLOv5	Back, Top Down	95.1	33	Detection	Detection
Remarks				** accuracy* *** iou*		** average all vehicle types*	

2.2 Traffic estimation using Modified Betweenness Centrality

The utilization of BC (Betweenness Centrality) has been found to be useful in the estimation of traffic volume and the explanation of the movement of traffic volume from one node to road sections within the defined road network. When combined with other factors that could explain the traffic demand for a specific road section, the utilization of BC can significantly improve the accuracy of traffic flow estimation.

2.2.1 Traffic Flow Estimation

The collection of traffic count data is crucial for estimating the flow of traffic along each road section and for determining the origin-destination (O-D) matrix. To obtain accurate traffic flow estimates, the traffic counting process must be carefully planned and executed, taking into account factors such as the date, time, number of counting points, and location. However, obtaining a large enough dataset for traffic count information can be a significant challenge, requiring substantial time, cost, and manpower (Tsapakis, Holik, Vandervalk, Turner, & Habermann, 2017).

To address this challenge, alternative methods have been introduced that require lower costs, less time, and fewer resources. These methods aim to provide efficient solutions to urban transportation problems and can be an effective way to quickly estimate traffic flow within a road network. There are various methods that have been proposed to estimate traffic flow, including the use of Betweenness Centrality (BC). BC can be used to explain the selection of a route for travel, but it falls short of providing a complete explanation of traffic flow, as it does not consider factors that are dependent on traffic demand.

To achieve a highly accurate estimation of traffic volume, it is necessary to modify traditional BC methods and to combine them with other techniques. This combined approach can help to consider the various factors that impact traffic flow and lead to more reliable and precise traffic flow estimates.

2.2.2 BC (Betweenness Centrality)

Traffic flow estimation is an important aspect in understanding the behavior of transportation systems in urban areas (Jayasinghe, Sano, & Nishiuchi, 2015). The measurement of centrality can be applied to describe the structure of nodes within a road network and provide insight into the movement of traffic (Freeman, 1977; Krafta, 2001; Sevtsuk & Kalvo, 2016; Sevtsuk & Mekonnen, 2012). Betweenness Centrality (BC) is a widely adopted method in this regard, which takes into account the influential nodes within a road network based on their connectivity and the volume of traffic passing through them (Maurya, Liu, & Murata, 2019).

The concept of BC is based on the notion that a node is considered central when many vehicles pass through it (Maurya et al., 2019). In other words, a node with a high value of BC implies that it lies on many shortest paths that drivers choose to travel to their destination. The BC of a node n within a road network can be calculated using equation (1) which takes into consideration the number of shortest paths that vehicles use to travel from node m to node p, and the total number of paths that pass through node n (Maurya et al., 2019).

$$BC(n)= \sum_{m \neq n \neq p} \frac{\sigma_{mp}(n)}{\sigma_{mp}} \tag{2.1}$$

The concept of BC has been applied in the estimation of traffic flow in a road network. However, it is distinct from other methods in its consideration of the embedment of geographic space and dynamic properties. When drivers make their way to their destination, they choose a route based on their physical needs, which contrasts with the idea behind BC that people typically choose the shortest route. Despite these limitations, the correlation between actual traffic flow and the flow estimated from BC has been studied and found to be high, with correlations reaching up to 0.8 (Crucitti, Latora, & Porta, 2006; Porta, Crucitti, & Latora, 2006; Porta et al., 2009). However, this method is only suitable for small road networks due to its limitations, as human behavior such as drivers choosing familiar routes rather than the shortest ones tends to affect the accuracy of the results (Tomko, Winter, & Claramunt, 2008).

2.2.3 Google Traffic Data

Google Maps, which was introduced in 2005, is a widely used navigation tool that offers drivers a convenient way to reach their destinations. With continuous improvements, including the addition of new features and user interface upgrades, Google Maps has become a valuable resource for drivers. The platform provides real-time traffic data and density for each road section, enabling drivers to make informed decisions about which route to take. The tool also provides real-time traffic updates, suggesting the shortest route and quickest time to reach the destination (Wallin, 2020). Additionally, Google Maps has been used to identify bottlenecks and estimate queue lengths (Kumarage et al., 2018). The traffic data provided by Google Maps are collected through the location and density of cell phone users. The traffic situation on the platform is represented by four colors (green, yellow, red, and dark red) on each road section, with green and yellow indicating light and medium traffic, respectively, red indicating traffic congestion causing delays, and dark red indicating heavy traffic congestion.

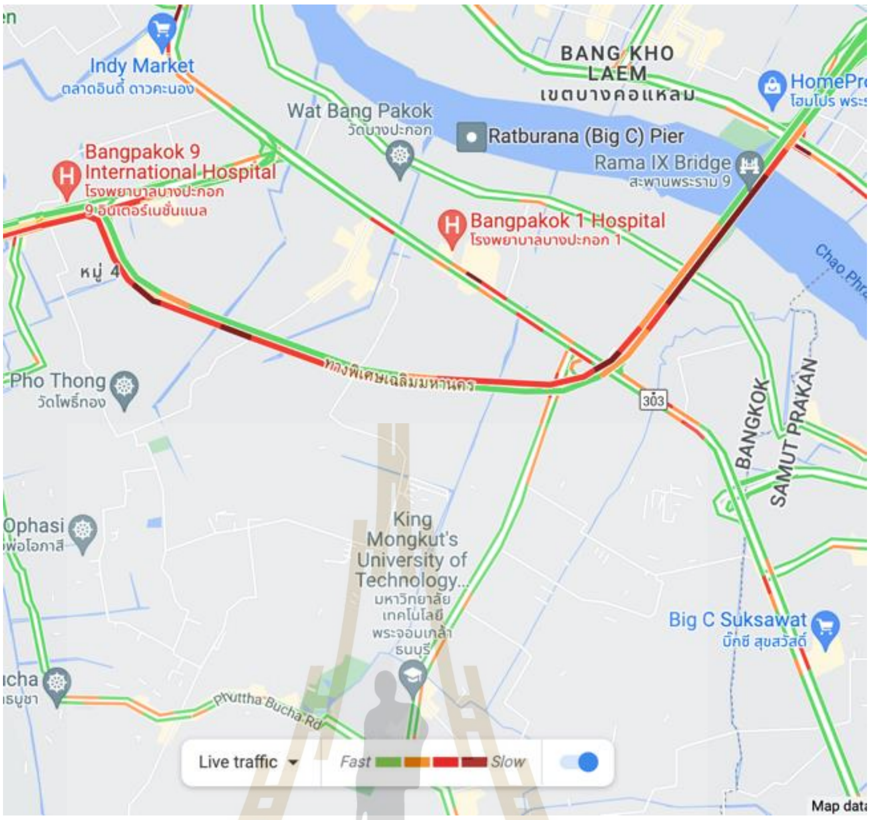

Figure 2.1 Theory of planned behavior model. (Ajzen, 1985)

Google Maps traffic data is a widely utilized tool for estimating traffic flow and congestion. It uses GPS data obtained from a variety of sources, such as drivers' mobile phones, transport departments, and road sensors (Sharma, 2017). The traffic situation on a particular route section is calculated based on the number of active mobile phones in the area (Petrovska & Stevanovic, 2015). The reliability of smartphones in estimating traffic situations has been studied by several researchers. Alvarez et al. (2014) found that the traffic data obtained from smartphones had nearly 99% reliability. Rezzouqi, Gryech, Sbihi, Ghogho, and Benbrahim (2018) compared the speed data from mobile phones with the actual road speed and found that the accuracy was approximately 75% on weekends and 83% on holidays. Ali and Abid (2021) observed the travel speed in relation to different traffic situations represented by Google Maps. They found that when the road section was green, the average speed was higher than 64 km/h, from 35-64 km/h when yellow, 18-35 km/h when red, and less than 18 km/h when dark red. Azad and Islam (2021) proposed a model for estimating traffic flow using Google traffic data, and the results showed that it can be a useful tool for estimating traffic volume.

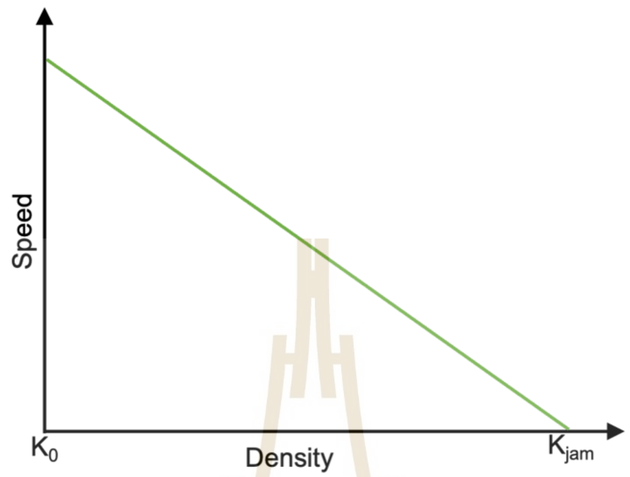

Figure 2.2 The Relationship Between Vehicle Speed and Traffic Density

The relationship between traffic speed and density is established, with speed decreasing as density increases (Venkatachalam & Gnanavelu, 2009). García Ramírez (2020) explored the feasibility of using Google traffic data to estimate both speed and volume. The results demonstrated a consistent relationship between speed and density as indicated in Figure 2.2. However, the challenge remains in determining familiar routes preferred by drivers over the shortest routes. Nevertheless, the traffic demand based on human activity can be represented by the traffic situation determined through the Google Maps engine as described in Section 2.2.2 BC (Betweenness Centrality)

CHAPTER 3
MATERIALS AND METHODS

The Materials and Methods section of this study can be divided into two main components: (1) Vehicle Detection based on YOLO and (2) Traffic Flow Estimation using the Proposed Enhanced Betweenness Centrality. The overall experiments are structured around the following framework:

A. Traffic Cameras: CCTV are used to capture video footage of traffic flow at specific locations in the study areas.

B. Traffic Volume Extraction: Traffic volume data are extracted from the video footage obtained from the traffic cameras. This process involves using computer vision techniques to analyze the video and count the number of vehicles, particularly passenger cars, passing through the monitored area.

C. Volume Prediction: Once the traffic volume data is obtained, a graph network method (Betweenness Centrality) is employed to predict future traffic volume.

D. Evaluation and Analysis: The evaluation and analysis of the prediction's accuracy and performance involve comparing the predicted traffic volume with the actual observed volume. This comparison allows for an assessment of how well the prediction aligns with the real data. Additionally, the coefficient of correlation is determined to quantify the level of explainability and the strength of the relationship between the model's predictions and the actual observed volume.

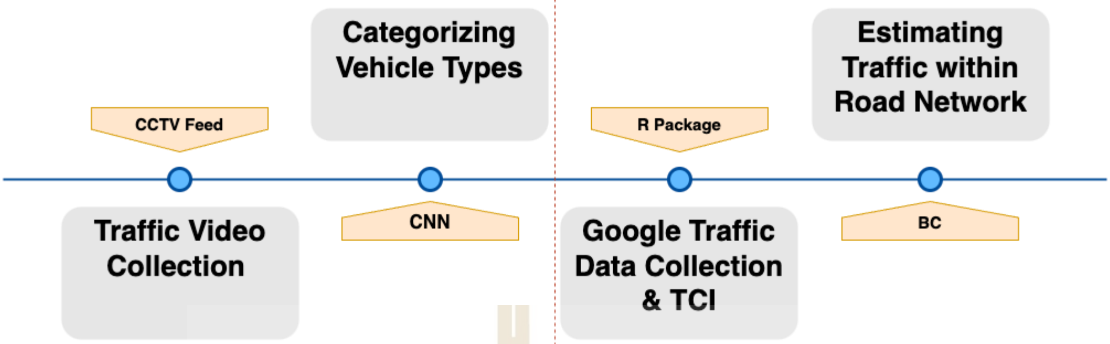

Figure 3.1 Overview of Framework

3.1 Vehicle Detection based on YOLO

The methodology of this detection task is summarized in Figure 4. The first step involved the construction of the dataset, which was achieved by collecting a variety of images and videos. These images were then annotated to identify regions of normal and overcrowded public transport passenger terminals (PTPT). To further enhance the dataset, various techniques were applied as listed in Table 2. The next step was to fine-tune the pretrained YOLOv5 models, obtained from the official GitHub of the authors (WZMIAOMIAO, 2022a), to better suit our requirements. In the final evaluation stage, the finetuned models were compared based on their mean Average Precision (mAP) and speed in terms of Frames per Second (FPS) for the inference task.

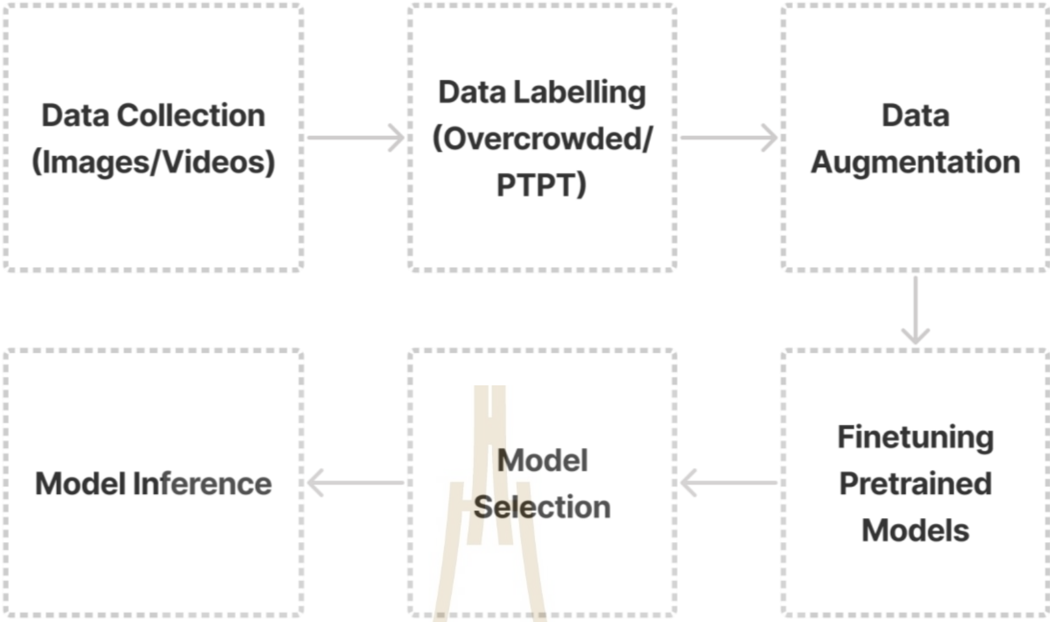

Figure 3.2 Process Flow Diagram

Table 3.1 Summary of previous studies on WTP for risk reduction and findings.

Steps	Detail
Preprocessing	Fix Orientation Resize: Fill (with center crop) in 640 × 640 Auto Adjust Contrast: Using Histogram Equalization
Augmentation	Flip Horizontally Crop: 0% Minimum Zoom, 2.5% Maximum Zoom Rotation: Between −15° and +15° Shear: ±15° Horizontal, ±15° Vertical Grayscale: Apply to 10% of images Hue: Between −25° and +25° Saturation: Between −25% and +25% Brightness: Between −25% and +25% Exposure: Between −25% and +25% Blur: Up to 1.75 px Noise: Up to 5% of pixels

3.1.1 Datasets

The sample images were obtained from a variety of sources including Google Image Search, photos taken by the authors in local city areas, video clips from news media, and traffic cameras (Hikvision 4MP Bullet Network Camera) installed approximately 12 m high on three different pedestrian bridges above highways in nearby cities. The traffic camera footage captured high-resolution images of the roadways and surrounding areas from an elevated perspective. A sample image from the traffic camera can be observed in Figure 3.3. The authors also used an Apple iPhone 12's main camera (12 million pixels) to capture still images and 1080 p video at 30 FPS from a car's dashboard level. The sample comprised two main categories of PTPT: 512 images of modified pickups (PTPT) and 431 images of overcrowded ones (overcrowded PTPT) and 346 background images, resulting in a total of 1,289 images. The input images were augmented, as detailed in Table 3.1, and the final dataset comprised 2,975 images. These images were randomly separated into 80% training set, 10% validation set, and 10% test set. Examples of both the datasets are shown in Figure 3.4 and Figure 3.5.

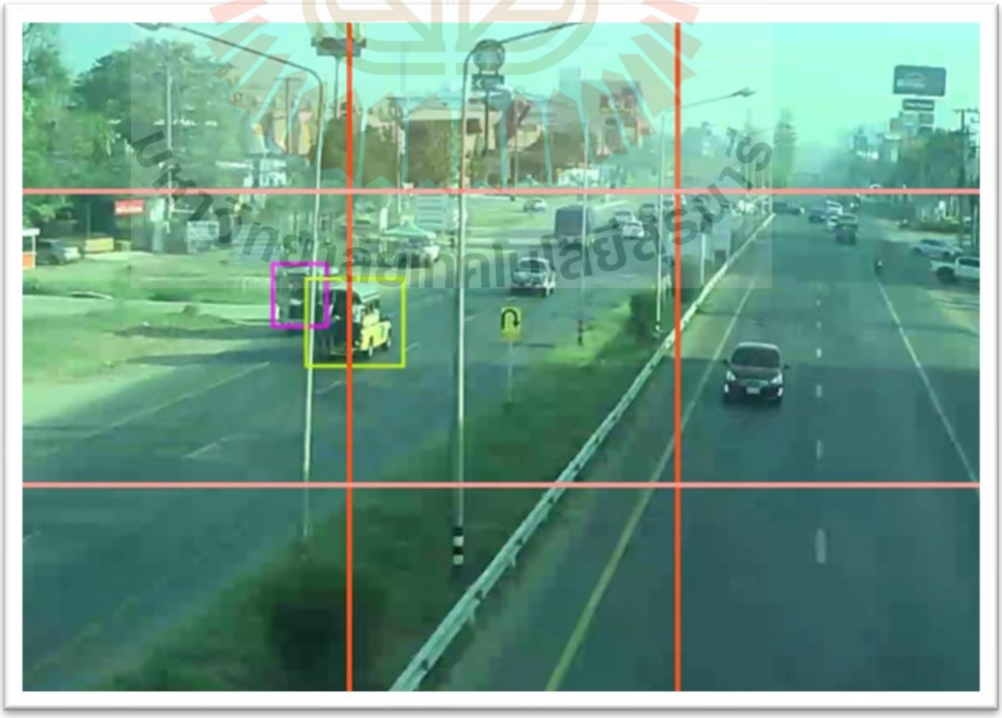

Figure 3.3 S × S Grid

Figure 3.4 Public Transport Pickup Trucks (PTPT)

Figure 3.5 Overcrowded Public Transport Pickup Trucks (Overcrowded PTPT)

3.1.2 YOLOv5 Architecture

YOLO, which was first introduced by Redmon, Divvala, Girshick, and Farhadi (2015) in 2015, uses a CNN for object detection and bounding box prediction all at once in a single forward propagation. YOLO takes an input image and divides it into an equal grid (s x s size), as shown in Figure 3.3. Furthermore, image classification and bounding box prediction are applied to each grid. The algorithm then outputs the corresponding class probabilities (i.e., overcrowded or PTPT), coordinates of the predicted object, and the confidence score of the boxes. In the subsequent YOLOv2, Redmond et al. added batch normalization, which was initially introduced by Ioffe and Szegedy (2015), and a few other techniques to improve the mean average precision (Redmon & Farhadi, 2016). They have also used Darknet-53 in the feature extraction layers of the network, improving detection speed and accuracy (Redmon & Farhadi, 2018). YOLOv4 was then proposed to use CSP-Darknet53 as a backbone to address YOLOv4's issue in detecting large objects (Bochkovskiy, Wang, & Liao, 2020).

The YOLOv5 (v6.0/6.1) continued to use CSPDarknet53 as a backbone and replaced the Focus layer, which was introduced in the earlier version to a 6 × 6 Conv2d. Furthermore, the spatial pyramid pooling was then replaced with the SPPF (F for fast). The SPPF layer increases the receptive field and separates the most important features from the backbone. The YOLOv5 version 6.0/6.1 (WZMIAOMIAO, 2022b) used the same head as in YOLOv3 and YOLOv4. Figure 3.6 shows the architecture of the latest YOLOv5 model L. In summary, Yolov5 now consists of three parts, namely, backbone (new CSP-Darknet53) for feature formation, neck (SPPF, new CSP-PAN) for feature extraction, and head (YOLOv3 Head) for detection. C3 denoted in backbone and neck comprises three convolutional layers and a module cascaded by two types of bottlenecks. Finally, the last three convolution layers in the head of the network are detection modules that generate three different feature maps. It is worth mentioning that since the YOLOv5 version 4.0 (Jocher, 2022), all activation functions (LeakyReLU or Leaky rectified linear unit (Xu, 2015) and HardSwish (Howard, 2019) used in convolutional layers are replaced by the Sigmoid linear unit (Elfwing, 2017) to simplify the architecture.

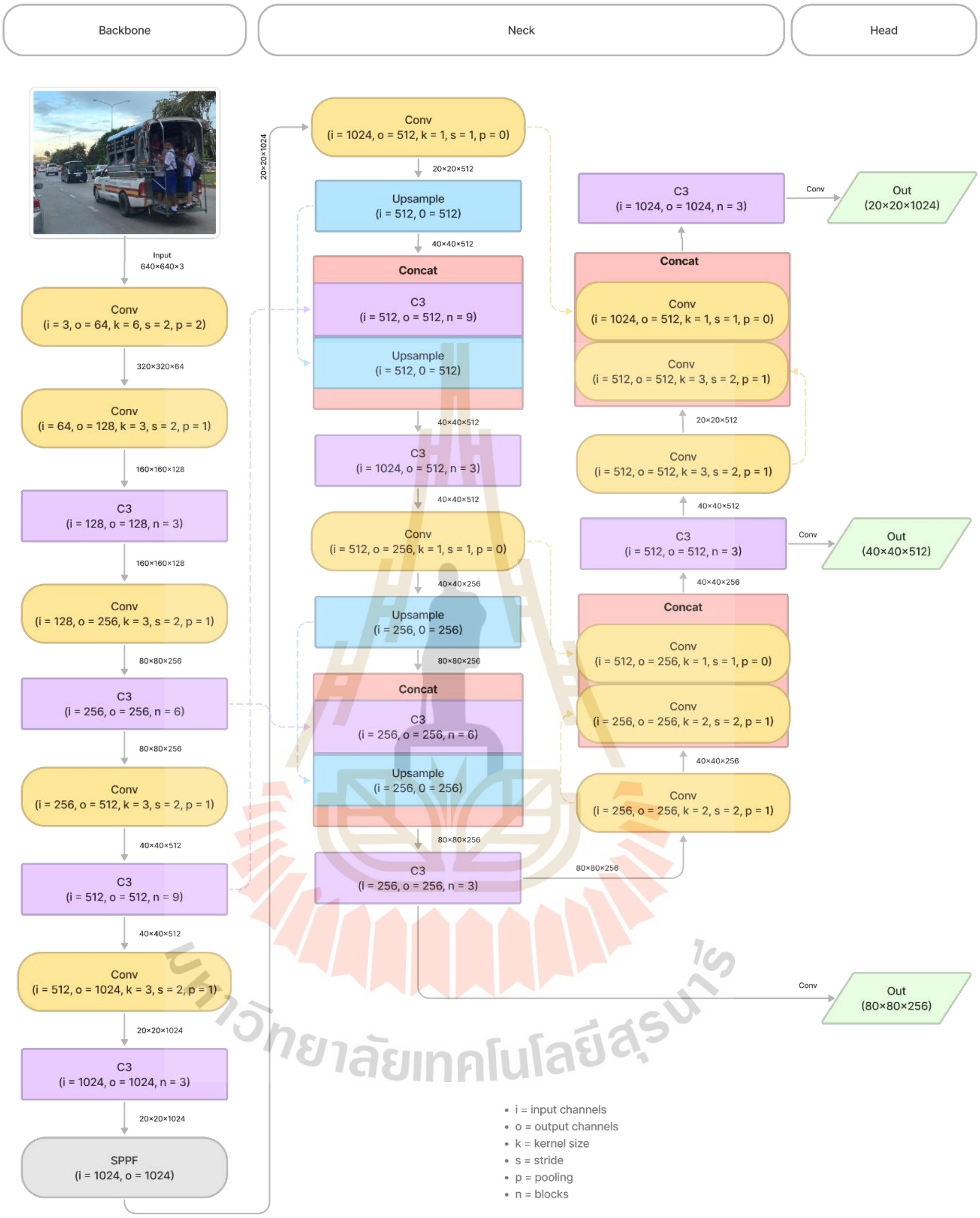

Figure 3.6 Model Structure of YOLOv5L (You Only Look Once)

3.1.3 Finetuning

In this study, the authors utilized a transfer learning approach to train the YOLOv5 models (N, S, M, L, and X) for recognizing overcrowded PTPT. The pretrained models were obtained from the official GitHub repository of the YOLOv5 authors (WZMIAOMIAO, 2022a). The training was conducted on an NVIDIA GeForce RTX2070 GPU using PyTorch 1.8 on Ubuntu 21.04. The authors set the training hyperparameters to the default values without freezing the backbone (F=0) for a total of 1,200 epochs with early stopping. The batch size was set to the maximum to fully utilize the GPU memory and the image size was set to 640x640 to match the pretrained models. The number of classes was defined as 2.

3.1.4 Performance Metrics

To test and compare the models, we used standard evaluation indicators of precision, recall, f1 score, average precision (AP), mAP, and FPS. Precision is the ratio of correctly detected to all detected labels, how many retrieved items are relevant. Recall is the ratio of correctly detected to all positive targets, how many relevant items are retrieved. F1 score is the harmonic mean of precision and recall. The AP metric is known to be a way to describe a precision-recall curve representing the average of all precision. The mean average precision is simply an average of the AP of all classes. The final evaluation metric is the FPS, which describes the number of images that can be processed in one second. Equations. (3.1)–(3.5) define the formulas as follows:

$$\text{precision}=\frac{TP}{TP+FP} \tag{3.1}$$

$$\text{recall}=\frac{TP}{TP+FN} \tag{3.2}$$

$$F_1=\frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}=\frac{2*TP}{2*TP+FP+FN} \tag{3.3}$$

$$AP=\int_0^1 p(r)dr \tag{3.4}$$

$$mAP=\frac{1}{n}\sum_{i=1}^{n} AP_i \tag{3.5}$$

where TP = true positive, FP = false positive, FN = false negative.

3.2 Traffic Flow Estimation using the Proposed Enhanced Betweenness Centrality

To deal with the limitations of traditional BC, it should be combined with other factors that will increase the accuracy of the traffic flow estimation. The methodological framework of this study is shown in Figure 3.7 - Methodological Framework. Traffic volume in any route could be estimated from O–D traffic demand and traffic volume that a driver passes through to reach a destination. The traffic volume of the second aspect can be estimated by BC, which will find the shortest route for a driver. The first term will be explained by the traffic density drawn from Google traffic data.

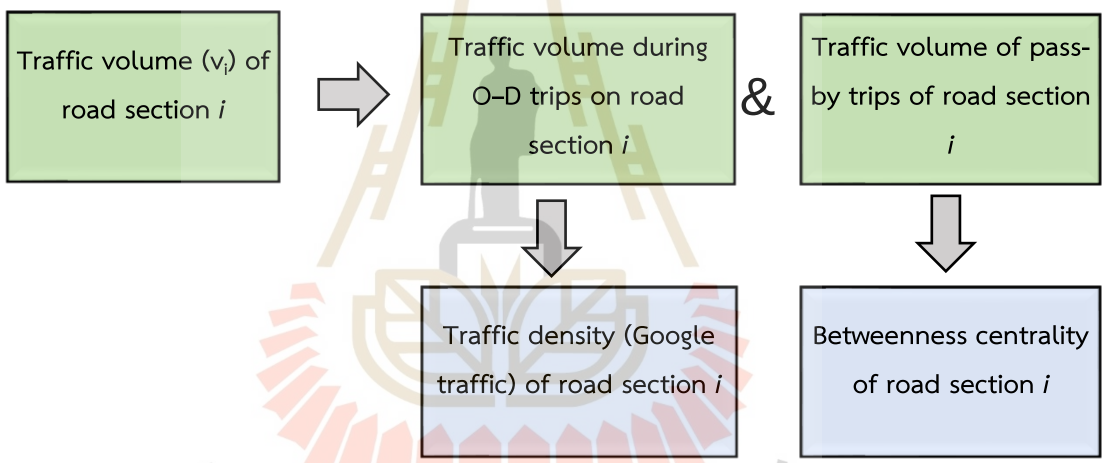

Figure 3.7 Methodological Framework

3.2.1 Study Area

The empirical examination was carried out in the city center of Nakhon Ratchasima, Thailand, as indicated in Figure 3.8. Traffic volume data from the northeast were obtained at Point A, as indicated in the figure. The data was collected from traffic cameras over the course of one week, from 6:00 AM to 9:00 PM, and was complemented by typical Google traffic data obtained via Google Maps during the same hours and days, with updates every 15 minutes for a total of one week. However, due to ongoing road construction work during that period, the traffic camera data was substituted with data obtained from traffic sensors. The collected data contained missing values that needed to be addressed. To handle this issue, the author employed a two-step approach. Firstly, when possible, missing values were filled using data from a comparable period, such as the same time last week. This approach helped to ensure that the missing values were replaced with relevant and representative data points. Secondly, in cases where data from the comparable period was not available, the author resorted to imputing the missing values by calculating the average value across the same timeframe for all periods. To calculate the TCI, the percentage of each color was determined within each 15-minute interval and was calculated using Equation (3.6). Additionally, traffic volume data at Point B was also collected in order to estimate the traffic movement from Point A through Point B. An hour difference in traffic data between Point A and Point B was observed for the traffic volumes distributed on the local roads within Nakhon Ratchasima city center.

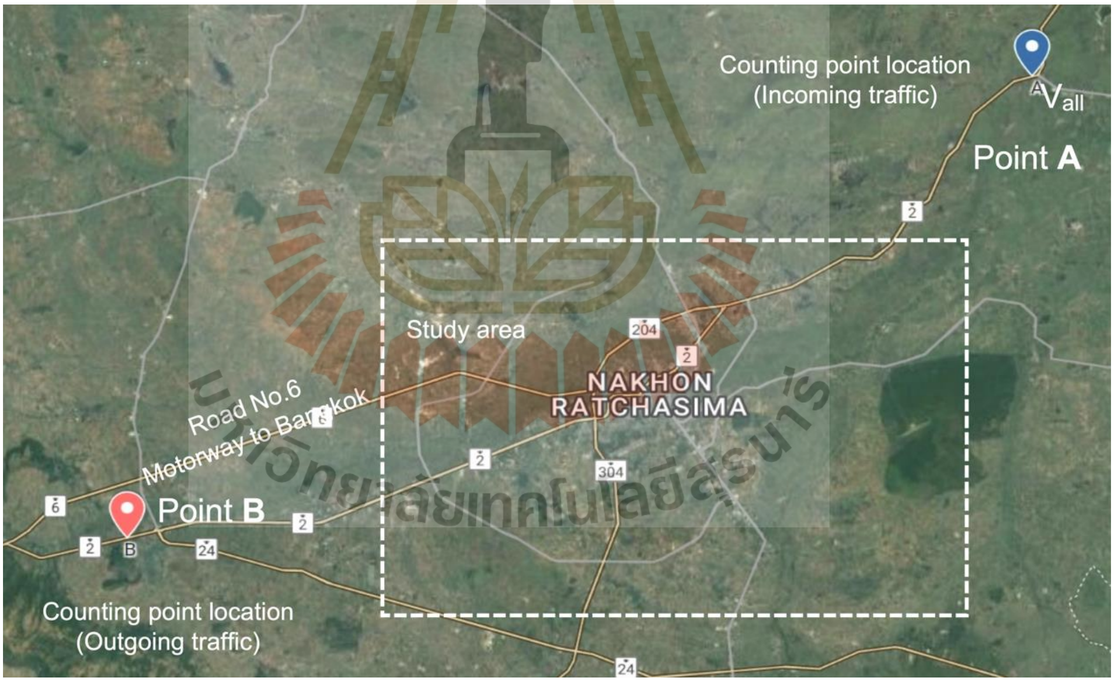

Figure 3.8 Study Area, City Center of Nakhon Ratchasima, Thailand

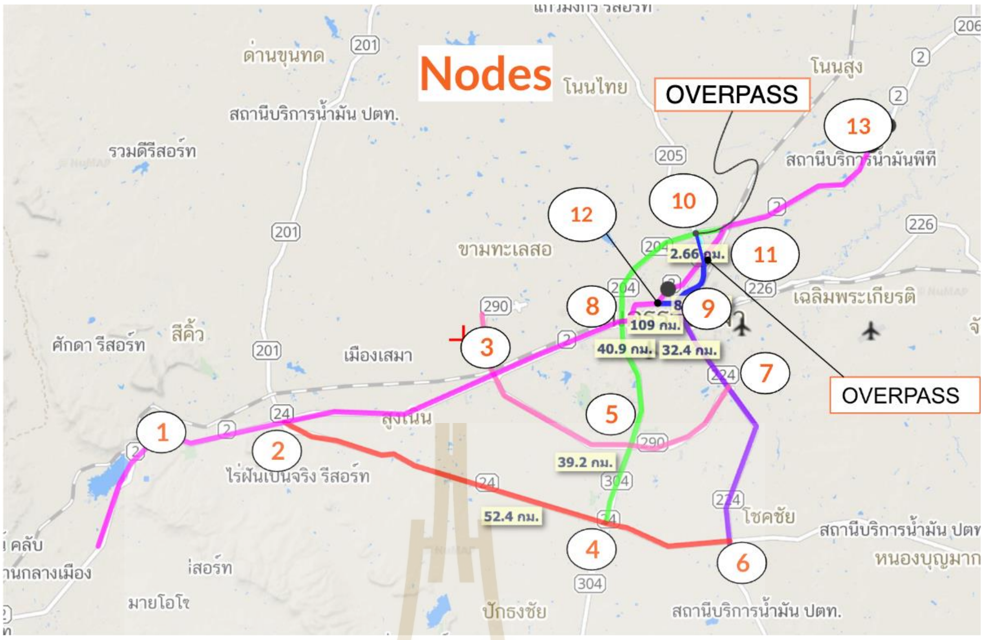

Figure 3.9 Nodes

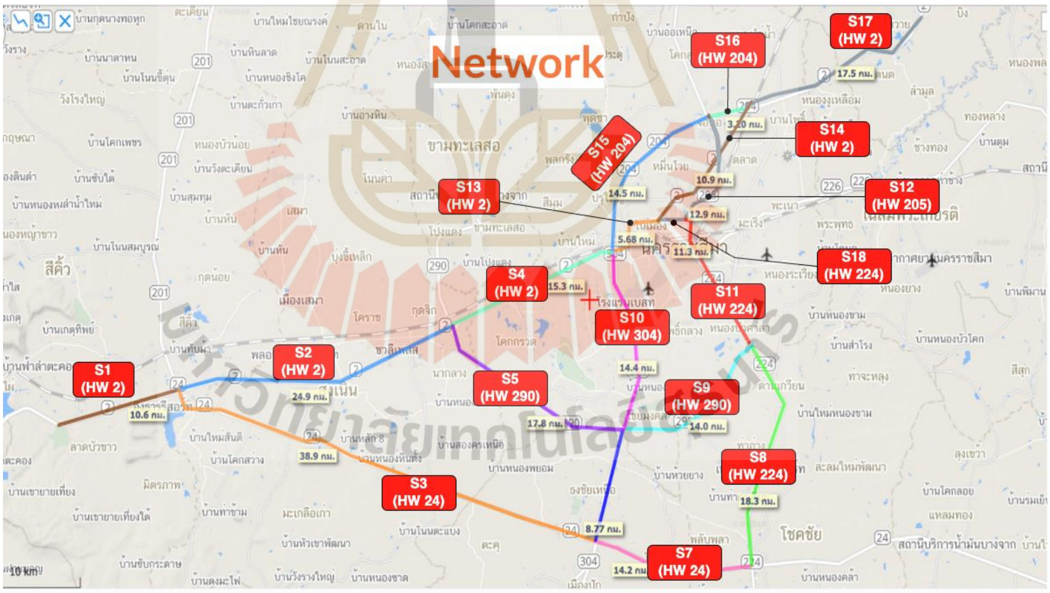

Figure 3.10 Road Network

The nodes and road sections (Figure 3.9 and Figure 3.10, respectively) within Nakhon Ratchasima city center are shown in Figure 3.11, consisting of 13 nodes and 18 road sections. The traffic volume passing through all the road sections is estimated by the difference in traffic volume at points A and B.

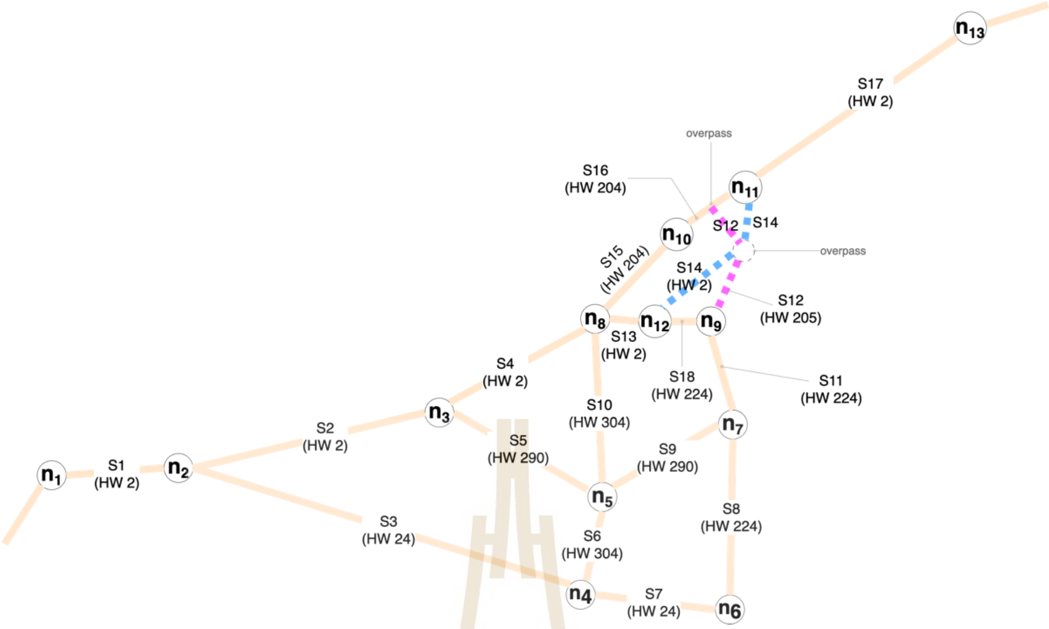

Figure 3.11 Nodes and Road Sections

3.2.2 Traffic Distribution Based on The BC Method

The traffic volume was simulated by varying the betweenness detour ratio between 20% and 50%. The index in each route section (i) was multiplied by the total traffic volume on the main route (V_{all}).

$$V_i = BC_i \times V_{all} \tag{3.6}$$

Note that the input traffic volume V_{all} is calculated as the difference between the input traffic volume at point A and the output traffic volume at point B, an hour later. This assumes that the through traffic from A to B takes 60 minutes, which is the average travel time for this route (109 km). By simulating the probability of traffic volume, this method accounts for the potential travel decisions drivers may make, excluding the through traffic.

3.2.3 Betweenness Centrality (BC)

In this study, betweenness centrality (BC) is used to predict the probability of traffic conditions on each local street. Betweenness centrality is calculated based on the number of shortest paths that pass-through a given street. A street with high betweenness centrality indicates a greater likelihood of it being used as a pathway to the destination. The equation of Betweenness Centrality is displayed in Equation (3.7).

$$Betweenness^{r}[i] = \sum_{j,k \in G-\{i\}; d[j,k] \leq r} \frac{n_{jk}[i]}{n_{jk}} \cdot W[j] \tag{3.7}$$

Note that n_{jk} is the number of shortest paths from j to k. n_{jk} [i] is the number of shortest paths passing thorough i, with j and k located within search radius r from i. W[j] is the weight of a specific destination j.

3.2.4 Traffic Congestion Index (TCI)

The traffic data shown by the colors in each section were collected every 15 minutes for one week, and were divided into four groups: green, yellow, red, and dark red. The traffic volume distribution from one node passing to another node passing through different routes is not equal and varies depending on the shortest route. When traffic speed is low and traffic density is high in a particular road section, the traffic density in that section (represented by color) is assumed to be high if high traffic volume from the origin node passes through that route section. The traffic situation (color) of a road section is assessed using the TCI, which converges with Equation (3.8). A low TCI (a high percentage of green) indicates low traffic volume, as only a small number of vehicles from the origin node pass through that route section (Bastea, 2016).

$$\mathrm{TCI} = (0 \times \mathrm{P0}) + (1 \times \mathrm{P1}) + (2 \times \mathrm{P2}) + (3 \times \mathrm{P3}) + (4 \times \mathrm{P4}) \tag{3.8}$$

The traffic index from Google Maps ranged from 1 to 4, where green = 1, yellow = 2, red = 3, and dark red = 4. The TCI can be calculated using the percentage of each color within a defined period. For example, within 15 minutes, section A was green for 10 minutes (66.67%) and yellow for 5 minutes (33.33%). The TCI is calculated as (0 x 66.67) + (1 x 33.33) = 33.33, indicating that traffic is flowing and traffic density within the road section is low. Table 3 shows an example of TCI calculation.

Table 3.2 An Example of the TCI of Different Road Sections

Section	P0	P1	P2	P3	TCI	Traffic situation
A	100	0	0	0	0	Flow/very low density
B	85.42	7.21	2.51	4.86	26.81	Low traffic congestion/low density
C	41.78	13.08	6.42	38.72	142.08	High congestion/high density

3.2.5 Statistical Analysis

We utilize the Pearson correlation coefficient to evaluate the association between the predicted betweenness centrality outcome and the TCI (Transitivity Clustering Index). The Pearson correlation coefficient is employed as a measure of the magnitude and direction of the linear relationship between the two variables under investigation. The statistical significance of the correlation coefficient is ascertained by examining the p-value, which furnishes evidence concerning the correlation coefficient between the outcomes derived from Equation (3.6) to (3.8). In this investigation, a significance level of 0.05 is set as the threshold for determining statistical significance.

3.2.6 Traffic Volume Estimation

We evaluated the proposed method by determining the correlation between the estimated traffic volume and the TCI of a particular road section. The correlation ranged between 0.1 and 1.0, where a strong positive correlation between the two factors was indicated by a value higher than 0.5. If the correlation between the estimated traffic volume based on BC and the TCI is higher than 0.5, the traffic volume from one area spreading to another can be estimated.

Raod section xx	Time	Betweenness ratio				Google traffic						
10		20%	30%	40%	50%	Mon	Tue	Wed	Thur	Fri	Sat	Sun
	06:00	31.31667	26.84465	26.84465	23.19939	1	1	1	1	1	1	1
	06:15	35.18333	30.15915	30.15915	26.06381	1	1	1	1	1	1	1
	06:30	37.61667	32.24501	32.24501	27.86643	1	1	1	1	1	1	1
	06:45	47.16667	40.43127	40.43127	34.94107	1	1	1	1	1	1	1
	07:00	65.56667	56.20375	56.20375	48.57179	1	1	1	1	1	1	1
	07:15	68.35	58.58962	58.58962	50.63368	1	1	1	1	1	1	1
	07:30	80.91667	69.36177	69.36177	59.94307	2	2	2	2	1	1	1
	07:45	86.31667	73.66205	73.66205	63.94339	2	2	2	2	2	1	1
	08:00	87.56667	75.06215	75.06215	64.86939	2	2	1	1	2	1	1
	08:15	84.26667	72.23339	72.23339	62.42475	2	2	2	2	2	1	1
	08:30	83.45	71.53334	71.53334	61.81976	2	2	1	2	1	1	1
	08:45	82.81667	70.99045	70.99045	61.35059	1	1	1	1	1	1	1
	09:00	86.23333	73.91921	73.91921	63.88165	1	2	1	1	1	1	1
	09:15	89.35	76.59082	76.59082	66.19048	1	1	1	1	1	1	1
	09:30	94.11667	80.67681	80.67681	69.72163	1	1	1	1	1	1	1
	09:45	93.55	80.19106	80.19106	69.30184	1	1	1	1	1	2	1
	10:00	104.2167	89.33453	89.33453	77.20371	2	2	1	2	2	2	2
	10:15	104.9167	89.93457	89.93457	77.72227	2	2	2	2	2	2	2
	10:30	111.15	95.27778	95.27778	82.33992	2	2	2	2	2	2	2
	10:45	113.9333	97.66365	97.66365	84.40181	2	2	2	2	2	2	2
	11:00	118.7333	101.7782	101.7782	87.95765	2	2	2	2	2	2	2
	11:15	116.9	100.2067	100.2067	86.59952	2	2	2	2	2	2	2
	11:30	121.4	104.0641	104.0641	89.93312	2	2	2	2	2	2	2
	11:45	124.3	106.55	106.55	92.08144	2	2	2	2	2	2	2
	12:00	127.55	109.3359	109.3359	94.48904	2	2	2	2	2	2	2
	12:15	128.0833	109.793	109.793	94.88413	2	2	2	2	2	2	2
	12:30	128.7833	110.3931	110.3931	95.40269	2	2	2	2	2	2	2
	12:45	129.1833	110.736	110.736	95.69901	2	2	2	2	2	2	2
	13:00	127.3333	109.1501	109.1501	94.32853	2	2	2	2	2	2	2
	13:15	128.4833	110.1359	110.1359	95.18045	2	2	2	2	2	2	2
	13:30	131.6333	112.8361	112.8361	97.51397	2	2	2	2	2	2	2
	13:45	127.7	109.4644	109.4644	94.60016	2	2	2	2	2	2	2
	14:00	132.7333	113.779	113.779	98.32885	2	2	2	2	2	2	2
	14:15	129.4	110.9217	110.9217	95.85952	2	2	2	2	2	2	2
	14:30	128.85	110.4502	110.4502	95.45208	2	2	2	2	2	2	2
	14:45	130.4167	111.7932	111.7932	96.61267	2	2	2	2	2	2	2
	15:00	122.1167	104.6784	104.6784	90.46403	2	2	2	2	2	2	2
	15:15	124.9	107.0643	107.0643	92.52592	2	2	2	2	2	2	2
	15:30	121.6833	104.307	104.307	90.14301	2	2	2	2	2	2	2
	15:45	121.65	104.2784	104.2784	90.11832	2	2	2	2	2	2	2
	16:00	120.4	103.2069	103.2069	89.19232	2	2	2	2	2	2	2
	16:15	121.6167	104.2498	104.2498	90.09363	2	2	2	2	2	2	1
	16:30	125.6833	107.7358	107.7358	93.10621	2	2	2	2	2	2	2
	16:45	117.9	101.0639	101.0639	87.34032	2	2	2	2	2	2	1
	17:00	118.0167	101.1639	101.1639	87.42675	2	2	2	2	2	2	2
	17:15	114.7167	98.33513	98.33513	84.98211	2	2	2	2	2	2	2
	17:30	108.6833	93.16335	93.16335	80.51261	2	2	2	2	2	2	2
	17:45	103.1167	88.39161	88.39161	76.38883	2	2	2	2	2	2	2
	18:00	95.35	81.73402	81.73402	70.63528	2	2	2	2	2	2	1
	18:15	92.38333	79.19099	79.19099	68.43757	2	2	2	2	2	2	1
	18:30	86.96667	74.54783	74.54783	64.42491	2	2	2	2	2	2	1
	18:45	82.46667	70.69043	70.69043	61.09131	2	2	2	2	2	2	2
	19:00	76.41667	65.50437	65.50437	56.60947	2	2	2	2	2	2	2
	19:15	74.1	63.51852	63.51852	54.89328	2	2	2	2	2	2	2
	19:30	76.71667	65.76153	65.76153	56.83171	2	2	2	2	2	2	2
	19:45	74.65	63.98998	63.98998	55.30072	2	2	2	2	2	2	2
	20:00	75.06667	64.34715	64.34715	55.60939	2	2	2	2	2	2	2
	20:15	74.23333	63.63281	63.63281	54.99205	2	2	2	2	2	2	2
	20:30	69.96667	59.97543	59.97543	51.83131	2	2	2	2	2	2	2
	20:45	69	59.1468	59.1468	51.1152	1	2	2	2	2	2	1
	21:00	64.88333	55.61799	55.61799	48.06557	1	1	1	2	2	1	1

แสดงผลลัพธ์โดยระบุ correlation ของ

PEARSON	10%	20%	30%	40%	50%
Mon		0.634164	0.63397	0.63397	0.634164
Tue		0.606767	0.606587	0.606587	0.606767
Wed		0.591993	0.591791	0.591791	0.591993
Thur		0.571909	0.571728	0.571728	0.571909
Fri		0.588642	0.588452	0.588452	0.588642
Sat		0.64276	0.64308	0.64308	0.64276
Sun		0.605781	0.606013	0.606013	0.605781

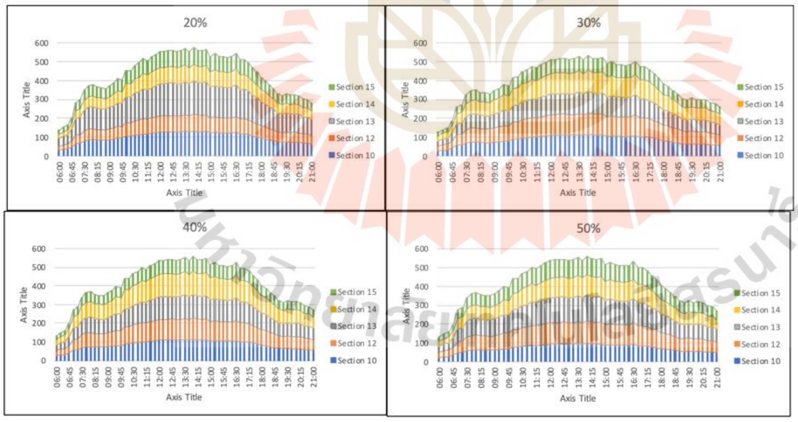

Figure 3.12 Example Calculations of Route Section 10

CHAPTER 4
RESULTS AND DISCUSSION

This section presents the results of the study, which are divided into two main components: the convolutional neural network (CNN) model and the traffic estimation model.

4.1 CNN

We compared the performance of five pretrained YOLO models, namely YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5L, and YOLOv5x, which can be obtained directly from GitHub WZMIAOMIAO (2022a). Mean average precision (mAP), as defined in Equation (3.5), was used as the standard metric to evaluate the object detection model, with higher mAP values indicating greater accuracy. All five models were trained for 1,200 epochs with early stopping, as indicated in the "Best epoch" column of Table 4.1. The precision, recall, and F1 scores are also presented in the same table and visualized in Figure 4.1. As shown in Figure 4.2, the loss of our models decreased sharply in the early steps and gradually converged, coinciding with an increase in accuracy.

Table 4.1 Evaluation Metrics for Different YOLOv5 Models

Model	Runtime (secs)	Best epoch	mAP_0.5 (%)	mAP_0.5:0.95 (%)	Precision (%)	Recall (%)	F_1 (%)
Model X	71973	190	96.44	70.63	94.47	92.29	93.36
Model L	39195	239	95.11	65.65	95.94	93.84	94.88
Model M	29382	296	95.66	69.14	92.60	95.38	93.97
Model S	15736	366	96.32	67.02	90.91	95.40	93.10
Model N	7691	297	96.06	66.63	97.51	90.70	93.98

Since our application requires real-time usage, we need to measure the frames per second (FPS), which is the reciprocal of the inference time taken by the model per image and is presented in Table 4.2. As shown, the chosen model should not be used on an edge device because the FPS is too low for real-time applications. It is worth noting that different batch sizes and computer environment settings may lead to varying detection speeds.

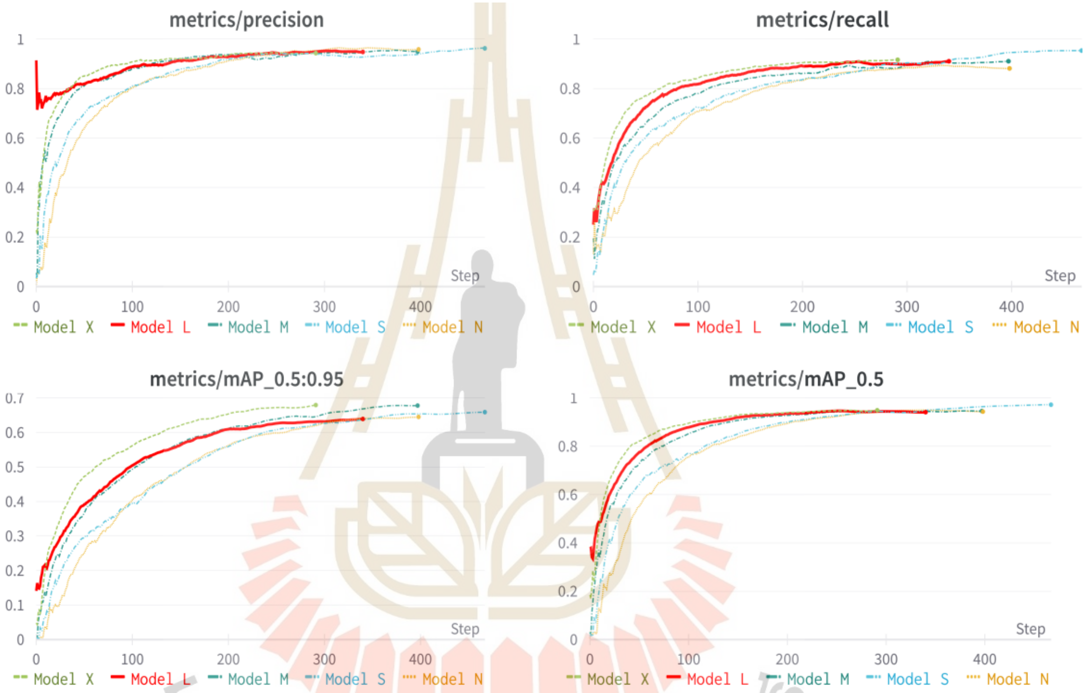

Figure 4.1 Precision, Recall, and Mean Average Precision on Five YOLOv5 Models

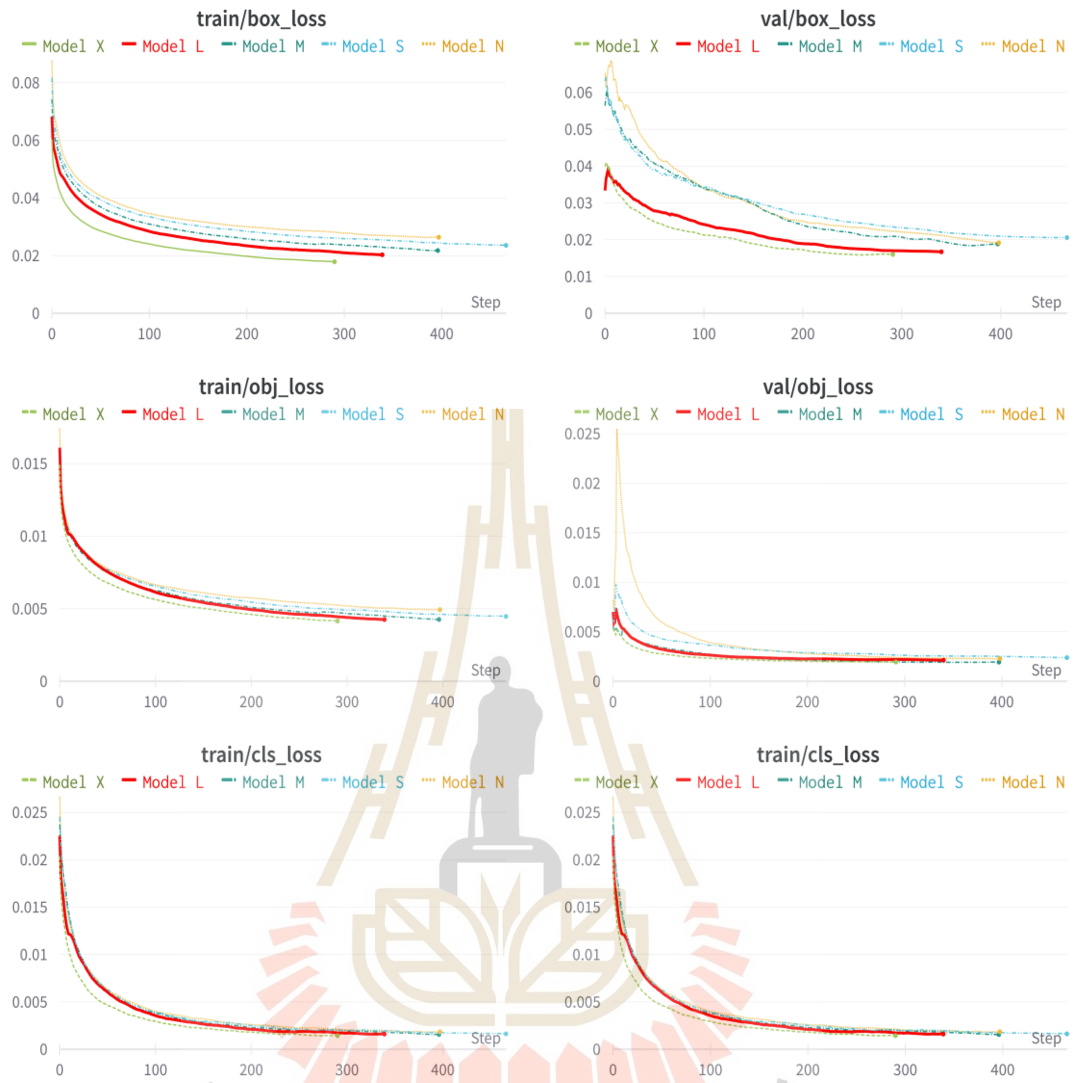

Figure 4.2 Train and Validation Loss

Table 4.2 Inference Speed Comparison.

Model	Batch Size	FPS (GPU)	FPS (CPU)
Model X	8	19	1
Model L	8	33	2
Model M	8	66	4
Model S	8	156	9
Model N	8	256	20

All the trained models exhibited similar precision and recall rates, as shown in Figure 4.3, and similar F1 scores, as shown in Figure 4.4. The confusion matrix diagrams in Figure 4.5 also demonstrated comparable results across the models. Consequently, the process of selecting the most appropriate model was a trial-and-error endeavor. In this study, we opted for the YOLOv5L model with a confidence threshold of 0.60 and an intersection-over-union (IOU) threshold of 0.55, as it produced satisfactory results on both the GPU and CPU, as illustrated in Figure 4.6 and Figure 4.7.

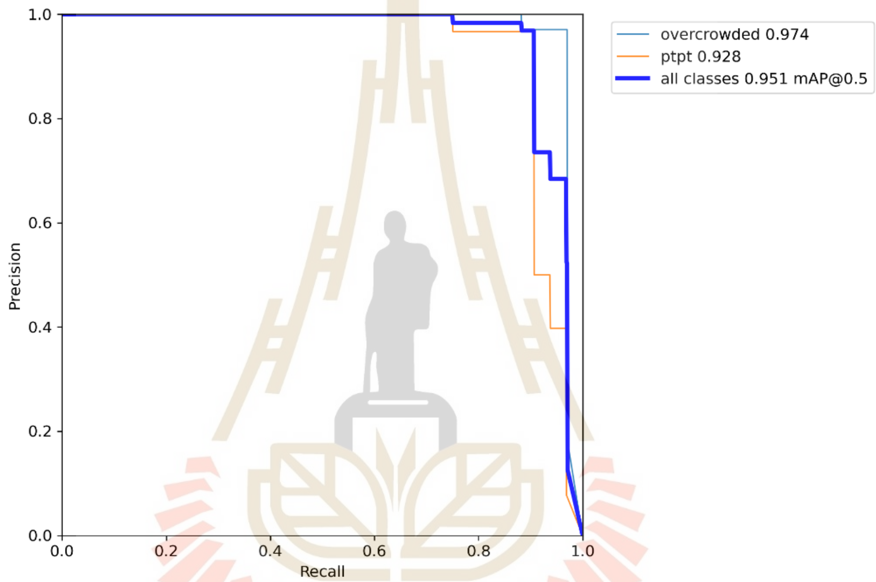

Figure 4.3 Precision-Recall Curve of YOLOv5L

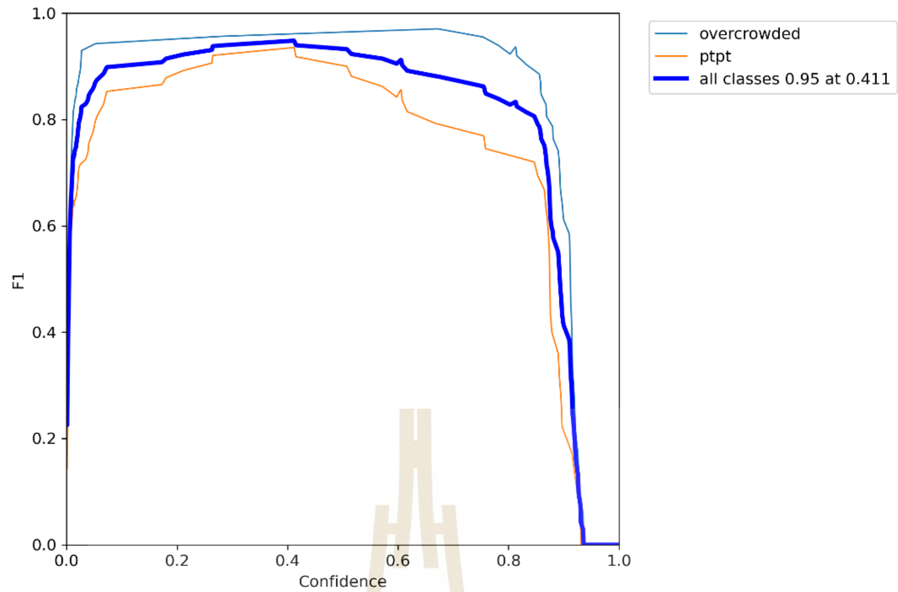

Figure 4.4 F1 Plot of YOLOv5L

Figure 4.5 Confusion Matrix of YOLOv5L

Figure 4.6 Overcrowded Detection by YOLOv5L (Online, 2019)

Figure 4.7 Multiclass Detection by YOLOv5L (Online, 2016)

It is worth noted that precision is a measure of how accurate the model is when it classifies an image as overcrowded. A high precision means that the model is very confident when it classifies an image as overcrowded, and it is unlikely to misclassify a regular image as overcrowded. On the other hand, recall is a measure of how many of the actual overcrowded images the model correctly classifies as overcrowded. A high recall means that the model is able to identify most of the actual overcrowded images, even if it is not always confident in its classifications. In this case, the selected YOLO model is able to correctly classify overcrowded PTPT significantly and misclassify regular ones as overcrowded with high probability, hence the lower recall. This means that the model is very confident when it classifies an image as overcrowded, but it is likely to miss some of the actual overcrowded images. The use of larger dataset might help improving the model though.

4.2 BC

The issue of traffic congestion within Nakhon Ratchasima district, particularly within the central urban region, is a significant concern. The high traffic density observed in this area results in significant delays and disrupts the normal flow of traffic.

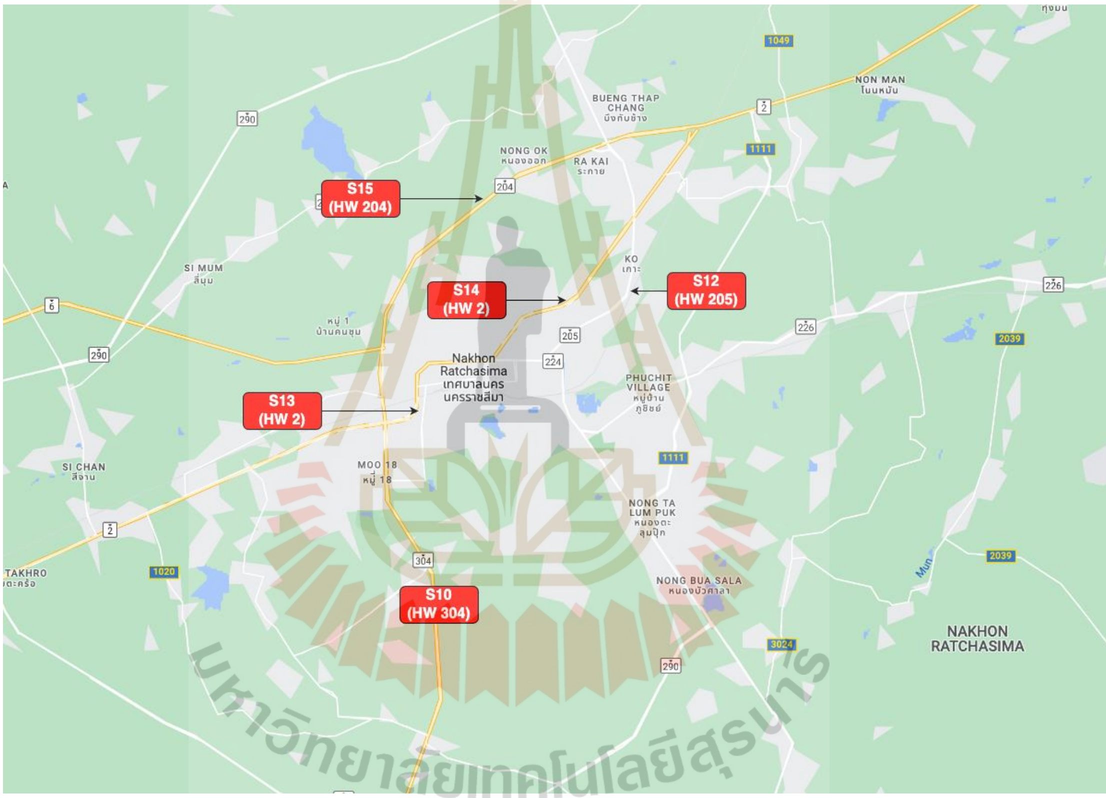

Figure 4.8 Selected Routes

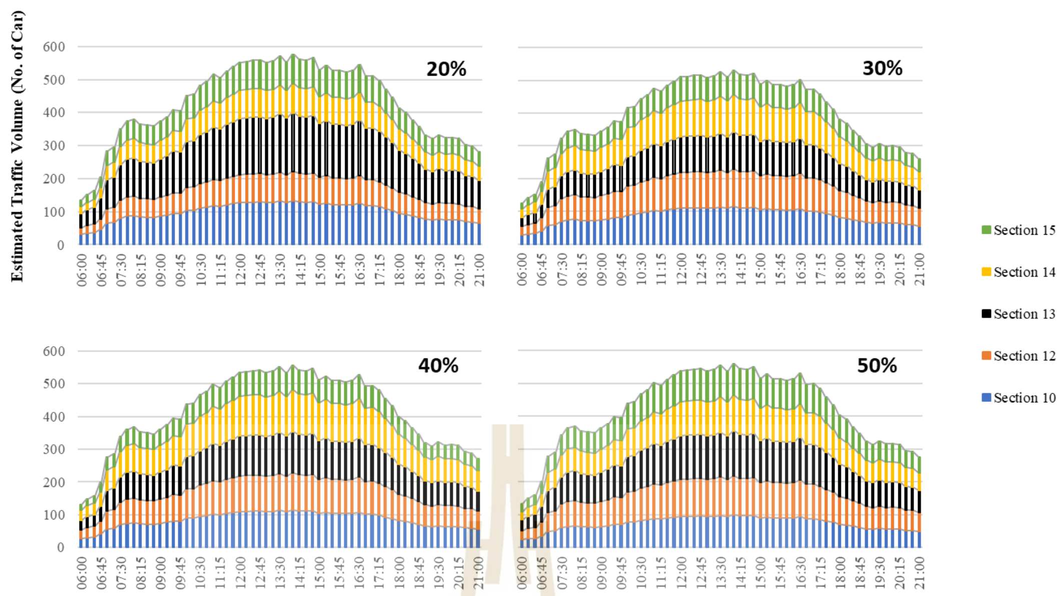

Figure 4.9 Estimated Traffic Volume of Different Road Sections

The cause of traffic congestion in the area cannot be attributed solely to the local traffic volume, but also to the volume of traffic originating from the northeast side and passing through the city in a southwesterly direction. For the purpose of correlation analysis, only five route sections (10, 12, 13, 14, and 15 - Figure 4.8) were included, as these were found to exhibit significant changes in traffic conditions as measured by the Google Traffic Index. Figure 4.9 displays the estimated traffic volume for these monitored routes within the BC search radius of up to 50% detour. The figure presents the relative traffic volume in time-temporal format based on a survey traffic volume at point A and a constant value of BC. The distribution of traffic for the 20% detour shows a significant proportion in Section 13, while the distribution is more evenly spread for detours of 30%, 40%, and 50%.

The accuracy of the estimated traffic volume (V_{all}) was evaluated by conducting a statistical analysis with the TCI obtained from Google Maps and determining the correlation between the estimated traffic volume based on BC and the TCI. As depicted in Figure 4.10, the correlation coefficients for all detour ratios exhibit a similar trend. Notably, the 20% detour ratio yielded the lowest correlation, while the 30%, 40%, and 50% detour ratios demonstrated a clustered pattern. Specifically, the 50% detour ratio displayed the highest correlation coefficient among all days, with the level of BC probability presented in Figure 4.11. Furthermore, the results for weekdays and weekends showed the best correlation on Monday and Saturday, respectively, with reported correlation coefficients of 0.56 and 0.57.

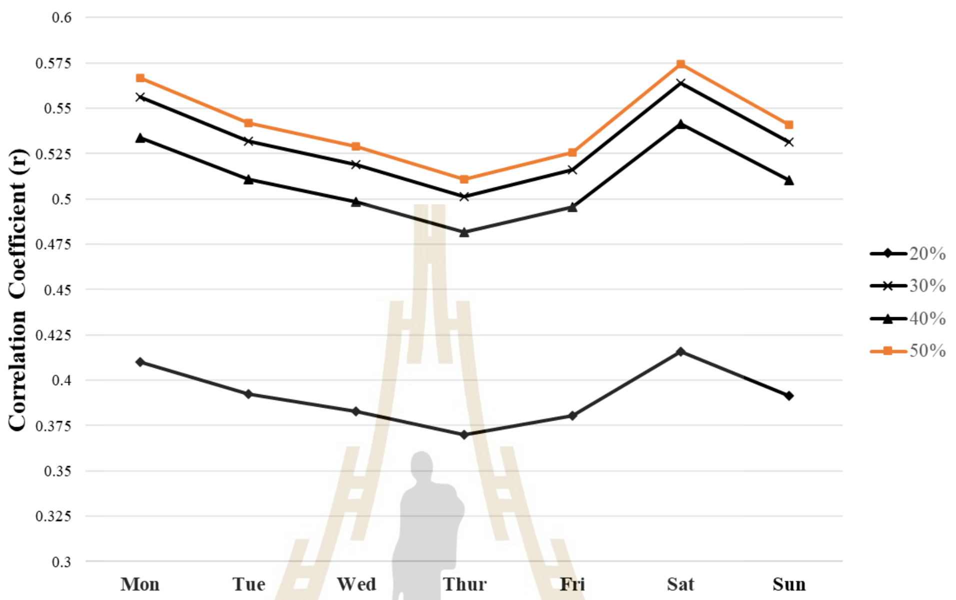

Figure 4.10 Correlation of Estimated Traffic Volume Based on BC and TCI

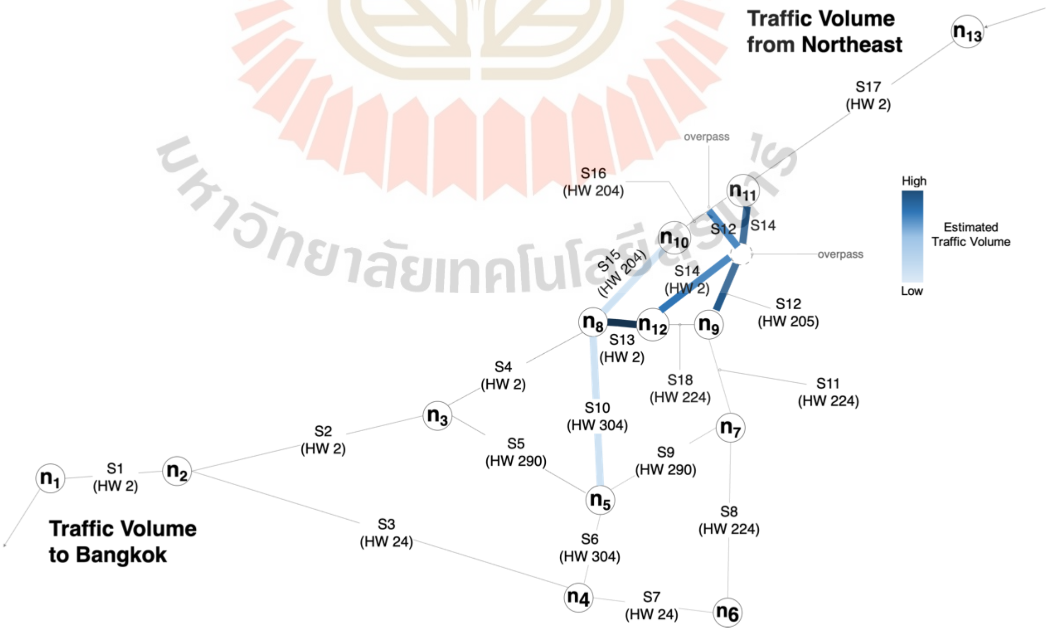

Figure 4.11 Level of the Estimated Traffic Volume at 50% Detour Ratio

Therefore, the proposed method using BC combined with the TCI demonstrated promising results in estimating traffic flow within a road network, specifically in the central area of Nakhon Ratchasima, Thailand. The strong positive correlation between the BC index and Google traffic data indicates the effectiveness of the proposed method. Furthermore, the significant correlation between the estimated traffic volume and the TCI retrieved from Google Maps suggests that the proposed measure is a reliable tool for describing the spread of traffic through the road sections effectively.

The correlation results revealed that the estimated traffic volume for Monday is the most significant among weekday observations, with a correlation coefficient of 0.56. Meanwhile, the result for Saturday is the most significant among weekends, with a correlation coefficient of 0.57. These findings highlight the importance of considering the day of the week when estimating traffic flow, as it can significantly impact the accuracy of the results.

Overall, the proposed method can provide valuable insights into traffic flow patterns and help inform traffic management strategies in urban areas, particularly in locations with high traffic density and congestion. Further research is necessary to assess the generalizability of the proposed method to other road networks and urban areas.

The proposed measure can be used to estimate the distribution of traffic volume from one area to another. However, it is necessary to compare the traffic flow estimated from BC in each road section with the actual traffic density in that section, which could be high due to traffic demand within the area and not from the northeast. Therefore, this method should be validated by comparing the estimated traffic volume with the actual traffic volume collected in the field. Another limitation is that the BC result only explains the impact of traffic congestion on monitored routes, which comprise only a fraction of the entire road network. Unmonitored routes that experience traffic congestion, such as through traffic routes, were excluded from this study. It is crucial to validate the simulation results of through traffic routes by considering other factors that may cause congestion, such as road width or the number of lanes. Thus, before confirming the simulation results, we recommend validating the through traffic routes.

CHAPTER 5
CONCLUSIONS

The key findings that addressed the two primary research questions can be summarized as follows:

How can computer vision aid in reducing manpower for transportation engineering tasks, such as vehicle counting and distinguishing between overcrowded public transport pickup trucks and normal ones?

We have presented a novel approach for the identification of illegal usage of public light pickup trucks, which have been known to pose a significant risk to road safety, was proposed. An object detection system was developed and trained using a variety of state-of-the-art, pretrained YOLOv5 models. The models were subsequently fine-tuned to optimize detection accuracy. The results of the study indicate that this approach is an effective method for identifying overcrowded vehicles and promoting road safety. The results of the study indicate that the large version (L) of the pretrained weights provided an acceptable balance between accuracy and speed, making it the most practical option for the real-world application of detecting overcrowded trucks and ensuring road safety.

We labeled our image dataset by hand and trained on well-known and up-to-date YOLOv5 (You Only Look Once) pretrained models from the smallest (N) to the biggest (X). We compared the finetuned models in terms of both accuracy and speed. We selected the large version (L) because it provided the best balance of speed and accuracy compared with the other models. The chosen model was tested on a video and displayed in the previous section. As per our findings, we conclude that using the YOLOv5L algorithm with our custom dataset is the best option for detecting overcrowded trucks and ensuring road safety.

We recommend further optimization and pruning of the fine-tuned model to reduce its size and improve real-time detection speed.

Can the use of Betweenness Centrality in graph theory be enhanced by combining it with Google Maps data to estimate traffic flow in a predefined network?

We have proposed a simple and cost-effective alternative approach utilizing the combination of the BC index and TCI was presented to estimate traffic flow within a road network. The road network of central Nakhon Ratchasima, Thailand was chosen as the case study to evaluate the efficacy of the proposed method. The simulation results revealed a strong positive correlation between the BC index and Google traffic data. The findings suggest that there is a significant correlation between the estimated traffic volume and the TCI obtained from Google Maps, indicating that the proposed method can effectively capture the distribution of traffic across the road segments. The correlation analysis also revealed that the estimated traffic volume for Mondays was the most significant among weekdays, and Saturdays were the most significant among weekends.

We recommend for the following future study:

1) Validation of the proposed method by comparing estimated traffic volume with actual traffic volume data collected in the field. This will help to determine the accuracy of the proposed method in estimating traffic flow distribution from one area to another.

2) Further investigation of through traffic routes that did not indicate congestion and were excluded from the study. Factors such as road width or number of lanes may be contributing to unchanging congestion on these routes and should be examined.

3) Consideration of traffic demand within a specific area as a factor in determining traffic density. Currently, the proposed method only compares traffic flow in a specific road section to the density of traffic in that section, but traffic demand within the area could also be contributing to the density.

4) Further research on the impact of traffic congestion on through traffic routes and the potential effects on traffic flow distribution in a road network.

In summary, this research presents a novel approach for the detection and classification of overcrowded pickup trucks in traffic using a combination of computer vision and graph theory. The proposed method can be utilized in various applications such as traffic management and road safety, which can help in reducing the number of road accidents caused by overcrowded pickup trucks in urban areas and improving the overall traffic flow. By integrating these advanced technologies, the proposed framework can provide a more efficient way to estimate traffic flow and identify traffic congestion hotspots, enabling policy makers to make more informed decisions on traffic management. Furthermore, by using existing open public data, the proposed method is cost-effective, scalable, and can be applied in multiple locations, making it a valuable tool for traffic management in urban areas. Overall, this research aims to provide a comprehensive solution for traffic management by focusing on both public pickup truck management and traffic flow in urban areas, ultimately leading to safer roads and a more efficient transportation system.

REFERENCES

Ali, K. S., & Abid, N. M. (2021). The Importance of Google Maps for Traffic in Calculating the Level of Service for the Road and Traffic Delay. *IOP Conference Series: Materials Science and Engineering, 1076*(1), 012015. doi:10.1088/1757-899x/1076/1/012015

Alobaidi, M. k., Badri, R. M., & Salman, M. M. (2020). Evaluating the Negative Impact of Traffic Congestion on Air Pollution at Signalized Intersection. *IOP Conference Series: Materials Science and Engineering, 737*(1), 012146. doi:10.1088/1757-899X/737/1/012146

Altshuler, Y., Puzis, R., Elovici, Y., Bekhor, S., & Pentland, A. (2012). Augmented Betweenness Centrality for Mobility Prediction in Transportation Networks. In.

Azad, A. K., & Islam, M. S. (2021, 22-24 Dec. 2021). *Traffic Flow Prediction Model Using Google Map and LSTM Deep Learning.* Paper presented at the 2021 IEEE International Conference on Telecommunications and Photonics (ICTP).

Bastea, T. (2016). Traffic Index: How it works. Retrieved from https://trafficindex.org/how-it-works/

Bernardo, V., Fageda, X., & Flores-Fillol, R. (2021). Pollution and congestion in urban areas: The effects of low emission zones. *Economics of Transportation, 26-27*, 100221. doi:https://doi.org/10.1016/j.ecotra.2021.100221

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934. Retrieved from https://ui.adsabs.harvard.edu/abs/2020arXiv200410934B

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8*(6), 679-698. doi:10.1109/TPAMI.1986.4767851

Carneiro, T., Medeiros Da Nobrega, R. V., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H. C., & Filho, P. P. R. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access, 6*, 61677-61685. doi:10.1109/access.2018.2874767

Carranza-García, M., Torres-Mateo, J., Lara-Benítez, P., & García-Gutiérrez, J. (2021). On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data. *Remote Sensing, 13*(1), 89. Retrieved from https://www.mdpi.com/2072-4292/13/1/89

Cristiani, A. L., Immich, R., Akabane, A. T., Madeira, E. R. M., Villas, L. A., & Meneguette, R. I. (2020). ATRIP: Architecture for Traffic Classification Based on Image Processing. *Vehicles, 2*(2), 303-317. doi:10.3390/vehicles2020017

Crucitti, P., Latora, V., & Porta, S. (2006). Centrality in networks of urban streets. *Chaos: An Interdisciplinary Journal of Nonlinear Science, 16*(1), 015113. doi:10.1063/1.2150162

Elfwing, S. a. U., Eiji and Doya, Kenji. (2017). Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *arXiv*. doi:10.48550/ARXIV.1702.03118

Eraqi, H. M., Soliman, K., Said, D., Elezaby, O. R., Moustafa, M. N., & Abdelgawad, H. (2021). Automatic Roadway Features Detection with Oriented Object Detection. *Applied Sciences, 11*(8), 3531. Retrieved from https://www.mdpi.com/2076-3417/11/8/3531

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry, 40*, 35-41. doi:10.2307/3033543

Gao, S., Wang, Y., Gao, Y., & Liu, Y. (2013). Understanding Urban Traffic-Flow Characteristics: A Rethinking of Betweenness Centrality. *Environment and Planning B: Planning and Design, 40*(1), 135-153. doi:10.1068/b38141

García Ramírez, Y. (2020). *Developing a Traffic Congestion Model based on Google Traffic Data: A Case Study in Ecuador*.

Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). YOLOX: Exceeding YOLO Series in 2021. arXiv:2107.08430. Retrieved from https://ui.adsabs.harvard.edu/abs/2021arXiv210708430G

Gholamhosseinian, A., & Seitz, J. (2021). Vehicle Classification in Intelligent Transport Systems: An Overview, Methods and Software Perspective. *IEEE Open Journal of Intelligent Transportation Systems, 2*, 173-194. doi:10.1109/OJITS.2021.3096756

Gigerenzer, G. (2008). *Rationality for mortals: How people cope with uncertainty*. New York, NY, US: Oxford University Press.

Henry, E., Bonnetain, L., Furno, A., Faouzi, N. E. E., & Zimeo, E. (2019, 5-7 June 2019). *Spatio-temporal Correlations of Betweenness Centrality and Traffic Metrics.* Paper presented at the 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS).

Hillier, B., Penn, A., Hanson, J., Grajewski, T., & Xu, J. (1993). Natural movement: or, configuration and attraction in urban pedestrian movement. *Environment and Planning B: Planning and Design, 20*(1), 29-66. doi:10.1068/b200029

Howard, A. a. S., Mark and Chu, Grace and Chen, Liang-Chieh and Chen, Bo and Tan, Mingxing and Wang, Weijun and Zhu, Yukun and Pang, Ruoming and Vasudevan, Vijay and Le, Quoc V. and Adam, Hartwig. (2019). Searching for MobileNetV3. *arXiv*. doi:10.48550/ARXIV.1905.02244

Intachak, T., & Wacharin, K. (2013). *Vehicle Detection and Classification System for Traffic Video Surveillance*. Paper presented at the The 5th Conference on Application Research and Development (ECTI-CARD), Bangkok, Thailand.

Ioffe, S., & Szegedy, C. (2015). *Batch Normalization Accelerating Deep Network Training by Reducing Internal Covariate Shift*. Paper presented at the 32nd International Conference on Machine Learning, ICML, Lille, France.

Jayasinghe, A., Sano, K., & Nishiuchi, H. (2015). Explaining Traffic Flow Patterns Using Centrality Measures. *International Journal for Traffic and Transport Engineering, 5*.

Jiang, B. (2009). Street hierarchies: a minority of streets account for a majority of traffic flow. *International Journal of Geographical Information Science, 23*(8), 1033-1048. doi:10.1080/13658810802004648

Jocher, G. (2022). V4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration. Retrieved from https://github.com/ultralytics/yolov5/releases/tag/v4.0

Kayid, A. M. (2018). Performance of CPUs/GPUs for Deep Learning workloads. *May*, 25. doi:10.13140/RG.2.2.22603.54563

Kazerani, A., & Winter, S. (2009). *Can Betweenness Centrality Explain Traffic Flow*.

Kim, J. (2020). Vehicle Detection Using Deep Learning Technique in Tunnel Road Environments. *Symmetry, 12*(12). doi:10.3390/sym12122012

Kim, J. a., Sung, J. Y., & Park, S. h. (2020, 1-3 Nov. 2020). *Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition.* Paper presented at the 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia).

Krafta, R. (2001). *Urban Centrality.* Paper presented at the 3rd International Space Syntax Symposium Atlanta 2001.

Kumarage, S., Bandara, S., & De Silva, D. (2018). *Use of Google Traffic Data in Traffic Analysis*.

Kwan, C., Gribben, D., Chou, B., Budavari, B., Larkin, J., Rangamani, A., Etienne-Cummings, R. (2020). Real-Time and Deep Learning Based Vehicle Detection and Classification Using Pixel-Wise Code Exposure Measurements. *Electronics, 9*(6). doi:10.3390/electronics9061014

Li, C., & Managi, S. (2021). Contribution of on-road transportation to PM2.5. *Scientific Reports, 11*(1). doi:10.1038/s41598-021-00862-x

Lurstwut, K., Phothiyod, W., & Ounban, C. (2018). Development Model of Participation to Prevent Road Traffic Accident by Engaging Multilateral Networks in Lamphun Province. *Lanna Public Health Journal, 14*(1), 59.

Mahmoody, A., Tsourakakis, C. E., & Upfal, E. (2016). *Scalable Betweenness Centrality Maximization via Sampling*.

Maungmai, W., & Nuthong, C. (2019, 23–25 February). *Vehicle Classification with Deep Learning.* Paper presented at the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore.

Maurya, S. K., Liu, X., & Murata, T. (2019, 2019). *Fast Approximations of Betweenness Centrality with Graph Neural Networks*.

Nepal, U., & Eslamiat, H. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors, 22*(2), 464. Retrieved from https://www.mdpi.com/1424-8220/22/2/464

Ofir, N., & Nebel, J.-C. (2021). Classic versus deep learning approaches to address computer vision challenges. arXiv:2101.09744. Retrieved from https://ui.adsabs.harvard.edu/abs/2021arXiv210109744O

Online, T. (2016, 10/11/2016). Thairath News Show. Retrieved from https://www.youtube.com/watch?v=ouJskQP6bDw

Online, T. (2019, 09/22/2019). Thairath News Show. Retrieved from https://www.youtube.com/watch?v=KZSGL1kDedE

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., . . . Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Retrieved from http://arxiv.org/abs/1912.01703

Petrovska, N., & Stevanovic, A. (2015, 15-18 Sept. 2015). *Traffic Congestion Analysis Visualisation Tool*. Paper presented at the 2015 IEEE 18th International Conference on Intelligent Transportation Systems.

Piniarski, K., Pawlowski, P., & Dabrowski, A. (2020). Tuning of Classifiers to Speed-Up Detection of Pedestrians in Infrared Images. *Sensors (Basel), 20*(16). doi:10.3390/s20164363

Porta, S., Crucitti, P., & Latora, V. (2006). The network analysis of urban streets: A dual approach. *Physica A: Statistical Mechanics and its Applications, 369*(2), 853-866. doi:10.1016/j.physa.2005.12.063

Porta, S., Strano, E., Iacoviello, V., Messora, R., Latora, V., Cardillo, A., . . . Scellato, S. (2009). Street Centrality and Densities of Retail and Services in Bologna, Italy. *Environment and Planning B: Planning and Design, 36*(3), 450-465. doi:10.1068/b34098

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640. Retrieved from https://ui.adsabs.harvard.edu/abs/2015arXiv150602640R

Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger. arXiv:1612.08242. Retrieved from https://ui.adsabs.harvard.edu/abs/2016arXiv161208242R

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv:1804.02767. Retrieved from https://ui.adsabs.harvard.edu/abs/2018arXiv180402767R

Rezzouqi, H., Gryech, I., Sbihi, N., Ghogho, M., & Benbrahim, H. (2018). *Analyzing the Accuracy of Historical Average for Urban Traffic Forecasting Using Google Maps.* Paper presented at the IntelliSys.

Roecker, M. N., Costa, Y. M. G., Almeida, J. L. R., & Matsushita, G. H. G. (2018a). *Automatic Vehicle type Classification with Convolutional Neural Networks*. Paper presented at the 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP).

Roecker, M. N., Costa, Y. M. G., Almeida, J. L. R., & Matsushita, G. H. G. (2018b). *Automatic Vehicle type Classification with Convolutional Neural Networks.* Paper presented at the 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), Maribor, Slovenia.

Roy, D., Panda, P., & Roy, K. (2020). Tree-CNN: A hierarchical Deep Convolutional Neural Network for incremental learning. *Neural Netw, 121*, 148-160. doi:10.1016/j.neunet.2019.09.010

S. Tabassum, M. S. U., N. H. Al-Nur and S. Shatabda. (2020). *Native Vehicles Classification on Bangladeshi Roads Using CNN with Transfer Learning*. Paper presented at the 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh.

Sevtsuk, A., & Kalvo, R. (2016). Urban Network Analysis Toolbox for Rhinoceros 3D. *Planning and Technology Today*(issue no. 112), 4–5.

Sevtsuk, A., & Mekonnen, M. (2012). Urban Network Analysis Toolbox. *International Journal of Geomatics and Spatial Analysis, 22*(2), pp. 287–305.

Sharma, R. (2017). How Google Maps Gets Its Remarkably Accurate Real-Time Traffic Data. Retrieved from https://gadgets360.com/apps/features/how-google-maps-gets-its-remarkably-accurate-real-time-traffic-data-1665385

Tomko, M., Winter, S., & Claramunt, C. (2008). Experiential hierarchies of streets. *Computers, Environment and Urban Systems, 32*(1), 41-52. doi:10.1016/j.compenvurbsys.2007.03.003

Transportation, D. o. L. (2021). *Transport Statistics Report in 2021*. Retrieved from Bangkok, Thailand:

Transportation, D. o. L. (2022). *Public Transport Accident Analysis Report*. Retrieved from https://web.dlt.go.th/statistics/

Tsapakis, I., Holik, W. A., Vandervalk, A., Turner, S. M., & Habermann, J. (2017). *Innovative Procedures in Traffic Volume Estimation*. Retrieved from

Venkatachalam, T. A., & Gnanavelu, D. (2009). Concentration of heterogeneous road traffic. In *Advanced Technologies*: IntechOpen.

Wallin, M. (2020). Traffic Data, Route Planning, and ETA: How Google Maps Predicts Travel Time. Retrieved from https://www.verizonconnect.com/resources/article/google-maps-travel-time/#:~:text=GPS%20data%20from%20individual%20phones,speed%20when%20conditions%20are%20clear.

Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2021). You Only Learn One Representation: Unified Network for Multiple Tasks. arXiv:2105.04206. Retrieved from https://ui.adsabs.harvard.edu/abs/2021arXiv210504206W

World Health Organization. (2018). *Global Status Report on Road Safety*. Retrieved from Geneva, Switzerland:

WZMIAOMIAO. (2022a). v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. Retrieved from https://github.com/ultralytics/yolov5/releases

WZMIAOMIAO. (2022b). YOLOv5 (6.0/6.1) brief summary #6998. Retrieved from https://github.com/ultralytics/yolov5/issues/6998

Xu, B. a. W., Naiyan and Chen, Tianqi and Li, Mu. (2015). Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv*. doi:10.48550/arxiv.1505.00853

Ye, P., Wu, B., & Fan, W. (2016). Modified Betweenness-Based Measure for Prediction of Traffic Flow on Urban Roads. *Transportation Research Record: Journal of the Transportation Research Board, 2563*(1), 144-150. doi:10.3141/2563-19

Zhu, J., Wang, Z., Wang, S., & Chen, S. (2020). Moving Object Detection Based on Background Compensation and Deep Learning. *Symmetry, 12*(12). doi:10.3390/sym12121965

APPENDIX

APPENDIX A

List of Publications

List of Publications

Suttanuruk, J., Jomnonkwao, S., Ratanavaraha, V., & Kanjanawattana, S. (2023). Convolutional Neural Network for Overcrowded Public Transportation Pickup Truck Detection. Computers, Materials & Continua, 74(3).

Computers, Materials & Continua
DOI: 10.32604/cmc.2023.033900
Article

Tech Science Press
check for updates

Convolutional Neural Network for Overcrowded Public Transportation Pickup Truck Detection

Jakkrit Suttanuruk[1], Sajjakaj Jomnonkwao[1,*], Vatanavong Ratanavaraha[1] and Sarunya Kanjanawattana[2]

[1]School of Transportation Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand
[2]School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand
*Corresponding Author: Sajjakaj Jomnonkwao. Email: sajjakaj@g.sut.ac.th
Received: 30 June 2022; Accepted: 03 October 2022

Abstract: Thailand has been on the World Health Organization (WHO)'s notorious deadliest road list for several years, currently ranking eighth on the list. Among all types of road fatalities, pickup trucks converted into vehicles for public transportation are found to be the most problematic due to their high occupancy and minimal passenger safety measures, such as safety belts. Passenger overloading is illegal, but it is often overlooked. The country often uses police checkpoints to enforce traffic laws. However, there are few or no highway patrols to apprehend offending drivers. Therefore, in this study, we propose the use of existing closed-circuit television (CCTV) traffic cameras with deep learning techniques to classify overloaded public transport pickup trucks (PTPT) to help reduce accidents. As the said type of vehicle and its passenger occupancy characteristics are unique, a new model is deemed necessary. The contributions of this study are as follows: First, we used various state-of-the-art object detection YOLOv5 (You Only Look Once) models to obtain the optimum overcrowded model pretrained on our manually labeled dataset. Second, we made our custom dataset available. Upon investigation, we compared all the latest YOLOv5 models and discovered that the YOLOv5L yielded the optimal performance with a mean average precision (mAP) of 95.1% and an inference time of 33 frames per second (FPS) on a graphic processing unit (GPU). We aim to deploy the selected model on traffic control computers to alert the police of such passenger-overloading violations. The use of a chosen algorithm is feasible and is expected to help reduce traffic-related fatalities.

Keywords: Public vehicle classification; passenger overloading detection; deep learning; CNN; YOLOv5

1 Introduction

Thailand is again included in the World Health Organization (WHO)'s infamous list of the deadliest traffic fatalities worldwide. In a 2018 global status report on road safety, the WHO ranked Thailand 8th out of 175 other countries, an improvement from the previous two surveys in which Thailand ranked 2nd [1]. In 2018, the death toll from road accidents was determined to be 22,491 people, with a fatality rate of around 32.7 per 100,000 people. The number could be significantly worse than it appears to be due to data collection incompetency [2]. Fatalities involving public transportation are typically high in terms of occupancy size (18,584 accidents in 2018, a 33.79% increase from the previous year) [2]. As of April 2022, motorcycles accounted for 52.7% (i.e., 21.8 million) of all vehicles registered in Thailand, followed by sedans (26.6%) and light pickup trucks (17%) [3]. One reason pickup trucks are in high demand is that the government state oil fund subsidizes 50% of every increase in diesel price. Moreover, biodiesel B20, which is a less expensive alternative to regular diesel, has become commercially viable and was endorsed by the Energy Ministry in 2019. This has resulted in the increase in terms of the market share of 1-ton trucks. As previously stated, pickup trucks are cost effective and therefore are modified to serve as a mode of public transportation. Approximately 100,000 public transport pickup trucks (PTPT) are currently registered with the Department of Land Transport, which is equivalent to the US Department of Motor Vehicles. These trucks have taken over the streets of the country, displacing regular buses. Depending on the structure and model, PTPTs can accommodate eight seated passengers and four to six standing passengers. Most truck operators are private companies that must pay additional taxes and fees to operate on designated routes. Therefore, the operators often carry as many passengers as they can. Typically, this means that they wait for passengers to arrive and fill the vehicle. This results in being behind schedule. Thus, they must exceed speed limits, breaking numerous traffic laws and increasing the risk of injury and death. The Thai traffic authority has a limited number of highway patrols to conduct traffic stops. Furthermore, they rely on roadblocks and closed-circuit television (CCTV) cameras to enforce traffic laws to minimize road accidents.

There are two broad categories related to overloading detection: physical sensors and computer vision. Sensors are used in intelligent transportation systems (ITS), such as Bluetooth, laser, radio wave, infrared, etc., to sense the weight of the vehicles. The use of weight sensors combined with loop induction sensors necessitates physical installations, i.e., opening a road surface and blocking traffic. Computer vision can use existing cameras. In case of a new installation, tampering with the road surface and road blocking is minimal. Recently, the use of computer vision has been developed and made available. Compared with ITS solutions, they are less expensive and easier to maintain.

In this study, we focus on computer vision category. Classic machine learning methods, such as feature extraction using histogram of oriented gradients (HOG), classification using support vector machine (SVM), and many others have been used and achieved satisfactory results. In recent years, deep learning techniques have become widely popular, especially since the introduction of various convolution neural network (CNN) algorithms in 2012. Hardware, particularly graphic processing units (GPU), also plays a significant role in reducing the time required to train a model because they are much faster than central processing units (CPU) [4]. However, a more powerful CPU comes at a higher cost and consumes more power. One could opt for online services for alternative cost-efficient solutions, such as Google Colab [5]. As a result, image classification has become more accurate. Specifically, vehicle classification in the modern era is accomplished using deep learning techniques, such as a CNN with the transfer learning method coupled with the power of modern GPUs, which allows for a significant reduction in training time. The pretrained network is considered highly effective because it has previously seen a tremendous number of images with thousands of categories. The mentioned CNN model subsequently adds extra interested classes to the last few layers of the network to enable its classification of newly added targets. In this case, the images of Thai-modified pickups and passenger-overloaded trucks are inputted into the networks to achieve the overcrowded PTPT classifier. However, the inference speed is slow, and we need to find bounding boxes or the locations of the objects of interest, i.e., one regular PTPT and two overcrowded ones, in the vicinity of the image from a CCTV video feed.

Region-based CNNs, such as R-CNN, have been proposed to identify the bounding box of an object of interest by splitting an image into regions. However, this approach is time-consuming. Fast R-CNN and Faster R-CNN are subsequently released to overcome the inference time issues. These methods are classified as two-stage detectors because they comprise, first, the generation of region proposals by selective search as in the R-CNN or by a Region Proposal Network and, second, object classification. Faster R-CNN is over 100 times faster than the traditional R-CNN. However, single-stage detectors, such as the YOLO (You Only Look Once) and its variants, such as the You Only Learn One Representation [6] and the YOLOX [7], an anchor-free version of the YOLO, have recently emerged by directly applying object classification and bounding box regression [8]. Generally, two-stage detectors are more accurate but slower than single-stage detectors, such as the YOLO.

Thus, in this study, we propose a way to accurately detect overcrowded PTPT in real time. Therefore, the algorithm should have an acceptable accuracy and inference speed in the said detection task when using existing traffic cameras. Currently, there is no model trained specifically for this task. Therefore, to bridge this gap, we propose training a model to detect the PTPT. Many peer-reviewed papers recommend the YOLOv5 for real-time object detection tasks [9–11]. We have compared all the pretrained YOLOv5 (version 6) models, such as models N, S, M, L, and X, to find an acceptable one to classify a unique characteristic of Thai public vehicles and their usage concerning public safety. Precisely, the model aims to identify whether or not public pickup trucks used for transportation are overcrowded. We have used the transfer learning technique on our manually labeled 1,239 images in the YOLOv5 PyTorch format [12] in order to recognize the overcrowded and the normal ones. The use of the model is expected to aid in the enforcement of traffic laws, reducing severe traffic mortality. The aforementioned contributions of this study can be summarized as follows:

1. We proposed a real-time deep learning model to detect overcrowded PTPT using existing traffic cameras.
2. We compared the existing state-of-the-art networks of deep learning for real-time Thai public pickup truck detection.
3. We open-sourced our manually labeled dataset of regular PTPT and overcrowded ones.

2 Literature Review

Traditional methods in computer vision are known approaches that do not rely on machine learning but on mathematical algorithms; often, they are outperformed by deep learning methods [13]. An example of these classic algorithms is the Canny edge detector [14], which is used to find curves in an image. Another study using classical approaches was conducted on Thai roadways in 2013 using various techniques, such as background subtraction, edge detection, and graph matching, which obtained 87.24% accuracy with effectiveness of five cars per second [15].

In 2019, a more recent study with a similar environment used a CNN to classify vehicle color and type, i.e., small, medium, large, and unknown. The models were trained on 917 images and achieved color and type classification accuracy of 70.09% and 81.62%, respectively [16].

Roecker et al. [17] used a CNN on the Beijing Institute of Technology-Vehicle dataset, which comprised 9,850 high-resolution vehicle images and obtained approximately 95% of precision, recall, and f-measure for truck classification tasks.

Cristiani et al. [18] proposed a traffic flow classification method based on image processing in conjunction with a luminosity sensor to overcome lighting issues at night and in other low-light environments, such as fog and rain. They used an Adaboost Classifier image processing technique to categorize road congestion. They achieved an overall accuracy of 92% in traffic flow classification tasks.

Meanwhile, Piniarski et al. [19] proposed a method to speed up pedestrian detection in infrared images, resulting in a 74% reduction in classification time. The classifiers analyzed in this study were a HOG with an SVM, an aggregate channel feature, and a deep CNN. The authors confirmed that their CNN model obtained the best accuracies for various databases.

Roy et al. [20] proposed the use of Tree-CNN to address a "catastrophic forgetting" problem in which adding new information, such as transfer learning techniques, tends to complicate the trained model. The tree-like architecture can grow and learn to accommodate new classes or data adaptively.

As mentioned in the previous section, recent approaches in object detection can be broadly categorized into two categories, namely, single-stage (YOLO) and two-stage algorithms (Faster R-CNN). Several studies have concluded that the YOLO algorithm outperformed the Faster R-CNN in terms of accuracy and inference time.

A study conducted by Kwan et al. [10] on real-time vehicle detection and classification adopted a method called pixel-wise coded exposure to condense multiple frames into a single frame. They directly captured sample videos from pixel-wise coded aperture cameras. The input image was resized to 448 × 448 pixels and input into a YOLOv1. The authors achieved decent results for real-time tasks in terms of speed and accuracy.

In 2020, Kim [21] introduced techniques to detect vehicles driving ahead in tunnel environments. The techniques used past object detection (YOLOv2) and discovered noise reduction and illuminance smoothing, which was applied to image frames obtained from dashcams. The authors reported a 94.1% improvement in vehicle detection accuracy. There is another study conducted by Zhu et al. [22] on moving vehicles (airplane, car, and person) detection using the YOLOv3 in conjunction with background subtraction image frames in videos. They reported 91% mAP and 27 frames per second (FPS).

In Bangladesh, researchers used the YOLO to detect their native vehicles, such as trucks and achieved a 73% IOU (intersect over union) at 55 FPS [23].

In Table 1, we have tabulated existing works that are closely related to this study. As previously stated, there is no existing detection model to identify pickup trucks that are overloaded with passengers. Consequently, the authors have developed a model to accomplish the said task.

Table 1: Summary of related studies

Author	Country (year)	Method	Camera angle	mAP (%)	FPS	Pickup truck	Overcrowded pickup truck
Maungmai et al. [16]	Thailand (2019)	CNN	Front	84.6*	NA	Classification	X

(Continued)

Table 1: Continued

Author	Country (year)	Method	Camera angle	mAP (%)	FPS	Pickup truck	Overcrowded pickup truck
Roecker et al. [17]	Brazil (2018)	CNN	Front	93.9∗	NA	Classification∗	X
Kim [21]	Korea (2020)	YOLOv2	Back	94.1∗	NA	Detection∗	X
Zhu et al. [22]	China (2020)	YOLOv3-SOD	All	91.0	27	Detection∗	X
Tabassum et al. [23]	Bangladesh (2020)	YOLO	All	73.0∗∗	NA	Detection	X
This research	Thailand	YOLOv5	Back, Top Down	95.1	33	Detection	Detection
Remarks				*∗ accuracy* *∗∗ iou*		*∗ average all vehicle types*	

3 Materials and Methods

The procedures of this study are summarized, as shown in Fig. 1. We began constructing our dataset as described in the next section by collecting various images and videos. Subsequently, we proceeded to annotate parts of still images that represent two classes of normal and overcrowded PTPT. Once we obtained the dataset, we augmented the originals by applying various techniques, as listed in Table 2. In the training phase, we finetuned the pretrained YOLOv5 models obtained from the authors' official GitHub [24]. Following that, we compared all finetuned models for the mAP and speed in terms of FPS for our inference task.

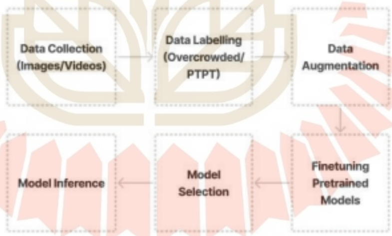

Figure 1: Process flow diagram

Table 2: Data augmentation

Steps	Detail
Preprocessing	Fix orientation Resize: Fill (with center crop) in 640 × 640 Auto adjust contrast: Using histogram equalization

(Continued)

Table 2: Continued

Steps	Detail
Augmentation	Flip horizontally Crop: 0% minimum zoom, 2.5% maximum zoom Rotation: Between −15° and +15° Shear: ±15° horizontal, ±15° vertical Grayscale: Apply to 10% of images Hue: Between −25° and +25° Saturation: Between −25% and +25% Brightness: Between −25% and +25% Exposure: Between −25% and +25% Blur: Up to 1.75 px Noise: Up to 5% of pixels

3.1 Datasets

The sample images were obtained from Google Image Search, photos taken by the authors in local city areas, video clips from news media, and traffic cameras (Hikvision 4 MP Bullet Network Camera) installed approximately 12 m high on three different pedestrian bridges above highways in nearby cities. A sample image from the traffic camera can be observed in Fig. 2. The authors used an Apple iPhone 12's main camera (12 million pixels) to capture still images and 1080 p video at 30 FPS from a car's dashboard level. The sample comprised two main categories of PTPT: 512 images of the modified pickups (PTPT), 431 images of the crowed ones (overcrowded PTPT), and 346 background images, which resulted in a total of 1,289 images. We manually annotated each image in a YOLOv5 PyTorch TXT format (class_id center_x center_y width height). As shown in Table 2, the input images were augmented per item and batch to a final total of 2,975 images. The dataset was randomly separated into 80% training set, 10% validation set, and 10% test set. An example of both datasets is shown in Figs. 3 and 4.

Figure 2: S × S grid

Figure 3: Public transport pickup trucks (PTPT)

Figure 4: Overcrowded public transportation pickup trucks (Overcrowded PTPT)

3.2 *YOLOv5 Architecture*

YOLO, which was first introduced by Redmon et al. [25] in 2015, uses a CNN for object detection and bounding box prediction all at once in a single forward propagation. YOLO takes an input image and divides it into an equal grid (s × s size), as shown in Fig. 2. Furthermore, image classification and bounding box prediction are applied to each grid. The algorithm then outputs the corresponding class probabilities (i.e., overcrowded or PTPT), coordinates of the predicted object, and the confidence score of the boxes. In the subsequent YOLOv2, Redmond et al. added batch normalization, which was initially introduced by Ioffe et al. [26], and a few other techniques to improve the mean average precision [27]. They Redmon et al. have also used Darknet-53 in the feature extraction layers of the network, improving detection speed and accuracy [28]. YOLOv4 was then proposed to use CSP-Darknet53 as a backbone to address YOLOv4's issue in detecting large objects [29].

The YOLOv5 (v6.0/6.1) continued to use CSPDarknet53 as a backbone and replaced the Focus layer, which was introduced in the earlier version to a 6 × 6 Conv2d. Furthermore, the spatial pyramid pooling was then replaced with the SPPF (F for fast). The SPPF layer increases the receptive field and separates the most important features from the backbone. The YOLOv5 version 6.0/6.1 [24] used the same head as in YOLOv3 and YOLOv4. Fig. 5 shows the architecture of the latest YOLOv5 model L. In summary, Yolov5 now consists of three parts, namely, backbone (new CSP-Darknet53) for feature formation, neck (SPPF, new CSP-PAN) for feature extraction, and head (YOLOv3 Head) for detection. C3 denoted in backbone and neck comprises three convolutional layers and a module cascaded by two types of bottlenecks. Finally, the last three convolution layers in the head of the network are detection modules that generate three different feature maps. It is worth mentioning that since the YOLOv5 version 4.0 [30], all activation functions (LeakyReLU or Leaky rectified linear unit [31] and HardSwish [32]) used in convolutional layers are replaced by the Sigmoid linear unit [33] to simplify the architecture.

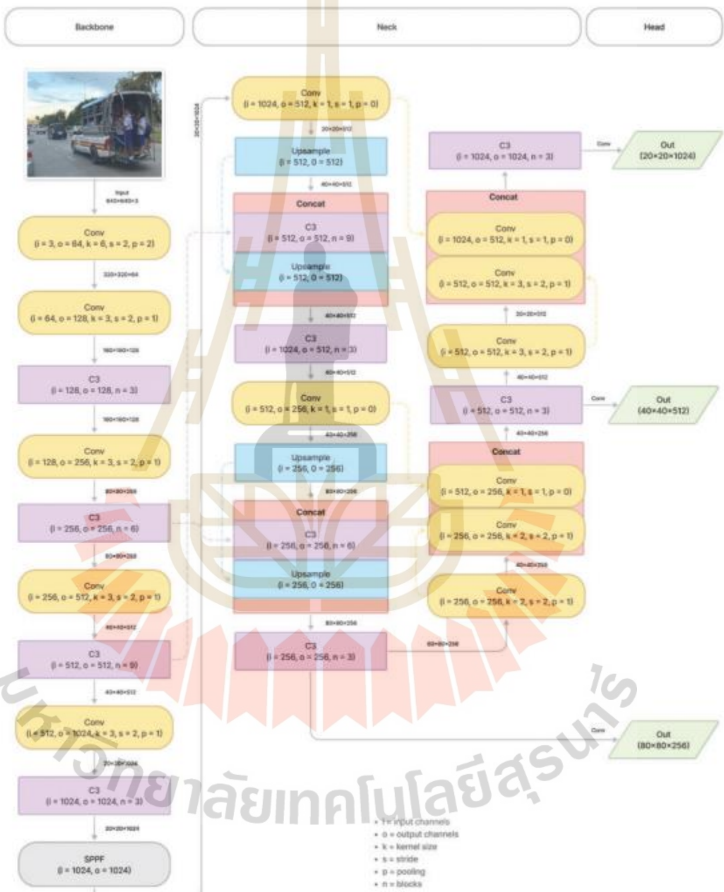

Figure 5: Model structure of YOLOv5L (You Only Look Once)

3.3 Finetuning

The authors employed a transfer learning technique to train the models because of its simplicity and shorter training time. We have used the pretrained YOLOv5 models N, S, M, L, and X to recognize overcrowded PTPT. All pretrained weights were obtained from the official GitHub repository. All models were trained and inferred to Ubuntu 21.04 with PyTorch 1.8 on an NVIDIA GeForce RTX2070. We then set the training hyperparameters to default without freezing the backbone (F = 0) for 1,200 epochs with early stopping. The training batch size was set to the maximum number to fully use the GPU memory. Furthermore, the image size was set to 640 × 640 to match the pretrained models. The number of classes was then set to 2.

3.4 Performance Metrics

To test and compare the models, we used standard evaluation indicators of precision, recall, f1 score, average precision (AP), mAP, and FPS. Precision is the ratio of correctly detected to all detected labels, how many retrieved items are relevant. Recall is the ratio of correctly detected to all positive targets, how many relevant items are retrieved. F_1 score is the harmonic mean of precision and recall. The AP metric is known to be a way to describe a precision-recall curve representing the average of all precision. The mean average precision is simply an average of the AP of all classes. The final evaluation metric is the FPS, which describes the number of images that can be processed in one second. Eqs. (1)–(5) define the formulas as follows:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = \frac{2 * \text{Precision} * Recall}{\text{Precision} + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{3}$$

where TP = true positive, FP = false positive, FN = false negative

$$AP = \int_0^1 p(r)\,dr \tag{4}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{5}$$

4 Results and Discussion

In this section, we compare the five pretrained YOLO models, which include YOLOv5n, YOLOv5s, YOLOv5 m, YOLOv5L, and YOLOv5x. These pretrained weights can be found and downloaded directly from GitHub [34]. The standard metric used to evaluate the object detection model is the mean AP, as described in Eq. (5). Higher mAP values were noted to imply better accuracy. All five models were trained for 1,200 epochs with early stopping, as seen in the "Best epoch" column shown in Table 3. The precision, recall, and F_1 values are also shown in the same table and plotted in Fig. 6. As shown in Fig. 7, our models dropped in losses steeply in the early steps and gradually converged. This coincided with an increase in accuracy.

Table 3: Evaluation metrics for different YOLOv5 models

Model	Runtime (secs)	Best epoch	mAP_0.5 (%)	mAP_0.5:0.95 (%)	Precision (%)	Recall (%)	F_1 (%)
Model X	71973	190	96.44	70.63	94.47	92.29	93.36
Model L	**39195**	**239**	**95.11**	**65.65**	**95.94**	**93.84**	**94.88**
Model M	29382	296	95.66	69.14	92.60	95.38	93.97
Model S	15736	366	96.32	67.02	90.91	95.40	93.10
Model N	7691	297	96.06	66.63	97.51	90.70	93.98

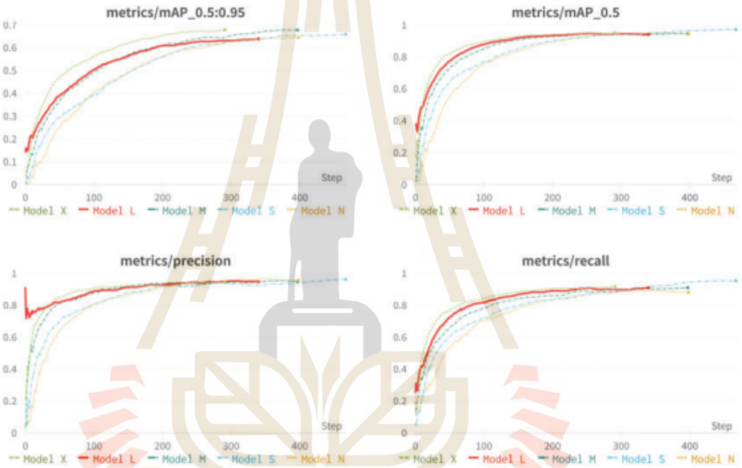

Figure 6: Precision, recall, and mean average precision on five YOLOv5 models

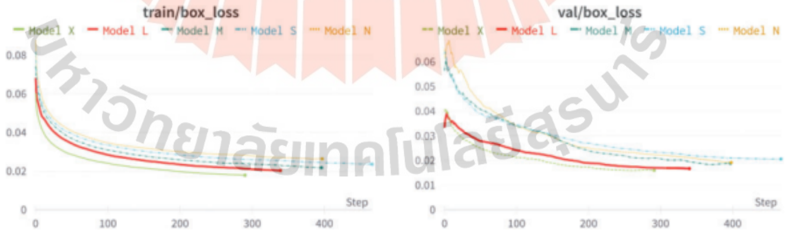

Figure 7: (Continued)

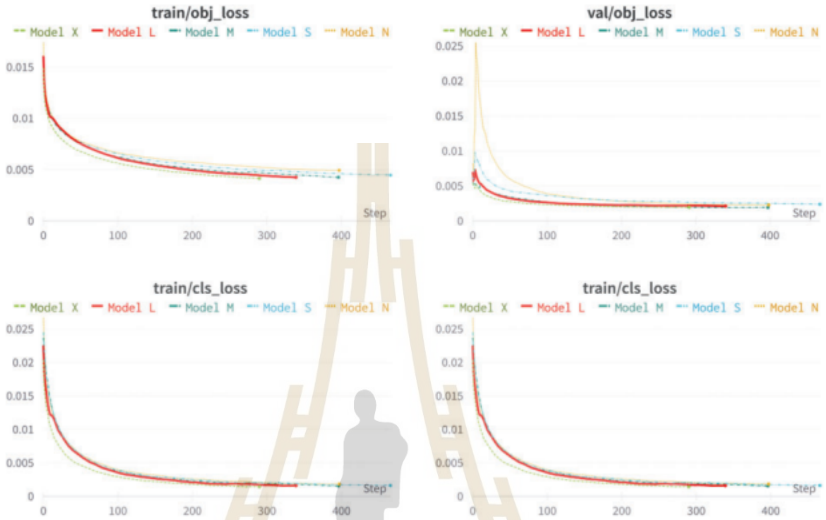

Figure 7: Train and validation loss

Because our application calls for real-time use, we need to also measure the FPS. The FPS is a reciprocal of the inference time taken by the model per image and is summarized in Table 4. As can be seen, the selected model should not be used on an edge device because the FPS is too low for real-time applications. It is also worth mentioning that different batch sizes and computer environment settings will result in different detection speeds.

Table 4: Inference speed comparison

Model	Batch Size	FPS (GPU)	FPS (CPU)
Model X	8	19	1
Model L	**8**	**33**	**2**
Model M	8	66	4
Model S	8	156	9
Model N	8	256	20

All trained models were similar in terms of precision and recall (see Fig. 8) as well as F_1 score (see Fig. 9). The models showed similar results in confusion matrix diagrams (see Fig. 10). Therefore, selecting the model to use was a trial-and-error process. In this study, we chose YOLOv5L with a confidence threshold of 0.60 and an IOU threshold of 0.55. These results were satisfied on both the GPU and CPU based on the accuracy, as shown in Figs. 11 and 12.

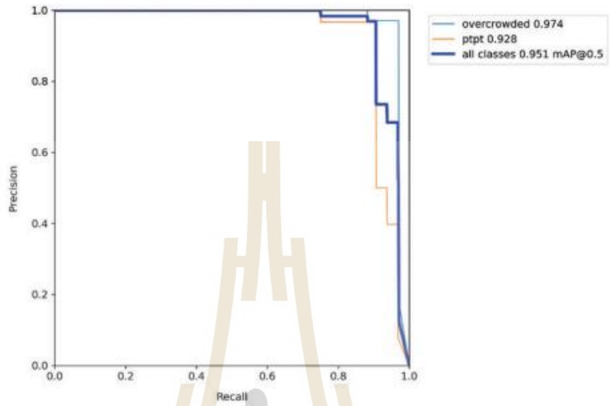

Figure 8: Precision-recall curve of YOLOv5L

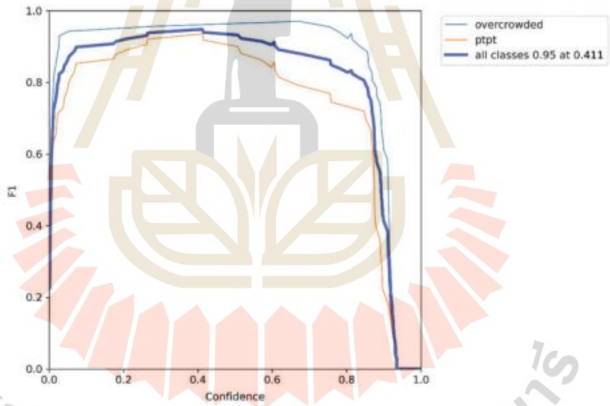

Figure 9: F1 plot of YOLOv5L

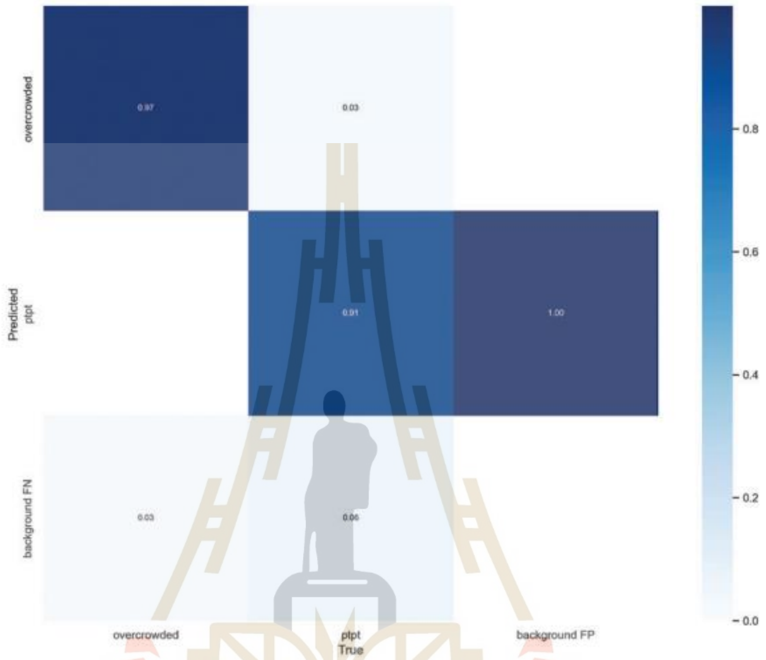

Figure 10: Confusion matrix of YOLOv5L

Figure 11: Overcrowded detection by YOLOv5L [35]

Figure 12: Multiclass detection by YOLOv5L [36]

5 Conclusions

In this study, we have focused in developing an object detection system to detect the illegal use of public light pickup trucks that could lead to road injury and fatality. We labeled our image dataset by hand and trained on well-known and up-to-date YOLOv5 (You Only Look Once) pretrained models from the smallest (N) to the biggest (X). We compared the finetuned models in terms of both accuracy and speed. We selected the large version (L) because it provided the best balance of speed and accuracy compared with the other models. The chosen model was tested on a video and displayed in the previous section. As per our findings, we conclude that using the YOLOv5L algorithm with our custom dataset is the best option for detecting overcrowded trucks and ensuring road safety. However, our model was performed at about 2 FPS on the CPU which considered too low for a real-time detection task. GPU utilization could be used to gain better FPS in the meantime. An optimization to achieve higher frame rate is recommended in the future. Furthermore, standing passengers in the back of an overcrowded PTPT could obscure the license plate from camera view. This would then obstruct a traffic enforcement system from capturing photographic evidence of violating vehicles. Improvements should thus be addressed in the future.

Acknowledgement: We are grateful to the people behind the libraries and frameworks used in this study, specifically Ultralytics and Facebook PyTorch, which provided all the tools needed in the modern deep learning industry.

Funding Statement: This work was supported by (i) Suranaree University of Technology, (ii) Thailand Science Research and Innovation, and (iii) National Science Research and Innovation Fund (Grant Number: RU-7-706-59-03).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding this study.

References

[1] World Health Organization, "Global status report on road safety," Geneva, Switzerland, 7 November 2018.
[2] K. Lurstwut, W. Phothiyod and C. Ounban, "Development model of participation to prevent road traffic accident by engaging multilateral networks in lamphun province," *Lanna Public Health Journal*, vol. 14, no. 1, pp. 59, 2018.
[3] Department of Land Transportation, "Transport statistics report in 2021," Bangkok, Thailand, 2021.

[4] A. M. Kayid, "Performance of cpus/gpus for deep learning workloads," Research report, Faculty of Media Engineering and Technology, The German University in Cairo, Cairo, Egypt, 2018.

[5] T. Carneiro, R. V. Medeiros Da Nobrega, T. Nepomuceno, G. -B. Bian, V. H. C. De Albuquerque *et al.*, "Performance analysis of google colaboratory as a tool for accelerating deep learning applications," *IEEE Access*, vol. 6, pp. 61677–61685, 2018.

[6] C. -Y. Wang, I. -H. Yeh and H. -Y. M. Liao, "You only learn one representation: Unified network for multiple tasks," arXiv:2105.04206, pp. 1–11, 2021.

[7] Z. Ge, S. Liu, F. Wang, Z. Li and J. Sun, "Yolox: Exceeding yolo series in 2021," arXiv:2107.08430, pp. 1–7, 2021.

[8] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez and J. García-Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sensing*, vol. 13, no. 1, pp. 89, 2021.

[9] J. -a. Kim, J. -Y. Sung and S. -h. Park, "Comparison of faster-rcnn, yolo, and ssd for real-time vehicle type recognition," in *2020 IEEE Int. Conf. on Consumer Electronics-Asia (ICCE-Asia)*, Seoul, Korea, pp. 1–4, 2020.

[10] C. Kwan, D. Gribben, B. Chou, B. Budavari, J. Larkin *et al.*, "Real-time and deep learning based vehicle detection and classification using pixel-wise code exposure measurements," *Electronics*, vol. 9, no. 6, 1014, pp. 1–21, 2020.

[11] U. Nepal and H. Eslamiat, "Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs," *Sensors*, vol. 22, no. 2, pp. 464, 2022.

[12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury *et al.*, "Pytorch: An imperative style, high-performance deep learning library," arXiv:1912.01703, pp. 1–12, 2019.

[13] N. Ofir and J. -C. Nebel, "Classic versus deep learning approaches to address computer vision challenges," arXiv:2101.09744, pp. 1–6, 2021.

[14] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[15] T. Intachak and K. Wacharin, "Vehicle detection and classification system for traffic video surveillance," in *The 5th Conf. on Application Research and Development (ECTI-CARD)*, Bangkok, Thailand, 2013.

[16] W. Maungmai and C. Nuthong, "Vehicle classification with deep learning," in *2019 IEEE 4th Int. Conf. on Computer and Communication Systems (ICCCS)*, Singapore, pp. 294–298, 23–25 February 2019.

[17] M. N. Roecker, Y. M. G. Costa, J. L. R. Almeida and G. H. G. Matsushita, "Automatic vehicle type classification with convolutional neural networks," in *2018 25th Int. Conf. on Systems, Signals and Image Proc. (IWSSIP)*, Maribor, Slovenia, pp. 1–5, 2018.

[18] A. L. Cristiani, R. Immich, A. T. Akabane, E. R. M. Madeira, L. A. Villas *et al.*, "Atrip: Architecture for traffic classification based on image processing," *Vehicles*, vol. 2, no. 2, pp. 303–317, 2020.

[19] K. Piniarski, P. Pawłowski and A. Dąbrowski, "Tuning of classifiers to speed-up detection of pedestrians in infrared images," *Sensors*, vol. 20, no. 16, pp. 4363, 2020.

[20] D. Roy, P. Panda and K. Roy, "Tree-cnn: A hierarchical deep convolutional neural network for incremental learning," *Neural Netw*, vol. 121, pp. 148–160, Jan 2020.

[21] J. Kim, "Vehicle detection using deep learning technique in tunnel road environments," *Symmetry*, vol. 12, no. 12, 2012, 2020.

[22] J. Zhu, Z. Wang, S. Wang and S. Chen, "Moving object detection based on background compensation and deep learning," *Symmetry*, vol. 12, no. 12, 1965, pp. 1–17, 2020.

[23] M. S. U. S. Tabassum, N. H. Al-Nur and S. Shatabda, "Native vehicles classification on Bangladeshi roads using cnn with transfer learning," in *2020 IEEE Region 10 Symp. (TENSYMP)*, Dhaka, Bangladesh, pp. 40–43, 2020.

[24] WZMIAOMIAO, "Yolov5 (6.0/6.1) brief summary #6998," Github. https://github.com/ultralytics/yolov5/issues/6998 (accessed on 03 March 2022).

[25] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," arXiv:1506.02640, pp. 1–10, 2015.

BIOGRAPHY

Jakkrit Suttanuruk, a researcher and engineer with a passion for innovation, has experienced the transition from the analog to the digital world. Born in the 1970s, Jakkrit has witnessed the remarkable technological advancements that have shaped our society. In 1996, Jakkrit began his academic journey by pursuing a degree in Transportation Engineering at Suranaree University of Technology. During this time, he gained a solid foundation in engineering principles and developed a good understanding in both civil and transportation fields.

Jakkrit embarked on a new adventure at the Army Signal School in Fort Gordon, Georgia. There, he focused on satellite communication and earned a certificate in this specialized field. After his time in the Army, Jakkrit continued his career in the civil and structural engineering fields, applying the valuable expertise he had acquired. He dedicated himself to refining his abilities as an engineer, contributing to various projects and gaining practical experience in these specialized areas.

While concurrently working in the industry, Jakkrit pursued higher education at the University of Nevada, Las Vegas, where he successfully completed a Master's degree in Civil Engineering. This educational endeavor fortified his understanding of the principles and practices in civil, structural, and transportation engineering.

Moreover, during his academic pursuits, Jakkrit expanded his skill set beyond traditional engineering domains. He ventured into the world of computer software and web development, acquiring proficiency in various programming languages. Jakkrit delved into the creation of web applications, honing his expertise in both frontend and backend development. Additionally, his exploration extended into the realm of artificial intelligence (AI), where he sought to understand its applications in the field of transportation engineering.

In 2018, Jakkrit embarked on a significant academic endeavor - a Ph.D. program under the guidance of Prof. Dr. Vatanavong Ratanavaraha and Asst. Dr. Sajjakaj Jomnonkwao. His research focused on "YOLO-BASED IMAGE ANALYSIS AND TRAFFIC FLOW ESTIMATION IN URBAN AREAS USING BETWEENNESS CENTRALITY." Jakkrit sought to bridge the worlds of civil, transportation, and structural engineering with computer technology and artificial intelligence (AI). Recognizing the transformative potential of these cutting-edge fields, he endeavored to apply them to address real-world challenges in his areas of expertise.