การแบ่งส่วนภาพเชิงความหมายด้วยเทคนิคการเรียนรู้เชิงลึกบนชุดข้อมูลภาพท้องถนนในกรุงเทพมหานคร

นายกฤษพล ธิติสิริเวช

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564

Semantic Image Segmentation Using Deep Learning Techniques on the Bangkok Urbanscapes Dataset

Mr. Kitsaphon Thitisiriwech

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021

หัวข้อวิทยานิพนธ์	การแบ่งส่วนภาพเชิงความหมายด้วยเทคนิคการเรียนรู้เชิงลึกบนชุดข้อมูลภาพท้องถนนในกรุงเทพมหานคร
โดย	นายกฤษพล ธิติสิริเวช
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	อาจารย์ ดร.พิตติพล คันธวัฒน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พีรพล เวทีกูล)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
(อาจารย์ ดร.พิตติพล คันธวัฒน์)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.นวลวรรณ สุนทรภิษัช)

กฤษพล ธิติสิริเวช : การแบ่งส่วนภาพเชิงความหมายด้วยเทคนิคการเรียนรู้เชิงลึกบนชุดข้อมูลภาพท้องถนนในกรุงเทพมหานคร. (Semantic Image Segmentation Using Deep Learning Techniques on the Bangkok Urbanscapes Dataset) อ.ที่ปรึกษาหลัก : ศ. ดร.บุญเสริม กิจศิริกุล, อ.ที่ปรึกษาร่วม : อ. ดร.พิตติพล คันธวัฒน์

การแบ่งส่วนเชิงความหมายบนชุดข้อมูลภาพท้องถนนสามารถนำมาประยุกต์กับระบบขับเคลื่อนอัตโนมัติที่สามารถอำนวยความสะดวกแก่ผู้ขับขี่ และมีส่วนสำคัญในการลดอุบัติเหตุบนท้องถนน โดยระบบขับเคลื่อนอัตโนมัติที่ปลอดภัยนั้นจะต้องมีคุณสมบัติที่ดีคือสามารถทำงานได้อย่างแม่นยำในทุกภูมิประเทศ ซึ่งนำมาสู่ปัญหาในงานวิจัยนี้ โดยประการแรกการขาดแคลนชุดข้อมูลถนนประเทศไทยโดยเฉพาะในเมืองกรุงเทพมหานคร และประการที่สองสถาปัตยกรรมการเรียนรู้เชิงลึกโดยวิธีมาตรฐานนั้นยังให้ความแม่นยำไม่ได้มากพอที่จะนำไปประยุกต์กับระบบนี้ โดยวิทยานิพนธ์นี้จึงนำเสนอชุดข้อมูลถนนในกรุงเทพมหานครที่ประกอบด้วยภาพถ่ายนำเข้าและภาพผลเฉลยเป็นจำนวน 701 ภาพ ประกอบกับนำเสนอสถาปัตยกรรมใหม่ DeepLab-V3-A1 ด้วยการปรับปรุงโมเดล DeepLab-V3+ ด้วยการเพิ่มชั้นคอนโวลูชัน 1 x 1 ที่มีจำนวนแตกต่างกันในด้านดีโคดเดอร์ เพื่อเสริมประสิทธิภาพสถาปัตยกรรมต้นแบบ DeepLab-V3+ โดยชุดข้อมูลที่นำมาใช้วัดผลประกอบด้วยชุดข้อมูลถนนกรุงเทพมหานคร (The Bangkok Urbanscapes), The CamVid (ในเมืองเคมบริดจ์), และ The Cityscapes (50 เมืองจากยุโรปโดยเฉพาะในประเทศเยอรมัน) ผลการทดลองด้วยวิธีที่นำเสนอแสดงให้เห็นถึงประสิทธิภาพในการแบ่งส่วนภาพถ่ายเชิงความหมายได้ดีกว่าวิธีการมาตรฐานด้วยมาตรวัดเหล่านี้ Precision, Recall, F1 Score, และ Mean IoU

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อนิสิต .....
ปีการศึกษา 2564 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....
ลายมือชื่อ อ.ที่ปรึกษาร่วม .....

6270011821 : MAJOR COMPUTER SCIENCE

KEYWORD: Semantic segmentation, Deep learning, The Bangkok Urbanscapes dataset

Kitsaphon Thitisiriwech : Semantic Image Segmentation Using Deep Learning Techniques on the Bangkok Urbanscapes Dataset. Advisor: Prof. BOONSERM KIJSIRIKUL, Ph.D. Co-advisor: PITTIPOL KANTAVAT, Ph.D.

Semantic segmentation on the urbanscapes dataset can apply to the self-automation systems. It can assist the driver in reducing the workforce in the long journey. This accurate system can also significantly reduce traffic-accidental cases. This system cannot operate safely without self-localization driving which is appropriate for all landscapes. It leads to the problem in our thesis that lacking the dataset would be the main topic for developing this system to apply self-driving cars in Thailand. In addition, the baseline deep convolutional neural networks for semantic segmentation architectures are not suitable to apply because it is not outperforming for all measurements. This thesis proposes the Bangkok Urbanscapes dataset, which contains the pair of input images and labels for 701 images. Furthermore, we also propose the improved version of DeepLab-V3+ as DeepLab-V3-A1, which refines the decoder side of DeepLab-V3+ with the different number of 1 x 1 convolution kernels. All methods are measured for these datasets: The Bangkok Urbanscapes (our proposed dataset), the CamVid, and the Cityscapes datasets. The experimental results show that our proposed methods outperform in terms of Precision, Recall, F1 Score, and Mean IoU.

Field of Study: Computer Science Student's Signature .....

Academic Year: 2021 Advisor's Signature .....

Co-advisor's Signature .....

กิตติกรรมประกาศ

ขอขอบพระคุณ ศ. ดร. บุญเสริม กิจศิริกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้มอบหลายสิ่งให้กับผม อาจารย์ได้ทุ่มเทเวลาในการให้คำปรึกษาและคอยตรวจแก้ไขวิทยานิพนธ์รวมถึงวารสารเชิงวิชาการในระดับนานาชาติ เพื่อให้ผลงานเชิงวิชาการของผมมีความครบถ้วนและสมบูรณ์ที่สุด และอาจารย์ได้มอบความรักความเมตตาให้กับผมมาตลอดในช่วง 2 ปีที่ผ่านมานี้ ซึ่งทำให้ผมได้เรียนปริญญาโทได้อย่างมีความสุขปราศจากอุปสรรค์ใดๆทั้งปวง

ขอขอบพระคุณ ศ. ดร. ยูจิ อิวาโฮริ อาจารย์ที่ปรึกษาร่วมจากมหาวิทยาลัยชูบุ ที่ดูแลผมเป็นอย่างดีตลอดช่วงการทำวิจัย และได้มอบคำแนะนำที่เป็นประโยชน์ต่อวิทยานิพนธ์ฉบับนี้ และขอขอบพระคุณ อ. ดร. พิตติพล คันธวัฒน์ ที่มาเป็นอาจารย์ที่ปรึกษาร่วม

ขอขอบพระคุณ ผศ. ดร. พีรพล เวทีกูล และรศ. ดร. นวลวรรณ สุนทรภิษัช ที่ให้เกียรติมาเป็นประธานกรรมการสอบ และกรรมการภายนอก สำหรับการสอบวิทยานิพนธ์ในครั้งนี้ คำแนะนำของอาจารย์ทั้งสองท่านส่งผลบวกต่อวิทยานิพนธ์นี้เป็นอย่างมาก

ขอขอบคุณ ดร. ธีรพงศ์ ปานบุญยืน (พี่เก้า) ที่ได้ช่วยเหลือและให้คำปรึกษาเชิงเทคนิคในงานวิทยานิพนธ์ฉบับนี้ พี่เก้าเป็นแบบอย่างที่ดีทั้งในฐานะนักวิจัยและเป็นพี่ชายที่ใจดีสำหรับผม พี่เก้ามักจะมีข้อคิดดีๆ และได้มอบกำลังใจให้ผมอย่างสม่ำเสมอ

ขอขอบคุณทุนอุดหนุนวิทยานิพนธ์สำหรับนิสิตจากบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย และทุนจากโครงการ SATREPS Project of JST และทุน JICA ในหัวข้อ "Smart Transport Strategy for Thailand 4.0 Realizing better quality of life and low-carbon society" ที่ได้มอบโอกาสในการทำวิทยานิพนธ์นี้ครับ

ขอขอบคุณเพื่อนร่วมรุ่นในระดับมหาบัณฑิตภาควิชาวิศวกรรมคอมพิวเตอร์ รวมถึงพี่ๆ น้องๆ ใน MIND Lab และแลป Datamind รวมถึงมิตรสหาย ที่ได้มอบกำลังใจและช่วยเหลือดูแลกันจนจบการศึกษา

สุดท้ายนี้ ขอขอบพระคุณป๊ากับม๊า อาแปะ อาอึ้ม อาโกว อาเจ็ก น้องจู น้องธี ที่เป็นกำลังใจสำคัญในการเรียน และคอยสนับสนุนความฝันของผมในช่วง 2 ปีที่ผ่านมานี้

กฤษพล ธิติสิริเวช

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูปภาพ

หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันเทคนิคการรู้จำภาพถ่ายมีบทบาทสำคัญในการพัฒนาแอปพลิเคชันที่สามารถลดขั้นตอนการทำงานของมนุษย์ที่ซ้ำซ้อนลงได้ด้วยระบบอัตโนมัติที่มีความแม่นยำสูง มีหลายปัญหาที่นำเทคนิคนี้มาประยุกต์ใช้ ได้แก่ ระบบตรวจจับป้ายแผ่นป้ายทะเบียนรถยนต์ด้วยกล้องวีดีโออัตโนมัติ และ ระบบตรวจจับมะเร็งลำไส้ใหญ่อัตโนมัติด้วยกล้องส่องผ่าตัด โดยตัวอย่างทั้งสองนี้สามารถลดการใช้แรงงานของมนุษย์ด้วยเทคนิคการรู้จำภาพถ่ายที่มีความแม่นยำสูงโดยการเสริมขีดความสามารถของระบบเหล่านี้ด้วยปัญญาประดิษฐ์ โดยเทคนิคการรู้จำภาพถ่ายที่มีประสิทธิภาพสูงสามารถนำมาพัฒนารถยนต์ขับเคลื่อนอัตโนมัติที่เหมาะสำหรับการใช้งานในประเทศไทยในปัจจุบันด้วยชุดข้อมูลภาพถ่ายบนท้องถนนที่อ้างอิงถึงสภาพแวดล้อมจริงของการขับขี่ในภูมิประเทศดังกล่าวมีความจำเป็นต่อการสร้างระบบขับเคลื่อนอัตโนมัติที่เหมาะสำหรับต่อการใช้งานในกรุงเทพมหานคร ซึ่งจะทำให้อุบัติเหตุจากระบบขับเคลื่อนอัตโนมัติในกรุงเทพมหานครลดลง

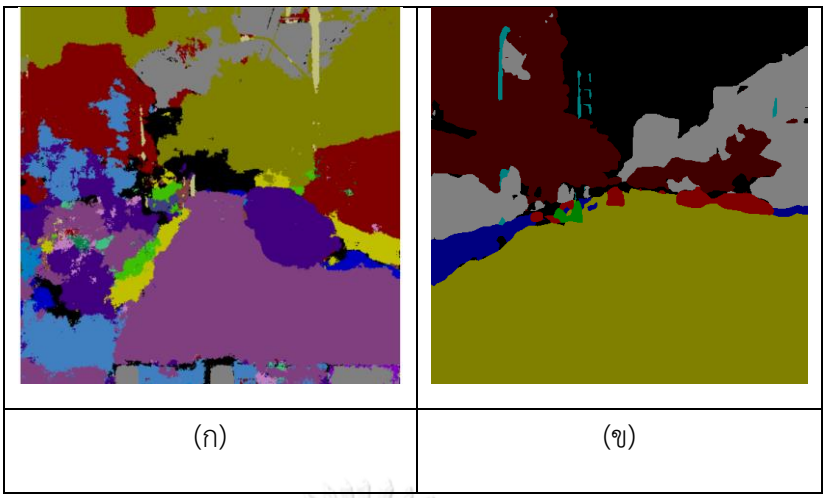
ในปัจจุบันมีงานวิจัยการแยกองค์ประกอบของยานพาหนะบนท้องถนนบนชุดข้อมูลกรุงเทพมหานครด้วยเทคนิคการรู้จำภาพถ่ายด้วยเทคนิคการเรียนรู้เชิงลึก (Transportation Mobility Factor Extraction Using Image Recognition) [1] โดย Pittipol K. และคณะในปี 2019 ได้นำเสนอวิธีรู้จำภาพถ่ายท้องถนน จำนวน 2 วิธี ได้แก่ 1. เทคนิคการตรวจจับประเภทของวัตถุบนท้องถนนด้วยแบบจำลองก่อนการเรียนรู้เชิงลึกโยโล่ เวอร์ชัน 3 (Object Detection by Using YOLO V3) [2] โดย J. Redmon และคณะในปี 2018 มาปรับใช้สำหรับการนับวัตถุแต่ละชนิดบนท้องถนน และ 2. เทคนิคการแบ่งส่วนภาพเชิงความหมายด้วยแบบจำลองก่อนการสอนของโมเดล (Pre-trained) 100 ชั้น ทิรามิสึ สำหรับการเชื่อมโยงเต็มรูปแบบของโครงข่ายคอนโวลูชันเดนซ์เน็ต (The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation) [3] โดย Simon J. ในปี 2017 นำมาปรับใช้สำหรับการคำนวณปริมาณเปอร์เซนต์ของพิกเซลของวัตถุแต่ละชนิดในภาพเพื่อนำมาสร้างแบบจำลองทำนายคุณภาพชีวิตของประชากรในจังหวัดกรุงเทพมหานคร

เทคนิคการรู้จำของชุดข้อมูลภาพถ่ายบนท้องถนนในกรุงเทพด้วยวิธีการเรียนรู้เชิงลึกในงานวิจัยข้างต้นนั้น ได้ทำการปรับไฮเปอร์พารามิเตอร์โดยอาศัยแบบจำลองก่อนการสอนโมเดลโดยที่

ไม่มีผลเฉลยจากชุดข้อมูล ผลลัพธ์จากเทคนิคการตรวจจับวัตถุให้ผลลัพธ์ที่ดีเยี่ยมสำหรับชุดข้อมูลนี้ แต่ผลลัพธ์จากเทคนิคการแบ่งส่วนภาพภาพเชิงความหมายโดยโมเดลก่อนการสอนทิรามิสีไม่สามารถทำนายรูปทรงของกลุ่มพิกเซลของวัตถุได้อย่างถูกต้องสำหรับชุดข้อมูลที่ไม่มีการทำผลเฉลย

งานวิทยานิพนธ์นี้ต้องการปรับปรุงประสิทธิภาพการทำนายพิกเซลรายคลาสของการแบ่งส่วนภาพถ่ายเชิงความหมายบนชุดข้อมูลท้องถนนกรุงเทพที่ไม่สามารถระบุรูปทรงของวัตถุในแต่ละคลาสได้อย่างชัดเจน ดังแสดงในรูปที่ 1(ก) ด้วยการทำผลเฉลยเป็นรูปภาพหลายเหลี่ยม (Polygon) ในชุดข้อมูลชุดภาพถ่ายกรุงเทพที่แสดงบริเวณพิกเซลของวัตถุในภาพถ่ายจากชุดข้อมูลนี้ จำนวน 701 ภาพ ที่มีคลาสของวัตถุทั้งหมด 11 คลาส ได้แก่ Road, Car, Tree, Pole, Footpath, Building, Motorcycle, Person, Trash, Crosswalk, และ Misc ประกอบกับการนำผลเฉลยมาฝึกด้วยสถาปัตยกรรมใหม่ที่ถูกนำเสนอ DeepLab-V3-A1 ด้วย Xception ดังแสดงในรูปที่ 1(ข) โดยมีคุณสมบัติการใช้ตัวกรองของชั้นคอนโวลูชันที่มีขนาดต่างกัน ที่สามารถไล่เก็บคุณลักษณะเด่นของรูปภาพได้ครบทุกระดับ ผนวกกับการทำฟีเจอร์คอนแคต อีกทั้งการนำ DeepLab-V3-A1 ด้วย Xception มาเปรียบเทียบประสิทธิภาพการทำนายกับวิธีการมาตรฐานด้วยสถาปัตยกรรมการแบ่งส่วนภาพเชิงความหมายทั้งหมด 6 รูปแบบได้แก่ SegNet [4], UNet [5], PSPNet [6], Tiramisu [3], DeepLab-V3+ [7] ด้วย ResNet-101 [8], และ DeepLab-V3+ ด้วย Xception [9] เพื่อเปรียบเทียบกับวิธีที่เรานำเสนอทั้ง 2 รูปแบบได้แก่ DeepLab-V3-A1 ด้วย ResNet-101 และ DeepLab-V3-A1 ด้วย Xception

อีกทั้งเราจะทำการทดลองเทคนิคใหม่ของเราทั้ง 2 วิธี เพื่อเปรียบเทียบกับวิธีมาตรฐานทั้งหมด 3 วิธี ได้แก่ Tiramisu [3], DeepLab-V3+ [7] ด้วย ResNet-101 [8], และ DeepLab-V3+ ด้วย Xception [9] บนชุดข้อมูลท้องถนนมาตรฐานสำหรับงานวิจัยสถาปัตยกรรมการแบ่งส่วนภาพถ่ายเชิงความหมายในปัจจุบัน ได้แก่ชุดข้อมูล CamVid [10] และ Cityscapes [11] ตามลำดับ



(ก) (ข)

รูปที่ 1 (ก) เปรียบเทียบการแบ่งส่วนภาพเชิงความหมายด้วย Pre-trained weight ที่ไม่มีผลเฉลยกำกับ (ข) ผลลัพธ์การแบ่งส่วนภาพเชิงความหมายด้วยวิธีการนำเสนอ DeepLab-V3-A1 ด้วย Xception และผลเฉลย

1.2 วัตถุประสงค์ของการวิจัย

เพื่อนำเสนอวิธีการปรับปรุงการประสิทธิภาพการแบ่งส่วนภาพเชิงความหมายบนชุดข้อมูลท้องถนนกรุงเทพมหานครที่เหมาะสม โดยในงานวิจัยนี้เรานำเสนอทั้งวิธีใหม่ DeepLab-V3-A1 ด้วย Xception ที่เรานำเสนอได้ ประกอบกับการสร้างผลเฉลยบนชุดข้อมูลนี้ เนื่องจากชุดข้อมูลภาพถ่ายท้องถนนในประเทศไทยนั้นยังไม่มีการทำชุดข้อมูลฝึกสำหรับภาพถ่ายบนท้องถนนมาก่อน ซึ่งแตกต่างจากต่างประเทศแถบยุโรปที่มีการวิจัยรถขับเคลื่อนอัตโนมัติอย่างกว้างขวาง ซึ่งชุดข้อมูลของเราสามารถเพิ่มขีดความสามารถของสถาปัตยกรรมวิธีมาตรฐาน (Baseline) ให้รู้จำลักษณะของวัตถุบนท้องถนนของกรุงเทพมหานครได้อย่างเหมาะสม อีกทั้งชุดข้อมูลของเราจะสามารถช่วยให้นักวิจัยระบบขับเคลื่อนอัตโนมัติ นำข้อมูลรูปภาพและผลเฉลยไปพัฒนาเทคนิคการแบ่งส่วนเชิงความหมายบนชุดข้อมูลบนท้องถนนเพื่อนำมาใช้กับระบบขับเคลื่อนอัตโนมัติบนถนนในกรุงเทพได้ในอนาคต ซึ่งจะส่งผลให้ระบบนี้สามารถลดแรงงานการขับขี่ของผู้ขับและช่วยลดอุบัติเหตุบนท้องถนนได้

1.3 ขอบเขตงานวิจัย

ในขั้นตอนแรกเราทำการสร้างผลเฉลยด้วยรูปทรงหลายเหลี่ยม (Polygon) จากชุดข้อมูลท้องถนนกรุงเทพมหานครด้วยโปรแกรม LabelMe [12] จำนวน 701 ภาพ ที่ประกอบด้วยวัตถุทั้งหมด 11 คลาส ได้แก่ Road, Car, Tree, Pole, Footpath, Building, Motorcycle, Person, Trash, Crosswalk, และ Misc เพื่อใช้ในการทดลองกับวิธีมาตรฐาน (Baseline) ทั้งหมด 6 วิธี ได้แก่

SegNet [4], UNet [5], PSPNet [6], Tiramisu [3] และ DeepLab-V3+ [7] ด้วย ResNet-101 [8], และ DeepLab-V3+ ด้วย Xception [9] และในขั้นตอนถัดไปเราจะฝึกวิธีที่เรานำเสนอ เพื่อเปรียบเทียบกับการฝึกวิธีมาตรฐาน 6 วิธีบนชุดข้อมูลของเรา เพื่อพิสูจน์ว่าวิธีของเรายังคงมีประสิทธิภาพไม่เฉพาะแค่บนชุดข้อมูลที่เรานำเสนอ เราจึงฝึกวิธีที่เรานำเสนอบนชุดข้อมูลท้องถนนอื่นๆ ที่เป็นมาตรฐานในงานวิจัยแบ่งส่วนเชิงความหมาย ได้แก่ CamVid [10] และ Cityscapes [11] ตามลำดับ เพื่อเปรียบเทียบประสิทธิภาพของวิธีการที่เรานำเสนอกับวิธีมาตรฐาน (Baseline) ได้แก่ Tiramisu [3], DeepLab-V3+ [7] ด้วย ResNet-101 [8], และ DeepLab-V3+ ด้วย Xception [9]

1.4 วิธีดำเนินงานวิจัย

1. ทบทวนวรรณกรรมและงานวิจัยที่เกี่ยวข้องกับหัวข้อวิจัย เช่น การเรียนรู้เชิงลึก, โครงข่ายคอนโวลูชัน, โปรแกรมการทำผลเฉลย, และเทคนิคการประมวลผลภาพดิจิตอล
2. นำชุดข้อมูลมาสร้างผลเฉลย
3. ค้นคว้าวิธีการแปลงผลเฉลยให้เป็นเวกเตอร์
4. เขียนโค้ดเพื่อทดสอบสมมติฐานกับชุดข้อมูล จากงานวิจัยในขั้นตอนทบทวนวรรณกรรม
5. ดำเนินการทดลองเพื่อหาความเป็นไปได้ สำหรับงานวิจัยด้วยชุดข้อมูลนี้
6. เขียนโครงร่างและตีพิมพ์ผลงานในงานประชุมเชิงวิชาการ
7. จัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

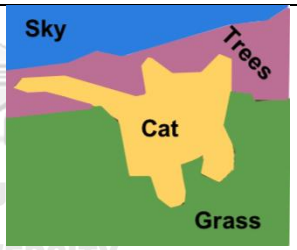
1. นักวิจัยด้านวิศวกรรมการขนส่งสามารถนำผลลัพธ์การทำนายชุดข้อมูลภาพถ่ายบนท้องถนนในกรุงเทพมหานคร ในงานวิทยานิพนธ์ชิ้นนี้นำไปต่อยอดและตีความในเชิงลึกได้ในด้านการสร้างแบบจำลองทำนายคุณภาพชีวิตของประชากรในกรุงเทพมหานคร
2. นักวิจัยการแบ่งส่วนภาพเชิงความหมายสามารถนำชุดข้อมูลภาพถ่ายบนท้องถนนในกรุงเทพมหานครเป็นชุดข้อมูลมาตรฐานสำหรับวัดประสิทธิภาพการทำนายของสถาปัตยกรรมการเรียนรู้เชิงลึกได้

3. นักวิจัยการแบ่งส่วนภาพเชิงความหมายสามารถนำสถาปัตยกรรมใหม่ DeepLab-V3-A1 ด้วย Xception ไปปรับใช้กับชุดข้อมูลอื่นเพื่อเป็นสถาปัตยกรรมวิธีมาตรฐาน (Baseline) ได้
4. นักวิจัยรถขับเคลื่อนอัตโนมัติในประเทศไทยสามารถนำองค์ความรู้ (Pre-train weight) จากสถาปัตยกรรม DeepLab-V3-A1 ด้วย Xception ที่ถูกฝึกบนชุดข้อมูลโดยอ้างอิงภาพจากแวดล้อมถนนในกรุงเทพมหานครที่แท้จริงมาประยุกต์ใช้กับระบบควบคุมอัตโนมัติได้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 การแบ่งส่วนภาพเชิงความหมาย

เป็นการสอนคอมพิวเตอร์ให้จำแนกประเภทของวัตถุในภาพทุกพิกเซล โดยกำหนดให้ข้อมูลฝึกที่ประกอบไปด้วยภาพถ่ายและผลเฉลยที่ถูกกำหนดขอบเขตของวัตถุที่เราต้องการให้โมเดลเรียนรู้ หลังจากขั้นตอนการฝึก โมเดลที่ถูกฝึกจะสามารถทำนายวัตถุจากชุดข้อมูลทดสอบได้ทุกพิกเซล โดยความแตกต่างระหว่างการจำแนกประเภทของภาพ กับ การแบ่งส่วนภาพเชิงความหมายดังแสดงในรูปที่ 2

Input		
Output	แมว	
วิธีการ	การคัดแยกประเภทของภาพ (Image Classification)	การแบ่งส่วนภาพเชิงความหมาย (Semantic Image Segmentation)

รูปที่ 2 ความแตกต่างระหว่างการแบ่งส่วนภาพกับการแบ่งส่วนภาพเชิงความหมาย[1]

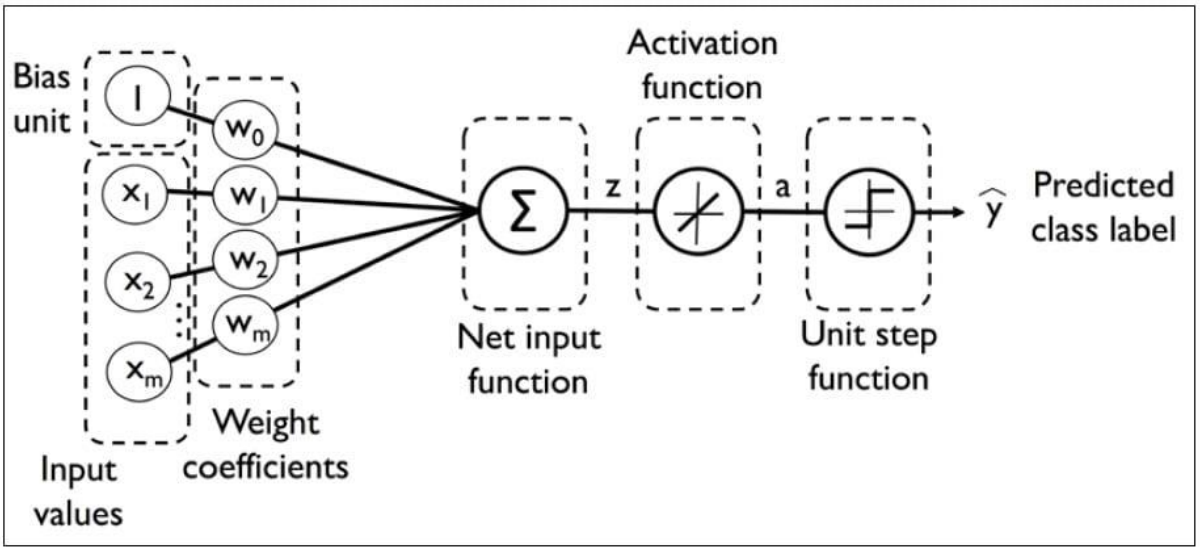
1 อ้างอิงจาก https://www.simplilearn.com/multilayer-artificial-neural-network-tutorial

2.2 โครงข่ายประสาทเทียม (Artificial Neuron Networks)

ในปัจจุบันงานวิจัยด้านการมองเห็นทางคอมพิวเตอร์อาศัยสถาปัตยกรรมที่ประกอบด้วยโครงข่ายประสาทเทียมเป็นองค์ประกอบหลักในการแบ่งส่วนภาพเชิงความหมาย โดยโครงข่ายประสาทเทียมได้รับแรงบันดาลใจมาจากการทำงานของเซลล์ประสาทในสมองของมนุษย์

การทำงานของแบบจำลองนี้เกิดจากการนำข้อมูลฝึกมาทำการคำนวณผลรวมดอทโปรดักท์ (Dot product) กับค่าน้ำหนักที่ถูกการสุ่มตอนต้น และผลลัพธ์การทำนายของโมเดลนี้ ถูกนำผลลัพธ์การคำนวณดอทโปรดักท์ข้างต้นมาผ่านฟังก์ชันกระตุ้น (Activation Function) ที่มีคุณสมบัติเป็นฟังก์ชันไม่เชิงเส้น (Non-Linearity Function) หลังจากโมเดลจะเรียนรู้ด้วยการปรับค่าน้ำหนักของแบบจำลองด้วยการแพร่ย้อนกลับ (Backpropagation) เมื่อเทียบกับผลเฉลยของชุดข้อมูลฝึก

โครงสร้างของโครงข่ายประสาทเทียมโดยรวมดังแสดงในรูปที่ 3



รูปที่ 3 ภาพรวมของการทำงานของโครงข่ายประสาทเทียม[2]

2 อ้างอิงจาก https://www.simplilearn.com/multilayer-artificial-neural-network-tutorial

หลักการทำนายของแบบจำลองโครงข่ายประสาทเทียมถูกอธิบายด้วยสมการเหล่านี้

$$\hat{y} = f(x) = \begin{cases} 1 & if \sum_{i=1}^{m} w_i x_i + b > 0 \\ 0 & otherwise \end{cases} \quad (1)$$

กำหนดให้ w_i คือค่าน้ำหนัก (weight), b คือค่าไบแอส (Bias), m คือ ขนาดของชุดข้อมูลฝึก โดยกระบวนการปรับปรุงการเรียนรู้ของโมเดลโครงข่ายประสาทเทียมถูกอธิบายด้วยสมการที่ 1 – 3

$$w_i \leftarrow w_i + \Delta w_i \quad (2)$$

โดยที่ $\Delta w_i = \eta(\hat{y} - y)x_i$ (3)

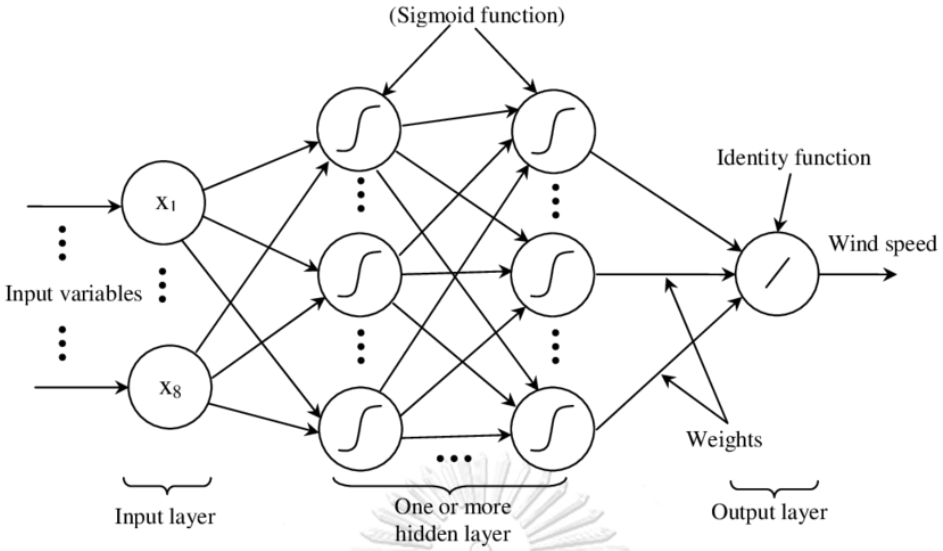
พารามิเตอร์ η แสดงถึงค่าอัตราการเรียนรู้ (Learning rate) ที่ถูกกำหนดอัตราการเรียนรู้เพื่อใช้ปรับค่าน้ำหนักให้ผลลัพธ์การเรียนรู้ได้ค่าน้ำหนักที่ใช้ทำนายกับงานได้เหมาะสมกับงาน หากค่าอัตราการเรียนรู้มากเกินไปจะทำให้จำนวนรอบการฝึกน้อยแต่ความแม่นยำต่ำมาก ถ้าหากกำหนดอัตราการเรียนรู้ต่ำเกินไปจะทำให้กระบวนการเรียนรู้การปรับน้ำหนักที่เหมาะสมนั้นต้องใช้จำนวนรอบในการฝึกมากเกินไป

ผลลัพธ์การคำนวณโครงข่ายประสาทเทียมดังแสดงในรูปที่ 4 ถูกอธิบายด้วยสมการการคำนวณโครงข่ายประสาทเทียมของแต่ละชั้นด้วยสมการที่ 4 และ 5

กำหนดตัวแปรแทนการคำนวณไปข้างหน้า (Feed forward) ได้ผลลัพธ์การคำนวณเพอร์เซปตรอนตัวที่ k คือ a_k^{d-1}มาจากการคำนวณเพอร์เซปตอนจากเลเยอร์ก่อนหน้า และ w_{jk} หมายถึงน้ำหนักของเพอร์เซปตรอนตัวที่ j ในชั้นที่ d โดย g คือฟังก์ชันกระตุ้น และกำหนดให้ n เป็นจำนวนของเพอร์เซปตรอนในเลเยอร์ที่ $d-1$ และ b_j^l คือค่าไบแอสในเลเยอร์ที่ d

$$z_j^d = \sum_{k=1}^{n} w_{jk}^d a_k^{d-1} + b_j^d \quad (4)$$

$$a_k^d = g(z_j^d) \quad (5)$$



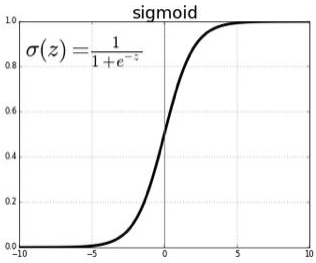
รูปที่ 4 โครงสร้างของโครงข่ายประสาทเทียมแบบป้อนไปข้างหน้า[3]

2.2.1 ฟังก์ชันกระตุ้น (Activation Function)

ทำหน้าที่นำผลลัพธ์หลังจากการทำดอทโปรดักท์ของเพอร์เซปตรอน ถูกนำไปแทนค่าในฟังก์ชันที่ไม่เป็นเส้นตรง (Non-Linear Function) เพื่อทำให้พื้นที่การตัดสินใจของโครงข่ายประสาทเทียมมีความซับซ้อนมากขึ้น เพื่อการแก้ไขปัญหาการแบ่งส่วนภาพที่มีความซับซ้อนสูง

- ฟังก์ชันซิกมอย (Sigmoid function)

เป็นฟังก์ชันที่ถูกดัดแปลงมาจากสมการการถดถอยแบบลอจิสติก โดยให้ผลลัพธ์การทำนายเพียงแค่ค่า 0 หรือ 1 เท่านั้น โดยกำหนดให้สมการและกราฟการคำนวณฟังก์ชันซิกมอยดังแสดงในรูปที่ 5 และหลักเกณฑ์การตัดสินใจของฟังก์ชันซิกมอยดังแสดงในสมการที่ 6



รูปที่ 5 กราฟของฟังก์ชันซิกมอย[4]

3 อ้างอิงจาก https://medium.com/engineer-quant/multilayer-perceptron-4453615c4337

4 อ้างอิงจาก https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7

หลักเกณฑ์การตัดสินใจ

$$f(z) = \begin{cases} 1 & if\ \sigma(z) > 0.5 \\ 0 & otherwise \end{cases} \quad (6)$$

- ฟังก์ชันค่าสูงสุดอย่างอ่อน (Softmax Function)

เป็นฟังก์ชันสำหรับทำนายความน่าจะเป็นด้วยการปรับผลลัพธ์ให้เรียบด้วยฟังก์ชันเอกโปเนนซ์เชียลจากผลลัพธ์การคำนวณด้วยโครงข่ายประสาทเทียมชั้นก่อนหน้า โดยผลลัพธ์การคำนวณ k ตัว ถูกกำหนดด้วย z และผลลัพธ์การคำนวณฟังก์ชันค่าสูงสุดอย่างอ่อนตัวที่ j ด้วยฟังก์ชัน f_i ดังแสดงในสมการที่ 7

$$f(z)_j = \frac{e^{z_j}}{\sum_{i=1}^{k} e^{z_i}} \quad (7)$$

- หน่วยเชิงเส้นเรคติไฟต์ (Rectified Linear Unit)

ในปัจจุบันฟังก์ชันไม่เชิงเส้นชนิดนี้เป็นที่นิยมสำหรับการปรับสัญญาณของผลลัพธ์คำนวณเพอร์เซปตรอนจากชั้นก่อนหน้าให้ชัดขึ้น เราแทนฟังก์ชันนี้ด้วย f ดังแสดงในสมการที่ 8

$$f(z) = \begin{cases} 0 & if\ z < 0 \\ z & if\ z \geq 0 \end{cases} \quad (8)$$

2.2.2 ฟังก์ชันต้นทุน (Cost Function)

เป็นฟังก์ชันที่ใช้วัดประสิทธิภาพผลการทำนายจากชุดข้อมูลด้วยโมเดล หากค่าของฟังก์ชันต้นทุนมีค่ามากในระหว่างการสอนโมเดลสามารถตีความได้ว่าประสิทธิภาพของโมเดลในการทำนายต่ำมาก หากค่าของฟังก์ชันต้นทุนระหว่างการสอนโมเดลมีค่าน้อยสามารถตีความได้ว่าประสิทธิภาพการทำนายผลของโมเดลนั้นมีประสิทธิภาพ

กำหนดให้ J แทนฟังก์ชันต้นทุนที่ใช้วัดชุดข้อมูลจำนวน n ตัวในการฝึกโมเดล y_i คือผลเฉลยจากชุดข้อมูลฝึก และ $\hat{y}_i$ คือผลลัพธ์ของการทำนายด้วยโมเดลจากชุดข้อมูล โดยในงานวิทยานิพนธ์นี้จะสนใจที่ค่าเฉลี่ยเอนโทปีไขว้ดังแสดงในสมการที่ 9

- ค่าเฉลี่ยเอนโทปีไขว้ (Binary Cross-entropy)

$$J = \frac{-1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{9}$$

ฟังก์ชันต้นทุนชนิดนี้เหมาะสำหรับการการแบ่งส่วนภาพภาพเชิงความหมาย ที่สามารถระบุทุกพิกเซลในภาพถ่ายว่าใช้ประเภทวัตถุที่เราสนใจใช่หรือไม่

2.2.3 ฟังก์ชันปรับเหมาะ (Optimization Function)

ฟังก์ชันที่ใช้ในการปรับน้ำหนักเพื่อการฝึกโมเดลการเรียนรู้เชิงลึกจากชุดข้อมูลฝึกเพื่อให้ค่าฟังก์ชันต้นทุนมีค่าต่ำที่สุด โดยงานวิทยานิพนธ์นี้ให้ความสนใจทีฟังก์ชัน Root Mean Squared Propagation [13] (RMSProp) โดยฟังก์ชันปรับเหมาะด้วย RMSProp มีคุณสมบัติในการทำให้ค่าของฟังก์ชันต้นทุนเข้าใกล้ค่าต่ำสุดที่เหมาะสม (Local optima) ด้วยการกำหนดฟังก์ชันความเร็วด้วยหลักการของ Exponential Moving Average (EMA) ยกกำลังสองดังแสดงในสมการที่ 10 – 13

โดยกำหนดให้สัญลักษณ์ดังนี้

$W^{[l]}$	คือค่าน้ำหนักของโมเดลที่เราต้องปรับค่าในแต่ละรอบของการเรียนรู้ ณ ชั้นที่ l
$\mathrm{b}^{[l]}$	คือค่าไบแอสของโมเดลที่เราต้องการปรับค่าในแต่ละรอบการเรียนรู้ ณ ชั้นที่ l
η	คืออัตราการเรียนรู้ (Learning rate)
$\mathrm{dW}^{[l]}$	คืออนุพันธ์ย่อยของฟังก์ชันต้นทุนเมื่อเทียบกับตัวแปรปรับน้ำหนักที่แสดงด้วย $\frac{\partial J}{\partial w}$ ณ ชั้นที่ l
$\mathrm{db}^{[l]}$	คืออนุพันธ์ย่อยของฟังก์ชันต้นทุนเมื่อเทียบกับตัวแปรไบแอสแสดงด้วย $\frac{\partial J}{\partial b}$ ณ ชั้นที่ l
ε	คือค่าคงที่ขนาดเล็กเพื่อป้องกันไม่ให้ตัวหารเป็น 0
β	คือค่าน้ำหนักของ EMA โดยกำหนดให้เป็นค่าเริ่มต้นซึ่งมีค่าเท่ากับ 0.9

กำหนดสมการกระบวนการปรับปรุงกระบวนการเรียนรู้ฟังก์ชันปรับเหมาะด้วย RMSProp ดังนี้

ขั้นตอนที่ 1 การคำนวณเวกเตอร์ของน้ำหนักเร่ง S โดยการใช้ EMA กำลังสองของอนุพันธ์ ดังแสดงในสมการที่ 10 และ 11

$$S_{\mathrm{dW}^{[l]}} = \eta\beta S_{\mathrm{dW}^{[l]}} + (1-\beta)\mathrm{dW}^{[l]^2} \tag{10}$$

$$S_{\mathrm{db}^{[l]}} = \eta\beta S_{\mathrm{db}^{[l]}} + (1-\beta)\mathrm{db}^{[l]^2} \tag{11}$$

ขั้นตอนที่ 2 การนำค่าน้ำหนักเร่ง $S_{\mathrm{dW}^{[l]}}$ และ $S_{\mathrm{db}^{[l]}}$ กลับไปอัพเดทค่าน้ำหนัก W และตัวแปรไบแอส b ของโครงข่ายประสาทเทียมในแต่ละชั้นเลเยอร์ที่ l ดังแสดงในสมการที่ 12 และ 13

$$W^{[l]} := W^{[l]} - \eta\left(\frac{\mathrm{dW}^{[l]}}{\sqrt{S_{\mathrm{dW}^{[l]}}} + \varepsilon}\right) \tag{12}$$

$$\mathrm{b}^{[l]} := \mathrm{b}^{[l]} - \eta\left(\frac{\mathrm{db}^{[l]}}{\sqrt{S_{\mathrm{db}^{[l]}}} + \varepsilon}\right) \tag{13}$$

2.2.4 การแพร่กระจายย้อนกลับ (Backpropagation)

เป็นขั้นตอนการปรับปรุงน้ำหนักในทุกโนดการคำนวนจากการเรียนรู้เชิงลึกอาศัยการคำนวณย้อนกลับของเพอร์เซปตรอนด้วยผลเฉลยของชุดข้อมูลฝึกในข้างต้น หลังการแทนค่าด้วยด้วยฟังก์ชันกระตุ้น โดยอาศัยมาตรวัดผลการทำนายของแบบจำลองนี้ด้วยฟังก์ชันต้นทุน และสมการการแพร่กระจายย้อนกลับจะสามารถอธิบายกระบวนการปรับปรุงการเรียนรู้สำหรับทุกโนดของโครงข่ายประสาทเทียมดังแสดงในสมการที่ 14 – 18 โดยกำหนดสัญลักษณ์ดังนี้

δ_j^l คือ ความคาดเคลื่อนของผลลัพธ์การคำนวณด้วยเพอร์เซปตรอนตัวที่ j ณ เลเยอร์ที่ l

J คือ ฟังก์ชันต้นทุน

z คือ ผลลัพธ์จากการคำนวณของเพอร์เซปตรอนก่อนหน้าที่จะถูกนำไปแทนค่าด้วยฟังก์ชันกระตุ้น

$$\delta_j^l = \frac{\partial J}{\partial z_j^l} = \frac{\partial J}{\partial a_j^l}\frac{\partial a_j^l}{\partial z_j^l} = \frac{\partial J}{\partial a_j^l} g'(z_j^l) \tag{14}$$

โดยการคำนวณ $\frac{\partial J}{\partial a_j^l}$ จากชั้นสุดท้ายจะถูกนำไปปรับค่าด้วยการคำนวณฟังก์ชันต้นทุนจากผลเฉลย สำหรับโนดจากการคำนวณเพอร์เซปตรอนของชั้นก่อนหน้า จะถูกคำนวณในลักษณะเดียวกันกับการคำนวณป้อนไปทางข้างหน้า

$$\frac{\partial J}{\partial a_j^l} = \sum_{k=1}^{m} \frac{\partial J}{\partial z_k^{l+1}}\frac{\partial z_k^{l+1}}{\partial a_j^l} = \sum_{k=1}^{m} \delta_k^{l+1} w_{kj}^{l+1} \tag{15}$$

กำหนดให้ m คือจำนวนโนดของเพอร์เซปตรอนในเลเยอร์ที่ $l+1$ และค่าความคลาดเคลื่อนของในแต่ละเลเยอร์จะถูกคำนวณย้อนกลับด้วยน้ำหนักและไบแอสดังแสดงในสมการที่ 16 และ 17

$$\delta_j^l a_k^{l-1} = \frac{\partial J}{\partial w_{jk}^l} = \frac{\partial J}{\partial z_j^l}\frac{\partial z_j^l}{\partial w_{jk}^l} \tag{16}$$

$$\delta_j^l = \frac{\partial J}{\partial a_j^l} = \frac{\partial J}{\partial z_j^l}\frac{\partial z_j^l}{\partial b_j^l} \tag{17}$$

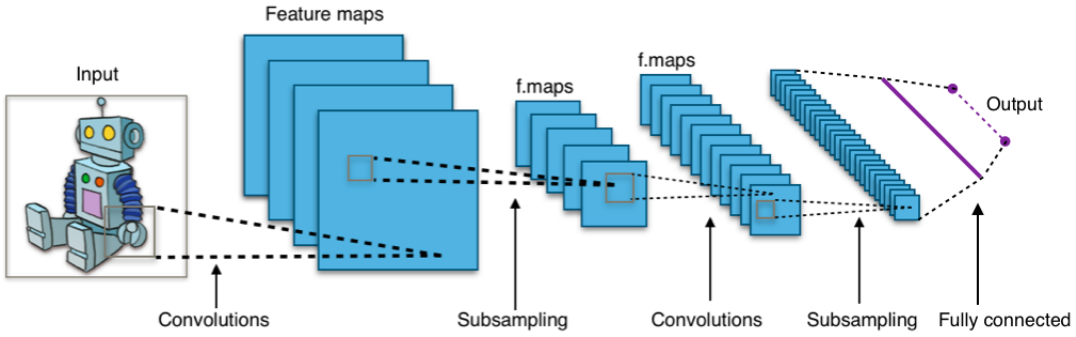
และเมื่อนำฟังก์ชันปรับค่าให้เหมาะสมคือ **RMSProp** มาประกอบการประเมินผลลัพธ์การปรับค่าน้ำหนักของเพอร์เซปตรอน w_{jk}^l จะได้สมการการปรับค่าน้ำหนักด้วยฟังก์ชันการปรับค่าให้เหมาะสมดังแสดงในสมการที่ 18

$$w_{jk,t}^l = w_{jk,t-1}^l + \eta a_{k,t}^{l-1} \delta_{j,t}^l \tag{18}$$

2.3 โครงข่ายประสาทคอนโวลูชัน (Convolutional Neuron Networks)

การเรียนรู้เชิงลึกด้วยโครงข่ายประสาทคอนโวลูชัน [14] จำลองมาจากการมองเห็นจากสัตว์เลี้ยงลูกด้วยนม โดยโครงข่ายประสาทเทียมแบบคอนโวลูชันประกอบด้วยน้ำหนักและค่าไบแอสเช่นเดียวกันกับโครงข่ายประสาทเทียม โดยแต่ละโนดจะถูกป้อนด้วยอินพุตที่มีการทำดอทโปรดักท์และถูกนำไปผ่านฟังก์ชันกระตุ้น ในลำดับถัดมาผลลัพธ์จากการทำคอนโวลูชัน จะถูกการลดขนาดมิติด้วยการทำ Sub-sampling เพื่อลดระยะเวลาการคำนวณ และส่งต่อไปยังชั้นที่เชื่อมโยงเต็มรูปแบบ

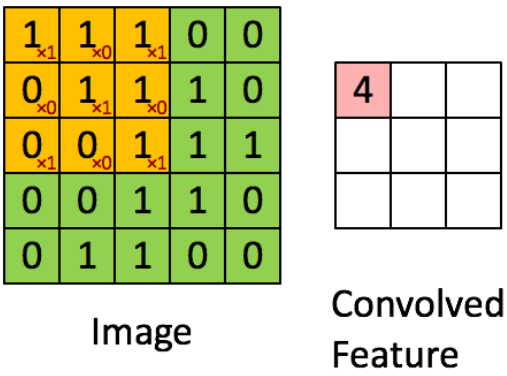
(Fully Connected Layer) เพื่อใช้ในการคัดแยกชนิดของรูปภาพ โดยภาพรวมของโครงข่ายประสาทเทียมในลักษณะนี้จะถูกแสดงดังรูปที่ 6



รูปที่ 6 ภาพรวมของโครงข่ายคอนโวลูชัน[5]

2.3.1 ชั้นคอนโวลูชัน (Convolutional Layer)

เป็นการหาความสัมพันธ์เชิงพื้นที่ของเวกเตอร์รูปภาพ ภาพนำเข้าซึ่งแสดงดังรูปที่ 7 ช่องสีเขียว และผลลัพธ์การทำดอทโปรดักท์เมทริกซ์ ซึ่งแสดงดังรูปที่ 7 ช่องสีชมพู ด้วยตัวกรอง (Filter) ซึ่งแสดงดังรูปที่ 7 ช่องสีเหลือง ชั้นคอนโวลูชันสามารถเรียนรู้คุณลักษณะเด่นของรูปภาพด้วยน้ำหนักเมทริกซ์ของตัวกรอง โดยน้ำหนักของตัวกรองจะถูกใช้ในการคำนวณกับทุกสมาชิกของเมทริกซ์รูปภาพดังแสดงในรูปที่ 7



รูปที่ 7 ขั้นตอนการทำฟีเจอร์แมพ[6]

5 อ้างอิงจาก https://en.wikipedia.org/wiki/Convolutional_neural_network

ขั้นตอนการทำคอนโวลูชันถูกอธิบายในสมการที่ 19 และ 20

โดยกำหนดสัญลักษณ์ดังนี้

z_{ij}^{l} คือผลลัพธ์ของการคำนวณฟีเจอร์แมพของชั้นคอนโวลูชัน ณ ชั้นที่ l

$w_{a,b}^{l}$ คือน้ำหนักของเมทริกซ์ตัวกรองที่มีจำนวนมิติเท่ากับ $m * m$

b^{l} คือค่าไบแอส ณ ชั้นที่ l

$A_{i+a,j+b}^{l-1}$ คือฟีเจอร์แมพของการคำนวณคอนโวลูชันจากชั้นก่อนหน้าที่ถูกผ่านด้วยฟังก์ชันไม่เชิงเส้น

g คือฟังก์ชันไม่เชิงเส้น

A_{ij}^{l} คือผลลัพธ์การคำนวณฟีเจอร์แมพจากชั้นที่ l หลังผ่านฟังก์ชันไม่เชิงเส้น โดยมีจำนวนมิติเท่ากับ $N * N$ ถูกการคำนวณด้วยการดอทโปรดักท์จากเมทริกซ์ตัวกรอง

$$z_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{a,b}^{l} A_{i+a,j+b}^{l-1} + b^{l} \tag{19}$$

$$A_{ij}^{l} = g(z_{ij}^{l}) \tag{20}$$

2.3.2 ขนาดของตัวกรอง (Size of Filter)

ขนาดของตัวกรองเรียกได้อีกชื่อว่ามิติที่ถูกกำหนดเพื่อใช้ในการคำนวณคอนโวลูชันเมทริกซ์ งานวิจัยส่วนใหญ่นิยมมิติของตัวกรองเมทริกซ์จัตุรัส

2.3.3 ชนิดการทำคอนโวลูชัน (Type of Convolution)

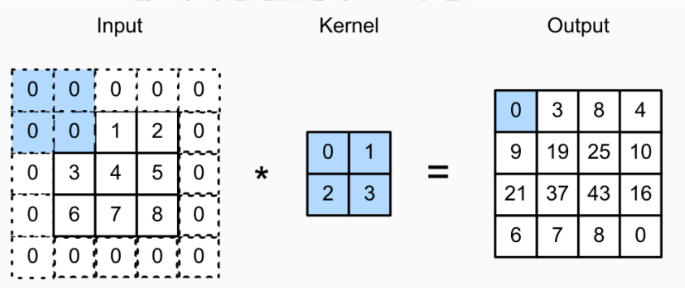
- การทำคอนโวลูชันแบบแคบ (Narrow Convolution) คือการคำนวณคอนโวลูชันแบบทั่วไปโดยที่ไม่เพิ่มขนาดของมิติของอินพุตเมทริกซ์ และผลลัพธ์การทำคอนโวลูชันแบบนี้จะให้ขนาดของลักษณะเด่นที่ถูกสกัดจากค่าน้ำหนักในคอนโวลูชันเมทริกซ์เรียกว่าฟีเจอร์แมพ (Feature Map) กำหนดให้ H x W คือจำนวนมิติของภาพ

---

6 อ้างอิงจาก https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

นำเข้า และกำหนดให้เมทริกซ์ตัวกรองที่เป็นเมทริกซ์จัตุรัส M x M ดังนั้นจำนวนมิติของเอาท์พุตที่เป็นฟีเจอร์แมพมีค่าเท่ากับ (H-M+1) x (W-M+1)

- การทำคอนโวลูชันแบบกว้าง (Wide Convolution) คือการทำคอนโวลูชันที่มีการเพิ่มขนาดเมตริกโดยนำขอบที่มีค่าเท่ากับ 0 มาเพิ่มรอบบริเวณของอินพุตเมทริกซ์ โดยการเพิ่มมิติในลักษณะนี้ถูกเรียกว่าการเพิ่มแพดดิง (Padding) โดยการทำคอนโวลูชันแบบกว้างจะมีช่วยให้ฟีเจอร์แมพยังคงมีข้อมูลส่วนของความเข้มของขอบรูปภาพปรากฏ โดยกำหนดให้ H x W คือจำนวนมิติของภาพนำเข้า และกำหนดให้เมทริกซ์ตัวกรองที่เป็นเมทริกซ์จัตุรัส M x M และกำหนดให้จำนวนแถวของการเพิ่มแพดดิงรอบข้อมูลนำเข้าด้วย P ดังนั้นจำนวนมิติของฟีเจอร์แมพของการทำคอนโวลูชันแบบกว้างเท่ากับ (H-M+P+1) x (W-M-P+1) ดังแสดงในรูปที่ 8

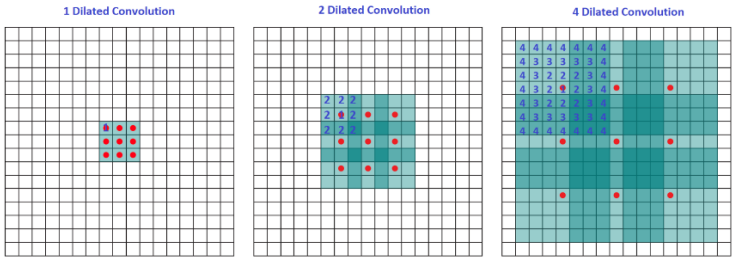


รูปที่ 8 การทำคอนโวลูชันแบบกว้าง[7]

- การทำคอนโวลูชันแบบหลุม (Atrous Convolution) คือการทำคอนโวลูชันลักษณะนี้มีต้นกำเนิดจาก Semantic Image Segmentation With Deep Con-Volutional Nets And Fully Connected Crfs (DeepLab-V1) [15] เป็นการปรับความกว้างของน้ำหนักของตัวกรองให้สามารถนำอินพุตที่อยู่ไกล มาพิจารณาการคำนวณฟีเจอร์แมพได้ เปรียบเทียบ ภาพที่ 9(ก) คือลักษณะการทำนำตัวกรองที่ไม่ได้กำหนดความกว้างมาคำนวณคอนโวลูชัน และ ภาพที่ 9(ข) คือการทำคอน

7 อ้างอิงจาก https://d2l.ai/chapter_convolutional-neural-networks/padding-and-strides.html

โวลูชันแบบหลุมด้วยอัตราความกว้างเท่ากับ 2 และ ภาพที่ 9(ค) คือการทำคอนโวลูชันแบบหลุมด้วยอัตราความกว้างเท่ากับ 4 ตามลำดับ

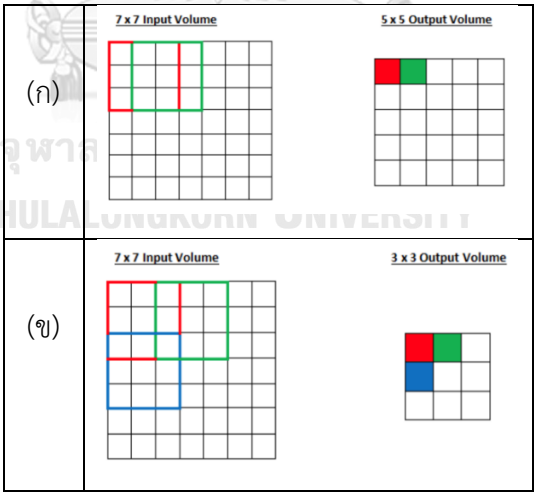


(ก) (ข) (ค)

รูปที่ 9 การทำคอนโวลูชันแบบหลุม[8]

- การทำคอนโวลูชันแบบก้าวกระโดด (Stride Convolution)

คือการลดมิติของฟีเจอร์แมพด้วยการทำให้ตัวกรองกระโดดไกลมากกว่า 1 ก้าว เมื่อพิจารณารูปที่ 10(ก) การทำคอนโวลูชันแบบก้าวกระโดด 1 ช่อง เปรียบเทียบกับรูปที่ 10(ข) การทำคอนโวลูชันที่มีอัตราการก้าวกระโดดเท่ากับ 2 ช่อง จะทำให้มิติของฟีเจอร์แมพลดลง



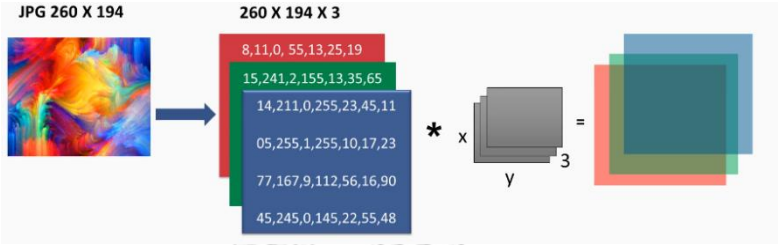
รูปที่ 10 (ก) คอนโวลูชันแบบก้าวกระโดด 1 ช่อง (ข) คอนโวลูชันแบบก้าวกระโดด 2 ช่อง[9]

8 อ้างอิงจาก https://towardsdatascience.com/understanding-2d-dilated-convolution-operation-with-examples-in-numpy-and-tensorflow-with-d376b3972b25

9 อ้างอิงจาก https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/

- จำนวนช่องสัญญาณ (Channels)

จำนวนช่องสัญญาณเกิดจากการผลลัพธ์ของการทำคอนโวลูชันตามจำนวนขนาดของตัวกรอง ยกตัวอย่างรูปภาพที่เกิดจากการผสมกันระหว่างช่องสี R G B ที่ประกอบด้วยเมทริกซ์ 3 ชั้นถูกซ้อนกัน หรือสามารถเรียกได้อีกแบบว่าความลึกของข้อมูลรับเข้าดังแสดงในรูปที่ 11



รูปที่ 11 ช่องสัญญาณของ R G B[10]

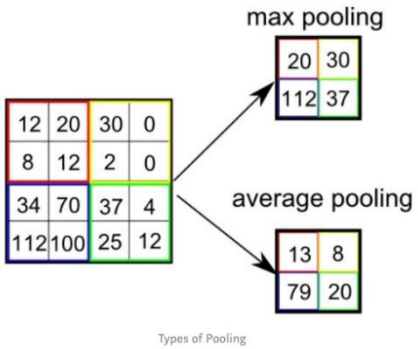
2.3.4 ชั้นพูลลิ่ง (Pooling Layers)

เป็นการลดสัญญาณรบกวนของข้อมูลด้วยการลดมิติของข้อมูลด้วยเมทริกซ์พูลลิงจากการคัดเลือกตัวแทนลักษณะเด่นจากฟีเจอร์แมพ (Feature map) โดยการทำพูลลิงมีอยู่ 2 ลักษณะหลัก ได้แก่ การทำพูลลิงโดยเลือกสมาชิกสูงสุด (Max pooling) และการทำพูลลิงโดยการเฉลี่ยสมาชิกในเมทริกซ์ (Average pooling)

1. การทำชั้นพูลลิงโดยการเลือกสมาชิกสูงสุด โดยการใช้พูลลิงเมทริกซ์ที่มีขนาด 2 x 2 เพื่อใช้เลือกสมาชิกจากฟีเจอร์แมพเพียง 1 ตัวที่มีค่ามากที่สุด $Max(12, 20, 8, 12) = 20$ ดังแสดงในรูปที่ 12 ที่ลูกศรด้านบน

2. การทำชั้นพูลลิงโดยการเฉลี่ยสมาชิกในพูลลิงเมทริกซ์ โดยการใช้พูลลิงเมทริกซ์ที่มีขนาด 2 x 2 เพื่อหาค่าเฉลี่ยจากสมาชิกจากฟีเจอร์แมพ โดยมีสมการดังนี้ $Avg(12, 20, 8, 12) = 13$ ดังแสดงในรูปที่ 12 ที่ลูกศรด้านล่าง

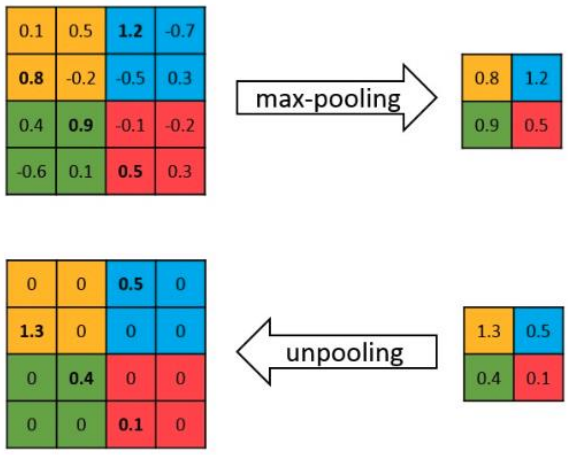
10 อ้างอิงจาก https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781789613964/2/ch02lvl1sec21/convolution-on-rgb-images



รูปที่ 12 ขั้นตอนการทำ Max pooling และ Average pooling[11]

2.3.5 ชั้นอันพูลลิง (Unpooling Layer)

เป็นการแปลงข้อมูลที่ผ่านการทำพูลลิงเลเยอร์ให้กลายเป็นข้อมูลนำเข้าก่อนการทำชั้นพูลลิงให้ได้มากที่สุด โดยการทำแพดดิ้งรอบเมทริกซ์ผลลัพธ์ดังแสดงในรูปที่ 13



รูปที่ 13 เปรียบเทียบระหว่าง Max-pooling และ Unpooling[12]

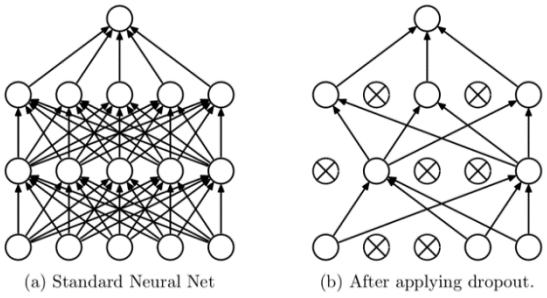
11 อ้างอิงจาก https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

12 อ้างอิงจาก https://www.researchgate.net/figure/Pooling-and-unpooling-layers-For-each-pooling-layer-the-max-locations-are-stored-These_fig2_306081538

2.3.6 ชั้นการเชื่อมโยงแบบเต็มรูปแบบ (Fully Connected Layer)

เป็นการนำผลลัพธ์ฟีเจอร์แมพหลังการทำคอนโวลูชันเลเยอร์ และการผ่านการเลือกสมาชิกจากชั้นพูลลิงแล้ว เมทริกซ์ที่ผ่านการทำพูลลิง จะถูกนำไปคำนวณในส่วนชั้นการเชื่อมโยงเต็มรูปแบบด้วยการคำนวณโครงข่ายประสาทเทียม (Artificial Neuron Networks) จากการเชื่อมโยงผลลัพธ์การคำนวณจากการทำคอนโวลูชันก่อนหน้าด้วยน้ำหนักและไบแอส และถูกคำนวณด้วยการคำนวณด้วยการป้อนไปทางข้างหน้า ประกอบกับการปรับปรุงการเรียนรู้ด้วยการแพร่กระจายย้อนกลับ ด้วยเทคนิคการปรับปรุงกระบวนการเรียนรู้ให้มีความเหมาะสมกับชุดข้อมูลด้วยการเพิ่มดรอปเอาท์ (Dropout) [16] เพื่อป้องกันการปรับเหมาะเกินไป (Overfitting) สำหรับชุดข้อมูลฝึก

ขั้นตอนการเรียนรู้ เส้นเชื่อมของขั้นตอนการเชื่อมโยงเต็มรูปแบบจะถูกสุ่มเชื่อมโยงด้วยการแจกแจงแบบเบอร์นูลี่ ถ้าค่าการสุ่มเส้นการเชื่อมโยงเท่ากับ 1 จะยังคงเส้นการเชื่อมโยงของเพอร์เซปตรอนไว้ และถ้าค่าการสุ่มเท่ากับ 0 ข้อมูลจากชั้นก่อนหน้าจะไม่ถูกนำไปคำนวณในเลเยอร์ถัดไป โดยอัตราการดรอปเอาท์จะสามารถถูกกำหนดให้เป็น 0.5 ดังแสดงในรูปที่ 14

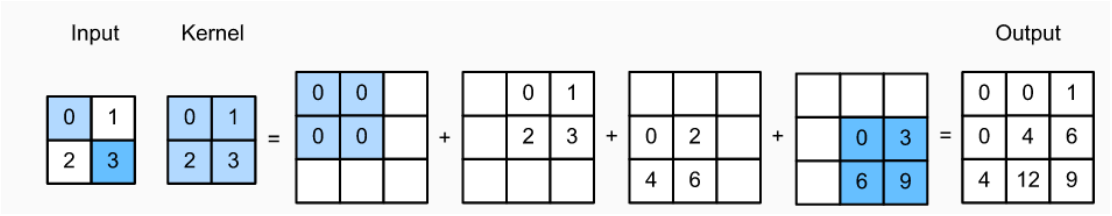


รูปที่ 14 เปรียบเทียบระหว่าง Neural Networks กับ Dropout Neural Networks[13]

2.3.7 ชั้นดีคอนโวลูชัน (Deconvolution Layer)

เป็นการนำฟีเจอร์แมพที่ผ่านการเรียนรู้ด้วยการผ่านชั้นเชื่อมโยงเต็มรูปแบบที่ผ่านการอัพเดทด้วยกระบวนการแพร่ย้อนกลับ (Backpropagation) มาทำการย้อนคำนวณค่ากลับเพื่อแสดงถึงลักษณะเด่นที่ตัวกรองเก็บไว้สำหรับการเรียนรู้การจำแนกภาพ โดยชั้นดีคอนโวลูชันมีบทบาทสำคัญต่อการแบ่งส่วนภาพเชิงความหมายดังแสดงในรูปที่ 15

13 อ้างอิงจาก https://miro.medium.com/max/981/1*EinUlWw1n8vbcLyT0zx4gw.png



รูปที่ 15 ตัวอย่างการคำนวณชั้นดีคอนโวลูชัน[14]

2.4 การประเมินประสิทธิภาพ (Evaluation)

ในงานวิจัยนี้เราจะทำการวัดความถูกต้องด้วยความแม่นยำรายคลาสและการอินเตอร์เซกชันโอเวอร์ยูเนี่ยน (Intersection Over Union), F1 Score เป็นหลัก

2.4.1 คอนฟิวชันเมตริกซ์ (Confusion Matrix)

กำหนดให้ตารางคอนฟิวชันเมทริกซ์ (Confusion Matrix) ดังแสดงในตารางที่ 1

ตารางที่ 1 ตารางคอนฟิวชันเมทริกซ์

	ผลลัพธ์การทำนาย เป็นคลาสจริง	ผลลัพธ์การทำนาย เป็นคลาสเท็จ
ผลเฉลยเป็นคลาสจริง	TP (True Positive)	FN (False Negative)
ผลเฉลยเป็นคลาสเท็จ	FP (False Positive)	TN (True Negative)

กำหนดให้สัญลักษณ์การทำนายในตารางมีความหมายดังนี้

TP หมายถึง จำนวนผลลัพธ์การทำนายเป็นคลาสบวกและผลเฉลยเป็นคลาสบวก

FP หมายถึง จำนวนผลลัพธ์การทำนายเป็นคลาสบวกและผลเฉลยเป็นคลาสลบ

FN หมายถึง จำนวนผลลัพธ์การทำนายเป็นคลาสลบและผลเฉลยเป็นคลาสบวก

TN หมายถึง จำนวนผลลัพธ์การทำนายเป็นคลาสลบและผลเฉลยเป็นคลาสลบ

14 อ้างอิงจาก http://d2l.ai/chapter_computer-vision/transposed-conv.html?highlight=deconvolution

2.4.2 ตัววัดประสิทธิภาพการแบ่งประเภท

มาตรวัดประสิทธิภาพการจำแนกรายคลาสประกอบด้วย Precision, Recall และ F1 โดยมีสูตรการคำนวณดังแสดงในสมการที่ 21 – 23

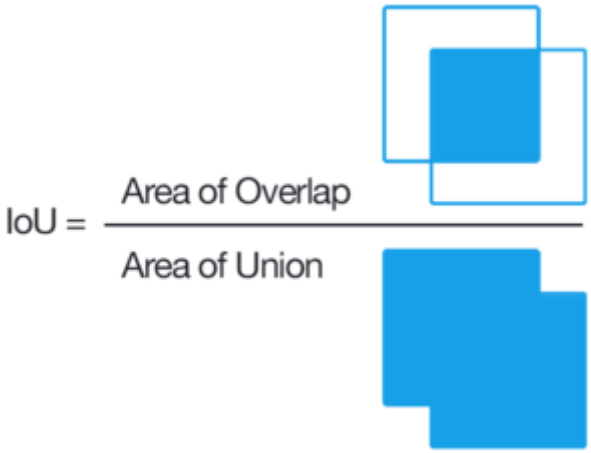
$$Precision = \frac{TP}{TP + FP} \tag{21}$$

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

$$F1 = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \tag{23}$$

2.4.3 มาตรวัดผลด้วยอินเตอร์เซกชันโอเวอร์ยูเนียน (Intersection over Union)

มาตรวัดผลชนิดนี้จะถูกนำมาหาค่าเฉลี่ยกับทุกคลาส เพื่อวัดประสิทธิภาพการทำนายโดยรวมของการแบ่งจำแนกรูปภาพเชิงความหมายดังแสดงในรูปที่ 16 และสมการการคำนวณมาตรวัด Mean IoU ดังแสดงในสมการที่ 24



รูปที่ 16 สูตรการคำนวณ Intersect over Union[15]

$$Mean\ IoU = \frac{1}{N}\sum_{i=1}^{N} IoU(i) \quad \text{เมื่อ } i \text{ คือ ชนิดของคลาส} \tag{24}$$

15 อ้างอิงจาก https://en.wikipedia.org/wiki/Jaccard_index

2.5 การปรับค่าให้เป็นปกติ (Normalization)

คือการปรับสเกลข้อมูลให้อยู่ในมาตรวัดเดียวกัน เนื่องจากชุดข้อมูลภาพถ่ายท้องถนนกรุงเทพมหานครนั้นประกอบด้วยช่องสัญญาณภาพ 3 สี ที่ประกอบไปด้วย สีแดง สีน้ำเงิน และสีเขียว โดยเราจะทำปรับค่าแต่ละช่องสัญญาณที่อยู่ช่วง [0,255] ให้แต่ละทุกช่องสัญญาณมีค่าความเข้มอยู่ในช่วงระหว่าง [0,1) จากการนำค่าเฉลี่ยจากทุกช่องสัญญาณมาลบออกด้วย [123.68, 116.78, 103.94] และหารด้วยส่วนเบี่ยงเบนมาตรฐานในแต่ละทุกช่องสัญญาณตามลำดับ

2.6 การประมาณค่าด้วยไบลิเนียร์ (Bilinear Interpolation)

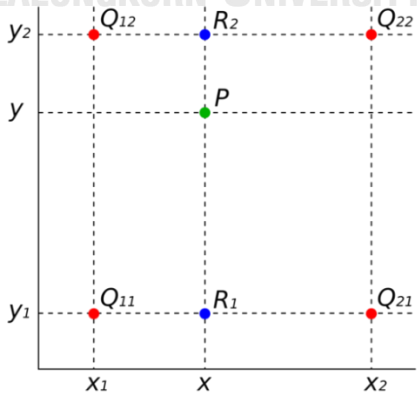
Bilinear Upsampling [17] เป็นการประมาณค่าเมทริกซ์ด้วยการขยายขนาดตัวกรองจากการประมาณด้วยอัตราการเติบโตแบบเส้นตรง สำหรับการทำนายค่าผลลัพธ์การแบ่งส่วนภาพเชิงความหมาย ในชั้นดีคอนโวลูชันในพิกัดระนาบ 2 มิติ ดังแสดงในสมการที่ 25 – 28 และรูปที่ 17 ตามลำดับ

$$f(x,y) = \big(1 - w_y\big) f(R_1) + w_y f(R_2) \tag{25}$$

$$f(R_1) = (1 - w_x) f(Q_{11}) + w_x f(Q_{21}) \tag{26}$$

$$f(R_2) = (1 - w_x) f(Q_{12}) + w_x f(Q_{22}) \tag{27}$$

$$w_x = \frac{x - x_0}{x_1 - x_0} \quad w_y = \frac{y - y_0}{x_1 - x_0} \tag{28}$$



รูปที่ 17 กระบวนการคำนวณการประมาณค่าด้วยไบลิเนียร์[16]

16 อ้างอิงจาก https://en.wikipedia.org/wiki/Bilinear_interpolation

2.7 คะแนนคุณภาพชีวิต (Quality of Life หรือ QOL Scores)

คะแนนคุณภาพชีวิตหมายถึง ระดับความรู้สึกของกลุ่มประชากรผู้เข้ารับการทำแบบทดสอบ ในกรณีของงานวิทยานิพนธ์นี้จะเน้นที่คะแนนความรู้สึกในการใช้ชีวิตในระแวกที่พักอาศัย เมื่อพิจารณาสภาพแวดล้อมบนท้องถนนในกรุงเทพมหานคร ซึ่งในคะแนนเหล่านี้สามารถคำนวณเชิงสถิติเพื่อนำไปตีความในเชิงสังคมศาสตร์ได้ โดยการตีความนี้จะช่วยให้การออกแบบเมืองในอนาคต จะมีความสอดคล้องกับการลักษณะดำรงชีวิตของประชากรกรุงเทพในระแวกนั้นได้ครบทุกแง่มุม

บทที่ 3
งานวิจัยที่เกี่ยวข้อง

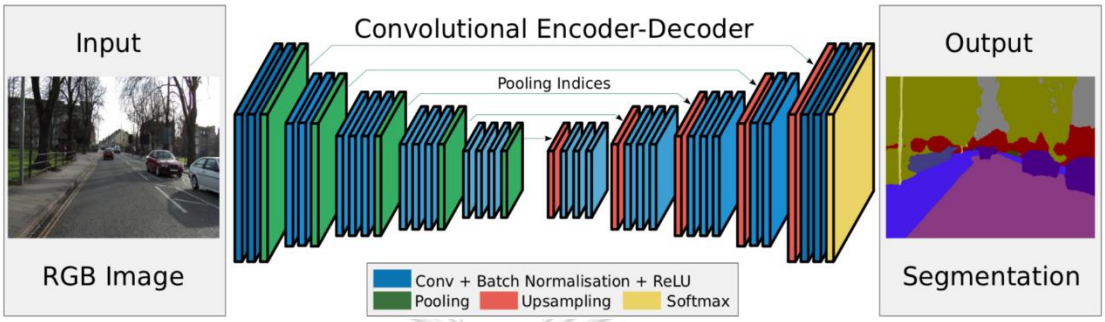
3.1 วิธีที่เกี่ยวข้อง

งานวิจัยการแบ่งส่วนภาพเชิงความหมายด้วยสถาปัตยกรรมการเรียนรู้เชิงลึกที่จะถูกยกขึ้นมาทำการทบทวนวรรณกรรม โดยวิธีเหล่านี้ได้แก่ SegNet, FCN, UNet, PSPNet, Tiramisu, DeepLab-V3+, และ FDNet จะถูกนำมาทบทวนวรรณกรรมอย่างละเอียด

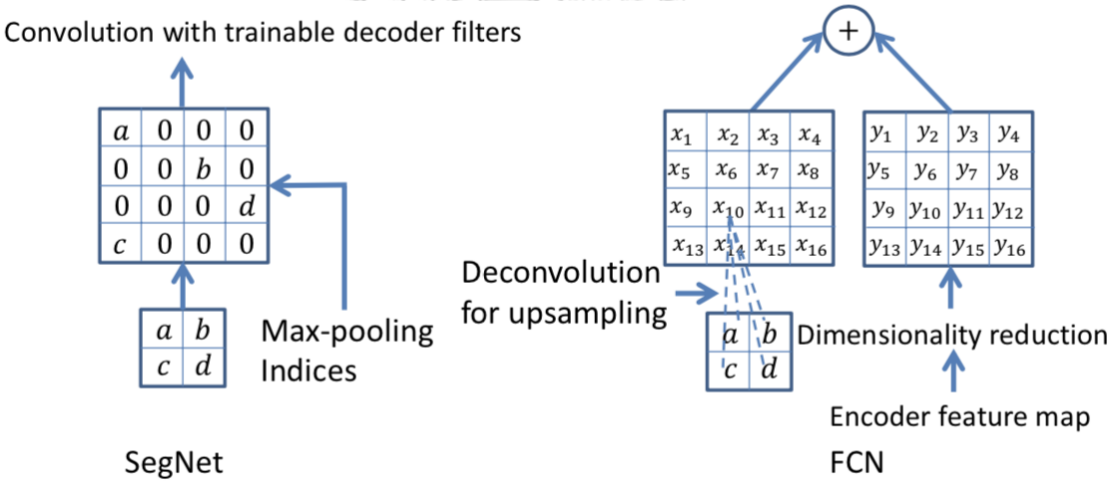
3.1.1 SegNet (A Deep Convolutional Encoder-Decoder Architecture for image Segmentation)

โดยสถาปัตยกรรมนี้จะประกอบไปด้วย 2 ส่วนหลักที่อาศัยโครงข่ายคอนโวลูชันด้วยการเรียนรู้เชิงลึกคือฝั่งเข้ารหัสทำหน้าที่ในการบีบอัดข้อมูล (Encoder) และฝั่งถอดรหัส (Decoder) ทำหน้าที่แปลงข้อมูลที่ถูกบีบอัดให้กลายเป็นผลลัพธ์สำหรับการทำนายการแบ่งส่วนภาพเชิงความหมายด้วยเทคนิคการเรียนรู้เชิงลึก ตัวอย่างสถาปัตยกรรมที่มีลักษณะนี้คือ SegNet: A Deep Convolutional Encoder-Decoder Architecture for image Segmentation [4] ถูกคิดค้นเมื่อปี 2016 โดย Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla โดยแบบจำลองนี้อาศัยได้นำโมเดลที่ผ่านการเรียนรู้การจำแนกประเภทรูปภาพที่มีประสิทธิภาพสูง ณ เวลานั้นบนชุดข้อมูล ImageNet [18] และชื่อของแบบจำลองคือ VGG-16 [19] ถูกนำมาประยุกต์ใช้ทางด้านฝั่งเข้ารหัสประกอบด้วยชั้นการทำคอนโวลูชันจำนวน 13 ชั้น และลบชั้นปลายของโมเดล VGG-16 [19] ที่ทำหน้าที่เพียงแค่การจำแนกประเภทรูปภาพออก เพื่อนำเพียงน้ำหนักที่ผ่านการเรียนรู้การจำแนกประเภทรูปภาพขนาดใหญ่มาใช้ในการบีบอัดข้อมูลด้านฝั่งการเข้ารหัสของ SegNet ด้วยกระบวนการยกองค์ความรู้จากงานประเภทอื่นเพื่อนำมาใช้ในงานลักษณะจำเพาะนี้ถูกเรียกว่า การโอนถ่ายการเรียนรู้ (Transfer Learning) [16] อีกทั้งโมเดล SegNet มีพารามิเตอร์ในการคำนวณที่น้อยกว่า VGG-16 จาก 134 ล้านพารามิเตอร์ เหลือเพียงแค่ 14.7 ล้านพารามิเตอร์ และภาพรวมของสถาปัตยกรรม SegNet ดังแสดงในรูปที่ 18 ฝั่งดีโคดเดอร์ของ SegNet มีความแตกต่างจากสถาปัตยกรรมอื่นด้วยการจำตำแหน่งดัชนีที่ให้ค่าสูงสุดในชั้นพูลลิง (Maxpooling Indices Layer) เพื่อลดระยะเวลาการคำนวณเมื่อเทียบกับสถาปัตยกรรมก่อนหน้าคือ FCN-8s [20] ของการอัพแซมปิง (Upsampling) เป็นขั้นตอนการฟื้นคืนข้อมูลหลังจากที่ถูกบีบอัดและชั้นพูลลิงดังกล่าวที่มีความสมมาตรกันกับฝั่งดีโคดเดอร์จะถูกมาใช้อัพเดทน้ำหนักด้วยชั้นคอนโวลูชันด้วยการแพร่กระจาย

ย้อนกลับ (Backpropagation) สำหรับคำนวณค่าความคลาดเคลื่อนจากชั้นการทำนาย (Classification Layer) ความแตกต่างของขั้นตอนอัพแซมปิงของ SegNet และ FCN-8s ดังแสดงใน รูปที่ 19



รูปที่ 18 ภาพรวมของสถาปัตยกรรม SegNet



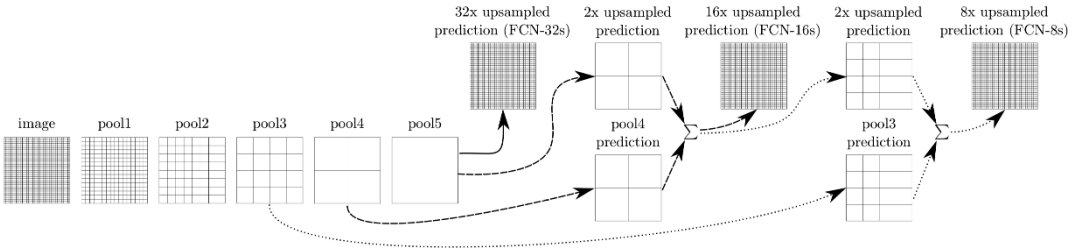
รูปที่ 19 เปรียบเทียบ Max Indices Pooling layer ของ SegNet เทียบกับ Upsampling Deconvolution ของ FCN

สถาปัตยกรรม SegNet ถูกนำมาฝึกจำนวน 140,000 รอบอีพอคบนชุดข้อมูลภาพจากกล้องหน้ารถ (CamVid) ที่มีทั้งหมด 11 คลาส ได้แก่ อาคาร, ต้นไม้, ท้องฟ้า, รถยนต์, สัญลักษณ์จราจร, ถนน, รั้ว, เสา, ทางเท้า โดยประสิทธิภาพของ SegNet ถูกวัดด้วย Mean IoU และ F1 Score บนชุดข้อมูลทดสอบ CamVid เท่ากับมีค่าเท่ากับ 60.10% และ 46.84% ตามลำดับ

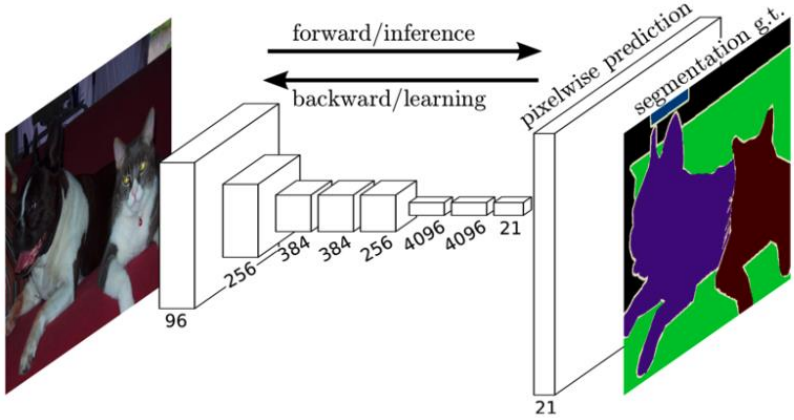
3.1.2 FCN (Fully Convolutional Networks for Semantic Segmentation)

สถาปัตยกรรมลักษณะนี้มีขั้นตอนการบีบอัดข้อมูลทางด้านฝั่งเอนโคดเดอร์และมีการฟื้นฟูข้อมูลที่ขาดหายจากการเข้ารหัสด้วยชั้นดีโคดเดอร์เหมือนกับสถาปัตยกรรม SegNet เหมือนกัน แต่สถาปัยตกรรมประเภทนี้มีความแตกต่างที่การนำฟีเจอร์แมพทางฝั่งเอนโคดเดอร์มาทำการบวกหรือคูณกันกับผลลัพธ์จากการทำดีโคดเดอร์ในแต่ละชั้น เพื่อทำให้สัญญาณข้อมูลที่ถูกบีบอัดไปตั้งแต่ตอนต้นมีความคมชัดขึ้นเพื่อให้การปรับปรุงกระบวนการเรียนรู้ด้วยการแพร่กระจายย้อนกลับมีประสิทธิภาพดีขึ้น

สถาปัตยกรรมชนิดนี้เป็นสถาปัตยกรรมยุคแรกของงานวิจัยการแบ่งส่วนภาพเชิงความหมาย โดยวิธีแรกคือ Fully Convolutional Networks for Semantic Segmentation (FCN) [20] ถูกคิดค้นโดย Jonathan Long, Evan Shelhamer, Trevor Darrell ในปี2015 โดยใช้การถ่ายโอนการเรียนรู้จากสถาปัตยกรรม AlexNet [21] สำหรับการคัดแยกรูปภาพบนชุดข้อมูลรูปภาพขนาดใหญ่ ImageNet เพื่อนำมาใช้สำหรับส่วนเอนโคดเดอร์ และนำฟีเจอร์แมพที่ผ่านการทำแม็กซ์พูลลิงในแต่ละชั้นทำการอัพแซมปิง (Upsampling) ขยายขนาดด้วยจำนวน 32, 16, 8 และ 2 เท่าตามลำดับ และฟีเจอร์แมพในชั้นที่มีมิติเท่ากันกับเมทริกซ์ที่ถูกอัพแซมปิง ถ้ามีจำนวนมิติเท่ากันจะถูกนำมามาบวกกัน เพื่อเพิ่มข้อมูลในส่วนที่ขาดหายและปรับปรุงสัญญาณเกรเดียน เรียกกระบวนการนี้ว่าสคริปคอนเนกชัน (Skip Connection) และนำผลลัพธ์จากการขยายสัญญาณนำไปผ่านชั้นสูงสุดอย่างอ่อนเพื่อทำนายความน่าจะเป็นสำหรับ 1 จุดพิกเซลนั้นควรจะเป็นคลาสไหน และผลลัพธ์ที่ดีที่สุดจากการทำอัพแซมปิงคือ FCN-8s ให้ค่า Mean IoU บนชุดข้อมูลทดสอบ PASCAL VOC 2012 [22] เท่ากับ 62.70% ที่ถูกนำผลลัพธ์การทำปรับปรุงเกรเดียนทุกชั้นมาพิจารณาในส่วนของ Pixel-wise Classification ซึ่งกระบวนการทำสคริปคอนเนกชันดังแสดงในรูปที่ 20 และภาพรวมของสถาปัตยกรรม Fully Convolutional Networks for Semantic Segmentation ดังแสดงในรูปที่ 21



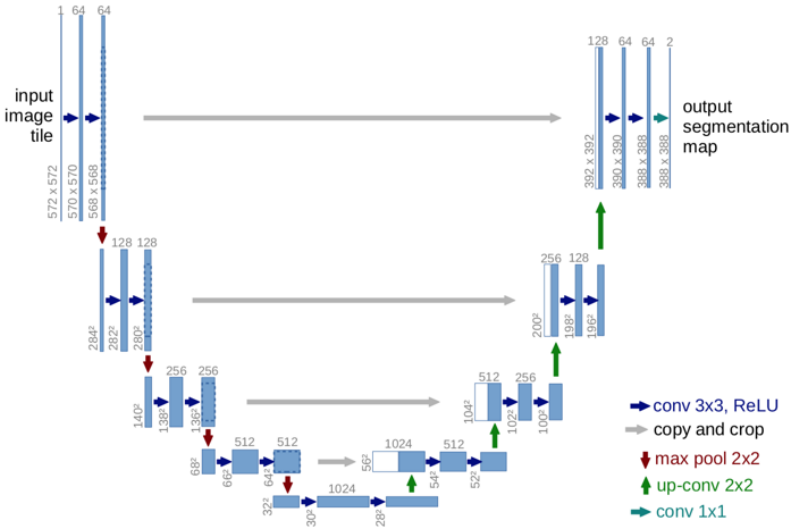
รูปที่ 20 การอัพแซมปิงด้วยสคริปคอนเนกชันในสถาปัตยกรรม Fully Convolutional Networks for Semantic Segmentation (FCN)



รูปที่ 21 ภาพรวมของสถาปัตยกรรม Fully Convolutional Networks for Semantic Segmentation (FCN)

3.1.3 UNet (Convolutional Networks for Biomedical Image Segmentation)

UNet Convolutional Networks for Biomedical Image Segmentation [5] ถูกคิดค้นโดย Olaf Ronneberger, Philipp Fischer, และThomas Brox ในปี 2015 โดยจุดเด่นของสถาปัตยกรรมนี้คือรูปร่างลักษณะที่มีความสมมาตรทั้งด้านดีโคดเดอร์และเอนโคดเดอร์ และมีการนำเมทริกซ์ฟีเจอร์แมพด้านที่สามาตรกับฝั่งอัพแซมปิ้งนำมาบวกกันครบทุกชั้นการคำนวณ ดังแสดงในรูปที่ 22 และผลลัพธ์การทำนายของโมเดลนี้บนชุดทดสอบ ISBI cell tracking challenge PhC-U373 dataset ให้ค่า Mean IoU เท่ากับ 92.03% ประกอบกับคุณสมบัติของการต่อชั้นการคำนวณด้วยวิธีที่เรียบง่ายและให้ประสิทธิภาพที่ดีส่งผลให้แบบจำลองนี้เป็นที่นิยมในงานวิจัยการแบ่งส่วนภาพเชิงความหมายในปัจจุบัน



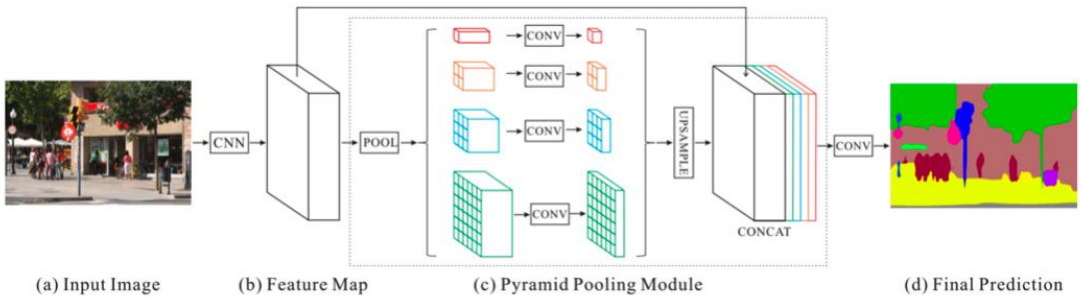
รูปที่ 22 ภาพรวมของสถาปัตยกรรมยูเน็ต (UNet)

3.1.4 PSPNet (Pyramid Scene Parsing Network)

สถาปัตยกรรมนี้ยังคงไว้ทั้งฝั่งเอนโคดเดอร์และดีโคดเดอร์เหมือนเดิม แต่มีความแตกต่างกันในส่วนของดีโคดเดอร์ที่นำฟีเจอร์แมพทางฝั่งเอนโคดเดอร์มาพิจารณา เพื่อปรับปรุงข้อมูลทางฝั่งดีโคดเดอร์สำหรับทำนายความน่าจะเป็นรายพิกเซล โดยฟีเจอร์แมพทางฝั่งเอนโคดเดอร์ถูกนำมาซ้อนกัน อีกทั้งสถาปัตยกรรมนี้จะไม่พิจารณาจำนวนมิติของฟีเจอร์แมพ โดยที่จำนวนมิติสองฝั่งจะต้องเท่ากัน เหมือนกับสถาปัตยกรรมเอนโคดเดอร์ดีโคดเดอร์ด้วยการฟีเจอร์ฟิวชัน จุดเด่นของสถาปัตยกรรมเอนโคดเดอร์ดีโคดเดอร์ด้วยการทำฟีเจอร์คอนแคต มีระยะเวลาการฝึกที่เร็วกว่าในส่วนของการอัพแซมปิง เพราะว่าไม่มีการนำฟีเจอร์แมพเมทริกซ์ฝั่งดีโคดเดอร์มาบวกกันกับฝั่งเอนโคดเดอร์

Pyramid Scene Parsing Network (PSPNet) [6] ได้ถูกคิดค้นขึ้นมาในปี 2017 โดย Hengshuang Zhao และคณะ โดยภาพรวมของสถาปัตยกรรม PSPNet ดังแสดงดังรูปที่ 23 จุดเด่นของสถาปัตยกรรมนี้คือสามารถปรับเปลี่ยนฝั่งเอนโคดเดอร์ได้ และในส่วนของปิรามิดโมดูล มีการนำแม็กพูลลิงจำนวนมิติที่แตกต่างกัน ได้แก่ Global Average Pooling [23] ที่รวบรวมลักษณะเด่นทั้งภาพด้วยการแสดงลักษะนั้นด้วยเมตริก 1 จุด, 2 x 2, 3 x 3, และ 6 x 6 เพื่อทำการเลือกลักษณะเด่นจากฟีเจอร์แมพที่ถูกสกัดมาจากแบบจำลองการจำแนกประเภทของรูปภาพที่ถูกผ่านการโอนถ่ายความรู้ได้แก่ ResNet101 [8], DenseNet100 [24] โดยผลลัพธ์จากการทำปิรามิดโมดูลถูกอัพแซมปิงผนวกกับผลลัพธ์จากการทำฟีเจอร์แมพของโมเดลที่ถูกถ่ายโอนความรู้มาจะถูกนำมาต่อกันโดยตรง ผลลัพธ์จากการทำปิรามิโมดูลและการต่อฟีเจอร์แมพจะถูกนำไปผ่านชั้นคอนโวลูชันเพื่อสกัดลักษณะ

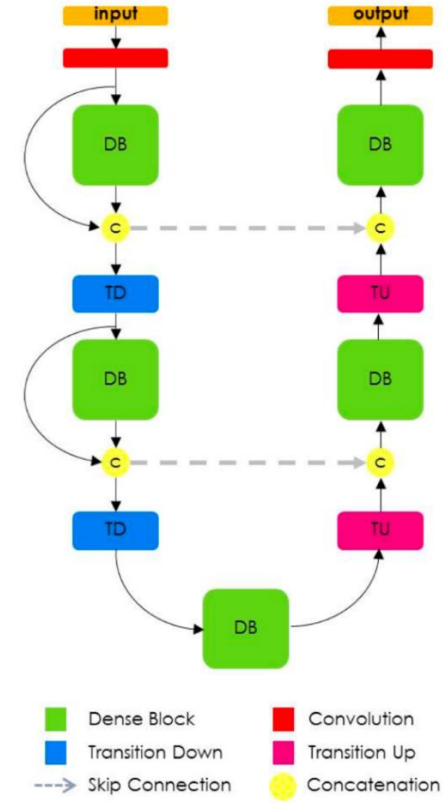
เด่นอีกรอบเพื่อการทำนายทุกพิกเซลจากชั้นดีโคดเดอร์ ผลลัพธ์การทำนายบนชุดทดสอบ PACAL VOC 2012 [22] และ Cityscapes [11] มีค่า Mean IoU เท่ากับ 80.20% และ 85.40% ตามลำดับ



รูปที่ 23 ภาพรวมสถาปัตยกรรม Pyramid Scene Parsing

3.1.5 Tiramisu (The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation)

The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation [3] ถูกคิดค้นขึ้นมาโดย Simon Je´gou และคณะในช่วงปลายปี 2017 โดยสถาปัตยกรรมทิรามิสีได้รับแรงบันดาลใจจาก U-Net [5] โดยนำบล็อกของ Dense Net มาซ้อนกันในลักษณะตัวยูทั้งฝั่งเอนโคดเดอร์และดีโคดเดอร์ ประกอบกับการนำฟีเจอร์แมพในแต่ละชั้นมาซ้อนทับกันแทนการบวกกันในฝั่งของดีโคดเดอร์ โดยฝั่งเอนโคดเดอร์ประกอบด้วย Dense Block จำนวน 5 ชุด และส่วน Dense Block ที่เป็น Bottle Neck ไว้ใช้ทำหรับการลดขนาดมิติของฟีเจอร์แมพด้วย Convolution 1 x 1 จากผลงานของ Network in Network [23] ส่วนฝั่งดีโดดเดอร์ประกอบด้วย Dense Block จำนวน 5 ชุดผสมกับเลอเยอร์อัพแซมปิงและชั้นคอนโวลูชันที่มีขนาด 3 x 3 และ m คือจำนวนฟีเจอร์แมพ ดังแสดงในรูปที่ 24 ผลการทดลองสถาปัตยกรรมทิรามิสีให้ค่า Mean IoU บนชุดข้อมูลทดสอบ CamVid เท่ากับ 66.90% ด้วยการฝึกด้วยการโอนถ่ายความรู้จากฝั่งเอนโคดเดอร์ด้วยโมเดล DenseNet 103 ชั้น โดยภาพรวมของสถาปัตยกรรมของทิรามิสีถูกแสดงดังรูปที่ 24



Architecture
Input, $m = 3$
3×3 Convolution, $m = 48$
DB (4 layers) + TD, $m = 112$
DB (5 layers) + TD, $m = 192$
DB (7 layers) + TD, $m = 304$
DB (10 layers) + TD, $m = 464$
DB (12 layers) + TD, $m = 656$
DB (15 layers), $m = 896$
TU + DB (12 layers), $m = 1088$
TU + DB (10 layers), $m = 816$
TU + DB (7 layers), $m = 578$
TU + DB (5 layers), $m = 384$
TU + DB (4 layers), $m = 256$
1×1 Convolution, $m = c$
Softmax

Layer
Batch Normalization
ReLU
3×3 Convolution
Dropout $p = 0.2$

Transition Down (TD)
Batch Normalization
ReLU
1×1 Convolution
Dropout $p = 0.2$
2×2 Max Pooling

Transition Up (TU)
3×3 Transposed Convolution $stride = 2$

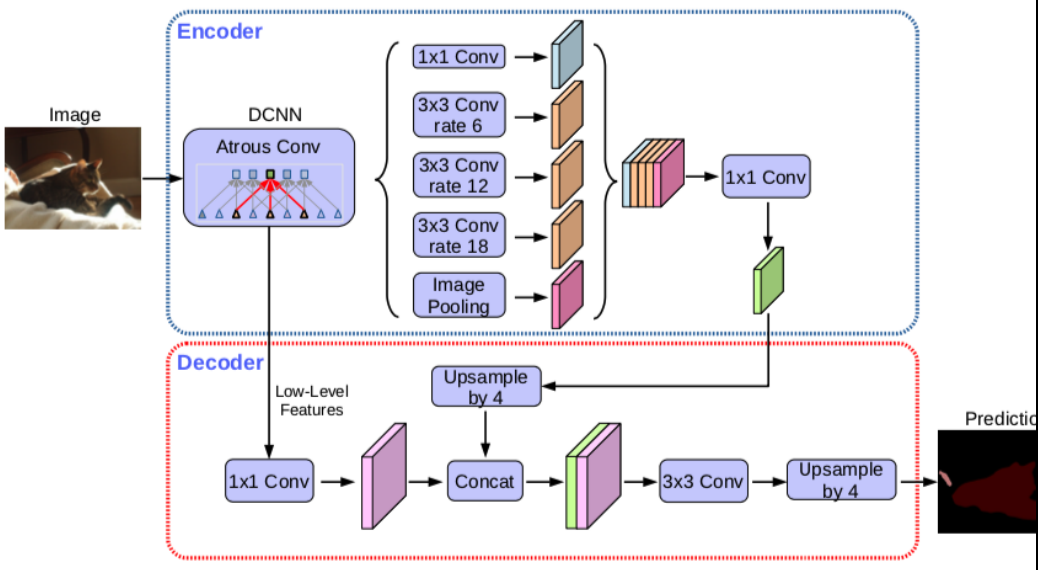
รูปที่ 24 ภาพรวมของสถาปัตยกรรมของทิรามิสึ

3.1.6 DeepLab-V3+ (Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation)

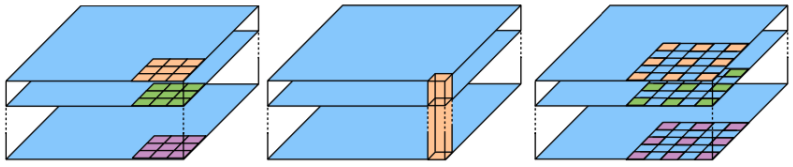
Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation (DeepLab-V3+) [7] โดย Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam ในช่วงปลายปี 2018 ได้มีการนำแนวคิดของการทำเอนโคดเดอร์ดีโคดเดอร์ด้วยการส่งต่อฟีเจอร์แมพมาใช้สำหรับในส่วนของดีโคดเดอร์ และมีการปรับปรุงในส่วนการสกัดฟีเจอร์แมพในส่วนฝั่งเอนโคดเดอร์ด้วยการทำคอนโวลูชันแบบหลุม (Atrous Convolution) ด้วยเรทที่ต่างกัน 3 ขนาดเพื่อให้ชั้นคอนโวลูชันสามารถสกัดข้อมูลหลากหลายขนาดเพื่อลดเวลาการคำนวณการทำคอนโวลูชัน ผนวกกับการทำคอนโวลูชันแบบลึก (1 x 1

Convolution) [18] เพื่อสกัดฟีเจอร์แมพทำให้ได้คุณลักษณะเด่นของข้อมูลอินพุตได้ครบทุกรูปแบบทั้งกว้างยาวและลึก โดยภาพรวมของสถาปัตยกรรม DeepLab-V3+ แสดงดังรูปที่ 25(ก) โดย Xception ที่ถูกพัฒนาสำหรับ DeepLab-V3+ ในชั้นพูลลิงทั้งหมดจะถูกแทนที่ด้วย Depth-wise separable convolution ที่ใช้ในการสกัดฟีเจอร์ทั้งแนวกว้างและลึก ดังแสดงในรูปที่ 25(ข) โดยมีรายละเอียดการปรับปรุงโครงสร้างโมเดล Xception [9] จากการเพิ่มชั้นการทำให้เป็นปกติแบบแบชและ ReLU สำหรับทุกชั้นที่ผ่านการทำ Depth-wise separable convolution ดังแดสงในรูปที่ 25 (ค) ซึ่งการปรับปรุง Xception สำหรับ DeepLab-V3+ ได้รับแรงบันดาลใจมากจาก Mobile-Net [25] ประกอบกับการทำการอัพแซมปิงทางฝั่งดีโคดเดอร์ด้วยการประมาณค่าด้วยฟังก์ชันไบลิเนียร์ 4 เท่า ประสิทธิภาพการทำนายของ Deep Lab-V3+ กับชุดข้อมูลทดสอบบน PASCAL VOC 2012 และ Cityscapes มีค่า Mean IoU เท่ากับ 87.80% และ 79.50% ตามลำดับ

(ก)

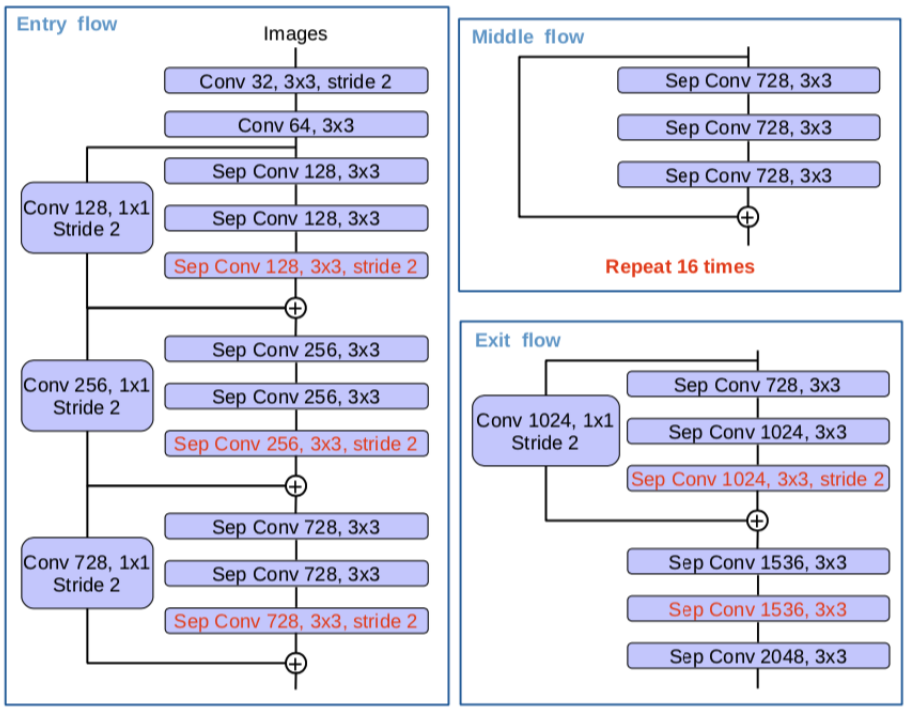


(ข)



(a) Depthwise conv. (b) Pointwise conv. (c) Atrous depthwise conv.

(ค)



รูปที่ 25 ภาพรวมของสถาปัตยกรรมของ DeepLab-V3+

3.1.7 FDNet (Learning Fully Dense Neural Networks for Image Semantic Segmentation)

Learning Fully Dense Neural Networks for Image Semantic Segmentation (FDNet) [26] ถูกคิดค้นในปี 2020 โดย Mingmin Zhen, Jinglu Wang, Lei Zhou, Tian Fang, Long Quan โดย FDNet ถูกคิดค้นขึ้นมาเพื่อการแบ่งส่วนภาพเชิงความหมายโดยที่ Dense Block ของ DenseNet [20] โดยภาพรวมของสถาปัตยกรรม FDNet ดังแสดงในรูปที่ 26(ก) เริ่มจากทางฝั่งเอนโคดเดอร์ของ FDNet จะถูกนำมาเชื่อมโยงกันแบบเต็มรูปแบบด้วยการทำฟีเจอร์คอนแคต กับ Dense Block ทุกชั้นของฝั่งดีโคดเดอร์ ผนวกกับการนำฟีเจอร์ทางฝั่งเอนโคดเดอร์มาทำการคอนแคตและนำมาผ่านชั้นพูลลิงหรือนำมาผ่านชั้นอัพแซมปิ้งด้านฝั่งดีโคดเดอร์ นิยามการนำฟีเจอร์แมพทั้งสองฝั่งมาผสมกันในลักษณะนี้ว่า Adaptive Aggregation ดังแสดงในรูปที่ 26(ข) และประโยชน์ของการเชื่อมโยงกันแบบเต็มรูปแบบของฟีเจอร์แมพด้วยการทำฟีเจอร์คอนแคตทางด้านเอนโคดเดอร์ทำให้ผลลัพธ์การอัพแซมปิงทางด้านดีโคดเดอร์มีฟีเจอร์แมพลักษณะเด่นที่ถูกสกัดทางฝั่งเอนโคดเดอร์ครบทุกชั้นการคำนวณ ส่งผลให้การทำชั้นดีโคดเดอร์มีประสิทธิการทำนายพิกเซลรายคลาสมากขึ้น

แต่ผู้วิจัยได้พบข้อเสียของการใช้ Cross Entropy เป็นฟังก์ชันต้นทุนสำหรับการวัดผลการทำนาย โดยการทำให้ชั้นคอนโวลูชันแรก ๆ ระบุพิกัดขอบของรูปภาพด้วยผลเฉลยวัตถุเพื่อการคำนวณฟีเจอร์แมพชั้นต้น (Low level feature) นั้นเป็นเรื่องที่ยาก ถ้าฟังก์ชันต้นทุนไม่สามารถระบุขอบเขตของพิกัดที่แท้จริงของขอบรูปภาพนำเข้าได้ ดังนั้นผู้วิจัยจึงปรับปรุงฟังก์ชันต้นทุนด้วยแนวคิดจาก Focal Loss [26] ดังแสดงในสมการที่ 29 มาพัฒนาในส่วนของ Weight attention เพื่อให้ฟังก์ชันต้นทุนสามารถระบุขอบเขตของเส้นขอบของรูปภาพที่ถูกชี้นำด้วยการทำ Weight attention ระหว่างการฝึกได้ โดย Weight Attention ถูกกำหนดด้วยสมการเหล่านี้ และพบว่าหากใช้ Dilated convolution ในการขยายขอบของภาพเพื่อสร้างเส้นสมมติในการบ่งบอกพิกัดขอบสำหรับส่วนของฟังก์ชันต้นทุนจะช่วยให้ประสิทธิภาพ Mean IoU มีค่ามากขึ้น ดังแสดงในรูปที่ 26(ค) และจำนวนความหนาขอบ Bandwidth ที่นำมาประมาณช่วงของขอบภาพ อยู่ที่ 40 พิกเซล จะเป็นค่าที่เหมาะสมที่สุด อีกทั้งประสิทธิภาพการทำนายของ FD Net บนชุดข้อมูลทดสอบ PASCAL VOC 2012 มีค่า Mean IoU เท่ากับ 84.20%

โดยกำหนดให้สัญลักษณ์

α_j คือ พารามิเตอร์ควบคุม Weight attention

$L_{i,c}$ คือ ผลลัพธ์จากการทำนายของโมเดลด้วย Softmax

$L_{i,c}^{gt}$ คือ ผลเฉลย

$w(L_{i,c})$ คือ Weight attention

I_i คือ พิกเซลที่ i ในภาพข้อมูลนำเข้า

λ คือ ตัวเลขที่นำมาปรับไฮเปอร์พารามิเตอร์ในฟังก์ชัน Weight attention

S_j คือ เซตของระยะห่างจุดพิกเซลในภาพที่ถูกคำนวณจากขอบภาพด้วยระยะห่างยูคลิเดียน

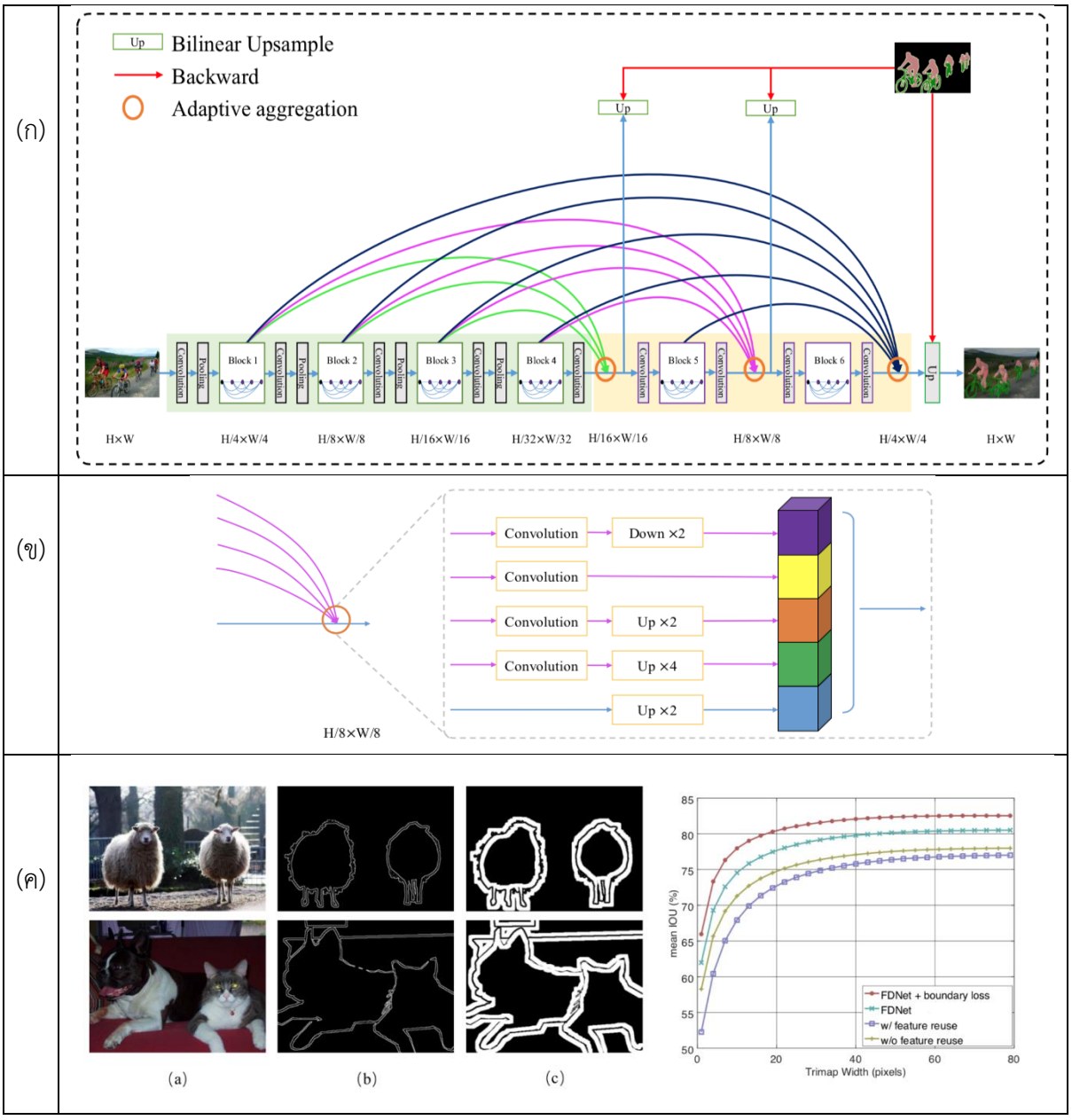
C คือ จำนวนคลาส

j คือ จำนวนตัวกรอง

$$loss(L, L^{gt}) = -\frac{1}{N}\sum_{J=1}^{K}\sum_{I_i \in S_j}\sum_{c=1}^{C}\alpha_j L_{i,c}^{gt} w(L_{i,c}) log L_{i,c} \quad (29)$$

เมื่อ $w(L_{i,c}) = (1 - L_{i,c})^{\lambda}$ *weight attention* ด้วยฟังก์ชันพหุนามและ

$w(L_{i,c}) = e^{-\lambda(1-L_{i,c})}$ *weight attention* ด้วยฟังก์ชันเอกโปเนนเชียล



รูปที่ 26 ภาพรวมของสถาปัตยกรรมของ FDNet

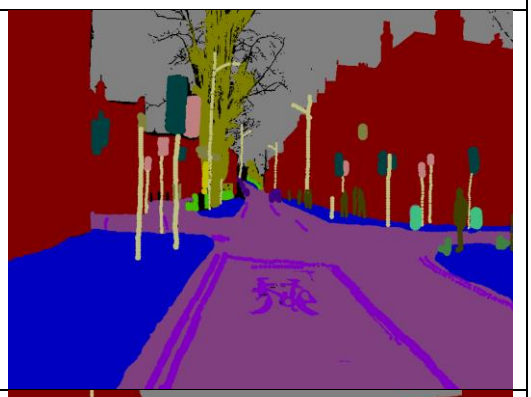
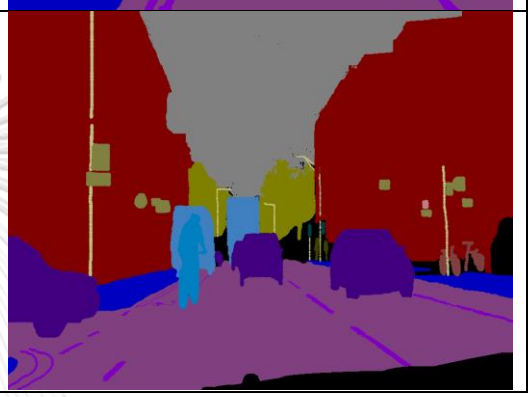
3.2 ชุดข้อมูลที่เกี่ยวข้อง

ในวิทยานิพนธ์นี้จะนำชุดข้อมูลภาพถ่ายท้องถนนที่เป็นมาตรฐานในการวัดผลที่ใช้ในงานวิจัยการแบ่งส่วนเชิงความหมาย มาทดลองเพื่อวัดผลในวิธีที่เรานำเสนอเมื่อเทียบกับวิธีการมาตรฐาน (Baseline) โดยชุดข้อมูลมาตรฐานที่จะถูกนำมาทบทวนวรรณได้แก่ ชุดข้อมูล CamVid และ Cityscapes

3.2.1 ชุดข้อมูล CamVid

ชุดข้อมูล Cambridge-driving Labeled Database (CamVid) [10] เป็นชุดข้อมูลแรกที่ถูกบันทึกในเมืองเคมบริดจ์ ประเทศอังกฤษ ชุดข้อมูลนี้ประกอบด้วยสภาพการขับขี่รถยนต์ที่หลากหลากหลาย ที่สะท้อนถึงสภาพการขับขี่จริงในเมืองเคมบริดจ์ ประกอบด้วยสภาพอากาศที่แตกต่างกัน ได้แก่ สภาพอากาศปกติ และอากาศมืดครึ้ม ดังแสดงในรูปที่ 27(ก) และ 27(ข) ตามลำดับ โดยมีการบันทึกสภาพขับขี่ในชุมชน จนไปถึงถนนใหญ่ การบันทึกวีดีโอในชุดข้อมูลนี้ถูกบันทึกด้วยกล้องวีดีโอ Panasonic HVX200 ซึ่งมุมมองในวีดีโอจะอยู่ในตำแหน่งผู้ขับเท่านั้น โดยมีความยาวทั้งหมด 2 ชั่วโมง วีดีโอมีความละเอียดที่ 960 x 720 พิกเซล และจำนวนภาพ 30 Frame per sec (FPS) ประกอบกับการใช้เทคนิค Calibration ที่กล้องวีดีโอก่อนการถ่ายทำด้วยวิธี Intrinsic และ Extrinsic ตามลำดับ และวีดีโอที่ได้ในข้างต้นจะถูกคัดเลือกนำมาใช้สร้างชุดข้อมูลภาพถ่ายท้องถนน CamVid เพียง 22 นาที ในลำดับถัดมา

วีดีโอที่ผ่านการคัดเลือกนั้นจะถูกทำให้เป็นภาพ จำนวน 701 ภาพ เพื่อนำไปสร้างผลเฉลยด้วยรูปทรงหลายเหลี่ยม (Polygon) จากโปรแกรม “InteractLabeler” ที่ทีมวิจัยทีได้สร้างขึ้นมาเพื่อให้อาสาสมัครใช้สำหรับการสร้างผลเฉลยบนชุดข้อมูล CamVid หลังตรวจสอบความถูกต้องของผลเฉลยทั้งหมดแล้ว ชุดข้อมูลรูปภาพทั้งหมด 701 ภาพ ทั้งภาพนำเข้า และ ผลเฉลยจะถูกแบ่งเป็น 3 ส่วน ได้แก่ ชุดข้อมูลฝึก 367 ภาพ, ชุดข้อมูลตรวจสอบ 101 ภาพ, และชุดข้อมูลทดสอบ 323 ภาพ ซึ่งมีทั้งหมด 32 คลาส โดยประกอบไปด้วยคลาสต่าง ๆ ได้แก่ Building, Wall, Tree, Road, Pedestrian, Car, และ Sidewalk ดังแสดงในรูปที่ 28

	ภาพถ่ายนำเข้า	ผลเฉลย
(ก)		
(ข)		

รูปที่ 27 ตัวอย่างชุดข้อมูลภาพถ่ายท้องบนถนน CamVid ที่ประกอบด้วยภาพข้อมูลนำเข้าและผลเฉลย (ก) สภาพอากาศปกติ (ข) สภาพอากาศมืดครึ้ม

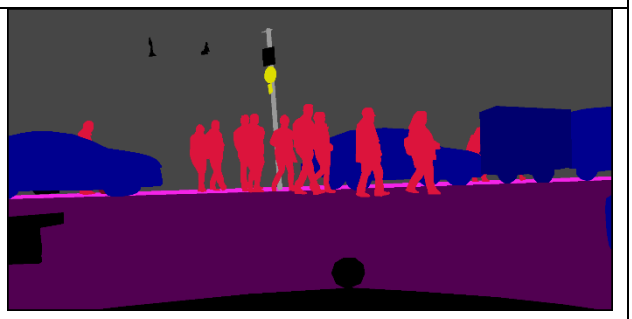
Void	Building	Wall	Tree	VegetationMisc	Fence
Sidewalk	ParkingBlock	Column_Pole	TrafficCone	Bridge	SignSymbol
Misc_Text	TrafficLight	Sky	Tunnel	Archway	Road
RoadShoulder	LaneMkgsDriv	LaneMkgsNonDriv	Animal	Pedestrian	Child
CartLuggagePram	Bicyclist	MotorcycleScooter	Car	SUVPickupTruck	Truck_Bus
Train	OtherMoving				

รูปที่ 28 คลาสของวัตถุในชุดข้อมูลรูปภาพท้องถนน CamVid ที่ถูกแสดงด้วยชุดสี

3.2.2 ชุดข้อมูล Cityscapes

ชุดข้อมูล Cityscapes [11] เป็นชุดข้อมูลภาพถ่ายท้องถนนมาตรฐานที่ได้รับการยอมรับในงานแบ่งส่วนเชิงความหมายสำหรับการประยุกต์ใช้บนระบบขับเคลื่อนอัตโนมัติ โดยชุดข้อมูลนี้ประกอบด้วยสภาพการขับขี่ในท้องถนนด้วยรถยนต์ 50 เมือง ในทวีปยุโรป โดยจะเน้นที่ประเทศเยอรมนีเป็นหลัก โดยระบบการบันทึกวีดีโอการขับขี่ในชุดข้อมูลนี้มีมาตรฐานสูงด้วยกล้องสเตอริโอ High Dynamic Rate (HDR) ประกอบกับระบบขับเซ็นเซอร์ 1/3 CMOS 2 ล้านพิกเซล ที่มี Rolling shutter speed 17 Hz ผนวกกับระบบระบุพิกัด GPS ตลอดเส้นทางการถ่ายทำ ชุดข้อมูลนี้ประกอบด้วยสภาพการขับขี่รถยนต์ที่หลากหลากหลาย ที่สะท้อนถึงสภาพการขับขี่จริงในยุโรป ประกอบกับการบันทึกสภาพขับขี่ในชุมชน จนไปถึงถนนใหญ่ ซึ่งมุมมองในวีดีโอจะอยู่ในตำแหน่งผู้ขับเท่านั้น วีดีโอมีความละเอียดที่ 2048 x 1024 พิกเซล โดยมีการแข่งขันเพื่อพัฒนาโมเดลสำหรับงานวิจัยด้วยชุดข้อมูล Cityscapes อยู่หลายประเภทได้แก่ การแบ่งส่วนภาพโดยฉับพลัน (Instance segmentation) และ การแบ่งส่วนภาพเชิงความหมาย (Semantic segmentation)

ในวิทยานิพนธ์นี้จะเน้นทบทวนวรรณกรรมสำหรับชุดข้อมูล Cityscapes ที่งานแข่งการแบ่งส่วนภาพเชิงความหมายเป็นหลัก โดยปริมาณของชุดข้อมูลงานแข่งการแบ่งส่วนเชิงความหมาย มีชุดข้อมูลภาพนำเข้าพร้อมภาพที่ผ่านการทำผลเฉลยโดยละเอียดเป็นจำนวน 5,000 ภาพ โดยถูกแบ่งเป็น 3 ส่วนได้แก่ ชุดข้อมูลฝึก 2,975 ภาพ, ชุดข้อมูลตรวจสอบ 500 ภาพ, และชุดข้อมูลทดสอบ 1,525 ภาพ อีกทั้งยังมีชุดข้อมูลผลเฉลยแบบหยาบจำนวน 20,000 ภาพ เพื่อเป็นตัวอย่างเสริมสำหรับการฝึกโมเดล กระบวนการสร้างผลเฉลยสำหรับชุดข้อมูล Cityscapes จะถูกสร้างด้วยอาสาสมัครที่ผ่านการอบรมการใช้โปรแกรม LabelMe [12] เพื่อสร้างผลเฉลยด้วยรูปทรงหลายเหลี่ยม (Polygon) ในกระบวนการควบคุมคุณภาพในชุดข้อมูลจะมีการตรวจสอบความถูกต้องของภาพผลเฉลยที่ได้รับจากอาสมัครอย่างเคร่งครัด โดยชุดข้อมูลนี้มีทั้งหมด 29 คลาส โดยคลาสของวัตถุในชุดข้อมูลนี้ถูกจัดกลุ่มเป็น 5 กลุ่ม ได้แก่ nature, vehicle, sky, object, human, และ void โดยตัวอย่างชุดข้อมูลฝึกดังแสดงในภาพที่ 29

ภาพถ่ายนำเข้า	ผลเฉลย
	

รูปที่ 29 ตัวอย่างชุดข้อมูลฝึกใน Cityscapes จากเมือง Aachen ประเทศเยอรมนี

บทที่ 4
แนวคิดและวิธีการดำเนินงาน

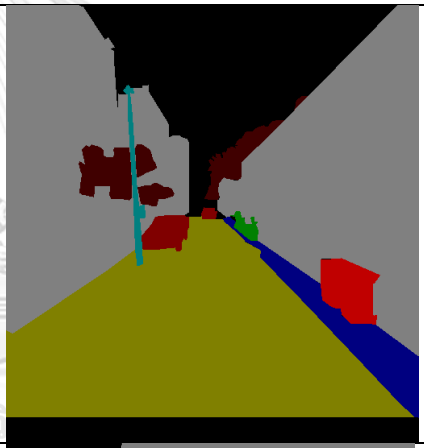
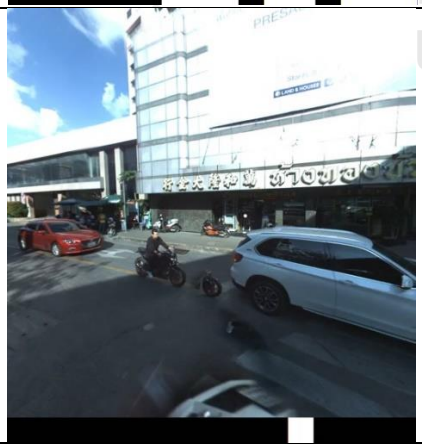
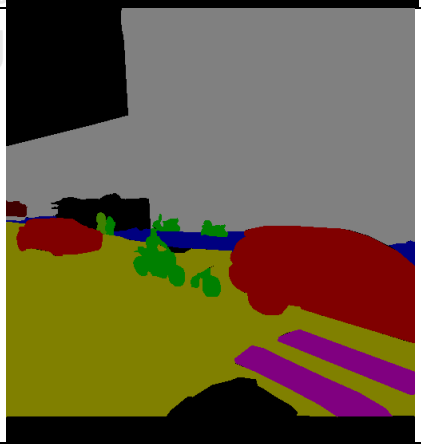
ในบทนี้ได้นำเสนอชุดข้อมูลรูปภาพถนนกรุงเทพมหานครเส้นสุขุมวิทที่รวบรวมโดย Iwahori Lab [27] ประกอบกับวิธีการที่นำเสนอ โดยมีขั้นตอนการดำเนินงานวิจัย 4 ขั้นตอนหลัก ดังแสดงในรูปที่ 30 1. การเตรียมชุดข้อมูลนี้จะถูกนำไปทำผลเฉลย 2. ขั้นตอนการแปลงข้อมูลภาพก่อนทำหน้าที่ป้อนรูปภาพเข้าโมเดลสำหรับขั้นตอนการฝึก 3. ขั้นตอนการฝึก โดยสถาปัตยกรรมที่ถูกคิดค้นขึ้นมาใหม่จะถูกฝึกด้วยผลเฉลย 4. ขั้นตอนการประเมินผล สำหรับวัดผลและปรับปรุงประสิทธิภาพของสถาปัตยกรรมที่นำเสนอเมื่อเปรียบเทียบกับสถาปัตยกรรมมาตรฐาน

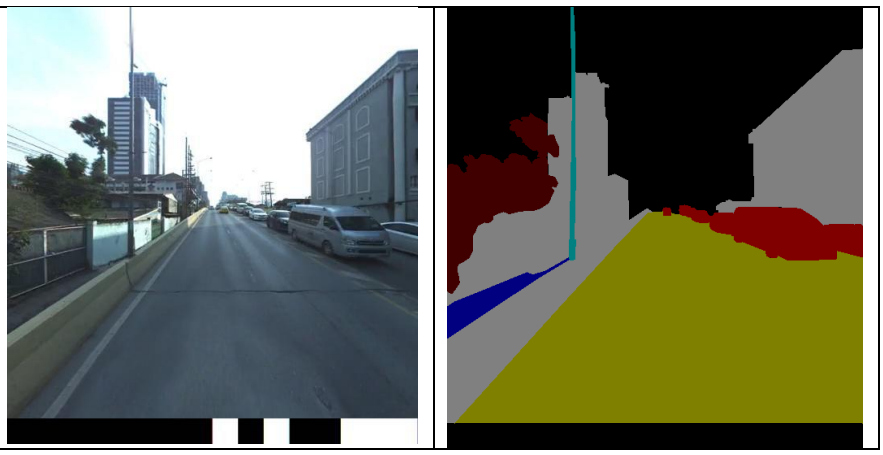
รูปที่ 30 ขั้นตอนดำเนินการ

4.1 ชุดข้อมูลถนนสุขุมวิท

ชุดข้อมูลภาพถ่ายท้องถนนกรุงเทพมหานคร ในชุดข้อมูลนี้ถูกบันทึกสภาพการจราจรในถนนสุขุมวิทเป็นหลัก โดยสมาชิกของ Iwahori Lab [27] และปริมาณภาพถ่ายในชุดข้อมูลนี้มีทั้งหมด 10,000 ภาพ โดยวิทยานิพนธ์นี้ได้คัดเลือกรูปภาพมาทั้งหมด 701 ภาพ โดยแบ่งสัดส่วนของชุดข้อมูลนี้ เป็น 3 ส่วนหลักได้แก่ ชุดข้อมูลฝึก 367 ภาพ, ชุดตรวจสอบ 101 ภาพ, และชุดทดสอบ 233 ภาพ โดยคลาสของวัตถุบนท้องถนนในชุดข้อมูลนี้มีทั้งหมด 11 คลาส ได้แก่ ถนน, ฉากหลัง, เสาไฟ, ฟุตปาธ, รถยนต์, รถจักรยานยนต์, ต้นไม้, คน, ถังขยะ, สิ่งก่อสร้าง, และทางม้าลาย โดยในรายละเอียดในการเก็บชุดข้อมูลถนนย่านสุขุมวิทแบ่งได้เป็น 2 กรณี ได้แก่การบันทึกวีดิโอถนนในซอยที่พักอาศัยบริเวณและถนนหลักในช่วงตอนกลางวัน โดยสภาพแสงแดดในชุดข้อมูลส่วนใหญ่จะมีแสงสว่างค่อนข้างดี และในบางรูปจะถูกเงาของต้นไม้ในบริเวณข้างถนนบัง เราได้ทำผลเฉลยชุดข้อมูลนี้ด้วยโปรแกรม LabelMe [12] จุดเด่นของโปรแกรมนี้คือเป็นซอฟต์แวร์ใช้ฟรี และสามารถนำมาผลเฉลยได้ทุกงานวิจัยการมองเห็นทางคอมพิวเตอร์รวมถึงงานการแบ่งส่วนภาพรูปภาพเชิงความหมาย และผลลัพธ์จากการทำผลเฉลยจะอยู่ในรูปไฟล์ .json เราจะแปลง .json ให้เป็นรูปภาพ .png เพื่อทำการ

ตรวจสอบผลลัพธ์การทำผลเฉลยและเทียบกับชุดสีของคลาสของผลเฉลย โดยสีของคลาสทั้งหมด 11 คลาส ในชุดข้อมูลถนนกรุงเทพฯ จะถูกเข้ารหัสสีโดยอ้างอิงคลาสวัตถุที่ตรงกันกับชุดสีของชุดข้อมูล PASCAL VOC 2011 ดังแสดงในรูปที่ 32 ระยะเวลาในการสร้างผลเฉลยจากภาพนำเข้าในชุดข้อมูลถนนกรุงเทพฯ ด้วยฟังก์ชันรูปทรงหลายเหลี่ยม (Polygon) ด้วยโปรแกรม LabelMe ต่อ 1 ภาพ ใช้เวลาขั้นต่ำโดยเฉลี่ยประมาณ 25 นาที และโดยมีการตรวจสอบ 2 รอบเป็นอย่างต่ำ เพื่อคัดเลือกผลเฉลยหลังการตรวจสอบเป็นผลเฉลยที่สมบูรณ์ที่สุดบนชุดข้อมูลถนนกรุงเทพมหานคร โดยมีตัวอย่างชุดข้อมูลภาพถ่ายท้องถนนกรุงเทพมหานครดังแสดงในรูปที่ 31

	ภาพถ่ายนำเข้า	ผลเฉลย
1.		
2.		

3. 

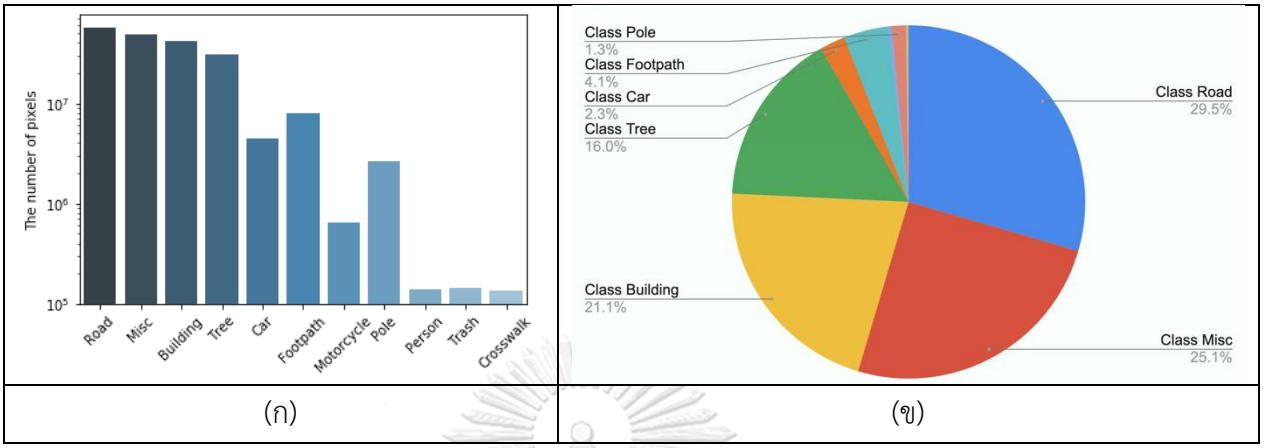
รูปที่ 31 ข้อมูลรูปภาพนำเข้าถนนสุขุมวิทและผลเฉลย

Colors											
Class	Road	Footpath	Crosswalk	Building	Pole	Trash	Car	Motorcycle	Person	Tree	Misc

รูปที่ 32 สีของผลเฉลยที่บ่งบอกถึงคลาสของวัตถุ

ในส่วนของการวิเคราะห์เชิงปริมาณ เราได้แสดงผลลัพธ์ของการทำผลเฉลยบนชุดข้อมูลกรุงเทพมหานคร เริ่มจากกราฟแท่งที่แสดงจำนวนพิกเซลของผลเฉลยของชุดข้อมูลนี้ในแกน Y ในรูปแบบของ logarithmic scale และในแกน X แสดงถึงประเภทคลาสของวัตถุ ดังแสดงในรูปที่ 33(ก) ประกอบกับกราฟวงกลมที่แสดงให้เห็นถึงสัดส่วนของคลาสที่เป็นองค์ประกอบในชุดข้อมูลถนนกรุงเทพฯ ในหน่วยของเปอร์เซ็นต์ ดังแสดงในรูปที่ 33(ข) โดยจากการวิเคราะห์ทั้งสองกราฟ พบว่ามีความเกี่ยวเนื่องกันโดยตรง ซึ่งสามารถสรุปได้ว่า มีคลาสหลักที่ปรากฏมากที่สุดในแต่ละภาพของผลเฉลย เป็นจำนวน 4 คลาส ได้แก่ คลาส Building, Misc, Building, และ Tree โดยคลาสหลักนั้นมีจำนวนไม่ต่ำกว่า 30 ล้านพิกเซล ดังแสดงในรูปที่ 33(ก) ประกอบกับในชุดข้อมูลของเรานั้นมีคลาสหลักเหล่านี้เป็นองค์ประกอบไม่ต่ำกว่า 91% โดยคลาส Building, Misc, Building, และ Tree ได้ปรากฏบนชุดข้อมูลภาพถ่ายถนนกรุงเทพฯ เป็นจำนวน 29.5%, 25.1%, 21.1%, 16.0% ตามลำดับ ดังแสดงในรูปที่ 33(ก) ในขณะที่มี 4 คลาส ได้แก่ Motorcycle, Person, Trash, และ Crosswalk เป็นคลาสส่วนน้อยที่ไม่ได้ถูกแสดงในแผนภูมิวงกลมดังแสดงในรูปที่ 33(ข) เนื่องจากคลาสส่วนน้อยทั้ง 4 คลาส มีพิกเซลในแต่คลาสที่ปรากฏบนชุดข้อมูลท้องถนนกรุงเทพฯ เป็นจำนวนน้อย โดยมีไม่ถึง

1 ล้าน พิกเซล โดยเราสามารถเห็นถึงความแตกต่างของหมวดหมู่คลาสหลักและคลาสย่อย จากกราฟแท่งดังแสดงในรูปที่ 33(ก)



(ก) (ข)

รูปที่ 33 (ก) กราฟแท่งที่แสดงให้เห็นถึงจำนวนพิกเซลในแต่ละคลาส ใน Logarithmic Scale ที่ปรากฏบนชุดข้อมูลภาพถ่ายท้องถนนกรุงเทพฯ (ข) แผนภูมิวงกลมที่แสดงให้เห็นถึงสัดส่วนของพิกเซลรายคลาสที่ปรากฏบนชุดข้อมูลภาพถ่ายท้องถนนกรุงเทพฯ ในหน่วยเปอร์เซ็นต์

4.2 การประมวลผลข้อมูลก่อน (Pre-Processing)

ในส่วนของขั้นตอนการประมวลผลก่อนมีเพื่อการจัดเตรียมชุดข้อมูลให้พร้อมสำหรับการฝึกเพื่อให้การฝึกแบบจำลองเกิดประสิทธิภาพสูงสุด ขั้นตอนการประมวลผลก่อนนั้นจะมีประกอบไปด้วย 3 ขั้นตอนหลัก โดยเริ่มจากการทำนอร์มัลไลเซชัน, การแต่งเติมชุดข้อมูล, และการแปลงชุดข้อมูลภาพถ่ายให้อยู่ในรูปของนัมพายอะเรย์ดังแสดงในรูปที่ 34 และ 35 ตามลำดับ

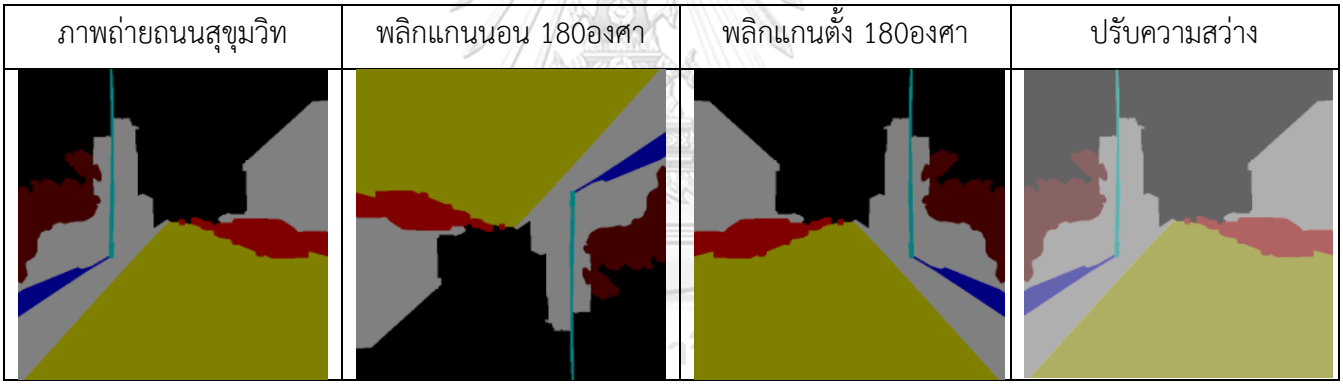
4.2.1 การปรับค่าให้เป็นปกติ (Normalization)

ในชุดข้อมูลภาพถ่ายถนนสุขุมวิทประกอบด้วยค่าสีในช่องสัญญาณ สีแดง สีเขียว และสีน้ำเงิน ที่มีค่าตั้งแต่ 0 ถึง 255 การปรับค่าให้ของช่องสัญญาณเหล่านี้ให้อยู่ระหว่าง 0 ถึง 1 ซึ่งเป็นความเข้มโทนสีขาวดำ ด้วยการลบออกด้วยค่าเฉลี่ยของสีในแต่ละช่องสัญญาณจากการคำนวณจากชุดข้อมูลด้วยค่าเฉลี่ย [123.68, 116.78, 103.94] และหารด้วยส่วนเบี่ยงเบนมาตรฐาน [123.68, 116.78, 103.94] ประโยชน์ของการปรับค่าให้เป็นปกติสามารถช่วยให้การคำนวณการแพร่กระจายย้อนระหว่างการฝึกโมเดลลู่เข้าสู่จุดต่ำที่สุดได้อย่างรวดเร็ว

4.2.2 การแต่งเติมชุดข้อมูล (Data Augmentation)

เนื่องจากข้อมูลฝึกมีจำนวนจำกัด เราจำต้องอาศัยการเพิ่มชุดข้อมูลด้วยการแต่งเติมข้อมูลด้วยการพลิกภาพด้านแกนตั้ง และแกนนอน ปรับความสว่างดังแสดงในรูปที่ 34 และ 35

รูปที่ 34 การเพิ่มชุดข้อมูลถนนสุขุมวิทด้วยการแต่งเติม



รูปที่ 35 การเพิ่มผลเฉลยด้วยการแต่งเติม

4.3 การแปลงชุดข้อมูลภาพเป็นนัมพายอาเรย์

ชุดข้อมูลรูปภาพจะต้องถูกแปลงจากค่าสีให้อยู่ในรูปแบบของเมตริกตัวเลข เพื่อให้คอมพิวเตอร์สามารถคำนวณด้วยโปรแกรมไพธอน ซึ่งนัมพายเป็นไลบารีหนึ่งของภาษาไพธอน โดยการแปลงชุดข้อมูลรูปภาพให้อยู่ในรูปแบบของเมตริกซ์นั้นจะสามารถลดเวลาการป้อนข้อมูลเข้าสู่โมเดลเพื่อทำการฝึก

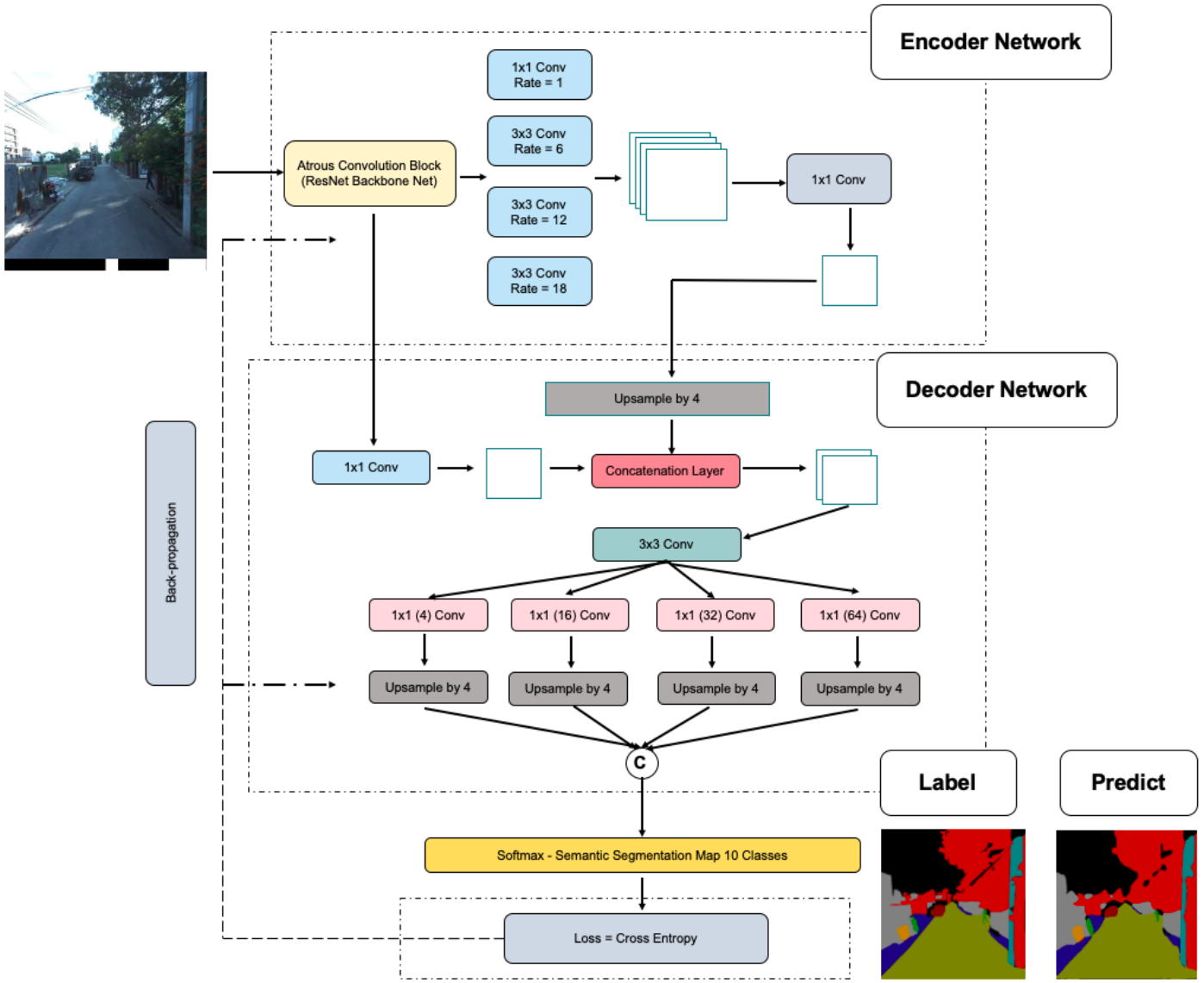
4.4 การแบ่งส่วนภาพเชิงความหมายด้วยวิธีการที่นำเสนอ

ในงานวิจัยการแบ่งส่วนภาพเชิงความหมายบนชุดข้อมูล PASCAL VOC 2012 ได้มีแบบจำลองที่เข้ามาแข่งขันและหนึ่งในโมเดลการแบ่งส่วนภาพเชิงความหมายด้วยการเรียนรู้เชิงลึกที่ได้ประสบผลสำเร็จมากที่สุดคือ ตระกูล DeepLab จากนักวิจัยบริษัท Google ที่ได้พัฒนา DeepLab มาทั้งหมด 4 เวอร์ชัน ตั้งแต่ DeepLab-V1 [15]: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs ที่ใช้การทำคอนโวลูชันแบบหลุม (Atrous Convolution) และการทำการประมวลผลลัพธ์หลังการทำนายด้วยชั้นเชื่อมโยงเต็มรูปแบบด้วยคอนดิชันนอลแรนดอมฟิลด์ (Fully Connected Conditional Random Field Layer) ประสิทธิภาพของ DeepLab-V1 ด้วย VGG16 บนชุดทดสอบ VOC 2012 มีค่า Mean IoU เท่ากับ 71.6%

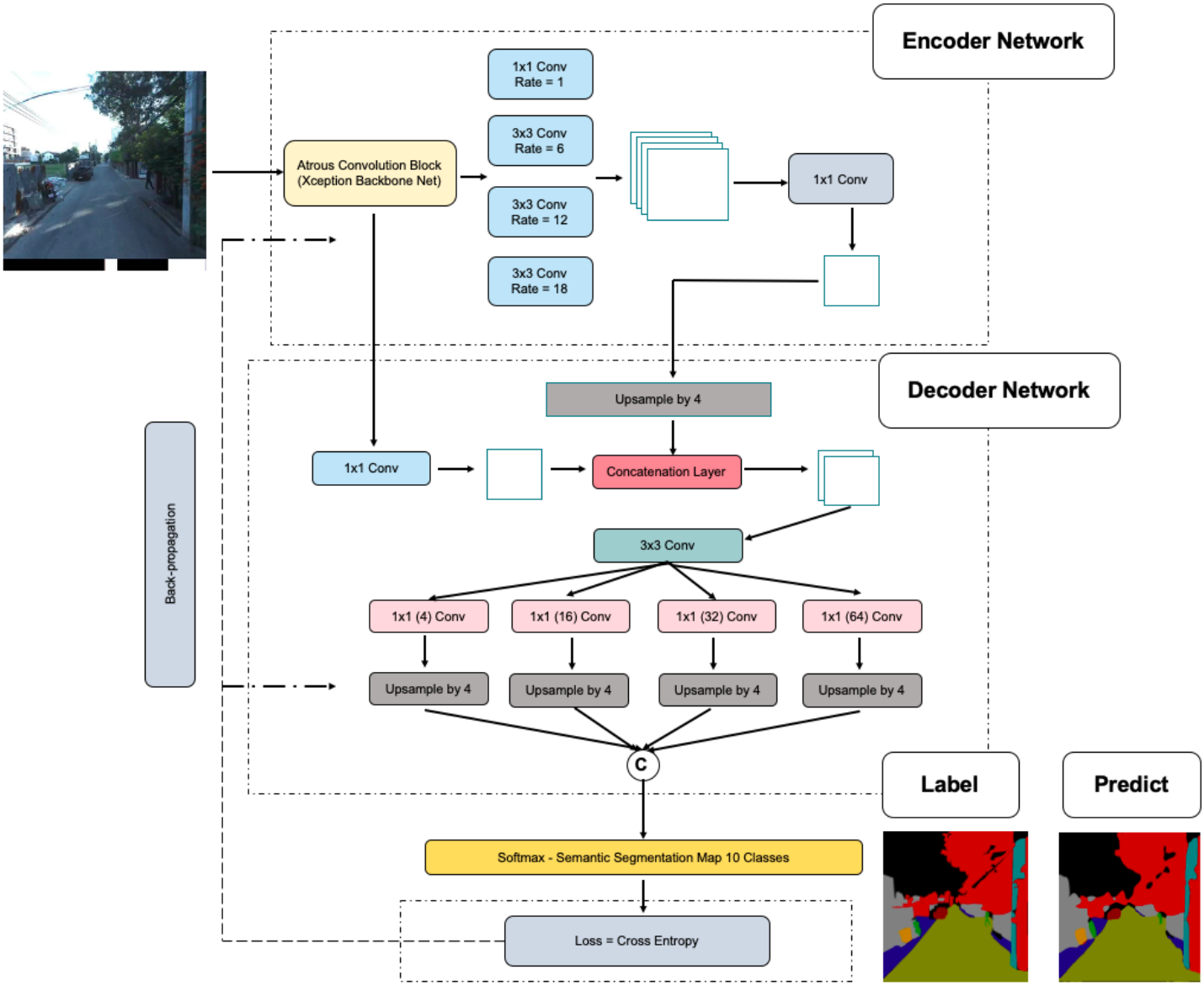
ในลำดับถัดมา ได้มีการคิดค้นสถาปัตยกรรมเรียนรู้เชิงลึกด้วย DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs (DeepLab-V2) [28] ได้มีการปรับปรุงในส่วนของเอนโคเดอร์ที่อาศัยกระบวนการทำคอนโวลูชันแบบหลุมให้มีหลากหลายขนาด (Atrous Spatial Pyramid Pooling) เพื่อขยายระยะของคอนโวลูชันหลุมให้สามารถสกัดลักษณะเด่นได้ทุกระดับ และนำฟีเจอร์แมพเหล่านั้นมาซ้อนทับกันเพื่อใช้ในทำนายด้านฝั่งดีโคดเดอร์ ประกอบกับการประมวลผลหลังการทำนายด้วย Fully Connected CRF Layer โดยประสิทธิภาพของ DeepLabV2-ResNet101 บนชุดข้อมูลทดสอบ PASCAL VOC 2012 มีค่า Mean IoU เท่ากับ 79.9% ในภายหลังมีการปรับปรุงสถาปัตยกรรมของ DeepLab-V2 เป็น Rethinking Atrous Convolution for Semantic Image Segmentation (DeepLab-V3) [29] โดยการนำชั้นการประมวลผลหลังการทำนายออก และเพิ่มชั้นการนอมอลไลซ์เซชันใน Atrous Spatial Pyramid Pooling เพื่อทำให้สะดวกต่อการฝึกและการคำนวณเกรเดียน และประสิทธิภาพการทำนายของ DeepLabV3-ResNet101 ให้ค่า Mean IoU บนชุดข้อมูลทดสอบบน PASCAL VOC 2012 เท่ากับ 85.7% และผลงานวิจัยล่าสุด DeepLab V3+ [7] เกิดจากการปรับปรุงชั้นดีโคดเดอร์ในส่วนของการสกัดลักษณะเด่นด้วย Xception [9] ที่ผ่านนำ Max-pooling ออกและถูกแทนที่ด้วยคอนโวลูชันด้านกว้าง โดยประสิทธิภาพของ DeepLab-V3+ บนชุดข้อมูลทดสอบ PASCAL VOC 2012 มีค่า Mean IoU เท่ากับ 87.80%

สำหรับในงานวิทยานิพนธ์ชิ้นนี้จึงเสนอการปรับปรุงสถาปัตยกรรมของ DeepLab-V3+ ด้วยสถาปัตยกรรมใหม่ 2 ชนิด ได้แก่ DeepLab-V3-A1 ด้วย ResNet-101 และ DeepLab-V3-A1 ด้วย Xception

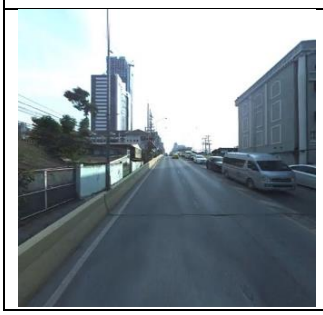
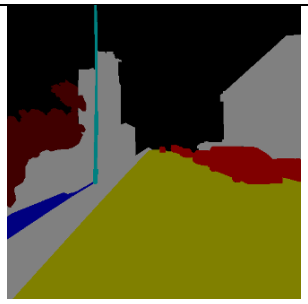
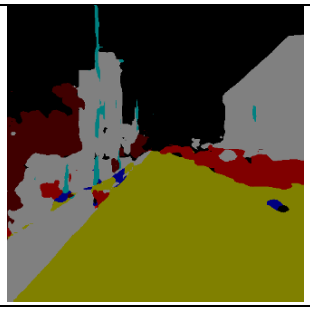
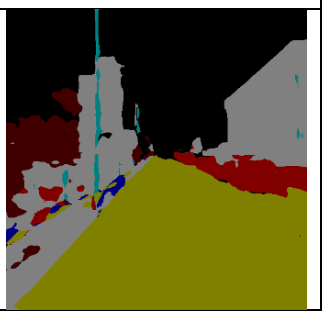
1. DeepLab-V3-A1 ด้วย ResNet-101 ดังแสดงในรูปที่ 36 เกิดจากการเพิ่มส่วนต่อขยายของ DeepLab-V3+ โดยการใช้เพียงตัวสกัดฟีเจอร์ด้วยการคัดแยกรูปภาพด้วย ResNet-101 ที่สกัดลักษณะเด่นจากรูปภาพป้อนเข้า และนำมาผ่านคอนโวลูชันแบบหลุมที่มีขนาดความกว้างแตกต่างกัน 4 ระดับ ได้แก่ 1, 6, 12, และ 18 โดยการทำ Padding Convolution สำหรับบางฟีเจอร์แมพที่มีขนาดเล็กเกินไป และฟีเจอร์แมพขั้นต้นถูกนำมาปรับมิติทางฝั่งเอนโคดเดอร์ด้วยการทำคอนโวลูชัน 1 x 1 พร้อมกับการทำอัพแซมปิงเพื่อเชื่อมต่อกับฟีเจอร์แมพระดับต่ำจาก ResNet-101 ในชั้นดีโคดเดอร์โดยการคอนแคต หลังจากที่นำฟีเจอร์แมพที่ผ่านการสกัดโดยคอนโวลูชันแบบหลุมที่มีขนาดความกว้างแตกต่างกัน 4 ระดับได้แก่ 1, 6, 12, และ 18 ถูกนำมาเชื่อมต่อกับฟีเจอร์แมพระดับต่ำจาก ResNet-101 และฟีเจอร์แมพทั้ง 2 จะถูกนำมาผ่านชั้นคอนโวลูชัน 3 x 3 ฟีเจอร์แมพที่ได้จากชั้นนี้จะถูกนำมาผ่านชั้นคอนโวลูชัน 1 x 1 ด้วยจำนวนชั้นกรองที่แตกต่างกัน เพื่อใช้ในการพิจารณาการอัพแซมปิงสำหรับการทำนายในแต่ละพิกเซลของภาพที่ประกอบไปด้วย 11 คลาส
2. DeepLab-V3-A1 ด้วย Xception ดังแสดงในรูปที่ 37 เกิดจากการปรับปรุงประสิทธิภาพการทำนายพิกเซลรายคลาสซึ่งคงไว้สถาปัตยกรรมดังเดิมด้วย DeepLab-V3-A1 เพียงเปลี่ยนแค่ตัวสกัดฟีเจอร์ด้วยการคัดแยกภาพจาก ResNet-101 เป็น Xception ที่มีประสิทธิภาพสูงบนชุดข้อมูล ImageNet และผลลัพธ์การทำนายระหว่าง DeepLab-V3-A1 ด้วย ResNet-101 กับ DeepLabV3-A1 ด้วย Xception บนชุดข้อมูลถนนกรุงเทพดังแสดงในรูปที่ 38



รูปที่ 36 วิธีการที่นำเสนอด้วยสถาปัตยกรรม DeepLab-V3-A1 ด้วย ResNet-101



รูปที่ 37 วิธีการที่นำเสนอด้วยสถาปัตยกรรม DeepLab-V3-A1 ด้วย Xception

รูปภาพนำเข้า	ผลเฉลย	DeepLab-V3-A1 ด้วย ResNet-101	DeepLab-V3-A1 ด้วย Xception
			

รูปที่ 38 วิธีการที่นำเสนอด้วยสถาปัตยกรรม DeepLab-V3-A1 ด้วย ResNet-101 และ DeepLab-V3-A1 ด้วย Xception

4.5 การฝึก

สำหรับการสอนภาพถ่ายจะถูกสุ่มตัด 512 x 512 พิกเซล ด้วยการป้อนชุดข้อมูลนำเข้าไปทีละแบช ซึ่งแต่ละแบชมีจำนวนรูปภาพเท่ากับ 8 ภาพ ซึ่งประกอบไปด้วยวัตถุจำนวน 11 คลาส โดยทำการฝึกจำนวน 300 รอบอิพอค (Epochs) ด้วยฟังก์ชันปรับค่าที่เหมาะสมที่สุดอัพเดทค่าน้ำหนักคือ RMSProp จากอัตราการเรียนรู้ทั้งทางฝั่งเอนโคดเดอร์และดีโคดเดอร์เท่ากับ 0.0001 โดยแต่ละรอบอิพอค (Epochs) ใช้เวลา 34.10 นาที สำหรับทั้ง DeepLab-V3-A1 ด้วย ResNet-101 และ DeepLab-V3-A1 ด้วย Xception และชั้นการแบ่งส่วนสำหรับทั้ง 2 สถาปัตยกรรมนี้คือ Softmax ที่ทำนายทุกพิเซลในภาพซึ่งประกอบมีทั้งหมด 11 คลาส สำหรับการทดลองทั้งหมดจะถูกประมวลผลด้วยภาษาไพธอนด้วยไลบารีเทนเซอร์โฟล์ (Tensorflow) และสเปคคอมพิวเตอร์ที่ใช้ในการทดลองคือ Intel Xeon Silver 4110 CPU@2.10GHz (8 Cores, 16 Threads per sockets), 128 GB of memory (RAM), with GPU: Nvidia Tesla V100 32GB x 2

4.6 การทดสอบ

โดยการทดสอบสถาปัตยกรรม DeepLab-V3-A1 Xception ที่ได้น้ำหนักที่ให้ค่า Mean IoU บนชุดข้อมูลตรวจสอบ ที่ดีสุดจะถูกนำไปใช้บนชุดข้อมูลทดสอบที่เตรียมไว้จำนวน 233 ภาพบทชุดข้อมูลถนนกรุงเทพมหานคร

บทที่ 5
การทดลองและผลการทดลอง

ในส่วนนี้เราจะทำการทดลองเพื่อยืนยันสมมติฐานว่าวิธีการของเราที่ถูกคิดค้นขึ้นนั้นมีประสิทธิภาพที่ดีไม่เพียงแต่ในชุดข้อมูลที่เรานำเสนอเท่านั้น โดยเราจะทำการทดลองทั้งหมดด้วยชุดข้อมูลทั้งหมด 3 ชุด โดยวิทยานิพนธ์นี้จะเน้นการทดลองชุดข้อมูลถนนกรุงเทพฯเป็นหลัก เราจะทำการทดลองวิธีที่เราคิดค้นได้แก่ DeepLab-V3-A1 ด้วย ResNet-101 และ DeepLab-V3-A1 ด้วย Xception เพื่อเปรียบเทียบกับวิธีการมาตรฐาน (Baseline) ทั้ง 6 วิธี ได้แก่ SegNet, UNet, PSPNet, Tiramisu, DeepLab-V3+ ด้วย ResNet-101 และ DeepLab-V3+ ด้วย Xception โดยเราจะคัดเลือกวิธีการมาตรฐานที่ดีที่สุดจากการพิจารณามาตรวัด Mean IoU ที่มีค่ามากที่สุดเป็นหลักจากการทดลองชุดข้อมูลถนนกรุงเทพฯ เพื่อนำวิธีมาตรฐานที่ดีที่สุดนำไปทดลองร่วมกับวิธีมาตรฐานต้นฉบับ ได้แก่ DeepLab-V3+ ด้วย ResNet-101 และ DeepLab-V3+ ด้วย Xception ในอีก 2 ชุดข้อมูลที่เหลือ ได้แก่ ชุดข้อมูล CamVid และ Cityscapes ตามลำดับ

ผลลัพธ์การทดลองบนชุดข้อมูลถนนกรุงเทพฯ พบว่าวิธี Tiramisu เป็นวิธีการมาตรฐานที่ดีที่สุด (Baseline) โดยมีประสิทธิภาพเท่ากับวิธีแรกของเราที่นำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 ในแง่ของมาตรวัด Mean IoU มีค่าเท่ากับ 57.64% ดังแสดงในตารางที่ 2 เราจะนำวิธี Tiramisu ไปทดลองร่วมกับวิธี DeepLab-V3+ โดยใช้โครงข่ายประสาทเทียมเชิงลึกที่แตกต่างกัน 2 วิธี ได้แก่ ResNet-101 และ Xception ในการทดลองถัดไปกับ 2 ชุดข้อมูลที่เหลือ ได้แก่ ชุดข้อมูล CamVid และ Cityscapes ซึ่งอยู่ในบทที่ 5.2 และ 5.3 ตามลำดับ

5.1 ผลการทดลองบนชุดข้อมูลถนนในกรุงเทพมหานคร

ตาราง 2 แสดงให้เห็นถึงผลลัพธ์การทดลองโดยรวมของวิธีที่เรานำเสนอได้แก่ DeepLab-V3-A1 ด้วย ResNet-101 และ Xception เมื่อเปรียบเทียบกับวิธีมาตรฐาน 6 วิธี ได้แก่ SegNet, UNet, PSPNet, Tiramisu, DeepLab-V3+ ด้วย ResNet-101 และ DeepLab-V3+ ด้วย Xception บนชุดข้อมูลถนนกรุงเทพฯ โดยวิธีที่เรานำเสนอคือ DeepLab-V3-A1 ด้วย Xception ให้ผลลัพธ์การทดลองที่ดีที่สุดในมาตรวัด Precision, Recall, และ F1 Score อีกทั้ง 3 หน่วยที่กล่าวมาข้างต้นมีค่าเกินกว่า 85% ซึ่งมีค่าเท่ากับ 87.44%, 86.08%, และ 85.93% ตามลำดับ และมีค่าความ

แม่นยำที่สูงที่สุดในคลาส Building เท่ากับ 88.27% ดังแสดงในตารางที่ 3 แต่อย่างไรก็ตามกลยุทธที่ดีที่สุดของเรา DeepLab-V3-A1 ด้วย Xception ให้ค่า Mean IoU ที่ต่ำกว่าวิธีแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 เล็กน้อยเพียง 0.03% ดังนั้นวิธีที่มีประสิทธิภาพที่สุดบนชุดข้อมูลถนนกรุงเทพฯ คือวิธี DeepLab-V3-A1 ด้วย ResNet-101 โดยการเลือกวิธีมาตรฐานที่ดีที่สุดจากชุดข้อมูลถนนกรุงเทพฯ เราจะพิจารณาที่ค่า Mean IoU เป็นหลัก เราพบ Tiramisu ที่ให้ค่า Mean IoU เท่ากันกับ DeepLab-V3-A1 ด้วย ResNet-101 ที่ 57.64% ดังแสดงในตารางที่ 3 ซึ่งเราจะนำวิธีการมาตรฐานที่ดีที่สุด Tiramisu ไปเปรียบเทียบกับวิธีที่นำเสนอตลอดการทดลองในบทที่ 5.1.1 และ 5.1.2 ตามลำดับ

5.1.1 DeepLab-V3-A1 ผลลัพธ์การทดลองจากการใช้โครงข่ายเชิงลึกด้วย ResNet-101 บนชุดข้อมูลถนนกรุงเทพฯ

ในส่วนนี้จะเปรียบเทียบประสิทธิภาพของวิธีแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 ที่เกิดจากการปรับปรุงวิธีมาตรฐานต้นตำรับ DeepLab-V3+ ด้วย Xception เราจะเห็นได้ว่ามีผลลัพธ์ที่ดีขึ้นในแง่ของประสิทธิภาพด้วยมาตรวัด Mean IoU ที่มากกว่า V3+ ด้วย Xception อย่างมีนัยสำคัญถึง 4.23% ดังแสดงในตารางที่ 2 อีกทั้งยังมีประสิทธิภาพที่ดีขึ้นในแง่ของหน่วยวัดผลที่เหลือ 3 หน่วย ซึ่งมีค่าเพิ่มขึ้นโดยรวมไม่ต่ำกว่า 2.00% ได้แก่ Precision, Recall, และ F1 Score เมื่อเปรียบเทียบวิธีการแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 กับวิธีการมาตรฐานต้นตำรับ DeepLab-V3+ ด้วย Xceptiom โดยวิธีของเรา DeepLab-V3-A1 ด้วย ResNet-101 มีค่า Precision, Recall, และ F1 เท่ากับ 87.36%, 85.97%, และ 85.84% ตามลำดับ แต่อย่างไรก็ตามประสิทธิภาพในแง่ของมาตรวัด Mean IoU ของวิธีแรกที่นำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 มีค่าเท่ากับวิธีมาตรฐานที่ดีที่สุด Tiramisu อยู่ที่ 57.64% ดังแสดงในตารางที่ 2

ในส่วนของการวิเคราะห์มาตรวัดความแม่นยำ เมื่อเราเปรียบเทียบวิธีแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 ให้ผลลัพธ์ที่ดีที่สุดในมาตรวัด Average accuracy โดยมีคลาสที่ได้ความแม่นยำมากที่สุดจากวิธี DeepLab-V3-A1 ด้วย ResNet-101 เป็นจำนวน 3 คลาส ได้แก่ Pole, Tree, and Misc มีค่าเท่ากับ 49.66%, 84.52%, และ 83.65% ตามลำดับ ดังแสดงในตารางที่ 3 ยิ่งไปกว่านั้น วิธีแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 มีค่าความแม่นยำในมาตรวัด Average accuracy มากกว่าวิธีการมาตรฐานต้นตำรับ DeepLab-V3+ ด้วย Xception

เกือบทุกคลาสยกเว้น คลาส Car ในทำนองเดียวกันการเปรียบเทียบประสิทธิภาพในมาตรวัด Accuracy วิธีแรกของเราที่นำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 เมื่อเทียบกับวิธีการมาตรฐานที่ดีที่สุด Tiramisu พบว่าวิธีแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 มีจำนวนคลาสที่มี Average accuracy สูงสุดมากกว่าวิธี Tiramisu เป็นจำนวน 1 คลาส

5.1.2 DeepLab-V3-A1 ผลลัพธ์การทดลองจากการใช้โครงข่ายเชิงลึกด้วย Xception บนชุดข้อมูลถนนกรุงเทพฯ

สำหรับผลการปรับปรุงโครงข่ายการเรียนรู้เชิงลึกเพื่อสกัดฟีเจอร์ของวิธีการที่เรานำเสนอวิธีแรก DeepLab-V3-A1 จาก ResNet-101 แทนที่ด้วย Xception พบว่าวิธีที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception มีผลลัพธ์ที่ดีกว่าในมาตรวัด Precision และ Recall ซึ่งมากกว่าเมื่อเทียบกับวิธี DeepLab-V3-A1 ด้วย ResNet-101 โดยประมาณ 0.10% ซึ่งวิธีการ DeepLab-V3-A1 ด้วย Xception มีค่า Precision และ Recall เท่ากับ 87.44% และ 86.08% ตามลำดับ ในทำนองเดียวกันวิธีที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception มีค่ามาตรวัด Precision และ Recall มากกว่าวิธีมาตรฐานที่ดีที่สุด Tiramisu อยู่ที่ 0.21% และ 0.51% ตามลำดับ เนื่องจาก DeepLab-V3-A1 ด้วย Xception มีค่ามาตรวัด Precision และ Recall ที่สูงที่สุดในการทดลองด้วยชุดข้อมูลถนนกรุงเทพฯ จึงส่งผลให้ค่ามาตรวัด F1 Score มีค่ามากที่สุดเท่ากับ 85.93% ดังแสดงในตารางที่ 2 แต่อย่างไรก็ตามก็มีผลลัพธ์ที่ไม่ดีจากการเปลี่ยน ResNet-101 ด้วย Xception ของ DeepLab-V3-A1 ซึ่งส่งผลให้ค่า Mean IoU ลดลงเล็กน้อยเพียง 0.03%

วิธีที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception มีประสิทธิภาพสูงที่สุดในมาตรวัดความแม่นยำ โดยมีคลาสที่ได้ความแม่นยำสูงที่สุดคือ คลาส Buiding มีค่าเท่ากับ 88.26% ดังแสดงในตารางที่ 3 และผลการเปรียบเทียบวิธี DeepLab-V3-A1 ด้วย Xception กับ วิธีการมาตรฐานด้วย Tiramisu พบว่ามีจำนวนคลาสที่ได้ Average accuracy สูงสุด น้อยกว่า Tiramisu เพียง 1 คลาส

ในส่วนของการวิเคราะห์กราฟการเรียนรู้ โดยมี 2 พีค ในกราฟมาตรวัด Mean IoU และ Average accuracy จากวิธีการที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception โดยมาตรวัดเหล่านี้ถูกวัดผล ในชุดข้อมูลตรวจสอบของถนนกรุงเทพฯ ซึ่งพีคเหล่านั้นถูกแสดงในกราฟเส้นดังแสดงในรูป 35(ก) และ 35(ข) ณ ตำแหน่งอิพอค ที่ 25 และ 231 ตามลำดับ

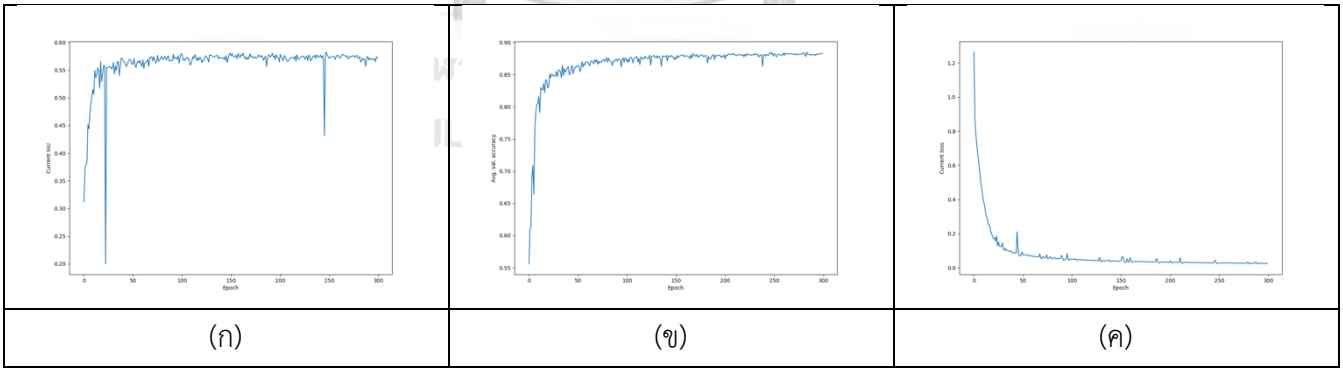
ในส่วนของการเปรียบเทียบประสิทธิภาพการทำนายบนชุดข้อมูลทดสอบของถนนกรุงเทพฯ ในเชิงคุณภาพของวิธีการที่เรานำเสนอทั้งสองวิธี DeepLab-V3-A1 ด้วย ResNet-101 และ DeepLab-V3-A1 ด้วย Xception ดังแสดงในรูปที่ 40(ง) และ 40(จ) ตามลำดับ กับวิธีการมาตรฐานกลุ่มแรกได้แก่ SegNet, UNet, และ PSPNet ดังแสดงในรูปที่ 40(ก), 40(ข), และ 40(ค) ตามลำดับ ประกอบกับวิธีการมาตรฐานในกลุ่มที่สอง ได้แก่ Tiramisu, DeepLab-V3+ ด้วย ResNet-101, และ DeepLab-V3+ ด้วย Xception ดังแสดงในรูปที่ 41(ก), 41(ข), และ 41(ค) ตามลำดับ

ตารางที่ 2 ผลการทดลองโดยรวมบนชุดข้อมูลทดสอบของถนนกรุงเทพฯ

	โมเดล	ตัวสกัดฟีเจอร์	Precision	Recall	F1 Score	Mean IoU
วิธีการมาตรฐาน	SegNet		82.22%	81.08%	79.99%	51.93%
	UNet		84.62%	83.05%	82.76%	52.14%
	PSPNet	ResNet-101	85.85%	84.16%	83.89%	55.20%
	Tiramisu	DenseNet-100	87.23%	85.57%	85.44%	**57.64%**
	DeepLab-V3+	ResNet-101	85.55%	82.45%	82.70%	50.30%
		Xception	85.58%	84.01%	83.86%	53.40%
วิธีการที่นำเสนอ	DeepLab-V3-A1	ResNet-101	87.36%	85.97%	85.84%	**57.64%**
		Xception	**87.44%**	**86.08%**	**85.93%**	57.61%

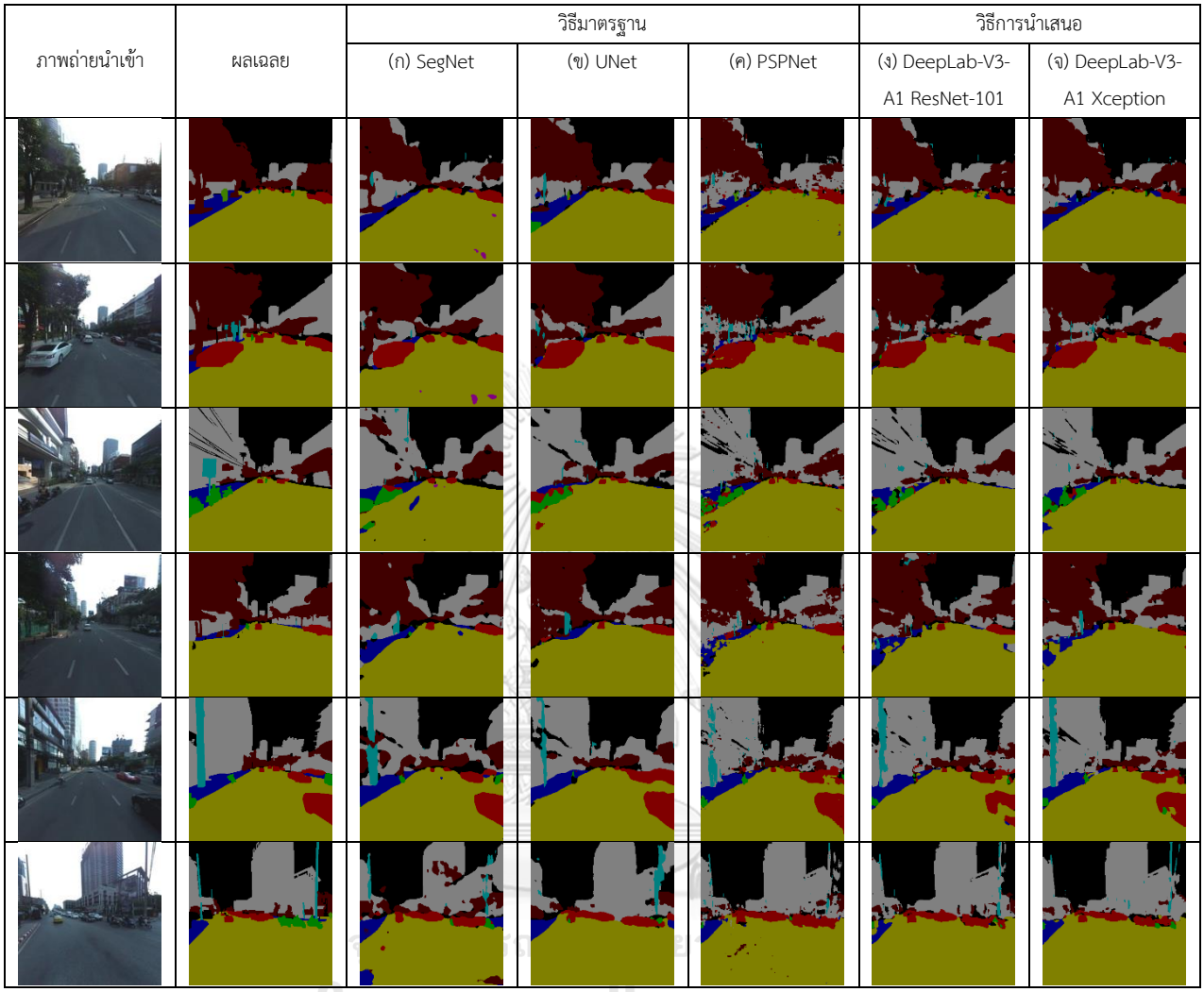
ตารางที่ 3 ผลการทดลองค่าความแม่นยำรายคลาสบนชุดข้อมูลทดสอบของถนนกรุงเทพฯ

	โมเดล	ตัวสกัดฟีเจอร์	Road	Footpath	Crosswalk	Building	Pole	Trash	Car	Motorcycle	Person	Tree	Misc
วิธีการมาตรฐาน	SegNet		93.66%	**62.34%**	98.36%	82.89%	43.16%	90.58%	75.26%	70.83%	85.04%	82.63%	78.40%
	UNet		94.02%	56.22%	98.32%	83.07%	31.89%	90.72%	73.13%	68.33%	85.24%	83.46%	79.69%
	PSPNet	ResNet-101	94.14%	56.62%	98.42%	86.64%	45.79%	90.63%	**84.81%**	**73.89%**	**85.93%**	84.95%	82.71%
	Tiramisu	DenseNet-100	94.39%	61.07%	**98.48%**	87.18%	49.39%	**91.19%**	76.86%	73.58%	85.84%	86.22%	80.76%
	DeepLab-V3+	ResNet-101	**95.08%**	39.73%	98.38%	87.29%	35.54%	90.66%	71.63%	68.36%	84.62%	79.44%	81.12%
		Xception	94.64%	54.99%	98.31%	83.16%	40.68%	90.57%	77.55%	67.63%	85.03%	84.00%	82.66%
วิธีการที่นำเสนอ	DeepLab-V3-A1	ResNet-101	94.53%	59.53%	98.39%	86.46%	**49.66%**	90.73%	75.85%	71.29%	85.44%	**87.52%**	**83.65%**
		Xception	94.37%	59.14%	98.40%	**88.27%**	49.31%	90.61%	76.56%	71.17%	85.32%	85.53%	83.20%

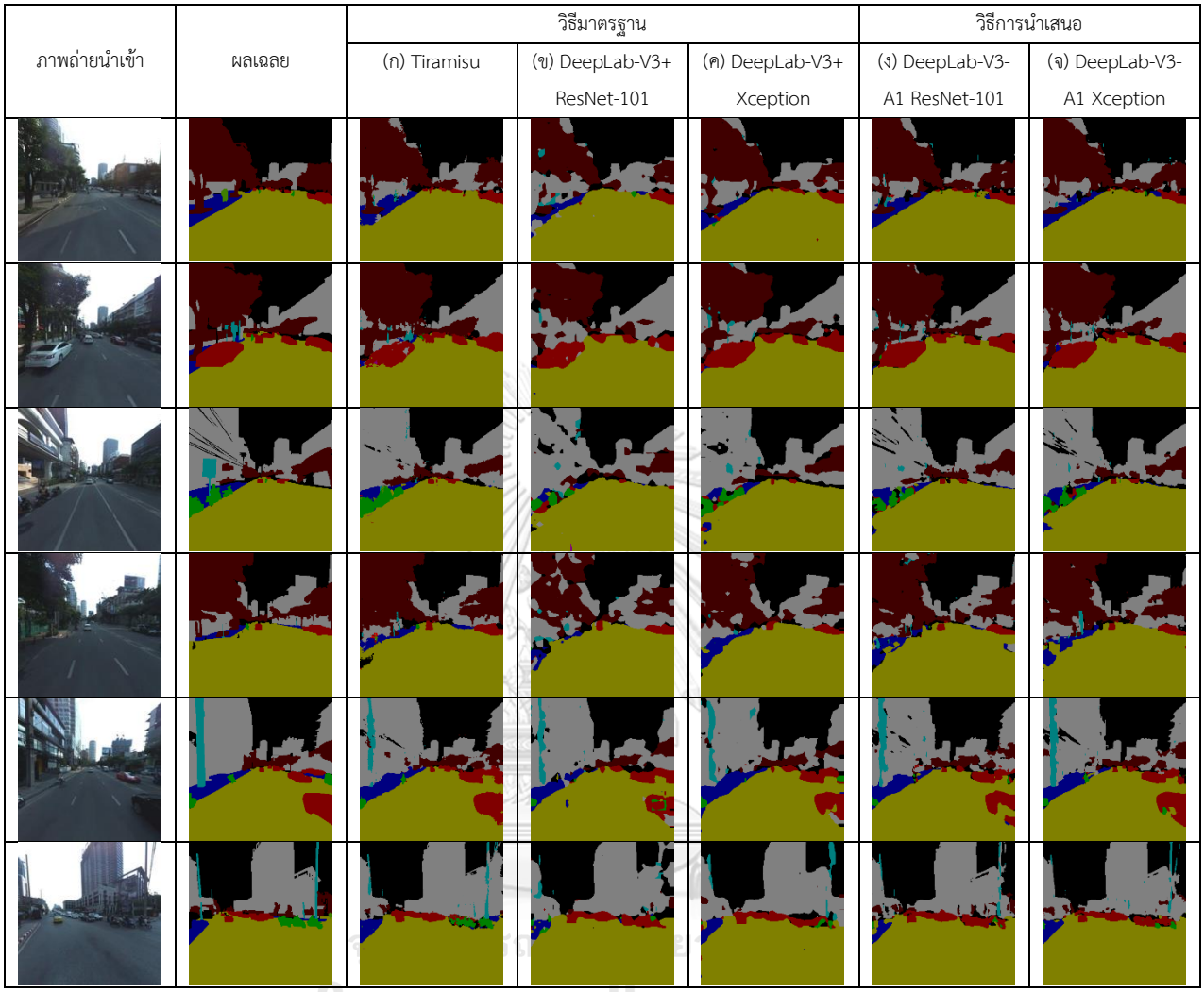


(ก) (ข) (ค)

รูปที่ 39 กราฟการเรียนรู้ DeepLab-V3-A1 ด้วย Xception บนชุดข้อมูลตรวจสอบของถนนกรุงเทพฯ ที่ฝึกจำนวน 300 รอบอิพอค (ก) หน่วย Mean IoU (ข) หน่วย Average accuracy (ค) กราฟค่าความสูญเสีย



รูปที่ 40 ผลลัพธ์ของการทำนายบนชุดข้อมูลทดสอบของถนนกรุงเทพฯ วิธีการมาตรฐาน (ก) SegNet (ข) UNet (ค) PSPNet เมื่อเทียบกับวิธีการที่นำเสนอ (ง) DeepLab-V3-A1 ด้วย ResNet-101 (จ) DeepLab-V3-A1 ด้วย Xception



รูปที่ 41 ผลลัพธ์ของการทำนายบนชุดข้อมูลทดสอบของถนนกรุงเทพฯ วิธีการมาตรฐาน (ก) Tiramisu (ข) DeepLab-V3+ ด้วย ResNet-101 (ค) DeepLab-V3+ ด้วย Xception เมื่อเทียบกับวิธีการที่นำเสนอ (ง) DeepLab-V3-A1 ด้วย ResNet-101 (จ) DeepLab-V3-A1 ด้วย Xception

5.2 ผลการทดลองในชุดข้อมูล CamVid

ชุดข้อมูล CamVid เป็นชุดข้อมูลที่ได้รับความนิยมในการนำโมเดลการแบ่งส่วนภาพเชิงความหมายมาทดสอบประสิทธิภาพของสถาปัตยกรรม โดยตารางที่ 4 แสดงให้เห็นถึงผลลัพธ์การทดลองโดยรวมในชุดข้อมูล CamVid ที่เปรียบเทียบทั้งวิธีการที่นำเสนอด้วย DeepLab-V3-A1 ด้วย ResNet-101 และ Xception รวมถึงวิธีการมาตรฐานที่ดีที่สุดในชุดข้อมูลถนนกรุงเทพฯ ในตารางที่ 2 คือ วิธีมาตรฐาน Tiramisu และ วิธีมาตรฐานต้นฉบับ DeepLab-V3+ ทั้ง ResNet-101 และ Xception โดยกลยุทธ์ที่ดีที่สุดของเราคือวิธีการที่นำเสนอ DeepLab-V3-A1 ด้วย Xception มีประสิทธิภาพมากที่สุดโดยพิจารณาจากมาตรวัด Mean IoU ที่มีค่าสูงที่สุดเท่ากับ 53.09% ดังแสดงในตารางที่ 4 ประกอบกับมีค่าในมาตรวัด 3 หน่วยที่เหลือมีค่าเกินกว่า 86.00% ได้แก่ Precision, Recall, และ F1 ซึ่งมีค่าเท่ากับ 87.56%, 86.12%, และ 86.00% ตามลำดับ วิธีของเรา DeepLab-V3-A1 ด้วย Xception มีประสิทธิภาพในแง่ของมาตรวัดความแม่นยำสูงที่สุด ถึง 4 คลาส ได้แก่ คลาส Sidewalk, Fence, Bicyclist, และ Pedestrain มีค่าเท่ากับ 86.08%, 85.60%, 85.80%, และ 72.04% ตามลำดับ ดังแสดงในตารางที่ 5 ยิ่งไปกว่านั้นกลยุทธ์ที่ดีที่สุดของเรา DeepLab-V3-A1 ด้วย Xception เมื่อเปรียบเทียบวิธีแรกที่นำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 พบว่ามีประสิทธิภาพในแง่ของมาตรวัด Mean IoU มากกว่า 0.19% ในทำนองเดียวกันการเปรียบเทียบวิธีการนำเสนอ DeepLab-V3-A1 ด้วย Xception กับ วิธีมาตรฐานที่ดีที่สุด Tiramisu พบว่ามีประสิทธิภาพในแง่ของ Mean IoU มากกว่าอย่างมีนัยสำคัญ ถึง 3.49% ดังแสดงในตารางที่ 4 ซึ่งเราจะนำวิธีการนำเสนอไปเปรียบเทียบกับวิธีการมาตรฐานตลอดตารางการทดลองในบทที่ 5.2.1 และ 5.2.2 ตามลำดับ

5.2.1 ผลลัพธ์การทดลอง DeepLab-V3-A1 จากการใช้โครงข่ายเชิงลึกด้วย ResNet-101 บนชุดข้อมูล CamVid

เราเปรียบเทียบประสิทธิภาพวิธีแรกที่นำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 ที่เกิดจากการปรับปรุงวิธีมาตรฐานต้นฉบับ DeepLab-V3+ ด้วย Xception มีพัฒนาการที่น่าประทับใจเมื่อเปรียบเทียบวิธีแรกที่นำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 กับวิธีการมาตรฐานต้นฉบับ DeepLab-V3+ ด้วย Xception ในแง่ของมาตรวัด Mean IoU ซึ่งมีค่าเพิ่มขึ้นอย่างมีนัยสำคัญถึง 9.52% อีกทั้งหน่วยวัดผลที่เหลือ ได้แก่ Precision, Recall, และ F1 มีค่าเพิ่มขึ้นโดยรวมถึง 3.41%

ดังแสดงในตารางที่ 4 เนื่องจาก DeepLab-V3-A1 ด้วย ResNet-101 มีประสิทธิภาพจากมาตรวัด Precision และ Recall ที่สูงที่สุดเท่ากับ 88.05% และ 86.20% ตามลำดับ จึงส่งผลให้ค่ามาตรวัด F1 Score มีค่ามากที่สุดเท่ากับ 86.15 % ดังแสดงในตารางที่ 4 ยิ่งไปกว่านั้นการเปรียบเทียบวิธีแรกของเรา DeepLab-V3-A1 ด้วย ResNet-101 เมื่อเทียบกับวิธีมาตรฐานที่ดีที่สุด Tiramisu พบว่าวิธีการของเรา DeepLab-V3-A1 ด้วย ResNet-101 มีประสิทธิภาพในแง่ของมาตรวัด Mean IoU ที่สูงกว่าอย่างมีนัยสำคัญถึง 3.30%

ในส่วนของการวิเคราะห์มาตรวัดความแม่นยำ DeepLab-V3-A1 ด้วย ResNet-101 ให้ผลลัพธ์ที่ดีที่สุดในหน่วยของ Average accuracy ดังแสดงในตารางที่ 5 โดยมีคลาสที่ได้ค่าความแม่นยำสูงสุดโดยวิธี DeepLab-V3-A1 ด้วย ResNet-101 เป็นจำนวน 4 คลาส ได้แก่ Building, Column_Pole, Car, และ MotorcycleScooter ซึ่งมีค่าเท่ากับ 89.46%, 50.33%, 84.59%, และ 97.68% ตามลำดับ ยิ่งไปกว่านั้นเมื่อเปรียบเทียบวิธีแรกที่นำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 มีค่าความแม่นยำมากกว่าวิธีมาตรฐานต้นตำรับ DeepLab-V3+ ด้วย Xception มากกว่าทุกคลาส ดังแสดงในตารางที่ 5 เช่นเดียวกันกับการเปรียบเทียบวิธีแรกที่นำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 กับวิธีมาตรฐานที่ดีที่สุด Tiramisu พบว่าวิธีแรกของเรามีจำนวนคลาสที่มีค่าสูงสุดด้วยมาตรวัดนี้มากกว่าวิธี Tiramisu เป็นจำนวน 2 คลาส

5.2.2 ผลลัพธ์การทดลอง DeepLab-V3-A1 จากการใช้โครงข่ายเชิงลึกด้วย Xception บนชุดข้อมูล CamVid

สำหรับผลการปรับปรุงโครงข่ายการเรียนรู้เชิงลึกของวิธีการที่เราคิดค้น DeepLab-V3-A1 จาก ResNet-101 แทนที่ด้วย Xception พบว่าวิธีการที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception มีผลลัพธ์ที่ดีจากการพัฒนาโครงข่ายการเรียนรู้เชิงลึก เมื่อเราเปรียบเทียบ DeepLab-V3-A1 จาก ResNet-101 กับ DeepLab-V3-A1 ด้วย Xception เราพบว่าประสิทธิภาพในมาตรวัด Mean IoU มีค่าเพิ่มขึ้น 0.19% แต่ก็ส่งผลให้มาตรวัด 3 หน่วยที่เหลือได้แก่ Precision, Recall, F1-Score มีค่าน้อยลงเมื่อเปรียบเทียบวิธีการ DeepLab-V3-A1 ด้วย ResNet-101 กับ DeepLab-V3-A1 ด้วย Xception เพียงเล็กน้อยประมาณ 0.10% ในทำนองเดียวกันการเปรียบเทียบวิธีที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception กับวิธีการมาตรฐานที่ดีที่สุด Tiramisu โดยวิธีของเรา DeepLab-V3-A1 ด้วย Xception มีประสิทธิภาพมากกว่าใน 3 มาตรวัดที่เหลือได้แก่ Precision,

Recall, และ F1-Score ที่มากกว่าวิธีการมาตรฐานที่ดีที่สุด Tiramisu อยู่ที่ 0.21%, 0.66%, และ 0.62% ตามลำดับ ยิ่งไปกว่านั้นการเปรียบเทียบวิธีของเรา DeepLab-V3-A1 ด้วย Xception เมื่อเทียบกับวิธีการมาตรฐานที่ดีที่สุด Tiramisu พบว่าวิธีของเรา DeepLab-V3-A1 ด้วย Xception มีประสิทธิภาพในแง่ของมาตรวัด Mean IoU สูงกว่าอย่างมีนัยสำคัญที่ 3.49% ดังแสดงในตารางที่ 4

วิธีที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception มีคลาสที่มีประสิทธิภาพสูงสุดในมาตรวัด Average accuracy เป็นจำนวน 4 คลาส ได้แก่ Sidewalk, Fence, Bicyclist, และ Pedestrian มีค่าเท่ากับ 86.08%, 85.60%, 85.80%, และ 72.04% ตามลำดับ ดังแสดงในตารางที่ 5 โดยผลการเปรียบเทียบวิธีที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception กับ วิธีแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 พบว่ามีจำนวนคลาสที่ได้ Average accuracy สูงสุดมีจำนวน 4 คลาสเท่ากัน และยิ่งไปกว่านั้นผลการเปรียบเทียบวิธีนี้กับวิธีมาตรฐาน Tiramisu พบว่าวิธีของเรา DeepLab-V3-A1 ด้วย Xception มีจำนวนคลาสที่ได้มาตรวัดนี้สูงสุด มากกว่า Tiramisu ถึง 2 คลาส

ในส่วนของการวิเคราะห์กราฟการเรียนรู้ของวิธีที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception บนชุดข้อมูล CamVid โดยมีกราฟเส้นที่แสดงถึงประสิทธิภาพในหน่วย Mean IoU, Average accuracy, และ กราฟค่าความสูญเสีย ดังแสดงในรูปที่ 42(ก), 42(ข) และ 42(ค) ตามลำดับ เราไม่พบความผิดปกติจากกราฟการเรียนรู้ทั้ง 3 แบบ ตลอดการฝึกจำนวน 300 รอบอิพอค

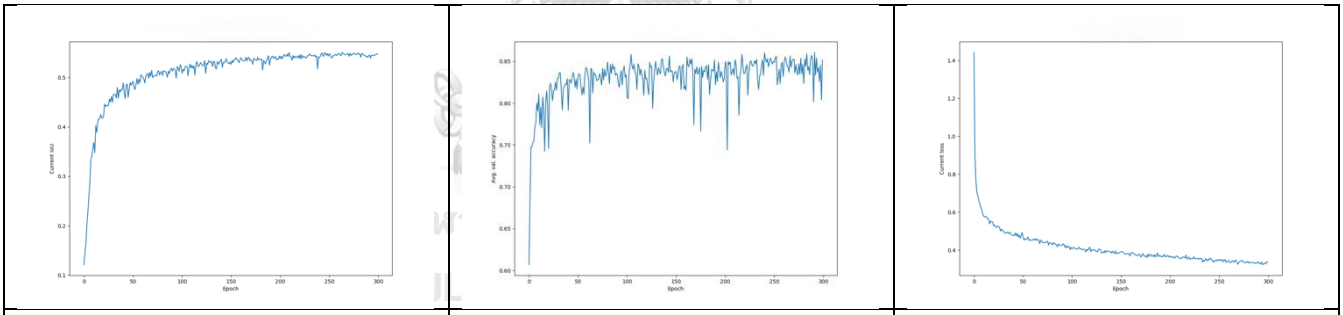
ในส่วนของการเปรียบเทียบประสิทธิภาพการทำนายบนชุดข้อมูลทดสอบของ CamVid ในเชิงคุณภาพด้วยวิธีการที่เรานำเสนอทั้งสองด้วย DeepLab-V3-A1 ด้วย ResNet-101 และ Xception ถูกแสดงในรูปที่ 43(ง) และ 43(จ) และวิธีการมาตรฐานได้แก่ DeepLab-V3+ ด้วย ResNet-101 และ DeepLab-V3+ ด้วย Xception ดังแสดงในรูปที่ 43(ก), 43(ข), และ 43(ค) ตามลำดับ

ตารางที่ 4 ผลการทดลองโดยรวมบนชุดข้อมูลทดสอบของ CamVid

	โมเดล	ตัวสกัดฟีเจอร์	Precision	Recall	F1 Score	Mean IoU
วิธีการมาตรฐาน	Tiramisu	DenseNet-100	87.36%	85.46%	85.38%	49.60%
	DeepLab-V3+	ResNet-101	83.96%	81.98%	81.47%	43.71%
		Xception	84.64%	82.31%	82.12%	43.38%
วิธีการที่นำเสนอ	DeepLab-V3-A1	ResNet-101	**88.05%**	**86.20%**	**86.15%**	52.90%
		Xception	87.56%	86.12%	86.00%	**53.09%**

ตารางที่ 5 ผลการทดลองค่าความแม่นยำรายคลาสบนชุดข้อมูลทดสอบของ CamVid

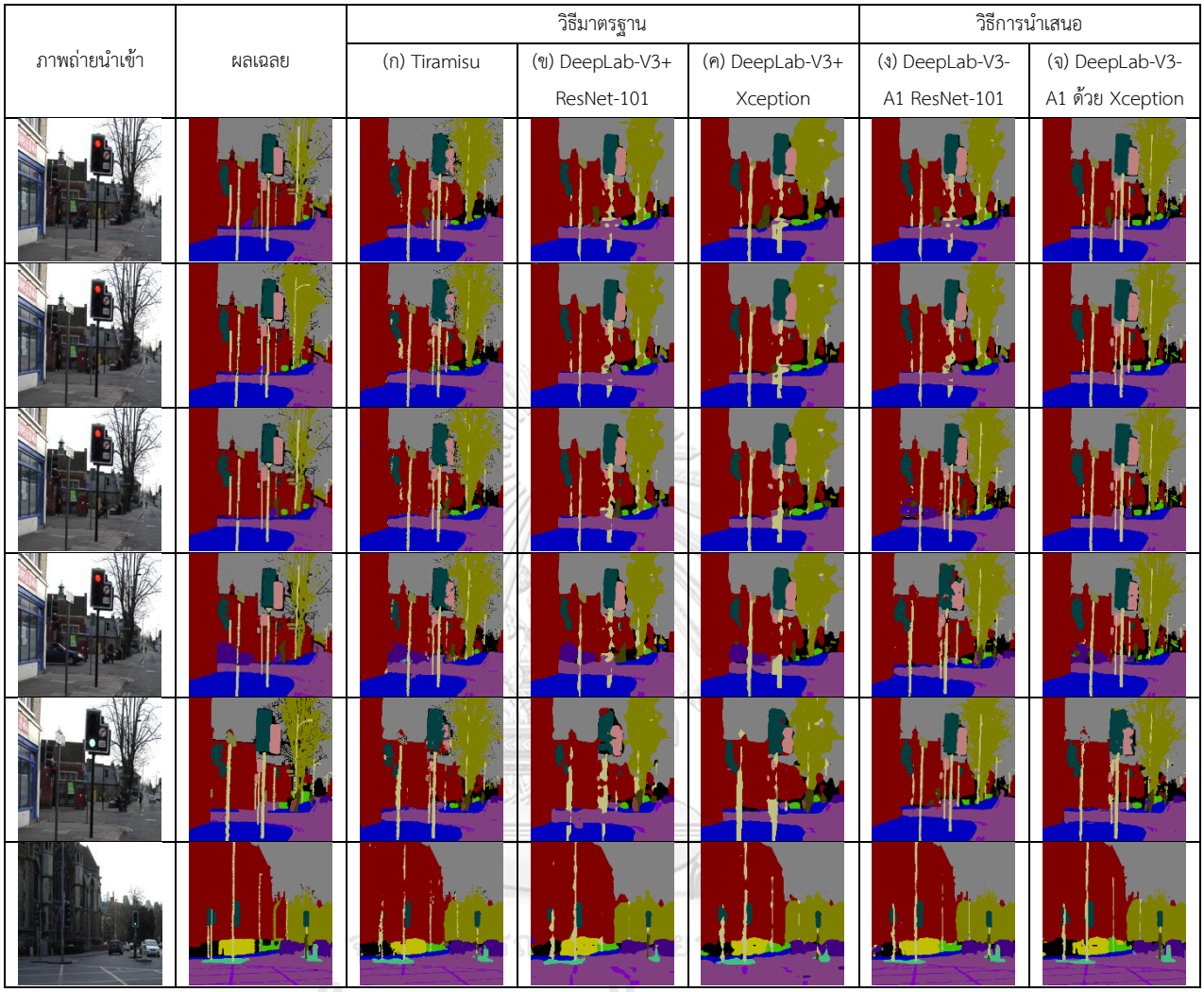
	โมเดล	ตัวสกัดฟีเจอร์	Road	Sidewalk	Fence	Building	Column_Pole	Car	Motorcycle Scooter	Bicyclist	Pedestrian	Tree	Sky
วิธีการมาตรฐาน	Tiramisu	DenseNet-100	91.61%	83.00%	82.90%	86.40%	41.52%	79.17%	97.62%	81.02%	63.12%	**85.26%**	**96.45%**
	DeepLab-V3+	ResNet-101	**92.12%**	76.55%	83.55%	87.56%	31.40%	74.55%	97.62%	80.49%	66.07%	80.72%	94.36%
		Xception	88.40%	79.27%	81.45%	86.74%	36.68%	76.23%	97.65%	80.41%	66.11%	82.01%	93.60%
วิธีการที่นำเสนอ	DeepLab-V3-A1	ResNet-101	90.48%	84.28%	84.76%	**89.46%**	**50.33%**	**84.59%**	**97.68%**	84.18%	69.07%	83.68%	95.47%
		Xception	92.10%	**86.08%**	**85.60%**	87.14%	50.26%	82.93%	97.62%	**85.80%**	**72.04%**	84.66%	95.38%



(ก) (ข) (ค)

รูปที่ 42 กราฟการเรียนรู้ DeepLab-V3-A1 ด้วย Xception บนชุดข้อมูลตรวจสอบของ CamVid

(ก) ในหน่วย Mean IoU (ข) ในหน่วย Average accuracy (ค) กราฟค่าความสูญเสีย



รูปที่ 43 ผลลัพธ์ของการทำนายบนชุดข้อมูลทดสอบของ CamVid วิธีการมาตรฐาน (ก) Tiramisu (ข) DeepLab-V3+ ด้วย ResNet-101 (ค) DeepLab-V3+ ด้วย Xception เมื่อเทียบกับวิธีการที่นำเสนอ (ง) DeepLab-V3-A1 ด้วย ResNet-101 (จ) DeepLab-V3-A1 ด้วย Xception

5.3 ผลการทดลองในชุดข้อมูล Cityscapes

ชุดข้อมูล Cityscapes เป็นชุดข้อมูลที่ได้รับความนิยมเป็นอย่างมากสำหรับการทดสอบประสิทธิภาพสถาปัตยกรรมการแบ่งส่วนภาพเชิงความหมาย เพื่อนำไปประยุกต์ใช้ในระบบขับเคลื่อนอัตโนมัติ ในการทดลองด้วยชุดข้อมูล Cityscapes เราจำลองชุดข้อมูลทดสอบใน Cityscapes โดยแบ่งรูปจากชุดข้อมูลตรวจสอบ (Validation) เป็นจำนวน 480 ภาพ และอีก 20 ภาพที่เหลือเป็นถูกกำหนดให้เป็นชุดข้อมูลตรวจสอบ (Validation) ในการทดลองในบทที่ 5.3

โดยตาราง 6 แสดงให้เห็นถึงผลลัพธ์การทดลองโดยรวมในชุดข้อมูล Cityscapes ที่เราเปรียบเทียบวิธีการที่เรานำเสนอ 2 วิธี ได้แก่ DeepLab-V3-A1 ด้วย ResNet-101 และ Xception กับวิธีมาตรฐาน (Baselines) ได้แก่วิธีการมาตรฐานที่ดีที่สุดในชุดข้อมูลถนนกรุงเทพฯ ดังแสดงในตารางที่ 2 คือ วิธีมาตรฐาน Tiramisu ร่วมกับวิธีมาตรฐานต้นฉบับ DeepLab-V3+ ด้วย ResNet และ Xception โดยวิธีที่เรานำเสนอด้วย DeepLab-V3-A1 ด้วย Xception มีประสิทธิภาพสูงในมาตรวัดเหล่านี้ ได้แก่ Precision, Recall, และ F1 โดยมาตรวัด 3 หน่วยที่กล่าวมาข้างต้นมีค่าเกินกว่า 80% ซึ่งมีค่าเท่ากับ 85.76%, 80.30%, และ 81.62% ตามลำดับ แต่อย่างไรก็ตามวิธี DeepLab-V3-A1 ด้วย Xception มีประสิทธิภาพที่ต่ำกว่าวิธีแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 เล็กน้อยที่ 0.94% ในแง่ของมาตรวัด Mean IoU ดังนั้นวิธีการแรกของเรา DeepLab-V3-A1 ด้วย ResNet-101 เป็นวิธีที่ดีที่สุดบนชุดข้อมูล Cityscapes โดยวิธีการที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception มีคลาสที่ได้มีประสิทธิภาพสูงสุดในมาตรวัด Average accuracy ถึง 3 คลาส ได้แก่ Road, Person, และ Sky มีค่าเท่ากับ 81.91%, 83.09%, 94.79% ตามลำดับ ดังแสดงในตารางที่ 7 อีกทั้งวิธีนี้มีประสิทธิภาพในแง่ของ Mean IoU ที่มากกว่า เมื่อเทียบกับวิธีการมาตรฐานที่ดีที่สุด Tiramisu อยู่ที่ 0.65% ซึ่งเราจะนำวิธีการที่เรานำเสนอทั้ง 2 แบบ ไปเปรียบเทียบประสิทธิภาพกับวิธีการมาตรฐานทั้ง 3 วิธี ด้วยชุดข้อมูลทดสอบและชุดข้อมูลตรวจสอบตามที่ได้กล่าวไว้ข้างต้นตลอดการทดลองในบทที่ 5.3.1 และ 5.3.2 ตามลำดับ

5.3.1 ผลลัพธ์การทดลอง DeepLab-V3-A1 จากการใช้โครงข่ายเชิงลึกด้วย ResNet-101 บนชุดข้อมูล Cityscapes

เราเปรียบเทียบประสิทธิภาพของวิธีนำเสนอด้วยวิธีแรก DeepLab-V3-A1 ด้วย ResNet-101 ที่เกิดจากการปรับปรุงวิธีการมาตรฐานต้นตำรับ DeepLab-V3+ ด้วย Xception เราพบว่ามี

พัฒนาการที่น่าประทับใจที่เกิดหลังการปรับปรุง โดยเฉพาะมาตรวัด Mean IoU ของวิธีแรกที่นำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 เมื่อเปรียบเทียบกับวิธี DeepLab-V3+ ด้วย Xception มีประสิทธิภาพเพิ่มขึ้นอย่างมีนัยสำคัญถึง 3.24% ดังแสดงในตารางที่ 6 อีกทั้งวิธีการแรกของเรา DeepLab-V3-A1 ด้วย ResNet-101 มีประสิทธิภาพมากกว่า DeepLab-V3+ ด้วย Xception ไม่ต่ำกว่า 4.90% ในมาตรวัดที่เหลือได้แก่ Precision, Recall, และ F1 Score ในทำนองเดียวกันการเปรียบเทียบประสิทธิภาพในมาตรวัด Mean IoU ระหว่างวิธีแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 กับวิธีมาตรฐานที่ดีที่สุด Tiramisu เราพบว่าวิธีของเรา DeepLab-V3-A1 มีประสิทธิภาพในแง่ของมาตรวัดผล Mean IoU ที่มากกว่าวิธีมาตรฐานที่ดีที่สุด Tiramisu อยู่ที่ 1.59% ดังแสดงในตารางที่ 6

ในส่วนของการวิเคราะห์มาตรวัดความแม่นยำของวิธีแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 มีคลาสที่มีมาตรวัดความแม่นยำสูงที่สุดได้แก่ คลาส Sidewalk, Pole, และ Vegetation มีค่าเท่ากับ 76.21%, 63.75%, และ 92.41% ตามลำดับ ดังแสดงในตารางที่ 7 เช่นเดียวกันกับการเปรียบเทียบวิธีแรกที่เรานำเสนอ DeepLab-V3-A1 ด้วย ResNet-101 กับวิธีมาตรฐาน Tiramisu เราพบว่าวิธีแรกของเรา DeepLab-V3-A1 ด้วย ResNet-101 มีจำนวนคลาสที่มีประสิทธิภาพจากมาตรวัด Average accuracy สูงสุดมากกว่าวิธีมาตรฐาน Tiramisu เป็นจำนวน 1 คลาส

5.3.2 ผลลัพธ์การทดลอง DeepLab-V3-A1 จากการใช้โครงข่ายเชิงลึกด้วย Xception บนชุดข้อมูล Cityscapes

สำหรับผลการปรับปรุงโครงข่ายประสาทเทียมสกัดฟีเจอร์ของ DeepLab-V3-A1 จาก ResNet-101 ที่ถูกแทนที่ด้วย Xception โดยการปรับปรุงตัวสกัดฟีเจอร์ส่งผลเสียต่อประสิทธิภาพในแง่ของมาตรวัด Precision และ Recall เมื่อเปรียบเทียบ DeepLab-V3-A1 ด้วย Xception กับ DeepLab-V3-A1 ด้วย ResNet-101 พบว่ามีค่า Precision และ Recall ลดลง 0.23% และ 0.57% ตามลำดับ ในทำนองเดียวกันการเปรียบเทียบวิธีที่เรานำเสนอ DeepLab-V3-A1 ด้วย Xception กับวิธีมาตรฐาน Tiramisu พบว่าวิธีของเรา DeepLab-V3-A1 ด้วย Xception มีค่า Precision และ Recall ที่น้อยกว่า Tiramisu อยู่ที่ 2.93% และ 1.74% ตามลำดับ เราพบและมีผลกระทบเชิงลบ

จากการเปลี่ยน ResNet-101 ที่ถูกแทนที่ด้วย Xception ของ DeepLab-V3-A1 ในชุดข้อมูล Cityscapes ซึ่งส่งผลให้มาตรวัด Mean IoU ลดลงเล็กน้อยเพียง 0.03% ดังแสดงในตารางที่ 6

วิธีที่เรานำเสนอด้วย DeepLab-V3-A1 ด้วย Xception มีประสิทธิภาพสูงในแง่ของมาตรวัด Average accuracy โดยมีจำนวนคลาสที่มีประสิทธิภาพสูงสุด ได้แก่ Road, Person, Sky มีค่าเท่ากับ 81.91%, 83.09%, 94.79% ตามลำดับ ดังแสดงในตารางที่ 7 และผลการเปรียบเทียบวิธีนี้กับวิธีมาตรฐาน Tiramisu พบว่ามีวิธีของเรา DeepLab-V3-A1 ด้วย Xception มีจำนวนคลาสที่ได้ Average accuracy สูงสุด นั้นมากกว่า Tiramisu 1 คลาส

ในส่วนของการวิเคราะห์กราฟการเรียนรู้ของวิธีนำเสนอ DeepLab-V3-A1 ด้วย Xception เราพบว่ามีช่วงการแกว่งตัวของกราฟ Average accuracy ขนาดใหญ่ ดังแสดงในรูปที่ 44(ข)

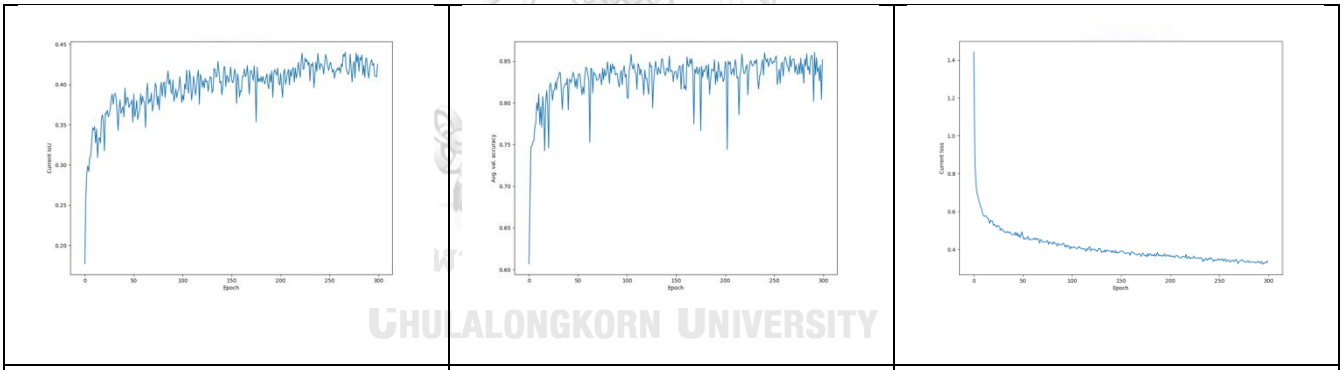
ในส่วนของเราเปรียบเทียบประสิทธิภาพการทำนายบนชุดข้อมูลทดสอบของ Cityscapes ในเชิงคุณภาพของวิธีการที่เรานำเสนอทั้งสองวิธี ได้แก่ DeepLab-V3-A1 ด้วย ResNet-101 และ Xception ถูกแสดงในรูปที่ 45(ง) และ 45(จ) และวิธีมาตรฐาน ได้แก่ Tiramisu, DeepLab-V3+ ด้วย ResNet-101, และ DeepLab-V3+ ด้วย Xception ดังแสดงในรูปที่ 45(ก), 45(ข), และ 45(ค) ตามลำดับ

ตารางที่ 6 ผลการทดลองโดยรวมบนชุดข้อมูลทดสอบของ Cityscapes

	โมเดล	ตัวสกัดฟีเจอร์	Precision	Recall	F1 Score	Mean IoU
วิธีการมาตรฐาน	Tiramisu	DenseNet-100	**88.70%**	**82.04%**	**84.20%**	39.99%
	DeepLab-V3+	ResNet-101	85.03%	79.98%	81.33%	38.35%
		Xception	77.13%	75.97%	75.14%	38.34%
วิธีการที่นำเสนอ	DeepLab-V3-A1	ResNet-101	86.00%	80.87%	82.22%	**41.58%**
		Xception	85.76%	80.30%	81.62%	40.64%

ตารางที่ 7 ผลการทดลองค่าความแม่นยำรายคลาสบนชุดข้อมูลทดสอบของ Cityscapes

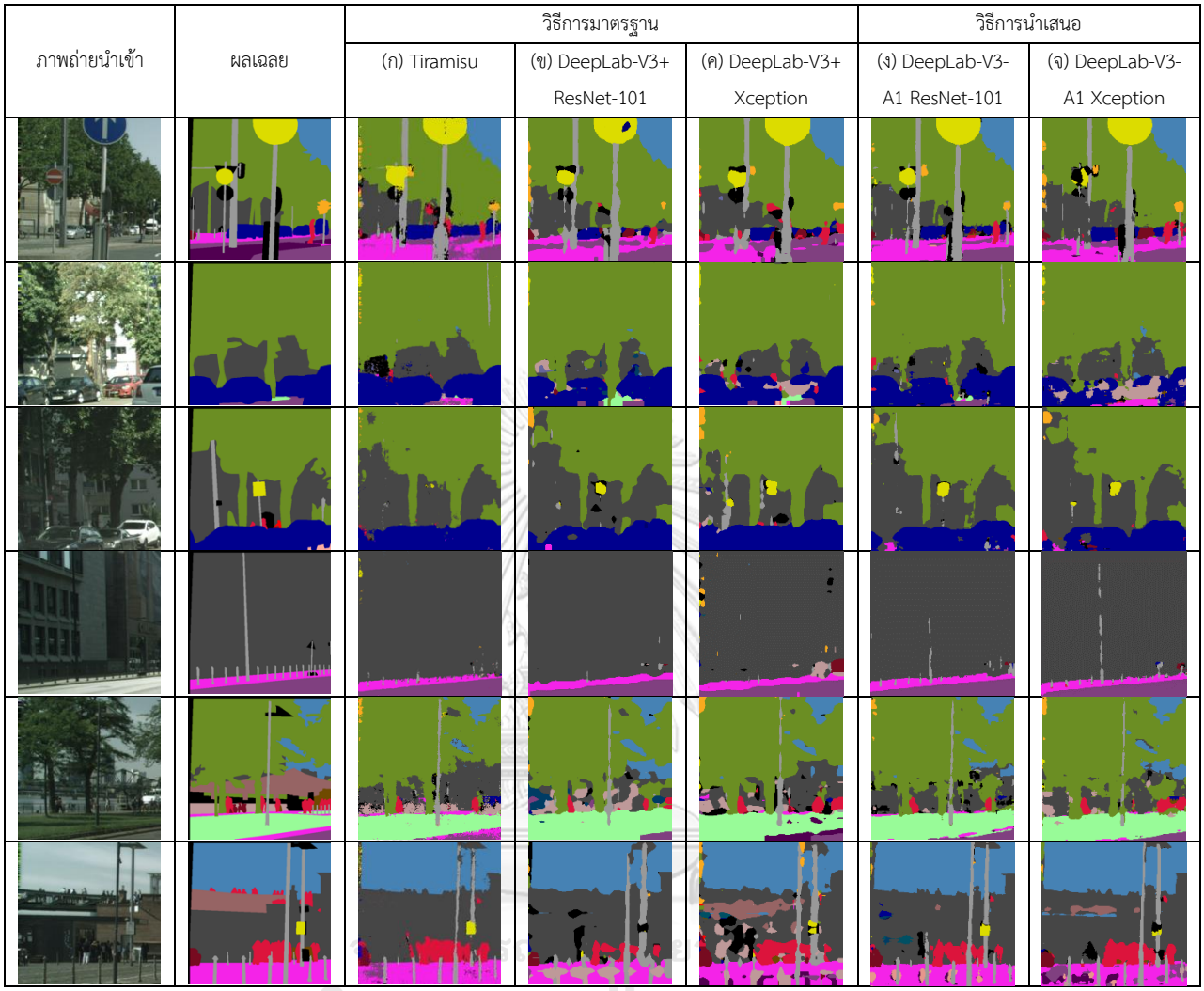
	โมเดล	ตัวสกัดฟีเจอร์	Road	Sidewalk	Fence	Building	Pole	Truck	Car	Motorcycle	Person	Vegetation	Sky
วิธีการมาตรฐาน	Tiramisu	DenseNet-100	79.08%	74.38%	87.89%	**88.37%**	51.23%	97.75%	**88.64%**	96.93%	79.32%	89.46%	94.31%
	DeepLab-V3+	ResNet-101	76.24%	71.52%	86.88%	82.52%	48.22%	97.79%	83.98%	97.00%	78.51%	88.64%	94.43%
		Xception	77.15%	70.37%	**91.15%**	72.11%	53.86%	**97.93%**	86.46%	**97.28%**	82.45%	92.20%	93.43%
วิธีการที่นำเสนอ	DeepLab-V3-A1	ResNet-101	79.79%	**76.21%**	85.93%	83.83%	**63.75%**	97.76%	86.59%	96.93%	80.91%	**92.41%**	93.98%
		Xception	**81.91%**	74.06%	86.40%	88.22%	53.66%	97.85%	83.14%	97.09%	**83.09%**	89.05%	**94.79%**



(ก) (ข) (ค)

รูปที่ 44 กราฟการเรียนรู้ DeepLab-V3-A1 ด้วย Xception บนชุดข้อมูลตรวจสอบของ Cityscapes

(ก) ในหน่วย Mean IoU (ข) ในหน่วย Average accuracy (ค) กราฟค่าความสูญเสีย



รูปที่ 45 ผลลัพธ์ของการทำนายบนชุดข้อมูลทดสอบของ Cityscapes วิธีการมาตรฐาน (ก) Tiramisu (ข) DeepLab-V3+ ด้วย ResNet-101 (ค) DeepLab-V3+ ด้วย Xception เมื่อเทียบกับวิธีการที่นำเสนอ (ง) DeepLab-V3-A1 ด้วย ResNet-101 (จ) DeepLab-V3-A1 ด้วย Xception

บทที่ 6
สรุปการวิจัยและแนวทางการวิจัยในขั้นถัดไป

6.1 สรุปการวิจัย

ในงานวิทยานิพนธ์นี้ได้นำเสนอวิธีการปรับปรุงการทำนายรายพิกเซลชุดข้อมูลถนนในกรุงเทพมหานคร ด้วยการสร้างชุดข้อมูลทั้งหมด 701 ภาพ ประกอบด้วยชุดข้อมูลฝึก 367 ภาพ, ชุดข้อมูลตรวจสอบ 101 ภาพ, และชุดข้อมูลทดสอบ 233 ภาพ เพื่อฝึกสถาปัตยกรรมการเรียนรู้เชิงลึกเพื่อการแบ่งส่วนภาพเชิงความหมาย จำนวน 11 คลาส อีกทั้งเรายังเสนอวิธีการเรียนรู้เชิงลึก 2 วิธี ได้แก่ DeepLab-V3-A1 ด้วย ResNet-101 และ DeepLab-V3-A1 ด้วย Xception ตามลำดับ ซึ่งเกิดจากการปรับปรุงส่วนดีโคดเดอร์ของวิธี DeepLab-V3+ ด้วยจำนวนคอนโวลูชัน 1x1 ที่มีจำนวนแตกต่างกันเพื่อช่วยในการฟื้นฟูฟีเจอร์แมพฝั่งดีโคดเดอร์ โดยวิธีการที่เรานำเสนอทั้งสองวิธีมีประสิทธิภาพสูงบนชุดข้อมูลท้องถนนในกรุงเทพมหานคร, CamVid และ Cityscapes ตามลำดับ ในทุกหน่วยมาตรวัดได้แก่ Precision, Recall, และ F1 โดย 3 หน่วย ที่กล่าวมานี้มีค่า ไม่ต่ำกว่า 80.00% ในทุกการทดลอง และในส่วนของมาตรวัด Mean IoU เราพบว่าวิธีนำเสนอทั้งสองวิธีมีผลลัพธ์ที่ดีกว่าวิธีมาตรฐาน

6.2 ข้อเสนอแนะเกี่ยวกับวิทยานิพนธ์นี้

นักวิจัยในสาขาอื่น ๆ สามารถนำสถาปัตยกรรม DeepLab-V3-A1 ทั้ง 2 วิธีไปประยุกต์ใช้กับชุดข้อมูลอื่น ๆ เพื่อนำองค์ความรู้จากโมเดลการแบ่งส่วนเชิงความหมายไปตีความเชิงเทคนิคบนชุดข้อมูลนั้น ๆ ได้ แต่อาจจะต้องปรับไฮเปอร์พารามิเตอร์ให้เหมาะสมกับชุดข้อมูลนั้น ๆ เพื่อเพิ่มประสิทธิภาพการฝึกบนชุดข้อมูลนั้น ๆ ประกอบกับนักวิจัยระบบขับเคลื่อนอัตโนมัติสามารถนำชุดข้อมูลถนนกรุงเทพมหานครไปฝึกกับโมเดลการแบ่งส่วนภาพเชิงความหมาย เพื่อทำให้ระบบขับเคลื่อนอัตโนมัติเข้าใจภูมิประเทศของการขับขี่บนถนนกรุงเทพฯ ในขั้นเบื้องต้นได้ เนื่องจากชุดข้อมูลถนนกรุงเทพฯ นี้มีข้อจำกัดในสภาพแสงแดดที่มีเฉพาะตอนกลางวัน และบางภาพอาจจะมีเงาต้นไม้บัง

6.3 แนวทางการวิจัยในขั้นถัดไป

สามารถปรับปรุงสถาปัตยกรรมการเรียนรู้ของเราด้วยตัวสกัดฟีเจอร์ที่มีประสิทธิภาพที่ดีกว่าในวิธีที่เรานำเสนอ โดยในปัจจุบันมีงานวิจัยที่พัฒนาการสกัดฟีเจอร์ด้วยเทคนิคการเรียนรู้เชิงลึกด้วย HRNet-V2 และเราจะทำการศึกษาความเป็นไปได้ในอนาคตที่จะสร้างชุดข้อมูลพร้อมผลเฉลยเพิ่มเติมจากชุดข้อมูลถนนในกรุงเทพมหานคร

บรรณานุกรม

1. Nakamura, K., et al., *Evaluation for Low-carbon Land-use Transport Development with QOL Indexes in Asian Developing Megacities: a Case Study of Bangkok.* 2015. 11: p. 1047-1063.
2. Redmon, J. and A.J.a.p.a. Farhadi, *Yolov3: An incremental improvement.* 2018.
3. Jégou, S., et al. *The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017.
4. Badrinarayanan, V., et al., *Segnet: A deep convolutional encoder-decoder architecture for image segmentation.* 2017. 39(12): p. 2481-2495.
5. Ronneberger, O., P. Fischer, and T. Brox. *U-net: Convolutional networks for biomedical image segmentation*. in *International Conference on Medical image computing and computer-assisted intervention*. 2015. Springer.
6. Zhao, H., et al. *Pyramid scene parsing network*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
7. Chen, L.-C., et al. *Encoder-decoder with atrous separable convolution for semantic image segmentation*. in *Proceedings of the European conference on computer vision (ECCV)*. 2018.
8. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
9. Chollet, F. *Xception: Deep learning with depthwise separable convolutions*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
10. Brostow, G.J., J. Fauqueur, and R.J.P.R.L. Cipolla, *Semantic object classes in video: A high-definition ground truth database.* 2009. 30(2): p. 88-97.
11. Cordts, M., et al. *The cityscapes dataset for semantic urban scene understanding*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
12. Wada, K. *labelme: Image Polygonal Annotation with Python*. 2016; Available

from: https://github.com/wkentaro/labelme.

13. Hinton, G., N. Srivastava, and K.J.C.o. Swersky, *Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.* 2012. 14(8): p. 2.
14. LeCun, Y., et al., *Backpropagation applied to handwritten zip code recognition.* 1989. 1(4): p. 541-551.
15. Chen, L.-C., et al., *Semantic image segmentation with deep convolutional nets and fully connected crfs.* 2014.
16. Srivastava, N., et al., *Dropout: a simple way to prevent neural networks from overfitting.* 2014. 15(1): p. 1929-1958.
17. Odena, A., V. Dumoulin, and C.J.D. Olah, *Deconvolution and checkerboard artifacts.* 2016. 1(10): p. e3.
18. Deng, J., et al. *Imagenet: A large-scale hierarchical image database*. in 2009 *IEEE conference on computer vision and pattern recognition*. 2009. Ieee.
19. Simonyan, K. and A.J.a.p.a. Zisserman, *Very deep convolutional networks for large-scale image recognition.* 2014.
20. Long, J., E. Shelhamer, and T. Darrell. *Fully convolutional networks for semantic segmentation*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
21. Yosinski, J., et al. *How transferable are features in deep neural networks?* in *Advances in neural information processing systems*. 2014.
22. Everingham, M., et al., *The pascal visual object classes (voc) challenge.* 2010. 88(2): p. 303-338.
23. Lin, M., Q. Chen, and S.J.a.p.a. Yan, *Network in network.* 2013.
24. Huang, G., et al. *Densely connected convolutional networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
25. Howard, A.G., et al., *Mobilenets: Efficient convolutional neural networks for mobile vision applications.* 2017.
26. Zhen, M., et al. *Learning Fully Dense Neural Networks for Image Semantic Segmentation*. in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019.
27. *Iwahori Lab Computer Vision Laboratory*. Available from:

http://www.cvl.cs.chubu.ac.jp.

28. Chen, L.-C., et al., *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.* 2017. 40(4): p. 834-848.

29. Chen, L.-C., et al., *Rethinking atrous convolution for semantic image segmentation.* 2017.

ประวัติผู้เขียน

ชื่อ-สกุล	นายกฤษพล ธิติสิริเวช
วัน เดือน ปี เกิด	3 เมษายน 2538
สถานที่เกิด	โรงพยาบาลเจริญกรุงประชารักษ์ จังหวัดกรุงเทพมหานคร
วุฒิการศึกษา	ปริญญาวิทยาศาสตร์บัณฑิต สาขาวิชาคณิตศาสตร์ มหาวิทยาลัยมหิดล
ที่อยู่ปัจจุบัน	507/454 ถนนสาธุประดิษฐ์ แขวงช่องนนทรี เขตยานนาวา จังหวัดกรุงเทพมหานคร