การลบภาพวัตถุสำหรับระบบจัดทำแผนที่ชนิดเคลื่อนที่โดยใช้โครงข่ายเจเนอเรทีฟแอดเวอเซอเรียล

นางสาวอาซีซ่าร์ ลอดิง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

การลบภาพวัตถุสำหรับระบบจัดทำแผนที่ชนิดเคลื่อนที่โดยใช้โครงข่ายเจเนอเรทีฟแอดเวอเซอเรียล

นางสาวอาซีซ่าร์ ลอดิง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง การลบภาพวัตถุสำหรับระบบจัดทำแผนที่ชนิดเคลื่อนที่โดยใช้โครงข่ายเจเนอเรทีฟแอดเวอเซอเรียล

โดย นางสาวอาซีซ่าร์ ลอดิง

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

..... คณบดีบัณฑิตวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร.สุพจน์ จันทร์วิพัฒน์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(ดร.มารุต บูรณรัช)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.ลือพล พิพานเมฆาภรณ์)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.นิกร สุทธิเสงี่ยม)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุวัจชัย กมลสันติโรจน์)

ชื่อ : นางสาวอาซีซ่าร์ ลอดิง
ชื่อวิทยานิพนธ์ : การลบภาพวัตถุสำหรับระบบจัดทำแผนที่ชนิดเคลื่อนที่โดยใช้โครงข่ายเจเนอเรทีฟแอดเวอเซอเรียล
สาขาวิชา : วิทยาการคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก : ผู้ช่วยศาสตราจารย์ ดร.ลือพล พิพานเมฆาภรณ์
ปีการศึกษา : 2567

บทคัดย่อ

ระบบจัดทำแผนที่ชนิดเคลื่อนที่ (mobile mapping system) เป็นเทคโนโลยีช่วยเก็บข้อมูลโดยใช้ยานพาหนะเพื่อบันทึกภาพแบบ 360 องศาที่มีความละเอียดสูงมาก (high definition) ระหว่างเคลื่อนที่ผ่านพื้นที่สำรวจ อย่างไรก็ตามในการนำภาพที่บันทึกไปใช้งาน จำเป็นจะต้องผ่านการประมวลผลเพื่อลบข้อมูลที่ไม่ต้องการออกจากภาพ แม้ว่าปัจจุบันมีงานวิจัยจำนวนหนึ่งนำเสนอเทคนิคกาารลบและเติมวัตถุในภาพถ่าย อย่างไรก็ตามเนื่องด้วยคุณลักษณะของภาพแบบ 360 องศา ที่แตกต่างจากภาพถ่ายดิจิทัลทั่วไป ส่งผลให้เทคนิคดังกล่าวอาจไม่สามารถใช้ลบวัตถุในภาพแบบ 360 องศาได้อย่างสมบูรณ์ และอาจต้องใช้เวลาในการประมวลผลค่อนข้างมาก งานวิจัยนี้มีวัตถุประสงค์เพื่อนำเสนอวิธีการลบและเติมวัตถุในภาพแบบ 360 องศาความละเอียดสูงมาก ด้วยเทคนิค Generative Adversarial Network (GAN) โดยภาพอินพุตจะถูกแปลงให้อยู่ในระบบพิกัดทรงกลม (Spherical coordinate system) เพื่อลบวัตถุเป้าหมายที่มีตำแหน่งอยู่ด้านล่างของภาพซึ่งคือรถสำรวจ จากนั้นจึงทำการเติมภาพส่วนที่ถูกลบออกโดย GAN ซึ่งถูกฝึกฝนให้สร้างภาพใหม่แทนที่ภาพวัตถุเป้าหมายที่ต้องการลบ วิธีการที่นำเสนอนี้มีข้อดีหลายประการ ได้แก่ 1) สามารถนำไปใช้ลบภาพวัตถุเป้าหมายในภาพได้ โดยไม่ต้องอาศัยข้อมูลอื่นมาช่วย 2) คุณภาพของภาพที่ถูกเติมใกล้เคียงกับภาพจริงและผสานอย่างกลมกลืนกับสภาพแวดล้อม นอกจากวิธีการดังกล่าวยังใช้เวลาในการประมวลไม่มาก จึงเป็นวิธีที่มีความเหมาะสมอย่างยิ่งสำหรับระบบจัดทำแผนที่ชนิดเคลื่อนที่ซึ่งต้องการความรวดเร็วในการประมวลผลภาพจำนวนมาก

(มีจำนวนทั้งสิ้น 107 หน้า)

คำสำคัญ : การลบภาพวัตถุ,การจัดทำแผนที่ชนิดเคลื่อนที่,โครงข่ายแบบเจเนอเรทีฟแอดเวอเซอเรียล

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

Name : Miss Arzeezar Lording
Thesis Title : Object Removals for Mobile Mapping System using Generative Adversarial Network
Major Field : Computer Science
King Mongkut's University of Technology North Bangkok
Thesis Advisor : Assistant Professor Dr. Luepol Pipanmekaporn, Ph.D.
Academic Year : 2024

ABSTRACT

Mobile mapping systems (MMS) capture geospatial data using vehicle-mounted sensors and cameras for generating digital maps from capturing equirectangular panoramic images with high definition. However, the collected images contain unwanted information that needs to be remove for further use. Although object removal approaches have been proposed, these methods may not be well-suited for MMS imagery due to their unique characteristics and processing requirements and require more time processing. In this study, we propose a Generative Adversarial Network (GAN)-based technique tailored to remove unwanted vehicles from MMS scenes. To handle the challenge of inpainting full panoramic scenes, we convert MMS images into spherical coordinate system to extract viewports containing target vehicles and remove target from image. Subsequently, a Pix2Pix-based inpainting network is employed to inpaint the target object in viewport. Our approach has many advantages such as 1) our method can inpaint target object not required information other than information in one image 2) our method preserves the output quality while efficiently processing large raw MMS image datasets. Considering the characteristics of MMS datasets, our approach is appropriate for MMS image datasets that require computational efficiency.

(Total 107 Pages)

Keywords: Object Removal,Mobile Mapping System,Generative Adversarial Network

_____ Advisor

กิตติกรรมประกาศ

วิทยานิพนธ์เรื่อง “การลบภาพวัตถุสำหรับระบบจัดทำแผนที่ชนิดเคลื่อนที่โดยใช้โครงข่ายเจเนอเรทีฟแอดเวอเซอเรียล” นี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์และความกรุณาจาก ผู้ช่วยศาสตราจารย์ ดร.ลือพล พิพานเมฆาภรณ์ อาจารย์ที่ปรึกษา ที่ได้ให้ความรู้และข้อเสนอแนะ ด้วยความเอาใจใส่และให้ความทุ่มเทตั้งแต่ช่วงเริ่มการวิจัยจนกระทั่งเสร็จสมบูรณ์ ขอขอบคุณผู้ช่วยศาสตราจารย์ ดร.นิกร สุทธิเสงี่ยม ที่สนับสนุนข้อมูลที่ใช้ในงานวิจัยและให้คำปรึกษาอันเป็นประโยชน์อย่างยิ่งแก่งานวิจัย นอกจากนี้ ผู้วิจัยขอขอบพระคุณคณาจารย์ของภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ทุกท่านที่ได้ให้ความรู้ต่างๆ ที่เป็นประโยชน์ต่อการวิจัยในครั้งนี้ ขอบพระคุณเจ้าหน้าที่ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศทุกท่านที่ได้ให้การช่วยเหลือในระหว่างการศึกษาเป็นอย่างดี ขอบพระคุณบิดา มารดาและทุกคนในครอบครัว และเพื่อนทุกคนตลอดจนบุคคลที่ไม่ได้กล่าวนามทุกท่าน ซึ่งได้ให้ความช่วยเหลือ ให้การสนับสนุนและกำลังใจตลอดระยะเวลาการศึกษาที่ผ่านมา ทำให้การศึกษานี้สำเร็จลุล่วงไปได้ด้วยดี หากสารนิพนธ์ฉบับนี้มีข้อผิดพลาดหรือข้อบกพร่องประการใด ผู้วิจัยต้องขออภัยมา ณ ที่นี้

อาซีซ่าร์ ลอดิง

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญภาพ

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบการจัดทำแผนที่แบบเคลื่อนที่ (Mobile Mapping System) [1] เป็นเทคโนโลยีที่ช่วยในการเก็บข้อมูลทางภูมิศาสตร์โดยใช้ยานพาหนะที่ติดตั้งอุปกรณ์สำหรับบันทึกภาพขณะยานพาหนะกำลังเคลื่อนที่ เพื่อลดเวลาในการสำรวจพื้นที่ขนาดใหญ่ อย่างไรก็ตามในการนำภาพที่ถูกบันทึกได้ไปจัดทำแผนที่ดิจิทัล (digital map) จำเป็นจะต้องนำภาพดังกล่าวไปทำการประมวลผลเพื่อลบข้อมูลที่มีแนวโน้มละเมิดสิทธิ์ความเป็นส่วนตัว (privacy) ของบุคคลอื่น เช่น ยานพาหนะบนท้องถนน ผู้ขับขี่ยานพาหนะ คนเดินอยู่บนทางเท้า ป้ายข้างทาง เป็นต้น เทคนิคทั่วไปที่มักถูกนำมาใช้ใช้ในการปกป้องความเป็นส่วนตัว ได้แก่ การเบลอภาพ (Blurring) การลดความละเอียด (Pixelization) การระบายสีทับ (Painting) ในบริเวณวัตถุที่ไม่ต้องการ ถึงแม้ว่าวิธีการดังกล่าวจะสามารถปิดบังความเป็นส่วนตัวได้ อย่างไรก็ตามเทคนิคดังกล่าวมักส่งผลให้คุณภาพของรูปภาพลดลง และยังเป็นจุดที่รบกวนสายตาซึ่งมักส่งผลต่อความพึงพอใจในการใช้งานแผนที่

จากปัญหาดังกล่าวนักวิจัยจึงมีแนวคิดในการจัดการกับภาพวัตถุที่ต้องการลบออกโดยการตกแต่งภาพในบริเวณที่ถูกลบให้ออกมาเป็นธรรมชาติมากที่สุด เรียกว่า การเติมภาพ (image inpainting) ซึ่งโดยทั่วไปเทคนิคเติมภาพสามารถแบ่งออกเป็น 2 หลักการ ได้แก่ 1) การเติมภาพโดยอาศัยข้อมูลจากบริเวณอื่นในภาพหรือจากภาพอื่นที่มีความสอดคล้องกับบริเวณที่ต้องการลบ หรือ Patch-based inpainting [2,3,4] 2) การเติมภาพโดยใช้วิธีการสร้างภาพในบริเวณวัตถุที่ต้องการลบขึ้นมาใหม่โดยใช้เทคนิคการเรียนรู้เชิงลึก (Deep learning) [5,6,7,8,9,10] สำหรับเป้าหมายของงานวิจัยนี้มุ่งเน้นในการลบและเติมภาพวัตถุจากภาพที่ถูกบันทึกจากกล้องบันทึกภาพแบบ 360 องศา ซึ่งเป็นภาพแบบพาโนรามา (Panorama) ที่มีความแตกต่างจากภาพดิจิทัลทั่วไปทั้งในแง่มุมมองของภาพ อัตราส่วนของภาพ และขนาดของภาพ ส่งผลให้เทคนิคการเติมภาพทั่วไปอาจไม่สามารถให้ผลลัพธ์ได้อย่างเป็นธรรมชาติ ยิ่งไปกว่านั้นเนื่องจากจำนวนภาพที่ต้องการนำไปประมวลผลสำหรับการจัดทำแผนที่ชนิดเคลื่อนที่มักมีปริมาณมาก เทคนิคการเติมภาพทั่วไปอาจใช้เวลาในการประมวลผลนาน โดยข้อจำกัดเหล่านี้ เทคนิคเติมภาพวัตถุจึงต้องถูกออกแบบอย่างเหมาะสมสำหรับระบบจัดทำแผนที่ชนิดเคลื่อนที่ได้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาเทคนิคการลบและเติมภาพวัตถุจากภาพพาโนรามาที่ถูกบันทึกได้จากกล้องบันทึกภาพแบบ 360 องศา สำหรับระบบจัดทำแผนที่ชนิดเคลื่อนที่ได้

1.3 ขอบเขตของการวิจัย

1.3.1 ชุดข้อมูลภาพที่นำมาสร้างและทดสอบในงานวิจัยนี้ ได้จากกล้อง Ladybug 360 องศา ที่ถูกติดตั้งบนรถสำรวจพื้นที่

1.3.2 วัตถุเป้าหมายที่ต้องการลบในงานวิจัยนี้คือ ส่วนของรถสำรวจพื้นที่

1.3.3 ใช้เทคนิค Generative Adversarial Network (GAN) เพื่อสร้างภาพแทนที่ภาพวัตถุเป้าหมายที่ต้องการลบได้อย่างเป็นธรรมชาติ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ลดขั้นตอนและเวลาในการประมวลผลภาพที่บันทึกได้จากยานพาหนะสำรวจพื้นที่ก่อนนำไปจัดทำแผนที่ดิจิทัล

1.4.2 เพิ่มคุณภาพของภาพในระบบแผนที่ดิจิทัล

1.5 นิยามคำศัพท์เฉพาะ

ภาพเฉลย (Ground truth) หมายถึง ภาพที่มีจากการเติมภาพโดยใช้วิธี Patch ซึ่งจะใช้ในการสอนโมเดลในการเติมภาพโดยใช้เทคนิค Deep learning

การเติมภาพทับ (Overlay) หมายถึง วิธีการเติมภาพโดยใช้ binary mask ในการระบุตำแหน่งการเติมภาพ โดยจะเติมภาพเฉพาะในส่วนที่ binary mask เป็นสีดำ

การสกัดคุณลักษณะ (Feature Extraction) เป็นวิธีการสกัดคุณลักษณะข้อมูลโดยการสร้างคุณลักษณะขึ้นมาใหม่ จากการทำผลรวมแบบเชิงเส้นของคุณลักษณะข้อมูลที่มีอยู่เดิม แล้วคัดเลือกเอา คุณลักษณะที่มีความสำคัญที่สุดออกมา

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 ระบบจัดทำแผนที่ชนิดเคลื่อนที่ (Mobile Mapping System)

ระบบจัดทำแผนที่ชนิดเคลื่อนที่ [1] เป็นระบบที่ติดตั้งเซนเซอร์ชนิดต่าง ๆ เข้ากับยานพาหนะ เพื่อให้ได้มาซึ่งข้อมูลที่ต้องการอย่างรวดเร็วและประหยัดค่าใช้จ่าย เซนเซอร์ที่มีความนิยมในการติดตั้งคือ อุปกรณ์กำเนิดแสงเลเซอร์เพื่อสำรวจพื้นที่แบบสามมิติ ที่ทำให้สามารถวัดขนาดและระยะทางต่าง ๆ ได้ พร้อมทั้งยังติดอุปกรณ์บันทึกภาพเพื่อเก็บข้อมูลพื้นที่รอบข้างยานพาหนะระหว่างสำรวจได้อย่างต่อเนื่อง ข้อมูลที่ได้มานอกจากนำไปสร้างแผนที่แล้ว ยังสามารถนำไปใช้งานด้านต่าง ๆ เช่น การสำรวจสภาพถนน สิ่งปลูกสร้าง การออกแบบเส้นทาง เป็นต้น โดยข้อมูลที่ได้จากระบบจัดทำแผนที่ชนิดเคลื่อนที่ประกอบด้วย 2 ส่วน ได้แก่

ข้อมูลภาพพาโนรามา 360 องศาที่ได้จากอุปกรณ์บันทึกภาพอยู่ในลักษณะภาพพาโนรามาแบบทรงกลม (Spherical panorama) แล้วแปลงภาพให้อยู่ในรูปแบบการฉายแบบ Equirectangular โดยข้อมูลนี้ถูกบันทึกในรูปแบบไฟล์ JPG

ภาพที่ 2-1 ข้อมูลภาพพาโนรามา

ข้อมูลพิกัดตำแหน่งทางภูมิศาสตร์ ตามพิกัด GPS ซึ่งถูกบันทึกตามตำแหน่งที่มีการบันทึกภาพพาโนรามา ข้อมูลพิกัดนี้อยู่ในระบบพิกัดกริดแบบยูทีเอ็ม (Universal Transverse Mercator) ซึ่งประกอบด้วย ลำดับของข้อมูลซึ่งตรงกับชื่อไฟล์ภาพพาโนรามาที่บันทึกขณะสำรวจพื้นที่ เวลาที่บันทึกพิกัด สัปดาห์ที่บันทึกตำแหน่ง ตำแหน่งพิกัดแกน X ตำแหน่งพิกัดแกน Y ตำแหน่งพิกัดแกน องศารอบแกน X องศารอบแกน Y และองศารอบแกน Z ข้อมูลพิกัดตำแหน่งนี้ถูกบันทึกในรูปแบบไฟล์ TXT

```
id,image,time,X,Y,Z,Heading,Roll,Pitch,Camera,Quality,Line,Color,AccuracyXyz
0,img_000000.jpg,1406946447.667,616102.775,1741800.188,31.123,-27.387,-0.326,0.210,0,1,1,C0;0;0;0;0;0,0.1
1,img_000001.jpg,1406946447.833,616102.449,1741800.800,31.121,-28.883,-0.357,0.247,0,1,1,C0;0;0;0;0;0,0.1
2,img_000002.jpg,1406946448.000,616102.103,1741801.420,31.123,-30.383,-0.252,0.200,0,1,1,C0;0;0;0;0;0,0.1
3,img_000003.jpg,1406946448.167,616101.728,1741802.038,31.122,-31.897,-0.309,0.198,0,1,1,C0;0;0;0;0;0,0.1
4,img_000004.jpg,1406946448.333,616101.333,1741802.650,31.119,-33.481,-0.465,0.270,0,1,1,C0;0;0;0;0;0,0.1
5,img_000005.jpg,1406946448.500,616100.914,1741803.268,31.118,-35.091,-0.485,0.298,0,1,1,C0;0;0;0;0;0,0.1
6,img_000006.jpg,1406946448.667,616100.471,1741803.886,31.120,-36.693,-0.471,0.269,0,1,1,C0;0;0;0;0;0,0.1
7,img_000007.jpg,1406946448.833,616100.005,1741804.496,31.121,-38.290,-0.494,0.340,0,1,1,C0;0;0;0;0;0,0.1
8,img_000008.jpg,1406946449.000,616099.508,1741805.108,31.119,-39.850,-0.554,0.418,0,1,1,C0;0;0;0;0;0,0.1
9,img_000009.jpg,1406946449.167,616098.979,1741805.723,31.118,-41.340,-0.598,0.475,0,1,1,C0;0;0;0;0;0,0.1
10,img_000010.jpg,1406946449.333,616098.424,1741806.341,31.120,-42.753,-0.651,0.577,0,1,1,C0;0;0;0;0;0,0.1
11,img_000011.jpg,1406946449.500,616097.833,1741806.973,31.122,-44.063,-0.533,0.639,0,1,1,C0;0;0;0;0;0,0.1
12,img_000012.jpg,1406946449.667,616097.209,1741807.618,31.123,-45.166,-0.299,0.636,0,1,1,C0;0;0;0;0;0,0.1
13,img_000013.jpg,1406946449.833,616096.549,1741808.265,31.128,-46.170,-0.300,0.603,0,1,1,C0;0;0;0;0;0,0.1
14,img_000014.jpg,1406946450.000,616095.850,1741808.923,31.134,-47.024,-0.307,0.610,0,1,1,C0;0;0;0;0;0,0.1
15,img_000015.jpg,1406946450.167,616095.130,1741809.601,31.130,-47.825,-0.049,0.681,0,1,1,C0;0;0;0;0;0,0.1
16,img_000016.jpg,1406946450.333,616094.380,1741810.284,31.134,-48.489,0.213,0.715,0,1,1,C0;0;0;0;0;0,0.1
17,img_000017.jpg,1406946450.500,616093.593,1741810.977,31.138,-49.142,0.269,0.701,0,1,1,C0;0;0;0;0;0,0.1
18,img_000018.jpg,1406946450.667,616092.778,1741811.679,31.144,-49.761,0.333,0.657,0,1,1,C0;0;0;0;0;0,0.1
```

ภาพที่ 2-2 ข้อมูลพิกัดตำแหน่ง

2.2 ภาพพาโนรามา (Panoramic image)

ภาพพาโนรามา [11] คือภาพถ่ายที่มีมุมมองกว้างและความสูงมากกว่าภาพถ่ายปกติ โดยถ่ายภาพหลาย ๆ ภาพ แล้วนำภาพมาต่อกัน ภาพพาโนรามาสามารถถ่ายภาพที่มีมุมมองการมองเห็นได้สูงถึง 360 องศา นิยมนำมาเป็นสื่อพาโนรามาเสมือนจริงที่สามารถมองเห็นได้รอบทิศทาง รูปแบบของภาพพาโนรามามี 4 รูปแบบ ดังนี้

ภาพพาโนรามาแบบระนาบ (Planer panorama) เป็นภาพพาโนรามาที่มีภาพใกล้เคียงกับภาพปกติ มีการบิดเบี้ยวของภาพเล็กน้อย หรือไม่มีการบิดเบี้ยวเลย การถ่ายภาพชนิดนี้จะถ่ายด้วยมุมมองไม่เกิน 180 องศา

ภาพที่ 2-3 ภาพพาโนรามาแบบระนาบ (Planer panorama)

[12]

ภาพพาโนรามาทรงกระบอก (Cylindrical panorama) เป็นพาโนรามาที่มีมุมมองด้านยาว 360 องศา ลักษณะของภาพเหมือนอยู่ในถังทรงกระบอก

ภาพที่ 2-4 ภาพพาโนรามาทรงกระบอก (Cylindrical panorama)

[12]

ภาพพาโนรามาทรงกลม (Spherical panorama) เป็นพาโนรามาที่มีด้านยาว (แนวนอน) 360 องศา และด้านกว้าง (แนวตั้ง) 180 องศา ลักษณะการมองเหมือนอยู่ในทรงกลม เมื่อคลี่ออก เรียกว่า Equirectangular และมีอัตราส่วนด้านยาวต่อด้านกว้าง เป็น 2 ต่อ 1 เสมอ

ภาพที่ 2-5 ภาพพาโนรามาทรงกลม (Spherical panorama)

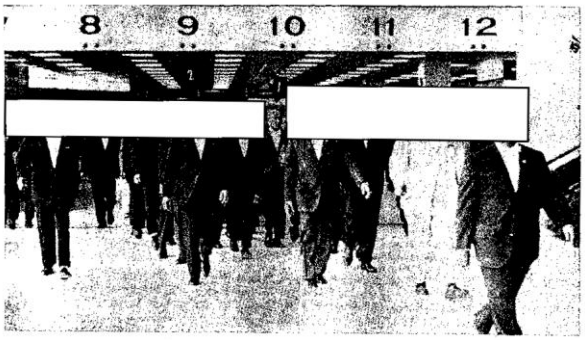
ภาพพาโนรามาทรงลูกบาศก์ (Cubic panorama) เป็นภาพพาโนรามาที่มีลักษณะภาพเหมือนมองจากด้านในกล่องสี่เหลี่ยม ซึ่งมีภาพ 6 ด้าน ภาพพาโนรามาทรงลูกบาศก์นี้เหมาะกับการนำภาพพาโนรามาแบบทรงกลมมาปรับแก้ส่วนที่ไม่สมบูรณ์

ภาพที่ 2-6 ภาพพาโนรามาทรงลูกบาศก์ (Cubic panorama)

2.3 การลบภาพวัตถุ (Object removals)

การลบวัตถุในภาพเป็นกระบวนการที่ใช้ในการแก้ไขภาพดิจิทัลเพื่อลบวัตถุที่ไม่ต้องการออกจากภาพนั้นๆ โดยทั่วไปการลบวัตถุในภาพมีวัตถุประสงค์เพื่อปิดบังข้อมูลหรือปกป้องความเป็นส่วนตัว (privacy) ของผู้อื่น โดยทั่วไปวิธีการลบวัตถุในภาพสำหรับการปกป้องความเป็นส่วนตัว มักใช้เทคนิคการประมวลผลภาพ (image processing) ได้แก่

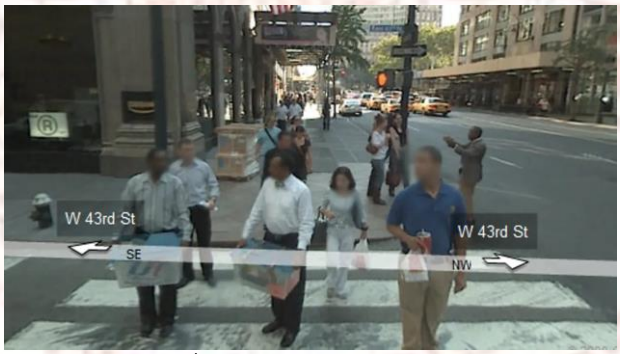
การระบายทับ (Painting) ในบริเวณวัตถุที่ไม่ต้องการ เป็นวิธีการที่ทำได้ง่ายที่สุด โดยการแทนที่พิกเซล (pixel) ของวัตถุเป้าหมายที่ต้องการลบด้วยพิกเซลสีอื่น ถึงแม้ว่าวิธีการดังกล่าวจะสามารถปิดบังความเป็นส่วนตัวได้ อย่างไรก็ตามเทคนิคนี้มักส่งผลกระทบในแง่ของคุณภาพของภาพลดลงอย่างมาก



ภาพที่ 2-7 การระบายทับ (Painting)

[13]

การเบลอ (Blurring) ซึ่งเป็นวิธีการที่ใช้อย่างแพร่หลาย โดยมีเทคนิคที่หลากหลาย เช่น Gaussian Blur [14] ซึ่งลดความเด่นของวัตถุที่ต้องการลบด้วยการใช้การกระจายค่าสีจากพื้นหลังของวัตถุ หรือ Lens Blur ซึ่งทำให้ภาพวัตถุถูกเบลอโดยการจำลองจากเลนส์กล้อง



ภาพที่ 2-8 การเบลอ (Blurring)

[15]

การลดความละเอียด (Pixelization) ซึ่งเป็นวิธีการแก้ไขภาพที่เปลี่ยนส่วนของภาพให้เป็นบล็อกของพิกเซลที่ถูกขยาย เช่น ภาพโมเสก (Mosaic) เพื่อปกปิดรายละเอียดของวัตถุที่ไม่ต้องการให้เห็น เช่น การปกปิดใบหน้า เป็นต้น เทคนิคนี้มีประสิทธิภาพในการปกปิดรายละเอียดโดยไม่ลบวัตถุออกจากภาพ และสามารถปรับระดับของการแสดงรายละเอียดวัตถุได้ตามต้องการ

ภาพที่ 2-9 การลดความละเอียด (Pixelization)

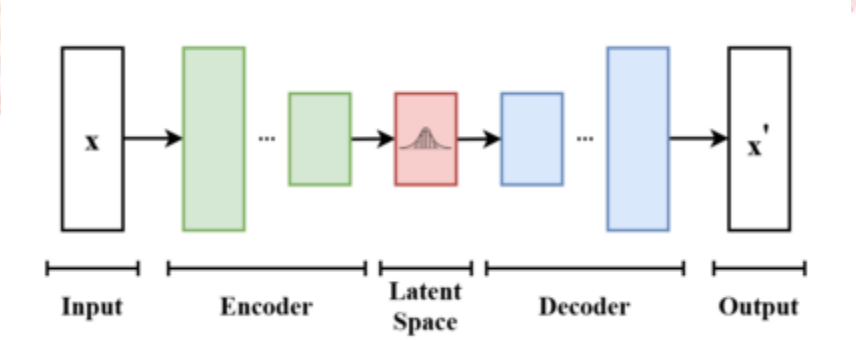
[16]

2.4 การเรียนรู้เชิงลึกแบบรู้สร้าง (Generative deep learning)

เป็นเทคนิคการเรียนรู้เชิงลึกเพื่อสร้างเนื้อหาของภาพขึ้นใหม่ ตามวัตถุประสงค์ของงานเป้าหมาย เช่น งานศิลปะ การสร้างภาพตัวละครในเกมส์ และการสร้างภาพสินค้า เป็นต้น ปัจจุบันเทคนิคการเรียนรู้เชิงลึกแบบรู้สร้างสามารถแบ่งออกเป็น 3 หลักการ ได้แก่

2.4.1 Variational Autoencoder (VAE)

VAE [17] เป็นโมเดลโครงข่ายประสาทเทียมที่สามารถสร้างข้อมูลใหม่ที่มีความคล้ายคลึงกับข้อมูลนำเข้า VAE ประกอบไปด้วย โครงข่ายประสาทเทียมย่อย 2 ตัว ได้แก่ ตัวเข้ารหัส (Encoder) และตัวถอดรหัส (Decoder) ตัวเข้ารหัสจะทำหน้าที่แปลงข้อมูลนำเข้าให้อยู่ในรูปของเวกเตอร์มิติต่ำแบบซ่อน (low-dimensional latent vector) เพื่อจำลอง distribution ของข้อมูล ขณะที่ตัวถอดรหัสจะนำ latent vector ที่สุ่มได้ไปสร้างข้อมูลใหม่ออกมา โดยการพยายามลอกเลียนข้อมูลนำเข้า ในการฝึกฝนโมเดล VAE สำหรับการเติมภาพจะใช้ชุดข้อมูลซึ่งประกอบไปด้วยภาพก่อนและหลังลบวัตถุเป้าหมาย



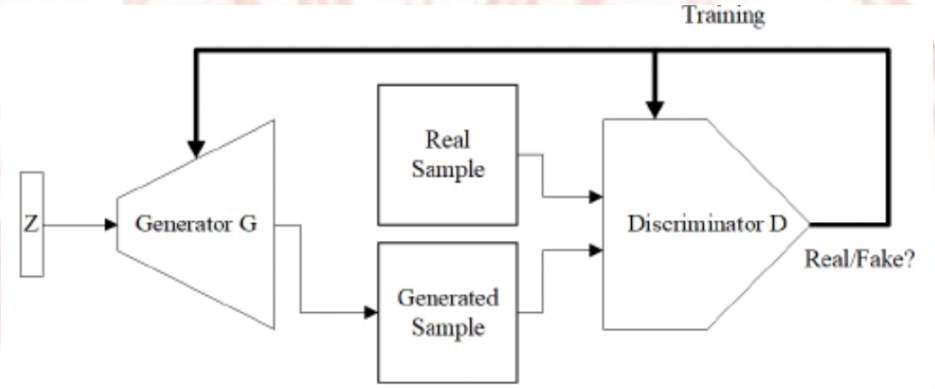
ภาพที่ 2-10 Variational Autoencoder (VAE)

[18]

2.4.2 Generative Adversarial Network (GAN)

Generative Adversarial Network (GAN) [19] เป็นโมเดลโครงข่ายประสาทเทียมที่ประกอบไปด้วย โครงข่ายประสาทเทียมย่อย 2 ส่วน ได้แก่ Generator ที่ทำหน้าที่สร้างรูปภาพสังเคราะห์ และ Discriminator ที่มีหน้าที่ในการแยกแยะภาพสังเคราะห์กับภาพจริงออกจากกัน โดยมีกระบวนการฝึกฝนโครงข่ายประสาทเทียมย่อยทั้งสองส่วนร่วมกันตามสมการที่ 2-1 โดยผลลัพธ์ที่ได้จากการฝึกฝนโมเดล GAN ก็คือ Generator ซึ่งมีความสามารถในการสร้างภาพใหม่จากภาพที่นำเข้า

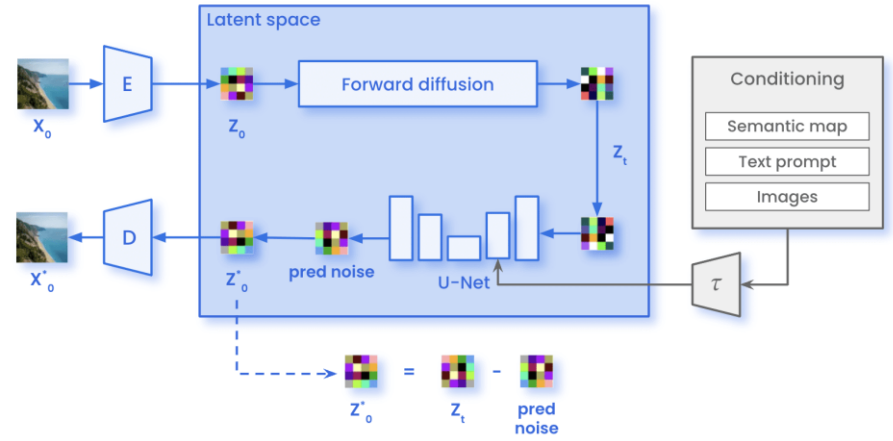
$$L_{GAN} = \min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[log\, D(x|y)] + \mathbb{E}_{z\sim p_z(z)}\left[log\left(1 - D\big(G(z|y)\big)\right)\right] \quad (2\text{-}1)$$



ภาพที่ 2-11 โครงข่าย Generative Adversarial Network (GAN) [20]

2.4.3 Stable Diffusion (SD)

SD [21] เป็นโมเดลโครงข่ายประสาทเทียมที่ใช้ในการสร้างหรือแก้ไขภาพตามข้อความ (text) โดย stable diffusion จะประกอบไปด้วย 2 ส่วนหลัก ได้แก่ ส่วน diffusion ซึ่งทำหน้าที่สร้าง noise และการลบ noise (denoising) ออกจากข้อมูล และส่วนโมเดล generative ซึ่งทำหน้าที่สร้างข้อมูลใหม่จากผู้ใช้ด้วยการป้อนข้อความ และค่า noise เพื่อควบคุมโมเดลสำหรับการสร้างภาพ อย่างไรก็ตามโมเดล Stable Diffusion ต้องการข้อมูลที่มีคุณภาพและความหลากหลายจำนวนมากเพื่อใช้ฝึกฝนโมเดล ยิ่งไปกว่านั้น SD ยังต้องการทรัพยากรคอมพิวเตอร์เพื่อใช้ในการสร้างภาพและประมวลผลค่อนข้างสูงเมื่อเทียบกับวิธีอื่น



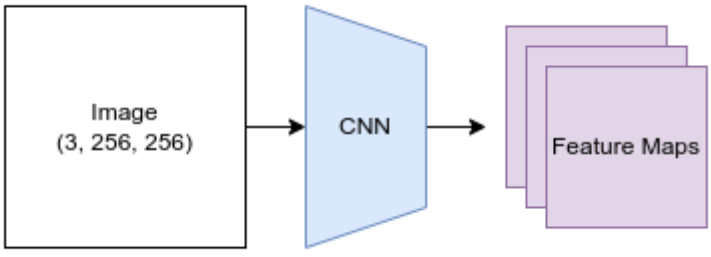
ภาพที่ 2-12 โครงข่าย Stable Diffusion (SD)

[22]

2.5 Perceptual loss

ในกระบวนการตรวจสอบความถูกต้องแบบดั้งเดิมมักมีการใช้ Mean Absolute Error (L1 loss) หรือ Mean Square Error (L2 loss) เป็นวิธีการเปรียบเทียบเพื่อหาความแตกต่างระหว่างภาพที่ทำการแต่งเติม กับภาพเฉลย แต่วิธีการที่กล่าวไปนั้น มักเจาะจงไปในระดับพิกเซลของภาพ (Pixel-wise) ไม่ได้มองภาพรวมของทั้งภาพ

งานวิจัย [23] นำเสนอการเปลี่ยนภาพสไตล์จากภาพหนึ่งไปสู่อีกภาพหนึ่ง โดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) ที่ผ่านการเรียนรู้จากชุดข้อมูลขนาดใหญ่ โดยใช้ Feature maps ซึ่งเป็นผลลัพธ์จากชั้นคอนโวลูชัน (Convolution) ที่มีความสามารถในการสกัดคุณลักษณะ (Feature extraction) ตามที่แสดงในภาพที่ 2-13



ภาพที่ 2-13 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN)

[24]

ซึ่ง Feature maps นี้สามารถนำมาคำนวณความแตกต่างระหว่างภาพที่มีการแต่งเติม กับภาพเฉลย โดยการใช้วิธีการนี้จะทำให้การเปรียบเทียบความแตกต่างในภาพรวมทั้งหมดได้ สามารถคำนวณได้ตามสมการที่ 2-2

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2}\sum_{i,j}\left(F_{ij}^{l} - P_{ij}^{l}\right)^2 \quad (2\text{-}2)$$

จากสมการ $\vec{\mathbf{p}}$ และ $\vec{\mathbf{x}}$ แทนภาพต้นฉบับ และภาพที่แต่งเติม $\mathbf{x}_{\mathbf{ij}}^{\mathbf{l}}$ คือ Feature maps ของฟิลเตอร์ที่ $\mathbf{i}^{\mathbf{th}}$ ที่ตำแหน่ง $\mathbf{j}$ ในชั้น (layer) $\mathbf{l}$ ของภาพต้นฉบับ $\mathbf{P}$ และภาพที่แต่งเติม $\mathbf{F}$ ตามลำดับ โดยการเปรียบเทียบความแตกต่างทางด้านสไตล์ของทั้ง 2 ภาพ ทำได้โดยใช้ squared-error loss กับ feature maps ของทั้ง 2 ภาพ

2.6 Style loss

ในการแต่งเติมรูปภาพนั้น นอกจากรายละเอียดของภาพต้องเหมือนกับภาพเฉลยแล้ว สี และสไตล์ของภาพต้องเหมือนกับภาพฉลยด้วยเพื่อให้ส่วนต่อระหว่างส่วนของภาพที่เติม กับส่วนของภาพต้นฉบับสามารถเข้ากันได้อย่างลงตัว

เพื่อคำนวณหาความแตกต่างทางสไตล์ของภาพ งานวิจัย [25] เสนอวิธีการคำนวณโดยใช้ Gram matrices ที่เอา Channel C จำนวน n_c มาทำการแผ่ (Flatten) ให้อยู่ในรูป [N,C,H,W] ซึ่งจะส่งผลให้คุณลักษณะเฉพาะอยู่ในขนาด [N x C, W x H] หลังจากนั้นจึงนำไปคูณกับ transpose ของตัวเองเพื่อให้ได้ Gram matrix ซึ่งคือค่าความสัมพันธ์ระหว่างคุณลักษณะ (feature correlation) ของภาพตามสมการที่ 2-3

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l} \quad (2\text{-}3)$$

ตามสมการ $\mathbf{G}_{\mathbf{ij}}^{\mathbf{l}}$ คือ Gram matrix ที่มาจากการคำนวณผลคูณภายใน (Inner Product) ระหว่างเวกเตอร์ Feature maps $\mathbf{i}$ และ $\mathbf{j}$ ในชั้น (layer) l

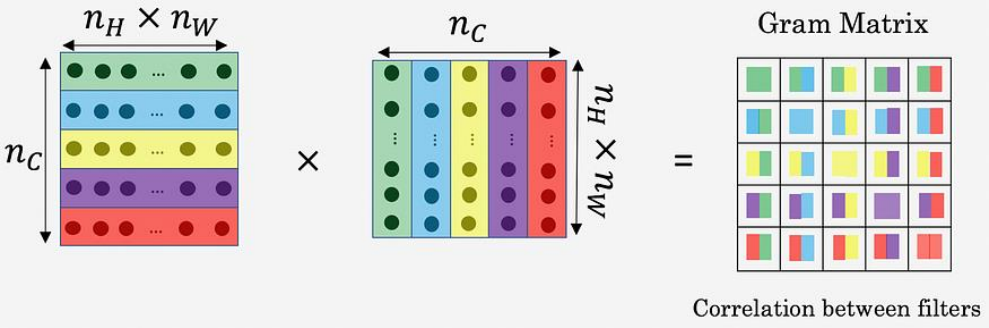
$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}\left(G_{ij}^{l} - A_{ij}^{l}\right)^2 \quad (2\text{-}4)$$

จากสมการที่ 2-4 $\mathbf{E}_\mathbf{l}$ คือระยะห่างระหว่างสไตล์ของภาพในชั้น (layer) ที่ $\mathbf{l}$ ที่มี feature maps จำนวน $\mathbf{N}_\mathbf{l}$ โดยแต่ละ Feature maps มีขนาด $\mathbf{M}_\mathbf{l}$, เมื่อ $\mathbf{M}_\mathbf{l}$ มาจากความกว้าง $\mathbf{n}_\mathbf{W}$ คูณด้วยความสูง $\mathbf{n}_\mathbf{H}$ ของ feature map ของภาพต้นฉบับ $\mathbf{A}_\mathbf{l}$ และของภาพสังเคราะห์ $\mathbf{G}_\mathbf{l}$

ในการคำนวณหาระยะห่างทางด้านสไตล์อาจใช้ feature map จากหลาย ๆ ชั้น (layer) มาช่วยคำนวณ สมการที่ใช้ในการเปรียบเทียบด้านสไตล์นั้นจึงมาจาก การรวมทุก ๆ ระยะห่างระหว่างสไตล์ของทุกชั้น (layer) คูณด้วยสัมประสิทธิ์ความสำคัญของแต่ละชั้น (weighted factor) ตามสมการที่ 2-5

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_t E_t \quad (2\text{-}5)$$

จากสมการ $\vec{a}$ และ $\vec{x}$ คือภาพต้นฉบับและภาพที่ผ่านการแต่งเติม w_l คือสัมประสิทธิ์ความสำคัญของแต่ละชั้น (weighted factor) วิธีการคำนวณความแตกต่างด้านสไตล์สามารถแสดงเป็นภาพที่มีรายละเอียดดังภาพที่ 2-14



ภาพที่ 2-14 รายละเอียดการคำนวณความแตกต่างด้านสไตล์

[26]

2.7 Total variation denoising

การปรับปรุงภาพโดยใช้ค่าความผันแปรทั้งหมด (Total Variation) [27] นั้นตั้งอยู่บนแนวคิดที่ว่า ภาพที่มีลักษณะเป็นสีขาวหรือดำลงอย่างกระทันหันคือภาพที่มีสัญญาณรบกวนอยู่มาก ซึ่งส่งผลให้มีค่าความผันแปรทั้งหมดสูงขึ้นไปด้วย โดยค่าความผันแปรทั้งหมดนิยามได้ตามสมการ

$$V(y) = \sum_{i,j} \sqrt{|y_{i+1,j} - y_{i,j}|^2 + |y_{i,j+1} - y_{i,j}|^2} \qquad (2\text{-}6)$$

โดยที่ $V(y)$ คือ ค่าความผันแปรทั้งหมด

จากสมการที่ 2-6 สามารถอธิบายกระบวนการทำงานของสมการได้ว่า ค่าความผันแปรทั้งหมดนั้นหาได้จากการหาผลรวมความแตกต่างของจุดสีที่อยู่โดยรอบ ซึ่งจากสมการนี้ เมื่อนำไปใช้กับภาพที่มีสีเดียวกันทั้งภาพ ค่าความผันแปรที่ได้จะมีค่าเป็นศูนย์ เนื่องจากไม่มีความแตกต่างของค่าสีภายในแต่ละจุดของภาพ ส่วนในกรณีที่ภายในภาพมีสีที่สลับกันไปมา ค่าความผันแปรทั้งหมดจะมีค่ามาก สมการหาความผันแปรทั้งหมดนี้ สามารถนำไปเพื่อการปรับปรุงคุณภาพได้ตามสมการที่ 2-7

$$\min_{y}(E(x,y) + \lambda V(y)) \qquad (2\text{-}7)$$

สมการที่ 2-7 เป็นสมการที่ใช้ควบคุมอัลกอริทึมการทำงานของการปรับปรุงคุณภาพของภาพ โดยที่ค่า $E(x,y)$ เป็นตัววัดค่าความแตกต่างระหว่างทั้งสองภาพ ในงานวิจัยนี้อาจกล่าวได้ว่าภาพที่ว่านี้ได้แก่ ภาพเฉลย และภาพสังเคราะห์ที่ทำการเติมภาพแทนที่วัตถุเป้าหมาย โดยใช้ Mean Squared Error เพื่อวัดค่าความแตกต่างของภาพ ส่วนค่า λ เป็นค่าตัวแปรควบคุมความสำคัญของค่าความผันแปรทั้งหมดที่มีต่อสมการควบคุม ที่ผู้ใช้นำไปปรับใช้เอง เพื่อให้มีความเหมาะสมกับความต้องการของผู้ใช้แต่ละคนสำหรับงานแต่ละงาน เมื่อค่า $\lambda = 0$ นั่นคือไม่เกิดกระบวนการปรับปรุงคุณภาพของภาพไม่ขึ้นกับค่าความผันแปรทั้งหมด เมื่อปรับให้ค่า λ มีค่าสูงขึ้น คือการ

กำหนดให้มีการปรับปรุงภาพโดยขึ้นกับค่าความผันแปรทั้งหมดมากขึ้น เป้าหมายในการใช้ค่าความผันแปรทั้งหมด คือ การปรับปรุงใช้ภาพที่มีค่าความผันแปรทั้งหมด $V(y)$ น้อยที่สุด และค่าความแตกต่างระหว่างภาพ $E(x, y)$ น้อยที่สุดเช่นเดียวกัน

ภาพที่ 2-15 ภาพก่อนและหลังจากกระบวนการการปรับปรุงภาพโดยใช้ค่าความผันแปรทั้งหมด (Total Variation)
[28]

2.8 งานวิจัยที่เกี่ยวข้อง

2.8.1 การเติมภาพด้วยการแทนที่ (patch-based approach)

เป็นเทคนิคการเติมภาพโดยใช้ข้อมูลจากบริเวณอื่นหรือจากภาพอื่นๆ ที่มีความเกี่ยวข้องกับภาพวัตถุเป้าหมายที่ต้องการลบจากนั้นทำการคัดลอกข้อมูลดังกล่าวมาเติมในส่วนที่ถูกลบเพื่อให้รูปภาพเกิดความสมบูรณ์ ข้อดีของวิธีการเติมภาพด้วยวิธีการแทนที่คือให้ผลลัพธ์สมบูรณ์และเป็นธรรมชาติเนื่องจากส่วนของภาพวัตถุเป้าหมายจะถูกแทนที่ด้วยภาพพื้นหลังจริง

งานวิจัย [2] ได้นำเสนอวิธีการลดเวลาในการค้นหาข้อมูลที่มีอยู่ในภาพเพื่อคัดลอกมาเติมส่วนที่ถูกลบ โดยใช้ความหนาแน่นของความเข้มสี (isophote) เพื่อจัดลำดับความสำคัญของข้อมูลเพื่อหาข้อมูลในภาพที่มีความเป็นไปได้ว่าจะเป็นส่วนที่หายไปมาเติมส่วนที่ถูกลบ

งานวิจัย [3] ได้นำเสนอวิธีการตรวจสอบความเข้ากันได้ของข้อมูลจากทิศทางของพิกเซลโดยการค้นหาข้อมูลที่มีความเป็นไปได้ที่จะนำมาเติมส่วนที่ถูกลบต้องสามารถผสานเข้ากับภาพส่วนที่ไม่ถูกลบได้อย่างแนบเนียน นอกจากนี้งานวิจัยนี้ยังได้นำเสนอการเลือกข้อมูลหลายขนาด เพื่อลดเวลาการค้นหาข้อมูลภายในภาพ ในกรณีที่ส่วนที่ถูกลบมีขนาดใหญ่

งานวิจัย [4] ได้นำเสนอวิธีการค้นหาข้อมูลในภาพจากการแบ่งที่มาของข้อมูลที่นำมาใช้ในการเติมภาพเป็นหลายกลุ่ม โดยพื้นที่แต่ละกลุ่มไม่ซ้อนทับกัน ซึ่งแต่ละพื้นที่ใช้การจัดลำดับความเป็นไปได้ตามความหนาแน่นของโครงสร้างของข้อมูล ทำให้การค้นหาข้อมูลแต่ละครั้ง สามารถทำได้ในหลายพื้นที่พร้อม ๆ กัน และเติมภาพได้อย่างรวดเร็วมากขึ้น

ข้อจำกัดของวิธีการเติมภาพดังกล่าวคือ ไม่เหมาะกับภาพที่มีรายละเอียดพื้นหลังที่ซับซ้อน ยิ่งไปกว่านั้นการค้นหาภาพพื้นหลังที่เกี่ยวข้องเพื่อนำมาใช้แทนที่ภาพวัตถุเป้าหมาย อาจไม่สามารถทำได้ง่ายนัก โดยเฉพาะอย่างยิ่งในระบบจัดทำแผนที่ชนิดเคลื่อนที่ จำเป็นต้องทราบพิกัดตำแหน่งของวัตถุเป้าหมายเพื่อค้นหาภาพที่เกี่ยวข้อง

2.8.2 การเติมภาพด้วยการเทคนิคการเรียนรู้เชิงลึก (Deep learning-based approach)

เป็นเทคนิคการเติมภาพโดยการสร้างเนื้อหาภาพขึ้นใหม่ตามบริบท (context) ของภาพ ข้อดีของวิธีการเติมภาพดังกล่าวคือ ไม่ต้องการข้อมูลพื้นหลังอ้างอิง และสามารถสร้างภาพที่มีรายละเอียดซับซ้อน โดยขึ้นอยู่กับความสามารถของโมเดลโครงข่ายประสาทเทียมเชิงลึกที่ถูกนำมาใช้

งานวิจัย [5] ได้นำเสนอวิธีการในการใช้โครงข่าย Unet ในการเติมภาพ โดยใช้ภาพที่ต้องการเติม พร้อมภาพที่ระบุตำแหน่งของพื้นที่ที่ต้องการเติมภาพ ในการฝึกสอนโมเดลให้เติมภาพส่วนที่ขาดหาย และคงภาพส่วนที่สมบูรณ์เอาไว้ ผลลัพธ์ที่ได้จากงานวิจัยนั้นออกมามีความสมจริง พร้อมทั้งได้พิสูจน์ว่าโมเดลนี้สามารถใช้กับพื้นที่ที่ต้องการเติมภาพได้หลากหลายขนาด และหลากหลายรูปแบบ

งานวิจัย [6] ได้นำเสนอวิธีการในการลบวัตถุจากภาพดาวเทียมโดยใช้ Conditional GAN พร้อมรูปภาพที่ปิดวัตถุที่ต้องการลบจนมิดเพื่อทำการลบขอบของวัตถุนั้นออกจากภาพ แล้วตรวจจับขอบของวัตถุภายในภาพ หลังจากนั้นจึงนำภาพที่ได้จากการตรวจจับขอบของวัตถุไปเข้าโมเดล Pix2PixHD ที่ฝึกการแปลงภาพขอบของวัตถุเป็นรูปภาพดาวเทียม เพื่อแปลงเป็นรูปภาพดาวเทียมซึ่งไม่ปรากฏวัตถุที่ต้องการลบภายในรูปภาพ

งานวิจัย [7] ได้นำเสนอวิธีการในการเติมภาพ 2 ขั้นตอนเริ่มจากการเติมภาพแบบหยาบ (coarse) หลาย ๆ รูปที่มีผลลัพธ์ในการเติมภาพที่เป็นไปได้ ต่อจากนั้นจึงปรับแต่งให้ผลลัพธ์มีความละเอียดมากขึ้น โดยใช้โมเดล Hierarchical VQ-VAE (Vector Quantized Variational Auto-Encoder) เพื่อป้องกันการเติมภาพที่มีรูปแบบการเติมซ้ำ ๆ แม้ว่าข้อมูลในรูปภาพนั้นมีความแตกต่างกัน (Posterior collapse) พร้อมทั้งมีความสามารถในการแยกข้อมูลโดยรวมจากทั้งรูปภาพและข้อมูลเฉพาะส่วนที่ต้องการเติมภาพออกจากกัน

งานวิจัย [8] ได้นำเสนอวิธีการการลบวัตถุโดยใช้โมเดล Denoising Diffusion Probabilistic ที่สามารถทำการเติมภาพได้จากภาพที่ระบุตำแหน่งบริเวณที่ต้องการเติมภาพในหลากหลายขนาดและรูปร่าง ใช้หลักการคล้ายคลึงกับการฟื้นฟูรายละเอียดจากรูปภาพ (image reconstruction) ซึ่งเริ่มด้วยการทำลายรูปภาพ แล้วฟื้นฟูภาพที่เสียหายกลับไปเป็นภาพต้นฉบับทำวนซ้ำไปมา ทำให้การเติมภาพในงานวิจัยนี้ได้ผลลัพธ์ที่มีความหลากหลาย แต่ใช้เวลาในการประมวลช้ากว่าการเติมภาพโดยใช้เทคนิคการเรียนรู้เชิงลึกแบบอื่น ๆ

งานวิจัย [9] นำเสนอการเติมภาพโดยการใช้มุมมองของจอแสดงผลแบบสวมศีรษะ (Head Mounted Display) โดยแบ่งภาพ Equirectangular ที่มีความเสียหายทั่วทั้งภาพออกเป็นหลาย ๆ ส่วน ร่วมกับภาพจาก E-to-V projection (Viewport) เพื่อหาความสัมพันธ์จากการเติมภาพตามขอบเขตการมองเห็นในแนวราบ (Latitude) ร่วมกับหาความสัมพันธ์ระหว่าง Viewport กับภาพที่มาจากการแบ่งภาพ Equirectangular เพื่อให้การเติมภาพมีความเรียบเนียน นอกจากนี้ยังมีการตรวจสอบความสม่ำเสมอของการเติมภาพจากมุมมอง 60 องศาที่เป็นขอบเขตการมองเห็นของมนุษย์ และ 120 องศาที่เป็นขอบเขตของจอแสดงผลแบบสวมศีรษะ

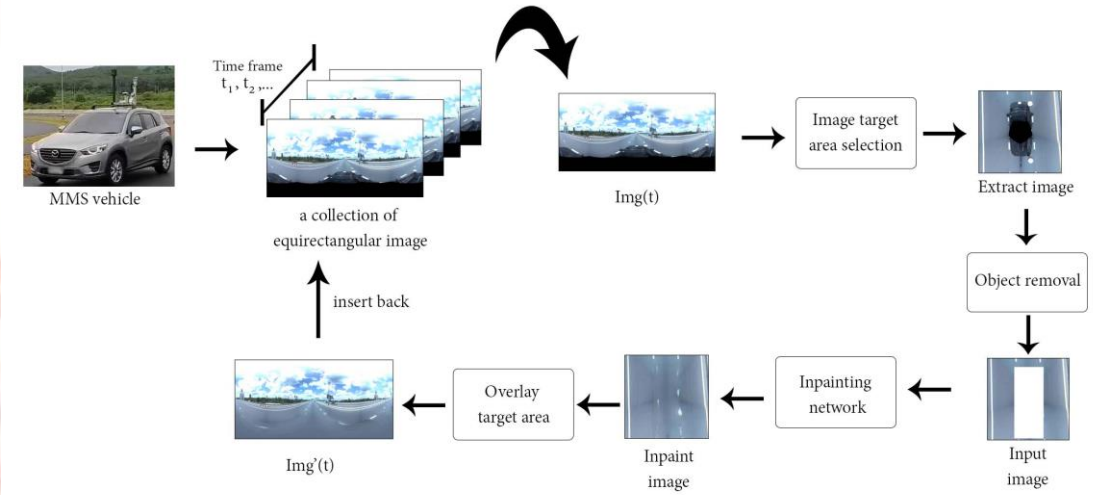
งานวิจัย [10] ได้นำเสนอวิธีการในการเติมภาพพาโนรามาโดยทำการแปลงรูปจาก Equirectangular เป็นรูปพาโนรามาทรงลูกบาศก์ เพื่อใช้กับโครงข่ายเจเนอเรทีฟแอดเวอเซอเรียล (Generative Adversarial Network) ที่ประกอบไปด้วย โครงข่ายประสาทเทียมย่อย 2 ตัว ได้แก่ Generator ที่มีหน้าที่ในการเติมภาพโดยใช้ และ Discriminator ที่มีหน้าที่ในการตรวจสอบความถูกต้องในการเติมภาพในแต่ละหน้าลูกบาศก์ และความต่อเนื่องของรูปภาพระหว่างหน้าลูกบาศก์ การเติมภาพวิธีการนี้สามารถทำได้โดยใช้ภาพที่ระบุตำแหน่งของบริเวณภาพที่ขาดหายไปเพื่อให้ Generator ได้เรียนรู้วิธีการเติมภาพ แต่เมื่อบริเวณที่ภาพขาดหายคาบเกี่ยวระหว่างหน้าลูกบาศก์ จะทำให้การเติมภาพมีความบิดเบี้ยว ซึ่งแสดงให้เห็นถึงความไม่ต่อเนื่องภายในภาพ ทำให้ส่งผลต่อคุณภาพของรูปภาพโดยรวม

ข้อจำกัดของวิธีการดังกล่าวคือ ต้องการชุดข้อมูลภาพจำนวนมากเพื่อใช้ฝึกโมเดล ยิ่งไปกว่านั้นคุณภาพของผลลัพธ์มักขึ้นอยู่ความสามารถของโมเดลโครงข่ายประสาทเทียมที่ถูกนำมาใช้

บทที่ 3
วิธีดำเนินงานวิจัย

3.1 ภาพรวมของงานวิจัย

งานวิจัยนี้ได้นำเสนอวิธีการลบภาพวัตถุเป้าหมาย ซึ่งวัตถุเป้าหมายที่กำหนดในงานวิจัยนี้คือยานพาหนะที่ใช้ในการสำรวจ โดยในขั้นตอนแรกจะต้องทำการเลือกส่วนของภาพที่มีวัตถุเป้าหมาย (Image target selection) ก่อนที่จะทำการลบภาพวัตถุเป้าหมาย (Object removals) แล้วนำเข้าโมเดล GAN เพื่อทำการเติมภาพในส่วนที่นำวัตถุเป้าหมายออกจากภาพ (Inpainitng network) เนื่องจากการเติมภาพนั้นอยู่ในส่วนของภาพที่มีวัตถุที่ต้องการเท่านั้น เพื่อให้ได้รูปพาโนรามาที่จะนำไปใช้งานต่อ จึงจำเป็นต้องทำการแทนที่ส่วนของภาพที่ถูกเติมลงไปบนภาพพาโนรามา (Overlay target area) ก่อนนำไปใช้งานต่อไป



ภาพที่ 3-1 ภาพรวมของกระบวนการเติมภาพ

3.1.1 การเลือกส่วนของภาพที่วัตถุเป้าหมายปรากฏ (Image target area selection)

เนื่องจากระบบ MMS นั้นเก็บข้อมูลภาพถ่ายพื้นที่สำรวจอยู่ในรูปแบบภาพพาโนรามาชนิด Equirectangular เมื่อพิจารณาตำแหน่งของรถสำรวจที่อยู่ด้านล่างของภาพพาโนรามาที่มีลักษณะถูกยืดออกไปตามด้านยาวของภาพพาโนรามา ซึ่งภาพที่ถูกยืดออกนี้มีความบิดเบี้ยวและท้าทายในการเติมภาพ [10] ให้ใกล้เคียงกับลักษณะของพื้นถนน เพื่อให้สามารถเลือกส่วนของภาพพาโนรามาที่วัตถุเป้าหมายปรากฏ จึงต้องทำการแปลงภาพจาก Equirectangular ที่อยู่ในระบบพิกัดฉาก (Cartesian coordinate system) ให้อยู่ในระบบพิกัดทรงกลม (Spherical coordinate system) โดยใช้สมการ (3-1) (3-2) และ (3-3)

$$r = \sqrt{x^2 + y^2 + z^2} \quad (3\text{-}1)$$

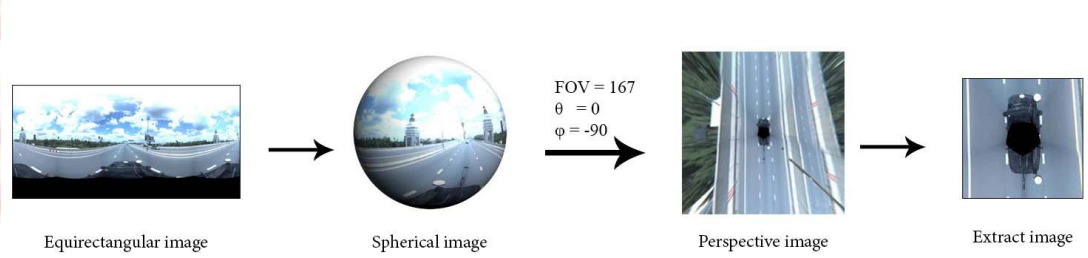
$$\theta = tan^{-1}\left(\frac{y}{x}\right) \quad (3\text{-}2)$$

$$\varphi = cos^{-1}\left(\frac{z}{r}\right) \quad (3\text{-}3)$$

ในการเลือกส่วนขอภาพที่ปรากฏวัตถุเป้าหมาย ต้องกำหนดค่าพิกัด ได้แก่ ค่ามุมมองการมองเห็น (Field of View) เป็นค่าที่กำหนดมุมของภาพที่ส่วนของภาพที่ต้องการ ค่ามุมเงย (φ) ซึ่งจะอยู่ในช่วง $0 \leq \varphi \leq \pi$ และค่าทิศ (θ) ที่อยู่ในช่วง $0 \leq \theta \leq 2\pi$ ซึ่งในงานวิจัยนี้ได้ทำการกำหนดพิกัด ค่ามุมเงย (φ) คือ -90 ซึ่งเป็นพิกัดด้านล่างของภาพพาโนรามา ค่าทิศ (θ) คือ 0 ซึ่งเป็นทิศที่ทำให้วัตถุเป้าหมายหันไปทางด้านหน้า

เพื่อรับประกันว่าภาพวัตถุเป้าหมายและภาพของถนนที่อยู่รอบข้างของวัตถุเป้าหมายมีความบิดเบี้ยวน้อยที่สุด ในงานวิจัยนี้ได้ตั้งค่าส่วนค่ามุมมองการมองเห็น (FOV) อยู่ที่ 167 องศา พร้อมกับตั้งค่าขนาดของภาพกำหนดเป็น 1000 x 1000 พิกเซล

เมื่อคำนึงถึงเวลาที่ใช้ในการประมวลผลของโมเดลเติมภาพ (Inpainting network) หากขนาดของ Input มีขนาดใหญ่ ช่วงเวลาที่ใช้ในการเติมภาพก็จะใช้เวลานานเช่นเดียวกัน เพื่อลดภาระการประมวลผล พร้อมทั้งเห็นภาพของวัตถุเป้าหมายทั้งหมดพร้อมพื้นที่รอบข้างของวัตถุเป้าหมาย จึงได้เลือกเฉพาะส่วนที่มองเห็นวัตถุเป้าหมายพร้อมทั้งภาพแวดล้อมที่แสดงถึงเอกลักษณ์ของถนน คือ เส้นถนน โดยงานวิจัยนี้เลือกใช้ภาพขนาด 256 x 256 พิกเซล



ภาพที่ 3-2 การเลือกส่วนของภาพที่วัตถุเป้าหมายปรากฏ

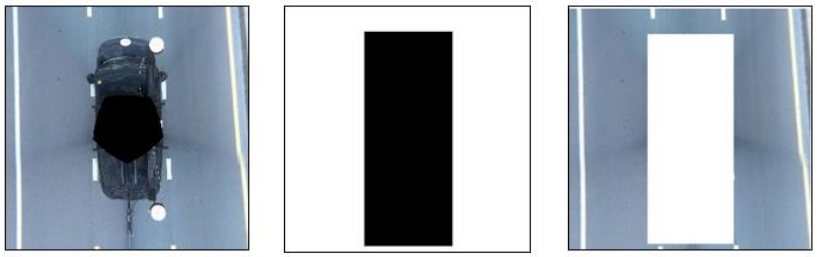
3.1.2 การลบภาพวัตถุเป้าหมาย (Object removals)

เมื่อได้มุมมองที่มองเห็นวัตถุเป้าหมายและภาพแวดล้อมแล้ว ต่อมาต้องทำการลบภาพวัตถุเป้าหมายออกจากภาพ ในงานวิจัยนี้ในส่วนของรถสำรวจนั้น ตำแหน่งของวัตถุเป้าหมายมีตำแหน่งแน่นอน การลบวัตถุเป้าหมายออกจากภาพสามารถคำนวณได้ดังสมการ (3-4)

$$x = (I_{in} \odot mask\ ) + [1 \odot (1 - mask)] \quad (3\text{-}4)$$

โดย $\mathbf{I_{in}}$ คือ ภาพจากขั้นตอนที่ 3.1.1 ที่ผ่านการ Normalize ค่าสีจากช่วง [0,255] ให้อยู่ในช่วง [0,1] ส่วน $\mathbf{mask}$ คือภาพที่แทนที่พิกเซลของวัตถุที่ต้องการลบด้วยพิกเซลสีดำ (พิกเซลสีดำแทนค่าด้วยค่า 0) ส่วนพิกเซลอื่นนั้นแทนที่ด้วยสีขาว (พิกเซลสีขาวแทนที่ที่ด้วยค่า 1) และส่วน $\mathbf{x}$

คือภาพหลังจากใช้ mask ในการลบส่วนพิกเซลสีดำออกจาก $\mathbf{I_{in}}$ ภาพในกระบวนการนี้แสดงในภาพที่ 3-3 ตามลำดับ

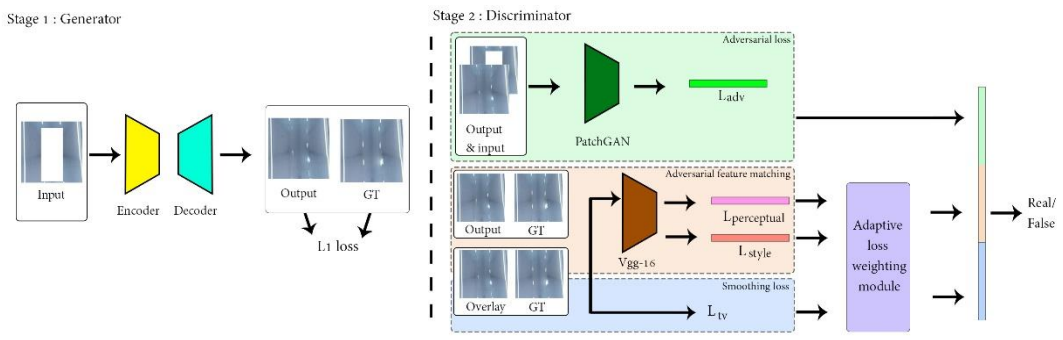


ภาพที่ 3-3 การลบภาพวัตถุเป้าหมาย

3.1.3 การเติมภาพแทนที่ส่วนที่ถูกลบ (Image inpainting)

ในการฝึกสอนโมเดลเติมภาพนั้นใช้การเรียนรู้แบบมีผู้สอน (Supervised learning) โดยใช้ภาพจากการเติมภาพวิธี patch-based ที่ใช้ส่วนของภาพก่อนที่รถสำรวจวิ่งผ่านพื้นที่เป้าหมายอ้างอิงตามพิกัด GPS

การเติมภาพนั้นทำโดยใช้โครงข่ายเจเนอเรทีฟแอดเวอเซอเรียลแบบมีเงื่อนไข (cGAN) งานวิจัยนี้ได้นำโครงสร้างของ Pix2Pix [29] ซึ่งประกอบด้วย Generator (**G**) ที่รับข้อมูลภาพที่ทำการลบวัตถุเป้าหมายจากขั้นตอนที่ 3.1.2 มาเพื่อผ่านกระบวนการเติมภาพ หลังจากนั้นจะใช้โครงข่าย Discriminator (**D**) ที่มีหน้าที่ในการพยายามตรวจจับภาพที่ Generator สร้าง จากภาพเฉลย (Ground truth) โดยใช้โครงสร้างที่มีชื่อว่า PatchGAN ที่มีความสามารถในแบ่งการแยกแยะภาพที่ถูกสร้างออกจากภาพเฉลยเป็นส่วน ๆ (patch) เป็นจำนวน $\mathbf{N_P}$ x $\mathbf{N_P}$ ภายใน 1 ภาพ โดยสมการที่โมเดลที่ใช้ในการเรียนรู้โดยใช้ PatchGAN เป็นไปตามสมการที่ (3-5)



ภาพที่ 3-4 โมเดลที่ใช้การเติมภาพแทนที่ส่วนที่ถูกลบ

$$L_{PGAN} = \mathbb{E}_{x,y}\left[\sum_{i=1}^{N_p} log\, D(x_i, y_i)\right] + \mathbb{E}_{x,z}\left[\sum_{i=1}^{N_p} log\left(1 - D\big(x, G(x,z)\big)\right)\right] \quad (3\text{-}5)$$

นอกจากนี้ Pix2Pix ยังเสนอการใช้ L1 Loss หรือ Mean Absolute Error ร่วมด้วยเพื่อเพิ่มประสิทธิภาพในการสร้างภาพของ Generator ให้ได้ผลลัพธ์ที่มีคุณภาพของภาพที่ดีขึ้น ตามสมการที่ (3-6)

$$L_G = \min_G \max_D L_{PGAN}(G,D) + \lambda_{L1}\mathcal{L}_{L1}(G) \quad (3\text{-}6)$$

เพื่อพัฒนาประสิทธิภาพในการเติมภาพ งานวิจัยนี้ได้นำเสนอวิธีการปรับปรุงประสิทธิภาพในการแยกระหว่างภาพสังเคราะห์กับภาพจริง จากการปรับปรุงสมการในการเรียนรู้ของ Discriminator ในสมการที่ (3-6) การใช้ความสามารถในการสกัดคุณลักษณะคุณลักษณะเฉพาะของภาพของ Deep learning (Feature extraction) มาช่วยในการดึงคุณลักษณะของภาพที่มีความสำคัญ ในการแยกภาพสังเคราะห์กับภาพจริงออกจากกัน โดยพิจารณาถึงความคล้ายคลึงเชิงความหมาย (semantic similarity) เพื่อช่วยให้ Generator มีความสามารถในการเติมภาพที่เหมือนจริงมากยิ่งขึ้น ตามสมการที่ (3-7)

$$\mathcal{L}_{perceptual} = \sum_{p=0}^{P-1} \frac{\left\| f_p^{I_{out}} - f_p^{I_{gt}} \right\|_1}{N_{f_p^{I_{gt}}}} + \sum_{p=0}^{P-1} \frac{\left\| f_p^{I_{over}} - f_p^{I_{gt}} \right\|_1}{N_{f_p^{I_{gt}}}} \quad (3\text{-}7)$$

จากสมการที่ (3-7) $\mathbf{I_{out}}$ คือภาพผลลัพธ์ของโมเดลเติมภาพ ส่วน $\mathbf{I_{over}}$ คือภาพ $\mathbf{I_{out}}$ ไปแทนที่ $\mathbf{I_{in}}$ เฉพาะในส่วนที่เติมภาพ โดยสมการ Perceptual loss นี้ใช้วัดค่าความต่างของภาพวัดจากการรับรู้ของคน เหมาะกับการวัดความแตกต่างของภาพที่มีความคล้ายกัน ไม่ได้เจาะจงในการใช้เพียงความแตกต่างกันในระดับพิกเซลเท่านั้น โดยใช้ความสามารถของ ImageNet pretrained VGG16 model ที่ถูกฝึกสอนให้สามารถแยกคุณลักษณะพิเศษของภาพ (High-level features) โดยใช้ L1 distance หรือ Mean Absolute Error ในการคำนวณความแตกต่างระหว่าง feature map โดย $\mathbf{f_p^{I_*}}$ คือ feature map ของ layer ที่ p ของ VGG model จาก $\mathbf{I_{out}}$, $\mathbf{I_{over}}$ และ $\mathbf{I_{gt}}$ ตามลำดับ โดย layer ที่นำ feature map มาใช้คือ block1_conv2, block2_conv2, block3_conv3, และ block4_conv3 ตามงานวิจัย [23]

นอกจากนี้งานวิจัยยังได้เพิ่มความสามารถในการฟื้นฟูภาพในเชิงสไตล์ (Style reconstruction loss) เพื่อให้การเติมภาพมีผลลัพธ์ออกมาในสไตล์เดียวกับภาพจริง ตามสมการที่ (3-8)

$$\begin{aligned}\mathcal{L}_{style} = &\sum_{p=0}^{P-1} \frac{1}{C_p C_p} \left\| K_p \left((f_p^{I_{out}})^T (f_p^{I_{out}}) - (f_p^{I_{gt}})^T (f_p^{I_{gt}}) \right) \right\|_1 + \\ &\sum_{p=0}^{P-1} \frac{1}{C_p C_p} \left\| K_p \left((f_p^{I_{over}})^T (f_p^{I_{over}}) - (f_p^{I_{gt}})^T (f_p^{I_{gt}}) \right) \right\|_1\end{aligned} \quad (3\text{-}8)$$

จากสมการที่ (3-8) $\mathbf{f_p^{I^*}}$ มีขนาด $\mathbf{(H_pW_p)} \times \mathbf{C_p}$ คือความสูง ความกว้าง และจำนวน channel ของ feature map $\mathbf{f_p^{I^*}}$ ตามลำดับ ทำให้ได้ Gram matrix ในขนาด $\mathbf{C_p} \times \mathbf{C_p}$ ส่วน $\mathbf{K_p}$ คือ ค่าที่ใช้ในการ Normalization feature map ในที่นี้คือ $\mathbf{1/C_pH_pW_p}$ ในแต่ละ layer ที่ $\mathbf{p}$ สมการนี้ถูกใช้กับทั้งภาพผลลัพธ์จากโมเดล $\mathbf{I_{out}}$ และภาพจากการแทนที่พื้นที่ของวัตถุเป้าหมายด้วยผลลัพธ์จากโมเดล $\mathbf{I_{over}}$ เช่นเดียวกันกับสมการที่ (3-7)

ส่วนสมการสุดท้ายที่ใช้ในงานวิจัยนี้คือ Total variation (TV) loss เพื่อลบ noise ออกจากภาพซึ่งส่งผลให้การเติมภาพได้ผลลัพธ์ที่เรียบเนียนมากขึ้น ตามสมการที่ (3-9)

$$\mathcal{L}_{tv} = \sum_{(i,j)\in R,(i,j+1)\in R} \frac{\left\| I_{over}^{i,j+1} - I_{over}^{i,j} \right\|_1}{N_{I_{over}}} + \sum_{(i,j)\in R,(i,j+1)\in R} \frac{\left\| I_{over}^{i,j+1} - I_{over}^{i,j} \right\|_1}{N_{I_{over}}} \quad (3\text{-}9)$$

จากสมการที่ (3-9) R คือขอบเขตของการขยาย 1 พิกเซลของขอบเขตพื้นที่วัตถุที่ถูกลบ และ $N_{I_{over}}$ คือจำนวนของพิกเซลใน I_{over}

จากสมการที่กล่าวมาทั้งหมดนั้น เพื่อพัฒนาประสิทธิภาพของของ Discriminator ในการแยกระหว่าง Ground truth กับ ภาพสังเคราะห์ งานวิจัยนี้ได้นำเสนอการรวมกันของสมการที่ (3-5) ที่มีวัตถุประสงค์ในการแยกระหว่างภาพจริงและภาพสังเคราะห์ สมการที่ (3-7) ที่ตรวจสอบ Semantic similarity สมการที่ (3-8) ที่ตรวจสอบสไตล์ของภาพ และสมการที่ (3-9) ที่มีหน้าที่ทำให้ส่วนที่เติมภาพนั้นมีความเรียบเนียนเข้ากับส่วนดั้งเดิมของภาพ ได้เป็นสมการที่ (3-10)

$$\mathcal{L}_D = L_{PGAN} + \alpha_{perceptual} * L_{percept} + \alpha_{style} * L_{style} + \alpha_{tv} * L_{tv} \quad (3\text{-}10)$$

แม้ว่าส่วนประกอบของสมการที่ใช้ในการเรียนรู้ของ Discriminator จะถูกเลือกมาเพื่อให้ตรวจสอบภาพที่ถูกสังเคราะห์เป็นไปได้อย่างมีประสิทธิภาพแล้ว การตัดสินใจในการเลือกสัมประสิทธิ์ที่เหมาะสมกับแต่ละส่วนนั้นใช้เวลานาน และสัมประสิทธิ์ที่ได้มาอาจไม่ใช่ค่าที่จะส่งผลดีต่อการเรียนรู้ของโมเดลได้ดีที่สุด

เพื่อให้สามารถตัดสินใจสัมประสิทธิ์ที่เหมาะสม ในงานวิจัยนี้ได้มีการนำเสนอในการใช้ในการใช้ $\boldsymbol{\alpha_k}$ ซึ่งคือสัมประสิทธิ์ไดนามิกของ $\mathbf{L}$ ที่ $\mathbf{k}$ ที่มีหน้าที่ในการปรับสัมประสิทธิ์ให้เหมาะสมโดยขึ้นกับค่าการสูญเสีย (error) ในขณะที่กำลังฝึกสอนโมเดล โดยการใช้ Softadapt [30] ที่คำนวณสัมประสิทธิ์ที่เหมาะสมโดยอ้างอิงจากค่าการสูญเสีย (error) สะสม เมื่อค่าการสูญเสีย (error) ของส่วนประกอบใดในการสมการการเรียนรู้มีค่าใกล้เคียงกับจุดต่ำสุด (minima) โดยคำนวณตามสมการที่ (3-11)

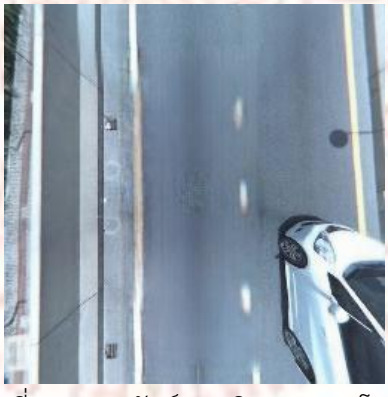
$$\alpha_k^i = \frac{f_k^i e^{\beta s_k^i}}{\sum_{\ell=1}^{n} f_\ell^i e^{\beta s_\ell^i}} \quad (3\text{-}11)$$

จากสมการที่ (3-11) ได้ทำการดัดแปลงสมการ softmax ที่จะให้ความสำคัญจากค่านำเข้าทั้งหมด ในที่นี้คือ s_k^i มาจากการหาค่าความเปลี่ยนแปลงจากการใช้ finite different เพื่อคำนวณอัตราการเปลี่ยนแปลงของค่าการสูญเสียของ f ที่ k ของรอบที่ i โดยหากค่า s_k^i มีค่าสูงแสดงว่าค่าการสูญเสียนั้นให้ประสิทธิภาพแย่ แต่หากว่า s_k^i มีค่าต่ำจะแสดงว่าค่าการสูญเสียนั้นให้ประสิทธิภาพแย่ แล้วทำการหารด้วยอัตราการเปลี่ยนแปลงของค่าการสูญเสียรวมตั้งแต่ $s_1^i, \ldots, s_n^i$

การมอบหมายค่าสัมประสิทธิ์ขึ้นกับตัวแปร β เมื่อ $\beta > 0$ จะมอบหมายค่าสัมประสิทธิ์มาก ให้กับอัตราการเปลี่ยนแปลงของค่าการสูญเสียของ f ที่มีค่ามาก เมื่อ $\beta < 0$ จะมอบหมายค่าสัมประสิทธิ์มาก ให้กับอัตราการเปลี่ยนแปลงของค่าการสูญเสียของ f ที่มีค่าต่ำ เมื่อ $\beta = 0$ จะมอบหมายค่าสัมประสิทธิ์อย่างเท่าเทียมกันให้กับค่าการสูญเสียของ f ทุกค่า

นอกจากการคำนึงถึงอัตราการเปลี่ยนแปลงแล้ว สมการที่ (3-11) มีการเพิ่มค่าการสูญเสีย f_k^i และ ส่วน f_ℓ^i คือ ค่าการสูญเสียของ f รวม ในกรณีค่าการสูญเสียมีอัตราการเปลี่ยนแปลง s_k^i มีค่าสูงแต่มีค่าใกล้เคียงกับจุดต่ำสุด (minima) จะส่งผลให้ค่าสัมประสิทธิ์จำนวนน้อยกับค่าการสูญเสียดังกล่าว

หลังจากฝึกสอนโปรแกรมเสร็จสมบูรณ์แล้ว ส่วนที่นำไปใช้คือ Generator ได้ภาพตามที่แสดงผลตามภาพที่ 3-5



ภาพที่ 3-5 ผลลัพธ์การเติมภาพจากโมเดล

3.1.4 การแทนที่ส่วนของภาพที่ถูกเติมในภาพพาโนรามา (Overlay target area)

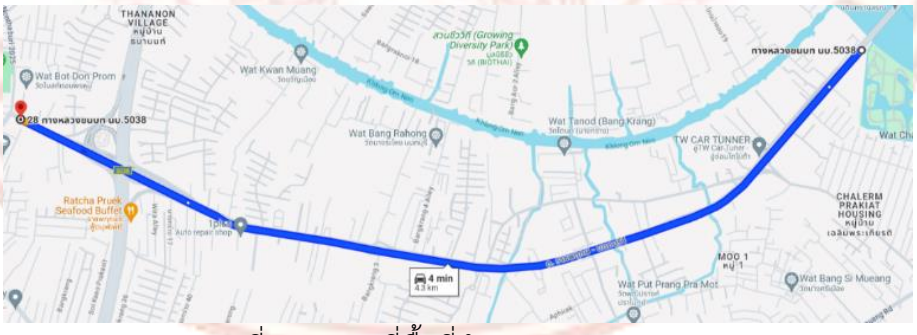
เมื่อผ่านการเติมภาพแล้ว ลักษณะของภาพที่ได้นั้นอยู่ในลักษณะภาพจัตุรัสขนาดความกว้าง 256 และความยาว 256 พิกเซล แต่ภาพที่มีการนำไปใช้งานนั้นอยู่ในลักษณะภาพสี่เหลี่ยมผืนผ้า ความกว้าง 2048 พิกเซล และความยาว 4096 พิกเซล ตามที่กล่าวไปในขั้นตอนที่ 3.1.1 ที่ภาพที่ใช้ในการเติมภาพนั้นมาจากภาพจัตุรัสขนาดความกว้าง 1000 พิกเซล และความยาว 1000 พิกเซล จึงต้องทำการเติมภาพเฉพาะส่วนไปยังภาพขนาด 1000 x 1000 พิกเซล ก่อนที่จะทำการคลี่ออกให้กลับไปอยู่ในรูปแบบภาพพาโนรามาตามภาพที่ 3-6 เพื่อให้สามารถนำไปใช้งานได้ต่อไป

ภาพที่ 3-6 ภาพผลลัพธ์สุดท้าย

3.2 ข้อมูลที่ใช้ในงานวิจัย

3.2.1 Bridge dataset

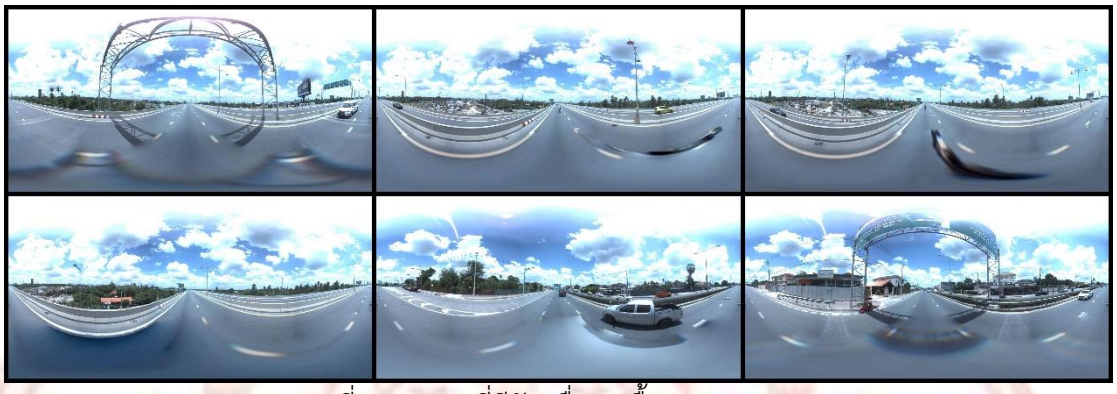
ภาพที่ใช้ในการงานวิจัยนี้มาจากการสำรวจถนนโดยใช้อุปกรณ์กล้อง Ladybug 3 ติดตั้งที่รถสำรวจเพื่อบันทึกภาพพาโนรามาขนาด 2048 x 4096 พิกเซล พร้อมด้วย GNSS คือ ระบบเครือข่ายดาวเทียมนำทางที่โคจรรอบโลก (Global Navigation Satellite System) เพื่อระบุตำแหน่งพิกัดบนพื้นผิวโลก โดยเส้นทางสำรวจเริ่มตั้งแต่สะพานมหาเจษฎาบดินทรานุสรณ์ ถึงวัดโบสถ์ดอนพรหม จังหวัดนนทบุรี ประเทศไทย ระยะทางสำรวจรวมทั้งหมด 4.3 กิโลเมตร ภาพพาโนรามาจำนวนรวม 1969 ภาพพร้อมข้อมูลพิกัด GPS ที่บันทึกในแต่ละพิกัดที่ทำการถ่ายภาพ



ภาพที่ 3-7 แผนที่พื้นที่สำรวจ Bridge dataset

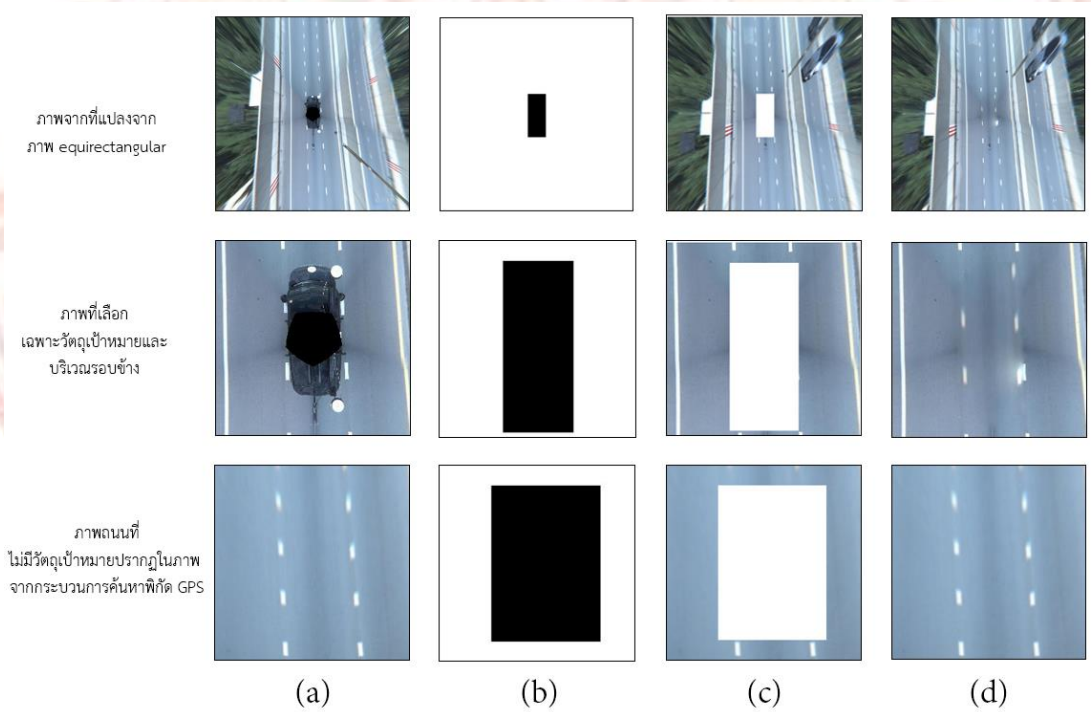
เพื่อฝึกสอนวิธีการที่มีการนำเสนอในงานวิจัยนี้ ผู้วิจัยได้ทำการสร้างภาพเฉลย (Ground truth) ที่มีการลบวัตถุเป้าหมายและเติมภาพสมบูรณ์แล้ว โดยใช้หลักการที่มีการนำเสนอใน [31] ซึ่งเสนอการใช้พิกัด GPS ที่ระบุตำแหน่งของรถสำรวจ ทำให้สามารถรู้ถึงภาพที่เป็นตำแหน่งเดียวกันแต่ไม่มีวัตถุเป้าหมายอยู่ตรงพิกัดนั้น จากการใช้ข้อมูลภาพที่ได้มานี้ทำให้สามารถนำข้อมูลภาพมาแทนบริเวณที่ลบวัตถุเป้าหมายออกไป หลังจากได้ผลลัพธ์จากการใช้หลักการที่กล่าวมา ผู้วิจัยได้ทำการคัด

ภาพที่มีวัตถุอื่นบนพื้นถนน (artifact) ยกตัวอย่างตามภาพที่ 3-8 ได้ดังนี้ ภาพที่พิกัด GPS เป้าหมายมีเงาพาดทับ ภาพที่พิกัด GPS เป้าหมายมีพาหนะอื่นขับผ่าน เป็นต้น ซึ่งตัวอย่างชุดข้อมูลที่ถูกคัดออกนี้ได้แสดงตามภาพด้านล่าง



ภาพที่ 3-8 ภาพที่มีวัตถุอื่นบนพื้นถนน (artifact)

การคัดภาพที่มีความผิดพลาดนี้ออกจะส่งผลให้โมเดลที่ทำการฝึกสอนสามารถเรียนรู้ลักษณะของถนนที่มีความถูกต้อง



ภาพที่ 3-9 ภาพชุดข้อมูล Bridge ที่ในงานวิจัย

จากภาพที่ 3-9 แสดงประเภทของชุดข้อมูลที่ใช้ในงานวิจัยนี้ ในแนวตั้งนั้นจะแสดงประเภทของข้อมูลในแต่ละชุดข้อมูล ประกอบด้วย ภาพนำเข้า (input image) ภาพแสดงจุดที่ลบวัตถุเป้าหมายออกไป (mask image) ภาพที่ทำการลบวัตถุเป้าหมายออกแล้ว (masked image) และ

ภาพเฉลยที่ทำการเติมภาพแทนที่วัตถุเป้าหมายแล้ว (Ground truth) ซึ่งภาพทั้งหมดอยู่ในขนาด 256 x 256 พิกเซล ภาพที่ใช้ในงานวิจัยนี้มีจำนวนรวม 5461 ภาพ

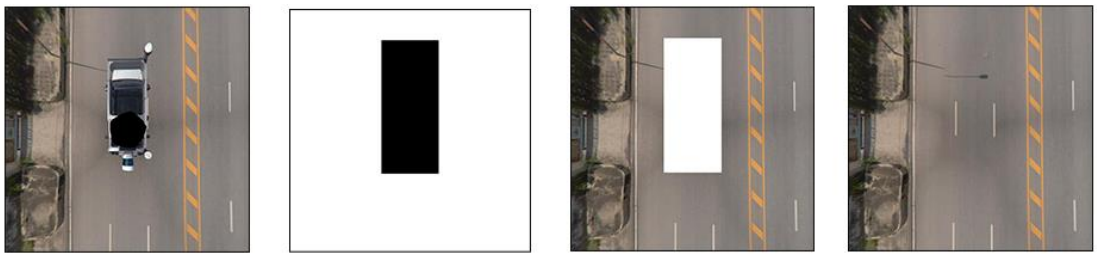
3.2.2 Nakorn dataset

ภาพที่ใช้ในการงานวิจัยนี้มาจากการสำรวจถนนโดยใช้อุปกรณ์กล้อง Ladybug 5 ติดตั้งที่รถสำรวจเพื่อบันทึกภาพพาโนรามาขนาด 4096 x 8192 พิกเซล พร้อมด้วย GNSS คือ ระบบเครือข่ายดาวเทียมนำทางที่โคจรรอบโลก (Global Navigation Satellite System) เพื่อระบุตำแหน่งพิกัดบนพื้นผิวโลก พื้นที่สำรวจคือจังหวัดนครสวรรค์ ประเทศไทย ระยะทางสำรวจรวมทั้งหมด 9.7 กิโลเมตร ภาพพาโนรามาจำนวนรวม 4075 ภาพ พร้อมข้อมูลพิกัด GPS ที่บันทึกในแต่ละพิกัดที่ทำการถ่ายภาพ

ภาพที่ 3-10 แผนที่พื้นที่สำรวจ Nakorn dataset (1)

ภาพที่ 3-11 แผนที่พื้นที่สำรวจ Nakorn dataset (2)

เนื่องจากภาพในชุดข้อมูลนี้มีจำนวนมาก ผู้วิจัยจึงเจาะจงไปที่การใช้ภาพที่เลือเฉพาะเป้าหมายและบริเวณรอบข้างเท่านั้น



ภาพที่ 3-12 ภาพชุดข้อมูล Nakorn ที่ในงานวิจัย

จากภาพที่ 3-12 แสดงประเภทของชุดข้อมูลที่ใช้ในงานวิจัยนี้ ในแนวตั้งนั้นจะแสดงประเภทของข้อมูลในแต่ละชุดข้อมูล ประกอบด้วย ภาพนำเข้า (input image) ภาพแสดงจุดที่ลบวัตถุเป้าหมายออกไป (mask image) ภาพที่ทำการลบวัตถุเป้าหมายออกแล้ว (masked image) และภาพเฉลยที่ทำการเติมภาพแทนที่วัตถุเป้าหมายแล้ว (Ground truth)

เพื่อให้ชุดข้อมูลอยู่ในขอบเขตของการวิจัยได้มีการปรับลดขนาดความกว้าง ความยาวของภาพพาโนรามาจาก 4096 x 8192 พิกเซล เป็น 2048 x 4096 พิกเซล และใช้ขนาดของภาพข้อมูลนำเข้า (input) ให้อยู่ในขนาด 512 x 512 พิกเซล หลังจากคัดภาพที่มีวัตถุอื่นบนพื้นถนน (artifact) ออกไปแล้ว ทำให้ชุดข้อมูลนี้มีจำนวนภาพรวม 4007 ภาพ

ในชุดข้อมูลนี้มีความแตกต่างจาก Bridge dataset ดังต่อไปนี้

ตารางที่ 3-1 ข้อแตกต่างระหว่าง dataset

ข้อแตกต่าง	Bridge dataset	Nakorn dataset
ขนาดของรถสำรวจ	198 x 100 พิกเซล	276 x 100 พิกเซล
ขนาดภาพ Input	256 x 256 พิกเซล	512 x 512 พิกเซล
Hardware specification	Ladybug 3	Ladybug 5
ความคมชัด	ต่ำกว่า	สูงกว่า

3.3 ขั้นตอนการดำเนินการ

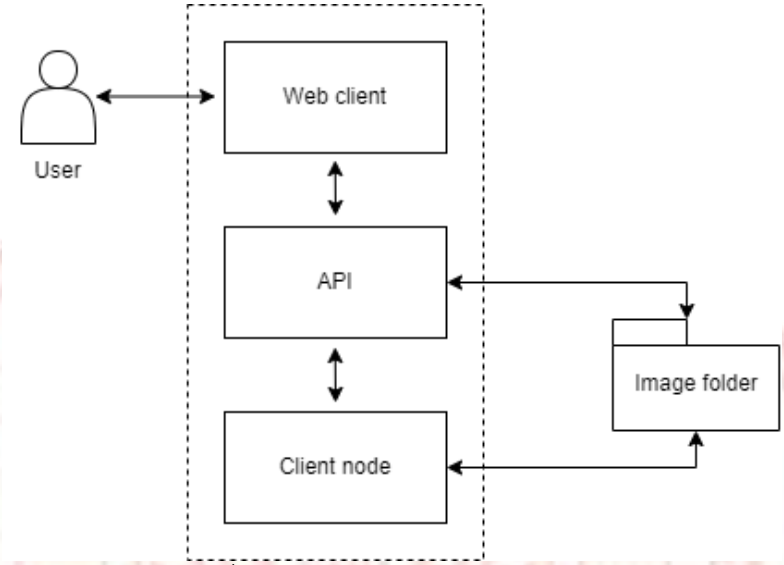
3.3.1 การทดลองวิธีการที่นำเสนอในงานวิจัย

ขั้นตอนการทดลองนั้นจะทำการแบ่งข้อมูลจากทั้งหมดเป็นข้อมูลฝึกสอนโมเดล (Training set) จำนวนร้อยละ 70 ส่วนร้อยละ 30 ที่เหลือจะเป็นส่วนของข้อมูลที่โมเดลไม่เคยเรียนรู้มาก่อน การทดลองนี้จะแบ่งข้อมูลที่เหลือนี้ออกเป็น ข้อมูลที่ใช้ทดสอบผลระหว่างฝึกสอนโมเดล (Validation set) จำนวนร้อยละ 20 และข้อมูลที่ใช้ทดสอบผลหลังจากฝึกสอนโมเดลสมบูรณ์แล้ว (Test set) จำนวนร้อยละ 10 ในระหว่างการฝึกสอนโมเดลนั้นได้ทำการวัดค่าการสูญเสียในการฟื้นคืนภาพ (reconstruction loss) โดยวัดจาก Mean Absolute Error (MAE) loss เพื่อตรวจสอบประสิทธิภาพของ Discriminator โดยการทดลองจะเริ่มจาก Bridge dataset ตามด้วย Nakorn dataset ตามลำดับ

3.3.2 การออกแบบและพัฒนาแอปพลิเคชันลบภาพวัตถุเป้าหมาย

เพื่อนำกระบวนการที่เสนอในงานวิจัยไปใช้งานต่อไป ผู้วิจัยได้ทำการพัฒนาแอปพลิเคชันเพื่อใช้ในการลบภาพวัตถุเป้าหมาย โดยมีรายละเอียดดังนี้

3.3.2.1 ภาพรวมการออกแบบระบบ ระบบที่พัฒนาในงานวิจัยนี้ ประกอบด้วย 3 ส่วน ได้แก่ ส่วนของ Web client ส่วนของ API และส่วนของ Client node ตามภาพที่ 3-13



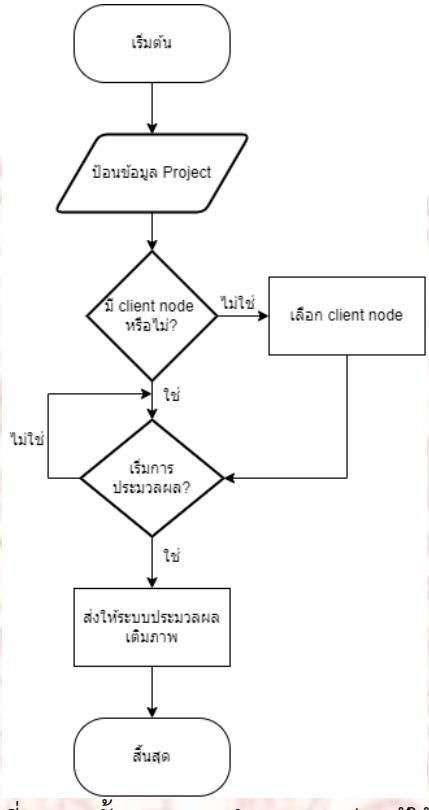
ภาพที่ 3-13 โครงสร้างของระบบโดยรวม

ส่วนของ Web client เป็นส่วนที่ติดต่อกับผู้ใช้งาน เพื่อให้ผู้ใช้งานได้ป้อนข้อมูลที่จำเป็นต่อการลบวัตถุเป้าหมาย ได้แก่ ข้อมูลโฟลเดอร์ที่มีภาพจากการสำรวจ และข้อมูลชื่อโฟลเดอร์ที่จะจัดเก็บผลลัพธ์หลังจากการลบภาพวัตถุเป้าหมาย นอกจากนี้ผู้ใช้งานยังสามารถยกเลิกการประมวลผลผ่านส่วน Web client นี้ได้เช่นกัน

ส่วนของ API เป็นส่วนที่ประสานงานระหว่างส่วนของ Web client และส่วนของ Client node มีหน้าที่ได้แก่ แบ่งงานให้แต่ละ Client node ประมวลผล แจ้งสถานะการประมวลผลไปที่ส่วน Web client

ส่วนของ Client node เป็นส่วนประมวลผลตามกระบวนการที่นำเสนอในงานวิจัยนี้ โดยรับคำสั่งจากส่วนของ API มีหน้าที่ได้แก่ ลบภาพรถสำรวจ บันทึกผลลัพธ์ลงโฟลเดอร์ปลายทาง และแจ้งสถานะการประมวลผลไปที่ส่วนของ API เพื่อแสดงผลให้ผู้ใช้งานต่อไป

3.3.2.2 การออกแบบการทำงานของระบบ

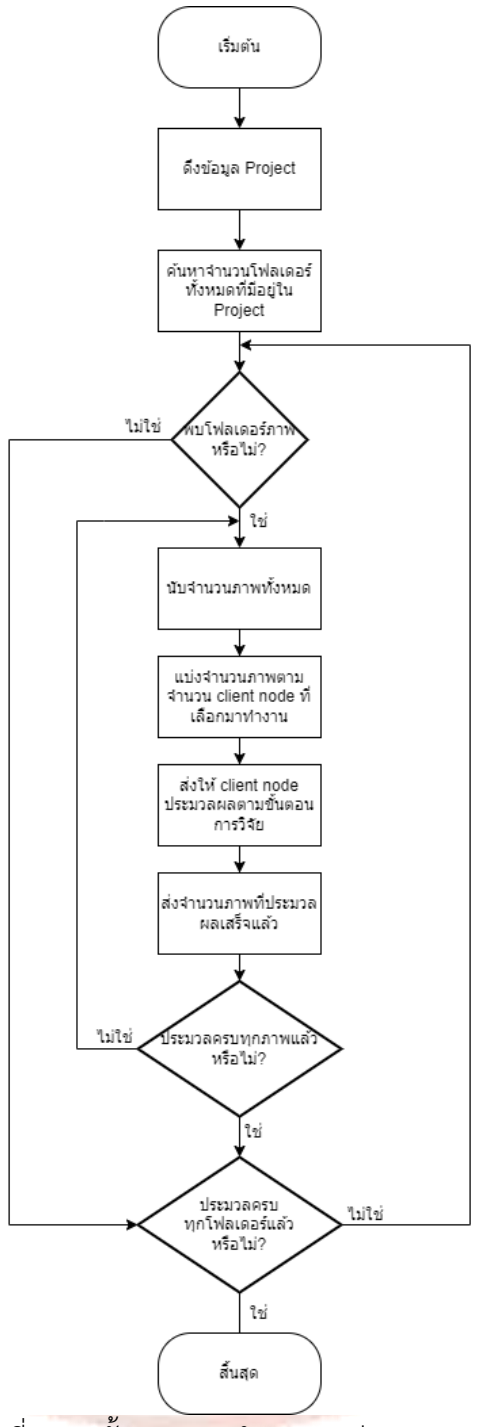


ภาพที่ 3-14 ขั้นตอนการทำงานของส่วนผู้ใช้งาน

เพื่อเริ่มต้นการทำงานผู้ใช้งานต้องกรอกข้อมูลที่จำเป็นต่อการลบวัตถุเป้าหมาย ได้แก่ ข้อมูลโฟลเดอร์ที่มีภาพจากการสำรวจ และข้อมูลชื่อโฟลเดอร์ที่จะจัดเก็บผลลัพธ์หลังจากการลบภาพวัตถุเป้าหมาย และเนื่องจากจำนวนของข้อมูลจากระบบจัดทำแผนที่ชนิดเคลื่อนที่ (MMS) ที่มีปริมาณมาก เพื่อให้การประมวลผลลบภาพรถสำรวจสามารถทำได้อย่างรวดเร็ว ระบบที่พัฒนานี้จึงสามารถเลือก Client node ได้มากกว่า 1 Node

โดยส่วนที่ใช้ติดต่อกับผู้ใช้งานนั้น ผู้วิจัยใช้เว็บแอปพลิเคชัน (Web application) เพื่อความสะดวกต่อการใช้งาน

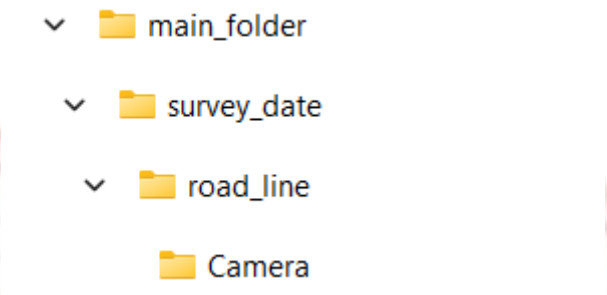
การทำงานในส่วนปนะมวลผลนั้นเริ่มจากเมื่อผู้ใช้งานเริ่มการประมวลผลจาก Web client โดยดำเนินการทำงานตามขั้นตอนที่นำเสนอในงานวิจัยนี้



ภาพที่ 3-15 ขั้นตอนการทำงานของส่วนประมวลผล

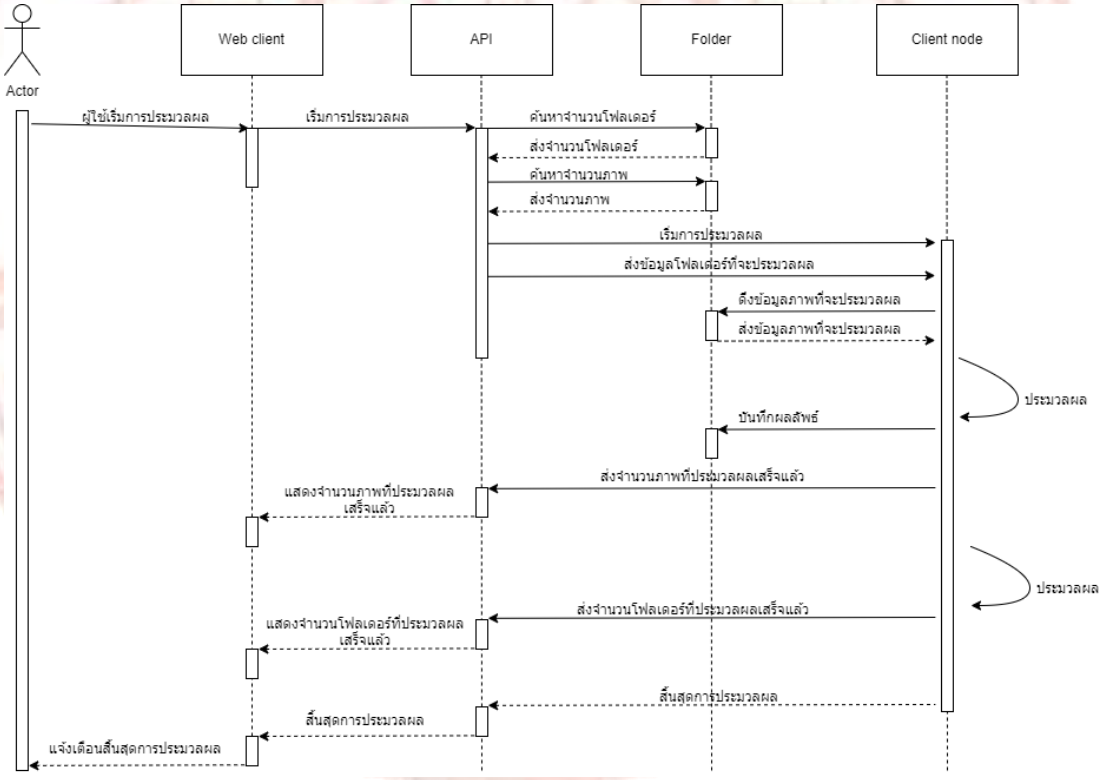
หลังจากได้รับข้อมูลจาก web client ระบบจะทำการค้นหาตามลำดับของโฟลเดอร์ที่แสดงในภาพที่ 3-16 ว่าภายในโฟลเดอร์ต้นทาง (main_folder) ว่าทั้งหมดกี่โฟลเดอร์และมีโฟลเดอร์ใดบ้างที่มีภาพนามสกุลไฟล์ JPG หากพบจะทำการนับจำนวนภาพทั้งหมดเพื่อแบ่งจำนวนภาพที่แต่ละ Client node ต้องทำการประมวลผลในจำนวนเท่า ๆ กัน โดยระหว่างที่

Client node ประมวลผลตามขั้นตอนที่เสนอในงานวิจัยนี้จะแสดงจำนวนภาพที่ประมวลผลเสร็จ เมื่อประมวลผลภายในโฟลเดอร์เสร็จเรียบร้อยแล้ว จะทำการตรวจสอบว่ายังมีโฟลเดอร์ที่ยังไม่ได้ประมวลผลอีกหรือไม่ และทำการประมวลต่อไปจนสำเร็จเรียบร้อย



ภาพที่ 3-16 ลำดับของโฟลเดอร์

การทำงานร่วมกันระหว่างส่วนของ Web client ส่วนของ API และส่วนของ Client node มีการรับ – ส่งข้อมูลระหว่างกันตามที่แสดงในภาพที่ 3-17



ภาพที่ 3-17 Sequence diagram แสดงการประมวลผลของระบบ

บทที่ 4
ผลการดำเนินงานวิจัย

การทดสอบนี้ได้นำแอปพลิเคชันที่ได้ออกแบบมาทำการทดสอบ ในการทดสอบจะใช้ภาพพาโนรามาขนาด 4096 x 2048 พิกเซล ที่ได้จากระบบจัดทำแผนที่ชนิดเคลื่อนที่มาทำการลบวัตถุเป้าหมายออกด้วยวิธีเติมภาพตามวิธีที่นำเสนอในงานวิจัยนี้ โดยแบ่งการทดสอบออกเป็น 2 ส่วน ได้แก่ การทดสอบเวลาที่ใช้ในการประมวลผล และการทดสอบคุณภาพของภาพที่ได้จากการเติมภาพ การทดสอบทั้งหมดนี้ได้ใช้เครื่องคอมพิวเตอร์ Intel (R) Xeon(R) Silver 4214 CPU@ 2.20GHz processor GPU Tesla V100-PCIE-16GB มาใช้ในการทดสอบ

4.1 การทดสอบเวลาที่ใช้ประมวลผล

การทดสอบนี้จะทดสอบระยะเวลาที่ใช้ในการประมวลผลในการเติมภาพ ผลลัพธ์ที่ได้จากการทดสอบ แสดงในตารางที่ 4-1 โดยเปรียบเทียบกับวิธีการวิจัยที่นำเสนอใน [31] แสดงให้เห็นว่าวิธีที่นำเสนอในงานวิจัยนี้ใช้เวลาในการประมวลผลเฉลี่ยน้อยกว่าใน Bridge dataset และใช้เวลาในการประมวลผลเฉลี่ยมากกว่าใน Nakorn dataset ทั้งนี้เวลาที่แสดงนั้นไม่ได้รวมเวลาที่ใช้ในกระบวนการคัดแยกภาพที่ที่มีวัตถุอื่นบนพื้นถนน (artifact) ในวิธีการวิจัยที่นำเสนอใน [31] ในขณะที่การใช้ Pix2Pix model นั้นจะไม่มีการสร้างภาพที่มีวัตถุอื่นบนพื้นถนน (artifact)

ตารางที่ 4-1 ผลการทดสอบเวลาที่ใช้ในการประมวลผล

Approach	Bridge dataset		Nakorn dataset	
	100 images (sec)	1000 images (sec)	100 images (sec)	1000 images (sec)
Patch-based model [31]	362	3260	**374**	**3742**
Pix2Pix model	**304**	**2970**	515	5163

4.2 การทดสอบคุณภาพของภาพที่ได้จากการเติมภาพ

4.2.1 วิธีการที่ใช้ทดสอบ

การทดสอบนี้ได้นำภาพจากการเติมภาพโดยอ้างอิงตำแหน่ง GPS และภาพผลลัพธ์จากวิธีที่นำเสนอในงานวิจัยนี้ โดยตัดภาพเฉพาะบริเวณที่ทำการเติมภาพ ซึ่งมีขนาด 100 x 220 พิกเซล และ 130 x 290 พิกเซล ในชุดข้อมูล Bridge dataset และ Nakorn dataset ตามลำดับ มาทำการเปรียบเทียบจากการใช้เทคนิคการวัดประสิทธิภาพ 3 วิธี ดังนี้

4.2.1.1 Reconstruction loss คือ การหาความแตกต่างกันของภาพในระดับพิกเซล มี 2 ประเภท ได้แก่

Mean Absolute Error (MAE) คือ การหาค่า error เฉลี่ยสัมบูรณ์ของข้อมูลทุกจุดของภาพ หากค่าที่วัดได้มีค่าน้อย แสดงว่าภาพมีความเหมือนกันมาก โดย error ชนิดนี้จะเปรียบเทียบข้อมูลแต่ละจุดในภาพอย่างเท่าเทียมกัน ในงานวิจัยนี้จะใช้วัดประสิทธิภาพระหว่างการฝึกสอนโมเดล

Mean Squared Error (MSE) คือ การหาค่า error เฉลี่ยของข้อมูลทุกจุดของภาพ แล้วนำมายกกำลังสอง หากค่าที่วัดได้มีค่าน้อย แสดงว่าภาพมีความเหมือนกันมาก โดย error ชนิดนี้จะมีความอ่อนไหวต่อ noise ภายในภาพ ทำให้ผลลัพธ์ที่ได้จะมีการให้ความสำคัญกับจุดที่มีความแตกต่างกันเป็นพิเศษ ในงานวิจัยนี้จะใช้วัดประสิทธิภาพผลลัพธ์จากโมเดล

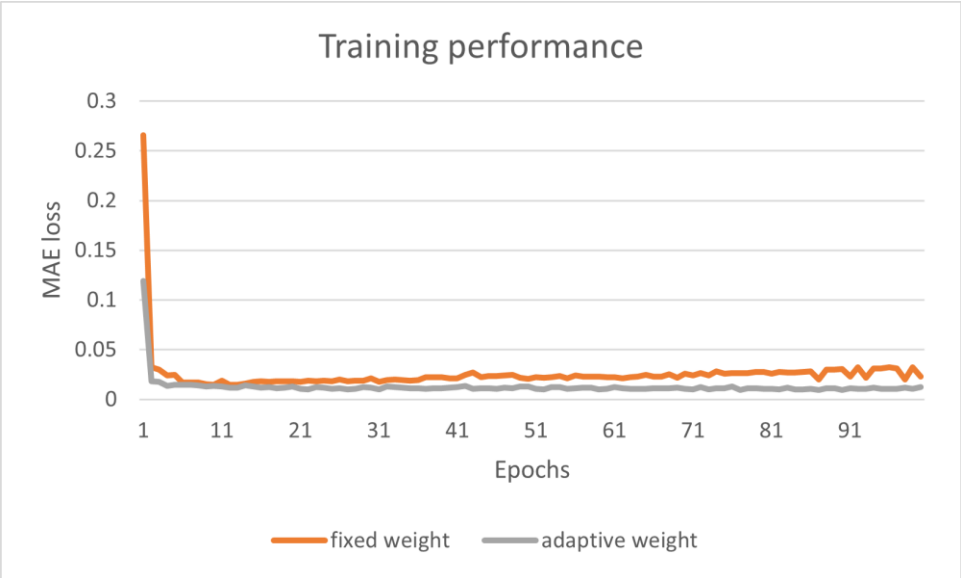
4.2.1.2 Peak signal-to-noised ratio (PSNR) คือ ค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุดของภาพ หากค่าที่วัดได้มีค่ามาก แสดงว่าภาพมีคุณภาพมาก

4.2.1.3 Structural similarity (SSIM) คือ การวัดดัชนีความคล้ายคลึงกันของโครงสร้าง ใช้สำหรับวัดความคล้ายคลึงกันของความสว่าง ความคมชัด และโครงสร้าง ค่าที่ได้จากการวัดจะอยู่ในช่วงระหว่าง 0 ถึง 1 ค่าที่วัดได้มีค่าเข้าใกล้ 1 แสดงว่าภาพมีคุณภาพมาก

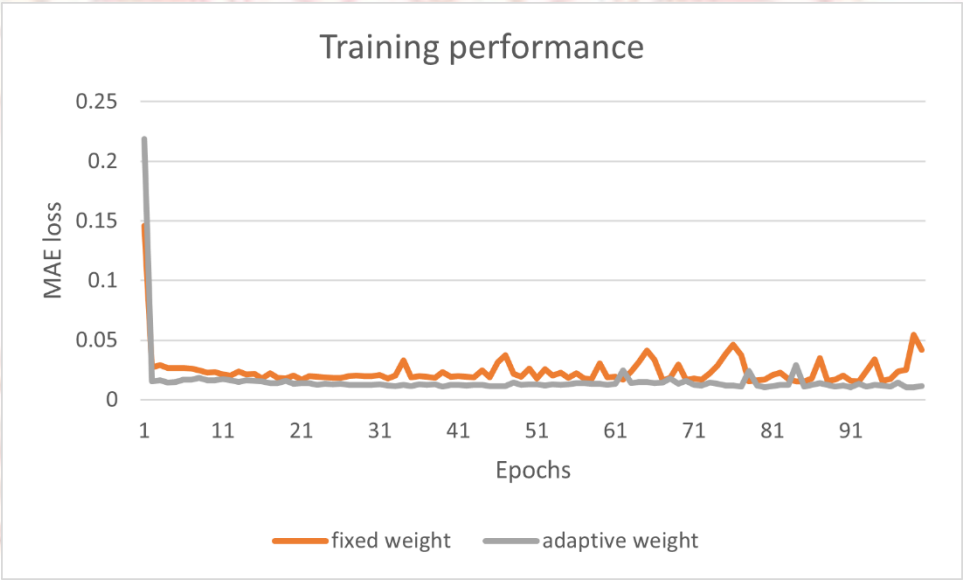
4.2.2 ผลการทดสอบ

4.2.2.1 Stable training

เพื่อทดสอบว่าวิธีการที่นำเสนอในงานวิจัยนี้ สามารถส่งผลให้กระบวนการฝึกสอนโมเดลเป็นไปได้อย่างมั่นคง ตามภาพที่ 4-1 เมื่อทดสอบกับ Bridge dataset จะเห็นได้ว่าค่าการสูญเสียของการฟื้นฟูภาพ (MAE loss) ของวิธีการฝึกสอนโมเดลโดยใช้ค่าสัมประสิทธิ์แบบไดนามิก มีค่าต่ำลงอย่างคงที่ เมื่อเปรียบเทียบกับวิธีการฝึกสอนโมเดลโดยใช้ค่าสัมประสิทธิ์แบบคงที่ และเพื่อทดสอบว่าวิธีที่นำเสนอในงานวิจัยนี้ สามารถใช้ได้ผลดีกับชุดข้อมูลที่มีคุณลักษณะแตกต่างกัน จึงได้มีการทดสอบกับ Nakorn dataset ตามภาพที่ 4-2 ซึ่งแสดงให้เห็นถึงประสิทธิภาพที่มีความใกล้เคียงกับการฝึกสอนโมเดลกับ Bridge dataset



ภาพที่ 4-1 ประสิทธิภาพช่วงขั้นตอนฝึกสอน (Training phase) ของ Bridge dataset

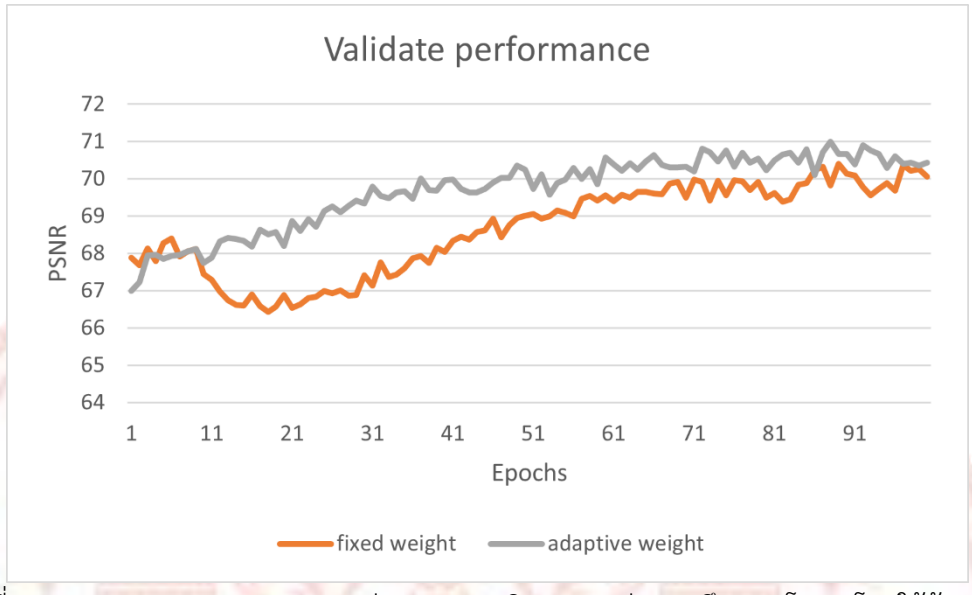


ภาพที่ 4-2 ประสิทธิภาพช่วงขั้นตอนฝึกสอน (Training phase) ของ Nakorn dataset

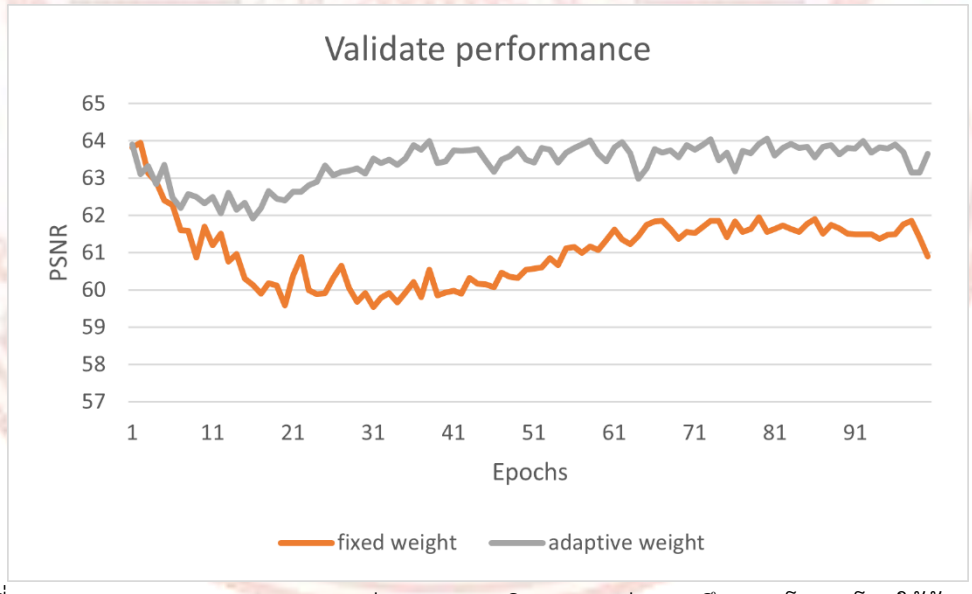
4.2.2.2 Generalization

เพื่อการทดสอบว่าวิธีการที่นำเสนอในงานวิจัยไม่ได้ให้ผลลัพธ์ที่ดีเฉพาะกับชุดข้อมูลเพียงชุดเดียว จึงได้มีการวัดคุณภาพของภาพโดยใช้ PSNR เพื่อวัดสัญญาณรบกวนภายในภาพ จากภาพที่ 4-3 จะเห็นได้ว่า ค่า PSNR ของวิธีการฝึกสอนโมเดลโดยใช้ค่าสัมประสิทธิ์แบบไดนามิกมีค่าสูงขึ้นเรื่อย ๆ เมื่อรอบการฝึกโมเดลมากขึ้น ต่างจากวิธีการฝึกสอนโมเดลโดยใช้ค่าสัมประสิทธิ์แบบคงที่ ที่ช่วงแรกจะมีค่า PSNR ลดลงอย่างมากแล้วจึงเริ่มสูงขึ้น และเพื่อทดสอบว่าวิธีที่นำเสนอในงานวิจัยนี้ สามารถใช้ได้ผลดีกับชุดข้อมูลที่มีคุณลักษณะแตกต่างกันจึงได้มีการทดสอบกับ

Nakorn dataset ตามภาพที่ 4-4 แม้ว่าในช่วงแรกค่า PSNR ลดลงแต่น้อยกว่า เมื่อเปรียบเทียบกับวิธีการฝึกสอนโมเดลโดยใช้ค่าสัมประสิทธิ์แบบคงที่

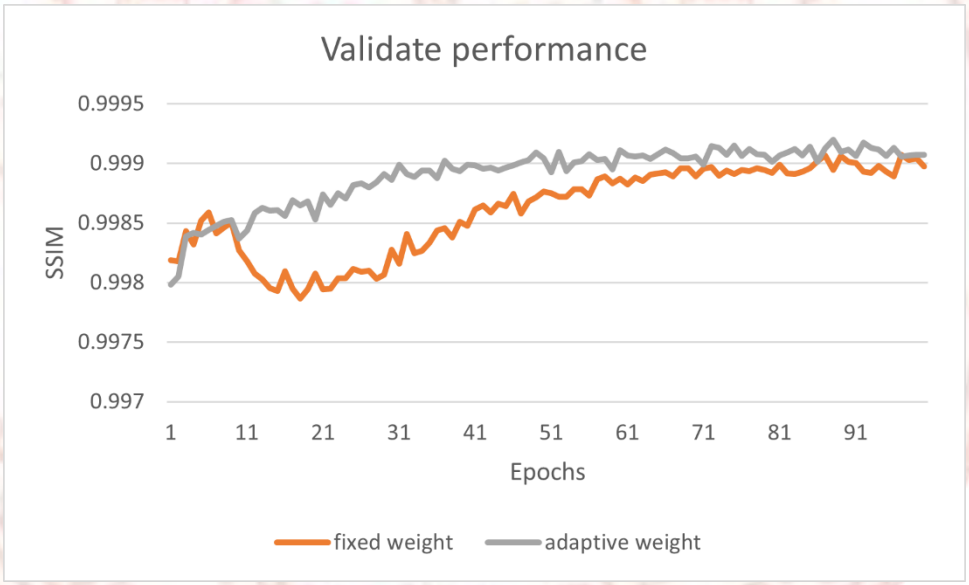


ภาพที่ 4-3 คุณภาพของภาพ (PSNR) ช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (Validate phase) ของ Bridge dataset

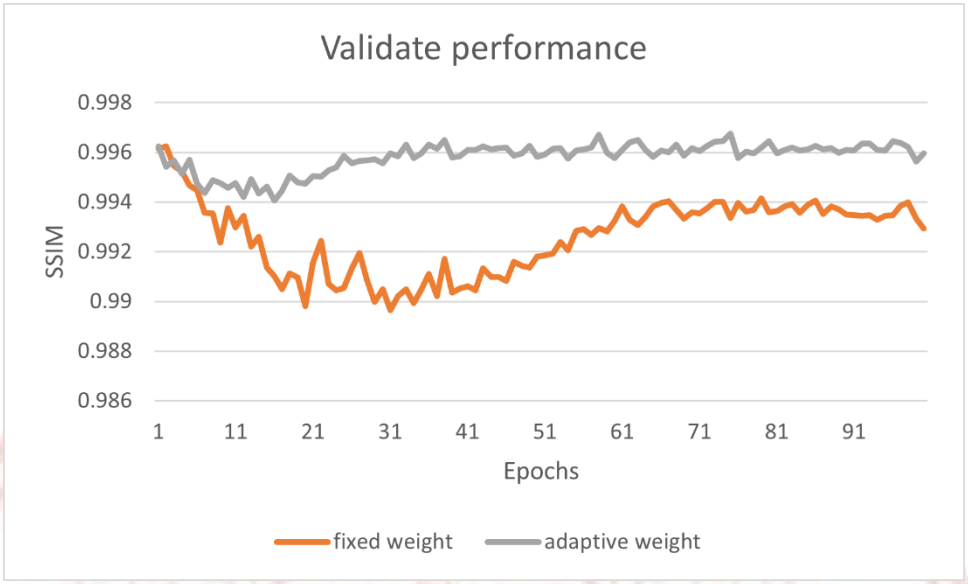


ภาพที่ 4-4 คุณภาพของภาพ (PSNR) ช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (Validate phase) ของ Nakorn dataset

นอกจากนี้ ได้มีการวัดคุณภาพของภาพโดยใช้ SSIM เพื่อวัดคุณภาพเชิงโครงสร้างภายในภาพ จากภาพที่ 4-5 จะเห็นได้ว่า ค่า SSIM ของวิธีการฝึกสอนโมเดลโดยใช้ค่าสัมประสิทธิ์แบบไดนามิก มีค่าสูงขึ้นเรื่อย ๆ เมื่อรอบการฝึกโมเดลมากขึ้น ต่างจากวิธีการฝึกสอนโมเดลโดยใช้ค่าสัมประสิทธิ์แบบคงที่ ที่ช่วงแรกจะมีค่า SSIM ลดลงอย่างมากแล้วจึงเริ่มสูงขึ้นจนสุดท้ายมีค่า SSIM ใกล้เคียงกับวิธีการฝึกสอนโมเดลโดยใช้ค่าสัมประสิทธิ์แบบไดนามิก และเพื่อทดสอบว่าวิธีที่นำเสนอในงานวิจัยนี้ สามารถใช้ได้ผลดีกับชุดข้อมูลที่มีคุณลักษณะแตกต่างกัน จึงได้มีการทดสอบกับ Nakorn dataset ตามภาพที่ 4-6 แม้ว่าในช่วงแรกค่า SSIM ลดลงแต่น้อยกว่าเมื่อเปรียบเทียบกับวิธีการฝึกสอนโมเดลโดยใช้ค่าสัมประสิทธิ์แบบคงที่ และผลลัพธ์ในรอบการฝึกสอนของทั้ง 2 วิธีการยังมีความแตกต่าง โดยที่วิธีการฝึกสอนโมเดลโดยใช้ค่าสัมประสิทธิ์แบบไดนามิกมีคุณภาพของภาพสูงกว่า



ภาพที่ 4-5 คุณภาพเชิงโครงสร้างของภาพ (SSIM) ช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (Validate phase) ของ Bridge dataset



ภาพที่ 4-6 คุณภาพเชิงโครงสร้างของภาพ (SSIM) ช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (Validate phase) ของ Nakorn dataset

4.2.2.3 Baseline comparison

การทดลองนี้จะทำการเปรียบเทียบผลลัพธ์กับวิธีการต่าง ๆ ต่อไปนี้

4.2.2.3.1 Adversarial loss เป็นมาตรฐานที่ใช้ในการเปรียบเทียบกับวิธีที่งานวิจัยนี้นำเสนอ (Baseline model)

$$\mathcal{L}_D = L_{PGAN} \tag{4-1}$$

4.2.2.3.2 Fixed weight เป็นการควบรวมสมการโดยใช้ค่าสัมประสิทธิ์แบบคงที่ โดยแทนที่ α_* ด้วย λ_*จากสมการที่ (3-10) ที่งานวิจัยนี้นำเสนอ ได้ตามสมการที่ (4-2)

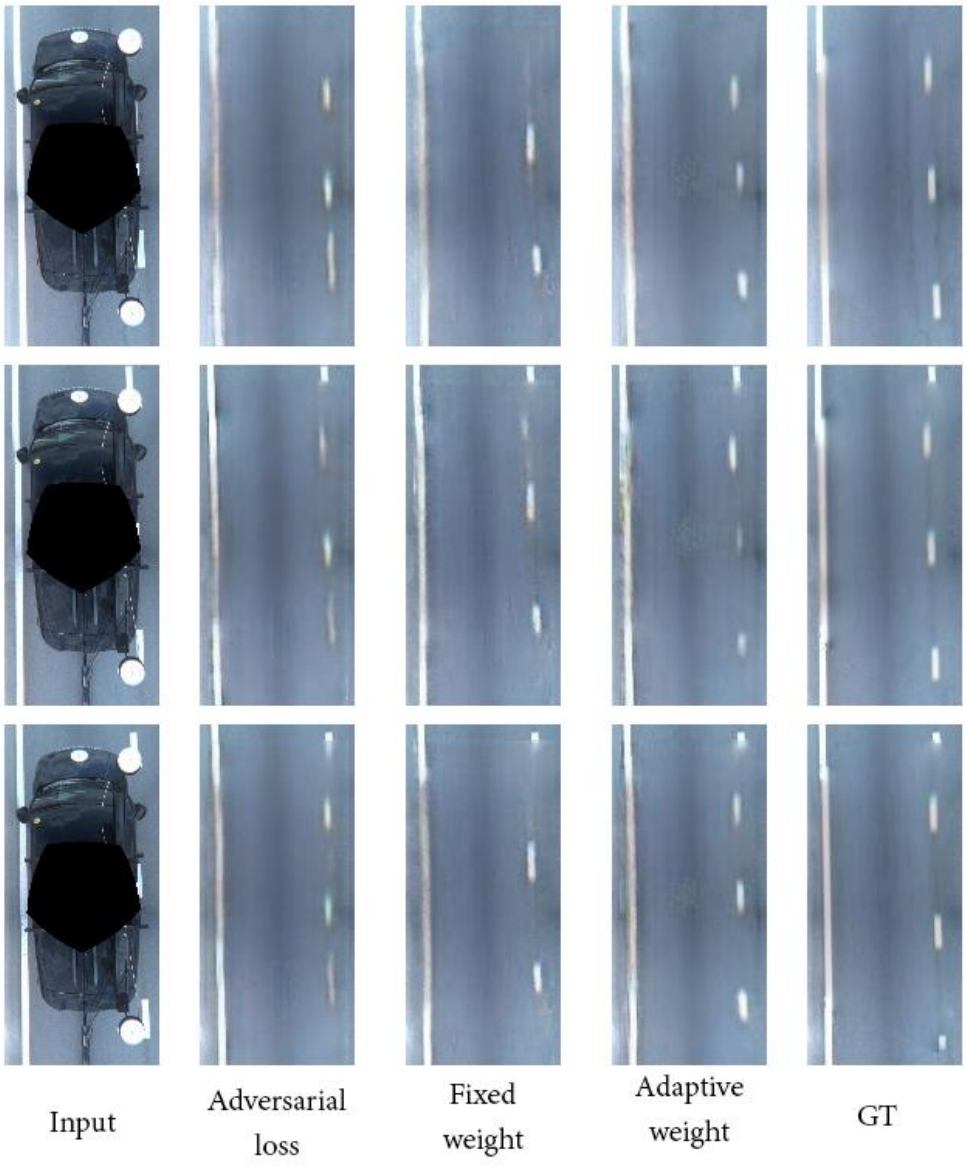
$$\mathcal{L}_D = L_{PGAN} + \lambda_{perceptual} * L_{percept} + \lambda_{style} * L_{style} + \lambda_{tv} * L_{tv} \tag{4-2}$$

4.2.2.3.3 Adaptive weight เป็นการควบรวมสมการโดยใช้ค่าสัมประสิทธิ์แบบคงที่ ที่งานวิจัยนี้นำเสนอ จากสมการที่ (3-10) ซึ่งในบทนี้จะกล่าวถึงตามสมการที่ (4-3)

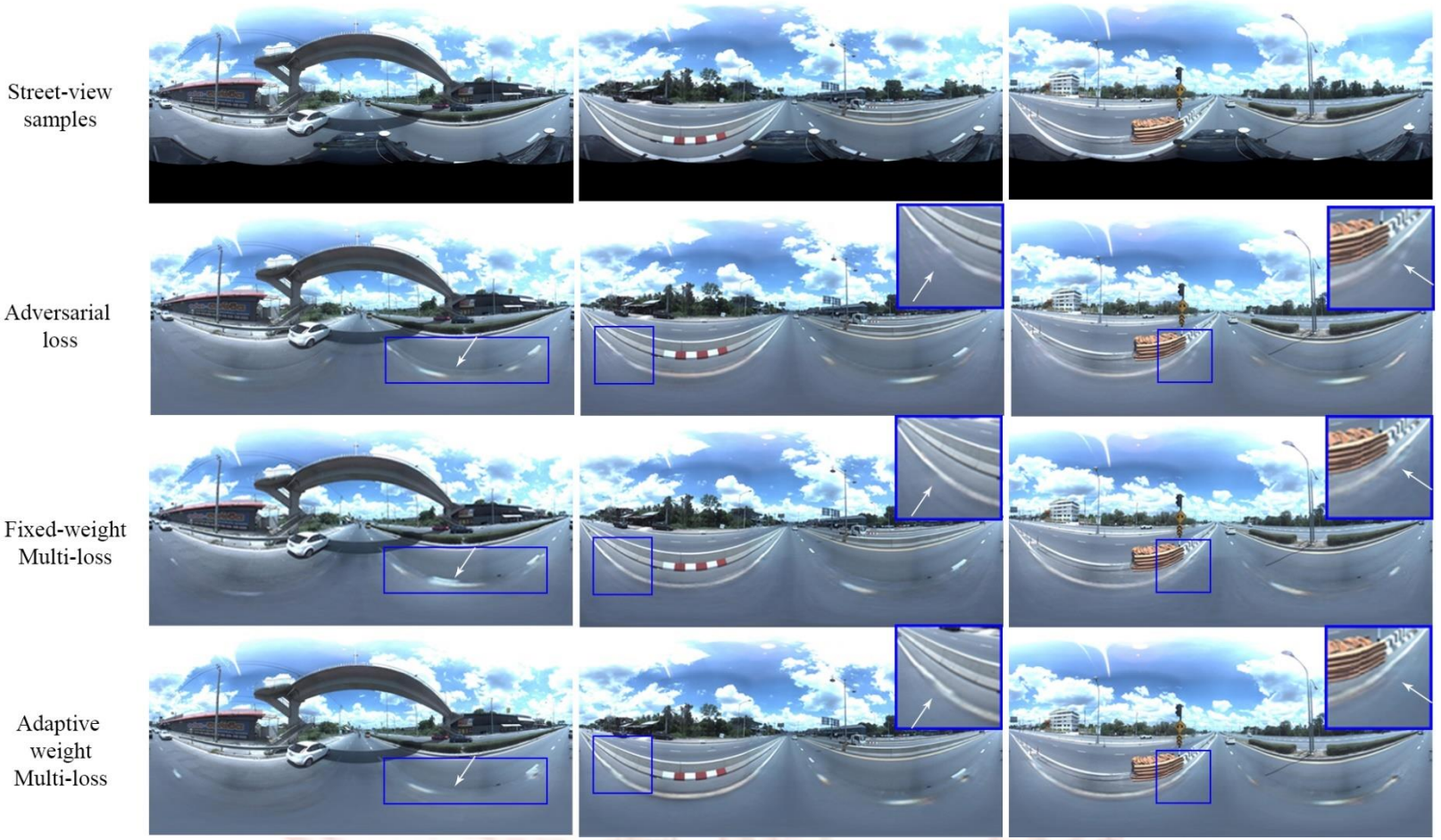
$$\mathcal{L}_D = L_{PGAN} + \alpha_{perceptual} * L_{percept} + \alpha_{style} * L_{style} + \alpha_{tv} * L_{tv} \tag{4-3}$$

เพื่อทำการวัดคุณภาพของวิธีการที่นำเสนอ งานวิจัยนี้ได้ทำการจำกัดขอบเขตของภาพให้อยู่ในพื้นที่เติมภาพเท่านั้นเทียบกับภาพเฉลยจากเทคนิคที่นำเสนอใน [31] ใน Bridge dataset ตามภาพที่ 4-7 แสดงให้เห็นว่าการเพิ่มสมการการเรียนรู้ที่มีการนำเสนอในงานวิจัยนี้ส่งผลให้รายละเอียดของภาพมีความใกล้เคียงกับภาพเฉลยมากกว่า Adversarial loss ซึ่งเป็น Baseline ที่ใช้เทียบกับวิธีการที่นำเสนอในงานวิจัยนี้ นอกจากนี้การเติมเส้นถนนของการใช้ค่าสัมประสิทธิ์แบบไดนามิก (Adaptive weight) นั้นให้ผลลัพธ์ที่มีความคมชัดมากกว่าวิธีอื่น เมื่อทำการเติมภาพทับ (Overlay) ลงบนภาพพาโนรามาในภาพที่ 4-8 ยังแสดงให้เห็นว่าการเติมภาพนั้นเข้ากันได้อย่างกลมกลืนเมื่อเปรียบเทียบกับ Adversarial loss ที่ยังเห็นถึงขอบระหว่างส่วนที่เติมภาพกับส่วนอื่น ๆ ของภาพ

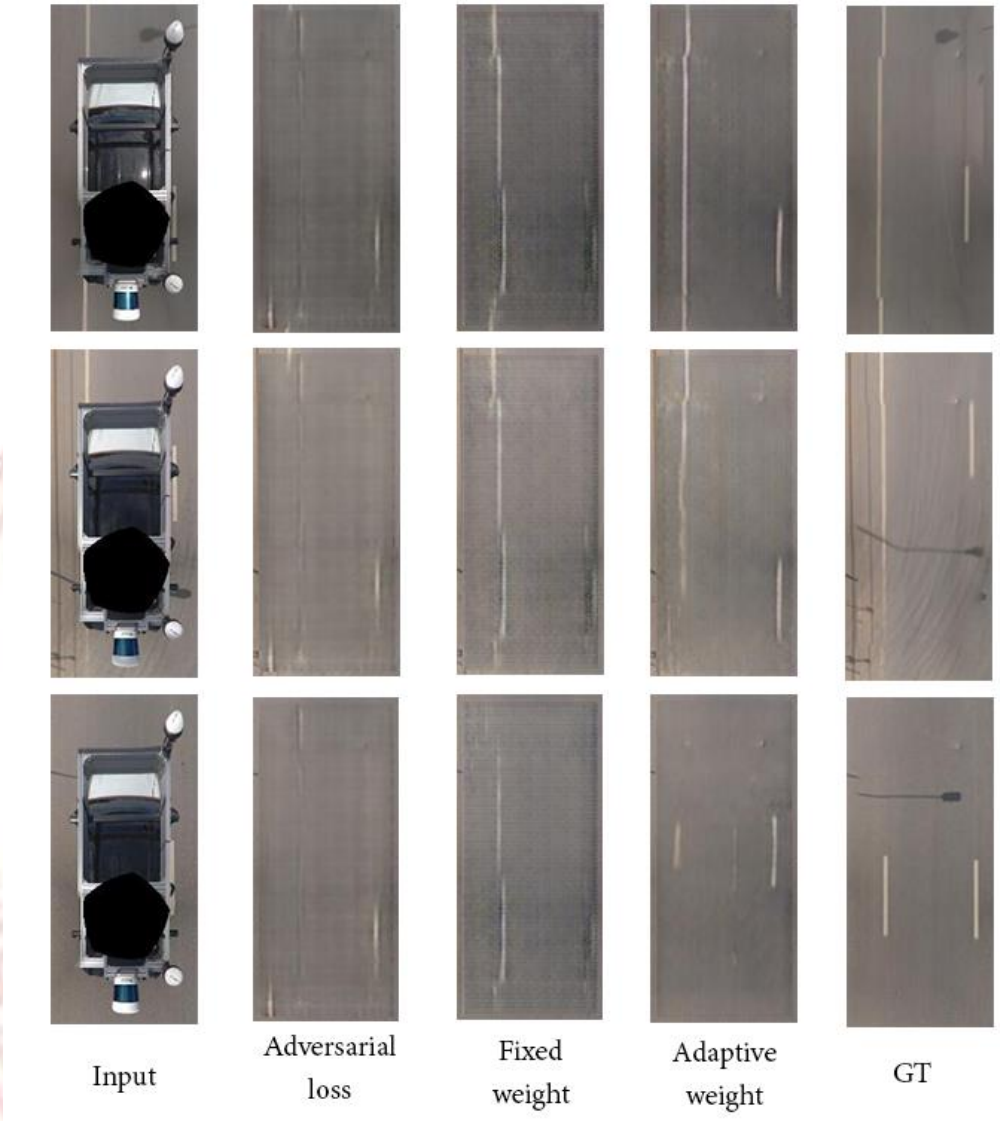
ใน Nakorn dataset ภาพที่ 4-9 แสดงให้เห็นอย่างชัดเจนว่าสมการการเรียนรู้นั้นช่วยให้การเติมรายละเอียดของภาพทำได้ดีมากกว่า Adversarial loss และการควบรวมสมการโดยใช้ค่าสัมประสิทธิ์แบบไดนามิก (Adaptive weight) ส่งผลให้การเติมภาพนั้นมีความใกล้เคียงกับภาพเฉลยมากกว่าการเติมภาพในวิธีการควบรวมสมการโดยใช้ค่าสัมประสิทธิ์แบบค่าคงที่ (Fixed weight) พร้อมทำการเติมภาพทับ (Overlay) ลงบนภาพพาโนรามาในภาพที่ 4-10



ภาพที่ 4-7 ผลลัพธ์การเติมภาพจากการทดลองวิธีการฝึกสอนโมเดลของ Bridge dataset



ภาพที่ 4-8 ผลลัพธ์สุดท้ายจากการทดลองวิธีการฝึกสอนโมเดลของ Bridge dataset



ภาพที่ 4-9 ผลลัพธ์การเติมภาพจากการทดลองวิธีการฝึกสอนโมเดลของ Nakorn dataset

ภาพที่ 4-10 ผลลัพธ์สุดท้ายจากการทดลองวิธีการฝึกสอนโมเดลของ Nakorn dataset

ผลการทดสอบการวัดคุณภาพของภาพที่ได้เปรียบเทียบกับโมเดลมาตรฐาน Adversarial loss แสดงในตารางที่ 4-2 ผลที่ได้จากการวัดคุณภาพโดยใช้วิธี Mean-squared error (MSE) จากวิธีการ Adaptive weight นั้นมีค่าน้อยที่สุด แสดงว่า ผลลัพธ์มีความเหมือนกับภาพเฉลย ส่วนค่าที่ได้จากวิธี Peak signal-to-noise ratio (PSNR) กับวิธี Structural similarity (SSIM) ได้ค่าสูงมากกว่า ทั้งในชุดข้อมูล Bridge dataset และ Nakorn dataset แสดงให้เห็นว่าวิธีการที่นำเสนอไปนั้นสามารถเติมภาพได้คุณภาพสูงกว่าวิธีการอื่น

ตารางที่ 4-2 ผลการทดสอบการวัดคุณภาพของภาพจากการทดลองวิธีการฝึกสอนโมเดล

Method	Bridge dataset			Nakorn dataset		
	MSE (↓)	PSNR (↑)	SSIM (↑)	MSE (↓)	PSNR (↑)	SSIM (↑)
Pix2Pix (adversarial loss) [29]	192.01	25.35	0.89	181.69	26.69	0.80
Pix2Pix (fixed-weight Multi- loss) [32]	177.84	25.73	0.90	223.35	25.62	0.73
Pix2Pix (Adaptive-weight Multi- loss) [33]	**159.33**	**26.19**	**0.91**	**165.47**	**26.98**	**0.82**

4.3 อภิปรายผล

4.3.1 ทดสอบผลกระทบของสมการที่ใช้ในการเรียนรู้ของ Discriminator

โดยการทดลองนี้ได้มีการใช้สมการตรวจสอบภาพที่แตกต่างกัน เพื่อทดสอบผลกระทบของสมการต่อคุณภาพของผลลัพธ์ การทดลองนี้นี้ได้กำหนดตัวแปร λ_x เป็นสัมประสิทธิ์แบบค่าคงที่เพื่อควบคุมผลกระทบของส่วนประกอบ x ในแต่ละสมการ ดังนี้

4.3.1.1 Standard loss function ประกอบไปด้วย Adversarial loss เป็นมาตรฐานที่ใช้ในการเปรียบเทียบกับวิธีที่งานวิจัยนี้นำเสนอ ตามสมการที่ (4-1)

4.3.1.2 Perceptual loss function ประกอบไปด้วย Adversarial loss และPerceptual loss

$$\mathcal{L}_D = L_{PGAN} + \lambda_{perceptual} * L_{percept} \quad (4\text{-}4)$$

4.3.1.3 Style loss function ประกอบไปด้วย Adversarial loss และ Style loss

$$\mathcal{L}_D = L_{PGAN} + \lambda_{style} * L_{style} \quad (4\text{-}5)$$

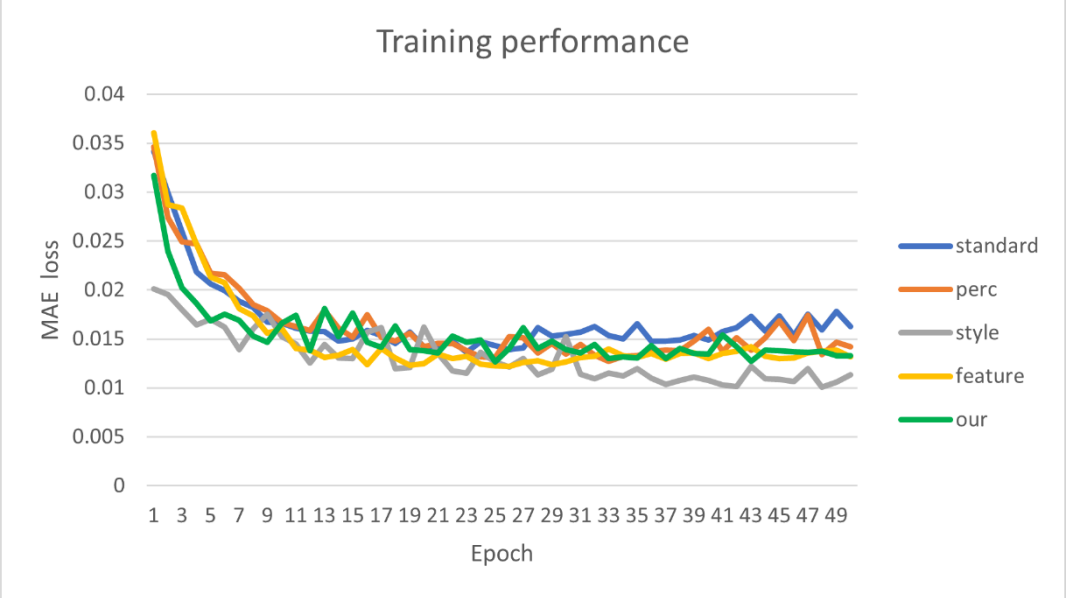
4.3.1.4 Feature loss function ประกอบไปด้วย Adversarial loss และ Adversarial feature matching loss ซึ่งประกอบด้วย Perceptual loss และ Style loss

$$\mathcal{L}_D = L_{PGAN} + \lambda_{perceptual} * L_{percept} + \lambda_{style} * L_{style} \quad (4\text{-}6)$$

4.3.1.5 Our loss function ประกอบไปด้วย Adversarial loss และ Adaptive feature matching loss ซึ่งเป็นวิธีที่งานวิจัยนี้นำเสนอ ตามสมการที่ (4-2)

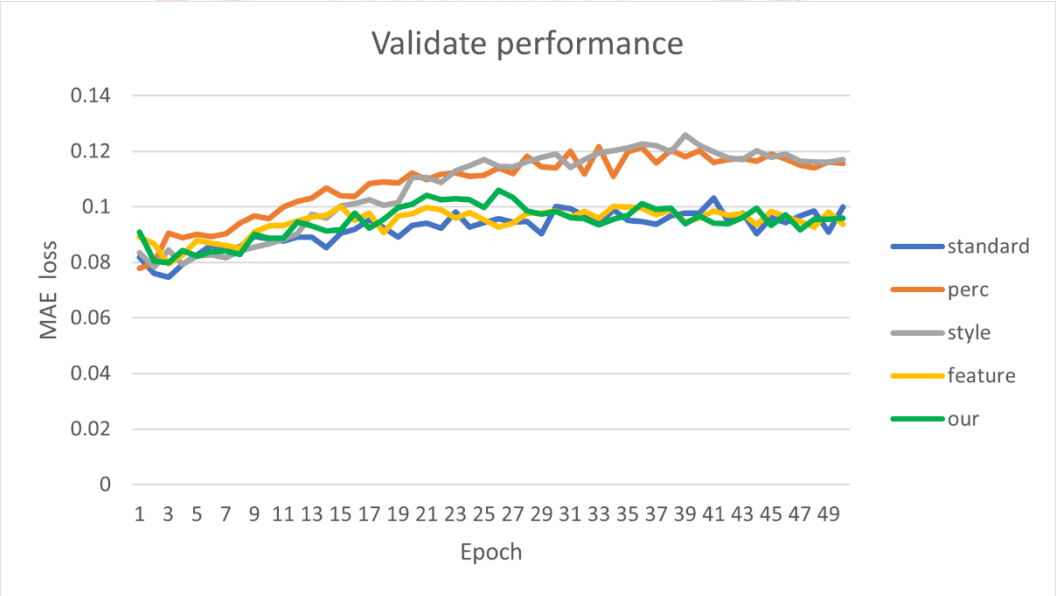
โดยโมเดลทั้งหมดใช้สมการการเรียนรู้ของ Generator (3-6) และค่าสัมประสิทธิ์ที่ใช้ในสมการควบคุม ดังนี้ λ_{L1}= 100, $\lambda_{perceptual}$= 0.05, λ_{style}= 120 และ λ_{tv}=0.1 ใช้เพื่อควบคุม MAE, perceptual loss, style loss และ total variation loss ตามลำดับ

จากการทดลอง ในช่วงขั้นตอนฝึกสอน (Training phase) แสดงให้เห็นว่าคุณภาพของผลลัพธ์ที่ดีที่สุดจากค่าการสูญเสียในการฟื้นคืนภาพให้ใกล้เคียงกับภาพเฉลยที่น้อยที่สุด คือ Style loss function นอกจากนี้ Perceptual loss function, Style loss function, Feature loss function และ Our loss function ยังแสดงถึงประสิทธิภาพในการเรียนรู้ซึ่งเพิ่มคุณภาพของผลลัพธ์ได้ดีกว่า Standard loss function ซึ่งเป็นมาตรฐานที่ใช้ในการเปรียบเทียบ โดยกราฟการวัดผลประสิทธิภาพแสดงตามภาพที่ 4-11



ภาพที่ 4-11 ประสิทธิภาพช่วงขั้นตอนฝึกสอน (Training phase) ของการทดลองผลกระทบของ multi-loss

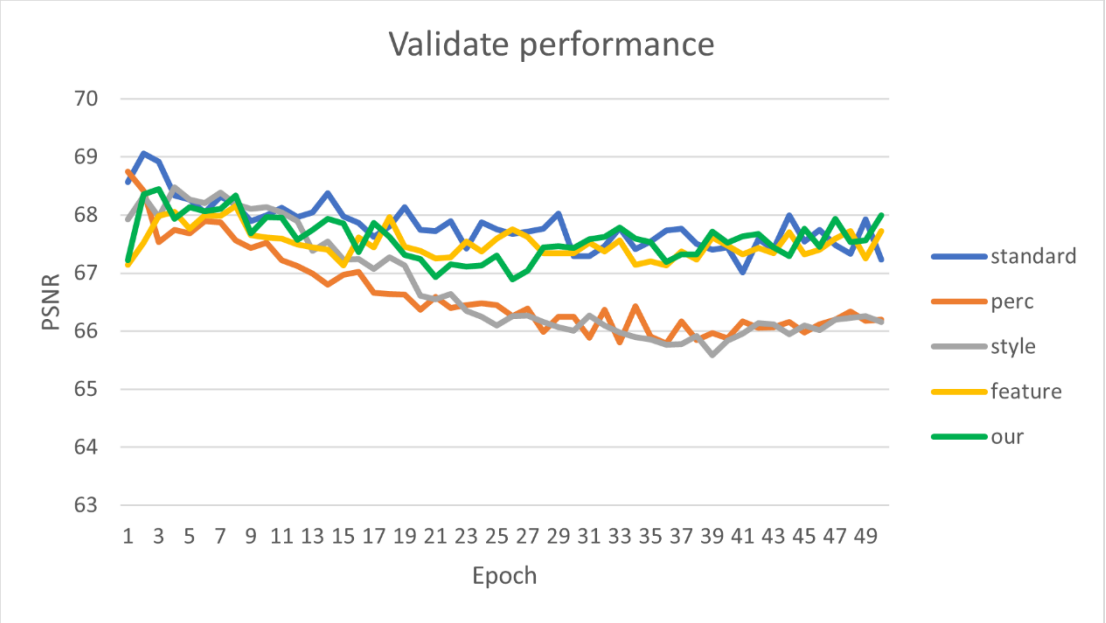
ในระหว่างฝึกสอนโมเดล ได้มีการทดสอบประสิทธิภาพของโมเดลเมื่อใช้กับข้อมูลที่ไม่เคยได้รับการเรียนรู้มาก่อน (unseen dataset) แสดงให้เห็นว่าสมการ Perceptual loss function และ Style loss function มีอัตราการเพิ่มขึ้นเมื่อผ่านรอบฝึกสอนโมเดลที่มากขึ้น ซึ่งอาจเป็นสัญญาณว่าสมการนี้ส่งผลให้โมเดลปรับตัวให้เข้ากับชุดข้อมูลที่ฝึกสอนมากเกินไป (overfit) ในขณะที่สมการที่เหลือมีอัตราค่าสูญเสียการฟื้นคืนภาพที่คงที่เมื่อรอบการฝึกสอนโมเดลเพิ่มมากขึ้น โดยกราฟการวัดผลประสิทธิภาพแสดงตามภาพที่ 4-12



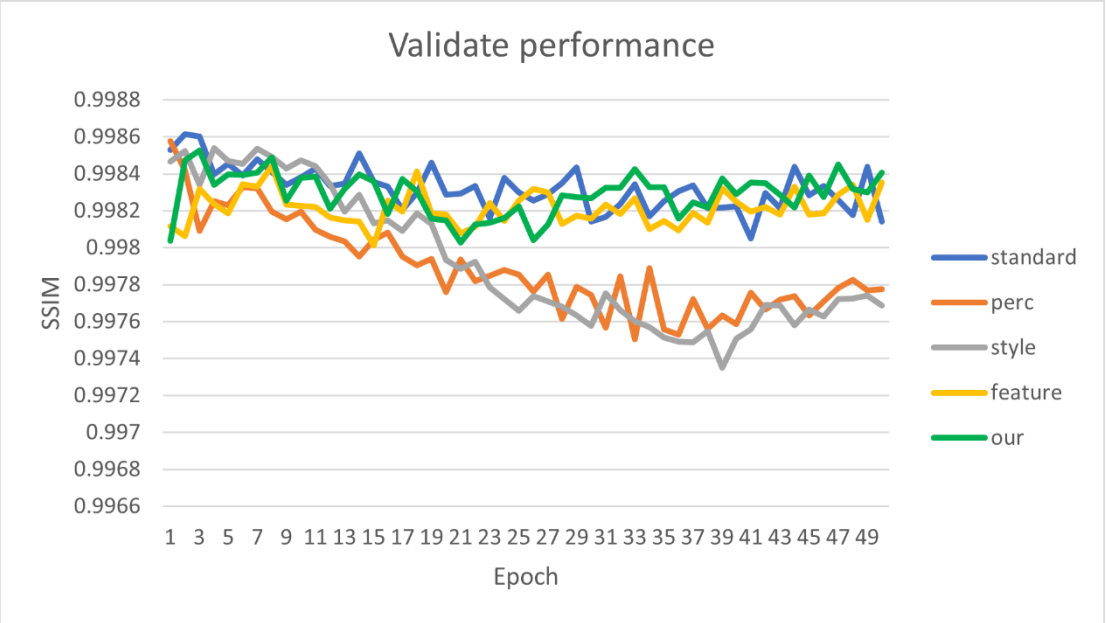
ภาพที่ 4-12 ประสิทธิภาพช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (Validate phase) ของการทดลองผลกระทบของ multi-loss

เมื่อสรุปผลจากในช่วงขั้นตอนฝึกสอน (Training phase) และช่วงการประเมินผลระหว่างการฝึกสอนโมเดล (Validate phase) สมการ Feature loss function และ Our loss function มีประสิทธิภาพใกล้เคียงกันในทั้ง 2 ช่วง

เพื่อค้นหาสมการที่ส่งผลให้โมเดลเติมภาพออกมาได้มีคุณภาพสูงที่สุด ผู้วิจัยได้ทำการพิจารณาโดยการใช้ค่า PSNR และ SSIM ตามภาพที่ 4-13 และ ภาพที่ 4-14 ตามลำดับ แสดงให้เห็นว่า Our loss function นั้น ส่งผลให้มีค่าสูงที่สุด แสดงว่าภาพนั้นมีคุณภาพสูงที่สุด



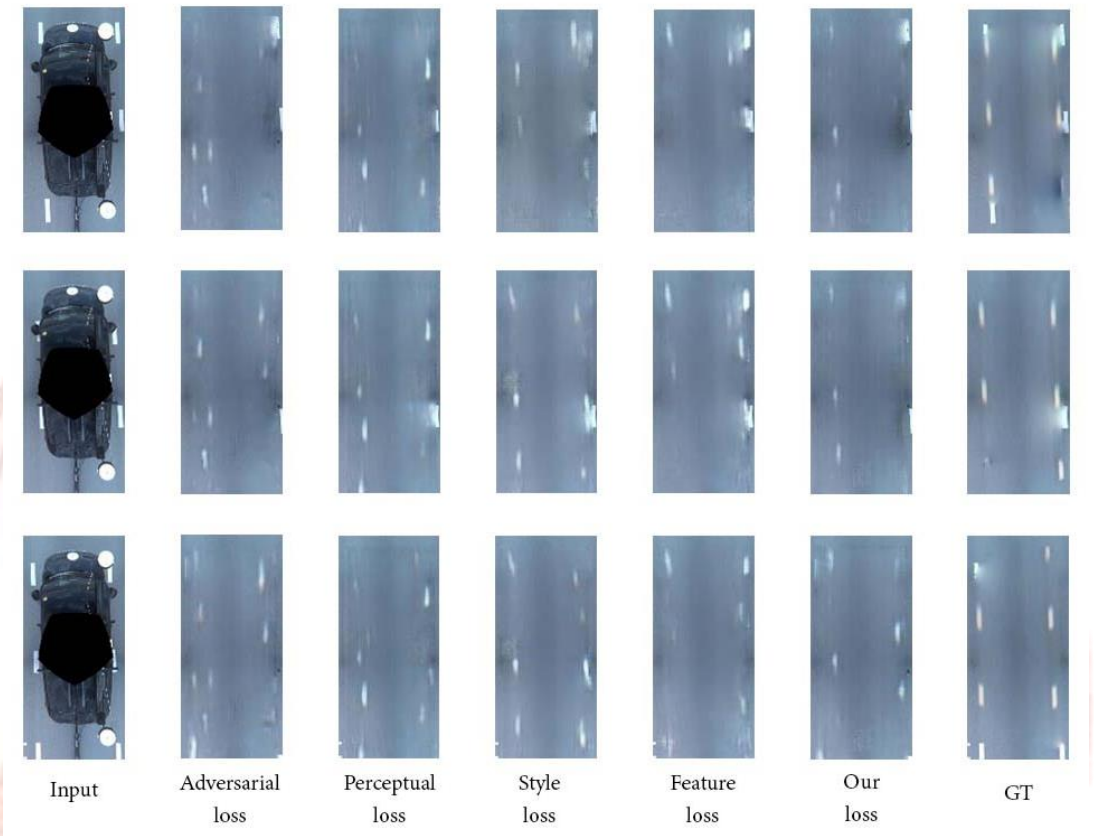
ภาพที่ 4-13 คุณภาพของภาพ (PSNR) ช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (Validate phase) ของการทดลองผลกระทบของ multi-loss



ภาพที่ 4-14 คุณภาพเชิงโครงสร้างของภาพ (SSIM) ช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (Validate phase) ของการทดลองผลกระทบของ multi-loss

เพื่อทำการวัดคุณภาพของวิธีการที่นำเสนอ งานวิจัยนี้ได้ทำการจำกัดขอบเขตของภาพให้อยู่ในพื้นที่เติมภาพเท่านั้นเทียบกับภาพเฉลยจากเทคนิคที่นำเสนอใน [31] ตามภาพที่ 4-15 โดยเทคนิคที่นำมาเปรียบเทียบ ได้แก่ Standard loss function จาก Pix2Pix [29], Perceptual loss,

Style loss, Adversarial feature matching loss และ Our loss ซึ่งคือวิธีการที่นำเสนอในงานวิจัยนี้ [32] ตามลำดับ



ภาพที่ 4-15 ผลลัพธ์การเติมภาพจากการทดลองผลกระทบของ multi-loss ที่นำมาทดสอบ

ผลการทดสอบการวัดคุณภาพของภาพที่ได้ในแต่ละ Objective function ที่แตกต่างกันทั้ง 5 วิธี โดยวัดผลที่ได้จากวิธีที่นำเสนอในงานวิจัยนี้ แสดงในตารางที่ 4-3 ผลที่ได้จากการวัดคุณภาพของภาพจากทั้ง 5 วิธีนี้ การวัดโดยใช้วิธี Mean-squared error (MSE) จากวิธีการที่นำเสนอนั้นมีค่าน้อยที่สุด แสดงว่า ผลลัพธ์มีความเหมือนกับภาพเฉลย ส่วนค่าที่ได้จากวิธี Peak signal-to-noise ratio (PSNR) กับวิธี Structural similarity (SSIM) ได้ค่าสูงมากกว่า แสดงให้เห็นว่าวิธีการที่นำเสนอไปนั้นสามารถเติมภาพได้คุณภาพสูงกว่าวิธีการอื่น

ตารางที่ 4-3 ผลการทดสอบการวัดคุณภาพของภาพจากการทดลองผลกระทบของ multi-loss

Model	MSE (↓)	PSNR (↑)	SSIM (↑)
Pix2Pix+ standard loss [29]	177.26	25.75	0.90
Pix2Pix + $\mathcal{L}_{perceptual}$	194.05	25.34	0.89
Pix2Pix + $\mathcal{L}_{\text{style}}$	188.28	25.49	0.90
Pix2Pix + $\mathcal{L}_{feature}$	161.10	26.17	0.91
Pix2Pix + our loss [32]	**160.84**	**26.26**	**0.91**

4.3.2 ทดสอบความละเอียดในการแบ่งภาพของ Discriminator (Patch size)

ในการทดลองนี้ ผู้วิจัยจะทำการปรับเปลี่ยนความละเอียดในการแบ่งภาพของ Discriminator ใช้ในการแยกแยะว่าภายในส่วนของภาพเป็นภาพจริงหรือเป็นภาพที่ถูกสังเคราะห์ขึ้นมา (patch size) โดยสัดส่วนของภาพที่ทดลองจะทำการเพิ่มการนั้นมีทั้งหมด 4 ขนาด ได้แก่

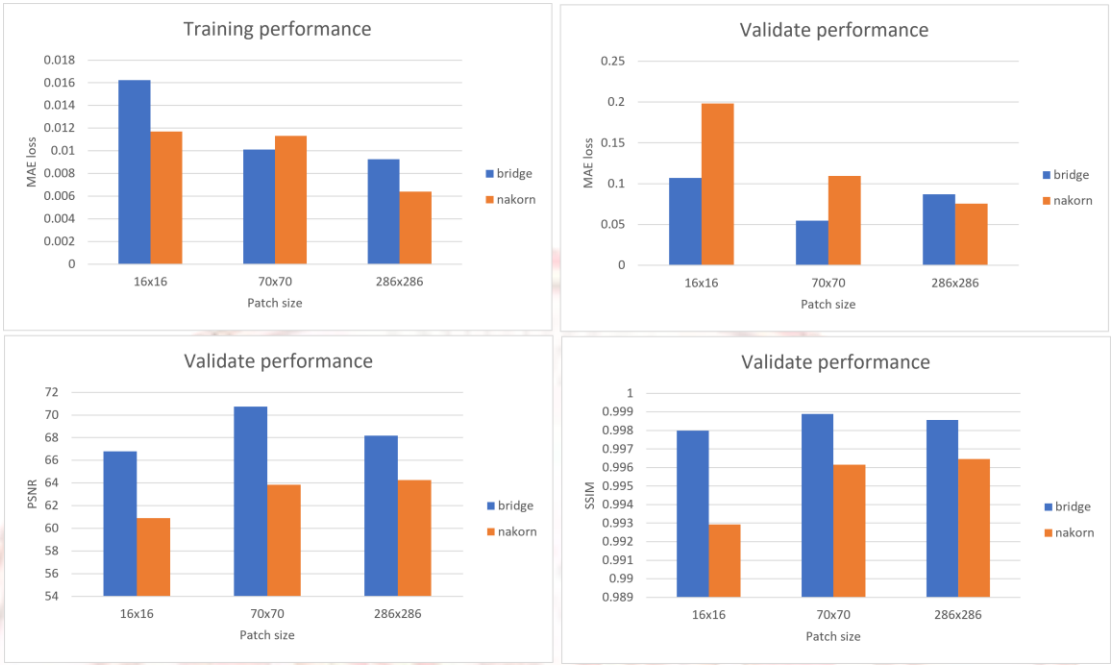
4.3.2.1 จำนวน 16 x 16 patch ต่อ 1 ภาพ

4.3.2.2 จำนวน 70 x 70 patch ต่อ 1 ภาพ

4.3.2.3 จำนวน 286 x 286 patch ต่อ 1 ภาพ

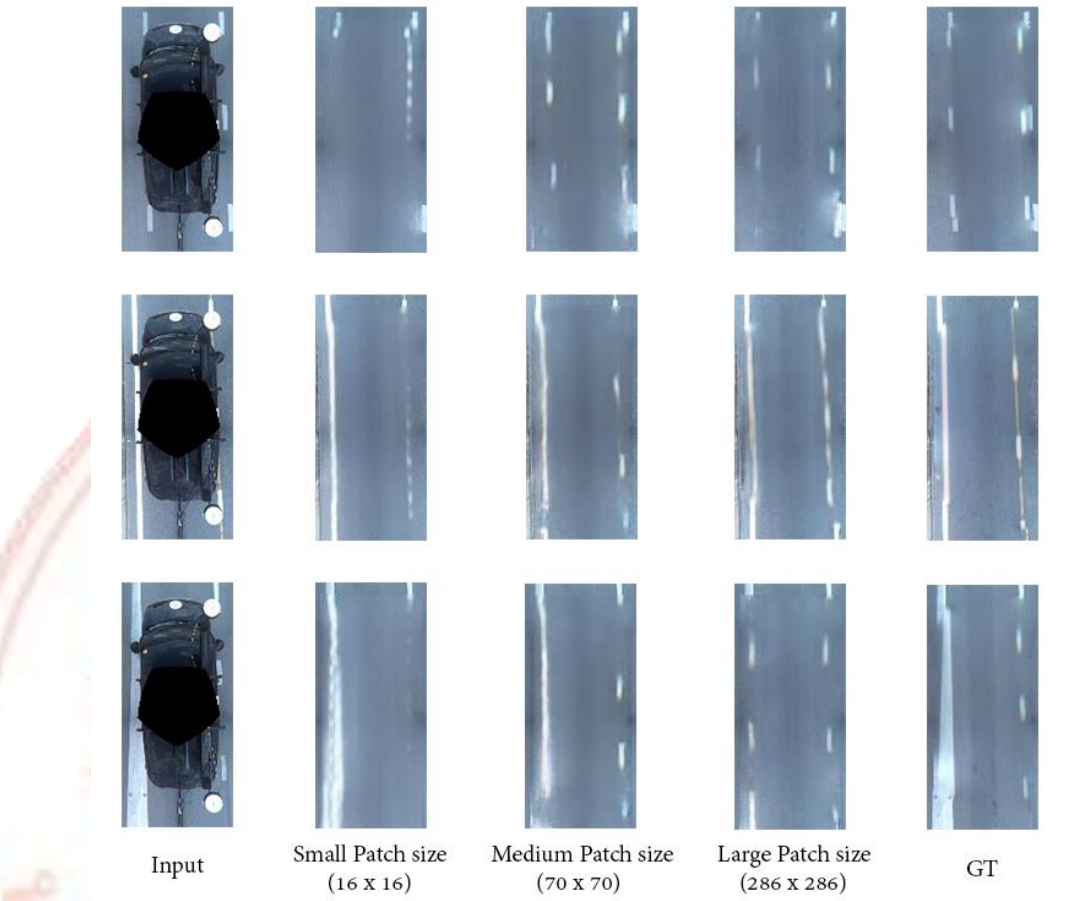
ทำการเปรียบเทียบผลลัพธ์สุดท้ายระหว่าง 2 dataset ได้ตามภาพที่ 4-16 โดยใช้ประสิทธิภาพการฟื้นฟูภาพช่วงขั้นตอนฝึกสอน (MAE training performance) ประสิทธิภาพการฟื้นฟูภาพช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (MAE Validate performance) คุณภาพของภาพช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (PSNR Validate performance) และคุณภาพเชิงโครงสร้างของภาพช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (SSIM Validate performance) ซึ่งแสดงให้เห็นว่าใน Bridge dataset การใช้ patch size ขนาด 70 x 70 ให้ผลลัพธ์ที่ดีที่สุด ในขณะเดียวกันใน Nakorn dataset การใช้ patch size ขนาด 286 x 286 ให้ผลลัพธ์ที่ดีที่สุด

จากการทดลองนี้ สามารถสรุปได้ว่าขนาดภาพที่ใช้ในการเติมภาพนั้น ส่งผลกระทบต่อการเลือก patch size ที่เหมาะสม ยิ่งขนาดภาพมีขนาดใหญ่ patch size ที่ใช้ก็ควรมีขนาดใหญ่ขึ้นด้วย



ภาพที่ 4-16 ประสิทธิภาพของโมเดลของการทดลองปรับขนาด patch size โดยรวม

เพื่อทำการวัดคุณภาพของวิธีการที่นำเสนอ งานวิจัยนี้ได้ทำการจำกัดขอบเขตของภาพในชุดข้อมูล Bridge dataset ให้อยู่ในพื้นที่เติมภาพเท่านั้นเทียบกับภาพเฉลยจากเทคนิคที่นำเสนอใน [31] โดยภาพที่ 4-17 แสดงภาพจาก patch size จำนวน 16x16 70x70 และ 286x286 ตามลำดับ



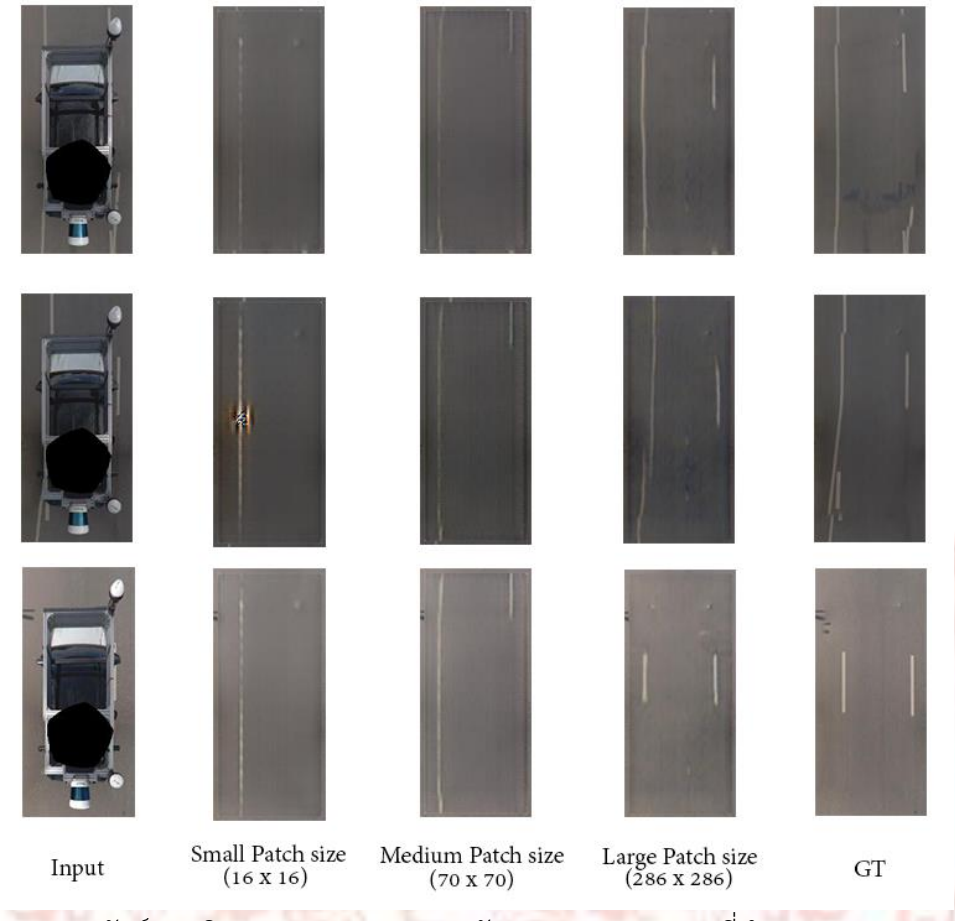
ภาพที่ 4-17 ผลลัพธ์การเติมภาพจากการทดลองปรับขนาด patch size ที่นำมาทดสอบของ Bridge dataset

ผลการทดสอบการวัดคุณภาพของภาพที่ได้ในแต่ละ Patch size ที่แตกต่างกัน โดยวัดผลที่ได้จากวิธีที่นำเสนอในงานวิจัยนี้ แสดงในตารางที่ 4-4 การวัดโดยใช้วิธี Mean-squared error (MSE) จากวิธีการที่นำเสนอนั้นมีค่าน้อยที่สุด แสดงว่า ผลลัพธ์มีความเหมือนกับภาพเฉลย ส่วนค่าที่ได้จากวิธี Peak signal-to-noise ratio (PSNR) กับวิธี Structural similarity (SSIM) ได้ค่าสูงมากกว่า แสดงให้เห็นว่าวิธีการที่นำเสนอไปนั้นสามารถเติมภาพได้คุณภาพสูงกว่าวิธีการอื่น นอกจากนี้ในการทดลองนี้ยังเพิ่มตัวแปรเวลาที่ใช้การฝึกสอนโมเดลแต่ละรอบมาด้วย เพื่อเปรียบเทียบเวลาที่จำเป็นต้องใช้ว่าแตกต่างกันอย่างไร ซึ่งแสดงให้เห็นว่ายิ่งเพิ่มจำนวนของ patch มากขึ้นเท่าใด เวลาที่ต้องใช้ก็ยิ่งมากเท่านั้น เนื่องจากจำนวน patch ที่ต้องตรวจสอบความถูกต้องมีจำนวนมากขึ้น

ตารางที่ 4-4 ผลการทดสอบการวัดคุณภาพของภาพจากการทดลองปรับขนาด patch size ของ bridge dataset

Patch size	Bridge dataset			
	MSE (↓)	PSNR (↑)	SSIM (↑)	Training time per epochs (min) (↓)
16 x 16	205.90	25.46	0.824	**49.66**
70 x 70	**168.63**	**25.99**	**0.864**	49.70
286 x 286	170.84	25.67	0.823	50.31

เพื่อทำการวัดคุณภาพของวิธีการที่นำเสนอ งานวิจัยนี้ได้ทำการจำกัดขอบเขตของภาพของภาพในชุดข้อมูล Nakorn dataset ให้อยู่ในพื้นที่เติมภาพเท่านั้นเทียบกับภาพเฉลยจากเทคนิคที่นำเสนอใน [31] โดยภาพที่ 4-18 แสดงภาพจาก patch size จำนวน 16x16 70x70 และ 286x286 ตามลำดับ



ภาพที่ 4-18 ผลลัพธ์การเติมภาพจากการทดลองปรับขนาด patch size ที่นำมาทดสอบของ Nakorn dataset

ผลการทดสอบการวัดคุณภาพของภาพที่ได้ในแต่ละ Patch size ที่แตกต่างกัน ตามตารางที่ 4-5 แสดงผลได้ว่าการใช้ patch size ที่ 286 x 286 ให้ผลลัพธ์ที่มีคุณภาพที่ดีที่สุด

ตารางที่ 4-5 ผลการทดสอบการวัดคุณภาพของภาพจากการทดลองปรับขนาด patch size ของ Nakorn dataset

Patch size	Nakorn dataset			
	MSE (↓)	PSNR (↑)	SSIM (↑)	Training time per epochs (min) (↓)
16 x 16	165.47	26.97	0.80	**91.98**
70 x 70	172.14	26.64	0.78	92.70
286 x 286	**143.39**	**27.33**	**0.82**	93.42

4.3.3 ทดสอบรอบการเปลี่ยนแปลงค่าของสัมประสิทธิ์ (Update epoch)

การทดลองนี้จะเปรียบเทียบระหว่างการใช้สัมประสิทธิ์แบบค่าคงที่ λ_x กับการใช้สัมประสิทธิ์แบบไดนามิก α_x โดยค่าสัมประสิทธิ์ตั้งต้น (initialize) ที่ใช้ในสมการควบคุม ดังนี้ λ_{L1}= 100, $\alpha_{perceptual}$= 1, α_{style}= 1 และ α_{tv}=0.1

การใช้สัมประสิทธิ์แบบไดนามิกที่จะทำการเปลี่ยนแปลงค่าของสัมประสิทธิ์ทุก ๆ n รอบ วิธีการที่นำมาเปรียบเทียบมี 5 วิธีการ ดังนี้

4.3.3.1 เปลี่ยนแปลงค่าของสัมประสิทธิ์ทุก ๆ 2 รอบ (n=2)

4.3.3.2 เปลี่ยนแปลงค่าของสัมประสิทธิ์ทุก ๆ 5 รอบ (n=5)

4.3.3.3 เปลี่ยนแปลงค่าของสัมประสิทธิ์ทุก ๆ 7 รอบ (n=7)

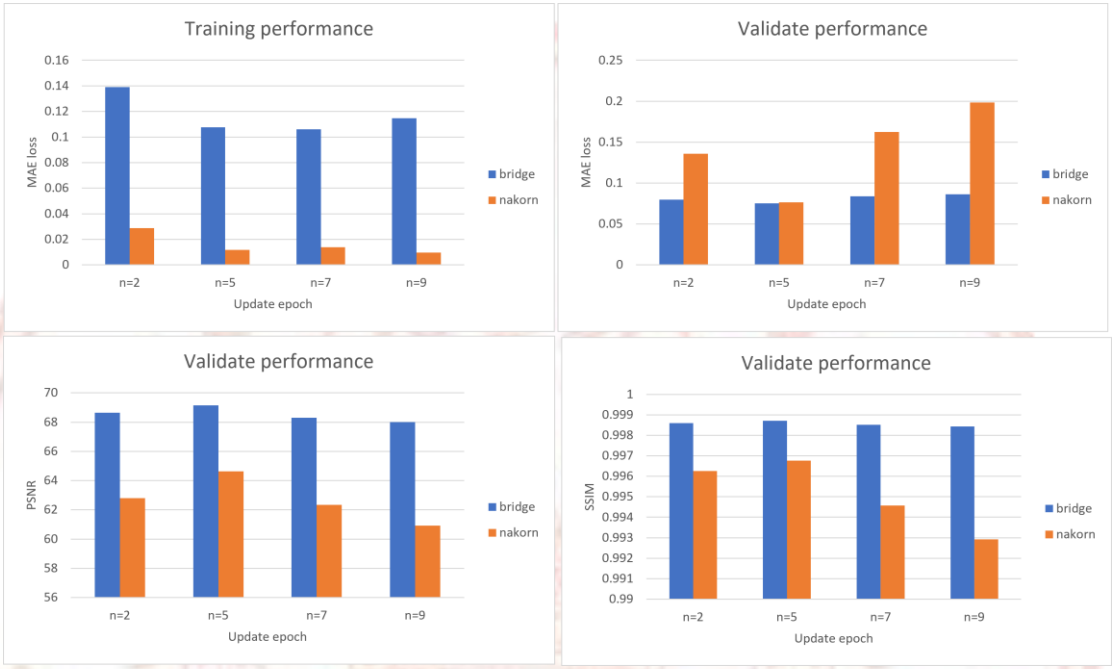
4.3.3.4 เปลี่ยนแปลงค่าของสัมประสิทธิ์ทุก ๆ 9 รอบ (n=9)

4.3.3.5 ใช้ค่าสัมประสิทธิ์คงที่ (Fixed weight)

ทำการเปรียบเทียบผลลัพธ์สุดท้ายระหว่าง 2 dataset ได้ตามภาพที่ 4-19 โดยใช้ประสิทธิภาพการฟื้นฟูภาพช่วงขั้นตอนฝึกสอน (MAE training performance) ประสิทธิภาพการฟื้นฟูภาพช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (MAE Validate performance) คุณภาพของภาพช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (PSNR Validate performance) และคุณภาพเชิงโครงสร้างของภาพช่วงการประเมินผลระหว่างการฝึกสอนโมเดลโดยใช้ข้อมูลที่ไม่เคยเรียนรู้ (SSIM Validate performance) ซึ่งแสดงให้

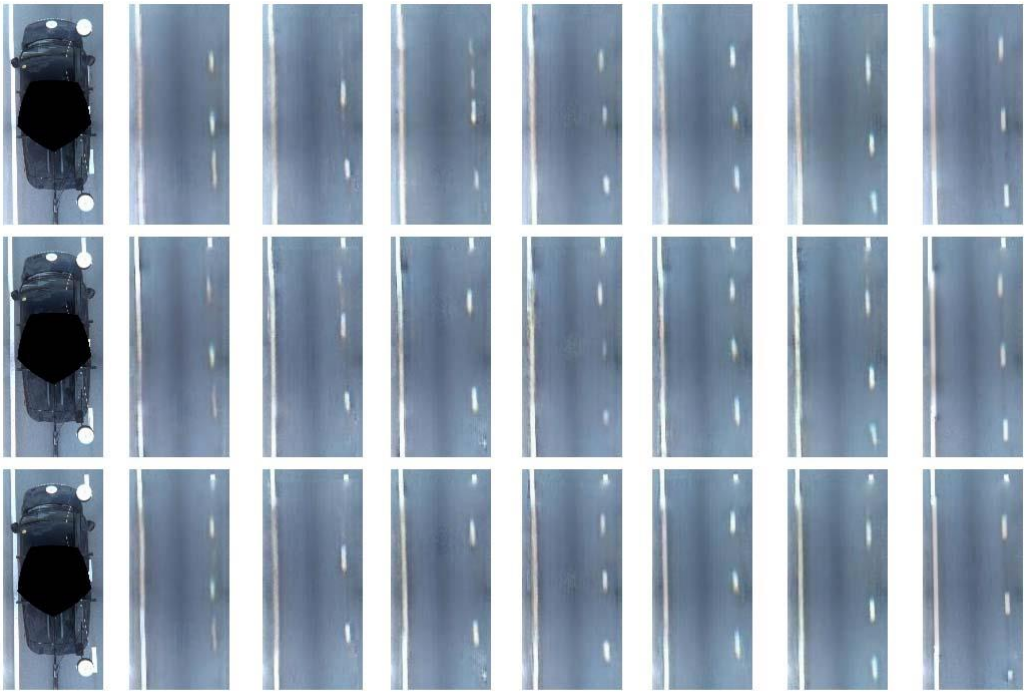
เห็นว่าทั้งใน Bridge dataset และ Nakorn dataset การเปลี่ยนแปลงค่าของสัมประสิทธิ์ทุก ๆ 5 รอบ (n=5) ให้ผลลัพธ์ที่ดีที่สุด

จากการทดลองนี้ สามารถสรุปได้ว่า แม้ว่ารูปแบบข้อมูลที่ใช้ในการฝึกสอนโมเดลนั้นมีลักษณะแตกต่างกัน จำนวนที่ใช้เปลี่ยนแปลงค่าสัมประสิทธิ์นั้นยังสามารถให้ผลลัพธ์ที่ดีได้

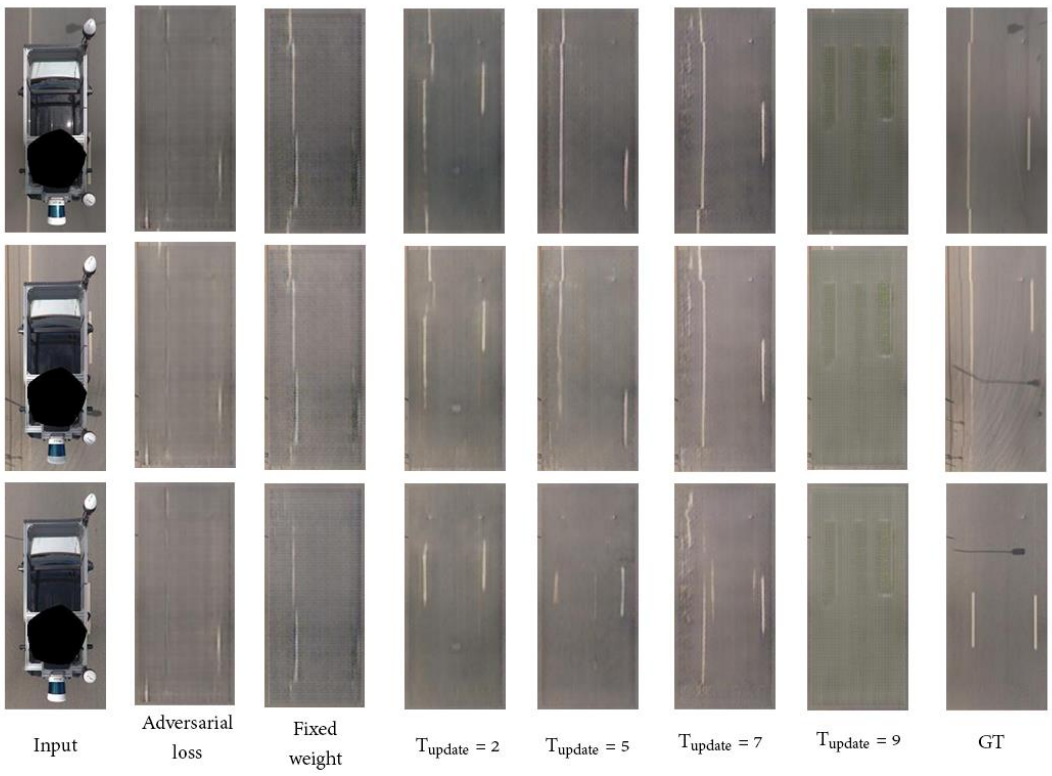


ภาพที่ 4-19 ประสิทธิภาพของโมเดลของการทดลองปรับสัมประสิทธิ์แบบไดนามิกโดยรวม

เพื่อทำการวัดคุณภาพของวิธีการที่นำเสนอ งานวิจัยนี้ได้ทำการจำกัดขอบเขตของภาพให้อยู่ในพื้นที่เติมภาพเท่านั้นเทียบกับภาพเฉลย โดยเทคนิคที่นำมาเปรียบเทียบ ได้แก่ Standard loss function จาก Pix2Pix [29], สมการควบคุมที่นำเสนอในงานวิจัยนี้ โดยใช้สัมประสิทธิ์ค่าคงที่ [32], ใช้สัมประสิทธิ์แบบไดนามิก [33] ที่อัพเดททุกๆ 2 5 7 และ 9 รอบ ตามลำดับ



ภาพที่ 4-20 ผลลัพธ์การเติมภาพจากการทดลองปรับสัมประสิทธิ์แบบไดนามิกของ Bridge dataset ที่นำมาทดสอบ



ภาพที่ 4-21 ผลลัพธ์การเติมภาพจากการทดลองปรับสัมประสิทธิ์แบบไดนามิกของ Nakorn dataset ที่นำมาทดสอบ

ผลการทดสอบการวัดคุณภาพของภาพที่ได้ โดยวัดผลที่ได้จากวิธีที่นำเสนอในงานวิจัยนี้ แสดงในตารางที่ 4-6 การวัดโดยใช้วิธี Mean-squared error (MSE) จากวิธีการที่นำเสนอนั้นมีค่าน้อยที่สุด แสดงว่า ผลลัพธ์มีความเหมือนกับภาพที่มาจากวิธี Patch-based ส่วนค่าที่ได้จาก Peak signal-to-noise ratio (PSNR) กับ Structural similarity (SSIM) ได้ค่าสูงมากกว่า แสดงให้เห็นว่าวิธีการฝึกสอนโมเดลโดยใช้สัมประสิทธิ์แบบไดนามิกที่นำเสนอในงานวิจัยนั้นสามารถเติมภาพได้คุณภาพสูงกว่าวิธีการอื่น

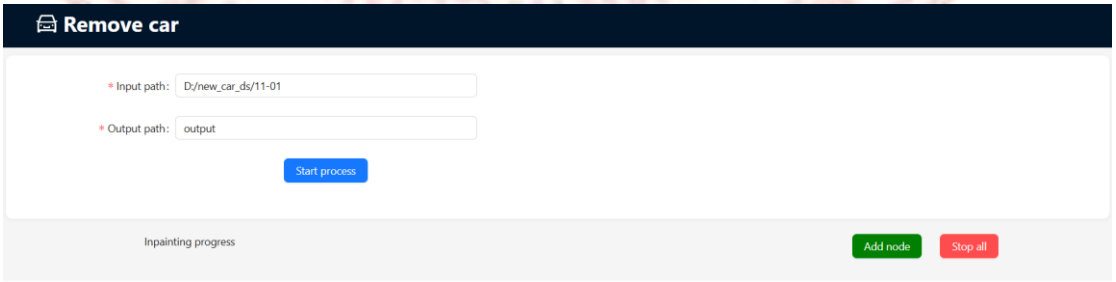
ตารางที่ 4-6 ผลการทดสอบการวัดคุณภาพของภาพจากการทดลองปรับสัมประสิทธิ์แบบไดนามิก

Method	Bridge dataset			Nakorn dataset		
	MSE (↓)	PSNR (↑)	SSIM (↑)	MSE (↓)	PSNR (↑)	SSIM (↑)
Pix2Pix (adversarial loss) [29]	192.01	25.35	0.89	181.69	26.69	0.80
Pix2Pix (fixed-weight Multi- loss) [32]	177.84	25.73	0.90	223.35	25.62	0.73
$n = 2$	181.77	25.61	0.90	225.35	25.69	**0.81**
$\boldsymbol{n = 5}$	**159.33**	**26.19**	**0.91**	**165.47**	**26.30**	0.79
$n = 7$	185.11	25.55	0.90	213.16	25.75	0.77
$n = 9$	195.57	25.31	0.89	238.67	25.58	0.78

4.4 การทดสอบระบบ

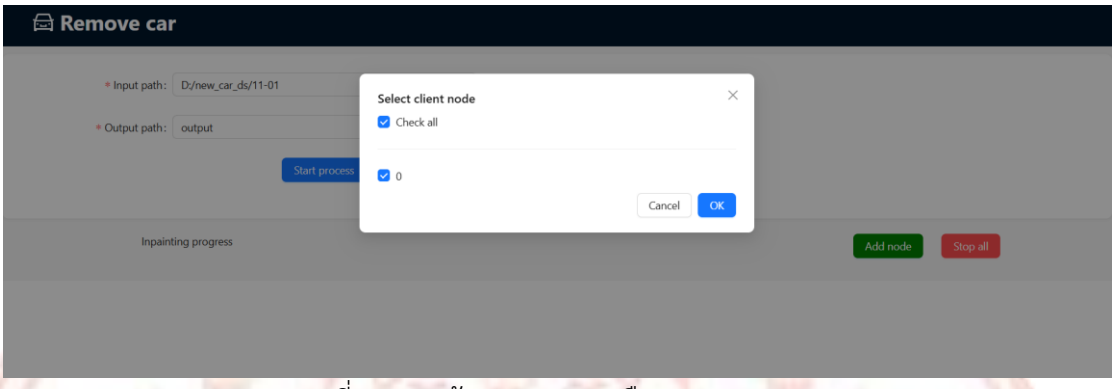
การทดสอบระบบที่ได้มีการพัฒนา มีรายละเอียดดังต่อไปนี้

4.4.1 การกรอกข้อมูล เริ่มต้นจากผู้ใช้งานกรอกข้อมูลโฟลเดอร์ที่ต้องการประมวลผลลบภาพรถสำรวจ พร้อมทั้งชื่อของโฟลเดอร์ที่จะใช้เก็บผลลัพธ์ ซึ่งโฟลเดอร์ที่เก็บผลลัพธ์นี้จะอยู่ภายในโฟลเดอร์ต้นทาง



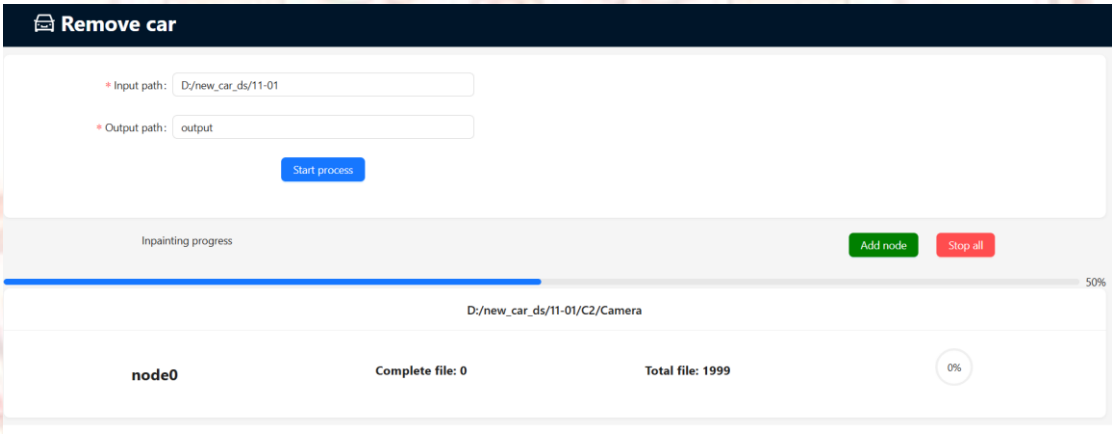
ภาพที่ 4-22 หน้าจอแสดงการกรอกรายละเอียด

4.4.2 การเลือก Client node เมื่อกรอกข้อมูลเสร็จเรียบร้อยแล้ว ผู้ใช้ต้องทำการจำนวน Client node ที่ต้องการใช้ในการประมวลผล ก่อนเริ่มการประมวลผล



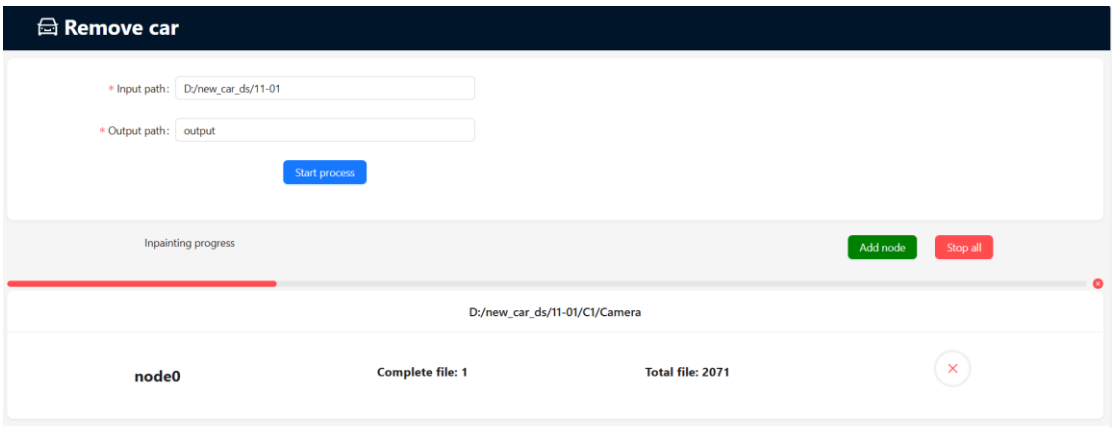
ภาพที่ 4-23 หน้าจอแสดงการเลือก Client node

4.4.3 ในระหว่างที่ Client node ประมวลผล ส่วนที่ติดต่อผู้ใช้งานจะแสดงจำนวนโฟลเดอร์ที่ผ่านการประมวลผล พร้อมกับจำนวนภาพที่ผ่านการประมวลผลของแต่ละ Client node



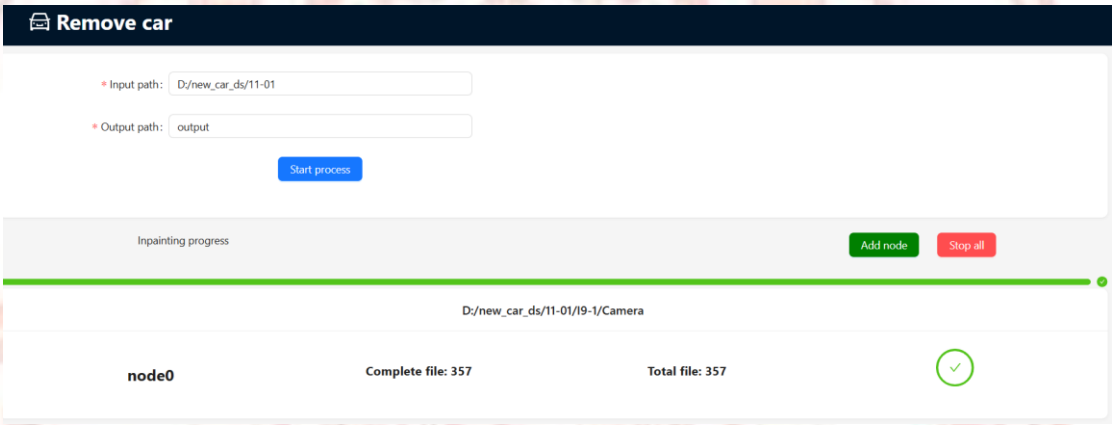
ภาพที่ 4-24 หน้าจอแสดงการประมวลผล

4.4.4 หากผู้ใช้งานมีการยกเลิกการทำงานระหว่างประมวลผล แถบสถานะการทำงานจะเปลี่ยนเป็นสีแดงเพื่อแสดงให้ผู้ใช้ทราบว่า Client node ได้หยุดการทำงานเรียบร้อยแล้ว



ภาพที่ 4-25 หน้าจอแสดงการยกเลิกการประมวลผล

4.4.5 เมื่อระบบทำการประมวลผลภาพสำเร็จทั้งหมดแล้ว แถบแสดงสถานะการทำงานจะเปลี่ยนเป็นสีเขียว



ภาพที่ 4-26 หน้าจอแสดงการประมวลผลสำเร็จ

บทที่ 5
สรุปผลการทดสอบ และข้อเสนอแนะ

5.1 สรุปผลการทดสอบ

งานวิจัยนี้ได้นำเสนอเทคนิคที่ใช้ลบวัตถุเป้าหมายโดยใช้วิธีการสังเคราะห์ภาพขึ้นมาเพื่อแทนที่ภาพวัตถุเป้าหมายที่ถูกลบออกไปจากภาพ ซึ่งวัตถุเป้าหมายในงานวิจัยนี้คือ ยานพาหนะที่เก็บข้อมูลภาพขณะสำรวจ โดยพัฒนาโมเดล Pix2Pix โดยปรับสมการที่ใช้ในการเรียนรู้เพื่อเพิ่มความสามารถในการตรวจจับภาพของ Discriminator ในด้านการเก็บรายละเอียดเชิงความหมายจาก Perceptual loss ด้านการเก็บรายละเอียดเชิงสไตล์จาก Style loss และเพิ่มความสามารถในการเติมภาพอย่างแนบเนียนจากการใช้ Total variation loss นอกจากนี้เพื่อให้ประสิทธิภาพในการทำงานของสมการเรียนรู้ที่เพิ่มมานั้นสามารถทำงานได้อย่างเต็มประสิทธิภาพ งานวิจัยนี้ได้เสนอวิธีการควบรวมสมการโดยใช้สัมประสิทธิ์แบบไดนามิก (Adaptive weight) เพื่อให้ผลลัพธ์มีคุณภาพสูงที่สุด และยังใช้ได้กับข้อมูลที่มีคุณลักษณะที่แตกต่างกัน หลังจากการฝึกสอนการเติมภาพในส่วนของการใช้งานนั้นเพียงอ้างอิงเฉพาะข้อมูลภายในภาพนั้น ๆ จากผลการทดสอบด้านคุณภาพโดยเปรียบเทียบกับวิธีก่อนและหลังการปรับปรุงที่นำเสนอในงานวิจัยนี้ โดยเทคนิคที่ใช้ในงานวิจัยนี้ให้คุณภาพสูงกว่าวิธีการก่อนการปรับปรุง จากการทดสอบด้านเวลาการประมวลผลเปรียบเทียบกับงานวิจัยที่มีขอบเขต และชุดข้อมูลเดียวกัน สามารถแสดงได้ว่าวิธีการที่งานวิจัยนี้นำเสนอสามารถเติมภาพได้ภายในระยะเวลาสั้น

นอกจากนี้เมื่อเปรียบเทียบกับข้อจำกัดกับวิธีการที่นำเสนอก่อนหน้าในการอ้างอิงพิกัดตำแหน่งและทิศของยานพาหนะเป้าหมายนั้น วิธีการที่งานวิจัยนี้ยังสามารถเติมภาพในกรณีการสำรวจกระทำในพื้นที่สาธารณะที่มียานพาหนะอื่น นอกจากยานพาหนะเป้าหมายที่ปรากฏบนพิกัดตำแหน่งเดียวกันกับยานพาหนะเป้าหมายที่จะนำมาใช้เติมภาพ ทำให้ภาพวัตถุอื่นปรากฏบนพื้นที่ที่ลบยานพาหนะเป้าหมาย แต่งานวิจัยที่นำเสนอนั้นจะไม่ปรากฏวัตถุอื่นในการเติมภาพ เนื่องจากการเรียนรู้ข้อมูลภายในภาพเดียว

5.2 ข้อเสนอแนะ

5.2.1 ในกรณีที่ภาพมีความคมชัดเพิ่มมากขึ้น วิธีการเติมภาพที่นำเสนอในงานวิจัยนี้มีการผสมภาพ (Image blending) เพื่อให้ส่วนที่เติมภาพนั้นมีความเข้ากันได้กับพื้นที่รอบข้างมากยิ่งขึ้น

5.2.2 วิธีการที่ได้นำเสนอในงานวิจัยนี้ใช้เพื่อรักษาทัศนียภาพด้านล่างของภาพ เป็นส่วนที่มีรายละเอียดได้แก่ พื้นผิวถนน เส้นถนน ซึ่งเป็นส่วนของภาพที่มีรายละเอียดน้อย หากรายละเอียดของ

องค์ประกอบภายในภาพมีรายละเอียดสูง เช่น ภาพเอกซเรย์ (X-ray) ที่มีโครงสร้างของร่างกาย วิธีการในงานวิจัยนี้อาจไม่สามารถเติมรายละเอียดได้อย่างถูกต้องครบถ้วนได้

5.3 งานในอนาคต

5.3.1 สามารถนำเทคนิคที่มีการนำเสนอนั้นไปใช้กับการลบวัตถุเป้าหมายชนิดอื่น ๆ ที่ไม่ต้องการในพื้นที่สำรวจได้ โดยการฝึกสอนจากการใช้ Mask ที่หลากหลาย เพื่อเพิ่มความหลากหลายของรูปร่างของวัตถุที่ต้องการเติมภาพ

5.3.2 พัฒนาเทคนิคจากวิธีที่นำเสนอนงานวิจัยที่นำเสนอการใช้ Conditional GAN เป็นการฝึกสอนโมเดลแบบ Unconditional GAN เพื่อเพิ่มประสิทธิภาพการฝึกสอนโมเดลมากยิ่งขึ้น เนื่องจากในเทคนิคปัจจุบันจำเป็นต้องใช้มนุษย์ในการคัดเลือกภาพเพื่อช่วยให้โมเดลมีประสิทธิภาพในการเติมภาพได้อย่างสูงสุด

บรรณานุกรม

1. Elhashash, M., Albanwan, H., & Qin, R. (2022). A Review of Mobile Mapping Systems: From Sensors to Applications. *Sensors, 22*(11), 4262. doi:https://doi.org/10.3390/s22114262
2. Zhou, T., Johnson, B., & Li, R. (2016). *Patch-based texture synthesis for image inpainting*.
3. Zhang, N., Ji, H., Liu, L., & Wang, G. (2019). Exemplar-based image inpainting using angle-aware patch matching. *EURASIP Journal on Image and Video Processing, 2019*(1), 70. doi:10.1186/s13640-019-0471-2
4. Yang, S., Liang, H., Wang, Y., Cai, H., & Chen, X. (2020). Image inpainting based on multi-patch match with adaptive size. *Applied Sciences, 10*(14), 4921.
5. Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., & Catanzaro, B. (2018). *Image inpainting for irregular holes using partial convolutions.* Paper presented at the Proceedings of the European conference on computer vision (ECCV).
6. Mansourifar, H., & Simske, S. J. (2023). GAN-Based Object Removal in High-Resolution Satellite Images. *arXiv preprint arXiv:2301.11726*.
7. Peng, J., Liu, D., Xu, S., & Li, H. (2021). *Generating diverse structure for image inpainting with hierarchical VQ-VAE.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
8. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Van Gool, L. (2022). *Repaint: Inpainting using denoising diffusion probabilistic models.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

9. Shang, Z., Liu, Y., Li, G., Zhang, Y., Miao, J., Liu, J., & Wang, L. (2022). *Viewport-Oriented Panoramic Image Inpainting.* Paper presented at the 2022 IEEE International Conference on Image Processing (ICIP).

10. Han, S. W., & Suh, D. Y. (2020). PIINET: A 360-degree panoramic image inpainting network using a cube map. *arXiv preprint arXiv:2010.16003*.

11. Lertbumroongchai, K. (2563). High Dynamic Range Panorama Image Processing to Create Virtual Reality Tour. [https://so05.tci-thaijo.org/index.php/jmctrmutp/article/view/251886]. *journal of Mass Communication Technology, RMTP, 5*(1), 87 - 98.

12. Lertbumroongchai, K. (2019). [online]. Panorama Photography. [cited 25 November 2567] Retrieved from https://touchpoint.in.th/panorama-part1/

13. Wikipedia. (2024). [online]. Censor bars. [cited 25 November, 2024] Retrieved from https://en.wikipedia.org/wiki/Censor_bars

14. Garske, V., & Noack, A. (2022). *Revisiting the privacy of censored credentials*.

15. Shankland, S. (2008). [online]. Google begins blurring faces in Street View. [cited 25 November, 2024] Retrieved from https://www.cnet.com/culture/google-begins-blurring-faces-in-street-view/

16. Wikipedia. (2024). [online]. Pixelization. [cited 25 November, 2024] Retrieved from https://en.wikipedia.org/wiki/Pixelization

17. Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning, 12*(4), 307-392.

18. Wikipedia. (2021). [online]. Variational autoencoder. [cited 25 November, 2024] Retrieved from https://en.wikipedia.org/wiki/Variational_autoencoder

19. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems, 27*.

20. Ongun, C., & Temizel, A. (2019). Paired 3D Model Generation with Conditional Generative Adversarial Networks. In (pp. 473-487).

21. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). *High-resolution image synthesis with latent diffusion models.* Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

22. Aristimuño, I. (2023). [online]. An Introduction to Diffusion Models and Stable Diffusion. [cited 25 November, 2025] Retrieved from https://blog.marvik.ai/2023/11/28/an-introduction-to-diffusion-models-and-stable-diffusion/

23. Johnson, J., Alahi, A., & Fei-Fei, L. (2016). *Perceptual losses for real-time style transfer and super-resolution.* Paper presented at the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14.

24. Whitaker, J. (2023). [online]. An Explanation of Style Transfer With a Showdown of Different Techniques. [cited 25 November, 2024] Retrieved from https://wandb.ai/johnowhitaker/style_loss_showdown/reports/An-Explanation-of-Style-Transfer-With-a-Showdown-of-Different-Techniques--VmlldzozMDIzNjg0#content-loss

25. Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). *Image style transfer using convolutional neural networks.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

26. Rajaa, S. (2018). [online]. Neural Style Transfer - the math & code. [cited 25 November, 2024] Retrieved from https://medium.com/@shangethrajaa/neural-style-transfer-the-math-code-71ef46200ca8

27. Chambolle, A. (2004). An Algorithm for Total Variation Minimization and Applications. *Journal of Mathematical Imaging and Vision, 20*(1), 89-97. doi:10.1023/B:JMIV.0000011325.36760.1e

28. Wikipedia. (2024). [online]. Total variation denoising. [cited 25 November, 2024] Retrieved from https://en.wikipedia.org/wiki/Total_variation_denoising

29. Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). *Image-to-image translation with conditional adversarial networks.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

30. Heydari, A. A., Thompson, C. A., & Mehmood, A. (2019). Softadapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions. *arXiv preprint arXiv:1912.12355*.

31. Inprom, E., & Pipanmekaporn, L. (2021). *Object Inpainting in Panoramic media for Mobile Mapping System.* (Master of Science (Computer Science)). King Mongkut's University of Technology North Bangkok,

32. Lording, A., Pipanmaekaporn, L., Kamonsantiroj, S., & Srisomboon, K. (2024). *GAN-Driven Vehicle Inpainting for Mobile Mapping Scenes.* Paper presented at the 2024 9th International Conference on Control and Robotics Engineering (ICCRE).

33. Lording, A., Pipanmaekaporn, L., Kamonsantiroj, S., & Srisomboon, K. (2024). *Adaptive Multi-Loss GAN for Object Removals in Street-View Scenes.* Paper presented at the the 5th Research, Invention, and Innovation Congress (RI2C 2024), Bangkok, Thailand.

ภาคผนวก ก

ผลงานวิทยานิพนธ์ที่นำเสนอในที่ประชุมวิชาการ

CO-SPONSORED BY

PATRONS

9 798350 372694
ISBN: 979-8-3503-7269-4
IEEE Catalog Number: CFP24JAA-ART
ISSN 2835-3722

2024 9th International Conference on Control and Robotics Engineering (ICCRE 2024)
ISBN: 979-8-3503-7269-4
IEEE Catalog Number: CFP24JAA-ART

ICCRE 2024

2024 9th International Conference on Control and Robotics Engineering

May 10-12, 2024 / Osaka, Japan

GAN-Driven Vehicle Inpainting for Mobile Mapping Scenes

Arzeezar Lording
Department of Computer and Information Science
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
s6504062810021@email.kmutnb.ac.th

Luepol Pipanmaekaporn
Department of Computer and Information Science
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
luepol.p@sci.kmutnb.ac.th;

Suwatchai Kamonsantiroj
Department of Computer and Information Science
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
suwatchai.k@sci.kmutnb.ac.th;

Kanabadee Srisomboon
Department of Electrical and Computer Engineering
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
kanabadee.s@eng.kmutnb.ac.th

***Abstract*—Mobile mapping systems (MMS) capture geospatial data using vehicle-mounted sensors and cameras for generating digital maps. However, the collected images contain sensitive information requiring privacy protection methods. Although object removal approaches have been proposed, these methods may not be well-suited for MMS imagery due to their unique characteristics and processing requirements. In this study, we propose a Generative Adversarial Network (GAN)-based technique tailored to remove unwanted vehicles from MMS scenes. Our approach leverages Pix2Pix, a GAN architecture for image-to-image translation. To handle the challenge of inpainting full panoramic scenes, we extract viewports containing target vehicles from the raw panoramic image. Subsequently, a Pix2Pix-based inpainting network is employed to inpaint the target object in viewport. We enhance the standard Pix2Pix framework by incorporating adversarial feature matching to generate realistic results. Experimental results demonstrated that our method preserves the output quality while efficiently processing large raw MMS image datasets, outperforming traditional methods.**

Keywords*—Image Inpainting, Mobile Mapping System, Generative Adversarial Network

I. Introduction

Mobile Mapping Systems (MMS) [1] have rapidly become essential for gathering geospatial data, including panoramic images, LiDAR point clouds, GPS coordinates, aiming to create detailed digital maps from vehicle-mounted platforms. Street-level MMS, in particular, are widely used for mapping urban environments. However, the street-view imagery unintentionally captures individuals and vehicles, leading to privacy concerns. Early approaches to privacy protection is to detect the objects of interest (e.g. pedestrian and license plates) and blurring them [2][3]. Although these approaches ensure the privacy, these approaches tend to generate unrealistic textures in the replaced areas.

To produce more realistic results, previous studies have focused on image inpainting to reconstruct missing regions in images. These works explored various methods, including copying pixels (patches) from other image regions [4][5][6] and propagating information from surrounding pixels to fill in inpainted areas [7]. However, these methods face challenges in MMS due to the potential size of missing regions and complex scenes.

Recent advancements in Generative Adversarial Networks (GANs) [8] have significantly improved image inpainting capabilities by synthesizing more realistic content for the missing regions. One of the pioneering works in this domain is the Context Encoder (CE) [9], employing an encoder-decoder architecture to fill in the inpainted content based on the contextual information surrounding the missing area. In [10], Globally and Locally Consistent Image Completion (GLCIC) has been introduced. In GLCIC, a two-stage network that leverages two discriminators: a global discriminator ensures coherence between the inpainted region and the entire image, while a local discriminator focuses on preserving fine-grained details and textures within the inpainted area. In [11], the authors introduced free-form inpainting to address arbitrary-shaped and large missing regions. The authors proposed partial convolution layers to capture and propagate information from context to the missing areas. Directly applying existing GAN-based methods to images captured by MMS often poses significant challenges. Firstly, MMS widely uses equirectangular panoramic projections, whose distortions can adversely affect inpainting performance. Secondly, the large size and high resolution of MMS panoramas often require computational complexity. Thirdly, the ratio of equirectangular images can hinder deep learning models trained on planar images. Thus, a tailored approach is required to overcome the unique characteristics of equirectangular imagery in MMS.

This study introduces a synthesized-based approach to image inpainting for panoramic MMS images using Generative Adversarial Network (GAN). Our method utilizes Pix2Pix [12], a GAN architecture for image-to-image translation tasks, to effectively remove unwanted vehicles in MMS images. To address inpainting panoramic images, we introduce a view selection method that renders a narrow field of view containing the target vehicles. We also highlight that the standard loss function used in Pix2Pix may fall short in synthesizing the high quality of images. Instead, we propose a novel adversarial training strategy that incorporates semantic feature matching. In other words, high-level features are extracted from both real and synthesized images and compared during training, enforcing the

generator network to produce more realistic results. Experimental results, conducted on MMS image datasets demonstrated that the proposed method achieves inpainting vehicles from street-view panoramic images while efficiently processing a large number of images.

The remainder of this paper is structured as follows: Section II introduces background knowledge that is used in our work, Section III introduces our proposed method, Section IV showcases the experiments conducted, Section V presents the experiment of our proposed method, and finally, Section VI concludes the paper.

II. Ease of Use

A. Generative Adversarial Network (GAN)

Generative Adversarial Network (GAN) was first introduced in [8] as a powerful generative model. The GAN consists of two neural networks: a generator and a discriminator. The generator learns to fit the distribution of training data. The discriminator is used to spot the input data synthesized by the generator. In GAN, both the generator and the discriminator are trained simultaneously by optimizing the adversarial loss function below:

$$L_{GAN} = \min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D(G(z|y))\right)\right] \quad (1)$$

A random noise z obeying the $pz(z)$ distribution generates x from the generator G and then inputs the generated data and the real data x into the discriminator D that identifies the input x is a real sample or a generated sample by G. At present, GANs have become the state-of-the-art in image inpainting to achieve more realistic results.

B. Pix2Pix

Pix2Pix [12] is a conditional GAN model designed for image-to-image translation tasks. This network learns to generate new images based on reference and target images. Pix2Pix network contains a U-NET architecture as the generator, which takes a reference image and learns to generate a new image based on target. The discriminator is a PatchGAN that analyzes small patches of the input image as real or generated and combines the results to improve the generator. The loss function of Pix2Pix GAN is a combination of adversarial loss and L1 function. The L1 loss function acts to minimize the sum of errors between the target image and the generated image, producing less blur in the output image.

$$L_{Pix2Pix} = \min_G \max_D L_{PGAN}(G,D) + \lambda_{L1}\mathcal{L}_{L1}(G) \quad (2)$$

where $L_{L1} = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]$ denotes L1 loss function, λ_{L1} is a hyper-parameter that specifies the importance of L1 error and L_{GAN} can be defined as follows:

$$L_{GAN} = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}\left[\log\left(1 - D(x,G(x,z))\right)\right] \quad (3)$$

Where x denotes the input sample and y is the target output image and $G(x)$ denotes the generated output image. In Pix2Pix, the discriminator aims to classify whether each $n\times n$ patch in an image is real or fake. The loss function of PatchGAN discriminator can be written as follows:

$$\mathcal{L}_{PGAN} = \mathbb{E}_{x,y}\left[\sum_{i=1}^{Np}\log D(x_i,y_i)\right] + \mathbb{E}_{x,z}\left[\sum_{i=1}^{Np}\log\left(1 - D(x,G(x,z))\right)\right] \quad (4)$$

where Np is the total number of N*N patches in the image.

III. PROPOSED METHOD

The proposed method is illustrated in Fig. 1. Given a collection of panoramic images captured by MMS vehicle, each image containing a target vehicle is selected to a view selection process. This process aims to generate a narrow viewport focused on the target vehicle within the panoramic scene. Subsequently, the area occupied by the target vehicle is marked in the extracted viewport. This marked viewport is then fed into the inpainting network, which produces a realistic inpainted image with the vehicle removed. The inpainted viewport is then re-projected back into the equirectangular format and integrated into the original panoramic image collection, replacing the corresponding region containing the target vehicle.

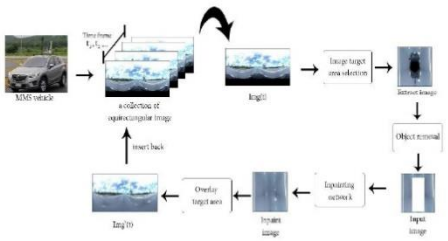


Fig. 1. The inpainting pipeline.

A. Target View Selection

Panoramic images typically capture omnidirectional visual information spanning a wide field of view. Processing and inpainting these panoramic scenes directly can lead to a significant increase in computational complexity and processing time requirements. Furthermore, directly applying inpainting techniques designed for planar images may produce unrealistic results in panoramic projections. Although some project methods [13][14] can be used to facilitate the inpainting process, they are not specifically design the particular task of targeted object (e.g. vehicles) removal from panoramic scenes.

Since panoramic images offer a full 360-degree view, we can easily extract a specific view to focus on a target object, like a vehicle. To do this, we first convert the panoramic image into a format in spherical coordinates.

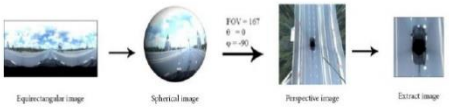


Fig. 2. Target View Selection Pipeline

This research budget was allocated by National Science, Research and Innovation Fund (NSRF), and King Mongkut's University of Technology North Bangkok (Project no. KMUTNB-FF-67-B-34).

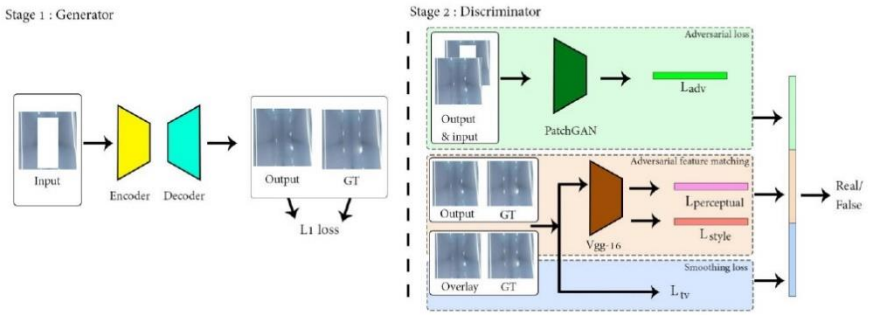


Fig. 3. The inpainting network

We then pinpoint the vehicle's location using cartesian coordinates. Next, we adjust our field of view to 167 degrees, ensuring we capture the vehicle and the surrounding road without distortion. We take a 1000 x 1000 pixel snapshot of this view and then crop it down to a smaller 256 x 256 pixel image, keeping the vehicle centered. The view selection process is show in Fig. 2. Once the target view is extracted, we use binary mark to overlay the area of the target object for image inpainting using Pix2Pix.

B. Inpainting

Once we have marked the object for removal in the selected view, we use an inpainting network to eliminate it. We feed a 256×256 RGB image with the marked object into the network. In our network, the generator built on a U-Net architecture, is used to fill in the marked region. The U-Net has the encoder that breaks the image down into smaller parts (256×256, 128×128, 64×64, 32×32, 16×16, 8×8, 4×4 and 1×1) and the decoder that reconstructs the image with resolutions of 1×1, 4×4, 8×8, 16×16, 32×32, 64×64, 128×128 and 256×256. The discriminator network is based on PatchGAN. The PatchGAN evaluates the inpainting result by dividing the image into 7×7 patches and classifying each patch as real or fake. The final output is an average of the patch-level decisions.

C. Adversarial feature matching

In order to make a realistic image, the adversarial loss is improved by incorporating a feature matching loss based on discriminator. This loss leverages the discriminator network to make the generator focuses not only fool the discriminator, but also produces similar high-level features to the real images. In this study, we use pre-trained VGG16 model to generate and compare high-level features of both real and generated images.

Inspired by [15], we first take into account perceptual loss to ensure both the real and generated images maintain a consistent visual style.

$$\mathscr{L}_{perceptual} = \sum_{p=0}^{P-1} \frac{\left\| f_p^{I_{out}} - f_p^{I_{gt}} \right\|_1}{N_{f_p^{I_{gt}}}} + \sum_{p=0}^{P-1} \frac{\left\| f_p^{I_{over}} - f_p^{I_{gt}} \right\|_1}{N_{f_p^{I_{gt}}}} \quad (5)$$

where I_{over} is the network prediction overlay I_{in}, the perceptual loss is measure perceptual difference in content and style between image from using an ImageNet-pretrained VGG-16 [16] by measure different by using L1 distance between generated data both I_{over} and I_{out} and the ground truth I_{gt}. $f_p^{I_*}$ is the feature map of the pth selected layer given I_{out}, I_{over}, and I_{gt}, respectively. We use layers block1_conv2, block2_conv2, block3_conv3, and block4_conv3.

We also include style loss [17] to guarantee the inpainted area that matches the overall style [18] and textures of the target image. Like perceptual loss, we extract features that represent textures and color patterns using pre-trained VGG16 network and compute the styles from I_{out} and I_{gt} on convolutional feature map. The style loss can be defined as follow:

$$\mathscr{L}_{\text{style}} = \sum_{p=0}^{P-1} \frac{1}{C_p C_p} \left\| K_p \left((f_p^{I_{out}})^T (f_p^{I_{out}}) - (f_p^{I_{gt}})^T (f_p^{I_{gt}}) \right) \right\|_1 + \sum_{p=0}^{P-1} \frac{1}{C_p C_p} \left\| K_p \left((f_p^{I_{over}})^T (f_p^{I_{over}}) - (f_p^{I_{gt}})^T (f_p^{I_{gt}}) \right) \right\|_1 \quad (6)$$

We note that the matrix operation assumes that the high-level features $f(x)_p$ is shape $(H_p W_p) \times C_p$, resulting in a $C_p \times C_p$ Gram matrix, and K_p is the normalization factor $1 / C_p H_p W_p$ for the pth selected layer.

By combining loss function in eq. (5) (6) into a set of equations. We denote this set of equations as Adversarial feature matching loss.

$$\mathscr{L}_{feature} = \lambda_{perceptual} \mathscr{L}_{perceptual} + \lambda_{style} \mathscr{L}_{style} \quad (7)$$

In addition, we consider total variation loss (L_{tv}) [12] that promotes smoothness and reduces noise in the inpainted area,

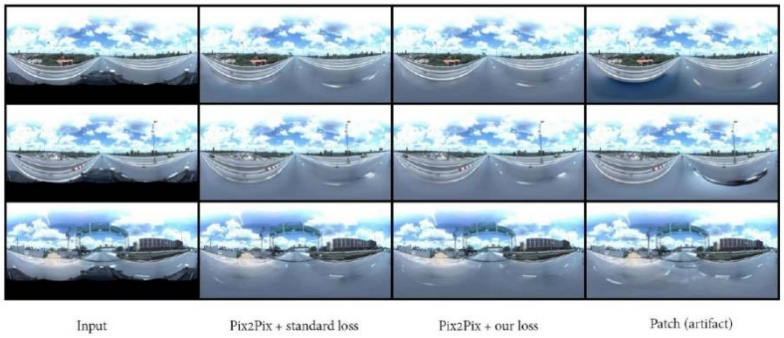


Fig. 7. Inpainting results on test image samples

Fig. 7 illustrates the inpainted results after removing vehicles in panoramic input images. As shown in Fig. 7, Pix2Pix with our proposed loss gives better inpainting quality compared to the baseline Pix2Pix model, successfully removing vehicles while preserving the contextual details and structural consistency of the surrounding scene. Fig. 8 presents a perspective view image that zooms in on the inpainted area, demonstrating that our method can effectively retain the structure and details of road lanes after inpainting and removing vehicles.

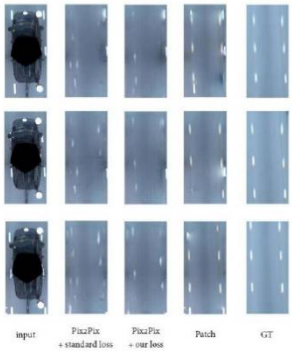


Fig. 8. Inpainted results on reconstructing road lanes after vehicle removal where GT denotes ground truth images

Since a large collection of panoramic images can be obtained from MMS surveying, we compare the processing time required for inpainting, as shown in Table II.

TABLE II. COMPARISON RESULT ON TIME PROCESSING

Approach	100 image (sec)	1000 image (sec)
Patch-based model	362	3260
Pix2Pix model	**304**	**2970**

As seen in Table II, we compare the processing time required for inpainting 100 and 1000 panoramic images using two different methods: a patch-based model [20] and the Pix2Pix model proposed in this study. For inpainting 100 images, the patch-based model took 362 seconds, while the Pix2Pix model was more efficient, requiring only 304 seconds. For the case of 1,000 images, The patch-based approach took 3260 seconds, whereas the Pix2Pix model completed the task in 2970 seconds. These results demonstrate that the proposed Pix2Pix model offers a significant improvement in computational efficiency compared to traditional patch-based inpainting methods, especially when dealing with large-scale datasets commonly encountered in mobile mapping applications.

V. CONCLUSION

This study presented a novel Generative Adversarial Network-based approach for inpainting unwanted vehicles from panoramic scenes in mobile mapping system (MMS). Our method leverages Pix2Pix, a powerful image-to-image translation GAN architecture, to effectively remove vehicles while preserving the overall scene quality. To address the challenges of inpainting in panoramic images, we introduced a view selection process that focuses on specific regions containing the target vehicles. Furthermore, we enhanced Pix2Pix framework by incorporating feature matching losses, leading to a significant improvement in generating more realistic. Experimental results on real-world MMS image datasets demonstrated that our proposed method not only achieves superior inpainting quality compared to standard Pix2Pix, but also efficiently handle large-scale image processing. These results suggest new possibilities for improving privacy protection of sensitive information in the field of MMS.

ACKNOWLEDGMENT

This research budget was allocated by National Science, Research and Innovation Fund (NSRF), and King Mongkut's University of Technology North Bangkok (Project no. KMUTNB-FF-67-B-34).

REFERENCES

[1] M. Elhashash, H. Albanwan, and R. Qin, "A Review of Mobile Mapping Systems: From Sensors to Applications," *Sensors*, vol. 22, Article 11, 4262, June 2022.

[2] A. Frome, G. Cheung, A. A., M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, "Large-scale privacy protection in Google Street View," in *Proc. of the IEEE International Conference on Computer Vision*, 2009, pp. 2373 - 2380.

[3] K. Chinomi, N. Nitta, Y. Ito, and N. Babaguchi, "PriSurv: privacy protected video surveillance system using adaptive visual abstraction," in *Proc. of the 14th international conference on Advances in multimedia modeling (MMM'08)*, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 144-154.

[4] T. Zhou, B. Johnson, and R. Li, "Patch-based texture synthesis for image inpainting," 2016. [Online]. Available: arXiv:1605.01576 [cs.CV].

[5] N. Zhang, H. Ji, L. Liu, and G. Wang, "Exemplar-based image inpainting using angle-aware patch matching," *EURASIP Journal on Image and Video Processing 2019*, Article 70, July 2019.

[6] S. Yang, H. Liang, Y. Wang, H. Cai, and X. Chen, "Image Inpainting Based on Multi-Patch Match with Adaptive Size," *Applied Sciences*, 10, Article 14, 4921, July 2020.

[7] H. Li; W. Luo; and J. Huang, "Localization of Diffusion-based Inpainting in Digital Images," *IEEE Trans. on Information Forensics and Security*, Vol. 12, no. 12, pp. 3050 - 3064, December 2017.

[8] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." *Advances in neural information processing systems* 27 2014.

[9] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T. and Efros, A. A. "Context encoders: Feature learning by inpainting". In *Proceedings of the IEEE conference on computer vision and pattern recognition* pp. 2536-2544. 2016.

[10] Iizuka, S., Simo-Serra, E., & Ishikawa, H. "Globally and locally consistent image completion". ACM Transactions on Graphics (ToG), 36(4), pp. 1-14, 2017.

[11] Iizuka, S., Simo-Serra, E. and Ishikawa, H. "Globally and locally consistent image completion". ACM Transactions on Graphics (ToG), 36(4), pp. 1-14, 2017.

[12] G. Liu, F. A. Reda, K. J. Shih, T. C. Wang, A. Tao, and B. Catanzaro, "Image Inpainting for Irregular Holes Using Partial Convolutions," Computer Vision – ECCV 2018, 2018, pp. 89-105.

[13] Isola, P., Zhu, J. Y., Zhou, T. and Efros, A. A. "Image-to-image translation with conditional adversarial networks". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125-1134, 2017.

[14] Z. Shang, Y. Liu, G. Li, Y. Zhang, J. Miao, J. Liu, and L. Wang, "Viewport-Oriented Panoramic Image Inpainting," presented at the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 2022, pp. 3031 - 3035.

[15] S. W. Han, and D. Y. Suh, "PIINET: A 360-degree panoramic image inpainting network using a cube map," 2022. Available: arXiv:2010.16003.

[16] Johnson, J., Alahi, A. and Fei-Fei, L. "Perceptual losses for real-time style transfer and super-resolution". In Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, pp. 694-711, 2016.

[17] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, Available: arXiv:1409.1556.

[18] Debbie Honghee Ko, Ammar Ul Hassan, Jungjae Suk, and Jaeyoung Choi, "Korean Font Synthesis with GANs," International Journal of Computer Theory and Engineering vol. 12, no. 4, pp. 92-96, 2020.

[19] Hore, A., and Ziou, D. "Image quality metrics: PSNR vs. SSIM". In the 20th international conference on pattern recognition, pp. 2366-2369, 2010.

[20] Goyal, P., & Diwakar, S. "Fast and enhanced algorithm for exemplar based image inpainting". In 2010 Fourth Pacific-Rim Symposium on Image and Video Technology, pp. 325-330, 2010.


STRI
KMUTNB
INVENTION TO INNOVATION
RI2C
2022
BANGKOK, THAILAND
Research, Invention, and Innovation Congress
PROCEEDINGS
2022 Research, Invention, and Innovation Congress
Innovation Electricals and Electronics (RI2C)
A VIRTUAL CONFERENCE
4-5 August 2022 | Bangkok, Thailand
At King Mongkut's University of Technology North Bangkok
www.ri2c-2022.com
Technical Co-sponsors
ECTI Association
IEEE Computational Intelligence Society Thailand Chapter
IEEE THAILAND SECTION
THAI TOHKEN THERMO CO.,LTD.

Adaptive Multi-Loss GAN for Object Removals in Street-View Scenes

Arzeezar Lording
Department of Computer and Information Science
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
s6504062810021@email.kmutnb.ac.th

Luepol Pipanmaekaporn
Department of Computer and Information Science
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
luepol.p@sci.kmutnb.ac.th

Suwatchai Kamonsantiroj
Department of Computer and Information Science
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
suwatchai.k@sci.kmutnb.ac.th

Kanabadee Srisomboon
Department of Electrical and Computer Engineering
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
kanabadee.s@eng.kmutnb.ac.th

***Abstract*— Street-view images often contain personal and sensitive information like pedestrians and vehicles, leading to privacy concerns. It is essential to anonymize these sensitive objects from street-view images without sacrificing image quality. In this study, we introduce a novel synthesis-based object removals in street-view images, named Adaptive Multi-Loss GAN. Our proposed method first extracts area of target objects within a raw street-view image through viewport extraction. Subsequently, a GAN-based network with multiple loss functions is employed to reconstruct the inpaint area. However, a simple weighted average of multi-loss functions may be insufficient and unstable for optimizing the inpaint network. To overcome this issue, we introduce an adaptive weighting strategy. This dynamic approach adjusts the weights of the loss functions based on gradient descent information, facilitating the optimization of the GAN-based network. Extensive experimental evaluations on street-view images datasets demonstrate that our approach achieves the improvement of standard metrics, including Mean-Squared-Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) over traditional methods that employ a fixed weighted average.**

Keywords—Object inpaints, Generative Adversarial Network, multi-loss functions, adaptive weighting

I. INTRODUCTION

Mobile Mapping Systems (MMS) [1] have emerged as indispensable tool for rapidly collecting geospatial data, including 360 degree images, LiDAR point clouds and GPS coordinates, aiming to generate digital maps from vehicle platforms. Among these systems, street-level MMS are widely used for mapping urban landscapes. Nonetheless, street-view images from MMS often capture images of individuals and vehicles, raising privacy issues. Early efforts to mitigate these privacy concerns involves anonymizing these sensitive objects within the raw images by applying blurring techniques [2][3]. Although these methods enhance privacy protection, they often introduce artificial textures, which can detract from the image quality.

Previous works have addressed this challenge by inpainting, which fills in the area of object's image by using image backgrounds. These methods explored various techniques like copying background image patches from other areas [4, 5, 6] or even reconstructing the missing region with surrounding pixels [7]. However, these techniques struggle with MMS due to potentially large missing regions and the complexity of the scenes.

Recently, Generative Adversarial Networks (GANs) [8] have been applied in image inpainting, producing realistic content to fill missing image regions. Early breakthroughs include the Context Encoder (CE) [9] used encoder-decoder architecture to leverage surrounding context for inpainting. A two-stage network employing separate discriminators to ensure both global image structure and local texture details within the repaired area has been introduced [10]. Free-form inpainting [11] that utilizes partial convolution layers, enabling to handle missing regions of any shape and size. Nevertheless, directly using these GAN-based methods for inpainting street-view imagery poses significant challenges. Firstly, distortions from equirectangular projections can negatively impact inpainting quality. Second, the size and resolution of street-view images often lead to significant computational burdens. Thirdly, the unique ratio of equirectangular images pose a challenge for deep learning models trained on standard planar images. Thus, GAN-based inpainting for street-view images requires a specialized approach to handle the unique challenges.

Recently, a GAN-based approach for inpainting street-view images was introduced [12]. This method utilizes a modified Pix2Pix architecture [13] to fill in the masked regions with realistic content. Furthermore, the GAN network was optimized by combining loss functions using a weighted average strategy. However, selecting the optimal weights for each loss function is a careful and time-consuming process. Moreover, training with fixed weights may lead to suboptimal performance, as they cannot adapt to different stages of training, leading to poor quality inpainting.

In this study, we propose a novel GAN-based method for inpainting objects in street-view imagery, called Adaptive Multi-loss GAN. Our method utilizes a modified Pix2Pix architecture as the inpainting network. Instead of combining multiple loss functions with a fixed weight average, we employ an adaptive weighting scheme to dynamically adjust the weights, enabling the model to balance the influence of each loss function during the training process. This results in enhanced inpainting performance and improved stability. Experimental results on street-view image datasets demonstrate that the proposed method achieves inpainting vehicles in street-view panoramic images while efficiently processing a large number of images.

II. PRELIMINARY

A. Generative Adversarial Network (GAN)

Generative Adversarial Network (GAN) was first introduced in [8] as a powerful generative model. The GAN consists of two neural networks: a generator and a discriminator. The generator learns to fit the distribution of training data. The discriminator is used to spot the input data synthesized by the generator. In GAN, both the generator and

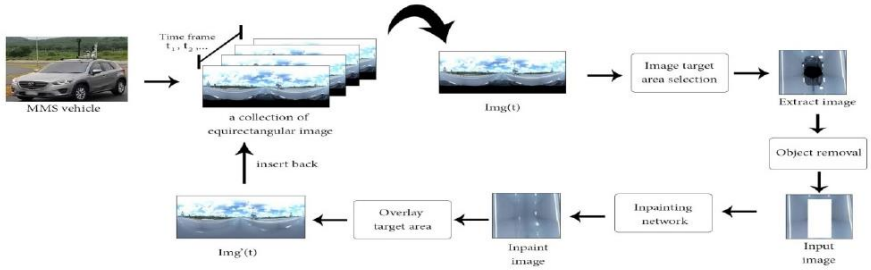


Fig. 1. The overall inpaint pipeline.

the discriminator are trained simultaneously by optimizing the adversarial loss function below:

$$L_{GAN} = \min_G \max_D V(D,G) = \mathrm{E}_{x\sim p_{data}(x)}\left[\log D(x\,|\,y)\right] + \mathrm{E}_{z\sim p_z(z)}\left[\log\left(1 - D(G(z\,|\,y))\right)\right] \quad (1)$$

random noise z obeying the $p_z(z)$ distribution generates x from the generator G and then inputs the generated data and the real data x into the discriminator D that identifies the input x is a real sample or a generated sample by G. At present, GANs have become the state-of-the-art in image inpainting to achieve more realistic results.

B. Pix2Pix

Pix2Pix [13] is a conditional GAN model designed for image-to-image translation tasks. This network learns to generate new images based on reference and target images. Pix2Pix network contains a U-NET architecture as the generator, which takes a reference image and learns to generate a new image based on target. The discriminator is a PatchGAN that analyzes small patches of the input image as real or generated and combines the results to improve the generator. The loss function of Pix2Pix GAN is a combination of adversarial loss and L1 function. The L1 loss function acts to minimize the sum of errors between the target image and the generated image, producing less blur in the output image.

$$\mathrm{L}_{Pix2Pix} = \min_G \max_D \mathrm{L}_{\mathrm{PGAN}}(G,D) + \lambda_{L1}\mathrm{L}_{L1}(G) \quad (2)$$

where $\mathscr{L}_{\mathrm{L1}} = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]$ denotes L1 loss function, λ_{L1} is a hyper-parameter that specifies the importance of L1 error and L_{GAN} can be defined as follows:

$$\mathrm{L}_{\mathrm{GAN}} = \mathrm{E}_{\mathrm{x,y}}\left[\log \mathrm{D}(\mathrm{x},\mathrm{y})\right] + \mathrm{E}_{\mathrm{x,z}}\left[\log\left(1 - \mathrm{D}(\mathrm{x},\mathrm{G}(\mathrm{x},\mathrm{z}))\right)\right] \quad (3)$$

Where x denotes the input sample and y is the target output image and $G(x)$ denotes the generated output image. In Pix2Pix, the discriminator aims to classify whether each $n\times n$ patch in an image is real or fake. The loss function of PatchGAN discriminator can be written as follows:

$$L_{PGAN} = \mathrm{E}_{x,y}\left[\sum_{i=1}^{N_p} \log D(x_i, y_i)\right] + \mathrm{E}_{x,z}\left[\sum_{i=1}^{N_p} \log\left(1 - D(x, G(x,z))\right)\right] \quad (4)$$

where N_p denotes the total number of N×N patches in the image.

III. Proposed Method

The proposed method is illustrated in Fig. 1. Given a collection of panoramic images captured by MMS vehicle, each image containing a target vehicle is selected to a view selection process. This process aims to generate a narrow viewport focused on the target vehicle within the panoramic scene. Subsequently, the area occupied by the target vehicle is marked in the extracted viewport. This marked viewport is then fed into the inpainting network, which produces a realistic inpainted image with the vehicle removed. The inpainted viewport is then re-projected back into the equirectangular format and integrated into the original panoramic image collection, replacing the corresponding region containing the target vehicle.

A. Target image selection

Panoramic images typically capture omnidirectional visual information spanning a wide field of view. Processing and inpainting these panoramic scenes directly can lead to a significant increase in computational complexity and processing time requirements. Furthermore, directly applying inpainting techniques designed for planar images may produce unrealistic results in panoramic projections. Although some project methods [14][15] can be used to facilitate the inpainting process, they are not specifically design the particular task of targeted object (e.g. vehicles) removal from panoramic scenes.

Since panoramic images offer a full 360-degree view, we can easily extract a specific view to focus on a target object, like a vehicle. To do this, we first convert the panoramic image into a format in spherical coordinates. We then pinpoint the vehicle's location using cartesian coordinates. Next, we adjust our field of view to 167 degrees, ensuring we capture the vehicle and the surrounding road without distortion. We take a 1000 x 1000 pixel snapshot of this view and then crop it down to a smaller 256 x 256 pixel image, keeping the vehicle

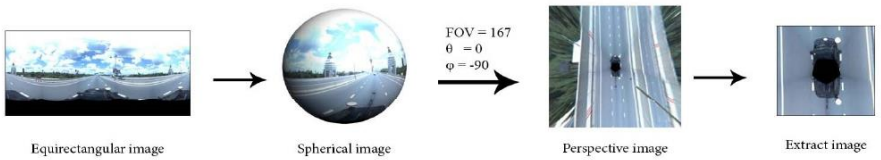


Fig. 2. The target image selection process

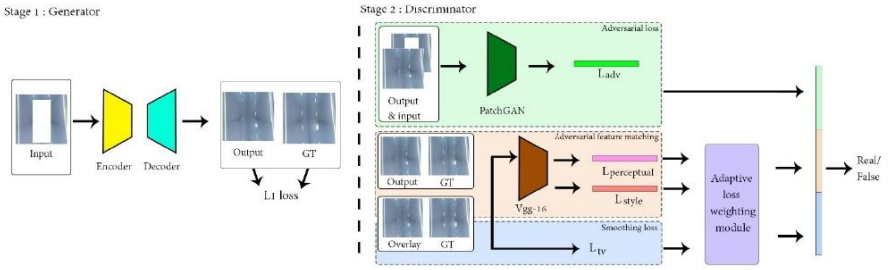


Fig. 3. The inpaining network

centered. The view selection process is shown in Fig. 2. Once the target view is extracted, we use binary mark to overlay the area of the target object for image inpainting using Pix2Pix.

B. The inpaint network

Fig. 3 illustrates the inpaint network architecture. This network is based on Pix2Pix architecture. It consists of two neural networks: a generator and discriminator. The generator takes an input image with masked (missing) regions. Once we have marked the object for removal in the selected view, we use an inpainting network to eliminate it. We feed a 256×256 RGB image with the marked object into the network. In our network, the generator built on a U-Net architecture, is used to fill in the marked region. The U-Net has the encoder that breaks the image down into smaller parts (256×256, 128×128, 64×64, 32×32, 16×16, 8×8, 4×4 and 1×1) and the decoder that reconstructs the image with resolutions of 1×1, 4×4, 8×8, 16×16, 32×32, 64×64, 128×128 and 256×256. The discriminator network is based on PatchGAN. The PatchGAN evaluates the inpainting result by dividing the image into 70×70 patches and classifying each patch as real or fake. The final output is an average of the patch-level decisions.

C. Adaptive Multi-Loss weighting

To enhance the GAN network ability for object inpainting, some methods [12] have proposed to optimize the network using various loss functions. The intuition is to enforce the network to balance these various objectives during training. We follow this scheme by modifying the adversarial loss in the discriminator network by incorporating the following additional loss functions.

Inspired by [16], our approach leverages perceptual loss to guarantee a consistent visual style across both the real images I_{gt} and the generated images I_{out}. This perceptual loss is defined as follows:

$$L_{percept} = \sum_{p=0}^{P-1} \frac{\left\| f_p^{I_{out}} - f_p^{I_{gt}} \right\|_1}{N_{f_p^{I_{gt}}}} + \sum_{p=0}^{P-1} \frac{\left\| f_p^{I_{over}} - f_p^{I_{gt}} \right\|_1}{N_{f_p^{I_{gt}}}} \quad (5)$$

where I_{over} denotes the network prediction overlay on the input image I_{in}. f_p^I denotes the feature map extracted from the p^{th} layer of a VGG-16 network [17] pre-trained on ImageNet. The perceptual loss quantifies the perceptual difference by comparing feature representations of the generated image, the overlaid prediction and the ground truth image. By minimizing this loss, the network learns to generate images that are stylistically consistent with the real images. In this study, we use VGG-16 layers of block1_conv2, block2_conv2, block3_conv3, and block4_conv3, respectively.

To enhance visual coherence, we incorporate style loss [18], ensuring that the texture generated within the inpainted regions blend with the overall style of the texture of the ground truth image. Similar to perceptual loss, we utilize a pre-trained VGG-16 network to extract features that capture textures and color patterns from both the generated images I_{out} and the ground truth images I_{gt}. The style loss can be defined as follow:

$$L_{style} = \sum_{p=0}^{P-1} \frac{1}{C_p C_p} K_p \left\| (f_p^{I_{out}})^T (f_p^{I_{out}}) - (f_p^{I_{gt}})^T (f_p^{I_{gt}}) \right\|_1 + \sum_{p=0}^{P-1} \frac{1}{C_p C_p} K_p \left\| (f_p^{I_{over}})^T (f_p^{I_{over}}) - (f_p^{I_{gt}})^T (f_p^{I_{gt}}) \right\|_1 \quad (6)$$

We note that the matrix operation assumes that the high-level features f_p^x is shape $(H_p W_p) \times C_p$, resulting in a $C_p \times C_p$ Gram matrix, and K_p is the normalization factor $1 / C_p H_p W_p$ for the p^{th} selected layer.

In addition, we consider total-variation loss (L_{tv}) [18] that encourages smoothness and discourages noise within the inpainted areas, leading to the enhanced visual quality . L_{tv} can be defined as follows:

$$L_{tv} = \sum_{(i,j)\in R,(i,j+1)\in R} \frac{\left\| I_{over}^{i,j+1} - I_{over}^{i,j} \right\|_1}{N_{I_{over}}} + \sum_{(i,j)\in R,(i,j+1)\in R} \frac{\left\| I_{over}^{i,j+1} - I_{over}^{i,j} \right\|_1}{N_{I_{over}}} \quad (7)$$

where R represents the inpainted region of the image. $I_{over}^{i,j}$ refers to the pixel intensity of the feature map from the input image at location i and j. $N_{I_{over}}$ denotes the total number of pixels in the inpainted region.

By combining the loss functions in eq. (5), (6) and (7) with the adversarial loss L_{GAN}, we formulate the discriminator loss for the discriminator network as shown in Eq. (8).

$$L_D = L_{GAN} + \alpha_{percept} * L_{percept} + \alpha_{style} * L_{style} + \alpha_{tv} * L_{tv} \quad (8)$$

where α_i represent a coefficient that controls the relative importance of the i^{th} loss function. Although training the network with a combination of these loss functions shows potential, determining the optimal weights for each component is a significant challenge. Manually tuning these weights is a laborious and time-consuming process. Moreover, relying on fixed weights during training can hinder performance, as the optimal balance between the different objectives may vary across training stages. The rigidity of the fixed weights can limit the quality of the inpainted results.

To address the challenge of weight optimization in multi-part loss functions, we leverage Softadapt [19]. This weighting method allows to dynamically adjust the weights assigned to each loss component based on their performance statistics during training. The weighting method favors to assign smaller weights to loss functions that are approaching their minima, as demonstrated by the following equation:

$$\alpha_k^i = \frac{f_k^i e^{\beta s_k^i}}{\sum_{\ell}^{n} f_\ell^i e^{\beta s_\ell^i}} \quad (9)$$

where α_k^i denotes a coefficient of the k^{th} loss component at iteration i. f_k^i is the value of the k^{th} loss function at iteration i. s_k^i indicates the average of the k^{th} loss function's values over recent iterations. β is a hyperparameter controlling the sensitivity of the weighting scheme to changes in loss values.

The pseudo-code for training the inpainting network is shown in Algorithm 1. The algorithm begins by creating a dataset to train the GAN network. This is done by extracting relevant regions within each raw street-view image in an image collection using the viewport selection algorithm described in Section III(A). Then, the images of target object within the extracted region are masked out using binary masks, enabling the generator network to fill in within the masked areas. During training, batches of masked images x and their corresponding ground truth I_{gt} are fed to the network. The generator attempts to reconstruct the mask regions by producing inpainted images I_{out}. Once the iteration t reaches at the defined interval T_{update}, the weights for the perceptual loss $\alpha_{percept}$, style loss α_{style}, and total variation loss α_{tv} are calculated based on Eq. (9) to balance the optimization. The discriminator (D) and generator (G) are then trained adversarially using losses from Eq. (8) and Eq. (2) respectively, with this process repeating until convergence is achieved.

Algorithm 1 Adaptive Multi-Loss GAN

Input: a collection of street-view images X,
a set of ground truth images I_{gt} ,
a set of binary mask M,
the batch size B,
an update interval T_{update}
Output: the trained generator G.

$P \leftarrow$ Viewport_Extraction (X)
$X \leftarrow$ Generate_Masked_Region (P, M)
$N_{batch} \leftarrow N_{lin} // B$
While not converged **do**
  for $i = 1, \ldots, N_{batch}$ **do**:
    for $j = 1, \ldots, B$ **do**:
      random an input sample x and its ground truth I_{gt}
      $I_{out} \leftarrow G(x)$
    end for
    if t mod $T_{update} = 0$ and t != 0 **then**
      update $\alpha_{percept}$, α_{style} and α_{tv} by Eq. (9)
    else:
      $s_{percept}^{t+1} = s_{percept}^{t} + l_{percept}^{t}$
      $s_{style}^{t+1} = s_{style}^{t} + l_{style}^{t}$
      $s_{tv}^{t+1} = s_{tv}^{t} + l_{tv}^{t}$
    end if
    $Disc_{real} \leftarrow D([x, I_{gt}])$
    $Disc_{gen} \leftarrow D([x, I_{out}])$
    $I_{over} \leftarrow$ Extract_Feature_Maps (I_{out}, I_{gt})
  end for
  Compute L_D and L_G using Eq. (8) and Eq. (2)
  D(t+1) ← Optimizer(D(t) , ∇L_D)
  G(t +1) ← Optimizer(G(t) , ∇L_G)
end while

IV. EXPERIMENTS

A. Data collection

The dataset was acquired by surveying the area of Maha-Chesadabodindranusorn Bridge to Wat Bot Don Prom in Bangkok Thailand, as illustrated in Fig. 4. The total surveyed distance was 4.3 kilometer, using a Ladybug camera equipped with a Global Navigation Satellite System (GNSS) to capture 4,096 × 2,048 resolution panoramic images along with their GPS coordinates, resulting in 1,969 panoramic images in total.

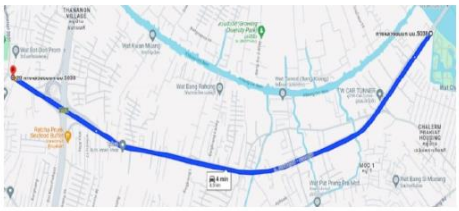


Fig. 4. Mobile Mapping Data Survey

B. Ground truth generation

Training a neural network requires ground truth data. However, in this dataset, ground truth images are not available. Specifically, we do not have images containing the background of the target vehicle. To address this, we can find the background region behind the vehicle from other images using GPS coordinates. Let (x_t, y_t) be the center coordinate of the target vehicle in the current image t and θ be the vehicle's direction. We calculate a point (x_q, y_q) located behind this vehicle at least a distance q using the following equations:

$$x_q = q_t * \cos(\theta_t + 180) + x_t$$

$$y_q = q_t * \sin(\theta_t + 180) + y_t$$

After obtaining the reference point (x_q, y_q), it is used to search for the image in the dataset that contain the background area corresponding to the location of the target vehicle and use this image to be the ground truth for the current image t. Fig . 5 illustrates example of training data.

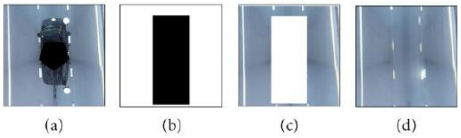


Fig. 5. a training sample, including (a) a target view image (b) binary mask (c) the input image and (d) the ground truth image respectively.

We split the dataset into 80% for training the network and 20% for testing.

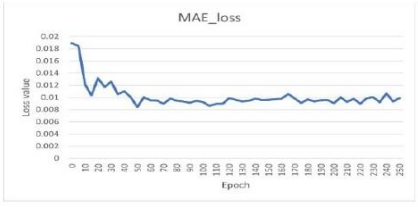


Fig 6. Training Loss

C. Experimental setting

To evaluate the effectiveness of our proposed method, we compare it against a fixed-weight multi-loss Pix2Pix as outlined in [12]. For the fixed-weight baseline, we set the weights to $\alpha_{percept}$ = 0.05, α_{style} = 120 and α_{tv} =0.1, respectively, as they are reported in their experiments obtained the best performance on this dataset. In contrast, our proposed method initializes all the loss function weights equally to 1.0, allowing the adaptive strategy to learn the optimal performance during training. Furthermore, the learning rate for both the generator and the discriminator is set to 0.0002. For the discriminator, we use 70 x 70 patches for the PatchGAN. We employed Adam's optimizer ($\beta 1$=0.5, $\beta 2$=0.99) to optimize the GAN loss, and used a batch size of 1. We conducted experiments with our proposed method on an Intel (R) Xeon(R) Silver 4214 CPU@ 2.20GHz processor with a Tesla V100-PCIE-16GB GPU.Fig. 6 plots training losses across the first 250 epochs.

D. Experiment results

We evaluate the quality of inpainted vehicles generated by different methods using three key metrics: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) [20]. Table I illustrates performance comparison of inpainting vehicles using different methods over three key metrics: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) [19]. As seen in Table 1, the proposed method (Pix2Pix with adaptive multi-loss outperforms the baseline Pix2Pix with fixed-weight multi-loss and Pix2Pix model that uses only adversarial loss. These quantitative results indicate that the adaptive multi-loss optimization enhances the Pix2Pix model to generate high-quality of the inpainted images.

TABLE I. PERFORMANCE COMPARISON ACROSS COMMON METRICS

Method	MSE (↓)	PSNR (↑)	SSIM (↑)
Pix2Pix (adversarial loss)	192.01	25.35	0.89
Pix2Pix(fixed-weight Multi- loss)[12]	177.84	25.73	0.90
Pix2Pix (adaptive-weight Multi-loss)	**150.33**	**26.19**	**0.91**

Fig. 7 illustrates the inpainted results on street-view samples. As shown in Fig. 7, the Adaptive-weight Multi-loss GAN produces higher-quality inpainted images compared to the method that uses only adversarial loss, which introduces artifacts. Moreover, compared to the fixed-weight multi-loss method, our proposed approach can generate sharper road lanes and mitigate blurriness over the inpainted regions. These results underscore the effectiveness of dynamically adjusting the weights for each loss function based on their performance.

E. Discussion

Table II examines the impact of different weight update intervals (T_{update}) on model performance. We observe that updating the multi-loss function weights every 5 epochs yields the best results. Fig. 8 shows inpainted results on test samples in different update intervals.

TABLE II. PERFORMANCE COMPARISON OF ADAPTIVE MULTI-LOSS GAN ACROSS VARYING INTERVAL UPDATE INTERVALS.

Interval update	MSE (↓)	PSNR (↑)	SSIM (↑)
$T_{update} = 2$	181.77	25.61	0.90
$T_{update} = 5$	**159.33**	**26.19**	**0.91**
$T_{update} = 7$	185.11	25.55	0.90
$T_{update} = 9$	195.57	25.31	0.89

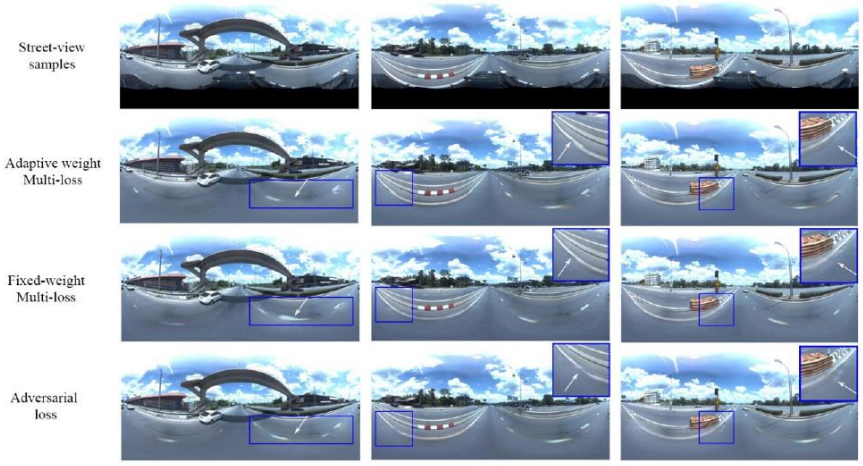


Fig. 7. Inpainted results on three test samples

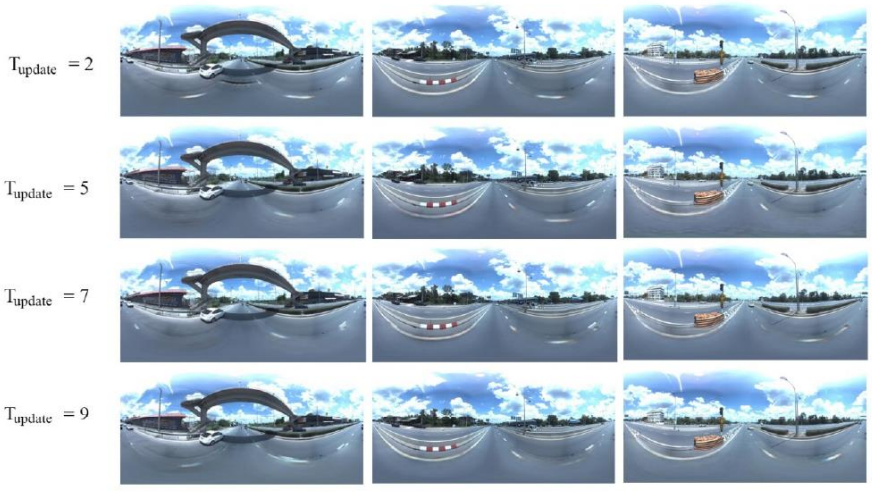


Fig. 8. Inpainted results of adaptive weighting multi-loss using different intervals

Fig. 9 demonstrates the impact of the generator loss over training epochs when using different update intervals. As shown in Fig. 9, the lowest loss, indicating an update interval of 7 epochs is optimal for this dataset while smaller intervals result in more fluctuations.

Fig 9. Generator Loss v.s. epoch using different intervals (n)

V. CONCLUSION

This study introduced Adaptive Multi-Loss GAN, a novel approach for anonymizing street-view images by effectively removing privacy-sensitive objects like pedestrians and vehicles. By incorporating an adaptive weighting strategy into the multi-loss function optimization of the GAN-based network, our method overcomes the limitations of traditional fixed-weight methods. This dynamic approach allows for a more balanced and effective training process, leading to higher-quality inpainted images. Extensive experiments conducted on street-view datasets demonstrate the superiority of our method, achieving significant improvements in MSE, PSNR, and SSIM metrics compared to traditional approaches. This research contributes a robust and efficient solution for privacy-preserving street-view image anonymization, effectively balancing privacy concerns with the need for high-quality visual data.

Future work could explore further refinements to the adaptive weighting scheme and investigate its application in other image editing tasks beyond object removal.

ACKNOWLEDGMENT

This research budget was allocated by National Science, Research and Innovation Fund (NSRF), and King Mongkut's University of Technology North Bangkok (Project no. KMUTNB-FF-67-B-34).

REFERENCES

[1] M .Elhashash, H .Albanwan, and R .Qin, "A Review of Mobile Mapping Systems :From Sensors to Applications, "*Sensors,* vol .22, Article 11, 4262, June 2022 .

[2] A .Frome, G .Cheung, A .A., M .Zennaro, B .Wu, A .Bissacco, H . Adam, H .Neven, and L .Vincent, "Large-scale privacy protection in Google Street View, "in *Proc .of the IEEE International Conference on Computer Vision*, 2009, pp .2373 -2380

[3] K .Chinomi, N .Nitta, Y .Ito, and N .Babaguchi, "PriSurv :privacy protected video surveillance system using adaptive visual abstraction, " in *Proc .of the 14th international conference on .Advances in multimedia modeling)MMM'08(*, Springer-Verlag, Berlin, Heidelberg, 2008, pp .144–154 .

[4] T .Zhou, B .Johnson, and R .Li, "Patch-based texture synthesis for image inpainting, "2016] .Online .[Available :arXiv:1605.01576]cs.CV .[

[5] N .Zhang, H .Ji, L .Liu, and G .Wang, "Exemplar-based image inpainting using angle-aware patch matching, "*EURASIP Journal on Image and Video Processing 2019*, Article 70, July 2019 .

[6] S .Yang, H .Liang, Y .Wang, H .Cai, and X .Chen, "Image Inpainting Based on Multi-Patch Match with Adaptive Size, "*Applied Sciences,* 10, Article 14, 4921, July 2020 .

[7] H .Li; W .Luo; and J .Huang, "Localization of Diffusion-based Inpainting in Digital Images, "*IEEE Trans .on Information Forensics and Security*, Vol.12, no.12, pp .3050 –3064, December 2017 .

[8] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio . "Generative adversarial nets "*.Advances in neural information processing systems* 27 2014 .

[9] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T .and Efros, A .A . "Context encoders :Feature learning by inpainting". In *Proceedings of the IEEE conference on computer vision and pattern recognition* pp . 2536-2544 .2016 .

[10] Iizuka, S., Simo-Serra, E .and Ishikawa, H ."Globally and locally consistent image completion". ACM Transactions on Graphics)ToG(, 36)4(, pp .1-14, 2017 .

[11] G .Liu, F .A .Reda, K .J .Shih, T.C .Wang, A .Tao, and B .Catanzaro, "Image Inpainting for Irregular Holes Using Partial Convolutions, " Computer Vision –ECCV 2018, 2018, pp .89-105 .

[12] Lording A., L. Pipanmaekaporn, S. Kamonsantiroj, K. Srisomboon. "GAN-Driven Vehicle Inpainting for Mobile Mapping Scenes". In Proceedings of the 9th International Conference on Control and Robotics Engineering (ICCRE), pp. 388-393, 2024.

[13] Isola, P., Zhu, J .Y., Zhou, T .and Efros, A .A ."Image-to-image translation with conditional adversarial networks". In Proceedings of the IEEE conference on computer vision and pattern recognition, pp . 1125-1134, 2017 .

[14] Z .Shang, Y .Liu, G .Li, Y .Zhang, J .Miao, J .Liu, and L .Wang, "Viewport-Oriented Panoramic Image Inpainting, "presented at the 2022 IEEE International Conference on Image Processing)ICIP(, Bordeaux, France, 2022, pp .3031 -3035 .

[15] S .W .Han, and D .Y .Suh, "PIINET :A 360-degree panoramic image inpainting network using a cube map, "2022 .Available : arXiv:2010.16003 .

[16] Johnson, J., Alahi, A .and Fei-Fei, L ."Perceptual losses for real-time style transfer and super-resolution". In Computer Vision–ECCV 2016 : 14th European Conference, Amsterdam, The Netherlands, pp.694-711, 2016 .

[17] K .Simonyan, and A .Zisserman, "Very deep convolutional networks for large-scale image recognition, "2014, Available :arXiv:1409.1556 .

[18] Debbie Honghee Ko, Ammar Ul Hassan, Jungjae Suk, and Jaeyoung Choi, "Korean Font Synthesis with GANs," International Journal of Computer Theory and Engineering vol. 12, no.4, pp. 92-96, 2020.

[19] Heydari, A. A., Thompson, C. A., & Mehmood, A. (2019). Softadapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions. *arXiv preprint arXiv:1912.12355*.

[20] Hore, A., and Ziou, D ."Image quality metrics :PSNR vs .SSIM". In the 20th international conference on pattern recognition, pp .2366-2369, 2010 .

ภาคผนวก ข

วิธีการติดตั้ง

วิธีการติดตั้งระบบ

1. การติดตั้ง Python

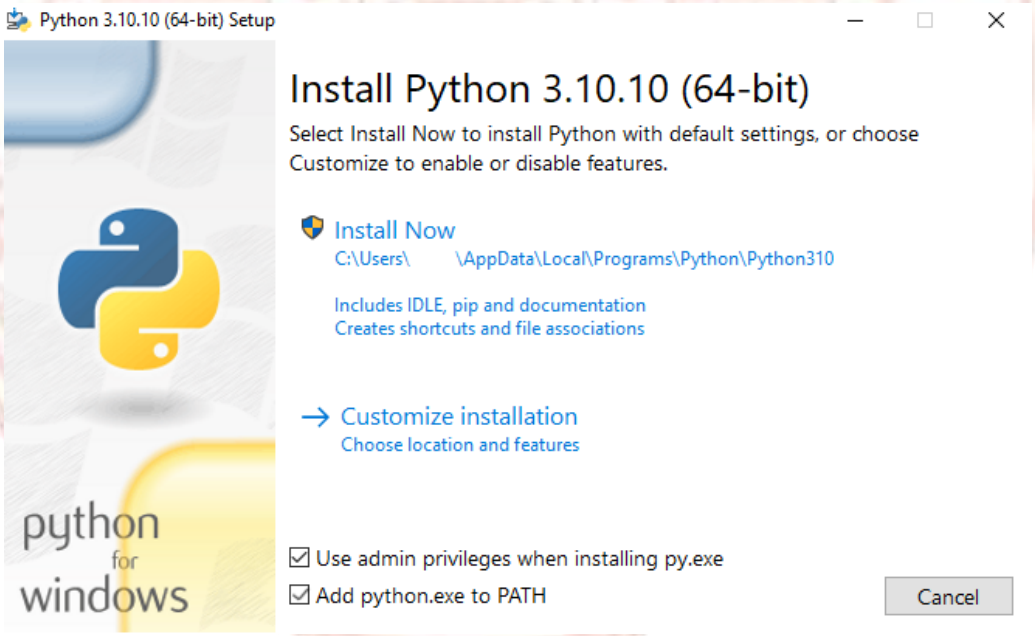
1.1 ดาวน์โหลดไฟล์ติดตั้งภาษา Python จากเว็บไซต์ https://www.python.org/downloads / โดยเลือก Python version 3.10 เป็นต้นไป

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore	
Gzipped source tarball	Source release		6dbe644dd1a520d9853cf6648084c346	24.9 MB	SIG	.sigstore	
XZ compressed source tarball	Source release		7bf85df71bbe7f95e5370b983e6ae684	18.7 MB	SIG	.sigstore	
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	892634724ab799569b512082c8f48c83	39.1 MB	SIG	CRT	SIG
Windows installer (64-bit)	Windows	Recommended	9735797853cba809b13c8396c91354a0	27.7 MB	SIG	CRT	SIG
Windows installer (32-bit)	Windows		a81b81687bc2575c05a30f4b31d6ea00	26.6 MB	SIG	CRT	SIG
Windows help file	Windows		448f8401ade49a7e2156d02512f2f9bf	9.0 MB	SIG	CRT	SIG
Windows embeddable package (64-bit)	Windows		f38a9e7e02a992daa62569b758d0a388	8.2 MB	SIG	CRT	SIG
Windows embeddable package (32-bit)	Windows		a681a7f9b242fe35b4d96d79e15e57d6	7.3 MB	SIG	CRT	SIG

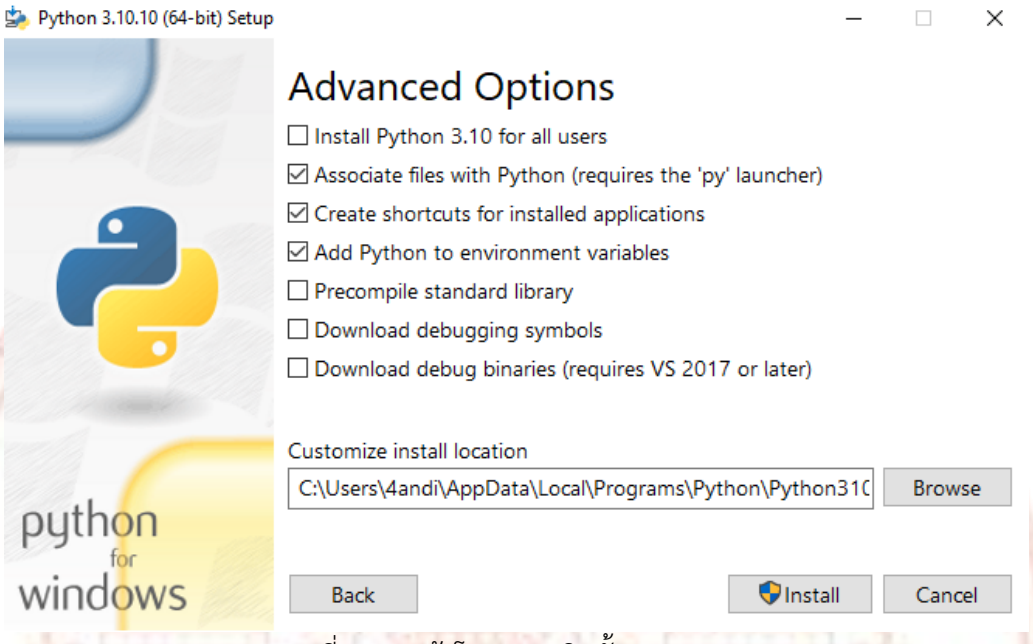
ภาพที่ ข-1 หน้าเว็บไซต์ดาวน์โหลด Python

1.2 เมื่อโปรแกรมดาวน์โหลดเสร็จ เปิดไฟล์เพื่อทำการติดตั้ง จะขึ้นหน้าต่างให้ทำการเลือก “Add Python 3.10 PATH” และกดปุ่ม “Customize installation”



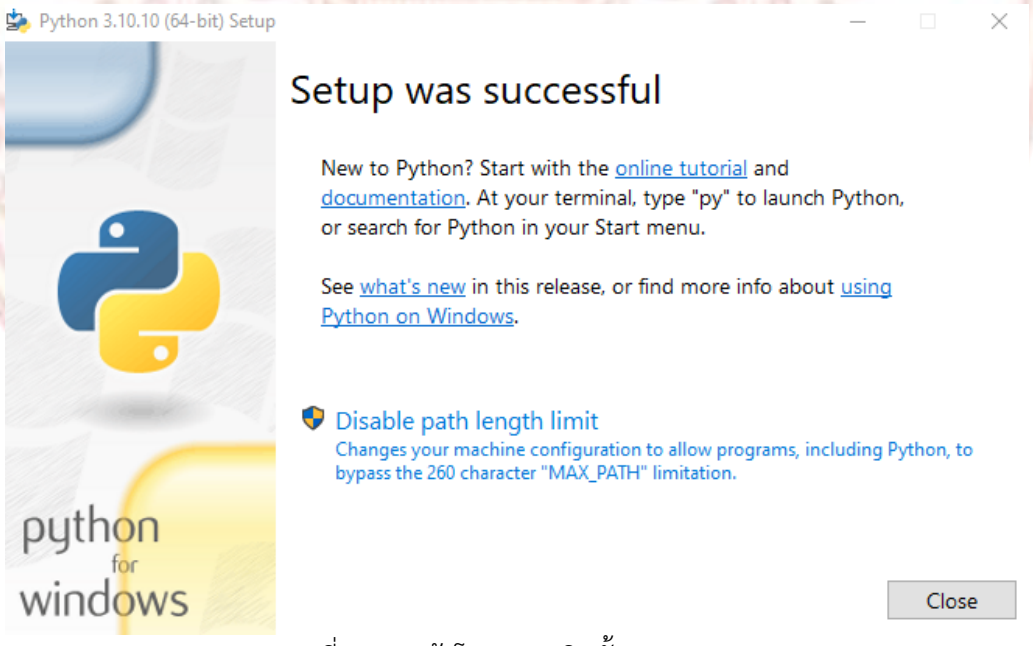
ภาพที่ ข-2 หน้าโปรแกรมติดตั้ง Python (1)

1.3 จากนั้นเลือก “install for all users” แล้วกดปุ่ม “Install”



ภาพที่ ข-3 หน้าโปรแกรมติดตั้ง Python (2)

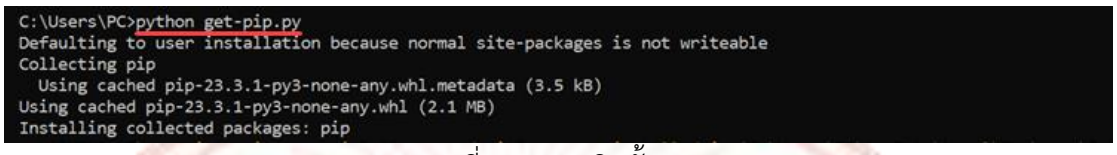
1.4 จากนั้นรอให้โปรแกรมทำการติดตั้งเสร็จ แล้วกดปุ่ม “Close”



ภาพที่ ข-4 หน้าโปรแกรมติดตั้ง Python (3)

1.5 เมื่อทำการติดตั้ง Python เสร็จ ก็ทำการติดตั้ง Packages เสริมของ Python ที่ชื่อ pip โดยให้ดาวน์โหลดจากเว็บไซต์ https://bootstrap.pypa.io/get-pip.py

1.6 เมื่อดาวน์โหลดเสร็จให้เปิด command prompt แล้วไปยัง path ของไฟล์ที่ดาวน์โหลดจากนั้นพิมพ์คำสั่ง “python get-pip.py”



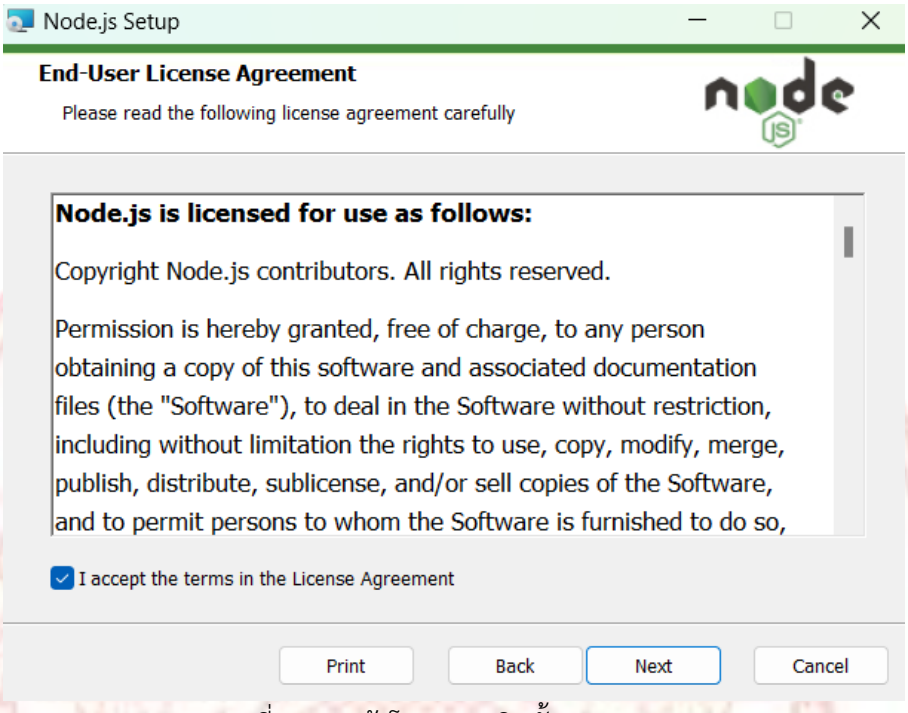
ภาพที่ ข-5 การติดตั้ง pip

2. การติดตั้ง Node.js

2.1 ดาวน์โหลดไฟล์ติดตั้ง Node.js จากเว็บไซต์ https://nodejs.org/en/download/ โดยเลือก “Windows” เลือก “x64 architecture” แล้วกด “Windows Installer (.msi)”

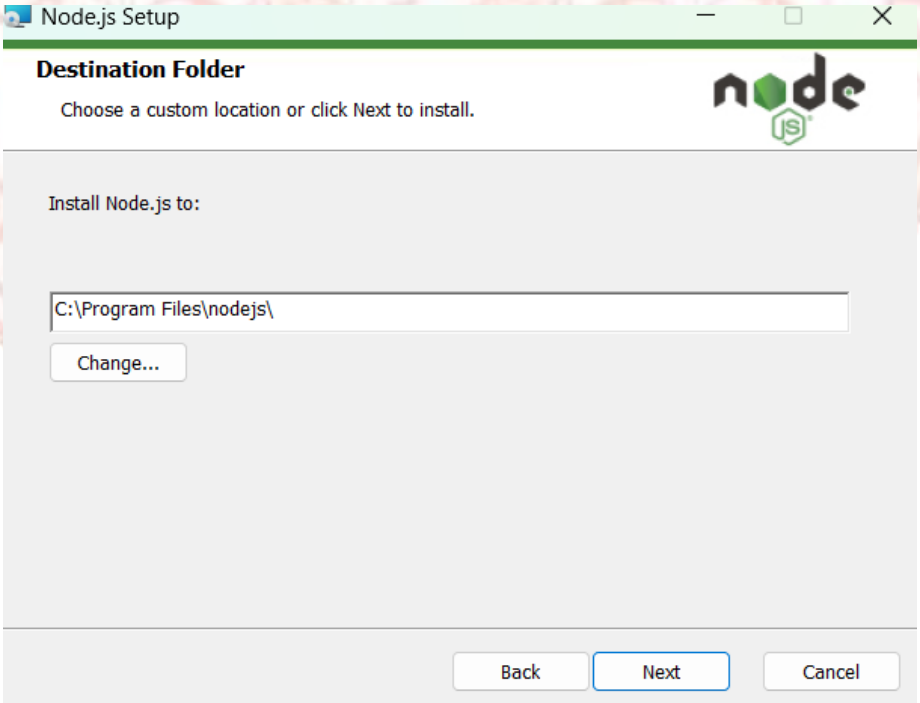
ภาพที่ ข-6 หน้าเว็บไซต์ดาวน์โหลด Node.js

2.2 เมื่อโปรแกรมดาวน์โหลดเสร็จ เปิดไฟล์เพื่อทำการติดตั้ง จะขึ้นหน้าต่างให้ทำการเลือก “I accept the terms and the License Agreement” จากนั้นกดปุ่ม “Next”



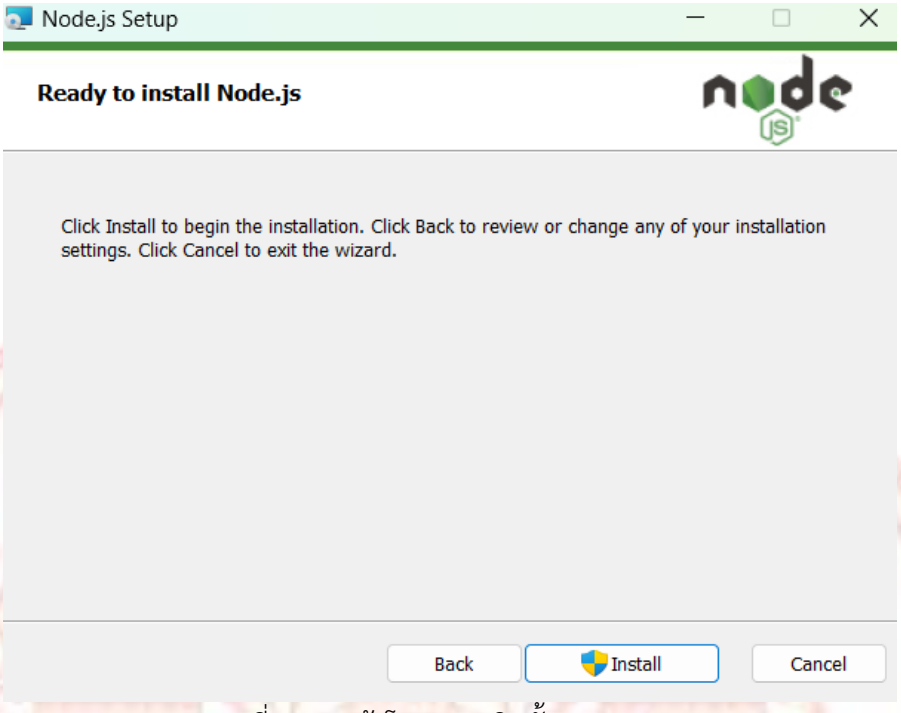
ภาพที่ ข-7 หน้าโปรแกรมติดตั้ง Node.js (1)

2.3 กำหนดตำแหน่งที่จะติดตั้งตามที่ต้องการ แล้วกดปุ่ม “Next”



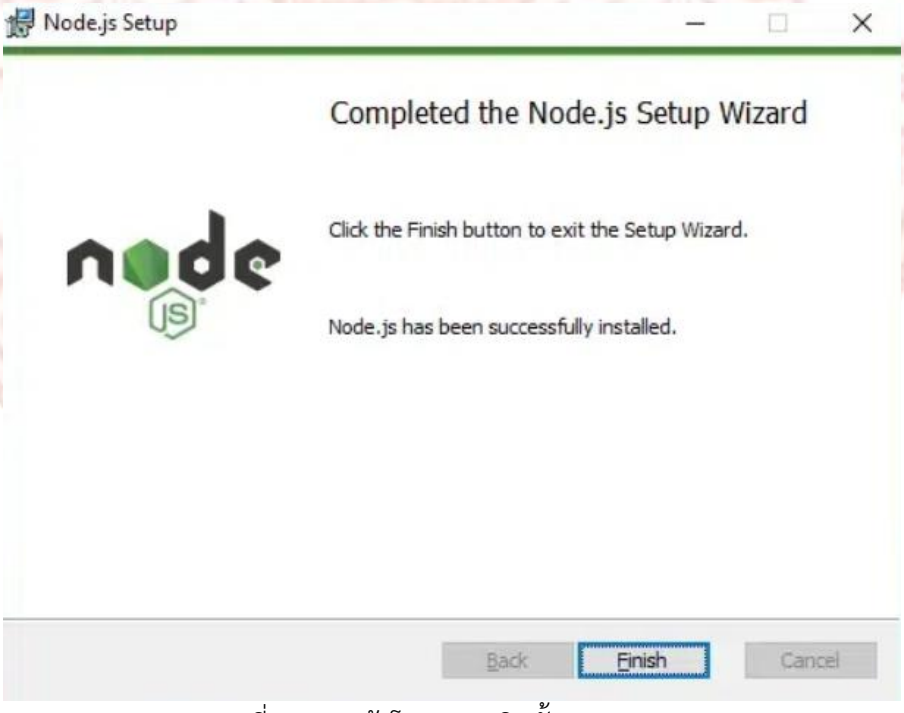
ภาพที่ ข-8 หน้าโปรแกรมติดตั้ง Node.js (2)

2.4 กดปุ่ม “Install” เพื่อเริ่มการติดตั้ง



ภาพที่ ข-9 หน้าโปรแกรมติดตั้ง Node.js (3)

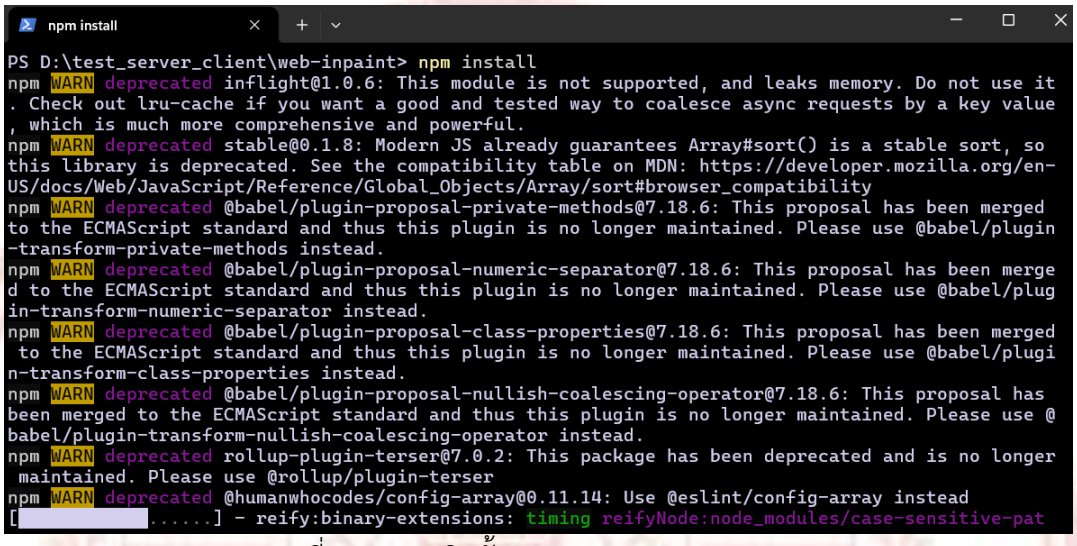
2.5 กดปุ่ม “Finish” เพื่อจบการติดตั้ง



ภาพที่ ข-10 หน้าโปรแกรมติดตั้ง Node.js (4)

3. การติดตั้ง Package ของ Web Client

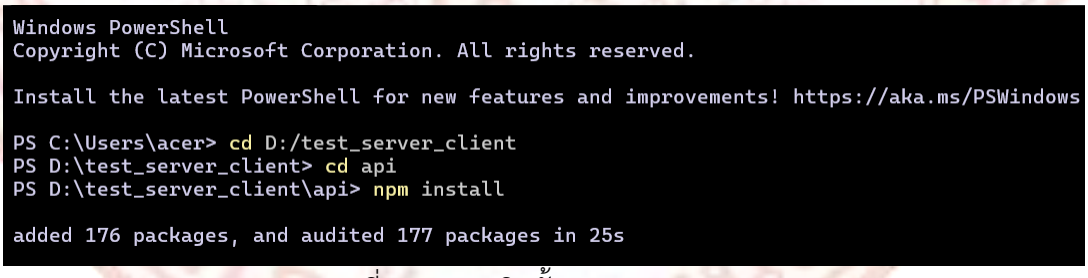
เปิด Command prompt หรือ Windows PowerShell แล้วไปยัง path ของโฟลเดอร์ web-inpaint แล้วพิมพ์คำสั่ง “npm install” เพื่อติดตั้ง Packages ที่เกี่ยวข้อง



ภาพที่ ข-11 การติดตั้ง Package ของ Web Client

4. การติดตั้ง Package ของ API

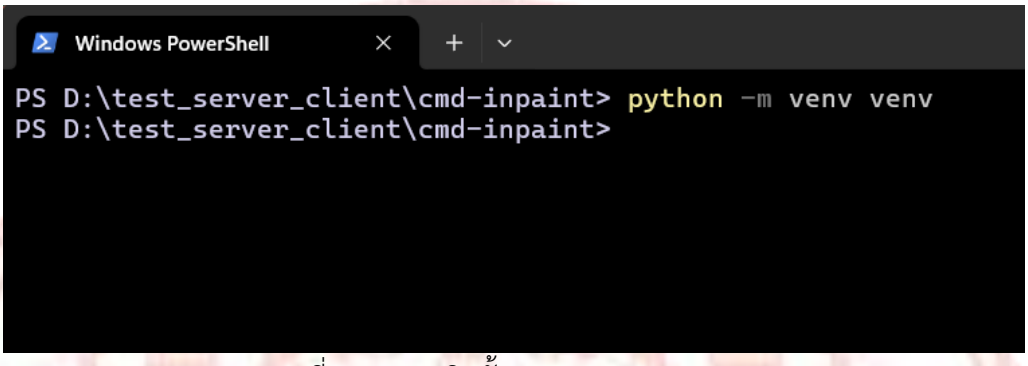
เปิด Command prompt หรือ Windows PowerShell แล้วไปยัง path ของโฟลเดอร์ api แล้วพิมพ์คำสั่ง “npm install” เพื่อติดตั้ง Packages ที่เกี่ยวข้อง



ภาพที่ ข-12 การติดตั้ง Package ของ API

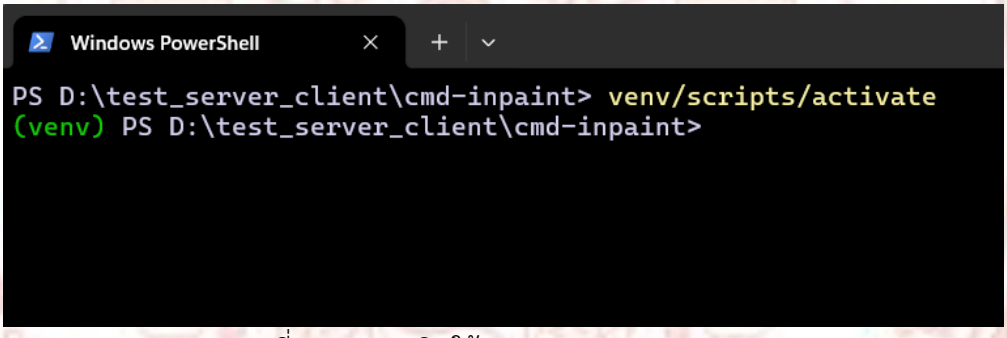
5. การติดตั้ง Package ของ Client Node

5.1 เปิด Command prompt หรือ Windows PowerShell แล้วไปยัง path ของโฟลเดอร์ source code แล้วพิมพ์คำสั่ง “python -m venv venv” เพื่อสร้าง virtual environment



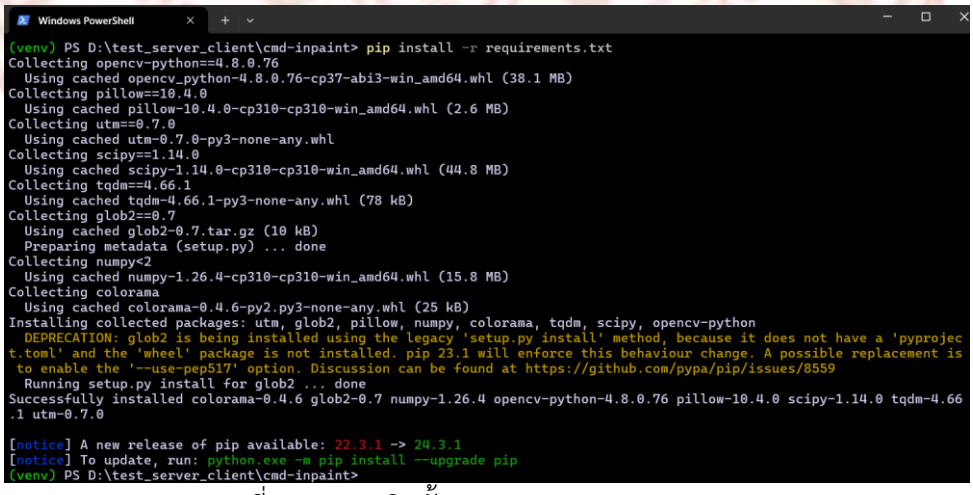
ภาพที่ ข-13 การติดตั้ง virtual environment

5.2 พิมพ์คำสั่ง “venv/scripts/activate” เพื่อเปิดใช้ virtual environment



ภาพที่ ข-14 การเปิดใช้งาน virtual environment

5.3 ติดตั้ง Packages ที่จำเป็น โดยใช้คำสั่ง pip install -r requirements.txt



ภาพที่ ข-15 การติดตั้ง Package ของ Client Node

5.4 กรณีที่ติดตั้ง OpenCV ไม่ได้ ซึ่งปัญหานี้ส่วนใหญ่จะเกิดขึ้นกับ Windows Server สามารถแก้ไขด้วยการติดตั้ง media feature pack จากเว็บไซต์ https://www.microsoft.com/en-us/software-download/mediafeaturepack หลังจากติดตั้งเสร็จให้กลับไปทำการติดตั้ง Packages ของ Client Node อีกครั้งในข้อที่ 5.3

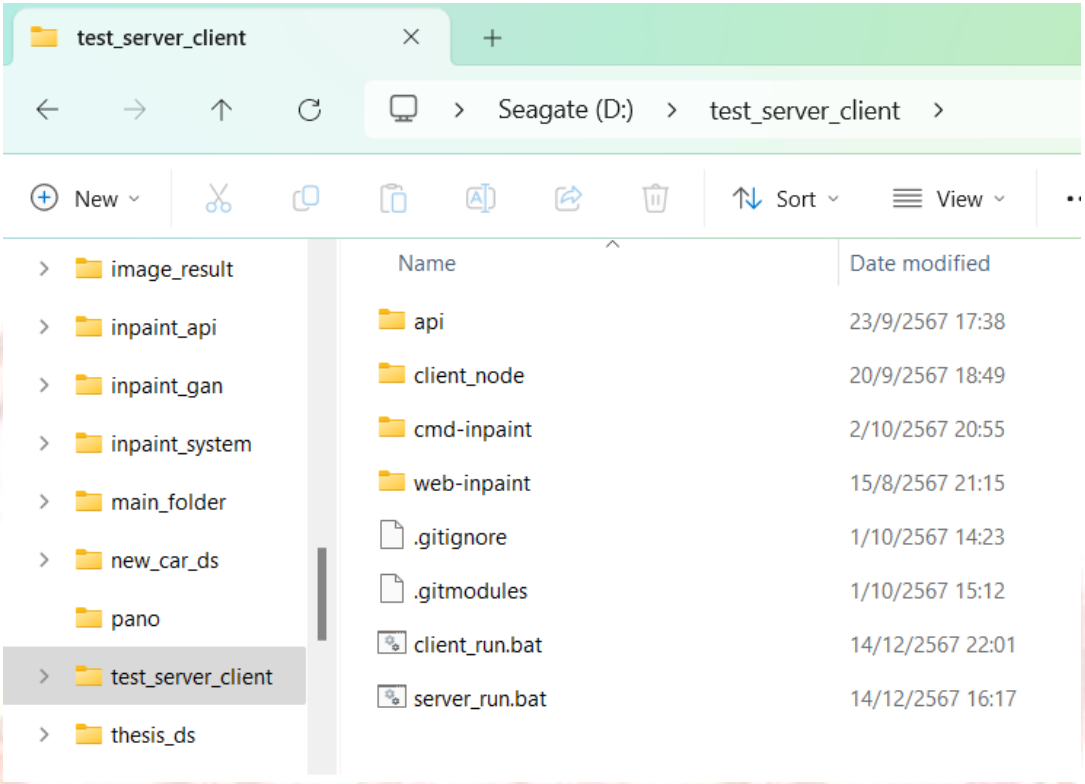
ภาคผนวก ค

คู่มือการใช้งาน

ขั้นตอนการใช้งานระบบ

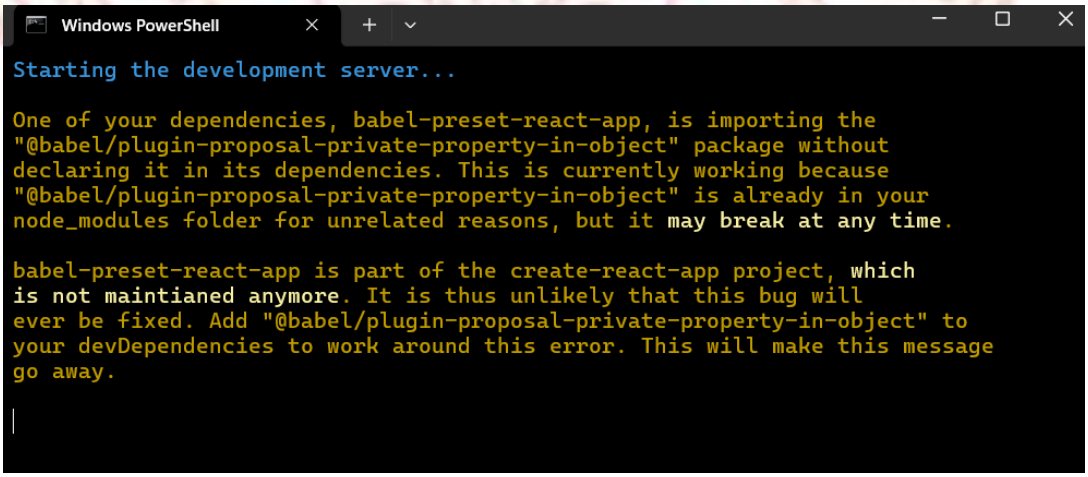
1. การเริ่มทำงานของระบบ

1.1 เปิดโฟลเดอร์ที่เก็บ source code



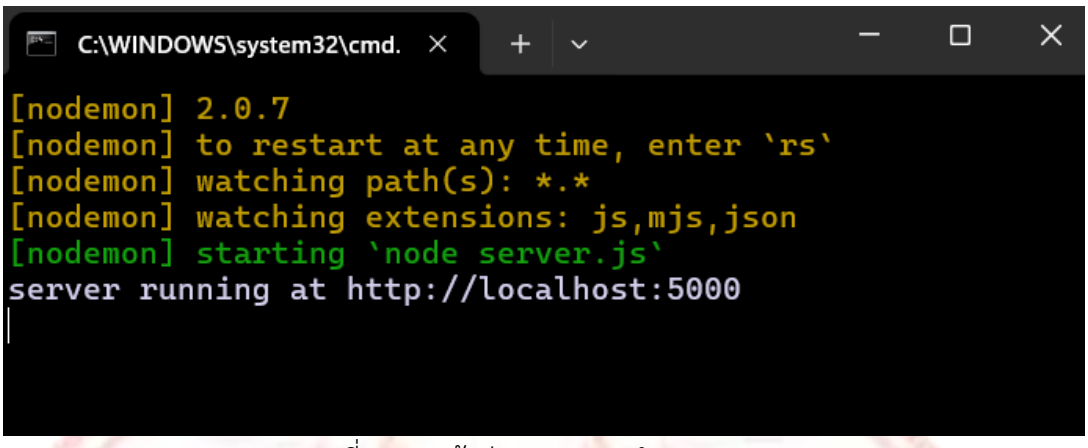
ภาพที่ ค-1 โฟลเดอร์ที่เก็บ source code

1.2 เปิดไฟล์ server_run.bat เพื่อเริ่มการทำงานในส่วนของ Web Client และส่วนของ API หลังจากเปิดไฟล์แล้วจะแสดง command prompt ขึ้นมา 2 หน้าต่าง



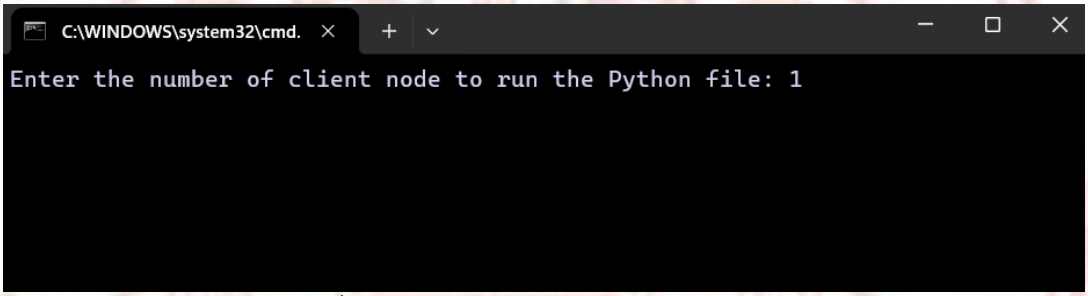
ภาพที่ ค-2 หน้าต่างแสดงการทำงานของ Web Client

.

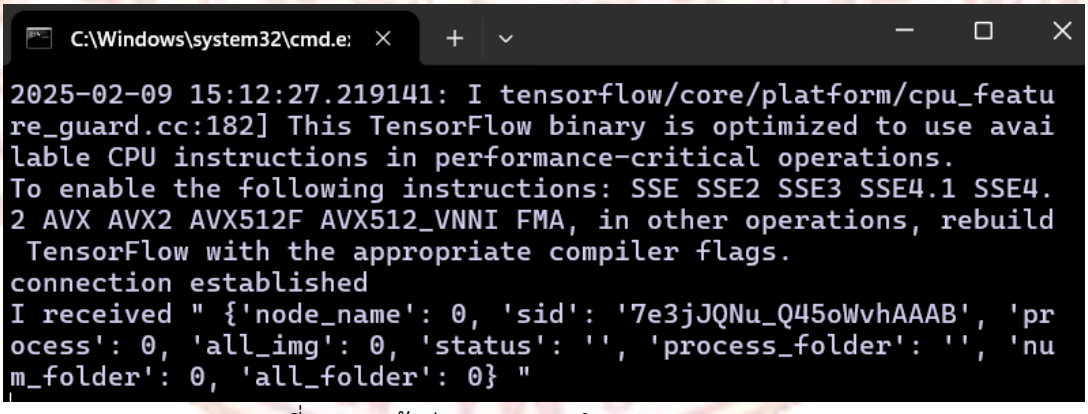


ภาพที่ ค-3 หน้าต่างแสดงการทำงานของ API

1.3 เปิดไฟล์ client_run.bat เพื่อเริ่มการทำงานในส่วนของ Client node แล้วทำการระบุจำนวน Client node ที่ต้องการ จากนั้นจะแสดง command prompt ตามจำนวนที่ระบุ



ภาพที่ ค-4 หน้าต่างกรอกจำนวน Client Node

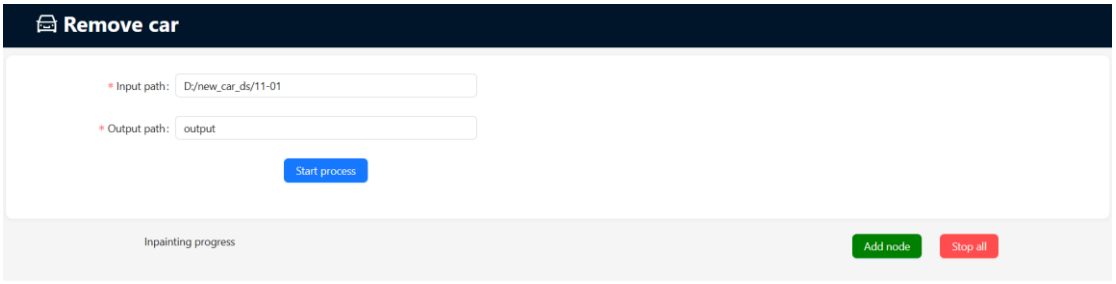


ภาพที่ ค-5 หน้าต่างแสดงการทำงานของ Client Node

2. ขั้นตอนการทำงานของระบบ

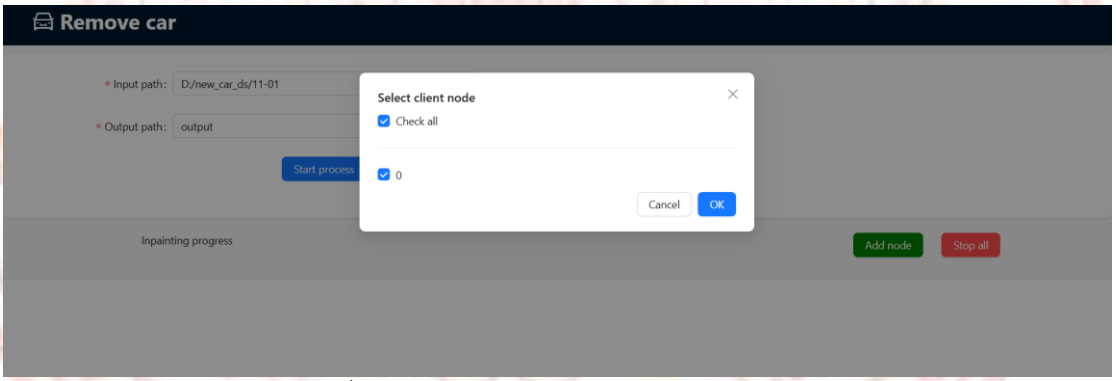
2.1 เปิดเว็บเบราว์เซอร์แล้วพิมพ์ URL เป็น http://localhost:3000/

2.2 กรอกข้อมูลโฟลเดอร์ที่ต้องการประมวลผลลบภาพรถสำรวจ พร้อมทั้งชื่อของโฟลเดอร์ที่จะใช้เก็บผลลัพธ์ ซึ่งโฟลเดอร์ที่เก็บผลลัพธ์นี้จะอยู่ภายในโฟลเดอร์ต้นทาง



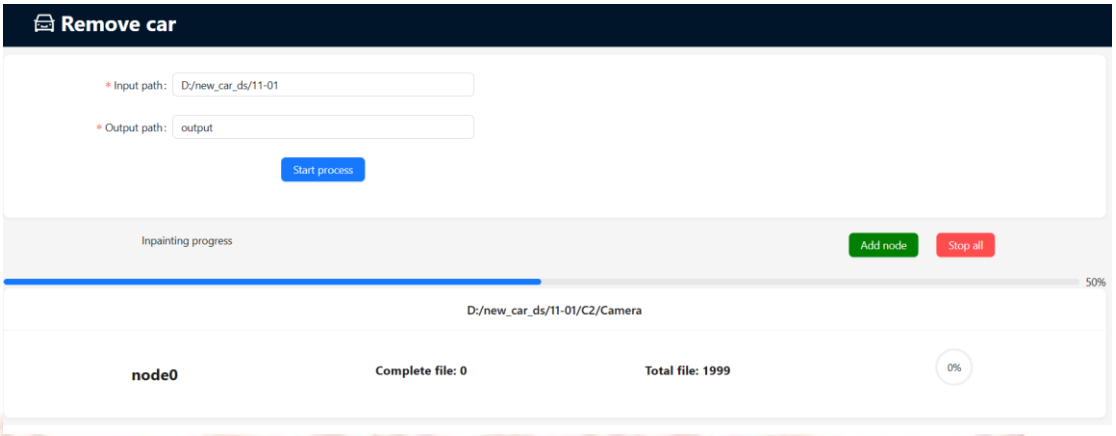
ภาพที่ ค-6 หน้าจอแสดงการกรอกรายละเอียด

2.3 เมื่อกรอกข้อมูลเสร็จเรียบร้อยแล้ว ผู้ใช้ต้องทำการจำนวน Client node ที่ต้องการใช้ในการประมวลผล ก่อนเริ่มการประมวลผล



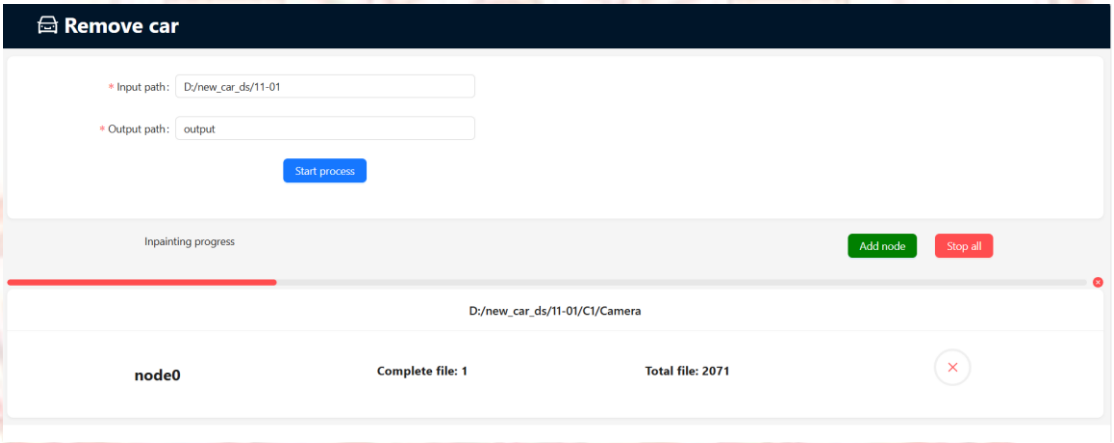
ภาพที่ ค-7 หน้าจอแสดงการเลือก Client node

2.4 ในระหว่างที่ Client node ประมวลผล ส่วนที่ติดต่อผู้ใช้งานจะแสดงจำนวนโฟลเดอร์ที่ผ่านการประมวลผล พร้อมกับจำนวนภาพที่ผ่านการประมวลผลของแต่ละ Client node



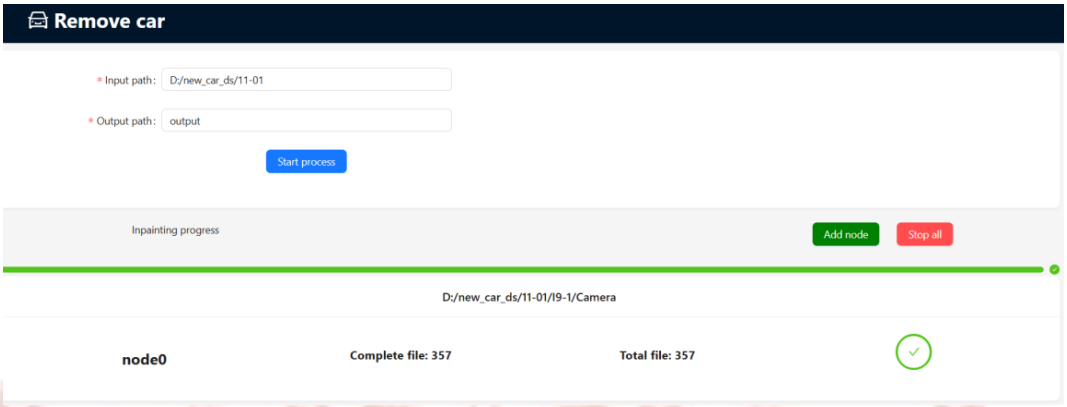
ภาพที่ ค-8 หน้าจอแสดงการประมวลผล

2.5 หากผู้ใช้งานมีการยกเลิกการทำงานระหว่างประมวลผล แถบสถานะการทำงานจะเปลี่ยนเป็นสีแดงเพื่อแสดงให้ผู้ใช้ทราบว่า Client node ได้หยุดการทำงานเรียบร้อยแล้ว



ภาพที่ ค-9 หน้าจอแสดงการยกเลิกการประมวลผล

2.6 เมื่อระบบทำการประมวลผลภาพสำเร็จทั้งหมดแล้ว แถบแสดงสถานะการทำงานจะเปลี่ยนเป็นสีเขียว



ภาพที่ ค-10 หน้าจอแสดงการประมวลผลสำเร็จ

ประวัติผู้เขียน

ชื่อ	อาซีซ่าร์ ลอดิง
ชื่อวิทยานิพนธ์	การลบภาพวัตถุสำหรับระบบจัดทำแผนที่ชนิดเคลื่อนที่โดยใช้โครงข่ายเจเนอเรทีฟแอดเวอเซอเรียล
สาขาวิชา	วิทยาการคอมพิวเตอร์
ประวัติ	จบการศึกษาปริญญาตรีสาขาวิชาวิทยาการคอมพิวเตอร์จากมหาวิทยาลัยเทคโนโลยีพระจอเกล้าพระนครเหนือ ปีการศึกษา 2564