



การพัฒนาวิธีการลบฝนออกจากภาพโดยใช้การเรียนรู้เชิงลึก



โดย  
นายพัสกร เอกผาชัยสวัสดิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2 ระดับปริญญาโทมหาบัณฑิต

ภาควิชาวิศวกรรมไฟฟ้า

มหาวิทยาลัยศิลปากร

ปีการศึกษา 2565

ลิขสิทธิ์ของมหาวิทยาลัยศิลปากร

การพัฒนาวิธีการลบฝนออกจากภาพโดยใช้การเรียนรู้เชิงลึก



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2 ระดับปริญญาโทมหาบัณฑิต

ภาควิชาวิศวกรรมไฟฟ้า

มหาวิทยาลัยศิลปากร

ปีการศึกษา 2565

ลิขสิทธิ์ของมหาวิทยาลัยศิลปากร

DEVELOPMENT OF IMAGE DERAINING METHOD USING DEEP LEARNING



By  
MR. Patsakorn AKEPHACHAISAWAT

A Thesis Submitted in Partial Fulfillment of the Requirements  
for Master of Engineering (ELECTRICAL AND COMPUTER ENGINEERING)

Department of ELECTRICAL ENGINEERING

Silpakorn University

Academic Year 2022

Copyright of Silpakorn University

หัวข้อ การพัฒนาวิธีการลบฝนออกจากภาพโดยใช้การเรียนรู้เชิงลึก  
โดย นายพัศกร เอกผาชัยสวัสดิ์  
สาขาวิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผนก ก แบบ ก 2 ระดับปริญญา  
มหาบัณฑิต  
อาจารย์ที่ปรึกษาหลัก อาจารย์ ดร. โสภณ ผู้มีจรรยา

---

คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากร ได้รับพิจารณาอนุมัติ  
ให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต



..... คณบดีคณะวิศวกรรมศาสตร์และ  
เทคโนโลยีอุตสาหกรรม  
(ผู้ช่วยศาสตราจารย์ ดร. อรุณศรี ลีจิระจำเนียร)

พิจารณาเห็นชอบโดย  
..... ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร. ระพีพันธ์ แก้วอ่อน)

..... อาจารย์ที่ปรึกษาหลัก  
(อาจารย์ ดร. โสภณ ผู้มีจรรยา)

..... ผู้ทรงคุณวุฒิภายใน  
(อาจารย์ ดร. ฆมร ศิลปพันธ์)

..... ผู้ทรงคุณวุฒิภายนอก  
(ผู้ช่วยศาสตราจารย์ ดร. วีรพล จิรจรีต)

630920053 : วิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2 ระดับปริญญามหาบัณฑิต

คำสำคัญ : โครงข่ายประสาทเทียมแบบสังวัตนาการ (CNN), Fusion model, Transfer learning, โครงข่ายเข้ารหัสอัตโนมัติ

นาย พัสกร เอกผาชัยสวัสดิ์: การพัฒนาวิธีการลบฝนออกจากภาพโดยใช้การเรียนรู้เชิงลึก  
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก : อาจารย์ ดร. โสภณ ผู้มีจรรยา

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการลบฝนออกจากภาพ แบ่งออกเป็นฝนเบา และฝนหนัก ซึ่งมีวิธีการลบฝนที่แตกต่างกัน เนื่องจากมีการแบ่งวิธีการลบฝนเป็นสองอย่าง จึงจำเป็นต้องมีการแยกฝนก่อนที่จะนำไปสู่ขั้นตอนการลบฝน โดยการจำแนกฝนนั้นจะใช้การทำ CNN ร่วมกับเทคนิคที่ชื่อว่า Fusion model เพื่อเพิ่มประสิทธิภาพของการจำแนก ซึ่งเป็นการฝึกสอนโมเดลที่ต้องใช้สองโมเดลในการรวมค่า Weight เพื่อเพิ่มประสิทธิภาพของการจำแนกฝน ต่อมาเป็นส่วนของการแยกฝนเบา และฝนหนัก ในฝนเบาจะใช้วิธีการที่ชื่อว่า SATLNet เป็นวิธีการลบฝนเบาที่รวมเทคนิคต่างๆที่ช่วยให้สามารถลบฝนได้ โดยวิธีการหลักแล้วจะใช้ Autoencoder ซึ่งเป็นวิธีที่ใช้ฝึกสอนภาพต้นทางให้เหมือนกับภาพปลายทาง โดยส่วนใหญ่แล้ววิธีการนี้ได้รับความนิยมในเรื่องของการกำจัดนอยส์ ในวิทยานิพนธ์ฉบับนี้จึงได้นำมาปรับใช้กับการลบฝน ซึ่งมองว่าฝนเป็นนอยส์ในอีกรูปแบบหนึ่ง โดยวัตถุประสงค์ของการลบฝนนั้น นอกจากจะช่วยเรื่องของความสวยงามแล้ว ยังสามารถช่วยในเรื่องของงานในด้านการมองเห็นของภาพ เช่น การตรวจจับวัตถุในขณะที่ฝนตก ซึ่งการลบฝนก่อนนั้นสามารถช่วยตรวจจับวัตถุได้ดีขึ้น ในส่วนของฐานข้อมูลภาพจะใช้สามฐานข้อมูลคือ Rain100L, Rain100H และ Rain800 ที่นำมาฝึกสอน และทดสอบของภาพฝนเบา และฝนหนัก ในส่วนของการทดสอบฝนจริง

630920053 : Major (ELECTRICAL AND COMPUTER ENGINEERING)

Keyword : Convolutional Neural Network (CNN), Fusion model, Transfer learning, Autoencoder

MR. Patsakorn AKEPHACHAISAWAT : Development of Image Deraining Method Using Deep Learning Thesis advisor : SOPON PHUMEECHANYA, Ph.D.

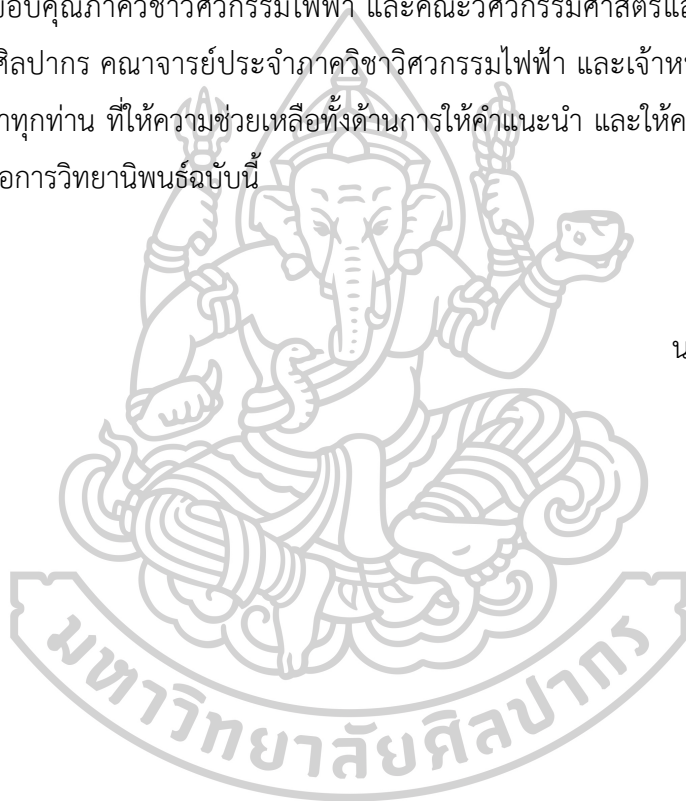
This thesis presents a method to remove rain from the image. It is divided into light rain and heavy rain, which have different methods of removing rain. Because there are two ways to remove the rain. Precipitation separation is thus required before leading to the precipitation removal step. In rain classification, CNN is combined with a technique called Fusion model to increase the efficiency of the classification. This is a training model that requires two models to combine Weights to increase the efficiency of rain classification. Next comes the division of light rain and heavy rain. In light rain, a method called SATLNet is used, a light rain removal method that combines various techniques that allow rain removal. The main method is to use Autoencoder, which is a method used to train the source image to be the same as the destination image. For the most part, this method is popular for eliminating noise. In this thesis, it has been applied to remove rain. which sees rain as noise in another form For the purpose of removing the rain In addition to helping with beauty Can also help in the field of vision, such as detecting objects in the rain. Removing the precipitation can help detect objects better. As for the image database, three databases, Rain100L, Rain100H and Rain800 are used for training. and test of images of light rain and heavy rain

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้รับปรึกษาจากอาจารย์ ดร.โสภณ ผู้มีจรรยา อาจารย์ที่ปรึกษา วิทยานิพนธ์ และคณะอาจารย์กรรมการสอบวิทยานิพนธ์ทุกท่าน ได้แก่ผู้ช่วยศาสตราจารย์ดร.ระพีพันธ์ แก้วอ่อน ประธานสอบวิทยานิพนธ์อาจารย์ ดร.ภมร ศิลาพันธ์กรรมการผู้ทรงคุณวุฒิภายใน และผู้ช่วย ศาสตราจารย์ดร.วีรพล จิรจรีต กรรมการผู้ทรงคุณวุฒิภายนอก ที่ให้คำปรึกษา ปรับปรุง และแนวทางการแก้ไขให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์ รวมถึงการนำเสนอผลงาน และกระบวนการวิจัย

ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้า และคณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากร คณาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้า และเจ้าหน้าที่และบุคลากรภาควิชา วิศวกรรมไฟฟ้าทุกท่าน ที่ให้ความช่วยเหลือทั้งด้านการให้คำแนะนำ และให้ความช่วยเหลืออำนวยความสะดวกต่างๆต่อการวิทยานิพนธ์ฉบับนี้

นาย พัสกร เอกผาชัยสวัสดิ์



## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	3
1.3 ขอบเขตของงานวิจัย.....	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution neural network).....	4
2.1.1 ชั้นคอนโวลูชัน (Convolution layer).....	4
2.2.2 ชั้นพูลลิ่ง (Pooling layer).....	6
2.2.3 ชั้นเชื่อมต่อโยงสมบูรณ์ (Fully Connected Layer).....	6
2.2 โครข่ายการเข้ารหัสอัตโนมัติ (Autoencoder).....	7
2.2.1 การเข้ารหัส (Encoder).....	7
2.2.2 ชั้นแฝง (Latent layer).....	7
2.2.3 การถอดรหัส (Decoder).....	8
2.3 Transfer learning.....	8



2.3.1 การใช้โมเดลที่มีการฝึกสอนมาก่อนหน้า (Pre-trained model).....	8
2.3.2 การทำ Fine tuning.....	9
2.4 Median stack .....	10
2.5 การเพิ่มจำนวนข้อมูลภาพ (Image augmentation).....	11
2.5.1 การหมุนภาพ (Rotation).....	11
2.5.2 การเปลี่ยนขนาดภาพ (Resizing) .....	12
2.5.3 การเพิ่มสิ่งรบกวน (Noise).....	12
2.6 YOLOV5 .....	12
2.7 Peak signal-to-noise ratio.....	14
2.8 Structural similarity index measure .....	14
2.9 Confusion matrix.....	15
2.10 การตรวจจับภาพขอบ (Edge detection).....	16
2.10.1 search-based.....	16
2.10.2 Zero-crossing .....	18
2.11 งานวิจัยที่เกี่ยวข้อง .....	18
บทที่ 3 วิธีดำเนินการวิจัย.....	30
3.1 ภาพรวมของวิธีการ.....	30
3.2 การตรวจจับวัตถุโดยใช้ Fusion model ร่วมกับ CNN.....	31
3.3 SATLNet สำหรับกรณีลบฝนเบา .....	33
3.3.1 Non Block-based Autoencoder.....	35
3.3.2 Block-based Autoencoder.....	35
3.3.3 Improve Block.....	36
3.3.4 Transfer Learning Autoencoder.....	39
3.4 Autoencoder สำหรับกรณีลบฝนหนัก .....	40

บทที่ 4 ผลการทดลองของงานวิจัย.....	41
4.1 ชุดข้อมูลภาพที่ใช้ในการทดลอง .....	41
4.1.1 ชุดข้อมูลที่ใช้สำหรับการจำแนกฝนเบา และฝนหนัก.....	41
4.1.2 ชุดข้อมูลไว้สำหรับการลบฝนออกจากภาพ .....	44
4.2 วัดประสิทธิภาพการจำแนกฝนเบา และฝนหนัก .....	49
4.3 วัดประสิทธิภาพของการลบฝน.....	58
4.3.1 ทดสอบประสิทธิภาพการลบฝนเบาโดยใช้ SATLNet.....	58
4.3.2 ทดสอบประสิทธิภาพการลบฝนหนักโดยใช้ Autoencoder.....	64
4.3.3 ทดสอบประสิทธิภาพการลบฝนทั้งระบบ .....	67
4.3.4 ทดสอบประสิทธิภาพการลบฝนจริง.....	67
4.4 วัดประสิทธิภาพของการตรวจจับวัตถุหลังจากลบฝนโดยใช้ YOLO .....	72
4.4.1 การวัดประสิทธิภาพโดยใช้ภาพฝนสังเคราะห์ .....	72
4.4.2 การวัดประสิทธิภาพโดยใช้ฝนจริง.....	74
บทที่ 5 สรุปและข้อเสนอแนะ .....	76
5.1 สรุปผลการวิจัย.....	76
5.2 ปัญหาและข้อเสนอแนะ .....	77
5.3 แนวทางการพัฒนาต่อยอด.....	78
รายการอ้างอิง .....	79
ประวัติผู้เขียน .....	91

## สารบัญตาราง

	หน้า
ตารางที่ 4.1 เปรียบเทียบค่าความแม่นยำของวิธีการจำแนกฝนในรูปแบบต่างๆ .....	57
ตารางที่ 4.2 เปรียบเทียบประสิทธิภาพของภาพฝนต้นฉบับ 1 รูป และพื้นฐานข้อมูล 150 รูป.....	59
ตารางที่ 4.3 เปรียบเทียบประสิทธิภาพของการลบฝนในทุกขั้นตอนของการลบฝน .....	63
ตารางที่ 4.4 เปรียบเทียบประสิทธิภาพของการลบฝนในงานวิจัยนี้กับ GCANet .....	64
ตารางที่ 4.5 เปรียบเทียบประสิทธิภาพของการลบฝนหนักในครั้งที่ 1 และครั้งที่ 2 .....	66
ตารางที่ 4.6 เปรียบเทียบประสิทธิภาพของการลบฝนในงานวิจัยนี้กับ GCANet .....	66
ตารางที่ 4.7 เปรียบเทียบประสิทธิภาพของการลบฝนหนัก และฝนเบาทั้งระบบ.....	67



## สารบัญภาพ

	หน้า
รูปที่ 1.1 ภาพการตรวจจับวัตถุโดยใช้ Yolov5.....	2
รูปที่ 2.1 การหา Feature map .....	5
รูปที่ 2.2 การหา pooling จากภาพ.....	6
รูปที่ 2.3 โครงข่ายคอนโวลูชัน.....	7
รูปที่ 2.4 โครงข่ายการเข้ารหัสอัตโนมัติ.....	8
รูปที่ 2.5 การทำ Transfer learning ระหว่างสองโมเดล.....	9
รูปที่ 2.6 การทำ Fine tuning .....	10
รูปที่ 2.7 การทำ Median stack.....	11
รูปที่ 2.8 การเพิ่มจำนวนรูปภาพ.....	12
รูปที่ 2.9 โครงสร้างของ Yolov5.....	13
รูปที่ 2.10 ตัวอย่าง Confusion matrix .....	15
รูปที่ 2.11 โครงสร้างสถาปัตยกรรม Semi-Supervised Image Deraining.....	19
รูปที่ 2.12 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น .....	19
รูปที่ 2.13 โครงสร้างสถาปัตยกรรม PreNet.....	20
รูปที่ 2.14 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น .....	21
รูปที่ 2.15 โครงสร้างสถาปัตยกรรม MSPFN.....	22
รูปที่ 2.16 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น .....	22
รูปที่ 2.17 โครงสร้างสถาปัตยกรรม SPANet.....	23
รูปที่ 2.18 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น .....	23
รูปที่ 2.19 โครงสร้างสถาปัตยกรรม DerainNet .....	24
รูปที่ 2.20 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น .....	24

รูปที่ 2.21 โครงสร้างสถาปัตยกรรม RCDNet .....	25
รูปที่ 2.22 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น .....	25
รูปที่ 2.23 โครงสร้างสถาปัตยกรรม SPDNet.....	26
รูปที่ 2.24 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น .....	26
รูปที่ 2.25 โครงสร้างสถาปัตยกรรม SAPNet.....	27
รูปที่ 2.26 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น .....	28
รูปที่ 2.27 โครงสร้างสถาปัตยกรรม GCANet.....	28
รูปที่ 2.28 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น .....	29
รูปที่ 3.1 ภาพรวมของขั้นตอนการลบฝน .....	30
รูปที่ 3.2 โครงสร้างของการจำแนกฝน .....	32
รูปที่ 3.3 โครงสร้างของ SATLNet.....	35
รูปที่ 3.4 โครงสร้าง Autoencoder ที่ใช้สำหรับลบฝน .....	35
รูปที่ 3.5 การแบ่งส่วนภาพสำหรับการฝึกสอน .....	36
รูปที่ 3.6 โครงสร้างของ Improve block.....	38
รูปที่ 3.7 ส่วนสุดท้ายของ Improve block # 2.....	39
รูปที่ 3.8 ขั้นตอนการลบฝนหนัก.....	40
รูปที่ 4.1 ตัวอย่างภาพฝนไว้สำหรับฝึกสอนในการลบฝนเบา .....	41
รูปที่ 4.2 ตัวอย่างภาพฝนไว้สำหรับการทดสอบในการลบฝนเบา.....	42
รูปที่ 4.3 ตัวอย่างภาพฝนไว้สำหรับฝึกสอนในการลบฝนหนัก.....	43
รูปที่ 4.4 ตัวอย่างภาพฝนไว้สำหรับการทดสอบในการลบฝนหนัก .....	43
รูปที่ 4.5 ตัวอย่างเปรียบเทียบภาพฝนกับ Ground truth ไว้สำหรับการทดสอบฝนเบาที่เพิ่มเข้ามา	44
รูปที่ 4.6 ตัวอย่างเปรียบเทียบภาพฝนกับ Ground truth ไว้สำหรับการฝึกสอนในฐานข้อมูล Rain100L .....	45

รูปที่ 4.7 ตัวอย่างเปรียบเทียบภาพฝนกับ Ground truth ไว้สำหรับการทดสอบในฐานข้อมูล Rain100L .....	46
รูปที่ 4.8 ตัวอย่างเปรียบเทียบภาพฝนกับ Ground truth ไว้สำหรับการฝึกสอนในฐานข้อมูล Rain100H.....	47
รูปที่ 4.9 ตัวอย่างเปรียบเทียบภาพฝนกับ Ground truth ไว้สำหรับการทดสอบในฐานข้อมูล Rain100H.....	48
รูปที่ 4.10 วัดประสิทธิภาพของ CNN model ของภาพฝนต้นฉบับ .....	49
รูปที่ 4.11 ตัวอย่างภาพที่มีการจำแนกผิดในฝนหนัก.....	50
รูปที่ 4.12 ตัวอย่างภาพขอบของแต่ละวิธีการ .....	51
รูปที่ 4.13 วัดประสิทธิภาพของ Fusion model โดยใช้ภาพขอบแบบ Sobel.....	51
รูปที่ 4.14 ตัวอย่างภาพที่มีการจำแนกผิดในฝนเบา.....	52
รูปที่ 4.15 ตัวอย่างภาพที่มีการจำแนกผิดในฝนหนัก.....	52
รูปที่ 4.16 วัดประสิทธิภาพของ Fusion model โดยใช้ภาพขอบแบบ Laplacian.....	53
รูปที่ 4.17 ตัวอย่างภาพที่มีการจำแนกผิดในฝนหนักโดยใช้ภาพขอบแบบ Laplacian.....	54
รูปที่ 4.18 วัดประสิทธิภาพของ Fusion model โดยใช้ภาพขอบแบบ Canny.....	55
รูปที่ 4.19 ตัวอย่างภาพที่มีการจำแนกผิดในฝนเบาโดยใช้ภาพขอบ Canny.....	56
รูปที่ 4.20 ตัวอย่างภาพที่มีการจำแนกผิดในฝนหนักโดยใช้ภาพขอบแบบ Canny.....	56
รูปที่ 4.21 ภาพตัวอย่างสำหรับทดสอบประสิทธิภาพการลบฝน .....	58
รูปที่ 4.22 เปรียบเทียบการลบระหว่าง ภาพฝน ภาพดั้งเดิม ภาพลบฝนโดยใช้วิธีการ Non-block-based autoencoder .....	59
รูปที่ 4.23 เปรียบเทียบการลบระหว่าง (ก) ภาพฝน (ข) ภาพดั้งเดิม (ค) ภาพลบฝนโดยใช้วิธีการ Block-based autoencoder.....	60
รูปที่ 4.24 เปรียบเทียบการลบระหว่าง ภาพฝน ภาพดั้งเดิม ภาพลบฝนโดยใช้วิธีการ Improve block.....	61
รูปที่ 4.25 เปรียบเทียบการลบระหว่าง (ก) ภาพฝน (ข) ภาพดั้งเดิม (ค) ภาพลบฝนโดยใช้วิธีการ Transfer learning.....	62

รูปที่ 4.26 เปรียบเทียบระหว่าง ภาพฝน ภาพดั้งเดิม ภาพการลบฝนครั้งที่ 1 และ การลบฝนครั้งที่ 2 .....	65
รูปที่ 4.27 เปรียบเทียบ ภาพฝนจริงดั้งเดิม ภาพลบฝนโดยใช้ GCANet และภาพที่ผ่านการลบฝนโดยใช้ SATLNet.....	68
รูปที่ 4.28 เปรียบเทียบ ภาพฝนจริงดั้งเดิม ภาพลบฝนโดยใช้ GCANet และภาพที่ผ่านการลบฝนโดยใช้ SATLNet.....	68
รูปที่ 4.29 เปรียบเทียบ ภาพฝนจริงดั้งเดิม ภาพลบฝนโดยใช้ GCANet และภาพที่ผ่านการลบฝนโดยใช้ SATLNet.....	69
รูปที่ 4.30 เปรียบเทียบภาพฝนจริงดั้งเดิมภาพการลบฝนโดยใช้ GCANet และภาพที่ผ่านการลบฝนโดยใช้โมเดลฝนหนัก.....	70
รูปที่ 4.31 เปรียบเทียบภาพฝนจริงดั้งเดิม ภาพการลบฝนโดยใช้ GCANet และภาพที่ผ่านการลบฝนโดยใช้โมเดลฝนหนัก.....	71
รูปที่ 4.32 เปรียบเทียบภาพฝนจริงดั้งเดิม ภาพการลบฝนโดยใช้ GCANet และภาพที่ผ่านการลบฝนโดยใช้โมเดลฝนหนัก.....	71
รูปที่ 4.33 ตัวอย่างการตรวจจับวัตถุจากภาพ Ground truthโดยใช้ Yolov5 .....	72
รูปที่ 4.34 ตัวอย่างภาพที่ไม่สามารถตรวจจับวัตถุได้ในภาพฝน และภาพลบฝน .....	73
รูปที่ 4.35 ตัวอย่างภาพที่ตรวจจับวัตถุเฉพาะภาพที่ลบฝน.....	73
รูปที่ 4.36 ตัวอย่างภาพที่ตรวจจับวัตถุเฉพาะภาพที่ลบฝน.....	73
รูปที่ 4.37 ภาพฝนจริงที่นำตรวจจับวัตถุในภาพ.....	74
รูปที่ 4.38 ภาพการลบฝนจริงที่นำไปตรวจจับวัตถุในภาพ.....	75
รูปที่ 5.1 ตัวอย่างการเปรียบเทียบระหว่างภาพต้นฉบับ และภาพที่ลบฝนออกแล้วแต่ยังหลงเหลือฝนอยู่.....	77
รูปที่ 5.2 แสดงตัวอย่างการเปรียบเทียบระหว่างภาพต้นฉบับ ภาพฝน และภาพฝนที่ถูกลบออกที่มีรายละเอียดพื้นหลังที่ลดลงไป .....	78

## บทที่ 1

### บทนำ

#### 1.1 ความสำคัญและที่มาของปัญหา

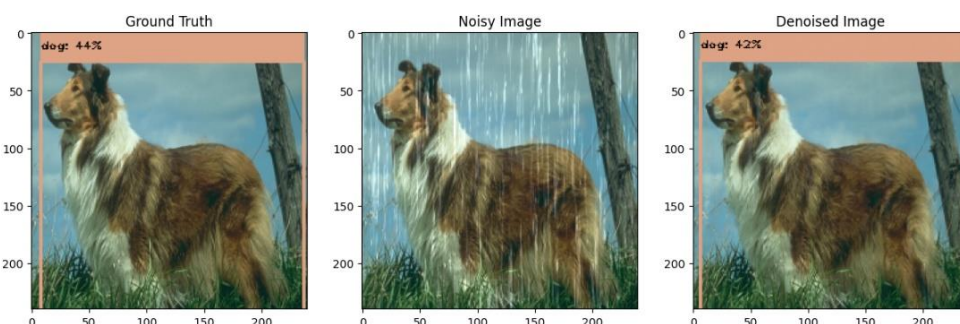
ในปัจจุบันการถ่ายภาพเป็นที่นิยมกันอย่างมาก แม้ว่าการถ่ายภาพสามารถทำได้ทุกที่ทุกเวลา แต่ในบางสถานการณ์มีสิ่งรบกวนที่บดบังภาพถ่าย ทำให้ภาพเกิด ความไม่สมบูรณ์ เช่น การถ่ายภาพขณะฝนตก จะได้ภาพที่มีสายฝนติดมากับภาพถ่าย ซึ่งเป็นสิ่งที่ผู้ถ่ายไม่ต้องการ แม้ว่าในสถานการณ์นี้สามารถหลีกเลี่ยงได้โดยการรอฝนหยุดตก แต่ต้องแลกมากับเวลาที่เสียไป ด้วยเหตุนี้จึงมีการนำเทคโนโลยีมาใช้กับภาพถ่ายเพื่อกำจัดสิ่งรบกวนที่ทำให้ภาพเกิด ความไม่สมบูรณ์ นอกจากนี้ในงานคอมพิวเตอร์วิทัศน์บางงานจำเป็นต้องใช้ภาพที่มีความชัดเจน มีสิ่งรบกวนที่น้อย เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด เช่น ในงานการตรวจจับวัตถุในภาพ การแยกพื้นที่ ภายในภาพ จึงทำให้การลบฝนมีความสำคัญอย่างมากกับงานที่ต้องตรวจจับวัตถุ หรือแยกพื้นที่ภายในได้ สภาพแวดล้อมที่ฝนตก ในงานวิจัยนี้จึงมุ่งเน้นไปที่การแก้ปัญหาสิ่งรบกวนจากภาพที่เกิดจากฝนเท่านั้น เนื่องจากเป็นสิ่งที่พบได้บ่อย อีกทั้งยังเป็นสิ่งรบกวนที่เกิดปัญหากับภาพอย่างมาก เม็ดฝนที่บดบังรูปภาพนั้นมีความถี่ ความหนา และกระจายอยู่ทั้งภาพ จึงสามารถมองเป็นสิ่งรบกวนชนิดหนึ่งได้

การแก้ปัญหาในปัจจุบันมีโปรแกรมมากมายที่ใช้ในการแก้ไขหรือตกแต่งภาพ ภายในโปรแกรมมีเครื่องมือต่างๆให้เลือกใช้งานตามความเหมาะสม แม้กระทั่งการลบฝนออกจากภาพถ่ายก็ยังสามารถทำได้ โดยใช้เทคนิคการตกแต่งภาพ แต่ต้องใช้ความชำนาญและประสบการณ์ของผู้แก้ไขภาพ เพื่อให้ได้ภาพออกมามีความสมจริงและสมบูรณ์มากที่สุด ซึ่งต้องเสียเวลาเป็นอย่างมากหากต้องลบฝนในงานที่เป็นภาพเคลื่อนไหวที่ต้องทำการลบทีละเฟรม จึงจะได้วิดีโอที่ถูกลบฝนออกจนหมด ด้วยเหตุนี้จึงมีการนำวิธีการ ประมวลผลภาพและการเรียนรู้เชิงลึกเข้ามาช่วยในการลบฝนออกจากภาพถ่าย

การลบฝนในรูปภาพเป็นขั้นตอนเบื้องต้นที่สำคัญในคอมพิวเตอร์วิทัศน์ที่เกี่ยวข้องกับการประมวลผลของภาพที่ต้องใช้ความแม่นยำ โดยใช้การเรียนรู้เชิงลึกออกแบบวิธีการลบฝนที่ช่วยเพิ่มประสิทธิภาพในงานที่เกี่ยวข้องกับการประมวลผลภาพ เช่น การตรวจจับวัตถุ, การจำแนกวัตถุ [1,2] จะเห็นว่าในงานที่กล่าวมา หากมีสิ่งรบกวนมาบดบังภาพจะทำให้ประสิทธิภาพการตรวจจับวัตถุ หรือการจำแนกวัตถุต่ำลง เนื่องจากแบบจำลองการจำแนกวัตถุส่วนใหญ่ได้ถูกฝึกสอนภายใต้สภาวะปกติ



ไม่มีสิ่งรบกวนเข้ามาบดบัง ทำให้เมื่อนำแบบจำลองมาใช้งานในส่วนของภาพที่รับเข้ามานั้น การกำจัด  
 ฝนออกจึงช่วยเพิ่มค่าความแม่นยำของวัตถุได้ได้มากขึ้น



รูปที่ 1.1 ภาพการตรวจจับวัตถุโดยใช้ YOLOv5

จากที่กล่าวมาข้างต้นจะเห็นว่า การกำจัดฝนก่อนทำการจำแนกวัตถุมีความสำคัญหากในภาพ  
 นั้นๆมีสิ่งรบกวนที่เยอะเกินไป ในงานวิจัยนี้จึงสนใจกำจัดสิ่งรบกวนในรูปแบบที่เป็นฝน เนื่องจากเป็น  
 สิ่งที่พบได้บ่อย และมีขนาดใหญ่ ซึ่งส่งผลต่อการจำแนกวัตถุได้อย่างชัดเจนดังรูปที่ 1.1 หากไม่มีการ  
 ลบฝนออกจากภาพก่อนจะไม่สามารถตรวจจับวัตถุได้ แต่เมื่อทำการลบฝนออกก่อนแล้วจึงนำเข้าสู่  
 กระบวนการตรวจจับวัตถุ จะพบว่าสามารถตรวจจับได้ และมีค่าความถูกต้องที่ใกล้เคียงกับภาพ  
 ต้นฉบับ

ในงานวิจัยนี้จึงได้นำการเรียนรู้เชิงลึกมาใช้เพื่อลบฝนออกจากภาพ โดยกระบวนการลบฝน  
 ออกจากภาพที่งานวิจัยนี้ได้นำเสนอ มีการลบฝนอยู่สองแบบคือฝนเบา และฝนหนัก โดยก่อนเริ่ม  
 กระบวนการลบฝนนั้นได้มีการออกแบบวิธีการคัดแยกฝนเบา และฝนหนักก่อน เนื่องจากลักษณะของ  
 ฝนทั้งสองรูปแบบมีความต่างกันอย่างมาก การแยกฝนเพื่อนำไปฝึกสอนแบบจำลองจึงสามารถช่วย  
 เพิ่มประสิทธิภาพของการลบฝนได้ ซึ่งในส่วนของการลบฝน มีส่วนประกอบสำคัญสามส่วน  
 ได้แก่ autoencoder [3] , improve block และ transfer learning ที่สามารถช่วยในการลบฝนได้  
 [4] โดยในส่วนแรกใช้สำหรับลบฝน และแยกเม็ดฝนออกจากภาพในเบื้องต้น เม็ดฝนที่ถูกแยก  
 ออกมาจะนำไปใช้สำหรับการทำ image augmentation เพื่อเพิ่มจำนวนรูปภาพที่มีลักษณะคล้ายกับ  
 ภาพฝนที่ต้องการ ทำให้เพิ่มประสิทธิภาพการทำนายได้ดียิ่งขึ้น [5] จึงเป็นสาเหตุที่ในส่วนสุดท้าย  
 จำเป็นต้องใช้ transfer learning

## 1.2 วัตถุประสงค์ของงานวิจัย

- 1.2.1 เพื่อศึกษาสถาปัตยกรรมการเรียนรู้เชิงลึกใช้สำหรับการลบฝนออกจากภาพ
- 1.2.2 เพื่อศึกษาวิธีการลบฝนโดยใช้เทคนิคการประมวลผลภาพแบบต่างๆ
- 1.2.3 เพื่อออกแบบวิธีการจำแนกประเภทฝนโดยใช้การเรียนรู้เชิงลึก
- 1.2.4 เพื่อออกแบบและพัฒนาวิธีการลบฝนออกจากภาพโดยใช้การเรียนรู้เชิงลึก

## 1.3 ขอบเขตของงานวิจัย

- 1.3.1 พัฒนาการจำแนกฝนในภาพออกเป็นฝนหนัก และฝนเบา
- 1.3.2 พัฒนาการลบฝนออกจากภาพ โดยปริมาณฝนมีความหนาแน่นในระดับน้อยถึงสูง
- 1.3.3 ใช้ภาพจากฐานข้อมูลสาธารณะ ได้แก่ Rain100L, Rain100H[6] , Rain800[7] และมีการเลือกภาพจากฐานข้อมูลใน [8] จำนวน 50 รูป เพื่อใช้ในการทดสอบภาพฝนเบาเพิ่มเติม
- 1.3.4 ออกแบบเฟรมเวิร์คที่ใช้สำหรับการลบฝนออกจากภาพ

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

ได้วิธีการใหม่สำหรับการลบฝนออกจากภาพ และวิธีการจำแนกภาพฝนใหม่ ที่ช่วยให้การลบฝนได้ดีขึ้น

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีและงานวิจัยที่เกี่ยวข้องได้นำเสนอพื้นฐานที่เกี่ยวข้องกับการพัฒนาวิธีการลบฝนออกจากภาพประกอบด้วย การเรียนรู้เชิงลึก โครงข่ายประสาทเทียมแบบคอนโวลูชัน(Convolution neural network) โครงข่ายการเข้ารหัส(Autoencoder) และงานวิจัยที่เกี่ยวข้องกับการลบฝนออกจากภาพ

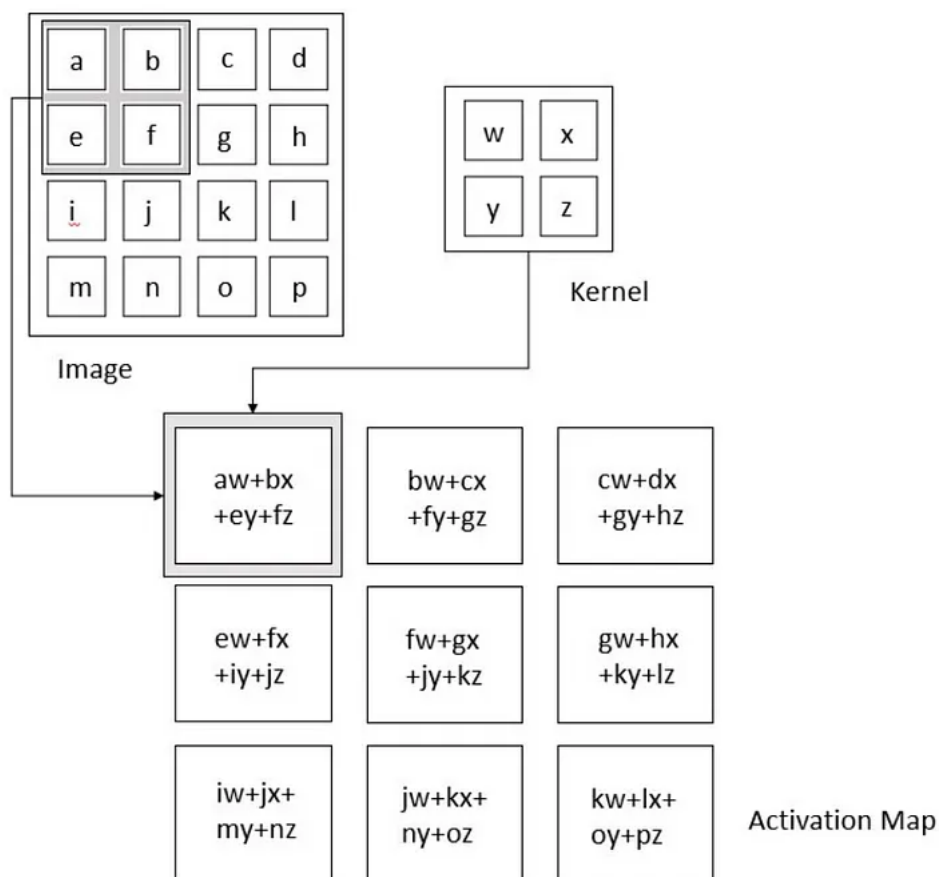
#### 2.1 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution neural network)

โครงข่ายประสาทเทียมแบบคอนโวลูชันเป็นโครงข่ายประสาทเทียมที่นิยมใช้กับรูปภาพเป็นข้อมูลนำเข้า มีแนวคิดมาจากการเรียนรู้เชิงลึกที่มีจำนวนชั้นที่ซ่อนอยู่เป็นจำนวนมาก แต่ในโครงข่ายประเภทนี้จะ เป็นชั้นของคอนโวลูชันแทน ซึ่งมีหน้าที่สกัดลักษณะเด่นของข้อมูลที่ได้รับเข้ามา เช่น เส้น รูปร่าง และสี ซึ่งเป็นการจำลองที่คล้ายกับการมองเห็นของมนุษย์ที่มองภาพเป็นส่วนๆจนครบทั้งภาพ เพื่อดูภาพรวมทั้งหมดของภาพว่าเป็นรูบอะไร โดยหลักการทำงานภายในชั้นของคอนโวลูชัน แบ่งออกเป็น 3 ส่วน ดังนี้

2.1.1 ชั้นคอนโวลูชัน (Convolution layer) เป็นชั้นที่แยกคุณสมบัติต่างๆของข้อมูลภาพที่ได้รับเข้ามา โดยจะมีตัวกรองที่ภายในประกอบไปด้วยค่าที่ไว้สำหรับสกัดคุณลักษณะเด่นของภาพ ตัวกรองจะทำการเลื่อนผ่านภาพที่ได้รับเข้ามาจนครบทุกพิกเซลจะได้ฟังก์ชันลักษณะของภาพออกมา เพื่อใช้สำหรับคอนโวลูชันในชั้นถัดไป โดยในเริ่มต้นของชั้นนี้จะทำการต่อทโพรดักระหว่างสองเมทริกซ์ เมทริกซ์แรกเป็นชุดของพารามิเตอร์ที่เรียนรู้ได้หรือเรียกอีกอย่างว่าเคอร์เนล ส่วนเมทริกซ์อีกตัวเป็นฟิลต์รับที่ให้เคอร์เนลได้เรียนรู้ข้อมูล โดยเคอร์เนลมีขนาดเล็กแต่มีความลึกมากกว่า ยกตัวอย่าง เช่น รูปภาพประกอบด้วยสามชั้น (RGB) ความสูงและความกว้างของเคอร์เนลจะมีขนาดเล็กกว่าเมทริกซ์ที่เป็นข้อมูลภาพ แต่ความลึกจะขยายไปถึงทั้งสามชั้นของภาพ เคอร์เนลจะเลื่อนไปตามความสูงและความกว้างทั้งหมดของรูปภาพ การเลื่อนของเคอร์เนลจะเรียกว่า Stride ซึ่งเป็นตัวบ่งบอกถึงระยะการเลื่อนของตัวเคอร์เนล หากมี Stride ที่มากจะได้คุณลักษณะของภาพที่น้อย แต่หากต้องการคุณลักษณะที่เท่ากับข้อมูลภาพต้นฉบับจำเป็นต้องใช้ Paddingคือการเติมข้อมูลไปบริเวณรอบๆของภาพ หากขนาดภาพนำเข้ามีขนาด  $W \times W \times D$  ซึ่ง  $W$  คือความกว้างของภาพ และ  $D$  คือ มิติของภาพ เราสามารถหาขนาดของภาพนำออกได้ดังนี้

$$W_{out} = \frac{W-F+2P}{S} + 1 \quad (2.1)$$

โดย F เป็นขนาดของเคอร์เนล P เป็นจำนวนของ Padding และ S เป็นขนาดของ Stride ซึ่งผลลัพธ์ที่ได้จะเป็นขนาดของ Feature map ที่สามารถคำนวณอย่างง่ายดังต่อไปนี้



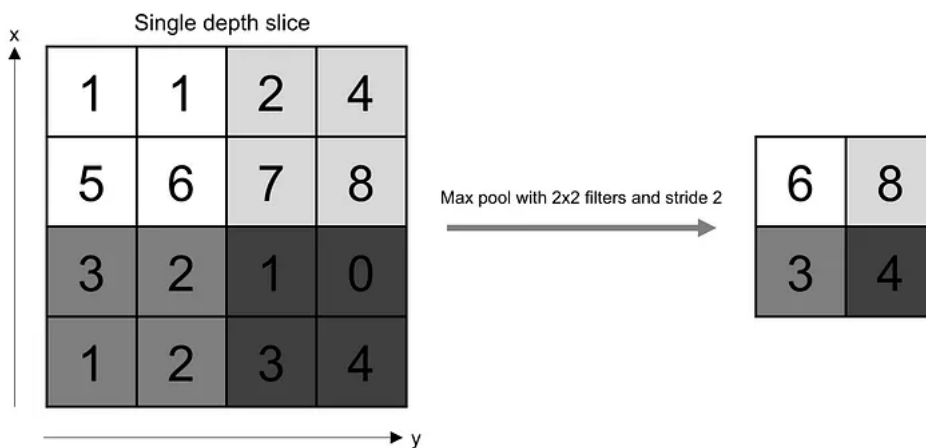
รูปที่ 2.1 การหา Feature map

ที่มา : [https://miro.medium.com/v2/resize:fit:720/format:webp/1\\*13ZUdVTQwWuhDPmo3JKag.png](https://miro.medium.com/v2/resize:fit:720/format:webp/1*13ZUdVTQwWuhDPmo3JKag.png)

13ZUdVTQwWuhDPmo3JKag.png

จากรูปจะเป็นการใช้เคอร์เนลขนาด  $2 \times 2$  ซึ่งต้องทำการเลื่อนผ่านภาพข้อมูลนำเข้าที่มีขนาด  $4 \times 4$  โดยมีการทำ Stride เท่ากับ 1 และไม่มีการทำ Padding ขนาดของ Feature map จะมีขนาดเท่ากับ  $3 \times 3$  เนื่องจากว่าไม่มีการทำ Padding ทำให้ Feature map ที่ได้มีขนาดเล็กกว่าภาพข้อมูลนำเข้า โดยเริ่มต้นต้องนำค่าภายในเคอร์เนลที่เลื่อนผ่านในขั้นแรกคูณกับค่าของพิกเซลภาพต้นฉบับตามตำแหน่ง จะได้  $aw+bx+ey+fz$  ตามภาพ หลังจากนั้นทำจนครบทั้งหมด 9 ครั้ง จะได้ Feature map ที่ไว้สำหรับใช้ต่อไป

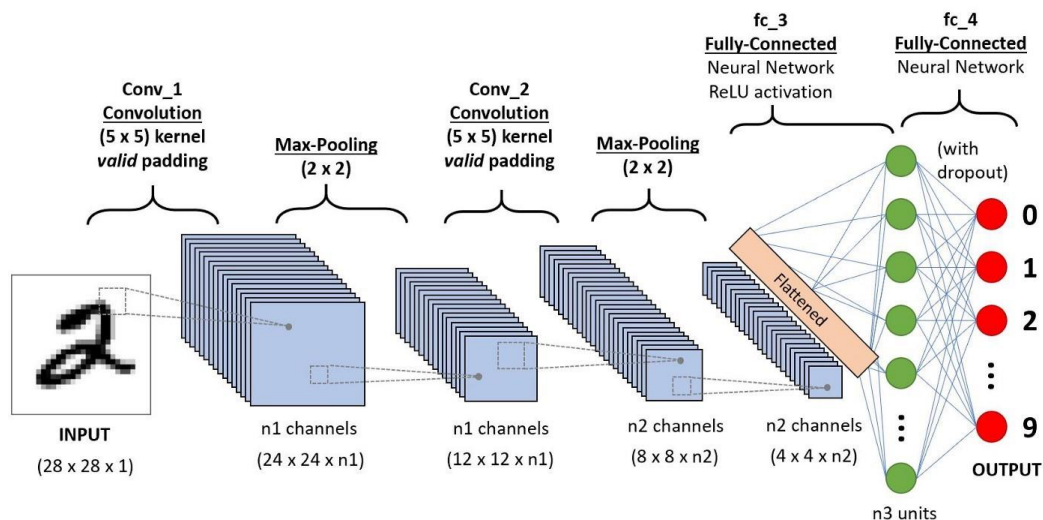
2.2.2 ชั้นพูลลิ่ง (Pooling layer) ชั้นพูลลิ่งเป็นชั้นที่อยู่ถัดจากชั้นคอนโวลูชัน โดยทำหน้าที่ย่อฟังก์ชันลักษณะให้เล็กลง เพื่อให้มีการทำงานที่ไวขึ้น โดยรายละเอียดของภาพไม่ได้ถูกลดลงไปจนไม่สามารถแยกคุณลักษณะเด่นได้ วิธีการพูลลิ่งมีหลายวิธี เช่น ใช้วิธีหาค่าต่ำสุด สูงสุด เฉลี่ย และผลรวม แต่ละวิธีมีความแตกต่างเพียงเล็กน้อย



รูปที่ 2.2 การทำ pooling จากภาพ

ที่มา : [https://miro.medium.com/v2/resize:fit:720/format:webp/1\\*sK7oP1m129V\\_oNGSsHlm\\_w.png](https://miro.medium.com/v2/resize:fit:720/format:webp/1*sK7oP1m129V_oNGSsHlm_w.png)

2.2.3 ชั้นเชื่อมต่อสมบูรณ์ (Fully Connected Layer) เป็นชั้นสุดท้ายของโครงข่ายคอนโวลูชัน นำผลลัพธ์จากชั้นก่อนๆ มาจำแนกให้ออกมาเป็นผลลัพธ์ที่ต้องการตามที่ได้ออกแบบจำนวนประเภทของข้อมูลที่ต้องการแบ่ง โดยใช้วิธีการเชื่อมต่อทุกๆ โหนดในทุกชั้นกับโหนดในชั้นถัดไป และจะมี ค่าน้ำหนักที่ต้องปรับเพื่อให้โมเดลสามารถเรียนรู้และทำนายผลลัพธ์ได้ดีขึ้น



รูปที่ 2.3 โครงข่ายคอนโวลูชัน

ที่มา : [https://miro.medium.com/max/1400/1\\*uAeANQIOQPqWZnnuH-VEyw.jpeg](https://miro.medium.com/max/1400/1*uAeANQIOQPqWZnnuH-VEyw.jpeg)

## 2.2 โครงข่ายการเข้ารหัสอัตโนมัติ (Autoencoder)

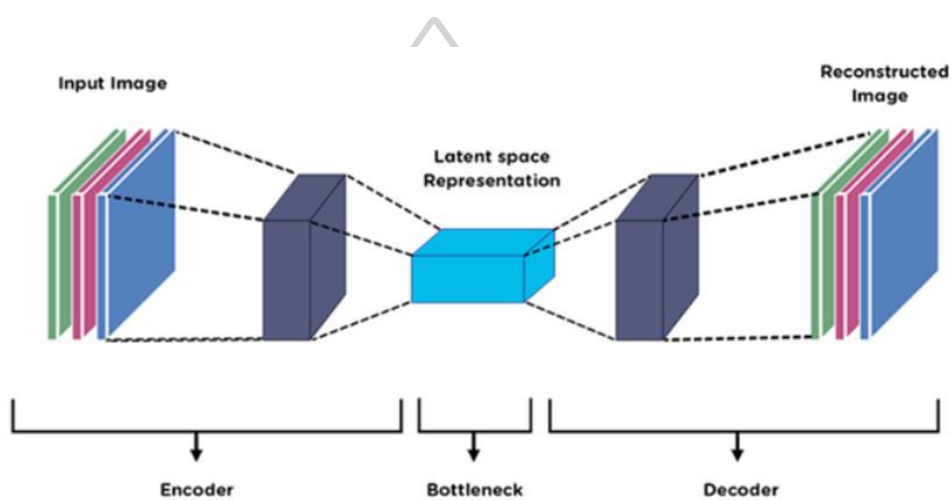
โครงข่ายการเข้ารหัสอัตโนมัติเป็นโครงข่ายที่สามารถเรียนรู้ได้ด้วยตนเอง ไม่จำเป็นต้องมี ผู้สอน ใช้เพียงข้อมูลนำเข้า และข้อมูลส่งออกสำหรับการเรียนรู้ที่ต้องการ ซึ่งจุดประสงค์ของโครงข่ายนี้ ต้องการให้ฝั่งอินพุตแปลงข้อมูลให้เหมือนในฝั่งเอาต์พุต ผลลัพธ์จะได้ข้อมูลที่มีลักษณะเหมือนฝั่งเอาต์พุตที่ได้ทำการระบุไปในโครงข่าย โดยตัวเข้ารหัสอัตโนมัติจำเป็นต้องใช้ข้อมูลสำหรับการฝึกที่มีลักษณะเฉพาะที่เหมือนกับฝั่งเอาต์พุตที่ต้องการ ในส่วนของเอาต์พุตนั้นอาจจะไม่เหมือนภาพต้นฉบับทั้งหมด เนื่องจากในระหว่างการฝึกสอนมีการสูญเสียข้อมูลเนื่องจากการบีบอัดข้อมูลในส่วนกลางของโครงสร้าง ส่วนประกอบที่สำคัญของโครงข่ายการเข้ารหัสอัตโนมัติสามารถแบ่งออกเป็น 3 ส่วนหลักได้ดังนี้

2.2.1 การเข้ารหัส (Encoder) เป็นส่วนที่ทำหน้าที่เข้ารหัสข้อมูล โดยมีพื้นฐานโครงข่ายมาจากชั้นคอนโวลูชัน โดยขนาดของข้อมูลที่รับเข้ามาจะมีขนาดเล็กลงไปทุกชั้นตามการออกแบบชั้นคอนโวลูชัน เปรียบเสมือนเป็นการบีบอัดข้อมูลให้เล็กลง เพื่อนำไปสู่ส่วนถัดไปเรียกว่า ชั้นแฝง (Latent layer)

2.2.2 ชั้นแฝง (Latent layer) เป็นส่วนที่ได้รับข้อมูลมาจากชั้นก่อนหน้าที่ได้ทำการลดขนาดข้อมูลลง ซึ่งเป็นส่วนสำคัญที่สุด เนื่องจากข้อมูลในชั้นนี้ต้องมีความพอดีในการออกแบบ ถ้าหากมี

ข้อมูลที่มีมากเกินไป ความสำคัญของข้อมูลก็จะน้อยลงทำให้ผลลัพธ์ที่ได้ไม่ค่อยดี แต่หากมีจำนวนน้อยไปผลลัพธ์ที่ได้ก็จะไม่ดีเหมือนกับกรณีแรก ดังนั้นการกำหนดจำนวนข้อมูลในชั้นนี้จึงมีความสำคัญอย่างมาก หลังจากนั้นข้อมูลจะถูกไปถอดรหัสในชั้นถัดไป เรียกว่า การถอดรหัส (Decoder)

2.2.3 การถอดรหัส (Decoder) เป็นส่วนที่ได้รับข้อมูลจากชั้นข้อมูลแฝง ซึ่งทำหน้าที่เพิ่ม ขนาดข้อมูลให้กลับเป็นขนาดเท่ากับตอนอินพุต มีพื้นฐานจากการทำคอนโวลูชัน ซึ่งเป็นการทำ เพิ่ม จำนวนข้อมูล ทีละชั้น จึงทำให้ข้อมูลเพิ่มขึ้นในทุกๆชั้น ผลลัพธ์ที่ได้จะเป็นไปตามการระบุเอาต์พุตที่โครงข่าย



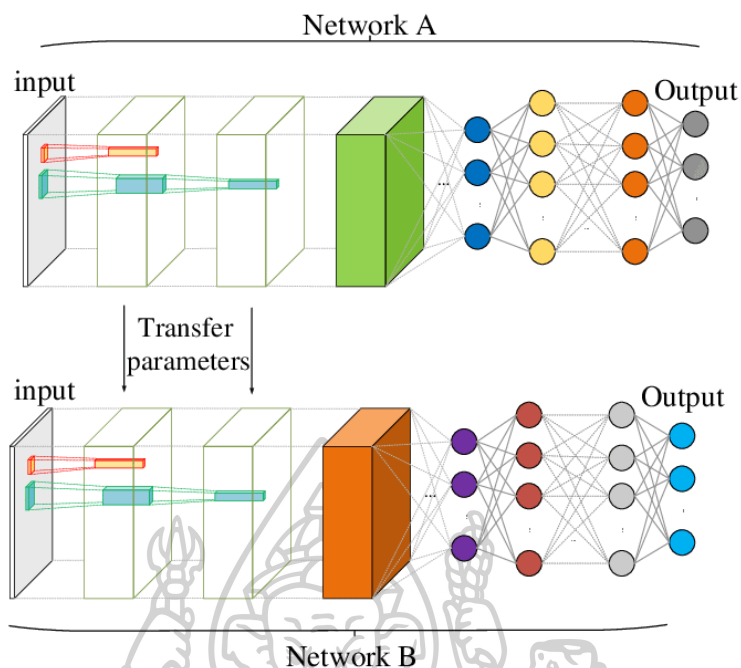
รูปที่ 2.4 โครงข่ายการเข้ารหัสอัตโนมัติ

ที่มา : [https://miro.medium.com/max/750/1\\*nqzWupxC60iAH2dYrFT78Q.png](https://miro.medium.com/max/750/1*nqzWupxC60iAH2dYrFT78Q.png)

## 2.3 Transfer learning

Transfer learning เป็นวิธีการหนึ่งที่จะช่วยเพิ่มประสิทธิภาพของโมเดลที่มีฐานข้อมูลจำนวนน้อย หรือต้องการประหยัดเวลาในการฝึกสอนโมเดล แต่ประสิทธิภาพลดลงเพียงเล็กน้อย โดยวิธีการของ Transfer learning มีอยู่หลายสามวิธีการหลัก คือ

2.3.1 การใช้โมเดลที่มีการฝึกสอนมาก่อนหน้า (Pre-trained model) ซึ่งเป็นโมเดลที่ถูกฝึกสอนไว้เป็นอย่างดีและมีการใช้ฐานข้อมูลขนาดใหญ่ในการฝึกสอน เพียงแต่โดเมนของโมเดลอาจจะไม่ตรงกับโดเมนฐานข้อมูลเฉพาะที่ต้องการ ดังนั้นจึงมีการนำ Weight จากโมเดลที่ถูกฝึกสอนมาก่อนโอนให้กับโมเดลใหม่ที่ต้องการฝึกสอนภายใต้ฐานข้อมูลเฉพาะที่ต้องการดังรูป 2.5



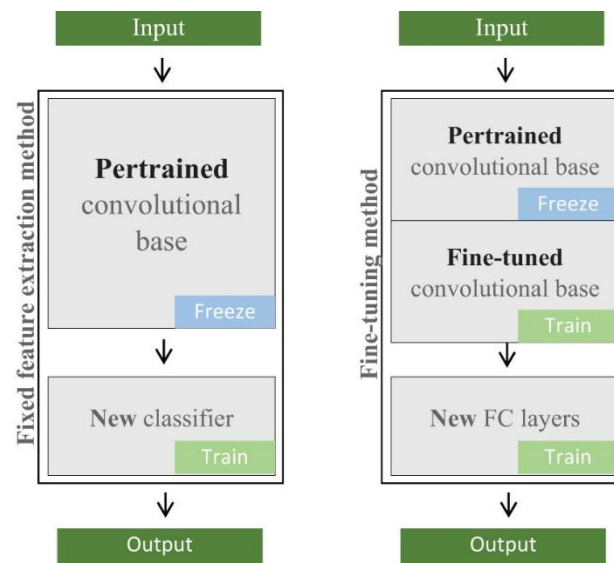
รูปที่ 2.5 การทำ Transfer learning ระหว่างสองโมเดล

ที่มา : [https://miro.medium.com/v2/resize:fit:720/format:webp/1\\*wHnkrPOjWZkmdzDYAdAuMg.png](https://miro.medium.com/v2/resize:fit:720/format:webp/1*wHnkrPOjWZkmdzDYAdAuMg.png)

จะเห็นว่าการทำ Transfer learning จะมีการโอน Weight ของโมเดลหนึ่งไปอีกโมเดลหนึ่งที่ต้องการ เฉพาะในส่วนของ CNN layer เท่านั้น และ Classification layer จะเป็นของโมเดลนั้นๆแทน ในส่วนของ CNN layer ที่ทำการโอนมานั้นจะทำการหยุดไว้ไม่ให้มีการฝึกสอน จะฝึกสอนเฉพาะ Classification layer เท่านั้น

2.3.2 การทำ Fine tuning ซึ่งเป็นวิธีการที่คล้ายกับวิธีการแรก เพียงแต่ในบางกรณี ประสิทธิภาพที่ได้อาจจะไม่ตามเป้าหมายที่ตั้งไว้ ดังนั้นจึงมีการใช้ Fine tuning เพื่อเพิ่มประสิทธิภาพ โดยการหยุดการเรียนรู้ของ CNN layer บางส่วนเท่านั้น และลบ CNN layer ช่วงปลายออก และเพิ่ม ชั้นเข้าไปใหม่ เพื่อให้โมเดลทำการฝึกสอน CNN layer ในส่วนปลาย และทำการฝึกสอนโมเดลใหม่ จะได้โมเดลที่ถูกฝึกสอนแบบวิธี Fine tuning สามารถปรับแต่งจำนวนชั้นได้ตามต้องการ เพื่อเพิ่ม ประสิทธิภาพ เนื่องจากในบางงานการเพิ่มจำนวนชั้นอาจจะทำให้เพิ่มประสิทธิภาพได้ แสดงดังรูป 2.6





รูปที่ 2.6 การทำ Fine tuning

ที่มา : [https://www.degruyter.com/document/doi/10.1515/jisys-2022-0198/asset/graphic/j\\_jisys-2022-0198\\_fig\\_001.jpg](https://www.degruyter.com/document/doi/10.1515/jisys-2022-0198/asset/graphic/j_jisys-2022-0198_fig_001.jpg)

#### 2.4 Median stack

Median stack เป็นเทคนิคที่ใช้ในงานประมวลผลภาพ หรืองานที่เกี่ยวกับภาพถ่าย ใช้วิธีการรวมภาพหลายๆภาพ โดยภาพที่นำมารวมกันต้องมีรายละเอียดบางส่วนของภาพที่แตกต่างกัน เพื่อให้ภาพผลลัพธ์ที่ได้มีคุณภาพในเรื่องของรายละเอียด และลดนอยส์ลงได้ [9] หลักการทำงานของ Median stack คือการหาค่ามัธยฐานของพิกเซลในภาพที่นำมารวมกัน โดยหาค่ามัธยฐานของพิกเซลที่ตรงกันในแต่ละภาพ เมื่อทำจนครบทุกพิกเซลจะได้ภาพเอาต์พุตที่ช่วยลดนอยส์ และเพิ่มรายละเอียดของภาพได้ดีขึ้น เนื่องจากพิกเซลที่เป็นนอยส์มักจะพบในบางภาพที่นำมาซ้อนกัน ทำให้เป็นค่าพิกเซลส่วนน้อยที่นำมาคำนวณในพิกเซลนั้นๆ จึงเป็นสาเหตุที่สามารถช่วยเพิ่มประสิทธิภาพของภาพได้เป็นอย่างดี นอกจากนี้การทำ Median stack ยังช่วยในเรื่องของโทนสี และความสว่างของภาพที่มีไม่เท่ากันในแต่ละภาพ ให้มีความใกล้เคียงกันอีกด้วย โดยสามารถแสดงสมการได้ดังนี้

$$I_{result}(x, y) = \text{Median}\{I_1(x, y), I_2(x, y), \dots, I_n(x, y)\} \quad (2.2)$$

เมื่อ  $I_{result}(x, y)$  คือค่าพิกเซลในตำแหน่ง  $(x, y)$  และ  $I_1(x, y), I_2(x, y), \dots, I_n(x, y)$  เป็นตำแหน่งของพิกเซลในแต่ละภาพที่ต้องการทำ Median stack



รูปที่ 2.7 การทำ Median stack

ที่มา : <https://frank.sauerburger.io/2021/12/28/median-stack.html>

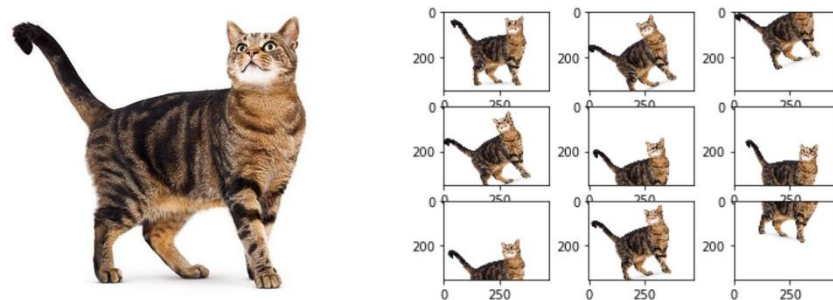
## 2.5 การเพิ่มจำนวนข้อมูลภาพ (Image augmentation)

เป็นวิธีการเปลี่ยนลักษณะหรือแก้ไขภาพให้มีความหลากหลาย เพื่อเพิ่มจำนวนข้อมูลไว้สำหรับการฝึกสอน นิยมใช้กับฐานข้อมูลที่มีจำนวนภาพน้อยหรือต้องการความหลากหลายของภาพมากยิ่งขึ้น ภาพที่ได้จะมีความแตกต่างเล็กน้อยจากภาพต้นฉบับ การเพิ่มจำนวนข้อมูลภาพยังช่วยให้การฝึกสอนโมเดลต่างๆมีประสิทธิภาพมากยิ่งขึ้น โดยวิธีแก้ไขภาพจะใช้เทคนิคต่างๆ เพื่อให้ภาพมีความหลากหลาย ซึ่งสามารถทำได้โดยการเปลี่ยนแปลงรูปแบบของภาพดังนี้

2.5.1 การหมุนภาพ (Rotation) เป็นการหมุนภาพให้มีความแตกต่างจากภาพต้นฉบับ ซึ่งสามารถหมุนไปในแนวนอนและแนวตั้งได้ การหมุนภาพจึงเป็นอีกหนึ่งวิธีที่ช่วยให้ภาพมีความหลากหลาย

2.5.2 การเปลี่ยนขนาดภาพ (Resizing) เป็นการปรับขนาดของภาพไม่ว่าจะเป็นการย่อภาพหรือขยายภาพในบริเวณที่ต้องการ การเปลี่ยนขนาดภาพนี้ทำให้ภาพมีความคมชัดและมีขนาดต่างๆ ซึ่งจะช่วยทำให้เพิ่มความหลากหลายมากยิ่งขึ้นในขณะทำการฝึกสอนแบบจำลอง

2.5.3 การเพิ่มสิ่งรบกวน (Noise) เป็นการเพิ่มความไม่แน่นอนให้กับภาพ โดยการเพิ่มเสียงรบกวนในภาพจะช่วยให้โมเดลสามารถเรียนรู้และทำนายภาพได้



รูปที่ 2.8 การเพิ่มจำนวนรูปภาพ

ที่มา : <https://149695847.v2.pressablecdn.com/wpcontent/uploads/2020/08/UKwFg.jpg>

## 2.6 YOLOV5

Yolov5 คือโมเดลสำหรับการตรวจจับวัตถุทั้งในภาพนิ่ง และภาพเคลื่อนไหว [10] (object detection) ที่พัฒนาขึ้นโดยใช้วิธีการเรียนรู้เชิงลึกร่วมกับ Convolutional Neural Network (CNN) โมเดลนี้เป็นตัวที่พัฒนาต่อจาก Yolov4 ซึ่งเป็นโมเดลที่อยู่ในกลุ่ม You Only Look Once (YOLO) ที่มีการพัฒนาในเรื่องของการตรวจจับวัตถุมาอย่างยาวนาน ในส่วนของ Yolov5 มีการวิเคราะห์ประสิทธิภาพจากโมเดลก่อน และนำมาแก้ไขใหม่จนพัฒนาระบบใหม่ในการสกัดลักษณะของวัตถุ (feature extraction) ซึ่งทำให้มีประสิทธิภาพในการตรวจจับวัตถุสูงขึ้น โดยเฉพาะการตรวจจับวัตถุที่มีขนาดเล็กหรืออยู่ในระยะไกลสามารถทำได้ดีกว่าโมเดลก่อนหน้า โดน Yolov5 มีส่วนประกอบที่สำคัญทั้งหมดสามส่วนได้แก่ Backbone, Neck และ Head

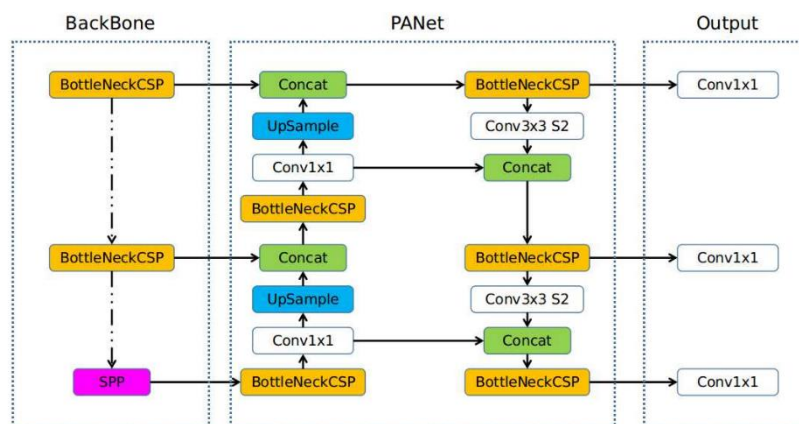
- Backbone โมดูลที่ใช้สำหรับการสกัดคุณลักษณะ (Feature Extraction) ของภาพ โดยใช้โมเดล CNN โดยมีตัวอย่างเช่น Darknet-53 ซึ่งประกอบไปด้วย Convolutional layers ที่ซับซ้อนและมีการใช้งาน skip connections ทำให้โมเดลมีความแม่นยำและความเร็วในการตรวจจับวัตถุ

- Neck โมดูลที่ใช้สำหรับการรวมคุณลักษณะที่ได้จาก Backbone เข้าด้วยกัน และเพิ่มความลึกของคุณลักษณะด้วยการใช้งาน Spatial Pyramid Pooling (SPP) เพื่อให้สามารถทำงานกับภาพขนาดต่างๆ ได้

- Head โมดูลที่ใช้สำหรับการคาดเดาความน่าจะเป็นของวัตถุ โดยใช้ layer ที่เชื่อมต่อกับ backbone และ neck เพื่อประมวลผลคุณลักษณะที่สกัดออกมา และทำการตรวจจับวัตถุ โดยมี layer สุดท้ายเป็น fully connected layer ที่ใช้สำหรับการทำนายขนาด ตำแหน่ง และหมวดหมู่ของวัตถุ โดย YOLOv5 ใช้หัวเดียวกับ YOLOv3 และ YOLOv4 มีส่วนประกอบคอนโวลูชันทั้งหมดสามชั้นที่ไว้ทำนายขอบเขต ของวัตถุที่ต้องการตรวจจับ ค่ะแนนและประเภทของวัตถุ

นอกจากนี้ YOLOv5 ยังมีการเพิ่มประสิทธิภาพด้วยการใช้โมเดล Backbone ที่ดีขึ้น ซึ่งเป็นโครงสร้างพื้นฐานของโมเดล CNN และใช้โมเดลตัวแรก (Anchor box) ที่เป็นส่วนประกอบสำคัญในการตรวจจับวัตถุ และยังมีการใช้งาน TensorRT เพื่อเพิ่มความเร็วในการประมวลผล โดยรวมแล้ว YOLOv5 เป็นโมเดลที่มีประสิทธิภาพและความแม่นยำ

Overview of YOLOv5



รูปที่ 2.9 โครงสร้างของ YOLOv5

ที่มา : <https://github.com/ultralytics/yolov5/issues/280>

## 2.7 Peak signal-to-noise ratio

Peak signal-to-noise ratio (PSNR) เป็นวิธีการเปรียบเทียบความใกล้เคียงกันของภาพทั้งสอง ถ้ามีค่าที่สูงบ่งบอกถึงมีประสิทธิภาพความใกล้เคียงกัน โดยสามารถคำนวณได้ดังสมการนี้

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \quad (2.3)$$

โดยในสมการนี้เป็นการหาค่าเฉลี่ยความผิดพลาดกำลังสอง ซึ่ง  $m$  และ  $n$  เป็นจำนวนพิกเซลของภาพในแนวตั้ง และแนวนอน และ  $I(i,j)$  เป็นค่าในพิกัดของภาพที่จะนำมาเปรียบเทียบ  $K(i,j)$  เป็นค่าในพิกัดของภาพต้นฉบับ หลังจากนั้นนำค่าที่ได้ไปหา PSNR ดังต่อไปนี้

$$PSNR = 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \quad (2.4)$$

โดยในส่วนของ  $MAX$  เป็นค่าสูงสุดของพิกเซลในภาพ

## 2.8 Structural similarity index measure

Structural similarity index measure (SSIM) เป็นอีกหนึ่งวิธีสำหรับหาความคล้ายกันของภาพทั้งสอง โดยมีช่วงของค่าอยู่ที่ 0 ถึง 1 หากค่าที่ได้ใกล้เคียง 1 มาก ความใกล้เคียงกันของทั้งสองภาพยิ่งใกล้เคียงกัน โดยมีสมการดังนี้

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2.5)$$

เมื่อ  $\mu_x, \mu_y$  คือค่าเฉลี่ยของ  $x, y$  และ  $\sigma_x, \sigma_y$  คือความแปรปรวนของ  $x, y$  ตามลำดับ และ  $\sigma_{xy}$  คือความแปรปรวนร่วมเกี่ยวของ  $x, y$

## 2.9 Confusion matrix

Confusion matrix เป็นเครื่องมือทางสถิติที่ใช้สำหรับการประเมินประสิทธิภาพของระบบการจัดการประเภทต่างๆ โดยเฉพาะในงานการเรียนรู้เชิงลึก ซึ่งจะช่วยให้ผู้วิเคราะห์ข้อมูล หรือทดสอบระบบ สามารถปรับปรุงโมเดลหรือระบบการจัดการประเภทข้อมูลให้ดียิ่งขึ้นได้ โดยจะประกอบไปด้วยตารางขนาด  $N \times N$  (โดย  $N$  คือจำนวนคลาส) ซึ่งแสดงผลการทำนายของโมเดล/ระบบ โดยแต่ละเซลล์ในตารางจะแสดงจำนวนของตัวอย่างที่ตัวทำนายแยกต่างคลาสดูถูกต้องหรือผิดพลาด

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

รูปที่ 2.10 ตัวอย่าง Confusion matrix

ที่มา : <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>

จากรูปข้างต้นนี้ เราสามารถสรุปผลการทำนายของโมเดลได้ดังนี้

-True Positive (TP) คือ ผลการทำนายที่ตรงกับผลลัพธ์จริง โดยที่โมเดลทำนายว่าเป็น positive และ ground truth ก็เป็น positive

-False Positive (FP) คือ ผลการทำนายที่ไม่ตรงกับผลลัพธ์จริง โดยที่โมเดลทำนายว่าเป็น positive แต่ ground truth เป็น negative

-False Negative (FN) คือ ผลการทำนายที่ไม่ตรงกับผลลัพธ์จริง โดยที่โมเดลทำนายว่าเป็น negative แต่ ground truth เป็น positive

-True Negative (TN) คือ ผลการทำนายที่ตรงกับผลลัพธ์จริง โดยที่โมเดลทำนายว่าเป็น negative และ ground truth ก็เป็น negative

## 2.10 การตรวจจับภาพขอบ (Edge detection)

Edge detection เป็นการตรวจจับขอบของภาพโดยใช้ความต่างของความเข้มของภาพที่อยู่ติดกันของสองพิกเซลเพื่อตรวจจับเส้นขอบ ซึ่งการตรวจจับภาพขอบเป็นสิ่งหนึ่งที่สำคัญกับงานทางด้านความรู้จำแบบ การแบ่งส่วนของภาพ โดย Edge detection สามารถมองได้ว่าเป็นตัวกรองชนิดหนึ่งที่สามารถสกัดคุณลักษณะของขอบภาพได้ จากการหาคอนทัวร์ที่ติดกันของพิกเซลในภาพ ซึ่งวิธีการเส้นขอบภาพสามารถแบ่งออกเป็นสองกลุ่มใหญ่ได้ดังนี้

2.10.1 search-based เป็นวิธีการตรวจจับขอบโดยการคำนวณการวัดความแข็งแรงของขอบก่อน นิยมใช้กับการหาขอบภาพโดยใช้ข้อมูลพิกเซลอันดับที่หนึ่ง จะใช้การหาค่า Gradient magnitude โดยพิจารณาจากพิกเซลทั้งหมดภาพในภาพ โดยไล่ไปจนครบพิกเซลในภาพ เมื่อค่า Gradient magnitude มากกว่าค่าเกณฑ์ของภาพเท่ากับว่าพิกเซลนั้นเป็นภาพขอบ โดยวิธีการการหาขอบภาพที่นิยมใช้มีดังนี้

- Sobel edge detection

Sobel operator ในบางครั้งเรียกว่า Sobel–Feldman operator หรือ Sobel filter [11] เป็นเทคนิคที่ไว้ใช้สำหรับการตรวจจับภาพขอบ โดยสามารถทำการตรวจจับได้ทั้งในแนวตั้ง และแนวนอน โดยจะมีเคอร์เนลขนาด 3x3 ที่ทำการคอนโวลูชันกับภาพต้นฉบับเพื่อหาความเข้มที่แตกต่างกันของพิกเซลรอบๆของเคอร์เนล โดยจะทำการหาทั้งแนวตั้ง และแนวนอน หลังจากนั้นจึงนำมารวมกัน โดยคำนวณได้ดังนี้

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \text{ and } G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (2.6)$$

ซึ่ง  $G_x$  เป็นผลลัพธ์จากนำเคอร์เนลไปคอนโวลูชันกับ A ที่ภาพที่ต้องการหาภาพขอบที่ต้องการหาขอบในแนวตั้ง ในส่วนของ  $G_y$  เป็นการหาภาพขอบในแนวนอน ซึ่งทั้งสองตัวนี้สามารถนำมารวมกันเพื่อหาภาพขอบทั้งสองแนวได้ดังนี้

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.7)$$

- Canny edge detection

Canny edge detection [12] เป็นการตรวจจับภาพขอบที่นิยมเป็นอันดับต้นๆ โดยมีขั้นตอนการทำหลักอยู่ 5 ขั้นตอนคือ

1. การลบรอยส้ เพื่อให้ได้ภาพที่มีความปราศจากสิ่งรบกวนมากที่สุด และให้ภาพที่เรียบยิ่งขึ้น โดยส่วนใหญ่จะใช้ Gaussian filter ในการลบรอยส้ โดยแสดงฟิลเตอร์ได้ดังต่อไปนี้

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} \quad (2.8)$$

เมื่อ  $G(x,y)$  เป็นค่าของ Gaussian filter และ  $x$  เป็นตำแหน่งของพิกเซลในแนวแกน  $x$  ส่วน  $y$  เป็นตำแหน่งพิกเซลในแนวแกน  $y$  และมี  $\sigma$  กำหนดความกว้างของเส้นขอบ Gaussian

2. Gradient Calculation เป็นขั้นตอนการตรวจจับความเข้มและทิศทางของเส้นขอบ ซึ่งในวิธีการของ Canny จะใช้วิธีการ Sobel operator เพื่อหาเส้นขอบในแนวตั้ง และแนวนอน

3. Non-maximum suppression เป็นการหาเส้นขอบที่ในพิกเซลที่ค่าสูงที่สุด ซึ่งจะทำการไล่ไปในทุกพิกเซล และหาทิศทางของเส้นขอบในแนวนั้นๆ เพื่อหาค่าสูงสุด โดยจุดประสงค์เพื่อต้องการเพียงแต่ขอบที่มีลักษณะเด่นที่สุดในทิศทางขอบของพิกเซลตำแหน่งนั้นๆ โดยจะทำให้ภาพขอบมีลักษณะเส้นที่บาง แต่ยังคงรายละเอียดได้ชัดเจนที่สุด

4. Double threshold เป็นการแบ่งเกณฑ์ค่าพิกเซลโดยจะแบ่งเป็นเกณฑ์สูง และเกณฑ์ต่ำ หากพิกเซลที่อยู่ระหว่างค่าทั้งสองเกณฑ์จะนำมาใช้เป็นภาพขอบต่อไป ซึ่งจะเป็นพิกเซลขอบที่มีความชัดเจนมากที่สุด

5. Edge Tracking by Hysteresis เป็นวิธีการที่ทำให้พิกเซลที่มีความโดดเด่นน้อยแปลงให้มีความโดดเด่นมากยิ่งขึ้น หากบริเวณรอบๆ พิกเซลนั้นมีค่าพิกเซลที่โดดเด่นอยู่บริเวณรอบๆ



2.10.2 Zero-crossing เป็นวิธีการตรวจจับขอบภาพที่ใช้อนุพันธ์อันดับที่สอง ซึ่งเป็นการแก้ปัญหาภาพขอบแบบแรกที่มีความหนาเกินไป โดยวิธีนี้จะทำการหาจุดตัดศูนย์ ซึ่งจะเป็นตำแหน่งของภาพขอบ โดยตัวอย่างวิธีการที่นำ Zero-crossing คือ Laplacian

- Laplacian edge

Laplacian edge detection [13] เป็นการตรวจจับขอบภาพ ซึ่งมีความแตกต่างจากสองวิธีการที่ได้กล่าวมาคือ เป็นการใช้อนุพันธ์อันดับสองในการหาเส้นขอบ การหาเส้นขอบจะมีวิธีการหาที่แตกต่างจากวิธีก่อนๆ โดยจะมีทั้งวิธีการหาแบบขอบใน และแบบขอบนอก ซึ่งจะใช้ฟิลเตอร์ที่ต่างกัันเรียกว่า ตัวดำเนินการเชิงบวก และตัวดำเนินการเชิงลบ โดยแสดงการฟิลเตอร์ทั้งสองแบบดังนี้

$$G_{positive} = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A$$

$$G_{negative} = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (2.9)$$

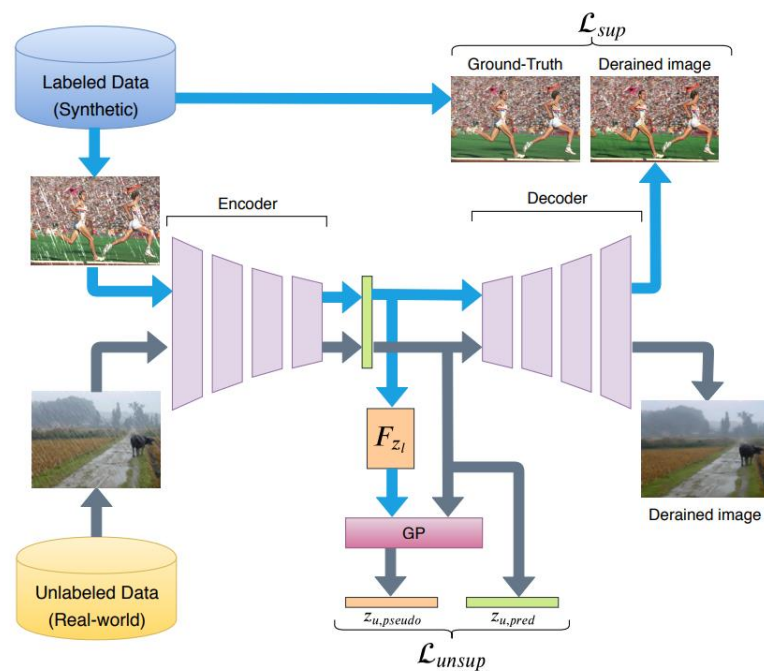
ซึ่ง  $G_{positive}$  เป็นภาพขอบจากการหาแบบเชิงบวก และ  $G_{negative}$  เป็นการหาภาพขอบจากเชิงลบ โดยมี A คือภาพที่ต้องการทำการหาภาพขอบ

## 2.11 งานวิจัยที่เกี่ยวข้อง

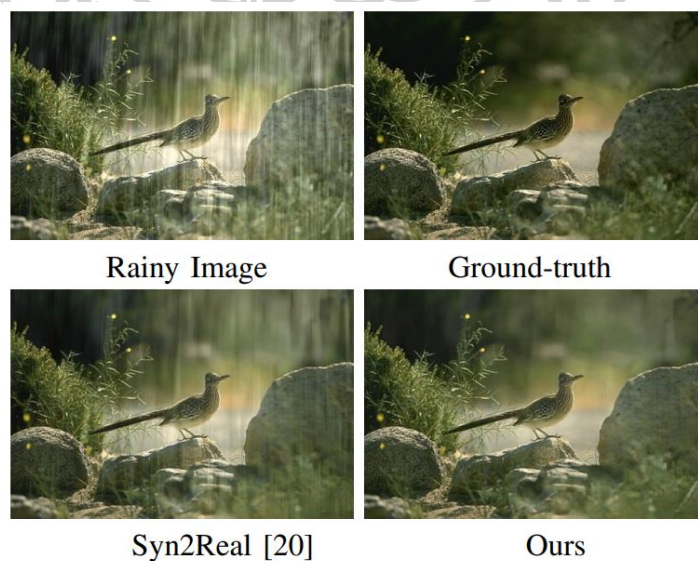
2.11.1 Semi-Supervised Image Deraining using Gaussian Processes(Rajeev Yasarla, Vishwanath A. Sindagi and Vishal M. Patel) [14]

ในงานวิจัยนี้ได้นำเสนอวิธีการลบฝน โดยใช้วิธีการ U-net ในการฝึกสอนใช้ข้อมูลทั้งที่มีการระบุตำแหน่งของฝน และไม่มีการระบุตำแหน่งของฝน เพื่อเพิ่มประสิทธิภาพให้กับการนำไปทดสอบกับภาพฝนจริง โดยมีการนำ Gaussian Processes ใช้หาค่าที่เหมาะสมระหว่างภาพที่ถูกระบุตำแหน่งฝน และไม่มีระบุตำแหน่งฝน โดยภายในโครงสร้างของ U-net ที่นำมาใช้ ภายในจะประกอบไปด้วยส่วน Encoder ที่เริ่มต้นมาใช้ Conv2d ขนาด 3 x 3 และใช้ Res2Block จำนวน 5 ชั้น โดยในระหว่างชั้นมี Downsample เพื่อลดข้อมูลในระหว่างชั้น ในส่วนของ Decoder เริ่มต้นใช้

Res2Block จำนวน 3 บล็อก โดยระหว่างชั้นมี Upsample เพื่อขยายข้อมูลระหว่างชั้น โดยในชั้นสุดท้ายใช้ Conv2d ขนาด 3 x 3 โดยผลลัพธ์จะได้ภาพ-ที่ถูกกำจัดฝน โดยมีการวัดประสิทธิภาพโดยใช้ฐานข้อมูลสามตัว คือ Rain800 Rain100L และ DDN โดยใช้ PSNR และ SSIM ในการวัดประสิทธิภาพ โดยมีการทดสอบให้เห็นระหว่างการนำข้อมูลที่ระบุตำแหน่งของฝนเพียงอย่างเดียวไป เทรนพบว่าประสิทธิภาพที่ได้ดีกว่าการนำภาพฝนที่ไม่ถูกระบุตำแหน่งไปฝึกสอนร่วมด้วย



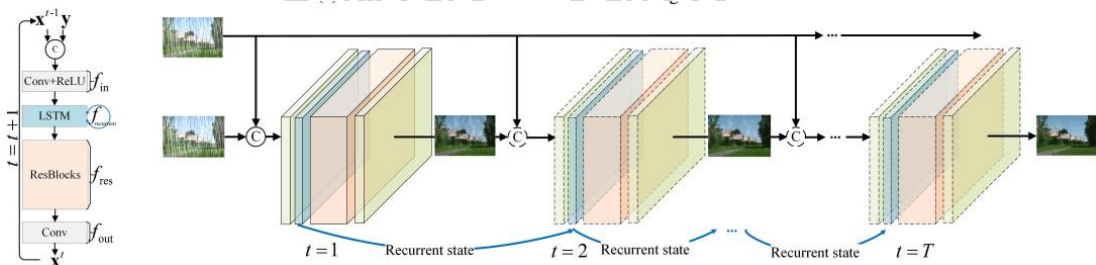
รูปที่ 2.11 โครงสร้างสถาปัตยกรรม Semi-Supervised Image Deraining



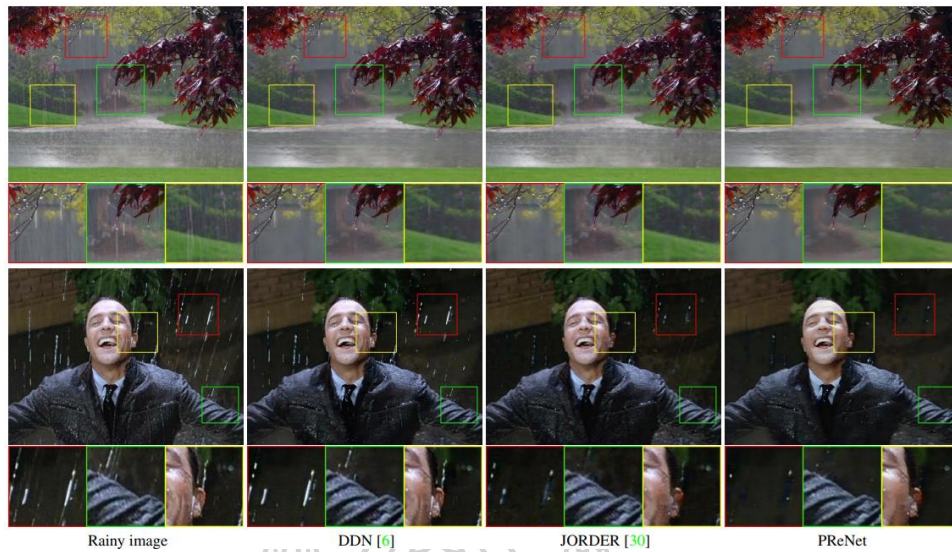
รูปที่ 2.12 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น

2.11.2 Progressive Image Deraining Networks: A Better and Simpler Baseline(Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu and Deyu Meng) [15]

ในงานวิจัยนี้ได้นำเสนอการลบฝนโดยใช้การคำนวณแบบวนซ้ำ โดยมีพื้นฐานมาจากการเรียนรู้เชิงลึก เรียกว่า Progressive recurrent network (PReNet) ซึ่งจะพิจารณาแก้ไขในส่วนของโครงข่ายเป็นหลัก ในโครงข่ายมีการใช้ Resblock ในแต่ละการวนซ้ำ ทำให้พารามิเตอร์มีการเพิ่มขึ้นอย่างมาก และเสี่ยงต่อปัญหาการ Over-fitting ดังนั้นจึงได้ทำการแทรก Long short term memory เพื่อลดปัญหาที่เกิดขึ้นในแต่ละการวนซ้ำ โดยในส่วนของฟังก์ชันเป้าหมายจะใช้วิธีการ Mean square error ร่วมกับ SSIM ซึ่งช่วยให้การปรับ hyper-parameter ได้ดีขึ้น ผลลัพธ์ที่ได้จากโมเดลที่ได้ทำการฝึกสอน มีการวัดประสิทธิภาพด้วย PSNR และ SSIM โดยเปรียบเทียบประสิทธิภาพที่ได้ในแต่ละการวนซ้ำ พบว่าค่าที่ได้เพิ่มขึ้นจนถึงรอบที่ 7 พบว่าประสิทธิภาพที่ได้ดีขึ้น จึงสรุปได้ว่าการวนซ้ำต้องมีจำนวนรอบที่พอดี จึงจะได้ประสิทธิภาพสูงสุด หลังจากนั้นมีการลองเปรียบเทียบกับวิธีการอื่นโดยใช้ฐานข้อมูลทั้งสามฐานข้อมูล คือ Rain100H Rain100L และ Rain12 พบว่าได้ค่าประสิทธิภาพได้ดีที่สุด



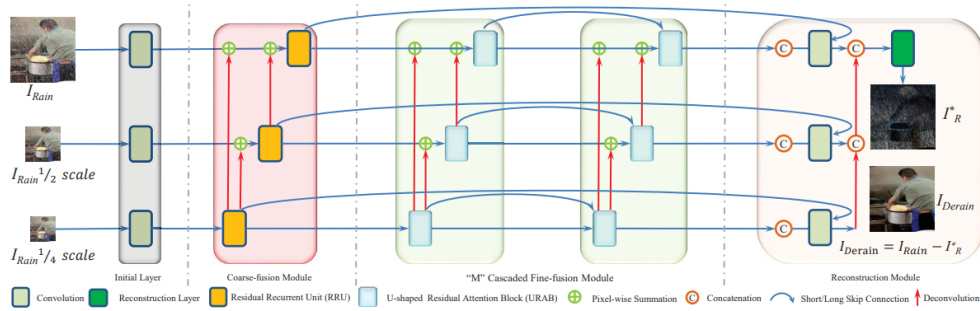
รูปที่ 2.13 โครงสร้างสถาปัตยกรรม PreNet



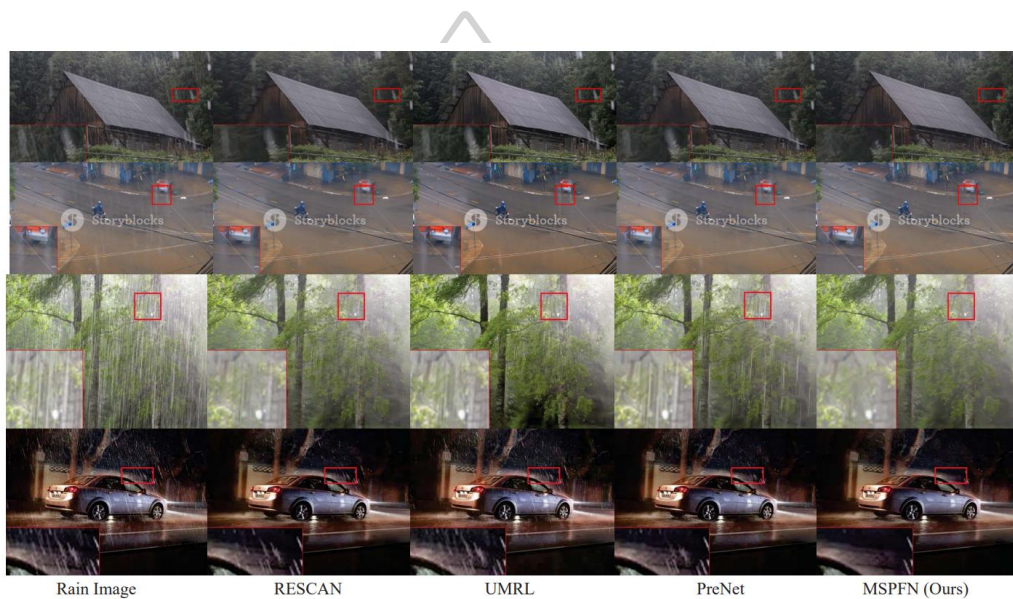
รูปที่ 2.14 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น

2.11.3 Multi-Scale Progressive Fusion Network for Single Image Deraining (Kui Jiang<sup>1</sup> Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma and Junjun Jiang)[1]

ในงานวิจัยนี้เสนอวิธีการลบฝนที่เรียกว่า MSPFN ซึ่งเป็นการใช้วิธีการย่อภาพให้มีขนาดที่ หลากหลาย โดยภาพที่ถูกย่อจะใช้วิธีการ Gaussian ในการย่อภาพ โดยภาพที่ถูกย่อจะมีรูปแบบของ ฝนที่มีบางส่วนที่คล้ายคลึงกันกับภาพต้นฉบับ ซึ่งทำให้เพิ่มประสิทธิภาพการสร้างชั้นของฝนใหม่ เพื่อ นำมาลบกับภาพต้นฉบับได้ โดยภายในโครงข่ายจะแบ่งออกเป็นสามส่วนใหญ่ ในส่วนแรกเรียกว่า Coarse Fusion ซึ่งภายในจะมีการรับภาพหลายขนาดมาเข้ากระบวนการเรียนรู้แบบวนซ้ำเพื่อสกัด คุณลักษณะต่างๆ แล้วนำผลลัพธ์จากภาพขนาดเล็กสุดรวมกับขนาดอื่นๆ หลังจากนั้นจะไปยังส่วนที่ สองเรียกว่า Fine Fusion ทำหน้าที่ปรับแต่งความสัมพันธ์ของข้อมูลของภาพในแต่ละขนาดที่ไม่ เท่ากัน ต่อมาในขั้นสุดท้ายเรียกว่า Reconstruction เป็นการนำผลลัพธ์จากทุกๆส่วน และในทุก ขนาดของภาพมาเข้าสู่กระบวนการคอนโวลูชัน เพื่อหาชั้นของฝน และนำไปลบกับภาพฝนต้นฉบับจะ ได้ผลลัพธ์ของฝน ซึ่งมีการทดสอบประสิทธิภาพทั้งในภาพฝนจริงและภาพฝนที่สังเคราะห์ขึ้นมา โดย ใช้ PSNR และ SSIM ในการเปรียบเทียบ พบว่าสามารถทำการลบฝนได้อย่างมีประสิทธิภาพทั้งคู่



รูปที่ 2.15 โครงสร้างสถาปัตยกรรม MSPFN



รูปที่ 2.16 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น

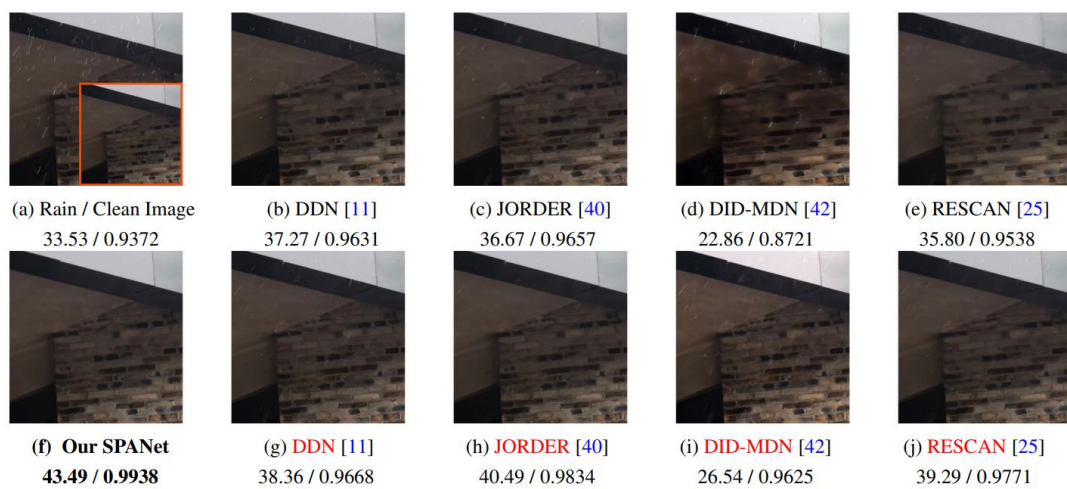
2.11.4 Spatial Attentive Single-Image Deraining with a High Quality Real Rain Dataset(Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang and Rynson W.H. Lau) [17]

ในทางวิจัยนี้เสนอวิธีการลบฝนชื่อว่า SPANet โดยมีการฝึกสอนโดยใช้ข้อมูลที่สังเคราะห์ขึ้นมาทั้งหมด เนื่องจากว่าภาพฝนจริงไม่สามารถมีการจับคู่กันระหว่างภาพฝน และภาพที่ไม่โดนฝนได้ โดยในวิธีนี้มีการรับภาพฝนเข้าไปโดยตรง ผลลัพธ์ที่ได้จะเป็นการลบฝนโดยตรง ในสถาปัตยกรรมนี้มีส่วนที่สำคัญคือ Spatial Attentive Block เป็นการใช้การเรียนรู้แบบวนซ้ำที่ช่วยให้การเรียนรู้ที่มีข้อมูลที่เรียงต่อกันจำนวนมาก สามารถฝึกสอนได้อย่างมีประสิทธิภาพ เมื่อทำการทดสอบผลลัพธ์ของ

การลบฝนกับ PSNR และ SSIM โดยใช้ทั้งภาพจริง และภาพสังเคราะห์ พบว่าสามารถลบฝนไปเป็น  
 อย่างดี แต่ก็ยังมีข้อบกพร่องอยู่ในบางภาพ เช่น ภาพที่เกิดหมอกเป็นจำนวนมาก วิธีการนี้จะไม่  
 สามารถลบฝนได้



รูปที่ 2.17 โครงสร้างสถาปัตยกรรม SPANet

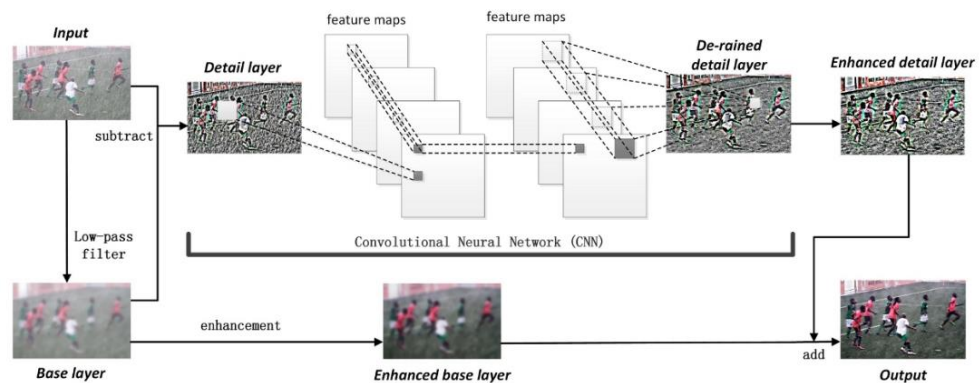


รูปที่ 2.18 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น

2.11.5 Clearing the Skies: A deep network architecture for single-image rain removal (Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao and John Paisley) [18]

ในงานวิจัยนี้ได้นำเสนอการลบฝนที่เรียกว่า DerainNet มีส่วนสำคัญคือการแยกภาพที่รับเข้ามาเป็นภาพ base layer และ detail layer และมีการใช้ CNN สำหรับช่วยในการลบฝน โดยภายใน CNN ประกอบไปด้วย Hidden layer จำนวน 2 ชั้น และ Output layer จำนวน 1 ชั้น ซึ่งในส่วนของการลบฝนจะใช้เพียง detail layer เนื่องจากในชั้นนี้คงรายละเอียดของภาพฝนไว้ทั้งหมด การจัดการในชั้นนี้จึงเป็นสิ่งที่ง่ายกว่านำภาพฝนต้นฉบับมาทำการลบฝนโดยตรง ถึงแม้ใน detail layer จะถูกลบฝนไปจนหมด แต่ในภาพยังคงหลงเหลือหมอกที่เกิดจากละอองฝน จึงใช้วิธีการปรับปรุงภาพใน detail layer อีกครั้งเพื่อกำจัดหมอกออก สำหรับฐานข้อมูลภาพที่นำมาฝึกสอนนั้น ได้มีการใช้ภาพฝนสร้าง

ขึ้นมาเอง ในบทความนี้ได้้นำ DerainNet วัดประสิทธิภาพกับฝนจริง และฝนที่สร้างขึ้นเองพบว่าสามารถลบฝนได้เป็นอย่างดีทั้งในฝนจริง และฝนที่สร้างขึ้นมาเอง และยังมีรายละเอียดพื้นหลังที่อยู่ครบ



รูปที่ 2.19 โครงสร้างสถาปัตยกรรม DerainNet

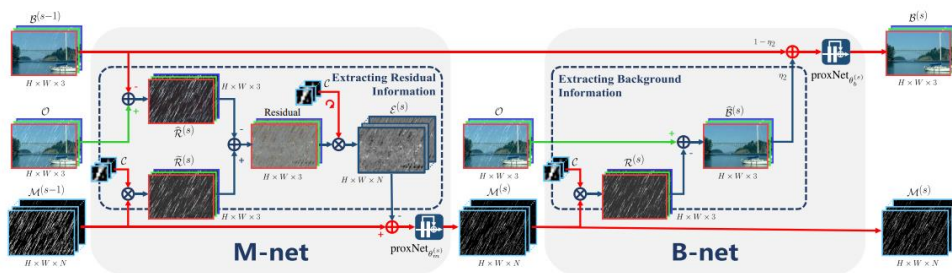


รูปที่ 2.20 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น

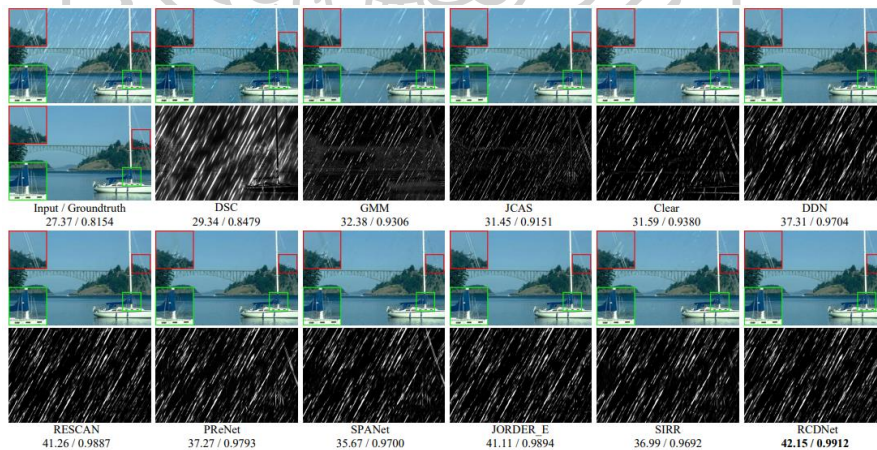
2.11.6 A Model-driven Deep Neural Network for Single Image Rain Removal(Hong Wang<sup>1</sup>, Qi Xie<sup>1</sup>, Qian Zhao<sup>1</sup> and Deyu Meng) [19]

ในงานวิจัยนี้ได้นำเสนอวิธีการ RCDNet เป็นวิธีลบฝน โดยมีการศึกษาลักษณะของฝนที่แท้จริง เพื่อทำการแยกระหว่างชั้นฝน และชั้นพื้นหลังออกจากกัน โดยในวิจัยนี้ได้กล่าวถึงว่าภาพฝนนั้นเกิดจากภาพพื้นหลังที่มีฝนรวมกับภาพของชั้นฝนที่มีเพียงในเท่านั้น จึงได้ทำการหาชั้นของฝนเพื่อ

นำมาลบกับภาพฝนดั้งเดิม จะได้ผลลัพธ์เป็นภาพพื้นหลังที่ไม่มีฝน โดยมีการนำเทคนิคการหาฝนจากฐานข้อมูลที่น่ามาทดสอบโดยเรียกว่า Convolution dictionary learning โดยในโครงสร้างของ RCDNet จะมีวิธีการทำซ้ำไปเรื่อยๆ จนกว่าจะได้ผลลัพธ์การลบฝนที่ดีที่สุด โดยจะแบ่งเป็นสองส่วนสำคัญ คือ M-net และ B-net โดย M-net ทำหน้าที่หาชั้นของฝน โดยหลักการคือต้องหารานละเอียด ภาพพื้นหลังที่ติดมากับชั้นฝนก่อน เพื่อนำไปลบกับชั้นฝนจะได้ชั้นฝนที่ไม่มีรายละเอียดพื้นหลังติดมา ต่อมาจะนำผลลัพธ์จากM-net ส่งให้กับ B-net ทำการลบฝนออกจากภาพต้นฉบับ และเข้าสู่ proxNet เป็นวิธีการ Optimization ภาพให้มีความละเอียดมากยิ่งขึ้น โดยภาพที่ได้ก็จะมี การลบฝนจนหมด ในวิจัยนี้มีการวัดประสิทธิภาพโดยใช้ PSNR และ SSIM เปรียบเทียบกับวิธีการอื่น ปรากฏว่า RCDNet มีประสิทธิภาพของการลบฝนดีมากที่สุดจากการทดสอบกับฝนจริง และฝนที่สร้างขึ้นมาเอง



รูปที่ 2.21 โครงสร้างสถาปัตยกรรม RCDNet

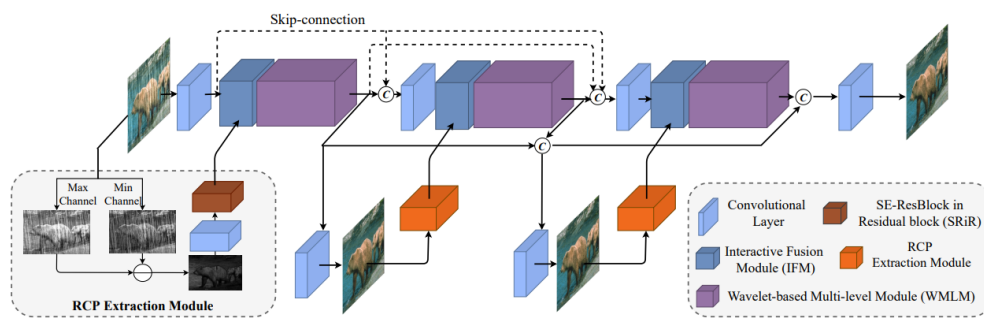


รูปที่ 2.22 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น

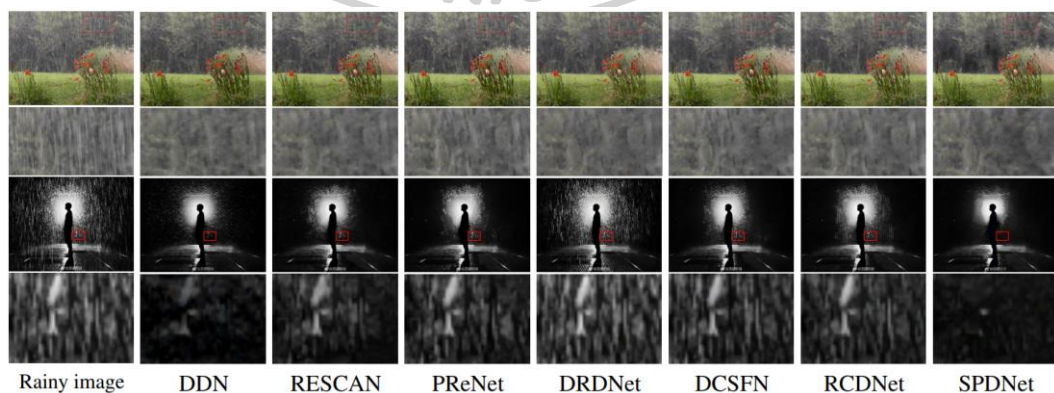


2.11.7 Structure-Preserving Deraining with Residue Channel Prior Guidance(Qiaosi Yi, Juncheng Li, Qinyan Dai, Faming Fang, Guixu Zhang and Tiejiong Zeng) [20]

ในวิจัยนี้ได้นำเสนอวิธีลบฝนที่เรียกว่า SPDNet เป็นการวนซ้ำผลลัพธ์ไปที่ละชั้นตอน เพื่อให้ได้ผลลัพธ์ที่มีการลบฝนที่ดีที่สุด มีส่วนสำคัญอยู่สองส่วนคือ Residue Channel Prior(RCP) ซึ่งเป็นการสกัดฝนออกในภาพทั้งหมด เหลือเพียงโครงสร้างที่แท้จริงของรายละเอียดพื้นหลังที่ไม่มีฝนปรากฏภายในภาพ ในส่วนสำคัญอีกส่วนหนึ่ง คือ Wavelet-based Feature Extraction Backbone(WMLM) ในวิธีการนี้ใช้วิธี discrete wavelet transform (DWT) และ Inverse DWT (IWT) แทนที่จะใช้การคอนโวลูชัน และดีคอนโวลูชันเนื่องจากวิธีการอื่นทั่วไปในการหาดำแหน่งฝนจำเป็นต้องมีการลดข้อมูลของภาพ ทำให้รายละเอียดของภาพลดน้อยลง ทำให้สามารถระบุตำแหน่งของฝนโดยที่ไม่รบกวนรายละเอียดพื้นหลัง



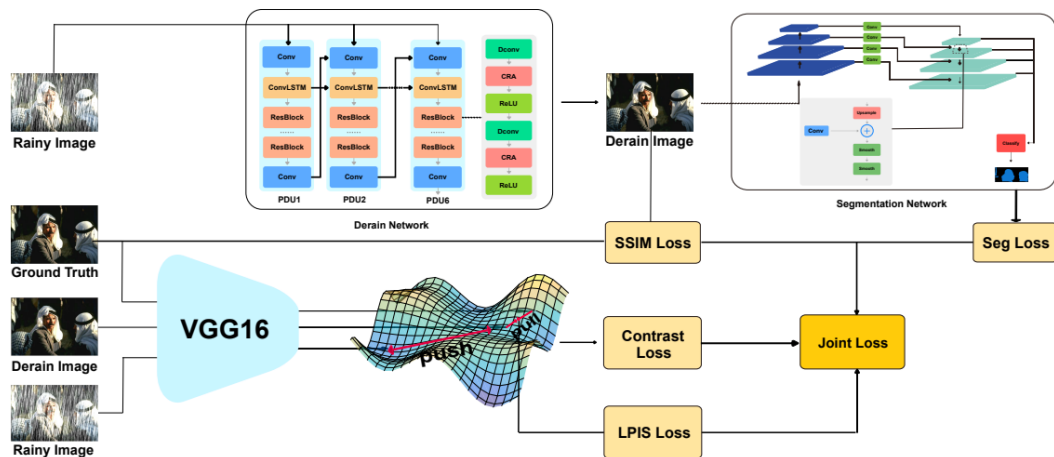
รูปที่ 2.23 โครงสร้างสถาปัตยกรรม SPDNet



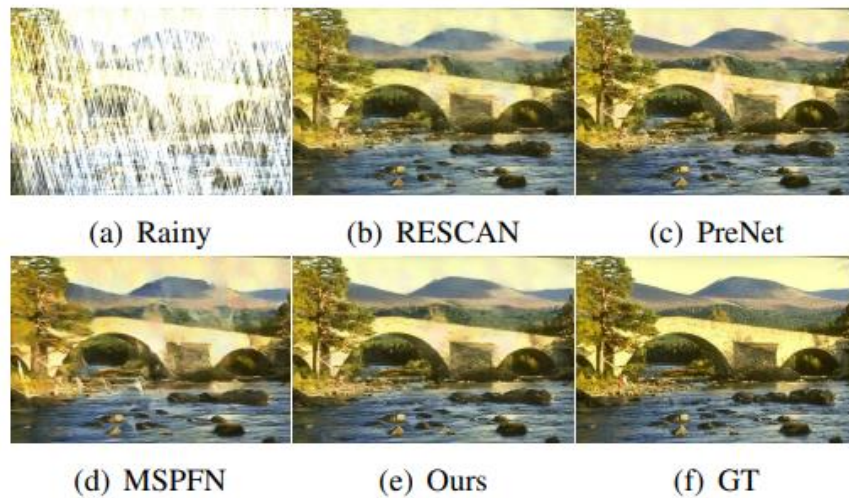
รูปที่ 2.24 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น

### 2.11.8 SAPNet: Segmentation-Aware Progressive Network for Perceptual Contrastive Deraining(Shen Zheng, Changjie Lu, Yuxiong Wu and Gaurav Gupta) [1]

ในวิจัยนี้มีการนำเสนอวิธีการลบฝนที่เรียกว่า SAPNet มีส่วนสำคัญ 3 ส่วน ในส่วนแรกเป็นส่วนของการลบฝน ซึ่งภายในประกอบไปด้วย โครงข่ายคอนโวลูชัน โครงข่ายการวนซ้ำ และ Resblock ซึ่งทำหน้าที่รับภาพฝนเข้ามาเพื่อทำการลบฝนออกทั้งหมด ในส่วนที่สองเป็นการหาค่าความสูญเสียของภาพฝน และภาพดั้งเดิม โดยใช้โมเดล VGG16 สำหรับหาค่าความสูญเสีย ในส่วนสุดท้ายเป็นการจำแนกส่วนต่างๆภายในภาพ ซึ่งมีการรับข้อมูลนำเข้าคือภาพฝนที่ถูกลบออกจากในกระบวนการแรก เพื่อเป็นการทดสอบประสิทธิภาพหลังจากการทำลบฝน เนื่องจากฝนนับเป็นสิ่งรบกวนหนึ่ง ที่ทำให้เกิดปัญหาการจำแนกส่วนในภาพ ดังนั้นในส่วนนี้จะทำการวัดประสิทธิภาพของการจำแนกส่วนในภาพเพื่อนำค่าความสูญเสียไปปรับปรุงภายในโครงข่าย SAPNet ต่อไป โดยการทดสอบประสิทธิภาพจะใช้ PSNR และ SSIM โดยที่ใช้ฐานข้อมูลภาพฝนที่มีความหนาแน่นต่ำ และความหนาแน่นสูง เทียบกับวิธีการอื่นพบว่าในวิธีนี้การได้ผลที่ดีกับฝนที่มีความหนาแน่นสูง แต่ในฝนที่มีความหนาแน่นต่ำทำได้มีหากเทียบกับวิธีการอื่น



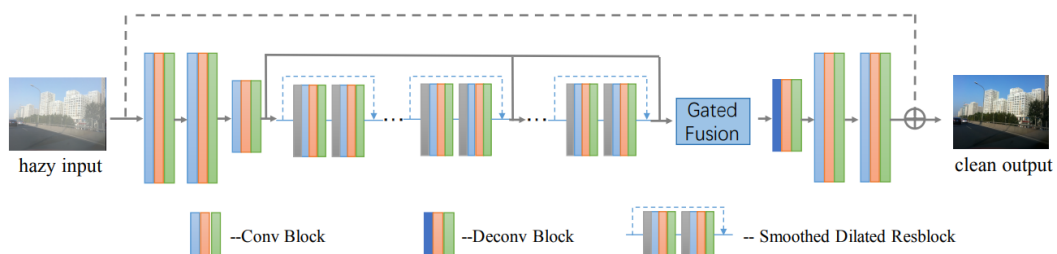
รูปที่ 2.25 โครงสร้างสถาปัตยกรรม SAPNet



รูปที่ 2.26 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น

2.11.9 Gated Context Aggregation Network for Image Dehazing and Deraining (Dongdong Chen, Mingming He, Qingnan Fan, et al.) [22]

ในวิจัยนี้มีการนำเสนอวิธีการที่เรียกว่า GCANet ซึ่งสามารถลบได้ทั้งฝน และหมอกในภาพได้ ซึ่งมีโครงสร้างหลักเป็น Autoencoder ซึ่งประกอบไปด้วยโครงข่ายคอนโวลูชัน 3 บล็อก สำหรับการเข้ารหัส ในส่วนของการถอดรหัสจะใช้โครงข่ายดีคอนโวลูชัน 1 บล็อก และโครงข่ายคอนโวลูชัน 1 บล็อก โดยระหว่าง Encoder กับ Decoder จะมีการแทรก Smoothed Dilated Resblock หลายบล็อก เพื่อเก็บข้อมูลในระหว่างการฝึกสอนที่สำคัญ หลังจากนั้นจะมีการใช้ Gated Fusion เพื่อรวบรวมข้อมูลในแต่ละบล็อกก่อนหน้า แล้วนำไปรวมกันเพื่อส่งต่อไปให้ในส่วน Decoder แล้วจึงออกมาเป็นภาพผลลัพธ์ของการลบหมอกหรือลบฝน



รูปที่ 2.27 โครงสร้างสถาปัตยกรรม GCANet



รูปที่ 2.28 เปรียบเทียบผลลัพธ์การลบฝนกับวิธีอื่น



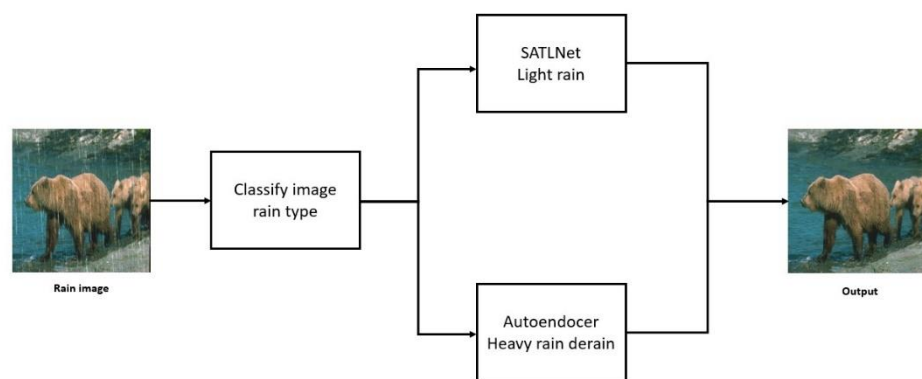
### บทที่ 3

#### วิธีดำเนินการวิจัย

ในส่วนของวิธีการดำเนินการวิจัยในวิทยานิพนธ์เล่มนี้ได้ออกแบบและพัฒนาโปรแกรมสำหรับการลบฝนออกจากภาพ โดยได้ทำการศึกษาการคัดแยกระดับของฝนที่มีลักษณะเบา และฝนที่มีลักษณะหนัก เนื่องจากการทดลองพบว่าการแยกฝึกสอนโดยแบ่งเป็นแต่ละโมเดล สามารถแก้ปัญหาในการลบฝนได้มากกว่าการฝึกสอนโดยใช้ข้อมูลภาพฝนที่ปนกันทั้งฝนเบา และฝนหนัก ในเบื้องต้นจึงได้มีการศึกษาวิธีเพิ่มประสิทธิภาพของการคัดแยกฝนก่อนเพื่อที่ได้ผลลัพธ์ที่ดีที่สุดก่อนเข้าสู่กระบวนการลบฝนออกจากภาพ ในส่วนของกระบวนการลบฝน ได้มีการศึกษาเกี่ยวกับชนิดและวิธีลบรอยสีในรูปแบบต่างๆ เพื่อนำมาพัฒนาวิธีการลบฝน พบว่าการใช้ Autoencoder ในการลบรอยสีเป็นที่นิยมอย่างมากในหลายงานวิจัย ในงานวิจัยนี้จึงได้เลือก Autoencoder เป็นส่วนสำคัญของการลบฝนออกจากภาพ

ในบทนี้กล่าวถึงขั้นตอนการดำเนินการวิจัย และอธิบายถึงโครงสร้างของกระบวนการลบฝน โดยในลำดับแรกจะเล่าถึงภาพรวมของกระบวนการลบฝนตั้งแต่เริ่มรับภาพเข้ามา หลังจากนั้นจึงอธิบายเกี่ยวกับการตรวจปริมาณของฝนเพื่อคัดแยกส่งไปให้กับขั้นตอนการลบฝนต่อไป โดยในส่วนของการลบฝนจะแบ่งเป็นสองส่วนด้วยกันคือ ลบฝนเบา และลบฝนหนักที่มีกระบวนการลบฝนที่ต่างกัน

#### 3.1 ภาพรวมของวิธีการ



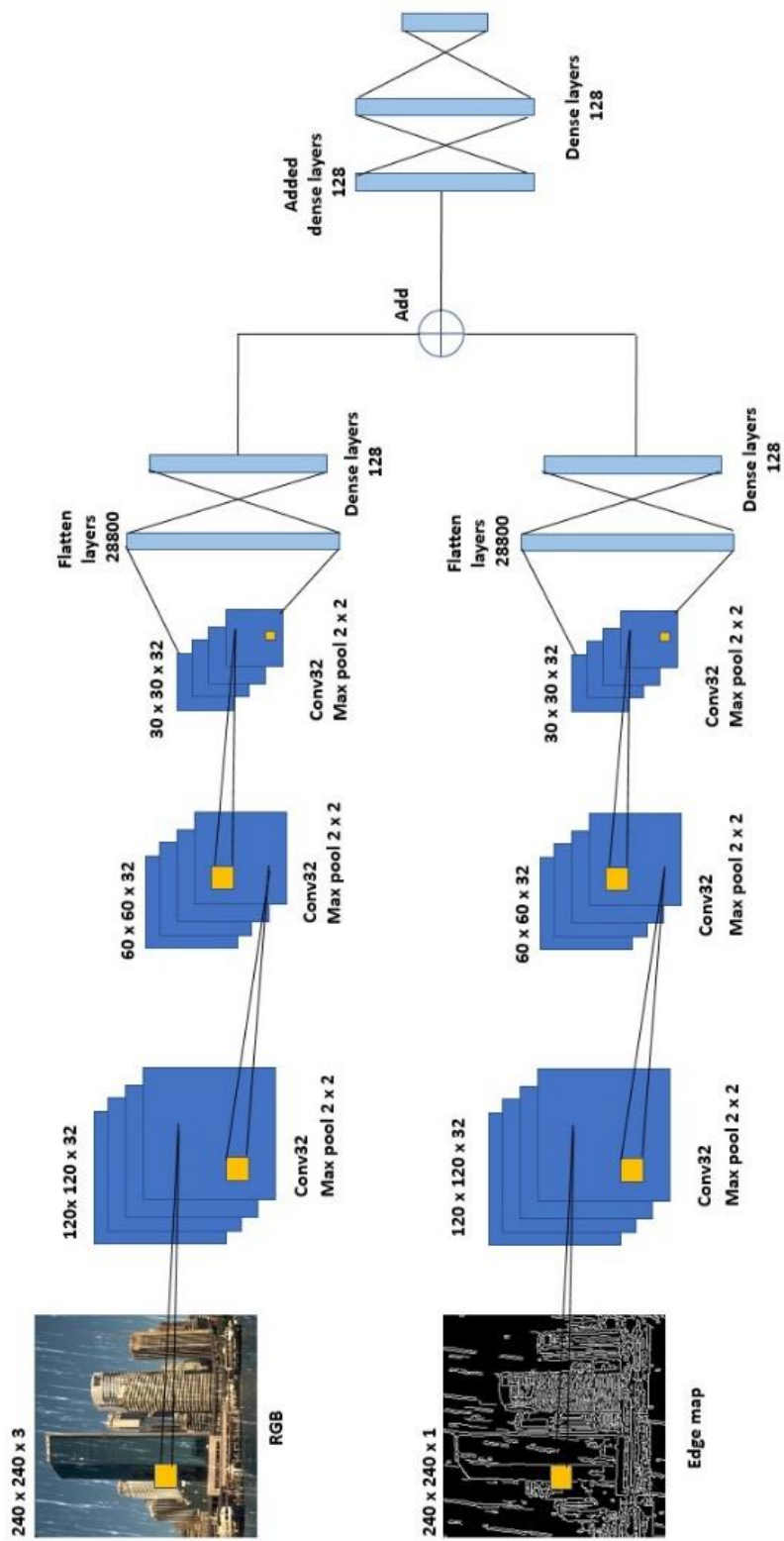
รูปที่ 3.1 ภาพรวมของขั้นตอนการลบฝน

ในส่วนแรกต้องรับภาพอินพุตเข้ามาก่อน ซึ่งเป็นภาพที่ต้องการลบฝนออก โดยภาพจะถูกส่งไปยังขั้นตอนตัดแยกฝนก่อน โดยในงานวิจัยนี้ได้แบ่งระดับฝนออกเป็นสองระดับ คือฝนเบา และฝนหนัก ซึ่งกระบวนการลบฝนของงานวิจัยนี้จะแบ่งออกเป็นสองส่วนคือ การลบฝนเบา และฝนหนัก จึงเป็นสาเหตุสำคัญที่ต้องทำการตัดแยกฝนก่อนที่จะเข้าสู่กระบวนการลบฝน โดยกระบวนการลบฝนเบา นั้นจะใช้วิธี A Self-Augmentation Transfer Learning Network for Image Deraining (SATLNet) ซึ่งถูกออกแบบให้ลบเฉพาะฝนเบาเท่านั้น แต่ส่วนสำคัญหลักของ SATLNet ยังคงเป็น Autoencoder เพียงแต่ได้ออกแบบโครงสร้างให้มีประสิทธิภาพการลบฝนให้ดีกว่าการใช้เพียง Autoencoder อย่างเดียว ในส่วนของการลบฝนหนักนั้นเป็นการใช้โครงสร้าง Autoencoder เพียงอย่างเดียว เมื่อผ่านสองกระบวนการนี้แล้วจะได้ผลลัพธ์เป็นภาพที่ถูกลบฝนออกดังรูป 3.1

### 3.2 การตรวจจับวัตถุโดยใช้ Fusion model ร่วมกับ CNN

จากการทดลองการลบฝนโดยแยกโมเดลที่ฝึกสอนจากฝนเบา และโมเดลที่ถูกฝึกสอนจากฝนหนักนั้น มีประสิทธิภาพมากกว่าโมเดลที่ถูกฝึกสอนรวมกัน การแยกภาพฝนก่อนเพื่อตัดแยกภาพให้เข้าสู่กระบวนการลบฝนโดยเฉพาะจึงช่วยให้ประสิทธิภาพดีขึ้น การตัดแยกฝนก่อนจึงเป็นส่วนสำคัญของงานวิจัยนี้

ในส่วนของโครงสร้างการตัดแยกฝนได้มีการใช้ CNN เป็นส่วนหลักในการตัดแยกฝน แต่เนื่องจากประสิทธิภาพการตัดแยกฝนที่ใช้เพียง CNN ประสิทธิภาพจะลดลงหากฐานข้อมูลที่น่าสนใจมีลักษณะที่ไม่ใกล้เคียงกับภาพข้อมูลนำเข้า ดังนั้นจึงมีการใช้เทคนิค Fusion model เพื่อเพิ่มประสิทธิภาพการตัดแยกฝนให้ดีขึ้น โดยมีหน้าที่นำโมเดลที่ถูกฝึกสอนทั้งสองโมเดลมารวมกัน ในงานวิจัยนี้จะใช้โมเดลที่ฝึกสอนโดยใช้ภาพฝน RGB กับโมเดลที่ฝึกสอนโดยใช้ภาพขอบ สาเหตุที่ต้องใช้ภาพขอบเข้ามาช่วยในการลบฝนนั้น เนื่องจากว่าการลดรายละเอียดที่ไม่จำเป็นออกเหลือเพียงแค่เส้นของฝนมีลักษณะเด่นมากกว่าเม็ดฝนที่เป็นภาพสี การนำโมเดลมารวมกันจึงสามารถช่วยเพิ่มประสิทธิภาพของการลบฝนได้ โดยมีโครงสร้างดังรูปต่อไปนี้



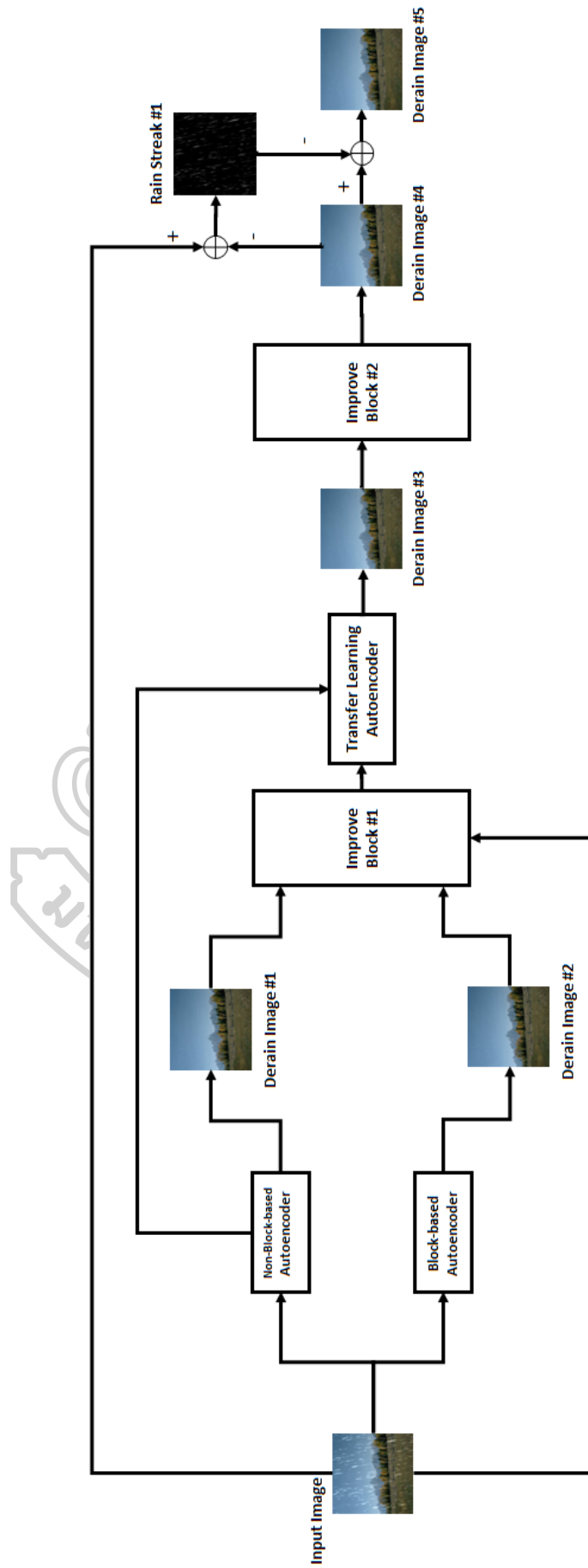
รูป 3.2 โครงสร้างของการจำแนกเฟส

จากรูปที่ 3.2 จะสังเกตเห็นว่ามีการใช้ CNN อยู่สองส่วน ซึ่งเป็นเทคนิคของ Fusion model ที่จะนำค่าในส่วนของ Dense ปลายทางของ CNN model มารวมกันเพื่อทำนายชนิดของฝน โดยเริ่มต้นของโครงสร้างนี้ จำเป็นต้องใช้ภาพที่นำมาเรียนรู้สองแบบคือ ภาพ RGB และภาพ Edge map ในส่วนของภาพ Edge map นั้นในงานวิจัยนี้ได้เลือกวิธีการหาภาพขอบแบบ Canny ซึ่งเป็นวิธีการหาภาพขอบที่สามารถแสดงรายละเอียดได้ของเส้นฝนได้ชัดเจนมากที่สุด ทำให้สามารถช่วยในการตัดแยกฝนได้เป็นอย่างดี โดยในส่วนของ CNN นั้นจะใช้ชั้นคอนโวลูชันทั้งหมด 3 ชั้น มี kernel เท่ากับ  $3 \times 3$  แต่ละชั้นจะมีการใช้ max pooling ขนาด  $2 \times 2$  โดยในส่วนสุดท้ายของแบบจำลองทั้งสองมีการทำ Flatten layer เพื่อจัดเรียงข้อมูลให้อยู่ในมิติเดียว แล้วจึงลดขนาด output ที่เหลือ 128 ในชั้น Dense layers หลังจากนั้นนำ Dense layers ทั้งสองผสานกัน โดยใช้วิธีการบวกค่าของโมเดลที่เป็นภาพ RGB กับโมเดลที่เป็นภาพขอบ ทำให้ได้ Dense layers ใหม่ที่ถูกรวมเข้ากันของทั้งสองแบบจำลอง โดยก่อนที่จะถึง outputs layers ได้เพิ่ม Dense layer ขนาด 128 อีกชั้นหนึ่ง เพื่อเพิ่มประสิทธิภาพของโครงสร้าง โดยในส่วนของ output นั้นมีการทำนายเพียงสองค่า คือ ฝนเบา และฝนหนัก

### 3.3 SATLNet สำหรับกรณีลบฝนเบา

ในส่วนนี้ เป็นการอธิบาย SATLNet แสดงรูปที่ 3.3 โดยประกอบด้วย 3 ส่วนสำคัญ คือ Autoencoder , Improve block และ Transfer learning โดยในส่วนแรกของ SATLNet เป็นการลบฝนเบื้องต้นโดยใช้ Autoencoder สองแบบ ได้แก่ Non Block-based autoencoder และ Block-based Autoencoder สาเหตุที่ต้องใช้ autoencoder 2 แบบ เนื่องจากผลลัพธ์ของการลบฝนมีข้อดีและข้อเสียที่แตกต่างกัน ในส่วนของ autoencoder แบบแรก จะมุ่งเน้นไปที่การลบฝนอย่างชัดเจน แต่รายละเอียดพื้นหลังถูกลดลง ในทางกลับกัน autoencoder แบบที่สอง ช่วยในเรื่องของการคงสภาพพื้นหลังแต่การลบฝนยังไม่ดีพอ ดังนั้นผลลัพธ์จากทั้งสองแบบจะเข้าสู่ Improve block#1 ที่ถูกสร้างขึ้นเพื่อเพิ่มประสิทธิภาพการลบฝนให้ดีขึ้นจากการนำข้อดีของผลลัพธ์การลบฝนทั้งสองแบบก่อนหน้า ผลลัพธ์ที่ได้จะถูกนำไปสกัดหาเม็ดฝนเพื่อนำไปเทรนต่อในส่วนของ Transfer learning Autoencoder ผลลัพธ์ที่ได้จะนำไปผ่าน Improve block#2 ซึ่งมีรายละเอียดที่แตกต่างกับ Improve block#1 เพียงเล็กน้อย ซึ่งมีการอธิบายอีกครั้งภายหลัง ในส่วนท้ายสุดของ SATLNet เป็นการเพิ่มประสิทธิภาพของการลบฝน โดยนำผลลัพธ์ที่ได้ลบกับภาพอินพุต เพื่อหาเม็ดฝนของภาพอินพุต แล้วนำไปลบกับภาพผลลัพธ์ก่อนหน้าอีกครั้ง วิธีนี้จะช่วยให้รายละเอียดพื้นหลังของภาพบางส่วนที่หายไปในช่วงตอนก่อนหน้ากลับมาได้

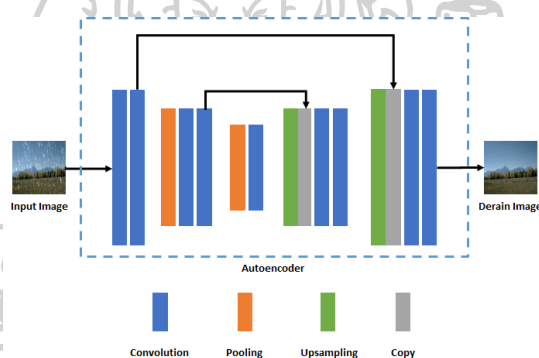




รูป 3.3 โครงสร้างของ SATLNet

### 3.3.1 Non Block-based Autoencoder

Autoencoder เป็น Neural Network ชนิดหนึ่งที่มีการออกแบบให้เรียนรู้ภาพอินพุตที่รับเข้ามา ให้เหมือนกับภาพเอาต์พุต จึงมีการนำไปใช้กับการลบ Noise ต่างๆ งานวิจัยนี้จึงใช้ประโยชน์จาก Autoencoder ในการลบฝนออกจากภาพ โดยโครงสร้างของ Autoencoder ที่นำมาใช้สำหรับการลบฝน แสดงรูปที่ 3.4 มีจำนวนทั้งหมด 15 ชั้น โดยใน 5 ชั้นแรกเป็นส่วนของ Encoder ที่มี Convolution จำนวน 5 ชั้น ต่อมาเป็นส่วนของ Latent layer ประกอบไปด้วย 2 ชั้น แล้วจึงเข้าสู่ Decoder ที่ประกอบไปด้วย Convolution ทั้งหมด 7 ชั้น โดยในบางชั้นจะมีการ Skip connection แบบ Concatenate กันระหว่างในส่วนของ Encoder กับ Decoder เพื่อป้องกันการสูญหายของข้อมูลระหว่างทาง โดยโมเดลนี้จะมีการใช้ activation แบบ Relu ทั้งหมด

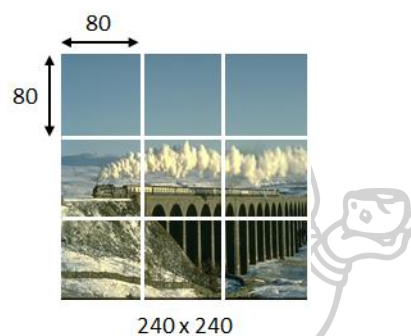


รูปที่ 3.4 โครงสร้าง Autoencoder ที่ใช้สำหรับลบฝน

### 3.3.2 Block-based Autoencoder

การทำ Block-base เป็นวิธีหนึ่งที่ช่วยฟื้นฟูรายละเอียดพื้นหลังของภาพ โดยเป็นการแบ่งภาพเดิมออกเป็นส่วนๆ ที่มีขนาดเท่ากัน แสดงรูปที่ 3.5 ในงานวิจัยนี้ ใช้ภาพขนาด 240x240 แบ่งออกเป็น 9 ภาพที่มีขนาดภาพละ 80x80 สาเหตุที่ต้องใช้ขนาดภาพ 80x80 เนื่องจากเม็ดฝนเป็น noise ที่มีขนาดใหญ่ หากใช้ขนาดภาพที่เล็ก รูปร่างของเม็ดฝนจะถูกแยกจนมีลักษณะที่ไม่ชัดเจน อีกทั้งอาจเกิด blocking artifacts จำนวนมาก จากการทดสอบจึงเลือกใช้การแบ่งภาพที่มีขนาด 80\*80 ซึ่งเป็นขนาดที่เหมาะสมกับภาพอินพุตที่รับเข้ามา โดยโครงสร้างของ Block-based Autoencoder จะเหมือนกับ Non Block-based Autoencoder ซึ่งผลลัพธ์การลบฝนที่ได้สามารถลบฝนได้ดีในระดับหนึ่ง แต่รายละเอียดพื้นหลังทำได้ดีกว่าแบบ Autoencoder

โดยสาเหตุการใช้วิธีการลบฝนแบบ Non-block-based และ Block-based เนื่องจากว่า ผลลัพธ์การลบฝนทั้งสองแบบมีข้อดี และข้อเสียที่ต่างกัน การลบฝนแบบแรกนั้น จะเน้นการลบฝนที่ยังมีสภาพของพื้นหลังค่อนข้างสมบูรณ์ แต่ฝนที่ได้ไม่สามารถลบออกได้ทั้งหมดเนื่องจากต้องมีการฝึกสอนให้โมเดลสามารถลบฝนที่มีลักษณะที่แตกต่างกันภายในภาพ ทำให้ประสิทธิภาพการลบฝนนั้นลดลง ดังนั้นจึงมีการใช้วิธีการลบฝนแบบ Block-based ที่จะแบ่งภาพออกเป็นส่วนๆ เพื่อให้การฝึกสอนสามารถจดจำรายละเอียดของฝนได้ดีขึ้น ทำให้สามารถลบฝนได้ดีกว่าวิธีการแรก แต่เนื่องจากการทำวิธีการนี้มีปัญหาในขั้นตอนการรวมภาพกลับมา ซึ่งจะเห็นรอยต่อของภาพ จึงทำให้เกิดกระบวนการต่อไปคือ Improve Block



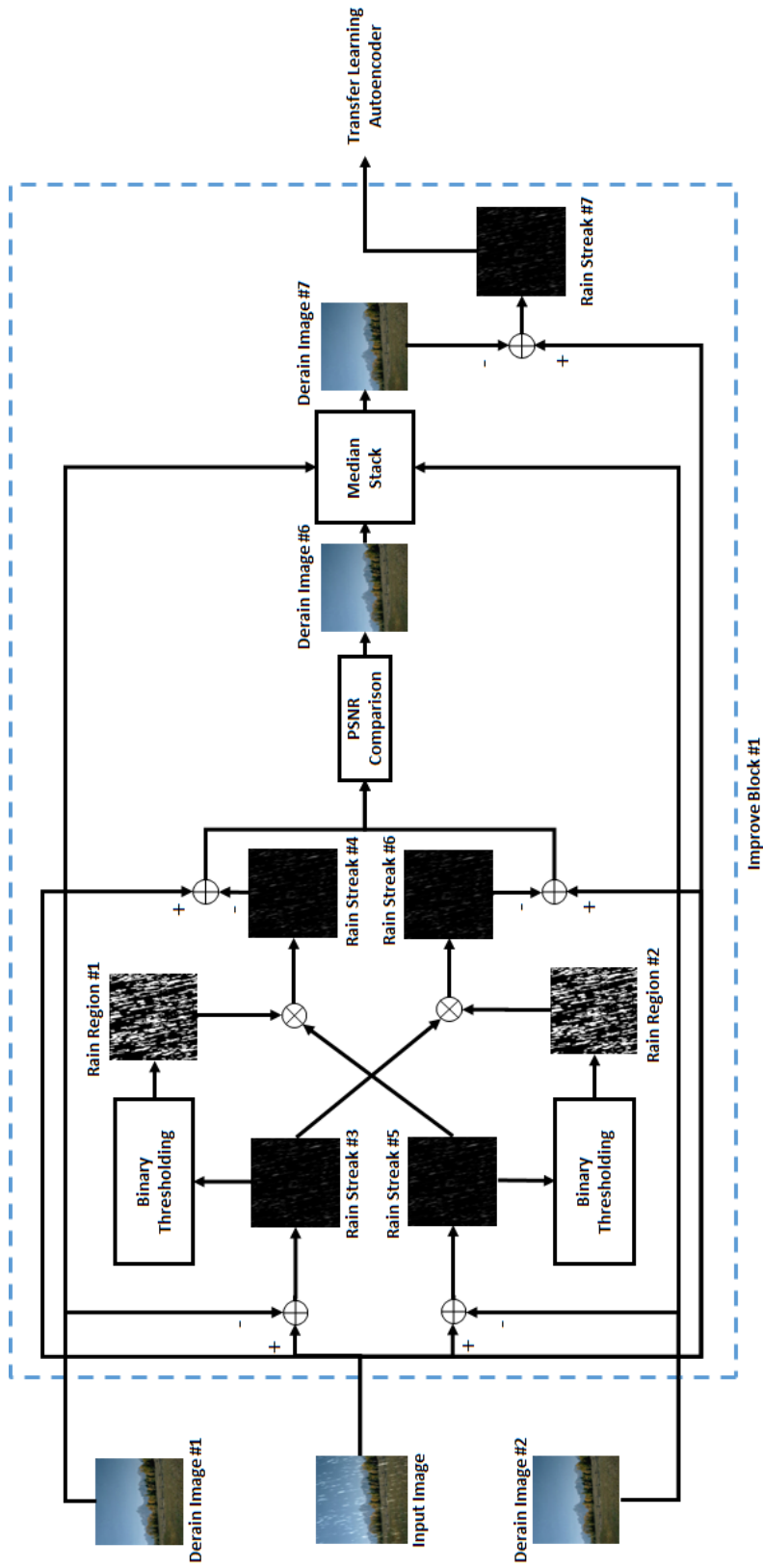
รูปที่ 3.5 การแบ่งส่วนภาพสำหรับการฝึกสอน

### 3.3.3 Improve Block

ในงานวิจัยนี้จะมีการใช้ Improve Block อยู่ 2 ช่วง ซึ่งมีโครงสร้างแตกต่างกันเล็กน้อย โดยในช่วงแรกจะเรียกว่า Improve Block #1 แสดงรูปที่ 3.6 เป็นส่วนที่ช่วยในเรื่องของการดึงจุดเด่นของภาพผลลัพธ์จาก Non Block-based Autoencoder และ Block-based Autoencoder โดยจะเป็นสลับคุณลักษณะของเม็ดฝนของแต่ละวิธีการ และเปรียบเทียบประสิทธิภาพเพื่อส่งผลลัพธ์ที่ดีที่สุดไปยังขั้นตอนต่อไป โดย Improve block # 1 มีการรับอินพุตมาทั้งหมด 3 ภาพ คือ ภาพอินพุต Derain Image #1 เป็นภาพการลบฝนจากautoencoderแบบธรรมดา และDerain Image #2 ที่ผ่านกระบวนการblock based ต่อมาต้องทำการหาภาพเม็ดฝนของทั้งสองภาพ โดยการนำไปลบกับภาพต้นฉบับ หลังจากนั้นนำไปผ่าน Binary thresholding เพื่อให้ได้ภาพเม็ดฝนที่เป็น Binary เรียกว่า Rain Region ในขั้นตอนนี้เราจะได้ Rain Region มาสองภาพ ได้แก่ Rain Region #1 และ Rain Region #2 ซึ่งทั้งสองภาพจะถูกนำไป bitwise and กับภาพเม็ดฝนอีกครั้ง โดย Rain Region #1 ถูกนำไป and กับภาพ Rain Streak #5 และ Rain Region #2 ถูกนำไป and กับภาพ Rain Streak #3 เนื่องจากในบางภาพของ autoencoder แบบธรรมดา กับแบบ Block based มีรายละเอียด และลักษณะของการลบฝนที่ต่างกัน จึงใช้วิธีการนำ Rain region ไป and กับ Rain

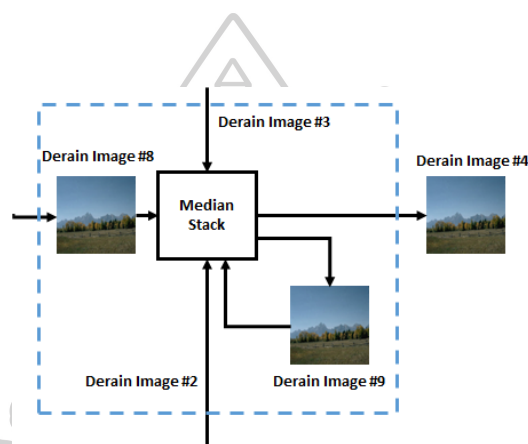
Streak เพื่อเป็นการคงตำแหน่งของฝนในแต่ละวิธีการไว้ และกำจัดรายละเอียดพื้นหลังที่ไม่จำเป็นออก หลังจากนั้นจะนำ Rain Streak ทั้งสองภาพที่ได้มาใหม่ย้อนกลับไปลบกับภาพฝนต้นฉบับเพื่อให้ได้ภาพที่ถูกลบฝน โดยภาพลบฝนจะถูกนำเข้าไปที่ PSNR Comparison เพื่อนำไปเปรียบเทียบประสิทธิภาพว่าภาพไหนมีประสิทธิภาพดีกว่าจะถูกส่งเป็น Output ต่อไปให้กับ Median Stack ซึ่งเป็นการหาค่ากลางของพิกเซลที่ถูกซ้อนกันอยู่ระหว่างภาพ Derain Image #1 ,#2 ,#6 และภาพฝนต้นฉบับ เพื่อเพิ่มประสิทธิภาพของการลบฝน สาเหตุการทำ Median Stack เนื่องจากว่าการลบฝนในแต่ละขั้นตอนก่อนหน้า มีการลดรายละเอียดพื้นหลังอย่างชัดเจน จึงทำให้ต้องใช้ภาพฝนต้นฉบับเพื่อฟื้นฟูรายละเอียดพื้นหลังขึ้นมาผ่านการทำ Median Stack หลังจากนั้นนำไปลบกับภาพฝนต้นฉบับเพื่อหาเม็ดฝนดังรูป Rain Streak #7 เพื่อส่งให้กับ Transfer Learning Autoencoder การทำฝึกสอนรูปเฉพาะเม็ดฝนที่ต้องการ





รูป 3.6 โครงสร้างของ Improve block

ในส่วนที่สองของการใช้ Improve Block เรียกว่า Improve block#2 แสดงรูปที่ 3.7 แตกต่างกับImprove block#1 เพียงเล็กน้อย โดยจะเป็นการเปลี่ยนภาพอินพุตจาก Derain Image #1 ซึ่งเป็นผลลัพธ์การลบฝนที่ผ่านAutoencoder ให้กลายเป็น Derain Image #3 ซึ่งเป็นผลลัพธ์จากการทำ Transfer learning โดยผลลัพธ์ที่ได้จากการทำ PSNR Comparison เป็นดังรูป Derain Image #8 นำมาทำ Median Stack กับ Derain Image #3 และ Derain Image #2 ที่เป็นผลลัพธ์ของการลบฝนที่ผ่าน Autoencoder แบบ Block based จะได้ภาพการลบฝนดังรูป Derain Image #9 แล้วนำมาซ้อนกลับมาทำ Median Stack อีกครั้ง จะได้ผลลัพธ์การลบฝนดังรูป Derain Image #4



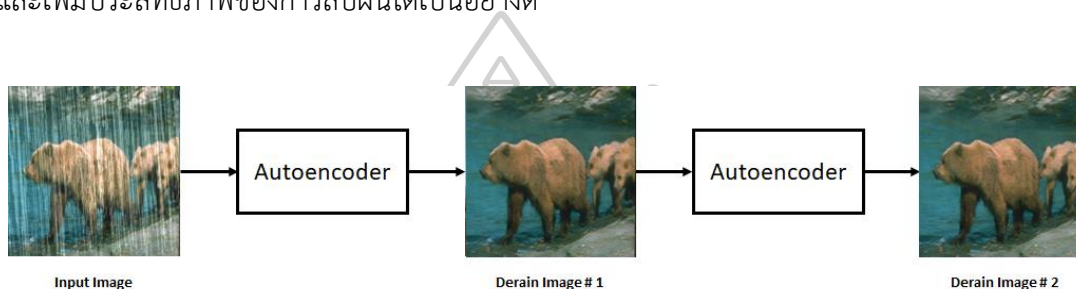
รูปที่ 3.7 ส่วนสุดท้ายของ Improve block # 2

### 3.3.4 Transfer Learning Autoencoder

การทำ Transfer learning เป็นการเรียนรู้จากโมเดลที่มีก่อนอยู่หน้ามาปรับใช้ให้เข้ากับข้อมูลที่ต้องการ โดยในส่วนนี้มีจุดประสงค์ต้องการฝึกสอนฐานข้อมูลที่มีเฉพาะภาพฝนที่มีลักษณะของฝนเหมือนกับภาพอินพุตที่นำมาทดสอบ เนื่องจากว่าการฝึกสอนเฉพาะลักษณะของฝนที่มีความใกล้เคียงกันกับภาพที่นำมาทดสอบ สามารถเพิ่มประสิทธิภาพของการลบฝนได้ดี โดยเริ่มแรกนำ weight จาก Model ที่ถูกเทรนจาก Non Block-based Autoencoder มาทำการ Transfer learning โดย Dataset ที่นำมาเทรน มาจากการทำ augmentation โดยใช้เม็ดฝนที่ถูกสกัดจากผลลัพธ์ในขั้นตอน Improve Block # 2 ทำให้ Dataset ที่นำมาใช้มีลักษณะเม็ดฝนที่เหมือนกับภาพอินพุตที่นำมาทดสอบ ทำให้ประสิทธิภาพของการลบฝนดีขึ้น

### 3.4 Autoencoder สำหรับกรณีลบฝนหนัก

ในกรณีของการลบฝนหนัก มีการใช้ Autoencoder ที่มีโครงสร้างแบบเดียวกับ Non Block-based Autoencoder ดังรูป 3.8 โดยเริ่มต้นจะมีการรับข้อมูลรูปฝนหนักเพื่อนำไปฝึกสอน โดยผ่าน Autoencoder ในขั้นแรกก่อน หลังจากได้ผลลัพธ์ภาพที่ลบฝนแล้ว จึงนำกลับมาฝึกสอนอีกรอบผ่าน Autoencoder ที่มีโครงสร้างเดิม เนื่องจากลักษณะของฝนหนักนั้นมีความทึบ และมีความถี่มากกว่าฝนเบา จึงเป็นสาเหตุที่ต้องมีการลบฝนถึงสองครั้ง โดยฝนครั้งแรกของการลบฝนสามารถลบได้บางส่วน และยังมีเหลือร่องรอยของการลบฝน แต่ในกรณีที่มีการลบฝนอีกครั้ง สามารถช่วยฟื้นฟูและเพิ่มประสิทธิภาพของการลบฝนได้เป็นอย่างดี



รูปที่ 3.8 ขั้นตอนการลบฝนหนัก



## บทที่ 4

### ผลการทดลองของงานวิจัย

การทดลองในงานวิจัยนี้ได้ใช้บริการ Cloud computing จาก Google ชื่อว่า Google Colaboratory

#### 4.1 ชุดข้อมูลภาพที่ใช้ในการทดลอง

##### 4.1.1 ชุดข้อมูลที่ใช้สำหรับการจำแนกฝนเบา และฝนหนัก

สำหรับการทดลองและการฝึกสอนการจำแนกฝนเบาใช้ฐานข้อมูลภาพชื่อว่า Rain100L ซึ่งเป็นภาพฝนที่ถูกสังเคราะห์ขึ้นมา โดยเม็ดฝนจะมีลักษณะที่แยกออกจากกันอย่างชัดเจน โดยมีขนาดรูป  $240 \times 240$  พิกเซล จำนวนทั้งหมด 300 ภาพ โดยฐานข้อมูล Rain100L นั้นได้มีการแบ่งภาพสำหรับฝึกสอน และภาพสำหรับทดสอบไว้แล้ว โดยมีการแบ่งสำหรับฝึกสอนทั้งหมด 200 ภาพ และ 100 ภาพสำหรับทดสอบประสิทธิภาพ ภาพดังตัวอย่างรูปที่ 4.1 และ รูปที่ 4.2



รูปที่ 4.1 ตัวอย่างภาพฝนไว้สำหรับฝึกสอนในการลบฝนเบา





รูปที่ 4.2 ตัวอย่างภาพฝนไว้สำหรับการทดสอบในการลบฝน

ในส่วนของฝนหนักจะใช้ฐานข้อมูลที่ชื่อว่า Rain100H เป็นฐานข้อมูลที่สังเคราะห์ฝนขึ้นมา เหมือนกับฐานข้อมูลภาพก่อนหน้า โดยจะมีลักษณะฝนที่มีความถี่ และอยู่ติดกันอย่างมาก ซึ่งฐานข้อมูลนี้มีการแบ่งไว้สำหรับฝึกสอนจำนวน 1087 รูป แต่ในขั้นตอนการแยกฝนจะเลือกใช้เพียง 350 รูปเท่านั้น และสำหรับภาพทดสอบมีจำนวน 100 รูป แสดงตัวอย่างดังรูป 4.3 และ 4.4



รูปที่ 4.3 ตัวอย่างภาพฝนไว้สำหรับฝึกสอนในการลบฝนหนัก



รูปที่ 4.4 ตัวอย่างภาพฝนไว้สำหรับการทดสอบในการลบฝนหนัก

#### 4.1.2 ชุดข้อมูลไว้สำหรับการลบฝนออกจากภาพ

ข้อมูลภาพที่ใช้กับการลบฝนจากภาพใช้ข้อมูลชุดเดียวกันกับการจำแนกภาพ คือ Rain100L และ Rain100H แต่ในส่วนของการทดสอบฝนเราได้มีการเพิ่มภาพจำนวน 50 ภาพในการทดสอบ ซึ่งนำมาจากฐานข้อมูล [7] เป็นรูปฝนที่มีลักษณะค่อนข้างบาง และถี่มากกว่าฐานข้อมูลฝนเบสฐานแรก ภายในฐานข้อมูลจะมี Ground truth เพื่อวัดประสิทธิภาพของการลบของฝน แสดงตัวอย่างดังรูป 4.5



Rain

Ground truth

รูปที่ 4.5 ตัวอย่างเปรียบเทียบภาพฝนกับ Ground truth ไว้สำหรับการทดสอบฝนเบาที่เพิ่มเข้ามา

ฐานข้อมูลที่น่ามาใช้สำหรับการลบฝนเบาเป็นฐานข้อมูลเดียวกันกับฐานข้อมูลที่ไว้จำแนกฝน แต่จะมีการใช้ภาพที่เป็น Ground truth ไว้สำหรับฝึกสอนจำนวน 200 รูป ซึ่งจะเป็นภาพพื้นหลังที่ไม่มีฝน ในส่วนของภาพฝนจะใช้การสังเคราะห์ฝน และนำมาใส่ใน Ground truth จำนวน 200 รูป ในส่วนของการทดสอบจะใช้ภาพจำนวน 100 รูป จากฐานข้อมูลเดิม โดยจะมีทั้ง Ground truth และภาพฝน โดยภาพไว้สำหรับฝึกสอน และสำหรับทดสอบแสดงตัวอย่างดังรูป 4.6 และ 4.7 ในส่วนของการลบฝนหนักนั้นใช้ฐานข้อมูลเดียวกันกับการจำแนกฝน แต่จะใช้ฐานข้อมูลภาพทั้งหมดแทนจำนวน 1087 รูป และส่วนของการทดสอบจะใช้ภาพทั้งหมด 100 รูป แสดงตัวอย่างดังรูป 4.8 และ 4.9



รูปที่ 4.6 ตัวอย่างเปรียบเทียบภาพฝนกับ Ground truth ไว้สำหรับการฝึกสอนในฐานข้อมูล

Rain100L



รูปที่ 4.7 ตัวอย่างเปรียบเทียบภาพฝนกับ Ground truth ไว้สำหรับการทดสอบในฐานข้อมูล Rain100L



รูปที่ 4.8 ตัวอย่างเปรียบเทียบภาพฝนกับ Ground truth ไว้สำหรับการฝึกสอนในฐานข้อมูล Rain100H



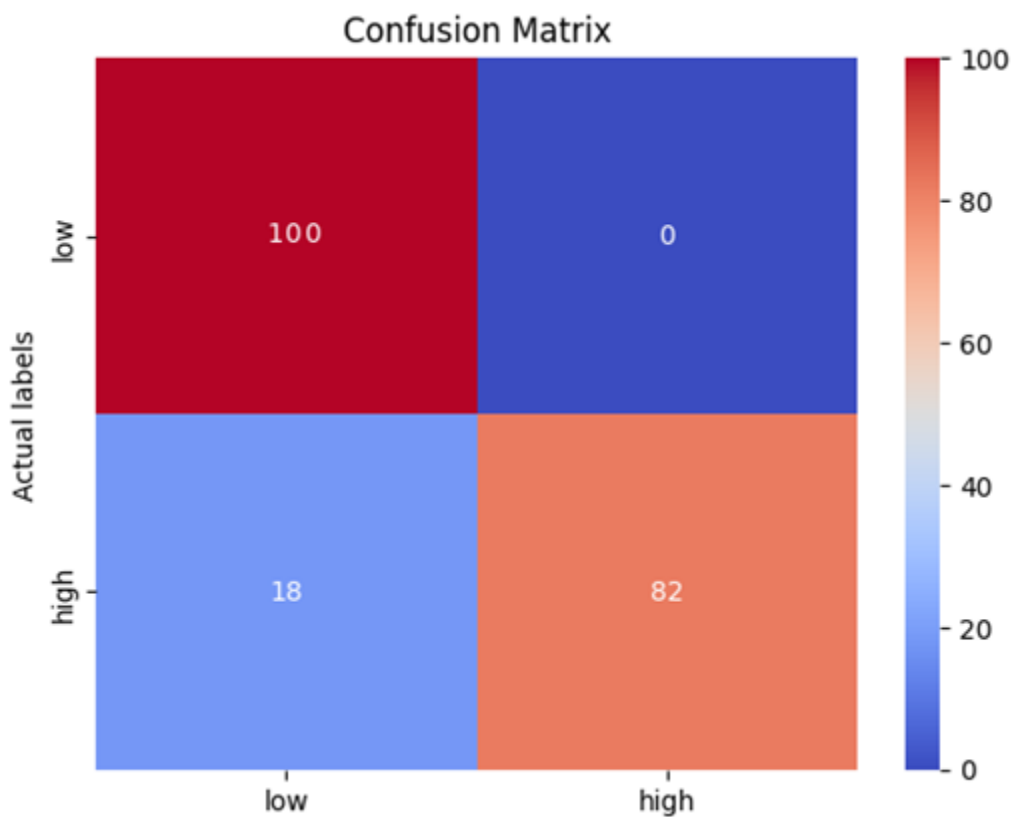
รูปที่ 4.9 ตัวอย่างเปรียบเทียบภาพฝนกับ Ground truth ไว้สำหรับการทดสอบในฐานข้อมูล

Rain100H

#### 4.2 วัดประสิทธิภาพการจำแนกฝนเบา และฝนหนัก

ในการทดสอบความแม่นยำของการจำแนกฝนหนัก และฝนเบา มีการทดสอบการจำแนกฝน โดยใช้เพียง CNN model กับภาพฝนต้นฉบับทั้งหมด 200 ภาพ แบ่งเป็นฝนหนัก 100 รูป และฝนเบา 100 รูป เพื่อวัดความแม่นยำเริ่มต้น หลังจากนั้นจึงมีการทดสอบวิธีการ Fusion model เพื่อเพิ่มความแม่นยำในการจำแนกฝน โดยได้มีการใช้ภาพขอบเข้ามามีส่วนในการฝึกสอนโมเดล โดยมีการทดลองเปลี่ยนวิธีการตรวจจับภาพขอบทั้งหมด 3 วิธีการ ได้แก่ Sobel , Laplacian และ Canny ซึ่งเป็นการเปลี่ยนในขั้นตอนการฝึกสอนในโมเดลที่เป็นการใช้ภาพขอบเพื่อฝึกสอนในรูปแบบที่ 3.2 การเปลี่ยนแปลงภาพขอบที่นำไปฝึกสอนมีผลต่อความแม่นยำเป็นอย่างมาก โดยจะใช้ Confusion matrix สำหรับแสดงผลวัดประสิทธิภาพความถูกต้องของการจำแนกฝนเบา และฝนหนัก

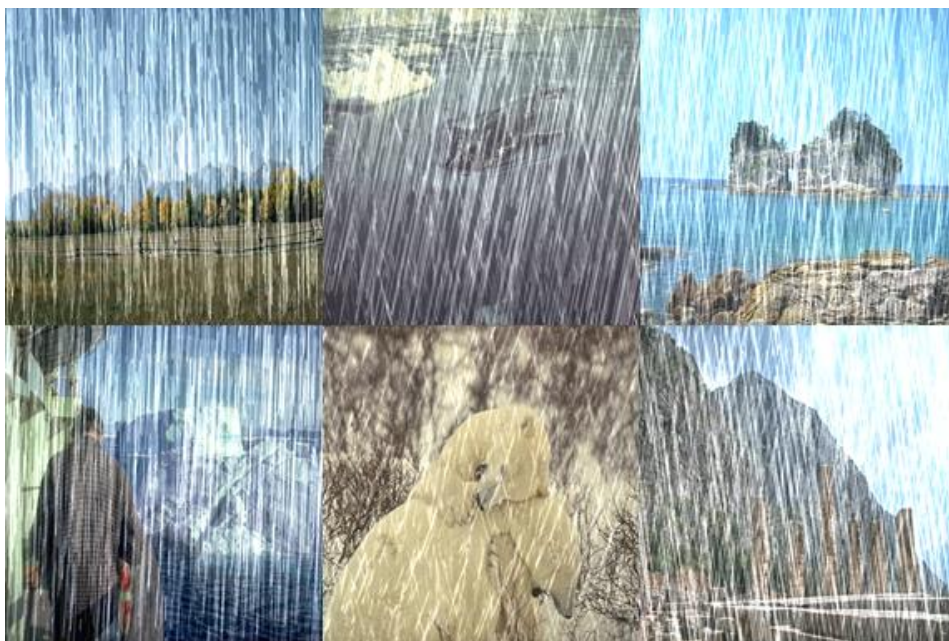
ในการทดสอบแรกจะใช้เพียงภาพฝนต้นฉบับในการฝึกสอนเพียงอย่างเดียว โดยจะใช้ CNN model ในรูปที่ 4.10 เพียงแต่จะไม่มีส่วนของ Fusion model เพื่อทดสอบความแม่นยำของการจำแนกฝนเบา และฝนหนักโดยไม่ผ่านการทำการกระบวนการอื่นๆ ผลลัพธ์การวัดความแม่นยำดังนี้



รูปที่ 4.10 วัดประสิทธิภาพของ CNN model ของภาพฝนต้นฉบับ



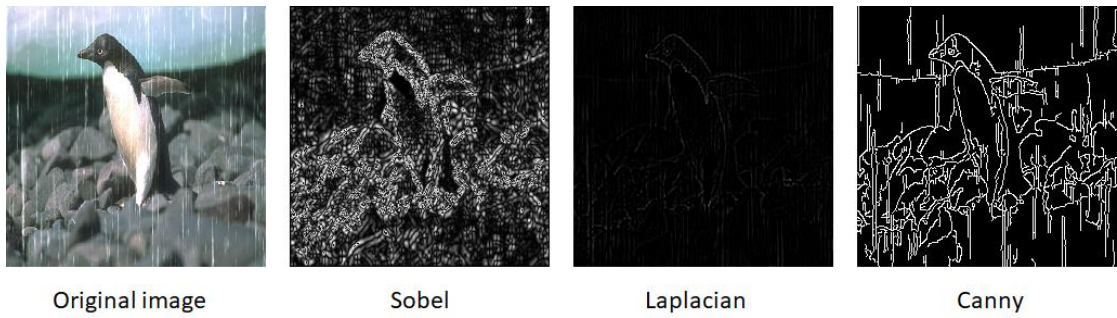
จากรูป 4.10 จะเห็นว่าการจำแนกภาพฝนเบา 100 รูป พบว่ามีการจำแนกเป็นฝนเบาทั้งสิ้น 100 รูป และจำแนกเป็นฝนหนัก 0 รูป ในส่วนของฝนหนักจำนวน 100 รูป พบว่าแยกเป็นฝนหนัก 82 รูป และจำแนกเป็นฝนเบา 18 รูป คิดเป็นเปอร์เซ็นต์ความถูกต้องได้เพียง 91 % ซึ่งเป็นความแม่นยำที่ค่อนข้างสูง แต่ปัญหาอยู่ที่การจำแนกฝนหนัก ที่มีการผิดพลาดถึง 18 รูป โดยตัวอย่างของรูปฝนที่มีการจำแนกผิดแสดงดังรูปที่ 4.11



รูปที่ 4.11 ตัวอย่างภาพที่มีการจำแนกผิดในฝนหนัก

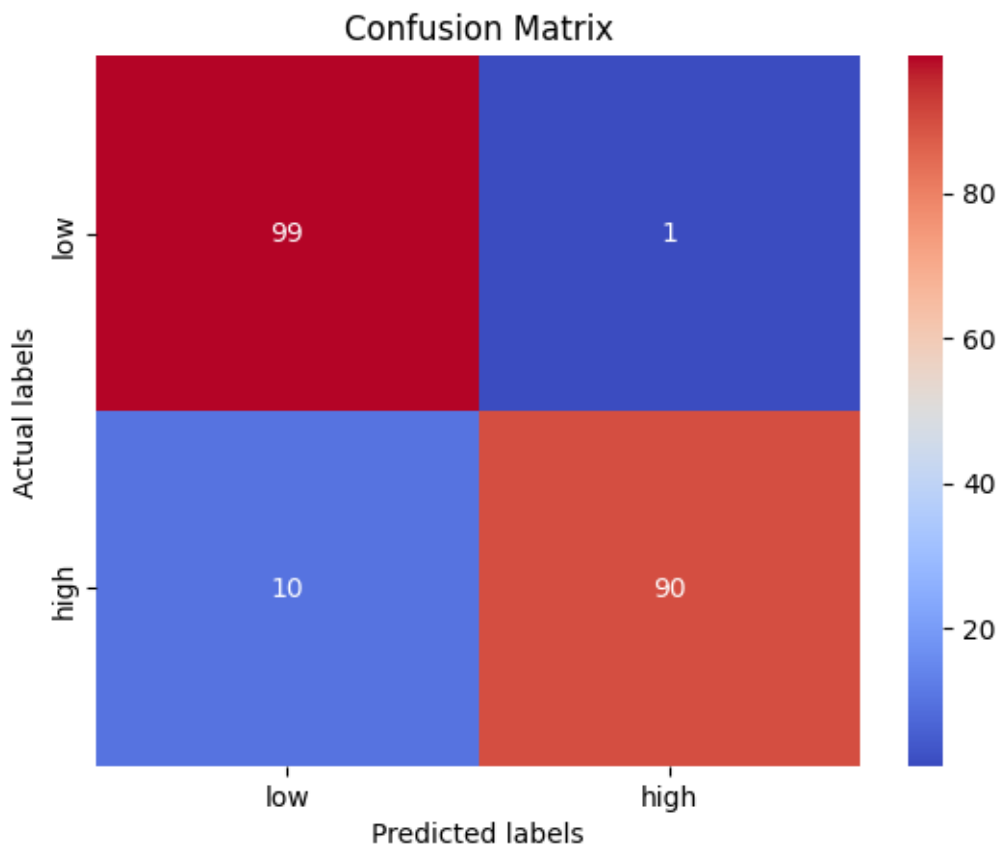
จากรูป 4.11 จะเห็นว่ารูปฝนที่มีการจำแนกผิดส่วนใหญ่จะมีลักษณะที่ภาพพื้นหลังที่เรียบ ไม่ค่อยมีรายละเอียดที่ซับซ้อน

ดังนั้นจึงมีการทดลองเพื่อเพิ่มประสิทธิภาพการจำแนกฝนให้มีความแม่นยำมากยิ่งขึ้น โดยการนำ Fusion model เข้ามามีส่วนร่วมกับการฝึกสอนโมเดล ซึ่งจะใช้ภาพขอบในการฝึกสอนโมเดล ร่วมกับภาพฝนต้นฉบับ โดยการทดลองนี้จะนำวิธีการหาภาพขอบทั้ง 3 วิธีมาใช้ ได้แก่ Sobel , Laplacian และ Canny แสดงตัวอย่างดังรูป 4.12 และมีการวัดความแม่นยำในทุกวิธีการหาภาพขอบ เพื่อเปรียบเทียบความแม่นยำที่มากที่สุดในการจำแนกฝน



รูปที่ 4.12 ตัวอย่างภาพขอบของแต่ละวิธีการ

การทดลองภาพขอบแบบแรกคือการทดสอบโดยใช้ Sobel หาภาพขอบ ซึ่งภาพขอบที่ได้มีลักษณะที่มองเห็นฝนไม่ชัดเจน หากภาพขอบได้บริเวณรายละเอียดพื้นหลังเท่านั้น เมื่อลองนำไปทดสอบกับการทำ Fusion model แล้วแสดงผลการทดสอบผ่าน Confusion matrix ดังนี้

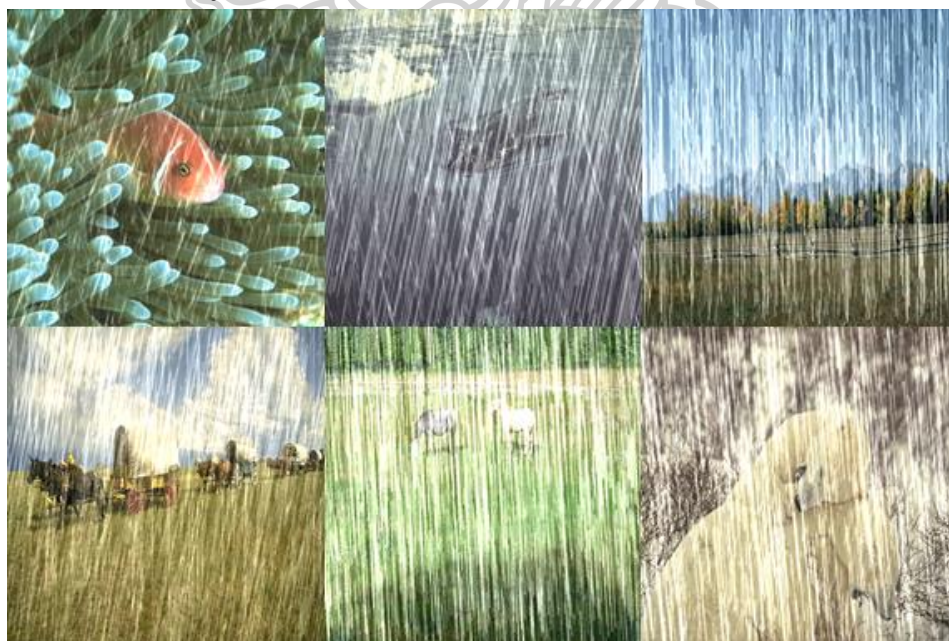


รูปที่ 4.13 วัดประสิทธิภาพของ Fusion model โดยใช้ภาพขอบแบบ Sobel

จากรูป 4.13 จะเห็นว่าความถูกต้องของการจำแนกฝน เมื่อเทียบกับวิธีการดั้งเดิมพบว่ามีค่าความแม่นยำที่เพิ่มขึ้นมาอย่างเห็นได้ชัดในเรื่องของการจำแนกฝนหนัก แต่ในกรณีของฝนเบา นั้นมีความผิดพลาดจำแนกเป็นฝนหนักเพียง 1 รูป และมีการจำแนกจากฝนหนักเป็นฝนเบาจำนวน 10 รูป โดยคิดเป็นเปอร์เซ็นต์ความถูกต้องของการจำแนกทั้งหมด คือ 94.5% โดยแสดงตัวอย่างรูปภาพที่มีการจำแนกผิดจากฝนเบาเป็นฝนหนักในรูปที่ 4.14 และตัวอย่างรูปฝนหนักเป็นฝนเบาในรูปที่ 4.15



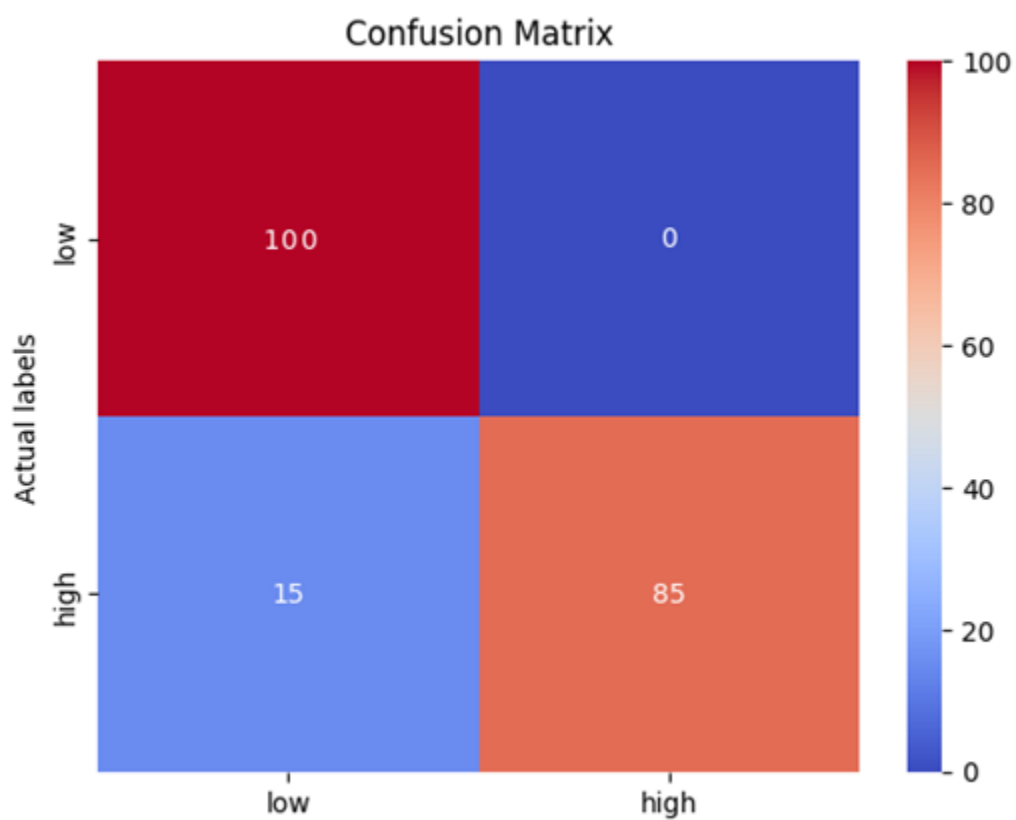
รูปที่ 4.14 ตัวอย่างภาพที่มีการจำแนกผิดในฝนเบา



รูปที่ 4.15 ตัวอย่างภาพที่มีการจำแนกผิดในฝนหนัก

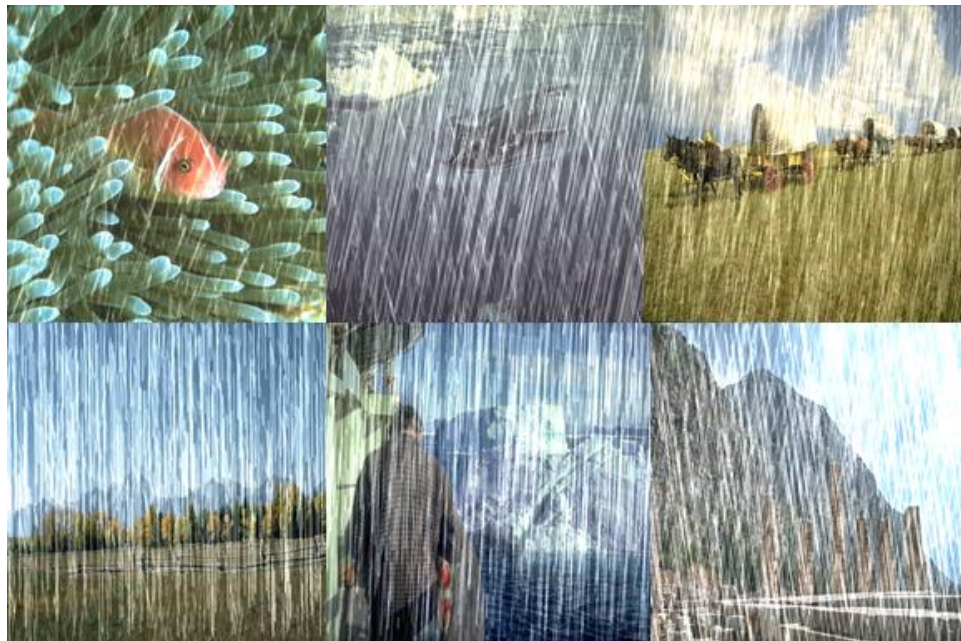
จากรูป 4.14 และ 4.15 จะเห็นว่าการจำแนกผิดของฝนมีลักษณะภาพที่คล้ายกับการจำแนกด้วยวิธีการเดิม คือรายละเอียดพื้นหลังที่มีค่อนข้างเรียบ เช่น ท้องฟ้า หรือแม่น้ำ แต่เนื่องจากการฝึกสอนโดยใช้ภาพขอบเพื่อแยกเม็ดฝนได้ชัดเจนมากขึ้น ทำให้ความผิดพลาดของการจำแนกน้อยลงกว่าเดิม

ในการทดลองต่อมาจะเปลี่ยนการใช้ภาพขอบเป็นวิธีการแบบ Laplacian และนำไปฝึกสอนในโมเดลเดิมเพื่อวัดความแม่นยำของการจำแนกฝนหนัก และฝนเบา โดยจะแสดงค่าความแม่นยำผ่าน Confusion matrix ดังรูปที่ 4.16



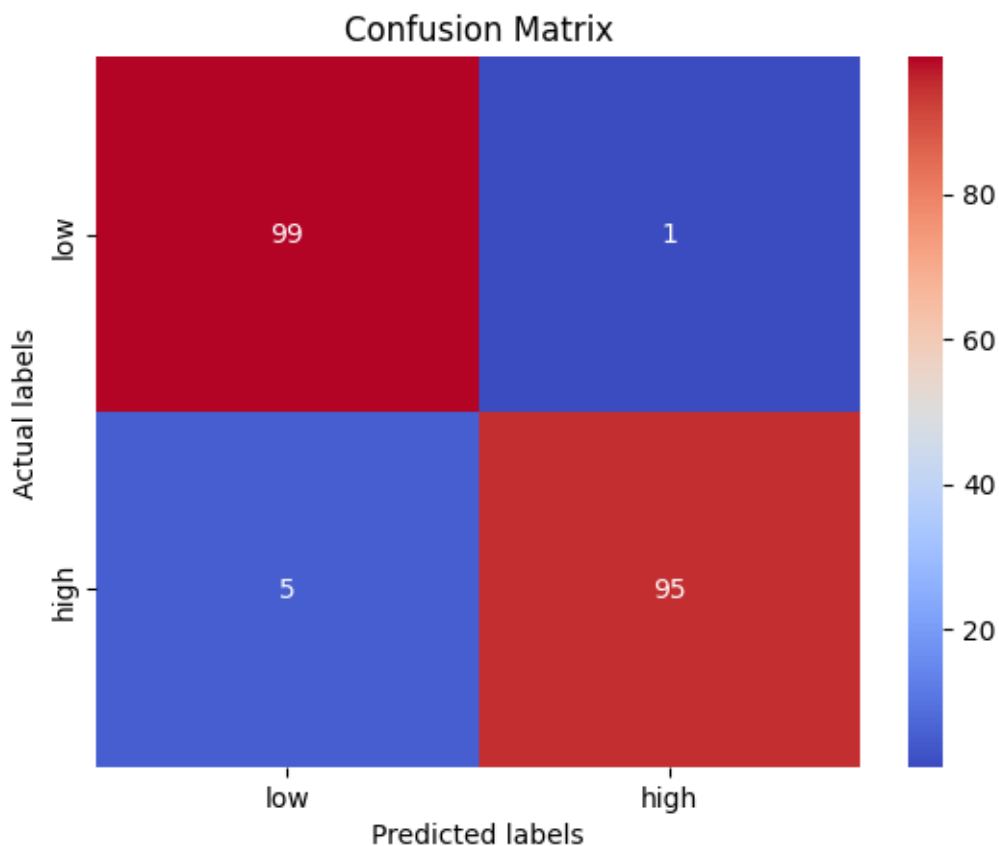
รูปที่ 4.16 วัดประสิทธิภาพของ Fusion model โดยใช้ภาพขอบแบบ Laplacian

จากรูปที่ 4.16 จะเห็นว่ากรการจำแนกฝนเบานั้นถูกต้งทั้งหมด แต่เมื่อมาดูการจำแนกฝนหน้กพบว่า มีการจำแนกจากฝนหน้กเป็นฝนเบาจำนวน 15 รูป หากเปรียบเทียบกับวิธีการหาภาพขอบแบบ Sobel แล้ว ความแม่นยำของการจำแนกฝนยังคงต่ำกว่า แต่การจำแนกฝนเบานั้นสามารถทำได้ถูกต้งทั้งหมด คิดเป็นค่าความแม่นยำได้ 92.5% โดยจะแสดงตัวอย่างรูปฝนที่มีการจำแนกผิดดังรูปที่ 4.17



รูปที่ 4.17 ตัวอย่างภาพที่มีการจำแนกผิดในฝนหน้กโดยใช้ภาพขอบแบบ Laplacian

ในการทดลองต่อมาเป็นการใช้ภาพขอบแบบ Sobel ในการฝึกสอนใน Fusion model ซึ่งเป็นภาพขอบที่สามารถหาขอบของฝนได้ละเอียดมากที่สุดใภาพขอบแบบก่อนหน้าทั้ง 2 แบบ โดยความแม่นยำของการจำแนกฝนเป็นดังรูปที่ 4.18



รูปที่ 4.18 วัดประสิทธิภาพของ Fusion model โดยใช้ภาพขอบแบบ Canny

จากรูป 4.18 จะเห็นว่าการจำแนกฝนเบาสามารถจำแนกถูกเป็นฝนเบา 99 และจำแนกเป็นฝนหนัก 1 รูป และในฝนหนัก มีการจำแนกถูกเป็นฝนหนัก 95 รูป จำแนกเป็นฝนบน 5 รูป ซึ่งการจำแนกฝนหนัก สามารถทำได้ดีมากกว่าภาพขอบก่อนหน้านี้อย่างมาก ซึ่งทำให้ค่าความแม่นยำจำแนกของ Fusion model แบบ Canny สูงถึง 97% ในส่วนของภาพฝนที่มีการจำแนกผิดในฝนเบาแสดงดังรูปที่ 4.19 และตัวอย่างภาพฝนหนักที่จำแนกผิดแสดงดังรูปที่ 4.20



รูปที่ 4.19 ตัวอย่างภาพที่มีการจำแนกผิดในฝนเบาโดยใช้ภาพขอบ Canny



รูปที่ 4.20 ตัวอย่างภาพที่มีการจำแนกผิดในฝนหนักโดยใช้ภาพขอบแบบ Canny

โดยสามารถสรุปค่าความถูกต้องของการจำแนกผ่นหน้า และผ่นขาโดยแบ่งเป็นการทดลองกับ CNN model ที่ไม่ปรับแต่งอะไร และการใช้ Fusion model ที่ใช้ภาพขอบทั้งสามแบบได้แก่ Sobel ,Laplacian และ Canny สรุปได้ดังตารางต่อไปนี้

ตารางที่ 4.1 เปรียบเทียบค่าความแม่นยำของวิธีการจำแนกผ่นในรูปแบบต่างๆ

วิธีการจำแนกผ่น	ค่าความแม่นยำของการจำแนกผ่น
CNN model	91 %
Fusion model + Sobel	94.5 %
Fusion model + Laplacian	92.5 %
Fusion model + Canny	97 %

จากตารางที่ 4.1 เป็นตารางเปรียบเทียบระหว่างวิธีการ CNN model ซึ่งมีโครงสร้างแบบทั่วไปออกแบบมาเพื่อจำแนกวัตถุ ซึ่งเมื่อนำมาใช้กับการจำแนกผ่นพบว่ามีความถูกต้องน้อยที่สุดได้เพียง 91 % จึงมีการใช้ Fusion model เพื่อเพิ่มความแม่นยำของการจำแนกผ่น จากตารางจะเห็นว่าการใช้ Fusion model แบบ Sobel ช่วยเพิ่มประสิทธิภาพการจำแนกผ่นได้ขึ้นมา 94.5% ต่อมาจึงเปลี่ยนมาใช้ภาพขอบแบบ Laplacian ซึ่งค่าความถูกต้องน้อยกว่าแบบ Sobel อยู่ที่ 92.5 % เนื่องจากการจำแนกผ่นหน้ายังทำได้ดีไม่เท่า แต่การจำแนกผ่นขาสามารถทำได้ดีกว่าแบบ Sobel ในส่วนสุดท้ายจะเป็นการใช้ Fusion model กับภาพขอบแบบ Canny ซึ่งเป็นภาพขอบที่สามารถหาเม็ดผ่นได้ดีที่สุด เนื่องจากภาพขอบแบบ Canny มีวิธีการหาภาพขอบที่มีการหาทิศทางของเส้นขอบได้เป็นอย่างดี ทำให้บริเวณผ่นที่อยู่ในทิศทางต่างๆสามารถเด่นชัดมากกว่าวิธีการอื่น และยังมีการแบ่งพิกเซลขอบอย่างชัดเจนด้วยวิธีการทำ Thresholding ที่แบ่งค่าพิกเซลที่มีความแข็งแรง และอ่อนแออย่างชัดเจน ในวิธีหาภาพขอบแบบ Canny มีค่าความถูกต้องได้สูงสุด 97 % เมื่อเทียบกับวิธีดั้งเดิมสามารถจำแนกผ่นได้ดีกว่าอย่างมาก



### 4.3 วัดประสิทธิภาพของการลบฝน

ในการทดสอบประสิทธิภาพของการลบฝน เป็นขั้นตอนที่ได้รับภาพอินพุตจากการจำแนกฝนมาก่อน เพื่อเข้าสู่โมเดลฝนเบา หรือฝนหนักที่ได้ทำการฝึกสอนที่เตรียมไว้ โดยใช้ PSNR และ SSIM ในการวัดประสิทธิภาพเทียบกับภาพ Ground truth ที่ไม่มีฝน ในการลบฝนเบาจะใช้โมเดลที่ชื่อว่า SATLNet ส่วนการลบฝนหนักจะใช้เพียง Autoencoder เท่านั้น

#### 4.3.1 ทดสอบประสิทธิภาพการลบฝนเบาโดยใช้ SATLNet

การทดลองครั้งนี้จะเป็นการทดลองเพื่อวัดประสิทธิภาพของการลบฝนให้มาประสิทธิภาพที่ดีที่สุด โดยยังไม่คำนึงถึงขั้นตอนการจำแนกฝนก่อนหน้า โดยจะใช้ฐานข้อมูลภาพทั้งหมด 150 รูป ที่เป็นฝนเบาทั้งหมด

ในการทดลองแรกจะเป็นการคำนวณหา PSNR และ SSIM ของภาพฝนที่ยังไม่มีการลบฝน เพื่อเป็นตัวชี้วัดให้กับภาพลบฝนที่นำมาเปรียบเทียบประสิทธิภาพของภาพฝนดั้งเดิม โดยจะมีการเปรียบเทียบการวัดประสิทธิภาพของฝน 1 รูป แสดงในรูปที่ 4.21 และเปรียบเทียบทั้งฐานข้อมูลแสดงได้ดังตารางต่อไปนี้



รูปที่ 4.21 ภาพตัวอย่างสำหรับทดสอบประสิทธิภาพการลบฝน

ตารางที่ 4.2 เปรียบเทียบประสิทธิภาพของภาพยนต์ฉบับ 1 รูป และพื้นฐานข้อมูล 150 รูป

	PSNR	SSIM
ทดสอบ 1 รูป	31.03	0.8677
ทดสอบพื้นฐานข้อมูล 150 รูป	26.63	0.8250

การทดลองถัดมาเป็นการทดสอบประสิทธิภาพการทำ Non-block-based autoencoder เป็นขั้นตอนที่นำภาพยนต์ไปฝึกสอนโดยตรงกับ Autoencoder โดยไม่ผ่านการแบ่งส่วนภาพเพื่อฝึกสอนทีละส่วน โดยประสิทธิภาพที่ได้จะมีการเปรียบเทียบทั้งสองแบบคือ เปรียบเทียบประสิทธิภาพกับรูปเดี่ยว และเปรียบเทียบประสิทธิภาพพื้นฐานข้อมูล โดยแสดงรูปตัวอย่างการลบฝนดังรูปที่ 4.22



(ก)

(ข)

(ค)

รูปที่ 4.22 เปรียบเทียบการลบระหว่าง (ก) ภาพฝน (ข) ภาพดั้งเดิม (ค) ภาพลบฝนโดยใช้วิธีการ Non-block-based autoencoder

จากการทดลองลบฝนจะเห็นว่าการลบฝนโดยใช้ Non-block-based สามารถลบฝนได้ดี แต่ยังคงเหลือร่องรอยของการลบฝนในบางส่วน เมื่อวัดประสิทธิภาพของรูปข้างต้นโดยใช้ PSNR และ SSIM พบว่าได้ 37.49 และ 0.9618 ตามลำดับ ซึ่งเมื่อเทียบกับค่าทั้งสองในภาพฝนแล้ว มีประสิทธิภาพที่เพิ่มขึ้นอย่างมาก หลังจากนั้นได้ทดสอบกับฐานข้อมูลฝนจำนวน 150 รูป โดยวัดจาก PSNR และ SSIM พบว่าได้ค่า 30.71 และ 0.9280 ตามลำดับ เมื่อเทียบกับภาพฝนแล้วมีค่าที่เพิ่มขึ้นอย่างมาก แต่เมื่อพิจารณาจากภาพเดี่ยวแล้ว ยังคงเหลือร่องรอยของฝน จึงนำไปสู่การทดลองต่อไปเพื่อเพิ่มประสิทธิภาพของการลบฝนให้ดีขึ้น

การทดลองต่อมาเป็นการทดลองโดยใช้วิธี Block-based autoencoder ซึ่งจะทำการแบ่งภาพเป็น 9 ส่วนเท่ากัน ซึ่งขนาดภาพที่แบ่งออกจะมีขนาด  $80 \times 80$  พิกเซล หลังจากนั้นจะนำไปฝึกสอนผ่าน Autoencoder ที่มีโครงสร้างเหมือนวิธีการก่อนหน้า โดยในการทดสอบประสิทธิภาพนั้นจะนำภาพที่แบ่งออกทั้งหมด 9 ส่วน มารวมกันเป็นภาพเดี่ยว และวัดประสิทธิภาพโดยใช้ PSNR และ SSIM โดยภาพผลการลบฝนโดยใช้วิธี Block-based เป็นดังรูป 4.23



รูปที่ 4.23 เปรียบเทียบการลบระหว่าง (ก) ภาพฝน (ข) ภาพดั้งเดิม (ค) ภาพลบฝนโดยใช้วิธีการ Block-based autoencoder

การทดสอบการลบฝนแบบ Block-based สามารถวัดประสิทธิภาพโดยใช้ PSNR และ SSIM ได้ 37.09 และ 0.9539 ตามลำดับ จะเห็นว่าภาพที่ทำการลบบนนั้นสามารถลบได้เกลี้ยงกว่าวิธีการแบบ Non-block-based แต่เมื่อดูจากค่า PSNR และ SSIM แล้วพบว่ามีความที่น้อยกว่าทั้งสองค่า เนื่องจากการทำ Block-based จะเกิด Blocking artifacts เกิดจากการฝึกสอนภาพทีละส่วน แล้วจึงนำมาต่อกัน แต่เมื่อลองทดสอบกับฐานข้อมูล 150 รูปแล้ว จะได้ค่า PSNR และ SSIM เท่ากับ 31.08 และ 0.9195 ตามลำดับ จะเห็นว่าในส่วนของคุณค่า PSNR สามารถทำได้ดีกว่าแบบ Non-block-based ซึ่งหมายความว่าในบางภาพของฐานข้อมูล สามารถลบฝนดีกว่าการทดลองแบบก่อนมาๆ ถึงแม้ว่าจะมีอาการ Blocking artifacts ก็ตาม ดังนั้นจึงนำไปสู่การทดลองถัดไปในการเพิ่มประสิทธิภาพการลบฝนให้ดีขึ้น

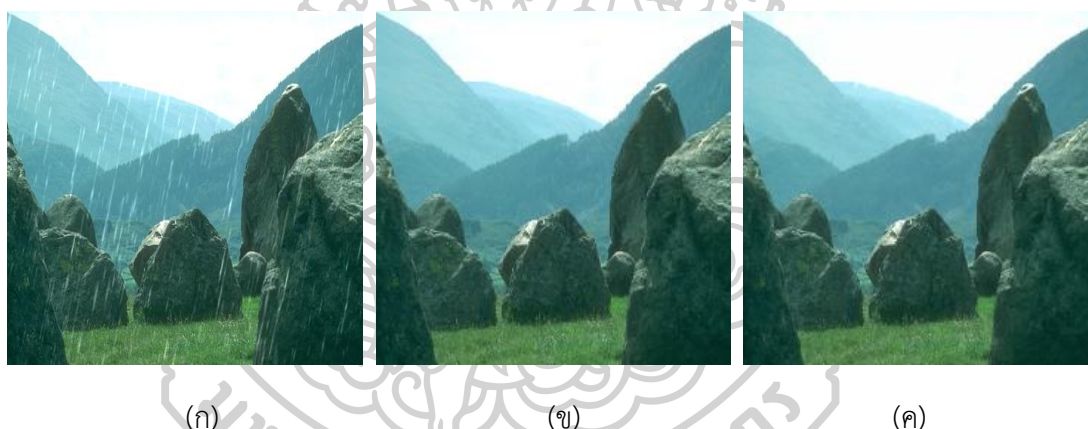
การทดลองต่อมาเป็นการทดลองโดยใช้ Improve block ที่ได้ออกแบบตามโครงสร้างดังรูปที่ 3.6 ซึ่งเป็นการดึงรายละเอียดของภาพที่ผ่านกระบวนการลบฝนแบบ Non-block-based และ Block-based เพื่อให้ได้ภาพการลบฝนใหม่ที่ดึงจุดเด่นของทั้งสองออกมา โดยภาพผลการลบฝนเป็นดังรูป 4.24



รูปที่ 4.24 เปรียบเทียบการลบระหว่าง (ก) ภาพฝน (ข) ภาพดั้งเดิม (ค) ภาพลบฝนโดยใช้วิธีการ Improve block

จากการลบฝนโดยผ่าน Improve block จะเห็นว่ามี การลบฝนที่มากกว่าวิธีการก่อนหน้า และมีรายละเอียดพื้นหลังที่ชัดเจนกว่าในบางจุด แต่ก็มีส่วนที่รายละเอียดพื้นหลังถูกลดลงไป เมื่อทดสอบประสิทธิภาพโดยใช้ PSNR และ SSIM ได้ 37.58 และ 0.9588 ตามลำดับ และเมื่อทดสอบกับ ฐานข้อมูล 150 รูป ได้ 31.46 และ 0.9241 ถึงแม้ว่าการใช้ Improve block จะได้ผลดีก็ตาม แต่ รายละเอียดบางส่วนได้หายไป ดังนั้นจึงนำไปสู่การทดลองต่อไป

การทดลองต่อมาเป็นการใช้ Transfer learning เข้ามาช่วยการลบฝนให้มีประสิทธิภาพยิ่งขึ้น โดยจะทำการสกัดฝนจากภาพฝนที่ทำการลบโดยใช้ Improve block และนำฝนที่สกัดได้ไปซ้อนกับ ฐานข้อมูลที่ไว้ฝึกสอนทั้งหมด และนำมาฝึกสอนใหม่ เพื่อให้ได้โมเดลที่มีลักษณะของภาพฝนที่ต้องการ ทดสอบ โดยผลของการลบฝนเป็นดังรูปที่ 4.25



รูปที่ 4.25 เปรียบเทียบการลบระหว่าง (ก) ภาพฝน (ข) ภาพดั้งเดิม (ค) ภาพลบฝนโดยใช้วิธีการ Transfer learning

จากภาพการลบโดยใช้ Transfer learning เข้ามาช่วยนั้น สามารถลบฝนได้เป็นอย่างดี และสามารถยังคงรายละเอียดจากภาพเดิมได้เป็นอย่างดี เมื่อวัดประสิทธิภาพโดยใช้ PSNR และ SSIM ได้ 38.42 และ 0.9651 ตามลำดับ จะเห็นว่าเมื่อเทียบกับวิธีการก่อนหน้า ค่าที่ได้เพิ่มขึ้นมาอย่างมาก เนื่องจากมีการฝึกสอนโดยใช้รูปแบบของฝนที่ใกล้เคียงกับภาพฝนมากที่สุด ทำให้มีการรู้ตำแหน่ง บริเวณที่ต้องการลบฝนได้ดีกว่าเดิม ในส่วนของการทดสอบฐานข้อมูล 150 รูป สามารถวัด PSNR และ SSIM ได้ 32.12 และ 0.9369 ตามลำดับ และยังคงสามารถทำได้ดีกว่าการทดลองก่อนหน้า

การทดลองสุดท้ายเป็นการใช้ Improve block อีกครั้งหนึ่ง เพื่อเพิ่มประสิทธิภาพของการลบฝน โดยภาพที่นำมาใช้จะเป็นขั้นตอนจากการทำ Improve block ครั้งแรกกับการทำ Transfer learning และหลังจากนั้นจะนำภาพผลลัพธ์ที่ได้ลบกับภาพฝนดั้งเดิม จะได้ภาพที่มีเฉพาะเม็ดฝนออกมา หลังจากนั้นนำไปลบกับการฝนที่ผ่าน Improve block ครั้งที่ 2 จะได้ภาพลบฝนในขั้นตอนสุดท้าย และเมื่อทำการทดสอบกับฐานข้อมูล 150 รูป สามารถสรุปได้ดังตารางที่ 4.3

ตารางที่ 4.3 เปรียบเทียบประสิทธิภาพของการลบฝนในทุกขั้นตอนของการลบฝน

Stages	Metrics	
	PSNR	SSIM
Input (150 รูป)	26.63	0.8250
Non-block-based autoencoder	30.71	0.9280
Block-based autoencoder	31.08	0.9195
Improve block#1	31.46	0.9241
Transfer learning autoencoder	32.12	0.9369
Improve block#2	32.40	0.9395
Output	32.43	0.9395

จากตารางที่ 4.3 เป็นการเปรียบเทียบค่า PSNR และ SSIM ในแต่ละขั้นตอน เพื่อดูแนวโน้มของประสิทธิภาพที่มีการเพิ่มขึ้นอย่างต่อเนื่องทั้งสองค่า โดยผลลัพธ์ของการลบฝนในขั้นตอนสุดท้ายสามารถลบฝนได้เป็นอย่างดี และมีรายละเอียดพื้นหลังที่ยังคงครบสมบูรณ์

ต่อมาเป็นการทดสอบการลบฝนเพื่อเปรียบเทียบกับงานวิจัยอื่น ได้แก่ GCANet ซึ่งเป็นการนำฐานข้อมูลภาพฝนเบ้าทั้งหมด 150 รูป ที่ได้ทำการทดสอบก่อนหน้ามาเปรียบเทียบอีกครั้งดังตาราง 4.4

ตารางที่ 4.4 เปรียบเทียบประสิทธิภาพของการลบฝนในงานวิจัยนี้กับ GCANet

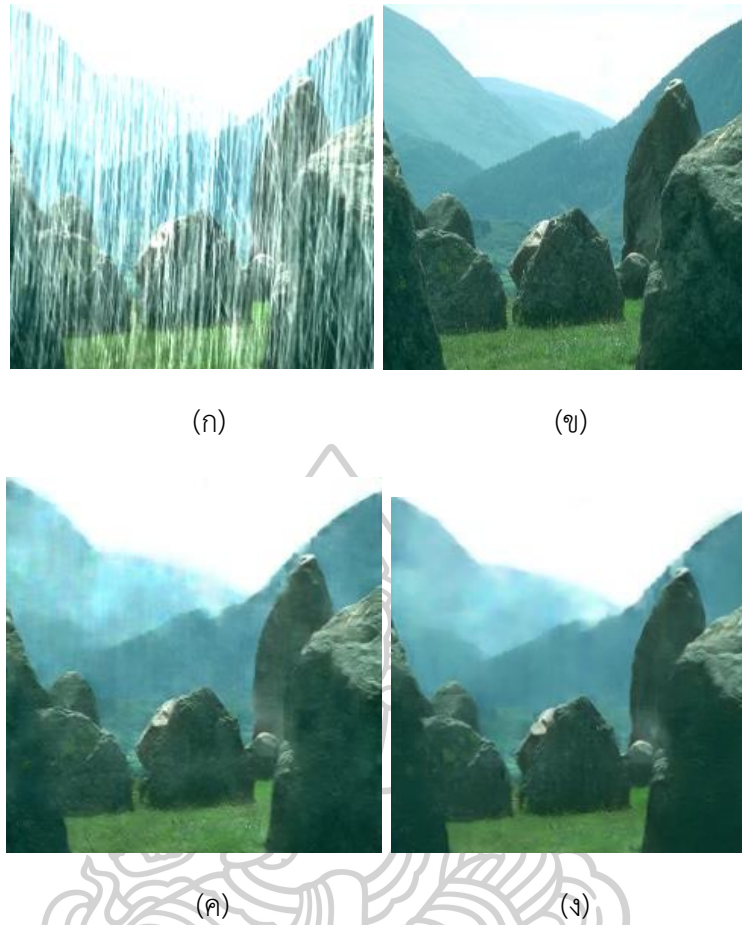
Method	Metrics	
	PSNR	SSIM
Input	26.63	0.8250
GCANet	27.19	0.8579
SATLNet	32.43	0.9395

จากตารางที่ 4.4 เป็นการเปรียบเทียบการลบฝนระหว่างในงานวิจัยนี้กับ GCANet ซึ่งผลลัพธ์ที่ได้จากการวัดประสิทธิภาพด้วย PSNR และ SSIM พบว่า SATLNet ของงานวิจัยนี้มี PSNR และ SSIM ที่สูงกว่าอย่างมาก

#### 4.3.2 ทดสอบประสิทธิภาพการลบฝนหนักโดยใช้ Autoencoder

สำหรับการลบฝนหนักในการทดลองนี้จะเป็นการทดลองเพื่อวัดประสิทธิภาพของรูปฝนหนักเพียงอย่างเดียว ยังไม่ได้นำรูปจากขั้นตอนการจำแนกฝนมาใช้ เนื่องจากต้องการให้เห็นถึงประสิทธิภาพของการลบฝนหนักเพียงอย่างเดียว

ในการทดลองนี้จะใช้วิธีการลบฝนผ่าน Autoencoder สองครั้ง เนื่องจากฝนหนักมีเม็ดฝนมีค่อนข้างใหญ่ทำให้การลบภายในครั้งแรกอาจไม่เพียงพอ โดยภาพตัวอย่างผลการลบฝนในครั้งที่ 1 และครั้งที่ 2 ดังรูปที่ 4.26



รูปที่ 4.26 เปรียบเทียบระหว่าง (ก) ภาพฝน (ข) ภาพดั้งเดิม (ค) ภาพการลบฝนครั้งที่ 1 และ (ง) การลบฝนครั้งที่ 2

จากรูป 4.26 จะเห็นว่าการลบฝนครั้งที่ 1 ยังหลงเหลือร่องรอยของการลบฝนอยู่ สังเกตได้จากบริเวณภูเขาด้านหลัง และเมื่อทำการลบฝนครั้งที่ 2 โดยใช้ภาพการลบฝนจากครั้งก่อนพบว่า บริเวณที่เหลือร่องรอยสามารถลบออกได้ดี จึงมีการทดลองลบฝนครั้งที่ 3 เพื่อเปรียบเทียบประสิทธิภาพ โดยจะมีการวัดประสิทธิภาพของภาพฝนหนักโดยใช้ฐานข้อมูล Rain100H จะเปรียบเทียบ PSNR และ SSIM ในการลบฝนทั้งหมด 3 ครั้ง เพื่อแสดงถึงประสิทธิภาพที่เพิ่มขึ้น



ตารางที่ 4.5 เปรียบเทียบประสิทธิภาพของการลบฝนหนักในครั้งที่ 1 และครั้งที่ 2

	PSNR	SSIM
ภาพฝน	19.19	0.6492
ลบฝนครั้งที่ 1	27.94	0.9048
ลบฝนครั้งที่ 2	28.44	0.9140
ลบฝนครั้งที่ 3	28.12	0.9098
ลบฝนครั้งที่ 4	27.92	0.9034
ลบฝนครั้งที่ 5	27.64	0.8991

จากตารางที่ 4.5 จะเห็นว่า การวนซ้ำเพื่อลบฝนนั้นช่วยเพิ่มประสิทธิภาพได้ทั้งในการวัดแบบ PSNR และ SSIM แต่ต้องอยู่ในจำนวนครั้งของการลบฝนที่เหมาะสม โดยในงานวิจัยนี้ได้ทำการทดลองทั้งหมด 5 รอบดังตาราง พบว่าการวนซ้ำ 2 รอบได้ผลดีที่สุด โดยสาเหตุของการวนซ้ำในรอบที่ 2 ขึ้นไปนั้นมีประสิทธิภาพลดลงโดยสังเกตจาก PSNR และ SSIM เนื่องจากการลบฝนในแต่ละรอบมีการลดทอนรายละเอียดพื้นหลังของภาพลงตลอดการวนซ้ำ เพื่อให้เม็ดฝนนั้นหายไปจากภาพ ทำให้ภาพที่ในแต่ละรอบสามารถลบฝนได้ดีหากมีการลบฝนวนซ้ำในจำนวนรอบท้ายๆ แต่รายละเอียดพื้นหลังก็ถูกลดลงตาม ดังนั้นการวนซ้ำรอบที่เหมาะสมจึงมีความสำคัญอย่างยิ่ง จากการทดลองจึงมีการใช้การวนซ้ำเพียงสองรอบ

ต่อมาเป็นการทดสอบการลบฝนเพื่อเปรียบเทียบกับงานวิจัยอื่น ได้แก่ GCANet ซึ่งเป็นการนำฐานข้อมูลภาพฝนหนักทั้งหมด 100 รูป ที่ได้ทำการทดสอบก่อนหน้ามาเปรียบเทียบกับอีกครั้ง ดังตาราง 4.6

ตารางที่ 4.6 เปรียบเทียบประสิทธิภาพของการลบฝนในงานวิจัยนี้กับ GCANet

Method	Metrics	
	PSNR	SSIM
Input	26.63	0.8250
GCANet	20.35	0.6596
Our method	28.44	0.9140

จากตารางที่ 4.6 เป็นการเปรียบเทียบการลบฝนระหว่างในงานวิจัยนี้กับ GCANet ซึ่งผลลัพธ์ที่ได้จากการวัดประสิทธิภาพด้วย PSNR และ SSIM พบว่าการลบฝนหนักของงานวิจัยนี้มี PSNR และ SSIM ที่สูงกว่าอย่างมาก เนื่องจาก GCANet ไม่ได้ถูกออกแบบมากับฝนหนัก การลบฝนจึงไม่สามารถลบได้ดี อีกทั้งยังทำให้บริเวณที่มีฝนเกิด

#### 4.3.3 ทดสอบประสิทธิภาพการลบฝนทั้งระบบ

ในการทดสอบประสิทธิภาพทั้งระบบ เป็นการทดสอบตั้งแต่ขั้นตอนการจำแนกฝน จนถึงขั้นตอนได้ภาพการลบฝน เพื่อเป็นการทดสอบวิธีการลบฝนที่มีการจำแนกฝนมาก่อนหน้า โดยการจำแนกฝนเบามีความถูกต้อง 149 รูป จำแนกผิด 1 รูป ในส่วนของฝนหนักมีการจำแนกถูก 95 รูป จำแนกผิด 5 รูป ดังนั้นรูปที่ต้องเข้าสู่โมเดลการลบฝนเบาทั้งหมด 154 รูป และรูปที่เข้าสู่โมเดลฝนหนักทั้งหมด 96 รูป โดยสามารถแสดงประสิทธิภาพทั้งตารางต่อไปนี้

ตารางที่ 4.7 เปรียบเทียบประสิทธิภาพของการลบฝนหนัก และฝนเบาทั้งระบบ

	PSNR	SSIM
ลบฝนเบา	31.16	0.9255
ลบฝนหนัก	28.89	0.9187

จากตารางที่ 4.7 จะเห็นว่าการลบฝนเบาเมื่อเทียบกับการลบฝนก่อนหน้า จะมีประสิทธิภาพที่ลดลงเนื่องจากต้องนำภาพฝนหนัก 5 รูปที่มีการจำแนกผิดเข้าสู่กระบวนการลบฝนเบา ซึ่งโมเดลการลบฝนเบาไม่สามารถทำการลบฝนหนักออกได้ แต่ในส่วนของกระบวนการลบฝนหนักที่มีประสิทธิภาพเพิ่มขึ้นมาเนื่องจากมีการแบ่งรูปไปให้ทางฝนเบา 5 รูป ซึ่งทำให้ค่าของ PSNR และ SSIM ขึ้นมาเพียงเล็กน้อย

#### 4.3.4 ทดสอบประสิทธิภาพการลบฝนจริง

การทดสอบประสิทธิภาพของฝนจริงนั้น เนื่องจากไม่สามารถหาภาพ Ground truth ได้จึงไม่สามารถเปรียบเทียบประสิทธิภาพโดยวิธีการ PSNR และ SSIM ได้ ในการทดลองนี้จึงมีการทดสอบเพียงลบฝนจริง และนำมาเปรียบเทียบกับภาพฝนดั้งเดิมเพียงเท่านั้น โดยการทดสอบภาพฝนจริงกับการใช้ลบฝนเบาเป็นดังรูปที่ 4.27 ถึง รูปที่ 4.29



รูปที่ 4.27 เปรียบเทียบ (ก) ภาพฝนจริงดั้งเดิม (ข) ภาพลบฝนโดยใช้ GCANet และ (ค) ภาพที่ผ่านการลบฝนโดยใช้ SATLNet



รูปที่ 4.28 เปรียบเทียบ (ก) ภาพฝนจริงดั้งเดิม (ข) ภาพลบฝนโดยใช้ GCANet และ (ค) ภาพที่ผ่านการลบฝนโดยใช้ SATLNet

ที่มา : [https://res.cloudinary.com/dk-find-out/image/upload/q\\_80,w\\_1920,f\\_auto/A-123rf-39618990\\_nqrp4y.jpg](https://res.cloudinary.com/dk-find-out/image/upload/q_80,w_1920,f_auto/A-123rf-39618990_nqrp4y.jpg)



(ก)

(ข)

(ค)

รูปที่ 4.29 เปรียบเทียบ (ก) ภาพฝนจริงดั้งเดิม (ข) ภาพลบฝนโดยใช้ GCANet และ (ค) ภาพ  
 ที่ผ่านการลบฝนโดยใช้ SATLNet

ที่มา : <https://www.maximum-inc.com/wp-content/uploads/2019/10/Rainy-Lamp-Post.jpg>

จากรูปแสดงการเปรียบเทียบการลบฝนแต่ละวิธีจะเห็นว่าสามารถลบฝนได้ในกรณีที่ฝนที่  
 ลักษณะที่เป็นเส้นตรง แต่ในกรณีที่เป็นเม็ดลักษณะกลม การลบฝนอาจจะยังทำได้ไม่ค่อยดีนัก สังเกต  
 ได้จากบริเวณต้นไม้ทางด้านซ้าย ที่มีฝนที่มีลักษณะคล้ายจุดกลม ซึ่งวิธีการลบฝนในงานวิจัยนี้ยังลบฝน  
 ลักษณะนี้ได้ในระดับหนึ่ง แต่หากเทียบกับวิธีการ GCANet จะเห็นว่าสามารถลบฝนได้บางส่วนของภาพ  
 เท่านั้น ในส่วนของการทดลองลบฝนหนักในภาพจริง แสดงดังรูปที่ 4.30 ถึงรูปที่ 4.32



(ก)

(ข)

(ค)

รูปที่ 4.30 เปรียบเทียบ (ก) ภาพฝนจริงดั้งเดิม (ข) ภาพการลบฝนโดยใช้ GCANet และ (ค)

ภาพที่ผ่านการลบฝนโดยใช้โมเดลฝนหนัก

ที่มา :

[https://www.omron.com/global/en/edge-link/uploads/554\\_1.jpg](https://www.omron.com/global/en/edge-link/uploads/554_1.jpg)



(ก)

(ข)

(ค)

รูปที่ 4.31 เปรียบเทียบ (ก) ภาพฝนจริงดั้งเดิม (ข) ภาพการลบฝนโดยใช้ GCANet และ (ค)

ภาพที่ผ่านการลบฝนโดยใช้โมเดลฝนหนัก

ที่มา :

[https://static.toiimg.com/thumb/msid-93975791,imsize-91980,width-](https://static.toiimg.com/thumb/msid-93975791,imsize-91980,width-400,resize-mode-4/93975791.jpg)

[400,resize-mode-4/93975791.jpg](https://static.toiimg.com/thumb/msid-93975791,imsize-91980,width-400,resize-mode-4/93975791.jpg)



(ก)

(ข)

(ค)

รูปที่ 4.32 เปรียบเทียบ (ก) ภาพฝนจริงดั้งเดิม (ข) ภาพการลบฝนโดยใช้ GCANet และ (ค) ภาพที่ผ่านการลบฝนโดยใช้โมเดลฝนหนัก

ที่มา : [https://files.ocula.com/anzax/a6/a63610a6-870b-43d5-a413-e628fe7a06be\\_750\\_556.jpg](https://files.ocula.com/anzax/a6/a63610a6-870b-43d5-a413-e628fe7a06be_750_556.jpg)

จากรูปที่ 4.30 ถึงรูปที่ 4.32 จะเห็นว่าวิธีการลบฝนของงานวิจัยนี้สามารถลบฝนได้ค่อนข้างดี ซึ่งฝนส่วนใหญ่สามารถลบออกได้ แต่เมื่อมาดูรายละเอียดพื้นหลัง มีการสูญเสียรายละเอียดที่ค่อนข้างมากจนทำให้มีลักษณะที่เบลอในบางจุด ในขณะที่การลบฝนหนักของ GCANet ไม่สามารถลบฝนได้ เนื่องจากว่ามีการฝึกสอนโดยใช้ฝนเบาเพียงเท่านั้น ทำให้ฝนที่มีความถี่มาก ๆ ไม่สามารถลบได้

จากการทดสอบฝนจริงทั้งสองภาพสรุปได้ว่าในงานวิจัยนี้ สามารถลบฝนจริงได้ทั้งแบบฝนเบา และฝนหนัก ซึ่งประสิทธิภาพอาจดีไม่เทียบเท่ากับการลบฝนแบบสังเคราะห์ เนื่องจากการฝึกสอนโมเดลใช้ฐานข้อมูลภาพที่สังเคราะห์ขึ้นมา การลบฝนจริงนั้นจึงทำให้มีประสิทธิภาพการลบฝนที่ไม่เทียบเท่า แต่ยังสามารถลบฝนได้

#### 4.4 วัดประสิทธิภาพของการตรวจจับวัตถุหลังจากลบฝนโดยใช้ YOLO

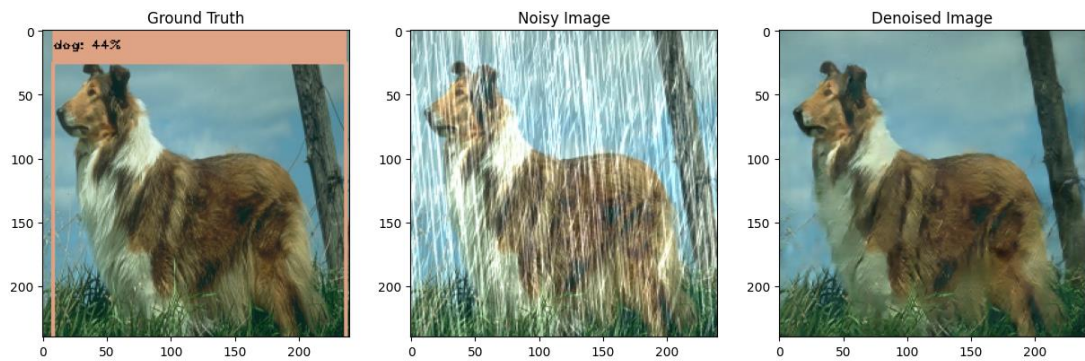
##### 4.4.1 การวัดประสิทธิภาพโดยใช้ภาพฝนสังเคราะห์

ในการทดลองนี้เป็นการวัดประสิทธิภาพของภาพเมื่อทำการลบฝนออก ใช้วิธีการตรวจจับวัตถุเพื่อทดสอบความแม่นยำของภาพ โดยใช้ YOLOv5 ในการตรวจจับวัตถุ โดยการทดลองครั้งนี้จะใช้ภาพเฉพาะที่สามารถตรวจจับได้ใน Ground truth เท่านั้น เนื่องจากว่า Ground truth เป็นภาพที่มีเฉพาะพื้นหลัง ไม่มีฝน หากไม่สามารถตรวจจับวัตถุใน Ground truth ได้ ภาพที่เกิดจากการลบฝนจึงยากต่อการตรวจจับวัตถุไปด้วย ดังนั้นจึงมีการทดสอบตรวจจับวัตถุในฐานข้อมูล Rain100L และ Rain100H ที่ไว้ใช้สำหรับทดสอบจำนวน 200 รูป พบว่าสามารถตรวจจับวัตถุในภาพ Ground truth ได้ทั้งหมด 36 รูป แสดงตัวอย่างดังรูป 4.33

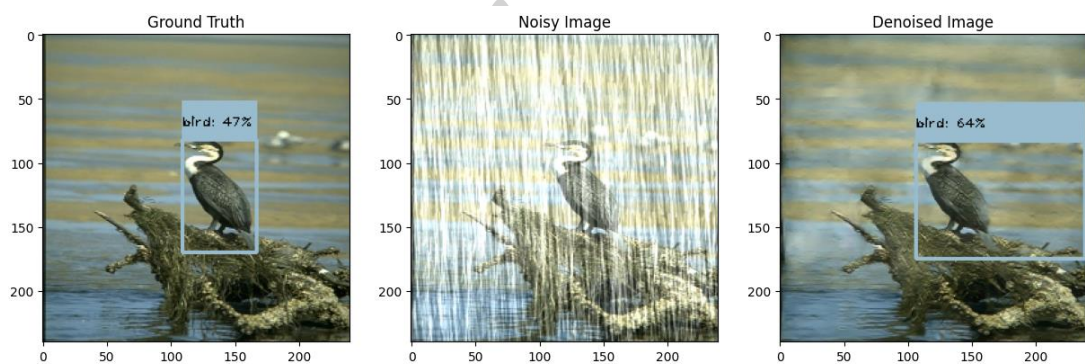


รูปที่ 4.33 ตัวอย่างการตรวจจับวัตถุจากภาพ Ground truth โดยใช้ YOLOv5

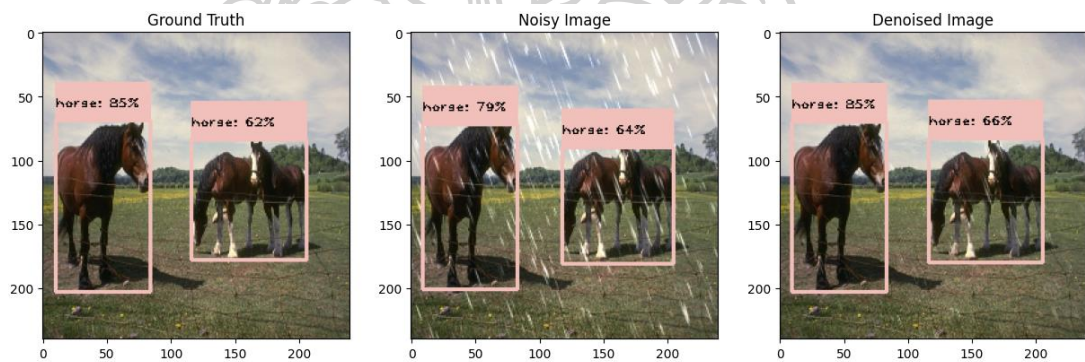
การวัดประสิทธิภาพจะนำภาพฝนที่ถูกลบออกมาตรวจจับวัตถุเทียบกับภาพฝนที่ยังไม่ถูกลบออก โดยจะคิดเปอร์เซ็นต์ความถูกต้องจากการนับภาพที่สามารถตรวจจับวัตถุได้ โดยไม่คำนึงถึงเปอร์เซ็นต์ความถูกต้องของการตรวจจับวัตถุ โดยในบางรูปของการลบฝนก็ไม่สามารถตรวจจับวัตถุได้ ตัวอย่างดังรูปที่ 4.34 หรือในบางกรณีสามารถตรวจจับได้เพียงภาพลบฝนเท่านั้นดังรูปที่ 4.35 หรือในภาพสามารถตรวจจับวัตถุได้ทั้งหมดดังรูปตัวอย่างที่ 4.36



รูปที่ 4.34 ตัวอย่างภาพที่ไม่สามารถตรวจจับวัตถุได้ในภาพฝน และภาพลบฝน



รูปที่ 4.35 ตัวอย่างภาพที่ตรวจจับวัตถุเฉพาะภาพที่ลบฝน



รูปที่ 4.36 ตัวอย่างภาพที่ตรวจจับวัตถุเฉพาะภาพที่ลบฝน

โดยจากการทดสอบภาพทั้งหมด 36 รูป พบว่ามีการตรวจจับวัตถุในภาพฝนได้ทั้งหมด 14 รูป จากทั้งหมด 38 รูป คิดเป็นความถูกต้องเท่ากับ 36.84% ในส่วนของการทดสอบภาพที่ถูกลบฝนออกแล้ว สามารถตรวจจับวัตถุได้จำนวน 28 รูป จาก 36 รูป คิดเป็น 77.77% จะเห็นว่าการลบฝนนั้นสามารถช่วยเพิ่มประสิทธิภาพของการตรวจจับวัตถุได้มากยิ่งขึ้น



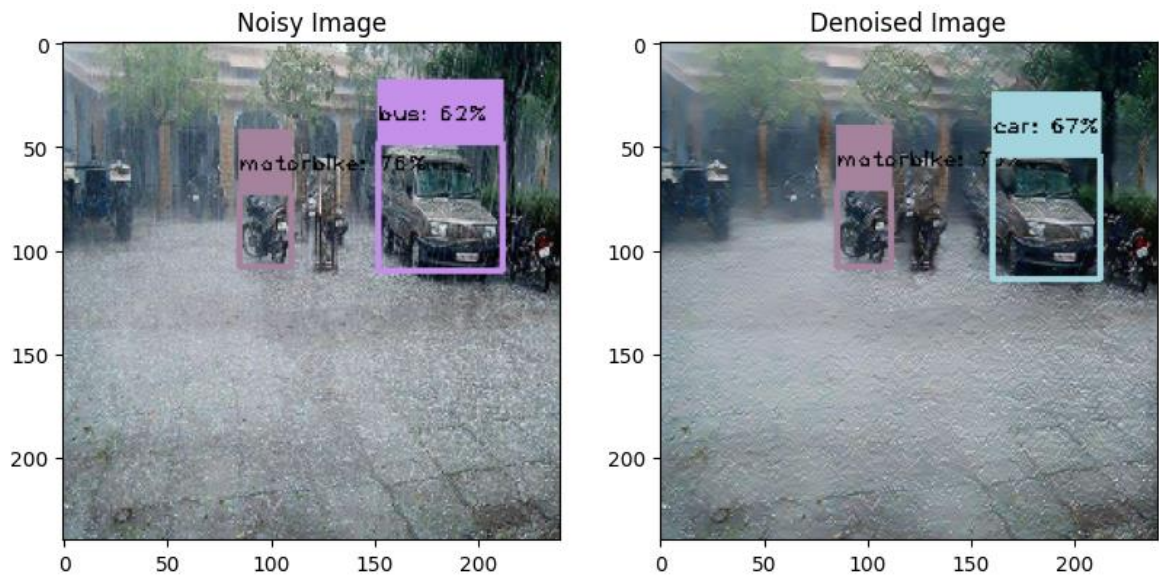
#### 4.4.2 การวัดประสิทธิภาพโดยใช้ฝนจริง

ในการทดลองนี้จะใช้ฝนจริงในการทดสอบตรวจจับวัตถุ โดยภาพฝนที่ใช้แสดงดังรูปที่ 4.37 ซึ่งจะเห็นว่าภายในภาพมีฝนตกที่ค่อนข้างหนัก จนทำให้รายละเอียดบางอย่างไม่ชัดเจน หรือโดนบดบังจากเม็ดฝน เมื่อนำไปลบฝนเพื่อนำไปตรวจจับวัตถุได้ดังรูปที่ 4.38



รูปที่ 4.37 ภาพฝนจริงที่นำตรวจจับวัตถุในภาพ

ที่มา : [https://resize.indiatvnews.com/en/resize/newbucket/730\\_-  
/2021/02/capture-1613549245.jpg](https://resize.indiatvnews.com/en/resize/newbucket/730_-/2021/02/capture-1613549245.jpg)



รูปที่ 4.38 ภาพการลบฝนจริงที่นำไปตรวจจับวัตถุในภาพ

จากรูปที่ 4.38 จะสังเกตว่าในภาพของฝนนั้นการตรวจจับมองว่ารถยนต์กลายเป็นรถบัส ซึ่งเกิดจากการที่มีฝนมาบังบริเวณรถยนต์ทำให้การทำงานผิดพลาด แต่เมื่อมีการลบฝนออกจากภาพก่อน และนำไปตรวจจับวัตถุพบว่า วัตถุที่ถูกลบออกสามารถตรวจจับวัตถุถูกต้อง โดยขึ้นว่าเป็นรถยนต์ ทำให้เห็นว่าการลบฝนออกจากภาพจึงช่วยเพิ่มประสิทธิภาพของการลบฝนได้



## บทที่ 5

### สรุปและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้นำเสนอวิธีการลบฝนออกจากภาพโดยใช้การเรียนรู้เชิงลึก มีวัตถุประสงค์เพื่อศึกษาวิธีการออกแบบโครงสร้างโมเดลใช้สำหรับจำแนกฝน และลบฝนออกจากภาพ โดยมีการใช้เทคนิคต่างๆจากงานประมวลผลภาพมาช่วยในการลบฝนออกจากภาพได้ดีขึ้น ซึ่งทำให้ช่วยในเรื่องงานทางด้านกรมองเห็นภาพ เช่น ในเรื่องของการตรวจจับวัตถุ หากตรวจจับวัตถุขณะมีฝนนั้นในบางครั้งไม่สามารถหาวัตถุที่ต้องการเจอได้ แต่หากทำการลบฝนก่อนที่จะตรวจจับวัตถุจะสามารถตรวจจับวัตถุได้เป็นอย่างดี ในงานวิจัยนี้ได้ทำการลบฝนอยู่สองแบบคือ ฝนเบา และฝนหนัก ซึ่งลักษณะของฝนที่แตกต่างกันอย่างมา ในกรณีของฝนเบา นั้นลักษณะของเม็ดฝนจะมีความแยกออกจากกัน และจะเห็นเป็นเม็ดมากกว่าฝนหนัก ในส่วนของฝนหนักนั้นจะมีความถี่ของฝน และอยู่ติดกันมากกว่าฝนเบา จะบดบังภาพมากกว่าในกรณีฝนเบาเป็นอย่างมาก

ในวิทยานิพนธ์นี้แบ่งการทำงานออกเป็น 2 ส่วน ได้แก่ 1. ส่วนของการจำแนกฝนเบา และฝนหนัก 2. ส่วนของการลบฝนออกจากภาพฝนเบา และฝนหนัก

ในส่วนแรกเป็นการจำแนกฝนเบา และฝนหนัก โดยสาเหตุที่ต้องมีการจำแนกฝนก่อน เนื่องจากขั้นตอนการลบฝนนั้น จะทำได้ดีเมื่อมีการฝึกสอนโมเดลที่มีลักษณะฝนแยกกัน ดังนั้นการจำแนกฝนเพื่อเข้าสู่โมเดลที่ถูกฝึกสอนไว้แยกกันช่วยเพิ่มประสิทธิภาพได้ โดยขั้นตอนการของการจำแนกฝน จะใช้คอนโวลูชันในการจำแนกฝน แต่เนื่องจากความถูกต้องที่ได้นั้นยังมีความผิดพลาด จึงมีการนำ Fusion model เพื่อเพิ่มประสิทธิภาพของการจำแนกฝน โดยการนำ Weight ของโมเดลทั้งสองมารวมกัน โมเดลแรกจะเป็นการฝึกสอนโดยใช้ภาพฝน RGB ส่วนโมเดลที่สองใช้ภาพขอบแบบ Canny ซึ่งให้ประสิทธิภาพดีที่สุดในการหาขอบของฝน โดยประสิทธิภาพที่จากการทดสอบภาพฝนทั้งหมด 200 ภาพ มีความถูกต้องที่ 98% มากกว่าการใช้คอนโวลูชันโมเดลเพียงอย่างเดียวที่มีความถูกต้องอยู่ที่ 94%

ในส่วนที่สองเป็นวิธีการลบฝนออกจากภาพ โดยจะรับภาพที่ถูกจำแนกมาจากขั้นตอนแรก จะมีการแบ่งออกเป็นสองส่วนคือ ส่วนของการลบฝนเบา และฝนหนัก ในส่วนของฝนเบา นั้นจะใช้วิธีการ SATLNet ซึ่งภายในโครงสร้างนี้มีส่วนที่สำคัญคือ Autoencoder , Improve block และ Transfer learning ซึ่งช่วยในการลบฝนเบาได้เป็นอย่างดีจากการทดสอบลบภาพฝนจำนวน 150 รูป

และนำไปเปรียบเทียบภาพต้นฉบับที่ไม่มีฝน พบว่าภาพส่วนใหญ่สามารถลบฝนได้เป็นอย่างดี และรายละเอียดพื้นหลังในภาพยังอยู่ครบใกล้เคียงกับภาพต้นฉบับ แต่ในบางภาพยังคงเหลือร่องรอยของฝนไว้บางส่วน ในส่วนของฝนหนักนั้นจะใช้เพียง Autoencoder เท่านั้น แต่มีการวนซ้ำฝนขั้นตอนการฝึกสอนทั้งหมดสองรอบ เนื่องจากลักษณะฝนที่มีมากทำให้ในการลบฝนครั้งแรกยังคงเหลือร่องรอยฝนอยู่ การลบฝนรอบสองนั้นจึงช่วยเพิ่มประสิทธิภาพการลบฝนได้ โดยมีการทดสอบกับภาพฝนหนัก 100 ภาพ พบว่าสามารถลบฝนได้ดี แต่รายละเอียดพื้นหลังจะถูกลดลงไปอย่างชัดเจน

## 5.2 ปัญหาและข้อเสนอแนะ

1. การลบฝนกรณีฝนเบาในบางภาพไม่สามารถลบได้เกลี้ยงทั้งหมด โดยส่วนใหญ่จะอยู่ที่บริเวณท้องฟ้า ตัวอย่างดังรูปที่ 5.1



รูปที่ 5.1 ตัวอย่างการเปรียบเทียบระหว่างภาพต้นฉบับ และภาพที่ลบฝนออกแล้วแต่ยังหลงเหลือฝนอยู่

2. ในกรณีของการลบฝนหนัก ถึงแม้จะสามารถลบฝนออกได้ทั้งหมด แต่รายละเอียดพื้นหลัง จะถูกลดลงไป แสดงตัวอย่างดังรูป 5.2



รูปที่ 5.2 แสดงตัวอย่างการเปรียบเทียบระหว่างภาพต้นฉบับ ภาพฝน และภาพฝนที่ถูกลบออกที่มี รายละเอียดพื้นหลังที่ลดลง

### 5.3 แนวทางการพัฒนาต่อยอด

1. พัฒนาต่อยอดโปรแกรมให้สามารถใช้กับภาพเคลื่อนไหวได้
2. พัฒนาให้สามารถใช้สำหรับลบฝนจริงได้อย่างมีประสิทธิภาพมากขึ้น



## รายการอ้างอิง

1. Zheng, S., et al. *SAPNet: Segmentation-Aware Progressive Network for Perceptual Contrastive Deraining*. 2021.
2. Xianghao Jiao, Y.L., Jiabin Gao, Xinyuan Chu, Risheng Liu, Xin Fan. *PEARL: Preprocessing Enhanced Adversarial Robust Learning of Image Deraining for Semantic Segmentation*. 2023.
3. Dor Bank, N.K., Raja Giryes. *Autoencoders*. 2021.
4. Yoonsik Kim, J.W.S., Gu Yong Park, Nam Ik Cho. *Transfer Learning From Synthetic to Real-Noise Denoising With Adaptive Instance Normalization*. in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.
5. Jason Wang, L.P. *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. 2017.
6. W. Yang, R.T.T., J. Feng, Z. Guo, S. Yan and J. Liu. *Joint Rain Detection and Removal from a Single Image with Contextualized Deep Networks*. 2020.
7. H. Zhang, V.S.a.V.M.P., *Image De-Raining Using a Conditional Generative Adversarial Network*. 2020.
8. X. Fu, J.H., X. Ding, Y. Liao and J. Paisley. *Clearing the Skies: A Deep Network Architecture for Single-Image Rain Removal*. 2017.
9. Moncef Gabbouj, E.C., Neal Gallagher. *An overview of median and stack filtering*. 1992.
10. Joseph Redmon, S.D., Ross Girshick, Ali Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*. 2016.
11. tutorialspoint. *Sobel Operator*. Available from: [https://www.tutorialspoint.com/dip/sobel\\_operator.htm&cd=4&hl=th&ct=clnk&gl=th](https://www.tutorialspoint.com/dip/sobel_operator.htm&cd=4&hl=th&ct=clnk&gl=th).
12. Sahir, S. *Canny Edge Detection Step by Step in Python — Computer Vision*. 2019; Available from: <https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123>.

13. tutorialspoint. *Laplacian Operator*. Available from: [https://www.tutorialspoint.com/dip/laplacian\\_operator.htm&cd=29&hl=th&ct=clnk&gl=th](https://www.tutorialspoint.com/dip/laplacian_operator.htm&cd=29&hl=th&ct=clnk&gl=th).
14. Wenzhe Shi, et al., *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*, in *CVPR 2016*. 2016.
15. Rajeev Yasarla, Vishwanath Sindagi, and V.M. Patel, *Semi-Supervised Image Deraining using Gaussian Processes*. 2020.
16. Dongwei Ren, et al., *Progressive Image Deraining Networks: A Better and Simpler Baseline*, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.
17. Kui Jiang, et al., *Multi-Scale Progressive Fusion Network for Single Image Deraining*, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
18. Tianyu Wang, et al. *Spatial Attentive Single-Image Deraining With a High Quality Real Rain Dataset*. in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019*. Long Beach, CA, USA.
19. Xueyang Fu, et al., *Clearing the Skies: A deep network architecture for single-image rain removal*. *IEEE Transactions on Image Processing*, 2017.
20. Hong Wang, et al., *A Model-Driven Deep Neural Network for Single Image Rain Removal*, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
21. Qiaosi Yi, et al., *Structure-Preserving Deraining with Residue Channel Prior Guidance*. 2021.

ภาคผนวก







ภาคผนวก ก

ผลงานที่ได้รับการตีพิมพ์



## CERTIFICATE OF PARTICAPATION

This is to certify that

**Patsakorn Akephachaisawat, Soapon Phumeechanya**

has successfully presented a paper entitled

**A Self-Augmentation Transfer Learning Network for Image Deraining**

in the 6th International Conference on Information Technology (InCIT2022),  
organized by Council of IT Deans in Thailand (CITT), during 10-11, November 2022  
at Panyapiwat Institute of Management (PIM), Nonthaburi, Thailand.

(Assoc.Prof.Dr. Pisit Charnkeitkong)

President, Council of IT Deans in Thailand (CITT)



IEEE  
COMPUTER  
SOCIETY  
Thailand Chapter



IEEE  
THAILAND SECTION



The 6<sup>th</sup> International Conference on Information Technology (InCIT2022)

**InCIT2022**  
@PIM, Nonthaburi

10<sup>th</sup> – 11<sup>th</sup> November 2022

Panyapiwat Institute of Management  
Nonthaburi, Thailand

## CALL FOR PAPERS

### OVERVIEW

The 6<sup>th</sup> International Conference on Information Technology (InCIT2022) is a premier forum for sharing research in areas related to information and communication technologies. We believe that fostering research encompassing intelligent technology and innovation for the future of society is vital. Authors involved in those research areas are cordially invited to submit papers and present them at InCIT2022. The conference will be an ideal opportunity to strengthen collaboration between researchers. It will provide many excellent opportunities for participants to exchange and discuss new innovative ideas and research results, as well as enabling exploration of future directions for cooperative research.

### SCOPE

- Artificial Intelligence
- Cloud Service and Computing
- Computer Animation and Game
- Database Technology
- Geo-informatics
- IT in Education
- IT Security and Privacy
- Intelligence Communications
- Network Security and Privacy
- Pattern Recognition
- Quantum Computing
- Natural Language Processing
- Smart and Expert Systems
- Wireless and Mobile Networks
- Data Science and Analytics
- Communications and Networking
- Digital Multimedia Technology
- E-Commerce, E-Education, E-Industry, E-Society
- Human Computer Interaction
- IT and Project Management
- IT and Mobile Application
- Internet of Things
- Augmented and Virtual Reality
- Platform Technologies
- Signal Processing
- Bio-medical Informatics
- Web and Internet Technologies
- Other Related Topics

### CONTACT INFORMATION

Faculty of Engineering and Technology  
Panyapiwat Institute of Management (PIM)  
85/1 Moo 2 Chaengwattana Rd., 11120  
Phone: +662 855 0327  
E-mail: [incit2022@citt.ac.th](mailto:incit2022@citt.ac.th)



<https://www.citt.or.th/incit2022>

### IMPORTANT DATE

Paper Submission Deadline	Notification of Acceptance	Camera-Ready Submission	Registration	Conference Dates
30 <sup>th</sup> June 2022	31 <sup>st</sup> July 2022	31 <sup>st</sup> August 2022	1 <sup>st</sup> - 30 <sup>th</sup> September 2022	10 <sup>th</sup> - 11 <sup>th</sup> November 2022



# A Self-Augmentation Transfer Learning Network for Image Deraining

Patsakorn Akephachaisawat  
 Department of Electrical Engineering  
 Faculty of Engineering and Industrial Technology  
 Silpakorn University  
 Nakhon Pathom, Thailand  
 akephachaisawat\_p2@su.ac.th

Sopon Phumeechanya  
 Department of Electrical Engineering  
 Faculty of Engineering and Industrial Technology  
 Silpakorn University  
 Nakhon Pathom, Thailand  
 phumeechanya\_s@su.ac.th\*  
 \*Corresponding author

**Abstract**— This paper proposes a self-augmentation transfer learning network, a new framework for removing rain from images that includes autoencoder, improve block, and transfer learning as key components. The first step is to remove rain using both non-block-based autoencoder and block-based autoencoder. Following that is the improve block step, which was designed to improve the efficiency of rain removal. The image will then be analyzed to identify rain streaks for additional training in the transfer learning autoencoder step. The results will then be run through the improve block again. In the experiment, we used PSNR and SSIM to measure the efficacy of rain removal using 150 test images. On average, our method yielded a PSNR of 32.43 and a SSIM of 0.935.

**Keywords**— image deraining; autoencoder; transfer learning; image augmentation; convolutional neural network

## I. INTRODUCTION

Image deraining is an essential preliminary step in computer vision that involves the processing of images with rain. Deep learning is now used in the design of image deraining methods to increase the efficiency of tasks such as object detection, image segmentation, image recognition, and so on. Convolutional neural networks (CNNs) are used in a variety of frameworks to remove rain and restore image quality in rain streak-observed areas. In addition, the intensity of the rain is the main factor and challenge of this work to achieve satisfactory output image.

In this paper, we design a novel framework for image deraining. There are three main components in this framework: autoencoder, improve block, and transfer learning. The first two components are in charge of preliminary removing rain from images and extracting the rain streak areas. In our image augmentation process, the additional rain streak images are created from the extracted rain streak areas together with images that have already been initially removed and are used in the transfer learning step. So we call the framework we proposed as a self-augmentation transfer learning network.

This paper proposes the SATLNet method for removing rain streak from the image. The method is designed to handle with two main issues.

- Handle with the rain streak that is large and opaque.
- Keep the tone of the output image the same as the input image without distortion of brightness and contrast.

## II. RELATED WORK

There have been various studies that have developed methods for removing rain from images. Each approach has its own advantages and similar components. The majority of them are based on deep learning and can be divided into two types: general CNN-based and autoencoder-based.

### A. General CNN-based method

Many researches focus on the design of the CNN model to remove the rain from the image [1], [2], [3], [4], [5] which is difficult and the complexity of designing structures for effective rain removal. As a result, some research handles on the input image before importing it into the CNN model. For example, MSPFN [6] minimizes the input image several sizes and then imports to the CNN model. DragNet [7] extracts input images using low-pass filters and high-pass filters before removing rain with the CNN Model. RCDNet [8] extracts all rain streaks to determine the details and patterns of the rain, and then performs the procedure of removing the rain. SPDNet [9] recreates the image channel as a way to prevent image details that may be lost during the rain removal process. JORDER [10] extracted key features of rain from rain images, and then entered the CNN model.

### B. Autoencoder-based method

Since the autoencoder method can be used to eliminate noise in images, there is a lot of research that uses this method to remove rain, which is seen as a form of noise in images. For example, gp-based

978-1-6654-8912-6/22/\$31.00 ©2022 IEEE

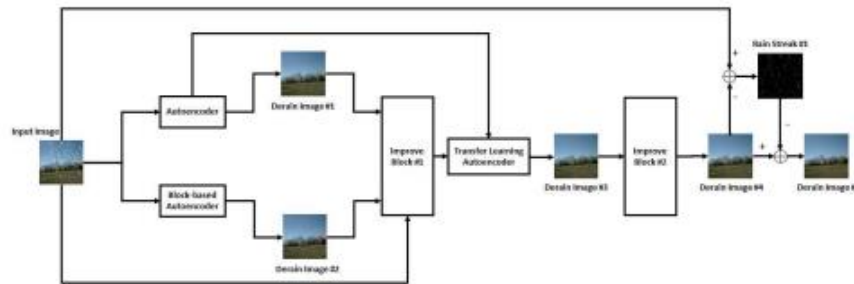


Figure 1. The framework of our self-augmentation transfer learning network (SATLNet).

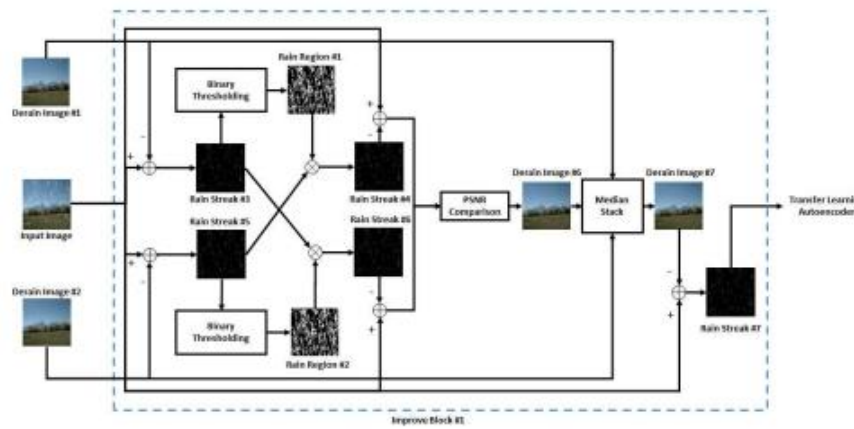


Figure 2. Improve block#1.

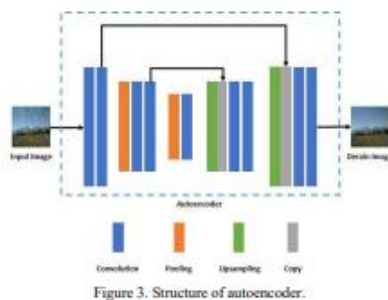


Figure 3. Structure of autoencoder.

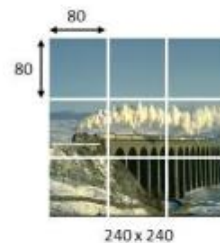


Figure 4. The input image is divided into 9 blocks (3x3).

SSL framework [11] uses autoencoder trained in both synthetic rain images and actual rain images. This makes it independent to remove rain in many forms. GCANet [12] was developed for fog removal and can also be used to remove rain. The structural design of this method uses the principle of autoencoder, switching from a straight layer to a method called Smoothed Dilated Resblock [13], which solves the problem of gridding artifacts of images from dilated convolutions. MPRNet [14] is a method of subtracting rain by dividing it into multiple states, each degrading the input image in multiple sizes, and using an autoencoder to learn contextualized features and then bring the resulting details together in the next stage.

### III. THE PROPOSED METHOD

This section describes our proposed method, called a self-augmentation transfer learning network, abbreviated as SATLNet. The structure of the framework, as shown in Fig. 1. The framework consists of three key parts: autoencoder, improve block, and transfer learning, with the first part being the rain remove using two types of autoencoders: non-block-based and block-based. The reason why both types of autoencoders are needed is because the ability to remove rain has different advantages and disadvantages. The first autoencoder section clearly focuses on removing rain, but the background details are also reduced. On the other hand, the second autoencoder contributes to the matter of maintaining the background, but removing the rain is not good enough. Once the results from the two autoencoder are obtained, it enters improve block#1 that was created to improve rain removal performance by taking advantage of the previous two rain removal results. The resulting images are extracted for rain streaks to be used for further training in the transfer learning autoencoder section. The resulting image, then goes through improve block#2, which has a slightly different detail to improve block#1, which is explained later. At the very end of the SATLNet framework, it increases the efficiency of rain removal by applying the resulting results to the input image, finding the rain streaks of the input image, and then re-subtracting it with the previous result image. This will help to return some of the background details of the image that were lost in the previous step.

#### A. Non-Block-based Autoencoder

Autoencoder is a type of neural network that is designed to learn the input image that is received to be the same as the output image, so it can be used to remove noise in the image. The structure of the autoencoder we use for removing rain is shown in Fig. 3. It has a total of 15 layers. In the first part is an encoder with 5 convolution layers. Next, it is a section of latent layer consisting of 2 layers. Then enter to the decoder consisting of 7 convolution layers. In some layers, there is a concatenation between the encoder and decoder to prevent data loss along the way.

#### B. Block-based Autoencoder

Using the Block-base technique is one way to help reconstruct the image. It divides the original image into sections of the same size as shown in Fig. 4. In this work, we used 240×240 input images, subdivided into 9 blocks with a size of 80×80. The reason why an image size of 80×80 is required is because the rain streaks are a large noise. If we use a small image size, the rain streaks are separated until they look ambiguous. There may also be a lot of blocking artifacts. As a result of the test, an image division of 80×80 was selected, which is the appropriate size of the input image. The structure of the block-based autoencoder is the same as the non-block-based autoencoder. The results can remove the rain well to some extent, but the background detail is better done than the non-block-based autoencoder.

#### C. Improve Block

In our framework, two improve blocks are used, which have a slightly different structure. The first block, called improve block#1, as shown in Fig 2., extracts the key points of the resulting image from both the non-block-based autoencoder and block-based autoencoder steps. It switches the rain streak characteristics of each step and compares the performance to deliver the best results to the next process. The improve block#1 has received three input images: input image, derain image#1, which is derived from non-block-based autoencoder, and derain image#2, which is processed from block-based autoencoder. Next, the rain streak images of the two images had to be found by deleting them with the input image. After that, take them through the binary thresholding process to obtain binary rain streak images called rain region. At this stage, we get two rain regions, rain region#1 and rain region#2, both of which are taken to AND with rain streak images again. The rain region#1 will be AND with rain streak image#5 and rain region#2 will be AND with rain streak image#3. Occasionally, the results obtained from the non-block-based autoencoder and block-based autoencoder have different details and characteristics of rain removal. We took the rain region to AND with the rain streak to preserve the location of the rain streak and eliminate unnecessary background details. After that, the two rain streak images are taken back to subtract with the original rain image to get result images without rain streaks. The result images are imported into the PSNR comparison to compare the performance of which images are better and are sent as output images to the median stack process. The median stack step, which is the determination of the median value of the pixels that are nested between the derain images#1, #2, #6, to increase the efficiency of rain removal. Next, the obtained output image will be subtracted from the original image for rain streaks as in rain streak#7 to be sent to the transfer learning autoencoder process.

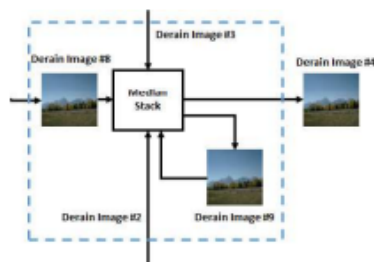


Figure 5. Some part of the improve block#2.

Some part of the improve block#2 is shown in Fig. 5, which has only slightly different details when compare to the improve block#1. The input image is the derain image#3, which is the resulting image of the transfer learning autoencoder. In addition, the result of the PSNR Comparison is the derain image #8, which is directed to the median stack stage with the derain image#3 and the derain image#2 as the result of removing rain through the block-based autoencoder. Next, we get the image as the derain image#9 and then put it back in the median stack again, resulting in the removal of the rain as in the derain image#4.

#### D. Transfer Learning Autoencoder

Transfer learning is about learning from previously trained models to adapt them to the data we need. The purpose of this section is to train images from the dataset that contain only rain images with the same characteristics of rain as the input images. It begins by applying weight from a non-block-based autoencoder model to transfer learning process, with the dataset being trained from image augmentation using rain streaks extracted from the resulting image in the stage of the improve block#2. For this reason, we call this a self-augmentation transfer learning technique. The dataset that is used in this process has the same rain streak characteristics as the input image. As a result, the efficiency of rain removal has improved.

## IV. EXPERIMENTS

### A. Datasets

In the experiment, we used 368 synthesized rain images from the Rain100L and Rain100H [10] datasets. The 200 images from Rain100L, and the 168 images from Rain100H are used in the training process. In the image augmentation step for the transfer learning process, only images from the Rain100L database are used. In testing, we used a total of 150 synthetic rain images, including: 100 test images from rain100L and selected 50 images from the dataset [7]. To test on real rain images, we used the input images from Rain800 [15] dataset.

### B. Metrics

To measure the effectiveness of our framework, we used the Peak Signal-to-Noise Ratio (PSNR) and

the structural similarity (SSIM), which are calculated to compare the resulting output image with the ground truth image. As for testing with real rain images, we don't have ground truth, so we can only make comparisons based on human view of how well the rain has been removed from the image.

### C. Training setting

There are several autoencoder parts in the SATLNet framework. For the training process, the Adam optimizer is used with a learning rate of 0.001, a batch size of 16, and an epoch of 500. Only the last three layers are trained with an epoch of 100 for the transfer learning step.

### D. Results of each stage in our framework

This section shows the effectiveness of each stage in our framework.

TABLE I. RESULTS OF EACH STAGE IN OUR FRAMEWORK WHEN TESTING ON ONLY A IMAGE.

Stage	Metrics	
	PSNR	SSIM
Input (1 image:rock)	31.03	0.8677
Non-block-based autoencoder	37.49	0.9618
Block-based autoencoder	37.09	0.9539
Improve block#1	37.58	0.9588
Transfer learning autoencoder	38.42	0.9651
Improve block#2	38.74	0.9671
Output	38.96	0.9691

TABLE II. RESULTS OF EACH STAGE IN OUR FRAMEWORK WHEN TESTING ON 150 IMAGES.

Stages	Metrics	
	PSNR	SSIM
Input (150 images)	26.63	0.8250
Non-block-based autoencoder	30.71	0.9280
Block-based autoencoder	31.08	0.9195
Improve block#1	31.46	0.9241
Transfer learning autoencoder	32.12	0.9369
Improve block#2	32.40	0.9395
Output	32.43	0.9395

Table 1 is a comparison of performance by using the rock image in Fig. 6 (the first row) as examples to show PSNR and SSIM values to compare the rain removal capacity of each stage. Starting with the non-block-based autoencoder versus the block-based autoencoder, it is clear that the PSNR and SSIM values have increased and in some areas of the output image still leave little trace of rain. Next is the stage of improve block#1, which can further increase the efficiency of removing rain from the image following the autoencoder's procedure. When entering the transfer learning autoencoder stage, it was found that it can further enhance performance, since it is trained using rain extracted directly from the previous step. After that, the image is loaded into the improve

block#2 to remove the rain as the final step in producing the output image.

Table 2 shows a comparison of the performance of the rain removal capacity of each step by showing the average of PSNR and SSIM from a total of 150 test images. It is clear that each step of our framework contributes to a continuous increase in the average of PSNR and SSIM.

#### E. Comparison

Table 3 shows a comparison of average PSNR and SSIM values from a total of 150 test images, comprising 100 images from the Rain100L dataset and 50 images from the dataset [6]. From the averages of input images, GCANet, and SATLNet, it is clear that our methods provide higher PSNR and SSIM values than GCANet. It has the ability to increase the PSNR value from the input image to approximately 6db and about 5db more than GCANet.

TABLE III. EXPERIMENTAL RESULTS ON 150 IMAGES FROM RAIN100L AND DATASET [7]

Dataset (150)	Rain100L (100) + dataset[7] (50)	
Metrics	PSNR	SSIM
Input	26.63	0.8250
GCANet [10]	27.19	0.8579
SATLNet	32.43	0.9395



Figure 6. Comparisons of GCANet, SATLNet for rock, snake, mountain and giraffe rainy images.

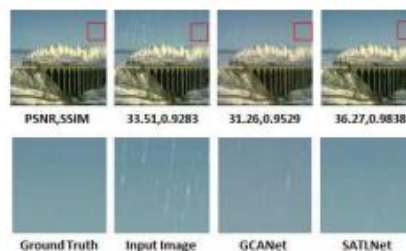


Figure 7. Comparisons of GCANet, SATLNet for train rainy image.

Fig. 6 shows a comparison of the effectiveness of rain removal by various methods, with the first column showing the original image without rain. The second column is a image with rain that needs to be removed. The third column is the result of removing rain using GCANet, and the last column is our rain removal method SATLNet. There are 4 examples of rocks, snakes, mountains and giraffes. The results of GCANet show that it could help remove rain to some extent, but the PSNR value in some images is less than the input image because this method distorts the colors from the input image. Whereas our SATLNet method can provide output images where the rain is reduced and retains the same tone as the input images. From the stone image, the difference in tone is clearly visible. The color of the output image obtained from GCANet is darker than the input image and the output image from our method. From the snake image, the SSIM value of our method is high, while GCANet provides a smaller SSIM value than the input image due to the loss of some of the image details. This is observed from the desert wave marks in the upper left corner that disappear.

Fig. 7 shows a train image with a zoom in the sky area of the image to observe the differences in the output image obtained by removing the rain of each method. GCANet provides an output image with a different tone than the input image, which is why PSNR is less, but nevertheless still has a good ability to remove smaller and not thicker rain streaks. GCANet still cannot handle large and opaque rains, leaving a trace of partial rain removal. While our SATLNet does not affect the tone of the output image when the rain is removed, it is also good to remove thick and solid rain.



Figure 8. Test on a real rainy image.

Fig. 8 is an experimental result of real rain images taken from Rain800 dataset by removing rain using GCANet and SATLNet methods. We found that removing rain using GCANet can well remove thin



and small rain streaks, but large and opaque rain streaks cannot be removed. While our SATLNet method is able to remove large and thick rain streaks, only a small amount of rain remains.

#### F. Testing time

The computational time for 150 images is investigated during testing. The GCANet method takes about 30 seconds, but the SATLNet method takes about 300 minutes. This is due to the image augmentation process that takes place during the transfer learning step.

#### V. CONCLUSION

We propose a framework for removing rain from images called SATLNet. Both non-block-based autoencoder and block-based autoencoder are used to remove rain in the first step. Then enter the stage of improve block, which improves the efficiency of rain removal by comparing the details of previous rain removals to obtain a well-detailed output image. The output image will be used in the process of finding rain streaks for image augmentation step. The transfer learning process uses only the extracted rain streaks for the input image to be re-trained by the same model in the first step. Finally, it will be re-entered into the improve block to increase efficiency, so that the rain in the image will be removed. Our SATLNet method has good ability to remove both thin and thick rain streaks. In addition, the color of the output image is not distorted when compares to the input image, resulting in PSNR and SSIM values are high.

#### ACKNOWLEDGMENT

The authors would like to acknowledge the contribution of Department of Electrical Engineering, Faculty of Engineering and Industrial Technology, Silpakorn University Scholarship for funding this research.

#### REFERENCES

- [1] W. Shi et al., "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1874-1883.
- [2] D. Ren, W. Zuo, Q. Hu, P. Zhu and D. Meng, "Progressive Image Deraining Networks: A Better and Simpler Baseline," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3932-3941.
- [3] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang and R. W. H. Lau, "Spatial Attentive Single-Image Deraining With a High Quality Real Rain Dataset," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12262-12271.
- [4] S. Zheng, C. Lu, Y. Wu and G. Gupta, "SAPNet: Segmentation-Aware Progressive Network for Perceptual Contrastive Deraining," 2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), 2022, pp. 52-62.
- [5] S. Anwar and N. Barnes, "Real Image Denoising With Feature Attention," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 3155-3164.
- [6] K. Jiang et al., "Multi-Scale Progressive Fusion Network for Single Image Deraining," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8343-8352.
- [7] X. Fu, J. Huang, X. Ding, Y. Liao and J. Paisley, "Clearing the Skies: A Deep Network Architecture for Single-Image Rain Removal," in IEEE Transactions on Image Processing, vol. 26, no. 6, pp. 2944-2956, June 2017.
- [8] H. Wang, Q. Xie, Q. Zhao and D. Meng, "A Model-Driven Deep Neural Network for Single Image Rain Removal," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 3100-3109.
- [9] Q. Yi, J. Li, Q. Dai, F. Fang, G. Zhang and T. Zeng, "Structure-Preserving Deraining with Residue Channel Prior Guidance," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 4218-4227.
- [10] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan and J. Liu, "Joint Rain Detection and Removal from a Single Image with Contextualized Deep Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 6, pp. 1377-1393, 1 June 2020.
- [11] R. Yasarla, V. A. Sindagi and V. M. Patel, "Semi-Supervised Image Deraining Using Gaussian Processes," in IEEE Transactions on Image Processing, vol. 30, pp. 6570-6582, 2021.
- [12] D. Chen et al., "Gated Context Aggregation Network for Image Dehazing and Deraining," 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 1375-1383.
- [13] Z. Wang and S. Ji, "Smoothed dilated convolutions for improved dense prediction," In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2486-2495. ACM, 2018.
- [14] S. W. Zamir et al., "Multi-Stage Progressive Image Restoration," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
- [15] H. Zhang, V. Sindagi and V. M. Patel, "Image De-Raining Using a Conditional Generative Adversarial Network," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 11, pp. 3943-3956, Nov. 2020.

## ประวัติผู้เขียน

ชื่อ-สกุล	พัสกร เอกผาชัยสวัสดิ์
วัน เดือน ปี เกิด	12 พฤศจิกายน 2540
สถานที่เกิด	ราชบุรี
วุฒิการศึกษา	วศ.บ.วิศวกรรมอิเล็กทรอนิกส์และระบบคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากร (2562)
ที่อยู่ปัจจุบัน	449/97 ถ.สวนตะไคร้ ต.สนามจันทร์ อ.เมืองนครปฐม จ.นครปฐม
ผลงานตีพิมพ์	Patsakorn Akephachaisawat and Sopon Phumeechanya, "A Self-Augmentation Transfer Learning Network for Image Deraining," 2022 6th International Conference on Information Technology (InCIT), Nonthaburi, Thailand, 2022, pp. 384-389.

