

ข้อเสนอโครงการวิศวกรรมไฟฟ้า วิชา 2102499

การเรียนรู้การให้ผลของแสงเพื่อสร้างภาพท้องฟ้า

Optical Flow Learning for Sky Image Generation

นายศรัณย์ภาร เมฆสกุลรุ่ง เลขประจำตัวนิสิต 6330509621

อาจารย์ที่ปรึกษา อ.ดร. สุวิชญา สุวรรณวิมลกุล

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2566

ลงชื่ออาจารย์ที่ปรึกษาหลัก

.....
นรธน
.....

(อ.ดร. นรธน พุฒานนวมณฑล)

วันที่ 26 เมษายน 2567

บทคัดย่อ

เมฆเป็นหนึ่งในปัจจัยหลักที่มีผลต่อความไม่แน่นอนในการประมาณค่ารังสีของแสงดวงอาทิตย์ โดยโครงงานนี้ได้นำเสนอวิธีการสร้างภาพถ่ายท้องฟ้าในอนาคตของคณะวิศวกรรมศาสตร์สาขาไฟฟ้า โดยใช้การเรียนรู้เชิงลึก และ ทฤษฎีเพิ่มเติมที่ช่วยในการทำนายภาพถ่ายท้องฟ้า เพื่อให้ได้ผลลัพธ์มีความแม่นยำและเหมาะสมกับความจริง โดยโครงงานนี้ได้ทำการฝึกฝนและทดสอบโมเดล SkyUNet, PhyDNet + GAN และ ConvLSTM สำหรับทำนายภาพถ่ายท้องฟ้า และ โมเดล SUNSET, Solarnet และ UNET สำหรับการทำนายรังสีแสงอาทิตย์ โดยจากการทดสอบโมเดล พบร่วมกับ PhyDNet และ โมเดล UNET มีประสิทธิภาพและความแม่นยำมากกว่าโมเดลอื่น ตามประเภทของงานที่ทำนาย

คำสำคัญ: การสร้างภาพถ่ายท้องฟ้าในอนาคต, การเรียนรู้เชิงลึก

Abstract

Cloud is the main factors of solar irradiance forecasting uncertainty. This project proposes a sky image generation of Chula Electrical Engineering's sky for accurate and realistic images by using deep learning and others theory for sky image. testing on SkyUNet, PhyDNet + GAN, ConvLSTM for sky image generation model and SUNSET, Solarnet, UNET for the solar irradiance forecasting model. The testing result showed that PhyDNet+GAN and UNET model outperformed others model in their tasks.

Keywords: Sky Image Generation, Solar Irradiance Forecasting, Deep Learning

สารบัญ

1	บทนำ	5
1.1	ที่มาและความสำคัญของโครงการ	5
1.2	วัตถุประสงค์ของโครงการ	5
1.3	ขอบเขตของโครงการ	5
1.4	ผลลัพธ์ที่คาดหวังจากโครงการ	6
2	โครงข่ายประสาทพื้นฐาน (Basic of Neural Network)	7
2.1	เครือข่ายประสาทแบบคอนโวลูชัน (Convolution Neural Network - CNN)	7
2.2	เครือข่ายประสาทความทรงจำระยะสั้น-ยาว (Long Short Term Memory - LSTM)	7
2.3	เครือข่ายคอนโวลูชันแบบความจำระยะสั้น-ยาว (Convolutional Long Short Term Memory - ConvLSTM)	10
3	การประยุกต์การเรียนรู้ไมโครไฟฟ้าสำหรับสร้างภาพถ่ายท้องฟ้า (Sky-Image Generation Learning)	11
3.1	สถาปัตยกรรมเครือข่ายแบบสกายยูเน็ต (SkyUNet Network)	11
3.1.1	ตัวเข้ารหัส (Encoder)	11
3.1.2	ตัวเชื่อมต่อ (Skip Connection)	12
3.1.3	ตัวถอดรหัส (Decoder)	12
3.1.4	คอขวด (Bottle Neck)	13
3.1.5	การเรียนรู้แบบ Supervised Learning	13
3.2	โมเดลไนมิกทางกายภาพ (PhyDNet)	14
3.2.1	PhyDNet Disentangling Architecture	15
3.2.2	เครือข่าย PhyCell	16
3.2.3	การเรียนรู้สมการทางกายภาพของ PhyCell ด้วย Moment Loss	17
3.2.4	การเรียนรู้แบบคู่ต่อสู้ช่วยสร้าง (Generative Adversarial Learning)	17
4	ประยุกต์การเรียนรู้ไมโครไฟฟ้าในการประมาณค่ารังสีแสงอาทิตย์โดยตรงจากอนุกรมของภาพถ่าย (Direct Irradiance Forecasting Approaches)	19
4.1	สถาปัตยกรรมเครือข่ายแบบชั้นเชือต (SUNSET)	19
4.2	สถาปัตยกรรมเครือข่ายแบบโซล่าเน็ต (Solarnet)	20
4.3	สถาปัตยกรรมเครือข่ายแบบยูเน็ต (UNET)	21
5	ผลลัพธ์จากการดำเนินการ	23
5.1	การเตรียมชุดข้อมูล (Data Preparation)	23
5.1.1	การจัดชุดข้อมูล (Data Partition)	24
5.2	การฝึกฝนโมเดลทำนายภาพถ่ายท้องฟ้า (Sky Images Model Training)	24

5.2.1	SkyUNet Model	24
5.2.2	ConvLSTM Model	25
5.2.3	PhyDNet + GANs Model	26
5.2.4	สรุปผลลัพธ์ที่ได้จากการฝึกฝนโมเดลสำหรับทำนายภาพถ่ายห้องฟ้า	27
5.3	การฝึกฝนโมเดลทำนายค่ารังสีแสงอาทิตย์ (Irradiance Forecasting Model Training)	27
5.3.1	การฝึกฝนโมเดล SUNSET	27
5.3.2	การฝึกฝนโมเดล Solarnet	28
5.3.3	การฝึกฝนโมเดล UNET	29
5.3.4	สรุปผลลัพธ์ที่ได้จากการฝึกฝนโมเดลสำหรับทำนายค่ารังสีแสงอาทิตย์	30
5.4	วิธีการวัดประสิทธิภาพ	31
5.4.1	อัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (PSNR: Peak Signal to Noise Ratio) . .	31
5.4.2	ค่าเฉลี่ยผลต่างกำลังสอง (MSE: Mean Squared Error)	31
5.4.3	ค่าเฉลี่ยผลต่างสัมบูรณ์ (MAE : Mean Absolute Error)	32
5.5	ผลจากการทดสอบโมเดล (Testing Model)	32
5.5.1	ผลลัพธ์จากการทดสอบโมเดลทำนายภาพถ่ายห้องฟ้า	32
5.5.2	ผลลัพธ์จากการทดสอบโมเดลทำนายรังสีแสงอาทิตย์	35
6	บทสรุป	38
6.1	สรุปผลการดำเนินการ	38
6.2	สรุปองค์ความรู้ที่ได้จากการโครงงาน	38
6.3	ปัญหา อุปสรรค และแนวทางการแก้ไข	38
6.4	กิจกรรมประการ	38

1 บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ในแต่ละวันโลกได้รับแสงแดดประมาณ 1366 วัตต์ (Watt) โดยการที่เราสามารถนำพลังงานจากแสงแดด (Solar Energy) ที่แฝงมาอยู่ในพลังงานให้เป็นพลังงานได้โดยไม่มีการสูญเสีย จึงทำให้พลังงานจากแสงอาทิตย์นั้นเป็นพลังงานที่มีความสำคัญและประโยชน์ในการอนุรักษ์พลังงาน [1]

หนึ่งในปัจจัยหลักที่ส่งผลต่อการประมาณค่าของรังสีของแสงอาทิตย์ (Solar Irradiance) ที่ส่องมายังบนพื้นโลกนั้นคือเมฆ เนื่องจากเมฆสามารถบดบังแสงแดดและลดปริมาณรังสีที่แผ่จากด้วยอาทิตย์ได้ และยังมีความไม่แน่นอนทางด้านปริมาณ รูปร่าง และ ความหนา จึงส่งผลให้เกิดปัญหาความไม่แน่นอนในการประมาณค่ารังสีของแสงอาทิตย์ หนึ่งในแนวทางการแก้ไขปัญหาความไม่แน่นอนนี้ สามารถทำได้โดยการเฝ้าดูทิศทางการเคลื่อนไหวของเมฆ (Cloud Movement) จากภาพถ่ายท้องฟ้า (Sky Image) เพื่อนำมาใช้ในทำนายค่ารังสีจากดวงอาทิตย์มีความแม่นยำมากยิ่งขึ้น [2, 3, 4, 5] ในปัจจุบันนี้ได้มีการนำความรู้ที่เกี่ยวกับการเรียนรู้เชิงลึก (Deep Learning) มาใช้ในการงานการประมาณค่าของรังสีของแสงอาทิตย์ ซึ่งจากการวิจัยล่าสุด [6] พบร่วมกัน ภาพถ่ายท้องฟ้าสามารถประมาณค่าของรังสีของแสงอาทิตย์ ได้แม่นยำมากกว่าภาพถ่ายจากดาวเทียม (Satellite Image) เนื่องจากการวางแผนการติดตั้งของกล้องภาพถ่ายท้องฟ้า (Sky Imager) และเขียนเชอร์วัดค่ารังสีของแสงอาทิตย์นั้น มักจะอยู่ในระยะที่ใกล้กัน และเก็บข้อมูลร่วมกันในบริเวณเดียวกัน

การทำนายค่าของรังสีของแสงอาทิตย์จากภาพถ่ายท้องฟ้า มีสองรูปแบบกล่าวคือ

1. การสร้างรูปแบบการการเคลื่อนไหวของเมฆ (Image Generation Approaches) แล้วนำภาพที่ถูกสร้างมาประมาณค่าพลังงานแสงอาทิตย์ เช่น SkyNet-Unet [2], ConvLSTM [7], PhyDNet [8], และ SkyGPT [9]
2. การประมาณค่าพลังงานแสงอาทิตย์โดยตรงจากอนุกรมของภาพถ่าย (Direct Irradiance Forecasting Approaches) เช่น Sunset [10], Solarnet [11], และ Unet [9]

อย่างไรก็ตามมีงานวิจัยไม่กี่ชิ้นที่วิเคราะห์ประสิทธิภาพของทั้งสองวิธี ดังนั้นโครงการนี้จึงถูกจัดทำขึ้นเพื่อศึกษาถึงความแตกต่างของทั้งสองรูปแบบ ทั้งนี้ เนื่องจากวิธีการสร้างรูปแบบการการเคลื่อนไหวของเมฆ สามารถนำไปต่อยอดได้ในงานอื่น เช่น การตรวจจับเมฆ การหาตำแหน่งดวงอาทิตย์ ข้าพเจ้าวัดผลโดยดูจากความถูกต้องของรูปภาพที่ได้ในอนาคต ในขณะที่การประมาณค่าพลังงานแสงอาทิตย์โดยตรงจะเป็นการวัดค่าความถูกต้องจากค่าของรังสีของแสงอาทิตย์

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาและออกแบบโมเดลสำหรับการสร้างภาพถ่ายท้องฟ้าในอนาคตและทำนายค่ารังสีแสงอาทิตย์
2. เพื่อพัฒนาและปรับปรุงโมเดลที่สามารถทำนายค่ารังสีแสงอาทิตย์จากข้อมูลภาพถ่ายท้องฟ้าและค่ารังสีแสงอาทิตย์ได้
3. เพื่อสร้างโมเดลที่สามารถสร้างภาพในอนาคต โดยที่วัตถุในภาพมีความอรูป เช่น การไฟลของเมฆ และ หมอก

1.3 ขอบเขตของโครงการ

1. โครงการนี้จะมุ่งเน้นการทำนายทางทิศทางการเคลื่อนไหวของเมฆเพื่อทำนายภาพถ่ายท้องฟ้า

2. โครงการนี้จะมุ่งเน้นการหาผลลัพธ์ตั้งกล่าวด้วยวิธีของการเรียนรู้เชิงลึก (Deep Learning)
3. ข้อมูลที่ใช้สำหรับการเรียนรู้ของโมเดลในโครงการนี้คือ ภาพถ่ายของห้องพักในแต่ละวันในช่วงเวลา 5.00 น. จนถึง 18.59 น. ถ่ายจากคณะวิศวกรรมศาสตร์สาขาไฟฟ้า โดยรูปภาพทั้งหมดมีขนาด 1080×1920 pixels และในเวลาที่แตกต่างกันในทุกๆ 1 นาที ซึ่งข้อมูลจากแหล่งทั้งสองมีข้อจำกัดคือไม่มีค่าที่แท้จริงในการให้ของแสง (Ground Truth Optical Flow)
4. โครงการนี้จะมุ่งเน้นวิธีการแก้ปัญหาการเรียนรู้ในกรณีที่ไม่มีข้อมูลจากค่าที่แท้จริงของการให้ของแสง

1.4 ผลลัพธ์ที่คาดหวังจากโครงการ

1. การเปรียบเทียบประสิทธิภาพระหว่างการสร้างรูปแบบการการเคลื่อนไหวของเมฆ (Image Generation Approaches) และ การประมาณค่าพลังงานแสงอาทิตย์โดยตรงจากอนุกรมของภาพถ่าย (Direct Irradiance Forecasting Approaches)
2. การเปรียบเทียบประสิทธิภาพของการสร้างรูปแบบการการเคลื่อนไหวของเมฆ
3. วิธีการของเทคนิคที่ใช้ในการเรียนรู้เชิงลึกและการเตรียมข้อมูล

2 โครงข่ายประสาทพื้นฐาน (Basic of Neural Network)

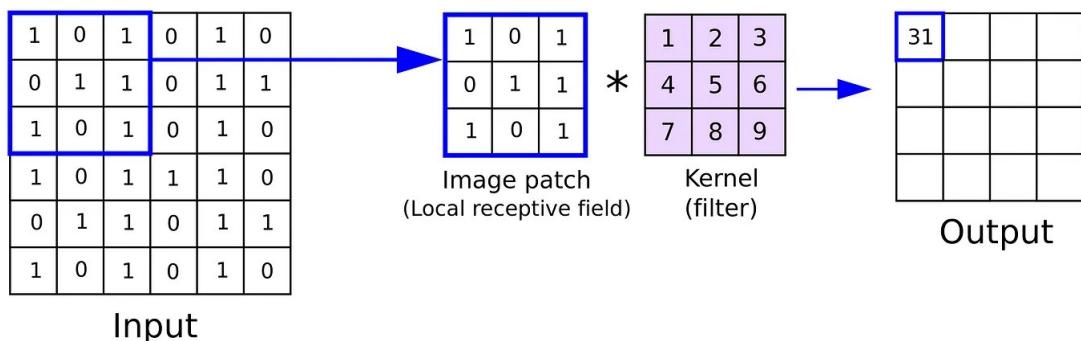
ในบทต่อไปนี้จะกล่าวถึงโครงข่ายพื้นฐานที่ถูกนำมาประยุกต์ใช้เป็นส่วนประกอบของโมเดลซึ่งถูกใช้ในการเรียนรู้เชิงลึก (Deep Learning) มาใช้ในการงานการประมาณค่าของรังสีของแสงอาทิตย์ เนื่องด้วยโมเดลเดลส่วนใหญ่เป็นการประยุกต์โมเดลพื้นฐานต่อไปนี้

- เครือข่ายประสาทแบบคอนโวลูชัน (Convolution Neural Network - CNN)
- เครือข่ายประสาทความทรงจำระยะสั้น-ยาว (Long Short Term Memory - LSTM)
- เครือข่ายคอนโวลูชันแบบความจำระยะสั้น-ยาว (Convolutional Long Short Term Memory - ConvLSTM)

โมเดลเหล่านี้มักถูกใช้เป็นองค์ประกอบพื้นฐานสำหรับสถาปัตยกรรมที่ถูกนำไปประยุกต์ในการสร้างรูปแบบการการเคลื่อนไหวของเมฆ (Image Generation Approaches) และ การประมาณค่าพลังงานแสงอาทิตย์โดยตรงจากอนุกรมของภาพถ่าย (Direct Irradiance Forecasting Approaches) ดังนั้นข้อมูลของโมเดลพื้นฐานเหล่านี้จะเป็นองค์ประกอบของการประยุกต์ในบทที่สาม

2.1 เครือข่ายประสาทแบบคอนโวลูชัน (Convolution Neural Network - CNN)

เครือข่ายประสาทแบบคอนโวลูชัน (CNN) คือ สถาปัตยกรรมเครือข่ายที่นิยมใช้การหาแบบแผน (Pattern) ของภาพเพื่อแยกคุณลักษณะต่างๆ (Features Extraction) ของภาพ โดยองค์ประกอบที่สำคัญในการแยกคุณลักษณะของภาพ คือ เคอร์เนล (Kernel) หรือ ตัวกรอง (Filter) ซึ่งทำหน้าที่หาคุณลักษณะของภาพจากการนำตัวกรองไปคำนวณผลรวมของการคูณในระดับสมาชิก (Element Wise Multiplication) กับข้อมูลภาพข้ามเข้า เพื่อโยง (Map) ไปผลลัพธ์ (Output) ตามลำดับในทุกข้อมูลข้ามเข้าดังรูปที่ 1

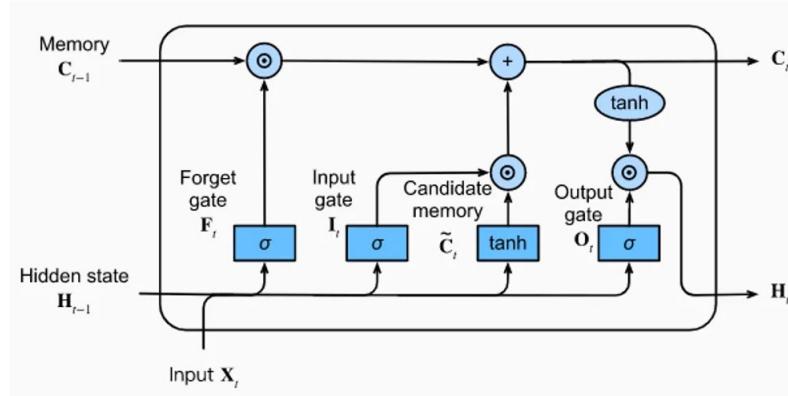


รูปที่ 1: กระบวนการคอนโวลูชัน [12]

2.2 เครือข่ายประสาทความทรงจำระยะสั้น-ยาว (Long Short Term Memory - LSTM)

เครือข่ายประสาท LSTM หรือ Long Short Term Memory เป็นหนึ่งประเภทของโครงข่ายสถาปัตยกรรมแบบนิวรัลเน็ตเวิร์กแบบวนกัน (Recurrent Neural Network หรือ RNN) ซึ่งออกแบบมาเพื่อประมวลผลข้อมูลแบบลำดับ (Sequence) โดยจะนำข้อมูลข้ามออก (Output) ของลำดับก่อนหน้ามาใช้เป็น ข้อมูลป้อนเข้า (Input) และถูกออกแบบมาเพื่อ

แก้ไขปัญหาความชัดสูญหาย (Vanishing Gradients) ของ RNN ที่มีลำดับข้อมูลที่ยาว ซึ่งโครงข่ายนี้สามารถจัดจำแล็บข้อมูลก่อนหน้าได้โดยมีองค์ประกอบที่ควบคุมการทำงานและลักษณะโครงสร้างตั้ง รูปที่ 2 [13, 14]



รูปที่ 2: โครงสร้างสถาปัตยกรรมเครือข่าย LSTM [13]

Input และ Output ของ LSTM

1. Input (X_t) คือ ข้อมูลข้าเข้าจากภายนอก ณ เวลา t
2. Cell State (C_{t-1}) คือ ข้อมูลที่จะจำจากลำดับการหน้าทั้งหมดโดยทำหน้าที่เป็นข้อมูลความจำระยะยาว (Long Term Memory)
3. Hidden State (H_{t-1}) คือ ข้อมูลจากลำดับก่อนหน้าโดยทำหน้าที่เป็นข้อมูลความจำระยะสั้น (Short Term Memory)

โดยที่ Cell State และ Hidden State คือ ข้อมูลข้าออกที่เดิมจาก LSTM ณ ลำดับเวลา ก่อนหน้า หรือ ณ เวลา $t - 1$ [13]

ประตู (Gate) ทำหน้าที่เป็นตัวเลือกข้อมูล หรือ เวกเตอร์ตัวเลือก (Selector Vector) โดยที่แต่ละประตูจะพิงก์ชันกระตุ้น (Activation Function) เป็นพิงก์ชันซิกมอยด์ (Sigmoid Function) ทำให้ได้ผลลัพธ์อยู่ในช่วง 0 ถึง 1 เพื่อนำไปคูณกับเวกเตอร์ข้อมูลต่างๆ ต่างๆ ภายใน LSTM ดังนี้

1. ประตูการลืม (Forget Gate) ทำหน้าที่เป็นเวกเตอร์ตัวเลือกในการจดจดหรือลบข้อมูลของ Cell State ลำดับก่อนหน้า (C_{t-1}) โดยจะนำข้อมูลข้าเข้า (X_t) และ Hidden State ลำดับก่อนหน้า (H_{t-1}) และผ่านพิงก์ชันกระตุ้นซิกมอยด์ (Sigmoid Activation Function) ดัง(1) นั่นก็คือในกรณีที่ค่าของประตูการลืมเป็น 0 คือการลบข้อมูลใน Cell State ก่อนหน้า ทั้งหมด และ ค่าเป็น 1 คือการจดจำข้อมูลใน Cell State ก่อนหน้า ทั้งหมด [14, 13]

$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + b_f) \quad (1)$$

โดยที่

σ คือ พิงก์ชันกระตุ้นซิกมอยด์

W_{xf}, W_{hf} คือ เมตริกถ่วงน้ำหนัก (Weight Matrix) ของข้อมูลข้าเข้า และ Hidden State ตามลำดับ

x_t คือ ข้อมูลข้าเข้าจากภายนอก ณ เวลา t

h_{t-1} คือ ข้อมูล Hidden State ของลำดับก่อนหน้า หรือ ณ เวลา $t - 1$

b_f คือ ค่าคงที่ของฟังก์ชัน f (Bias)

2. ประตูขาเข้า (Input Gate) และ หน่วยความจำผู้สมัคร (Candidate Memory) ทำหน้าที่เป็นเวกเตอร์ตัวเลือกในการเพิ่มข้อมูลใน Cell State ลำดับก่อนหน้า (C_{t-1}) โดยจะนำข้อมูลขาเข้า (X_t) และ Hidden State (H_{t-1}) ไปผ่าน Input Gate ที่เป็นฟังก์ชันกราฟตุนซิกมอยด์ ดังสมการ (2) และ Candidate Memory ที่เป็นฟังก์ชันกราฟตุนไฮเพอร์โบลิกแทนเจนต์ (Hyperbolic Tangent Activation Function - tanh) เพื่อทำให้ข้อมูลให้เป็นค่ามาตรฐาน (Normalize) หรือ กีศีอยู่ในช่วง -1 ถึง 1 ดังสมการ (3) และนำมารวบผลลัพธ์จาก Input Gate และ Candidate Memory และเพิ่มเข้าไปใน Cell State ลำดับก่อนหน้า และส่งไปยัง LSTM ตัวถัดไป ดังสมการ (4) [14, 13]

$$i_t = \sigma(W_{x^i}x_t + W_{h^i}h_{t-1} + b_i) \quad (2)$$

$$g_t = \tanh(W_{x^c}x_t + W_{h^c}h_{t-1} + b_c) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (4)$$

โดยที่

σ คือ ฟังก์ชันกราฟตุนซิกมอยด์

\tanh คือ ฟังก์ชันกราฟตุนไฮเพอร์โบลิกแทนเจนต์

W_{x^i}, W_{x^c} คือ เมตริกถ่วงน้ำหนัก (Weight Matrix) ของข้อมูลขาเข้า

W_{h^i}, W_{h^c} คือ เมตริกถ่วงน้ำหนัก (Weight Matrix) ของ Hidden State

x_t คือ ข้อมูลขาเข้าจากภายนอก ณ เวลา t

h_{t-1} คือ ข้อมูล Hidden State ของลำดับก่อนหน้า หรือ ณ เวลา $t - 1$

b_i, b_c คือ ค่าคงที่ของฟังก์ชัน i และ c ตามลำดับ (Bias)

c_t คือ Cell State ของลำดับปัจจุบัน หรือ ณ เวลา t

3. ประตูขาออก (Output Gate) ทำหน้าที่กำหนดค่า Hidden State ข้าออก ในปัจจุบัน หรือ ณ เวลา t โดยจะนำข้อมูลขาเข้า (X_t) และ Hidden State (H_{t-1}) ไปผ่าน Output Gate ที่เป็นฟังก์ชันกราฟตุนซิกมอยด์ ดังสมการ (5) ไปร่วมกับ State Cell ณ เวลา t ที่ผ่านฟังก์ชันกราฟตุนไฮเพอร์โบลิกแทนเจนต์ ดังสมการ (6) [14, 13]

$$o_t = \sigma(W_{x^o}x_t + W_{h^o}h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

โดยที่

σ คือ ฟังก์ชันกราฟตุนซิกมอยด์

\tanh คือ ฟังก์ชันกราฟตุนไฮเพอร์โบลิกแทนเจนต์

W_{x^o}, W_{x^o} คือ เมตริกถ่วงน้ำหนัก (Weight Matrix) ของข้อมูลขาเข้า

x_t คือ ข้อมูลขาเข้าจากภายนอก ณ เวลา t

h_t คือ ข้อมูล Hidden State ของลำดับปัจจุบัน หรือ ณ เวลา t

h_{t-1} คือ ข้อมูล Hidden State ของลำดับก่อนหน้า หรือ ณ เวลา $t - 1$

b_o, b_c คือ ค่าคงที่ของพัฟ์ชั่น o (Bias) c_t คือ Cell State ของลำดับปัจจุบัน หรือ ณ เวลา t

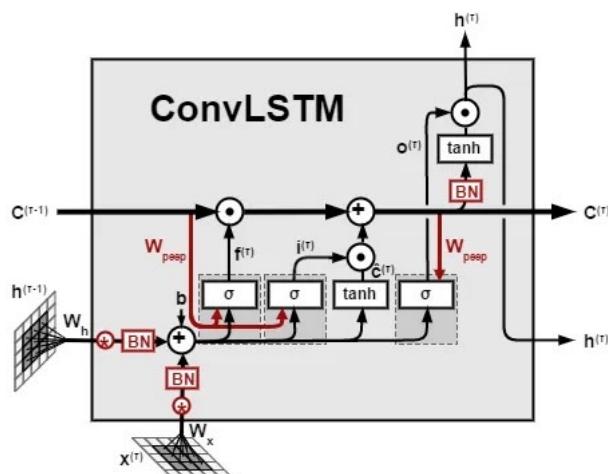
2.3 เครื่อข่ายคอนโวลูชันแบบความจำระยะสั้น-ยาว (Convolutional Long Short Term Memory - ConvLSTM)

เครื่อข่าย ConvLSTM คือ โมเดลที่คล้าย LSTM แต่แทนที่การ การคูณแบบจุด (dot product) ระหว่างค่าถ่วงน้ำหนัก กับ ข้อมูลขาเข้าและ ข้อมูล Hidden State ด้วยการคอนโวลูชันกับ Kernel ของ Conv2D ดังเช่นสมการต่อไปนี้ และมีโครงสร้างโครงข่ายดังรูปที่ 3

$$i_t = \sigma(W_{x^i} * x_t + W_{h^i} * h_{t-1} + b_i) \quad (7)$$

$$g_t = \tanh(W_{x^c} * x_t + W_{h^c} * h_{t-1} + b_c) \quad (8)$$

$$o_t = \sigma(W_{x^o} * x_t + W_{h^o} * h_{t-1} + b_o) \quad (9)$$



รูปที่ 3: โครงสร้างสถาปัตยกรรม ConvLSTM [13]

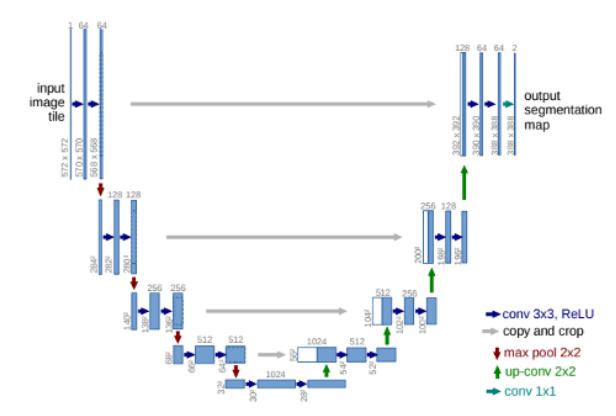
3 การประยุกต์การเรียนรู้โมเดลสำหรับสร้างภาพถ่ายท้องฟ้า (Sky-Image Generation Learning)

ในบทนี้จะกล่าวถึงโมเดลที่ถูกนำมาใช้ในการสร้างภาพถ่ายท้องฟ้า และการเรียนรู้ที่เกิดขึ้นภายในแต่ละโมเดลนั้นๆ

- สถาปัตยกรรมเครือข่ายแบบสกายเน็ต (SkyUNet)
- โมเดลไอนามิกทางภาพ (PhyDNet)
- การเรียนรู้แบบคู่ต่อสู้ช่วยสร้าง (Generative Adversarial Learning)

3.1 สถาปัตยกรรมเครือข่ายแบบสกายเน็ต (SkyUNet Network)

โมเดล SkyUNet [2] เป็นสถาปัตยกรรมเครือข่ายแบบบยูเน็ต เป็นหนึ่งในเครือข่ายที่นิยมใช้ในการทำงานด้าน วิสัยทัศน์ คอมพิวเตอร์ (Computer Vision) และ การแบ่งส่วนภาพ (Image Segmentation) โดยโครงสร้างของสถาปัตยกรรมโครงข่ายแบบบยูเน็ตเป็นดัง รูปที่ 4 [15]

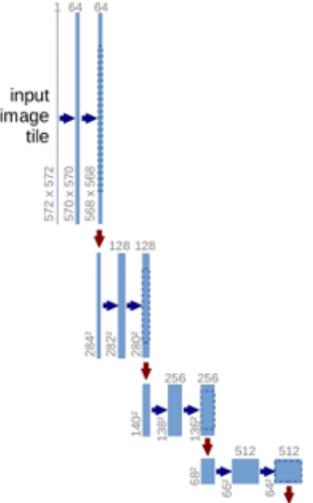


รูปที่ 4: โครงสร้างของสถาปัตยกรรมเครือข่ายแบบบยูเน็ต [15]

เครือข่ายบยูเน็ต(U-Net) เป็นเครือข่ายแบบตัวเข้ารหัส (Encoder) และตัวถอดรหัส (Decoder) ที่มีรูปร่างเป็นตัวบยู (U) ซึ่งประกอบด้วย 4 บล็อก (Block) ของตัวเข้ารหัสและตัวถอดรหัส ที่เชื่อมกันด้วยคอขวด (Bottle Neck) โดยตัวเข้ารหัส และตัวถอดรหัสในแต่ละชั้นจะมีการเชื่อมต่อ กันด้วยตัวเชื่อมต่อ (Skip Connection) [15, 16]

3.1.1 ตัวเข้ารหัส (Encoder)

ตัวเข้ารหัสทำหน้าที่เป็นตัวแยกคุณลักษณะ (Feature Extractor) ของภาพและเรียนรู้คุณลักษณะ (Features) ต่างๆ ในภาพที่ถูกป้อนเข้ามา โดยแต่ละบล็อกของตัวถอดรหัสนั้น ประกอบด้วย ชั้นของ 3x3 คอนโวลูชัน (Convolution) และ ชั้นของฟังก์ชันกระตุ้นเรียร์ (ReLU activation function) 2 ชุด และต่อ กับ ชั้นของ 2x2 max-pooling ซึ่งทำการลดขนาด (Downsample) ของมิติเชิงพื้นที่ (Spatial Dimensions) ลงครึ่งหนึ่ง และ เพิ่มจำนวนของชีวนั้น (Channel) เป็นสองเท่า ไปจนถึง คอขวด (Bottle Neck) [15, 16] ดังรูปที่ 5



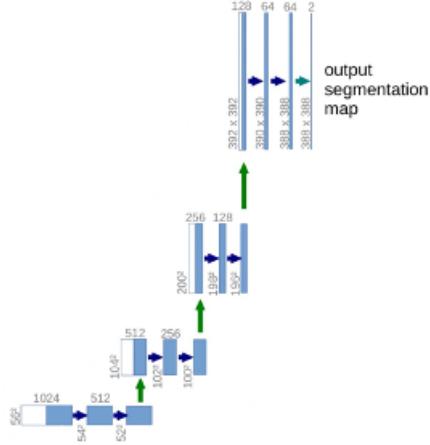
รูปที่ 5: ตัวเข้ารหัสในสถาปัตยกรรมโครงข่ายแบบบยเน็ต [15]

3.1.2 ตัวเชื่อมต่อ (Skip Connection)

ตัวเชื่อมต่อทำหน้าที่มอบข้อมูลเพิ่มเติมคุณลักษณะให้กับตัวถอดรหัส โดยจะเชื่อมต่อระหว่างตัวถอดรหัสและตัวเข้ารหัสในแต่ละขั้น เพื่อช่วยให้ตัวถอดรหัสสร้างคุณลักษณะของภาพได้ถูกต้องมากขึ้น และยังทำหน้าที่เป็นทางผ่านเพื่อช่วยให้เพิ่มการไหลของความซั่น (Gradient Flow) ในขณะการขยายพื้นที่กลับในเครือข่าย (Backpropagation) ให้เครือข่ายมีการเรียนรู้ที่ดีขึ้น [15, 16]

3.1.3 ตัวถอดรหัส (Decoder)

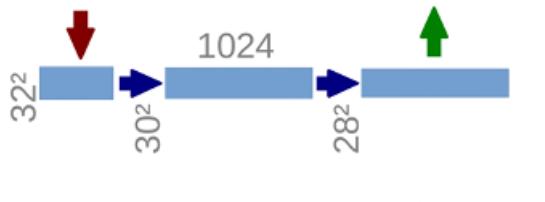
ตัวถอดรหัสทำหน้าที่สร้างคุณลักษณะของภาพให้ได้ภาพเดิม ซึ่งเชื่อมต่อกับตัวเชื่อม โดยในแต่ละบล็อกของตัวถอดรหัส จะประกอบด้วย ชั้นของ 3×3 convolution โคนไว้ชั้น และ ชั้นของฟังก์ชันกระตุนเรียรู 2 ชุด และต่อ กับ ชั้น 2×2 convolution โคนไว้ชั้นแบบสแลบ transpose (Transpose Convolution) ซึ่งทำการเพิ่มขนาด (Upsample) ของมิติเชิงพื้นที่ที่เป็นสองเท่า และลดจำนวนชั้นแนลงครึ่งหนึ่ง ไปจนถึง ตัวถอดรหัสสุดท้ายและส่งผลลัพธ์ต่อไปยัง ชั้นของ 1×1 convolution โคนไว้ชั้น และ ชั้นฟังก์ชันกระตุน sigmoid (Sigmoid Activation Function) เพื่อให้ได้ภาพที่สมบูรณ์ [15, 16] ดังรูปที่ 6



รูปที่ 6: ตัวถอดรหัสในสถาปัตยกรรมเครือข่ายแบบบุญเน็ต [15]

3.1.4 คอขวด (Bottle Neck)

คอขวดทำหน้าที่เป็นตัวเข้มและส่งข้อมูลระหว่างตัวเข้ารหัสและตัวถอดรหัสเพื่อให้เครือข่ายทั้งหมดสมบูรณ์ โดยประกอบด้วย ชั้นของ 3×3 convolutional และ ชั้นของฟังก์ชันกระตุ้นเรียร์ 2 ชุด [15, 16] ดัง รูปที่ 7



รูปที่ 7: โครงสร้างของสถาปัตยกรรมเครือข่ายแบบบุญเน็ต [15]

3.1.5 การเรียนรู้แบบ Supervised Learning

โมเดล SkyUNet เรียนรู้การสร้างรูปภาพถ่ายท้องฟ้า โดยวิธีการ Supervised learning กล่าวคือต้องให้รูปภาพถ่ายท้องฟ้าในอนาคตและมีการสร้างแบบจำลองการไฟลของแสงแบบเทียมขึ้นมาโดยใช้การเรียนรู้ด้วยฟังก์ชันต่างๆ ดังนี้

1. **การสูญเสียของความเข้ม (Intensity Loss)** เป็นการคำนวณค่าระยะห่าง L^2 -norm ของแต่ละพิกเซล (Pixel) ในมิติของสี (RGB Space) ระหว่างภาพที่นาย \hat{I} และ ภาพที่ถูกต้อง I ให้น้อยที่สุด เพื่อให้ค่าในแต่ละพิกเซลของภาพที่ท่านนายและภาพที่ถูกต้องมีความใกล้เคียงกันมากที่สุด ดังสมการ (10)

$$L_{int}(\hat{I}, I) = \|\hat{I} - I\|^2 \quad (10)$$

โดยที่

L_{int} คือ Intensity loss

\hat{I} คือ ภาพที่นาย

I คือ ภาพที่ต้อง

2. การสูญเสียความชัน (Gradient loss) เป็นการคำนวณและเทียบค่าของภาพทำนาย \hat{I} และภาพที่ต้อง I ในแต่ละ ดัชนีเชิงพื้นที่ (Spatial Indices) เพื่อช่วยให้ภาพที่ทำนายมีความคมชัดมากขึ้น ดังสมการ (11)

$$L_{gd}(\hat{I}, I) = \sum \| |\hat{I}_{i,j} - \hat{I}_{i-1,j}| - |I_{i,j} - I_{i-1,j}| \|_1 + \| |\hat{I}_{i,j} - \hat{I}_{i,j-1}| - |I_{i,j-1} - I_{i,j-1}| \|_1 \quad (11)$$

โดยที่

L_{gd} คือ Gradient Loss

\hat{I} คือ ภาพทำนาย

I คือ ภาพที่ต้อง

i และ j คือ ค่าดัชนีเชิงพื้นที่ของภาพ

3. การสูญเสียของการไหลของแสง (Optical Flow Loss) เป็นการคำนวณความแตกต่างจากการประมาณการไหลของแสง (Optical Flow Estimate) เพื่อให้ได้โมเดลสามารถทำนายภาพที่มีการเคลื่อนไหวที่แม่นยำและตรงกับความจริงมากที่สุด โดยในโครงงานนี้ได้มีการนำ เครือข่ายการไหลของแสง (Optical Flow Network) ซึ่งมีเป็นแบบเครือข่ายประสาทแบบconvolutional (Convolution Neural Network) ที่ถูกฝึกมาแล้ว [17] ใช้ในการคำนวณ ค่าประมาณการไหลของแสง ระหว่างภาพทำนาย \hat{I} และ ภาพที่ต้อง I ณ เวลา t และ $t+1$ ดังสมการ (12)

$$L_{op}(\hat{I}, I) = \| f(\hat{I}_{t+1}, I_t) - f(I_{t+1}, I_t) \|_1 \quad (12)$$

โดยที่

ฟังก์ชัน f เป็นฟังก์ชันประมาณการไหลของแสง (Optical Flow Estimate) จาก เครือข่ายการไหลของแสง (Optical Flow Network) ซึ่งเป็นการคำนวณค่าการลงโทษการเคลื่อนไหว (Motion Penalty) ของภาพ

L_{op} คือ Optical Flow Loss

\hat{I} คือ ภาพทำนาย

I คือ ภาพที่ต้อง

3.2 โมเดลไนมิกทางกายภาพ (PhyDNet)

PhyDNet คือ โมเดลการเรียนรู้เชิงลึก (Deep Learning Model) ที่ออกแบบมาเพื่อใช้ในงานที่เกี่ยวกับการทำนายวิดีโอแบบไม่มีผู้สอน (Unsupervised Video Prediction) โดยใช้หลักการจำลองความรู้ในนามิกทางกายภาพ (Physical Dynamic) ในการทำนายการเคลื่อนไหวของวัตถุภายในภาพดังสมการ (13) โดยแบ่งการอธิบายเป็น 2 ส่วน คือ Disentangling Architecture และ Physical-Constrained Recurrent Cell (PhyCell) [8]

$$\frac{\partial \mathbf{h}(t, \mathbf{x})}{\partial t} = \frac{\partial \mathbf{h}^p}{\partial t} + \frac{\partial \mathbf{h}^r}{\partial t} := \mathcal{M}_p(\mathbf{h}^p, \mathbf{u}) + \mathcal{M}_r(\mathbf{h}^r, \mathbf{u}) \quad (13)$$

กำหนดให้ ค่าของไนมิกทางกายภาพ (Physical Dynamic) และ องค์ประกอบเศษเหลือ (Residual Factors) ในพื้นที่แอบแฝง (Latent Space) \mathcal{H} มีการแยกกันแบบเชิงเส้น (Linearly Distentangled) โดยที่

$\mathbf{u} = \mathbf{u}(t, \mathbf{x})$ คือ ภาพ (Frame) ในลำดับวิดีโอ ณ เวลา t

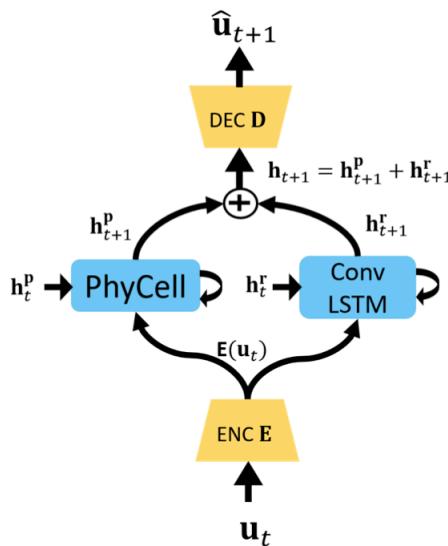
$\mathbf{x} = (x, y)$ คือ พิกัดเชิงพื้นที่ (Spatial Coordinate)

$\mathbf{h}(t, \mathbf{x}) \in \mathcal{H}$ คือ ค่าตัวแทนแอบแฝง (Latent Representation) ของวิดีโອอนถึง ณ เวลา t ซึ่งประกอบด้วย \mathbf{h}^p และ \mathbf{h}^r แสดงองค์ประกอบทางกายภาพ (Physical Component) และ Residual Component ตามลำดับ $\mathbf{h} = \mathbf{h}^p + \mathbf{h}^r$

$\mathcal{M}_p(\mathbf{h}^p, \mathbf{u})$ และ $\mathcal{M}_r(\mathbf{h}^r, \mathbf{u})$ คือ ค่าไดนามิกทางกายภาพ และ เชษヘルิโอ (Residual) ในพื้นที่แอบแฝง (Latent Space) ตามลำดับ

3.2.1 PhyDNet Disentangling Architecture

สถาปัตยกรรม PhyDNet มีจุดประสงค์ในการเรียนรู้การทำແຜນ (Mapping) จากข้อมูลลำดับข้ามเวลา (Input Sequence) ไปสู่ปริภูมิແຜນ (Latent Space) ของการประมาณองค์ประกอบทางกายภาพ (Physical Dynamic Component) และ องค์ประกอบ Residual ดังสมการ (13) โดยมีโครงสร้างสถาปัตยกรรมดัง รูปที่ 8 [8]



รูปที่ 8: โครงสร้างสถาปัตยกรรม PhyDNet) [8]

ตัวเข้ารหัส (Encoder) จะทำหน้าที่ทำແຜນ (Mapped) จากภาพ \mathbf{u}_t ณ เวลา t ไปสู่ปริภูมิตัวແຜນ \mathcal{H} จากนั้นผลลัพธ์ จากการตัวเข้ารหัส $E(\mathbf{u}_t)$ ส่งต่อไปยังเครือข่ายประสาทเวียนกลับ (Recurrent Neural Network) 2 เครือข่าย คือ

1. เครือข่าย PhyCell โดยทำหน้าที่ประมาณค่าองค์ประกอบทางกายภาพของสมการ (13) หรือก็คือ $\frac{\partial \mathbf{h}^p(t, \mathbf{x})}{\partial t} = \mathcal{M}_p(\mathbf{h}^p, \mathbf{u})$ และได้ผลลัพธ์เป็นค่าองค์ประกอบทางกายภาพ \mathbf{h}_{t+1}^p ณ เวลา $t+1$ หรือเวลาถัดไป
2. เครือข่าย ConvLSTM โดยทำหน้าที่ประมาณค่าองค์ประกอบ Residual ของสมการ (13) หรือก็คือ $\frac{\partial \mathbf{h}^r(t, \mathbf{x})}{\partial t} = \mathcal{M}_r(\mathbf{h}^r, \mathbf{u})$ และได้ผลลัพธ์เป็นค่าองค์ประกอบ Residual \mathbf{h}_{t+1}^r ณ เวลา $t+1$ หรือเวลาถัดไป

หลักจากนั้นจะทำการรวมองค์ประกอบทางกายภาพ \mathbf{h}_{t+1}^p และ Residual \mathbf{h}_{t+1}^r เข้าด้วยกันเป็น $\mathbf{h}_{t+1} = \mathbf{h}_{t+1}^p + \mathbf{h}_{t+1}^r$ และส่งไปยังตัวถอดรหัส (Decoder) เพื่อทำการแปลงข้อมูลให้เป็นภาพ \mathbf{u}_{t+1} ณ เวลา $t+1$ หรือเวลาถัดไป โดยที่ในโครงงานนี้ได้มีการฟังก์ชันสูญเสียจากเครือข่าย PhyCell มาให้ในการประมาณการเคลื่อนไหวของเมฆ และจะขออธิบายในบทถัดไป

3.2.2 เครือข่าย PhyCell

เครือข่าย PhyCell มีจุดประสงค์ในการประมาณ ค่าไดนามิกทางกายภาพ $\mathcal{M}_p(\mathbf{h}^p, \mathbf{u})$ โดยมีสมการดังสมการ (14) และ มีโครงสร้างเครือข่ายดังรูปที่ 9 [8]

$$\mathcal{M}_p(\mathbf{h}^p, \mathbf{u}) = \Phi(\mathbf{h}) + \mathcal{C}(\mathbf{h}, \mathbf{u}) \quad (14)$$

โดยที่

$\Phi(\mathbf{h})$ คือ ตัวทำนายทางกายภาพจาก ข้อมูลแห่ง \mathbf{h}

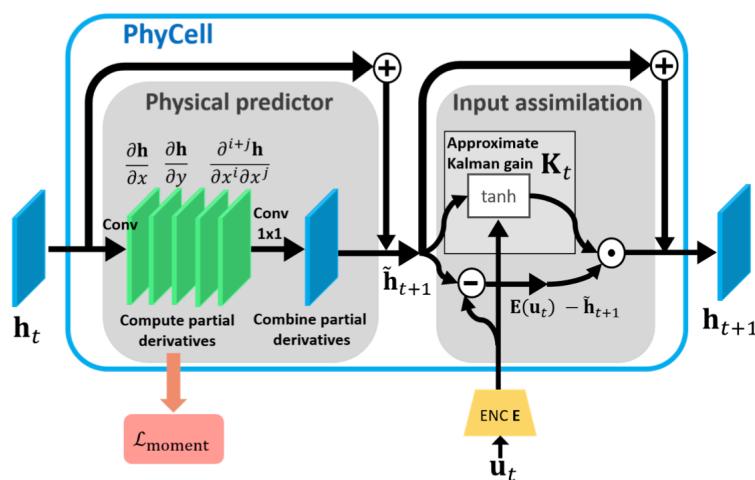
$\mathcal{C}(\mathbf{h}, \mathbf{u})$ คือ พจน์ความถูกต้องในการโมเดลลักษณะทางกายภาพจาก ข้อมูลแห่ง \mathbf{h} และ ข้อมูลข้าเข้า

ตัวทำนายทางกายภาพ (Physical Predictor) $\Phi(\mathbf{h})$ สามารถอธิบายได้ด้วยสมการ (15) โดยที่ $\Phi(\mathbf{h})$ เป็นการรวม ค่าอนุพันธ์เชิงพื้นที่ (Spatital Derivatives) และค่าสัมประสิทธิ์ $c_{i,j}$ จนถึงอนุพันธ์ลำดับ (Differential order) ที่ q

$$\Phi(\mathbf{h}(t, \mathbf{x})) = \sum_{i,j:i+j \geq q} c_{i,j} \frac{\partial^{i,j} \mathbf{h}}{\partial x^i \partial y^j}(t, \mathbf{x}) \quad (15)$$

ความถูกต้อง (Correction) $\mathcal{C}(\mathbf{h}, \mathbf{u})$ สามารถอธิบายได้ด้วยสมการ (16) ซึ่งเป็นการคำนวณความต่างของสถานะแห่ง (Latent State) หลังจากมีการเคลื่อนไหวทางกายภาพ (Physical Motion) $\mathbf{h}(t, \mathbf{x}) + \Phi(\mathbf{h}(t, \mathbf{x}))$ และค่าผลการเข้ารหัส ของข้อมูลข้าเข้าใหม่ $\mathbf{E}(\mathbf{u}(t, \mathbf{x}))$ โดยที่ $\mathbf{K}(t, \mathbf{x})$ คือค่าตัวประกอบ gating และ \odot คือ ผลคูณ Hadamard

$$\mathcal{C}(\mathbf{h}, \mathbf{u}) := \mathbf{K}(t, \mathbf{x}) \odot [\mathbf{E}(\mathbf{u}(t, \mathbf{x})) - \mathbf{h}(t, \mathbf{x}) + \Phi(\mathbf{h}(t, \mathbf{x}))] \quad (16)$$



รูปที่ 9: โครงสร้างเครือข่าย PhyCell [8]

3.2.3 การเรียนรู้สมการทางกายภาพของ PhyCell ด้วย Moment Loss

ค่าสูญเสียที่เกิดจากการประมาณค่าอนุพันธ์ $\frac{\partial^{i,j}}{\partial x^i u^j}$ ในเครือข่าย PhyCell โดยที่ $\{w_{p,i,j}^k\}_{i,j \geq k}$ มีขนาด $k \times k$ จะถูกควบคุมด้วย Moment Loss ดังสมการ (17)

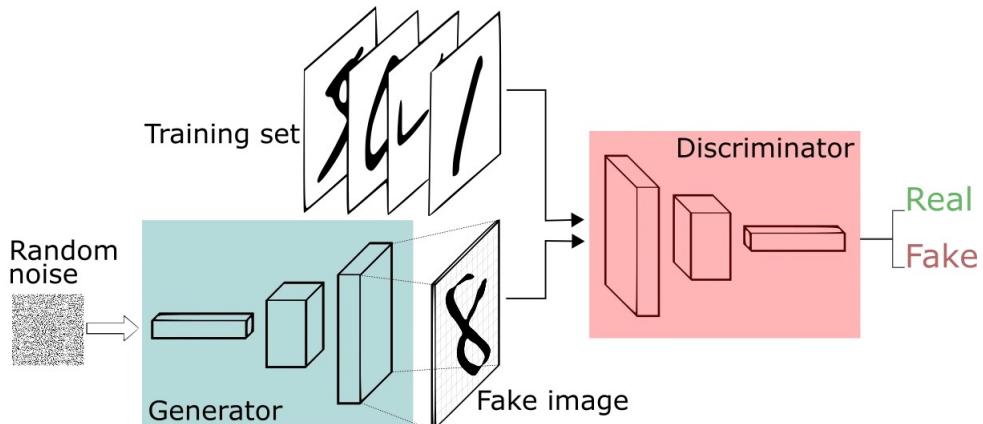
$$\mathcal{L}_{moment} = \sum_{i \leq k} \sum_{j \leq k} \| \mathbf{M}(w_{p,i,j}^k) - \Delta_{i,j}^k \|_F \quad (17)$$

การประมาณค่าประมาณกระบวนการอนุพันธ์ $\frac{\partial^{a,b}}{\partial x^a y^b}$ ด้วยเมตริก Moment $\mathbf{M}(w)_{i,j} = 0$ สำหรับ $i \neq a$ และ $j \neq b$ อธิบายด้วยสมการ (18) [8]

$$\mathbf{M}(w)_{i,j} = \frac{1}{i!j!} \sum_{u=-\frac{k-1}{2}}^{\frac{k-1}{2}} \sum_{v=-\frac{k-1}{2}}^{\frac{k-1}{2}} u^i v^j \mathbf{k}[u, v] \text{ for } i, j = 0, \dots, k-1 \quad (18)$$

3.2.4 การเรียนรู้แบบคู่ต่อสู้ช่วยสร้าง (Generative Adversarial Learning)

การเรียนรู้ของ PhyDNet ทำได้โดยการใช้การเรียนรู้แบบเครือข่ายคู่ต่อสู้ช่วยสร้าง (GANs) เป็นเครือข่ายมีจุดประสงค์ในการสร้างข้อมูลจากสัญญาณรบกวน (Noise) เพื่อให้ได้ข้อมูลที่มีลักษณะเหมือนกับข้อมูลที่ถูกต้อง (Ground Truth) ดังรูปที่ 10 โดย GANs ประกอบด้วยเครือข่ายหลัก 2 เครือข่าย คือ ตัวสร้างแบบ (Generator) ซึ่งในที่นี้คือ PhyDNet และตัวถอดแบบ (Discriminator) โดยในที่นี้ถูกสร้างด้วย CNN layers โดยที่ทั้งสองเครือข่ายนั้นจะทำการแข่งขันและกันนั้นทำให้เครือข่ายนี้สามารถสร้างข้อมูลที่มีความเหมือนข้อมูลต้นฉบับและมีความแม่นยำ



รูปที่ 10: โครงสร้างเครือข่ายคู่ต่อสู้ช่วยสร้าง (GANs) [18]

ตัวสร้างแบบ (Generator) ตัวสร้างแบบมีหน้าที่ในรับข้อมูลเข้ามาแบบสุ่ม (Random Input) หรือ สัญญาณรบกวน (Noise Input) เพื่อสร้างข้อมูลปลอม (Fake Data) ที่มีลักษณะเหมือนข้อมูลที่ใช้ในการฝึกฝน (Training Data) มากที่สุดเพื่อให้ตัวถอดแบบ (Discriminator) ไม่สามารถแยกออกว่าเป็นข้อมูลจริง (Real Data) โดยมีพังก์ชันจุดประสงค์ดังสมการ (19) [19]

$$\min_G V(G) = \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(Z)))] \quad (19)$$

โดยที่

$D(G(Z))$ คือ ความน่าจะเป็นของข้อมูลที่ถูกสร้างโดยตัวสร้างแบบว่าเป็นข้อมูลจริง หรือก็คือ $D(G(Z)) = 1$ คือ ตัว

ถอดแบบเดาว่าเป็นข้อมูลจริง

\mathbb{E}_z คือ ค่าความคาดหวัง (Expected Value) ของข้อมูลข้าเข้าแบบสุ่ม (Random Inputs)

ตัวถอดแบบ (Discriminator) คือตัวแยกชนิด (Classifier) ของเครือข่ายคู่ต่อสู้ที่สร้าง (GANs) ซึ่งจะทำหน้าเรียนรู้และแยกแยะว่าข้อมูลที่ได้รับมาจากตัวสร้างแบบว่าเป็นข้อมูลจริง หรือ ข้อมูลปลอม โดยมีพักรหัสจุดประสงค์ตั้งสมการ (20) [19]

$$\max_D V(D) = \max_D \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (20)$$

โดยที่

$D(x)$ คือ ความน่าจะเป็นของข้อมูลจริง x ว่าเป็นข้อมูลจริง

$D(G(Z))$ คือ ความน่าจะเป็นของข้อมูลที่ถูกสร้างโดยตัวสร้างแบบว่าเป็นข้อมูลจริง หรือก็คือ $D(G(Z)) = 1$ คือ ตัวถอดแบบเดาว่าเป็นข้อมูลจริง

$G(z)$ คือ ข้อมูลที่สร้างขึ้นโดยตัวสร้างแบบ จากการข้อมูลรบกวน z (Noise)

\mathbb{E}_x คือ ค่าความคาดหวัง (Expected Value) ของข้อมูลจริง (Real Data)

\mathbb{E}_z คือ ค่าความคาดหวัง (Expected Value) ของข้อมูลข้าเข้าแบบสุ่ม (Random Inputs)

4 ประยุกต์การเรียนรู้โมเดลการประมาณค่ารังสีแสงอาทิตย์โดยตรงจากอนุกรมของภาพถ่าย (Direct Irradiance Forecasting Approaches)

ในบทนี้จะกล่าวถึงประมาณค่าพลังงานแสงอาทิตย์โดยตรงจากอนุกรมของภาพถ่าย และการเรียนรู้ที่เกิดขึ้นภายในแต่ละโมเดลนั้นๆ ได้แก่

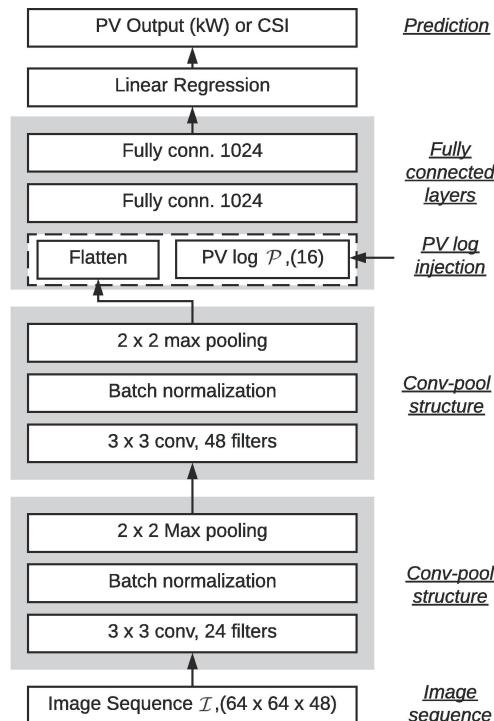
- สถาปัตยกรรมเครือข่ายแบบชั้นเช็ต (SUNSET) [10]
- สถาปัตยกรรมเครือข่ายแบบโซลาร์เน็ต (Solarnet) [11]
- สถาปัตยกรรมเครือข่ายแบบยูเน็ต (UNET) [9]

ทั้งนี้สถาปัตยกรรมเครือข่ายแบบ SUNSET ถูกสร้างมาเพื่อประมาณค่าพลังงานแสงอาทิตย์ (Solar Photovoltaic (PV) Power) โดยที่ Horizon ของ SUNSET ถูกสร้างมาสำหรับการพยากรณ์ระหว่างวัน (Intra-Day Forecasting) ในขณะที่สถาปัตยกรรมเครือข่ายแบบโซลาร์เน็ตถูกสร้างมาเพื่อพยากรณ์ค่ารังสีแสงอาทิตย์ (solar Irradiance) ภายใน Intra-Hour Horizon

ต่างจากทั้งสองโมเดลข้างต้น สถาปัตยกรรมเครือข่ายแบบยูเน็ต [9] นั้นถูกสร้างมาเป็น Regression Model ที่ให้ค่าประมาณค่าพลังงานแสงอาทิตย์ โดยการใช้ภาพถ่ายท้องฟ้า ณ เวลาหนึ่น

4.1 สถาปัตยกรรมเครือข่ายแบบชั้นเช็ต (SUNSET)

คอนโวลูชันเลเยอร์ (Convolutional Layer) เป็นองค์ประกอบหลักของสถาปัตยกรรมเครือข่ายแบบ SUNSET [10] โดยจะรับอินพุต (Input) ที่เป็นอนุกรมข้อมูลของเป็นภาพถ่าย (Image Series) จำนวน 48 ภาพในอดีต และค่า PV ในอดีต 16 ค่า ดังรูปที่ 11 โครงสร้างแบบนี้มีความพิเศษอยู่ที่ความสามารถในการสมอนุกรมข้อมูลจากภาพ และอนุกรมข้อมูลค่า PV ในอดีต



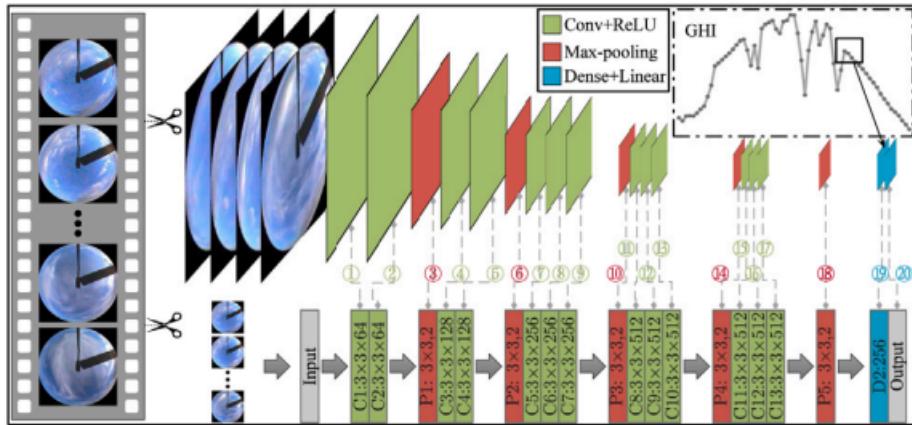
รูปที่ 11: โครงสร้างสถาปัตยกรรม SUNSET [10]

การเรียนรู้ของ SUNSET นั้นอยู่ในรูปแบบของการเรียนรู้แบบมีผู้สอน (Supervised Learning) นั่นคือหากว่า \hat{y} เป็นตัวแทนของค่า PV ที่โมเดลพยากรณ์ขึ้น (Predict) และ y เป็นตัวแทนของค่า PV ที่แท้จริง (Ground Truth) พารามิเตอร์ (Parameters) ของ SUNSET จะถูกเรียนรู้เพื่อลดความต่างระหว่าง \hat{y} และ y ที่วัดโดยค่าเฉลี่ยของผลต่างยกกำลังสอง (Mean Squared Error Loss)

4.2 สถาปัตยกรรมเครือข่ายแบบโซลาร์เน็ต (Solarnet)

สถาปัตยกรรมเครือข่ายแบบ Solarnet [11] นั้นเรียนแบบสถาปัตยกรรมของ VGG [20] โดยที่โมเดลจะรับภาพถ่ายท้องฟ้าในอดีตในรูปแบบของอนุกรมเพียงอย่างเดียว และพยากรณ์ค่ารังสีแสงอาทิตย์ ดังแสดงในรูป

การเรียนรู้ของ Solarnet นั้นอยู่ในรูปแบบของการเรียนรู้แบบมีผู้สอน (Supervised Learning) โดยวัดโดยค่าเฉลี่ยผลต่างสัมบูรณ์ (Mean Absolute Error Loss) ระหว่าง รังสีแสงอาทิตย์ (Irradiance) ที่โมเดลพยากรณ์ขึ้นและ รังสีแสงอาทิตย์ (Irradiance) ที่วัดได้จากสถานีภาคพื้น

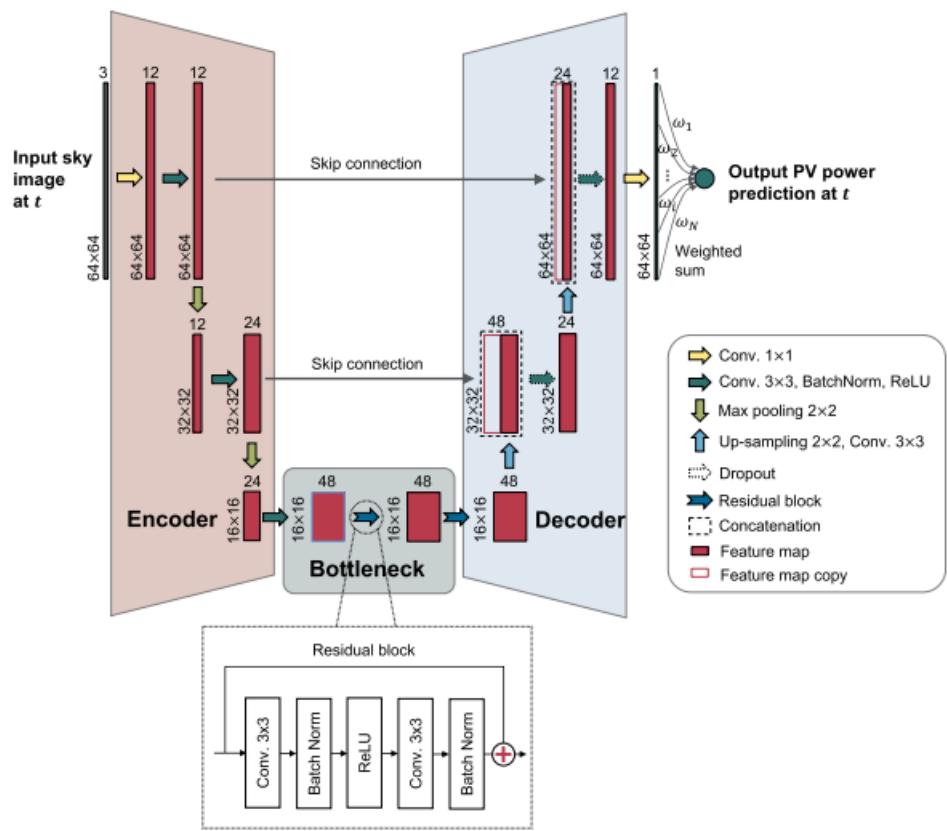


รูปที่ 12: โครงสร้างสถาปัตยกรรม Solarnet [11]

4.3 สถาปัตยกรรมเครือข่ายแบบยูเน็ต (UNET)

ใน [9] ได้ออกแบบสถาปัตยกรรมเครือข่ายคล้าย UNET [15, 16] เพื่อพยากรณ์ค่าพลังงานแสงอาทิตย์ ณ เวลาที่ต้องการ โดยการรับข้อมูลภาพถ่าย ณ เวลาหนึ่ง ซึ่งถือว่าเป็น Regression Model โดยเจตจำนงของการสร้างโมเดลนี้นั้นทำเพื่อเปลี่ยนผลจากการสร้างภาพในอนาคตมาสู่ค่าพยากรณ์พลังงานแสงอาทิตย์ ซึ่งอาจรับอินพุตมาจากผลของโมเดลสร้างภาพท้องฟ้าในอนาคต เช่น SkyNet-Unet, PhyDNet, และ SkyGPT โดยโครงสร้างของยูเน็ตเป็นดังรูป 13

การเรียนรู้ของ U-Net อยู่ในรูปแบบของการเรียนรู้แบบมีผู้สอน (Supervised Learning) โดยวัดโดยค่าเฉลี่ยของผลต่างยกกำลังสอง (Mean Squared Error Loss) ระหว่าง รังสีแสงอาทิตย์ (Irradiance) ที่โมเดลพยากรณ์ขึ้นและค่ารังสีแสงอาทิตย์ (Irradiance) ที่วัดได้จากสถานีภาคพื้น



รูปที่ 13: โครงสร้างสถาปัตยกรรม UNET [9]

5 ผลลัพธ์จากการดำเนินการ

ผลลัพธ์จากการดำเนินการค่าของรังสีของแสงอาทิตย์จากภาพถ่ายท้องฟ้า มีสองรูปแบบกล่าวคือ

1. การสร้างรูปแบบการการเคลื่อนไหวของเมฆ (Image Generation Approaches) และวิ่งภาพที่ถูกสร้างมาประมาณค่าพลังงานแสงอาทิตย์ ซึ่งได้ทำการเปรียบเทียบประสิทธิภาพของโมเดล ConvLSTM [7], SkyNet-Unet [2], และ PhyDNet [8]
2. การประมาณค่าพลังงานแสงอาทิตย์โดยตรงจากอนุกรมของภาพถ่าย (Direct Irradiance Forecasting Approaches) ซึ่งได้ทำการเปรียบเทียบประสิทธิภาพของโมเดล Sunset [10], Solarnet [11], และ Unet [9]

ทั้งนี้โครงงานนี้ได้ทำการทดสอบและพัฒนาโมเดลสำหรับทำนายภาพถ่ายท้องฟ้าโดยใช้ PyTorch ซึ่งเป็น ไลบรารี (Library) ในภาษาทางโปรแกรมมิ่ง (Programming) Python และใช้ข้อมูลภาพถ่ายท้องฟ้าของคณะวิศวกรรมศาสตร์สาขาไฟฟ้า บทที่ 5.1 จะกล่าวถึงการจัดเตรียมชุดข้อมูลที่ใช้ในการทดลองนี้ สำหรับการทดสอบ (Testing) และฝึกฝนโมเดล (Training) อยู่ในบทที่ 5.2 และผลการทดลองการสร้างรูปแบบการการเคลื่อนไหวของเมฆจะอยู่ในบทที่ 5.5.1 และ การประมาณค่าพลังงานแสงอาทิตย์โดยตรงจากอนุกรมของภาพถ่ายอยู่ในบทที่ 5.5.2

5.1 การเตรียมชุดข้อมูล (Data Preparation)

ชุดข้อมูลภาพถ่ายท้องฟ้าของคณะวิศวกรรมศาสตร์สาขาไฟฟ้า (CUEE Dataset) เป็นภาพถ่ายท้องฟ้าที่คณะวิศวกรรมศาสตร์สาขาไฟฟ้าโดยมีขนาดอยู่ที่ 1920×1080 พิกเซล (Pixel) และถูกบันทึกในช่วงเวลา 5.00 น จนถึง 18.59 น. ตั้งแต่วันที่ 15 มีนาคม 2566 จนถึง 22 สิงหาคม 2566 โดยแต่ละภาพถูกถ่ายในเวลาที่แตกต่างกันทุก 60 วินาที ดัง รูปที่ 14 และได้มีการจัดการชุดข้อมูลดังนี้

1. เลือกภาพที่ถูกถ่ายในช่วงเวลา 8.00 น. จนถึง 16.00 น. เนื่องจากเป็นช่วงเวลาที่เห็นแสงแดดชัดเจนมากที่สุด
2. เนื่องจากชุดข้อมูลภาพถ่ายท้องฟ้าของคณะวิศวกรรมศาสตร์สาขาไฟฟ้านั้นมีขนาดที่ใหญ่มากเราจึงทำการเปลี่ยนขนาด (Resize) ภาพจากขนาด 1920×1080 พิกเซล เป็น 64×64 พิกเซล เพื่อลดระยะเวลา และ หน่วยความจำที่ใช้งาน (Memory Usage) ในการฝึกฝน, ทดสอบโมเดล
3. แบ่งชุดข้อมูลเป็น 3 ชุดดังนี้
 - ชุดข้อมูลฝึกฝน (Training Data) ขนาด 17,752 ตัวอย่าง (Samples) หรือ 80%
 - ชุดข้อมูลตรวจสอบ (Validating Data) ขนาด 1,091 ตัวอย่าง หรือ 5%
 - ชุดข้อมูลทดสอบ (Testing Data) ขนาด 3,276 ตัวอย่าง หรือ 15%



รูปที่ 14: ตัวอย่างภาพถ่ายท้องฟ้าของคณะกรรมการมาตรฐานสาขาไฟฟ้า

5.1.1 การจัดชุดข้อมูล (Data Partition)

เพื่อให้ได้ผลลัพธ์ในการทำนายภาพถ่ายท้องฟ้าในระยะที่ยาวขึ้น จึงได้มีการเพิ่มภาพถ่ายท้องฟ้าของข้อมูลภาพข้าเข้า (Input Image) และ ภาพที่ถูกต้อง (Ground Truth Image) ในกลุ่มตัวอย่างที่ใช้ฝึกฝนโมเดลเพื่อให้ได้ผลลัพธ์ในการทำนายระยะยาว โดยกำหนดข้อมูลข้าเข้า (Input Data) เป็นภาพถ่ายท้องฟ้าที่เวลา 15 นาทีในอดีตจนถึงเวลาปัจจุบัน หรือ ทั้งหมด 16 ตัวหยุดเวลา ($I_{t-15}, I_{t-14}, \dots, I_{t-1}, I_t$) และ ข้อมูลข้าออก (Output) เป็นภาพถ่ายท้องฟ้าในอนาคต 15 นาที หรือ ทั้งหมด 15 ตัวหยุดเวลา ($\hat{I}_{t+1}, \hat{I}_{t+2}, \dots, \hat{I}_{t+14}, \hat{I}_{t+15}$) โดยที่ ข้อมูลข้าเข้าและข้อมูลข้าออกจะมีขนาด (Shape) เป็น ข้อมูล 5 มิติ (Batch Size x RGB Channel x Time stamp x Width x Height)

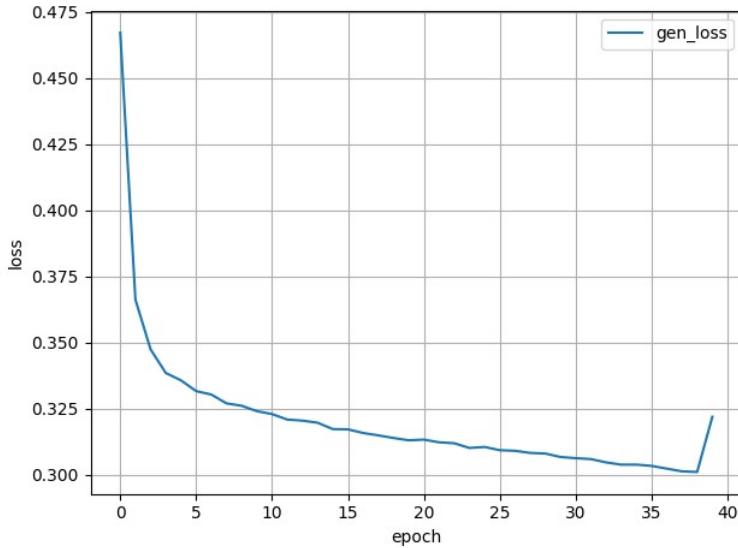
5.2 การฝึกฝนโมเดลทำนายภาพถ่ายท้องฟ้า (Sky Images Model Training)

ในโครงการนี้ได้มีการฝึกฝนโมเดลสำหรับการทำนายภาพถ่ายท้องฟ้าของ CUCEE ทั้งหมด 3 โมเดล ได้แก่ โมเดล ConvLSTM, PhyDNet+GAN และ SkyUNet โดยมีการรับข้อมูลข้าเข้าและข้อมูลข้าออกแบบระยะยาว (ตามในหัวข้อที่ 5.1.1) ในการฝึกฝนโมเดล

5.2.1 SkyUNet Model

SkyUNet เป็นโมเดลโครงข่ายสถาปัตยกรรมแบบยูเน็ต (U-Net Architecture) และใช้การประมาณการไอลของแสง (Optical Flow Estimate) จาก เครื่อข่ายการไอลของแสง (LiteFlowNet) ในการทำการเคลื่อนไหวของเมฆ โดยมี รายละเอียดการฝึกฝนโมเดลดังนี้

- จำนวนโมเดลพารามิเตอร์ (Model Parameters) = 7,731,757 พารามิเตอร์
- ขนาดชุดตัวอย่าง (Batch Size) = 4
- ค่าอัตราการเรียนรู้ (Learning Rate) = 0.0002
- จำนวนรอบการฝึกฝน (Training Epochs) = 40 รอบ
- เครื่องมือเพิ่มประสิทธิภาพแบบอัตโนมัติ (Adam Optimizer)
- พังค์ชันสูญเสีย (Loss Function) ประกอบด้วย Intensity Loss, Gradient Loss และ Optical Flow Loss โดยที่ ค่าถ่วงน้ำหนัก (λ) คือ 5, 0.00111 และ 0.010 ตามลำดับ

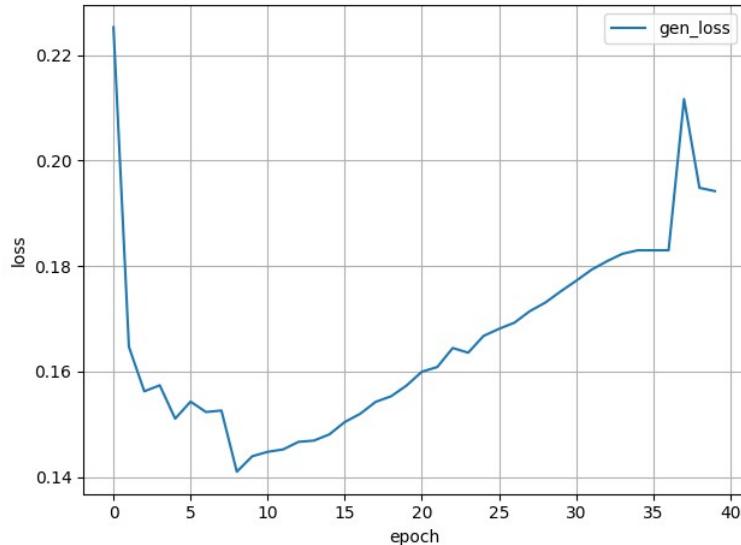


รูปที่ 15: กราฟแสดงค่าสูญเสียในการฝึกฝนโมเดล SkyUNet

5.2.2 ConvLSTM Model

ConvLSTM เป็นโมเดลเครือข่ายแบบconvโวลูชันแบบความจำรยะสั้น-ยาว (Convolutional Long Short Term Memory - ConvLSTM) โดยใช้หลักการเข้ารหัสและถอดรหัสภาพ ในการทำนายภาพถ่ายท้องฟ้าในอนาคต โดยมีรายละเอียดในการฝึกฝนโมเดลดังนี้

- จำนวนโมเดลพารามิเตอร์ (Model Parameters) = 2,227,139 พารามิเตอร์
- ขนาดชุดตัวอย่าง (Batch Size) = 4
- ค่าอัตราการเรียนรู้ (Learning Rate) = 0.001
- จำนวนรอบการฝึกฝน (Training Epochs) = 40 รอบ
- เครื่องมือเพิ่มประสิทธิภาพแบบอัตโนมัติ (Adam Optimizer)
- ฟังก์ชันสูญเสีย (Loss Function) ประกอบด้วย Encoder Loss และ Decoder Loss

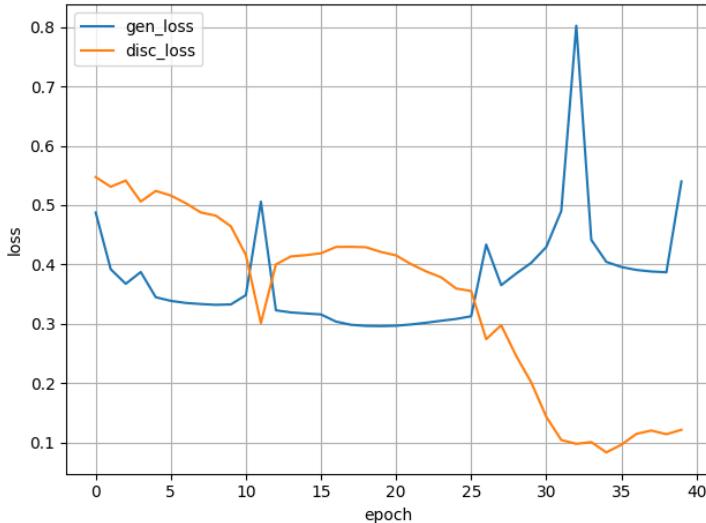


รูปที่ 16: กราฟแสดงค่าสูญเสียในการฝึกฝนโมเดล ConvLSTM

5.2.3 PhyDNet + GANs Model

PhyDNet + GANs เป็นโมเดลเครือข่ายคู่ต่อสู้ช่วยสร้าง (GANs) ที่มีการใช้หลักการจำลองความรู้ไดนามิกทางกายภาพ จาก โมเดลไดนามิกทางกายภาพ (PhyDNet) ในการทำนายการเคลื่อนไหวของเมฆ โดยมีรายละเอียดของโมเดลดังนี้

- จำนวนโมเดลพารามิเตอร์ (Model Parameters) = 3,119,222 พารามิเตอร์
- ขนาดชุดตัวอย่าง (Batch Size) = 4
- ค่าอัตราการเรียนรู้ (Learning Rate) = 0.002
- จำนวนรอบการฝึกฝน (Training Epochs) = 40 รอบ
- เครื่องมือเพิ่มประสิทธิภาพแบบบอดี้ม (Adam Optimizer)
- ฟังก์ชันสูญเสีย (Loss Function) ประกอบด้วย Moment Loss, Generator Loss และ Discriminator Loss



รูปที่ 17: กราฟแสดงค่าสูญเสียในการฝึกฝนโมเดล PhyDNet + GAN

5.2.4 สรุปผลลัพธ์ที่ได้จากการฝึกฝนโมเดลสำหรับทำนายภาพถ่ายห้องฟ้า

ในการฝึกฝนโมเดลสำหรับทำนายภาพถ่ายห้องฟ้าทั้งสิ้น 3 โมเดล ได้ผลลัพธ์ที่แตกต่างกันดังนี้ จากรูปที่ 15 จะเห็นว่าค่าของฟังก์ชันสูญเสียในการเรียนรู้ของโมเดล SkyUNet มีค่าลดลงอย่างต่อเนื่องในการฝึกฝนโมเดลจนถึงจุดฝึกฝนครั้งที่ 37 และ จากรูปที่ 16 จะเห็นว่าค่าของฟังก์ชันสูญเสียของโมเดล ConvLSTM มีค่าลดลงในการฝึกฝนโมเดลจนถึงจุดฝึกฝนครั้งที่ 8 และมีค่าสูญเสียที่เพิ่มขึ้นไปในการฝึกฝน

โมเดล PhyDNet + GAN โดยจาก รูปที่ 17 จะเห็นว่าค่าของฟังก์ชันสูญเสียของโมเดล PhyDNet + GAN ในขณะการฝึกฝนโมเดลนั้นมีการเปลี่ยนแปลงอย่างสม่ำเสมอไปจนถึงการฝึกฝนรอบที่ 30 เนื่องจากโมเดลตัวสร้าง (Generator) กำลังแข่งกันกับ โมเดลถอดแบบ (Discriminator) และจะเห็นได้ว่าโมเดลถอดแบบนั้นอาจชนะโมเดลสร้างแบบได้ในที่สุด

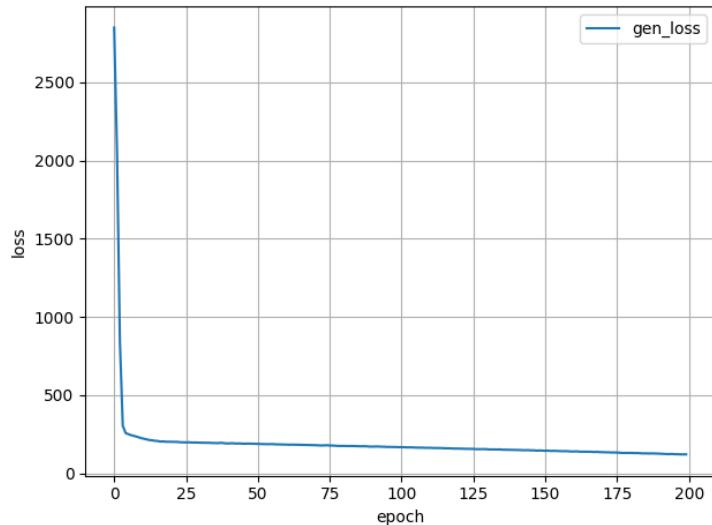
5.3 การฝึกฝนโมเดลทำนายค่ารังสีแสงอาทิตย์ (Irradiance Forecasting Model Training)

ในโครงการนี้ได้มีการฝึกฝนโมเดลสำหรับการทำนายค่ารังสีแสงอาทิตย์ ทั้งหมด 3 โมเดล ได้แก่ โมเดล SUNSET, Solarnet และ UNET โดยมีการรับข้อมูลข้าเข้าเป็นภาพถ่ายห้องฟ้าเพื่อทำนายค่ารังสีแสงอาทิตย์ และนำค่าที่ทำนายได้จากโมเดลไปเทียบกับค่ารังสีแสงแดดจากชุดข้อมูล EE Station 1 ซึ่งรายในชุดข้อมูลนี้จะมีค่ารังสีแสงแดดของคณะวิศวกรรมศาสตร์ไฟฟ้า เพื่อใช้เป็นข้อมูลรังสีแสงแดดที่ถูกต้อง (Ground Truth) ในการฝึกฝนและทดสอบโมเดล โดยมีโมเดลมีการเรียนรู้ตาม หัวข้อที่ 4 และมีรายละเอียดในการฝึกฝนโมเดลดังนี้

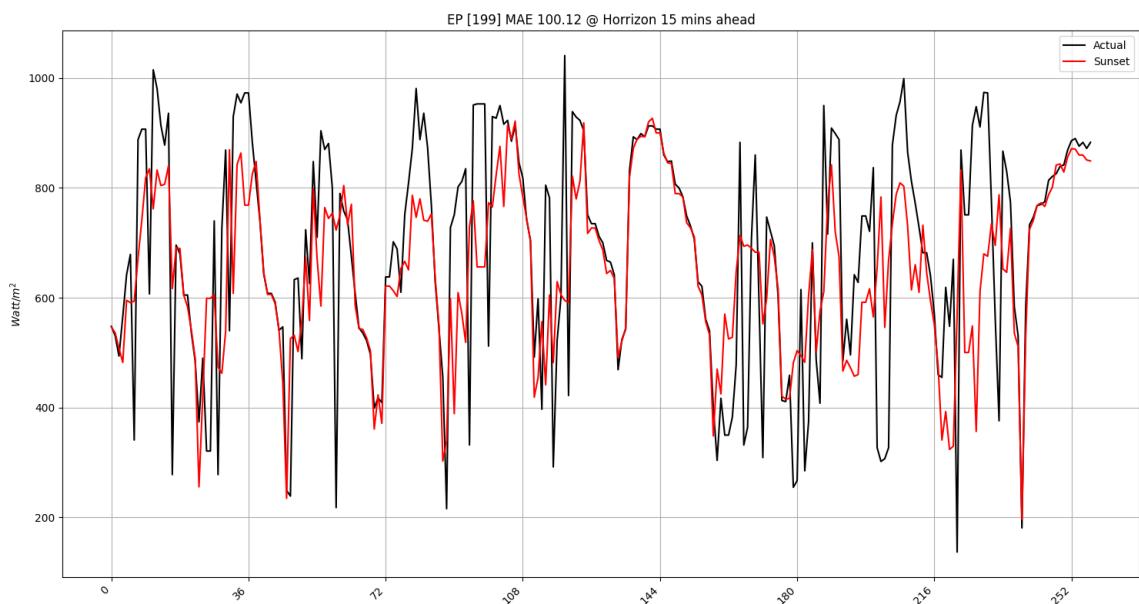
5.3.1 การฝึกฝนโมเดล SUNSET

- จำนวนโมเดลพารามิเตอร์ (Model Parameters) = 26,296,497 พารามิเตอร์
- ขนาดชุดตัวอย่าง (Batch Size) = 128
- ค่าอัตราการเรียนรู้ (Learning Rate) = 2×10^{-6}
- จำนวนรอบการฝึกฝน (Training Epochs) = 200 รอบ

- เครื่องมือเพิ่มประสิทธิภาพแบบดั้ม (Adam Optimizer)
- พังก์ชันสูญเสีย (Loss Function) คือ Mean Squared Error Loss



รูปที่ 18: กราฟแสดงค่าสูญเสียในการฝึกฝนโมเดล SUNSET

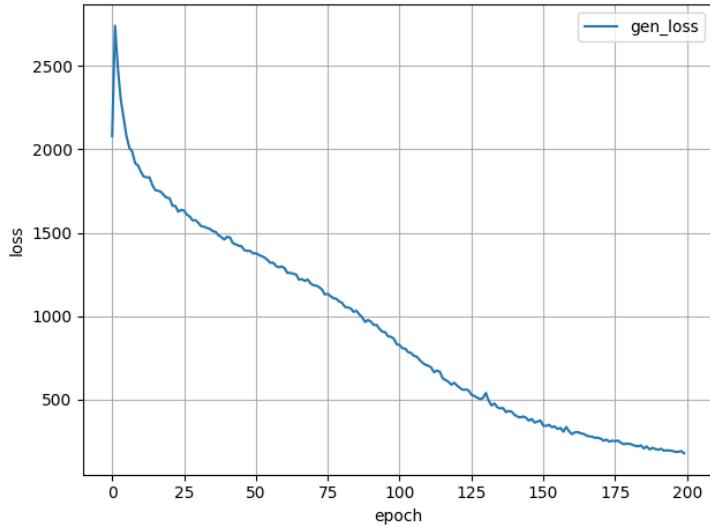


รูปที่ 19: ตัวอย่างผลลัพธ์ของการฝึกฝนโมเดล SUNSET

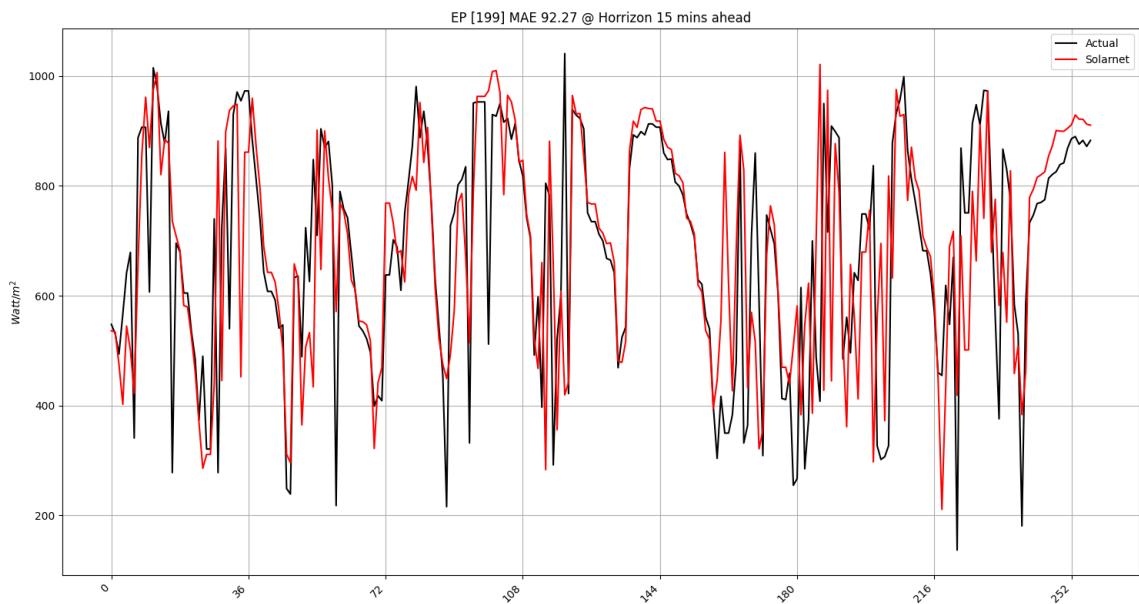
5.3.2 การฝึกฝนโมเดล Solarnet

- จำนวนโมเดลพารามิเตอร์ (Model Parameters) = 14,867,974 พารามิเตอร์
- ขนาดชุดตัวอย่าง (Batch Size) = 16
- ค่าอัตราการเรียนรู้ (Learning Rate) = 0.00001

- จำนวนรอบการฝึกฝน (Training Epochs) = 200 รอบ
- เครื่องมือเพิ่มประสิทธิภาพแบบอัตโนมัติ (Adam Optimizer)
- ฟังก์ชันสูญเสีย (Loss Function) คือ Mean Squared Error Loss



รูปที่ 20: กราฟแสดงค่าสูญเสียในการฝึกฝนโมเดล Solarnet

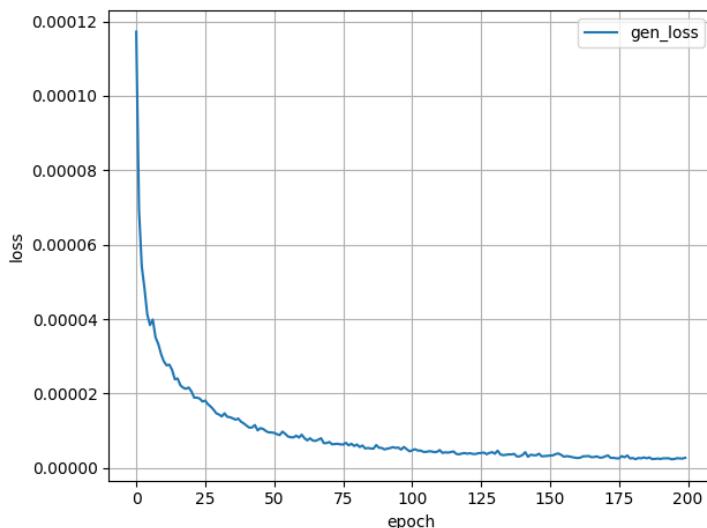


รูปที่ 21: ตัวอย่างผลลัพธ์ของการฝึกฝนโมเดล Solarnet

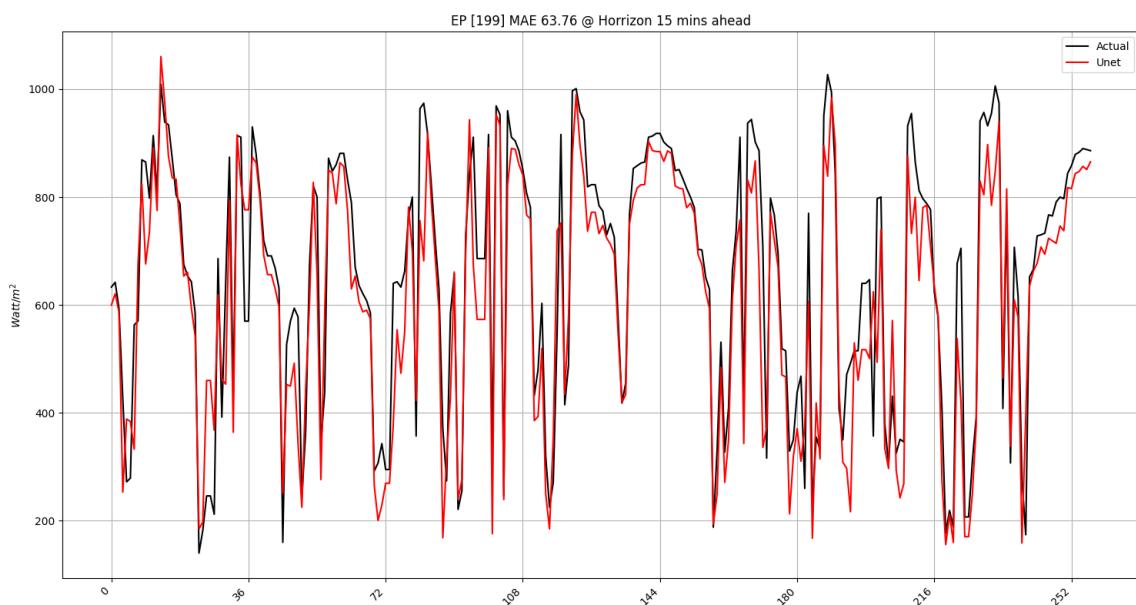
5.3.3 การฝึกฝนโมเดล UNET

- จำนวนโมเดลพารามิเตอร์ (Model Parameters) = 306,120 พารามิเตอร์
- ขนาดชุดตัวอย่าง (Batch Size) = 256

- ค่าอัตราการเรียนรู้ (Learning Rate) = 0.0001
- จำนวนรอบการฝึกฝน (Training Epochs) = 200 รอบ
- เครื่องมือเพิ่มประสิทธิภาพแบบอัตโนมัติ (Adam Optimizer)
- ฟังก์ชันสูญเสีย (Loss Function) คือ Mean Squared Error Loss



รูปที่ 22: กราฟแสดงค่าสูญเสียในการฝึกฝนโมเดล UNET



รูปที่ 23: ตัวอย่างผลลัพธ์ของการฝึกฝนโมเดล UNET

5.3.4 สรุปผลลัพธ์ที่ได้จากการฝึกฝนโมเดลสำหรับทำนายค่ารังสีแสงอาทิตย์

หลังจากได้ทำการฝึกฝนโมเดลสำหรับการทำนายค่ารังสีจากด้วยอาทิตย์ทั้ง 3 โมเดล 1. SUNSET 2. Solarnet และ 3. UNET จาก รูปที่ 18, รูปที่ 20 และ รูปที่ 22 จะเห็นว่าทั้งสามโมเดลมีค่าสูญเสียที่ลดลงตลอดการฝึกฝนโมเดล หรือก็คือ

โมเดลเกิดการเรียนรู้ขึ้นในขณะกระบวนการฝึกฝน และจากรูปที่ 19, รูปที่ 21 และ รูปที่ 23 จะเห็นว่าผลลัพธ์ที่ได้จากการฝึกฝนนั้นมีการประมาณค่ารังสีแสงอาทิตย์ที่มีแนวโน้มที่ดี โดยที่ โมเดล UNET นั้นมีค่าสูญเสียที่น้อยกว่าโมเดลอื่นอย่างมาก และผลลัพธ์ในการประมาณค่ารังสีแดงอาทิตย์มีความแม่นยำเป็นอย่างมากเมื่อเทียบกับโมเดลที่เหลือเนื่องจาก โมเดล UNET นั้นมีวิธีการเรียนรู้แบบ Regression ในขณะที่โมเดล SUNSET และ Solarnet เรียนรู้แบบ Forecasting

5.4 วิธีการวัดประสิทธิภาพ

5.4.1 อัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (PSNR: Peak Signal to Noise Ratio)

อัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด เป็นอัตราส่วนระหว่าง ค่ากำลังสูงสุดของสัญญาณ และ ค่ากำลังสัญญาณรบกวน เนื่องจากสัญญาณส่วนใหญ่มีช่วงไดนามิก (dynamic) มีค่าที่กว้างจึงนิยมแสดงประมาณในหน่วย เดซิเบล (Decibel/dB) โดยแสดงดังสมการ (21) [21]

$$PSNR = 20 \log_{10} \left(\frac{MAX_f}{\sqrt{MSE}} \right) \quad (21)$$

และ MSE (Mean Squared Error) คือ

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|f(i, j) - g(i, j)\|^2 \quad (22)$$

โดยที่

f คือ เมตริกข้อมูล (Matrix Data) ของภาพต้นฉบับ

g คือ เมตริกข้อมูล ของภาพที่ถูกลดTHON (Degraded Image)

m, n คือ จำนวนแถว (Row) และ หลัก (Column) ของพิกเซลในภาพ ตามลำดับ

i, j คือ ค่าดัชนี (index) ของแถวและหลักตามลำดับ

MAX_f คือ ค่าสูงสุดของสัญญาณในภาพต้นฉบับ

ค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุดระหว่างสามารถนำมาใช้ในการวัดประสิทธิภาพของภาพได้จากการ (21) โดยการหาค่าเฉลี่ยของผลต่างยกกำลังสอง (MSE) ทำให้เราสามารถเปรียบเทียบค่าในแต่ละพิกเซลของภาพที่ทำนายกับภาพที่ถูกต้อง ซึ่งทำให้เราทราบถึงความแตกต่างของภาพทั้งสอง

5.4.2 ค่าเฉลี่ยผลต่างกำลังสอง (MSE: Mean Squared Error)

ค่าเฉลี่ยผลต่างกำลังสอง (MSE) เป็นหนึ่งในค่าพื้นฐานที่ใช้ในการวัดความผิดพลาด (Error) ของโมเดล โดยหาผลต่างระหว่างข้อมูลทำนาย (Predicted) และ ข้อมูลจริง (Ground Truth) จากนั้นนำไปยกกำลังสอง และหาค่าเฉลี่ยในข้อมูลทุกตัวดังสมการ (23)

$$MSE = \frac{1}{N} \sum_{i=0}^N (\hat{y}_i - y_i)^2 \quad (23)$$

โดยที่

N คือ จำนวนข้อมูลทั้งหมด

\hat{y} คือ ข้อมูลที่โมเดลทำนาย (Predicted Data)

y คือ ข้อมูลจริงที่ถูกต้อง (Ground Truth)

5.4.3 ค่าเฉลี่ยผลต่างสัมบูรณ์ (MAE : Mean Absolute Error)

ค่าเฉลี่ยผลต่างสัมบูรณ์ (MAE) เป็นอีกหนึ่งค่าพื้นฐานที่ใช้ในการวัดความผิดพลาด (Error) ของโมเดล ซึ่งมีลักษณะคล้ายกับ MSE แต่ต่างกันตรงที่ เมื่อเราหาผลต่างระหว่างข้อมูลทำนาย (Predicted) และ ข้อมูลจริง (Ground Truth) และทำการหาค่าสัมบูรณ์ (Absolute) แทนที่จะยกกำลังสอง ดังสมการ (24)

$$MAE = \frac{1}{N} \sum_{i=0}^N |\hat{y}_i - y_i| \quad (24)$$

โดยที่

N คือ จำนวนข้อมูลทั้งหมด

\hat{y} คือ ข้อมูลที่โมเดลทำนาย (Predicted Data)

y คือ ข้อมูลจริงที่ถูกต้อง (Ground Truth)

5.5 ผลจากการทดสอบโมเดล (Testing Model)

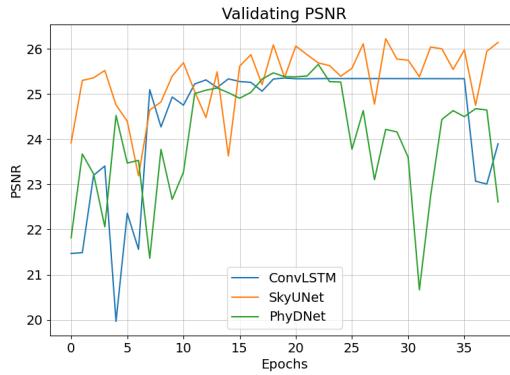
ในส่วนนี้ได้ไปจับตัวอย่างข้อมูลตรวจสอบทั้งหมด 1,091 ตัวอย่าง (5%) และ ชุดข้อมูลทดสอบทั้งหมด 3,276 ตัวอย่าง (15%) โดยได้มีการวัดประสิทธิภาพของโมเดลด้วยค่า Mean Squared Error (MSE), Mean Absolute Error (MAE) และ Peak signal to Noise Ratio (PSNR) เป็นตัววัดประสิทธิภาพ

5.5.1 ผลลัพธ์จากการทดสอบโมเดลทำนายภาพถ่ายท้องฟ้า

โมเดลทำนายภาพถ่ายท้องฟ้าที่มีประสิทธิภาพจะต้องให้ผลลัพธ์เป็นภาพที่มีความคมชัด และมีความคลาดเคลื่อนจากภาพจริงมากที่สุด จากการทดสอบโมเดลด้วยชุดตรวจสอบได้ผลลัพธ์ที่ดีที่สุดของแต่ละโมเดลดัง ตารางที่ 1 และจะพบว่าโมเดล SkyUNet นั้นมีค่า อัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (PSNR) ที่มากกว่าโมเดล ConvLSTM และ PhyDNet+GAN เล็กน้อย และยังมีค่าเฉลี่ยผลต่างกำลังสอง (MSE) และ ค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE) ที่น้อยกว่าอีกทั้ง 2 โมเดล

ตารางที่ 1: ตารางผลการตรวจสอบโมเดลทำนายรังสีแสงอาทิตย์

Model	Validating PSNR	Validating MSE	Validation MAE	Epoch
SkyUNet	26.22	512.73	12.91	28
ConvLSTM	25.35	526.17	14.53	19
PhyDNet + GAN	25.65	529.89	13.44	22



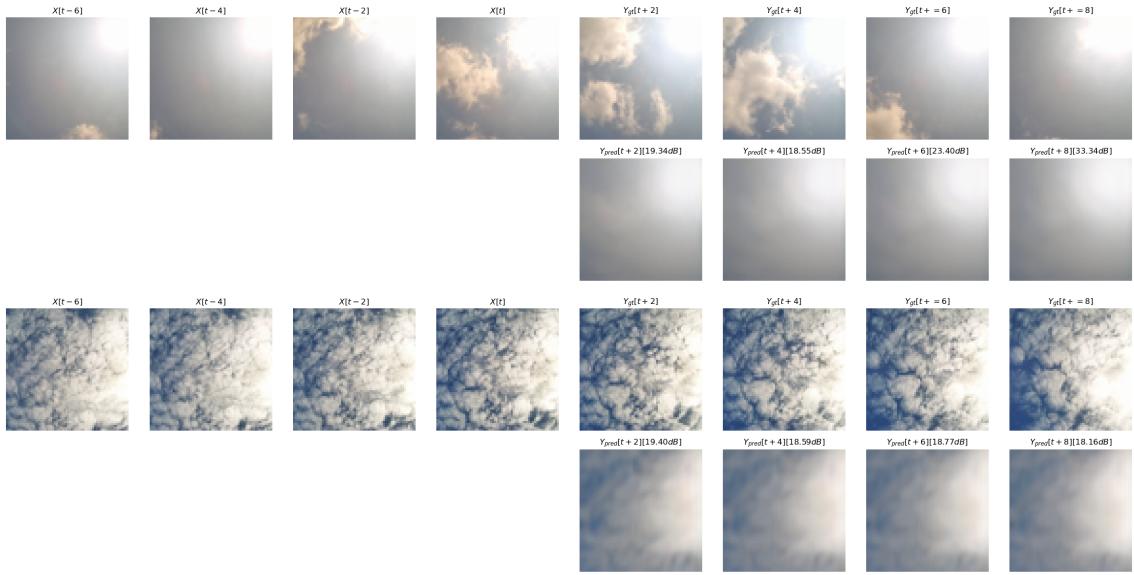
รูปที่ 24: กราฟแสดงค่า PSNR ในกระบวนการตรวจสอบเทียบในแต่ละโมเดลทำนายภาพถ่ายท้องฟ้า

หลังจากทำการตรวจสอบโมเดล (Validation) ทำให้ทราบว่าแต่ละโมเดลนั้นมีประสิทธิภาพมากที่สุดในการฝึกฝนกีครั้ง และนำโมเดลทั้งสามไปทำนายชุดข้อมูลทดสอบซึ่งได้ผลลัพธ์ดัง ตารางที่ 1 จะพบว่าในกระบวนการทดสอบโมเดล (Testing Models) โมเดล PhyDNet + GAN นั้นมีค่า อัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (PSNR) ที่มากที่สุดและมีค่าเฉลี่ยผลต่างกำลังสอง (MSE) และ ค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE) ที่น้อยที่สุดอีกด้วย

ตารางที่ 2: ตารางผลการทดสอบโมเดลทำนายภาพถ่ายท้องฟ้า

Model	Testing PSNR	Testing MSE	Testing MAE
SkyUNet	26.37	512.73	12.84
ConvLSTM	25.50	532.89	14.49
PhyDNet + GAN	26.99	414.44	11.35

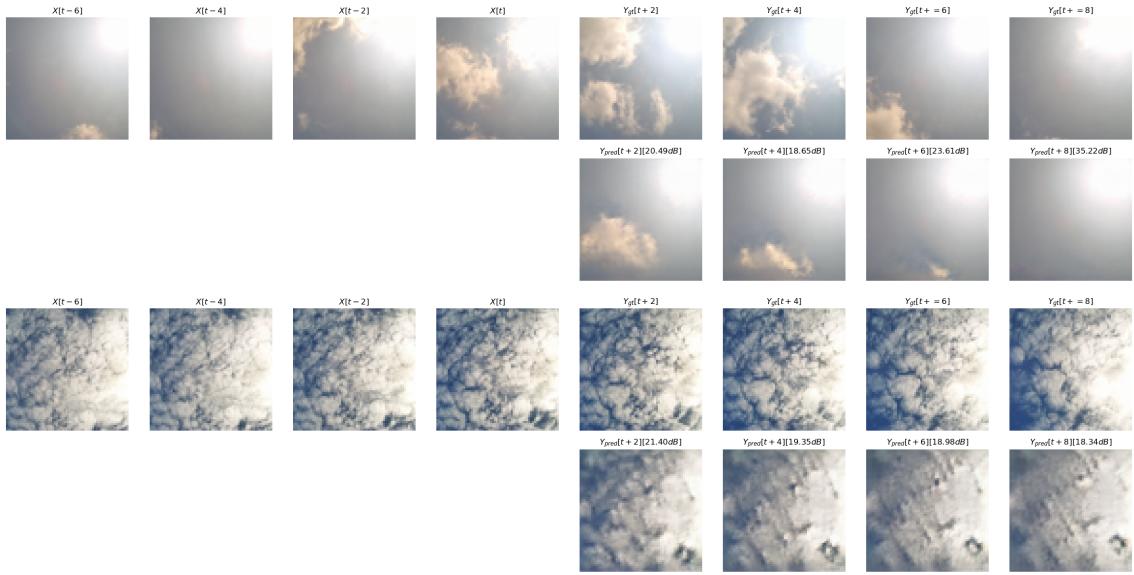
เมื่อทำการพิจารณาจากรูปตัวอย่างผลลัพธ์การทำนายภาพถ่ายท้องฟ้าของแต่ละโมเดลจะพบว่าใน รูปที่ 26 และ รูปที่ 25 โมเดล ConvLSTM และ SkyUNet ยังไม่สามารถที่จะให้ผลลัพธ์ที่มีความสมจริงและถูกต้องได้ถึงแม้ว่าค่า อัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (PSNR) ที่มากที่สุดและมีค่าเฉลี่ยผลต่างกำลังสอง (MSE) และ ค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE) จะไม่ได้น้อยกว่าของตัว โมเดล PhyDNet มากนักซึ่งจาก รูปที่ 27 จะเห็นว่าผลลัพธ์ที่ได้ออกมานั้นมีความคมชัดที่มากกว่า และ เหนือกว่าในความจริงมากกว่า



รูปที่ 25: รูปตัวอย่างการทำนายภาพถ่ายท้องฟ้าของโมเดล SkyUNet



รูปที่ 26: รูปตัวอย่างการทำนายภาพถ่ายท้องฟ้าของโมเดล ConvLSTM



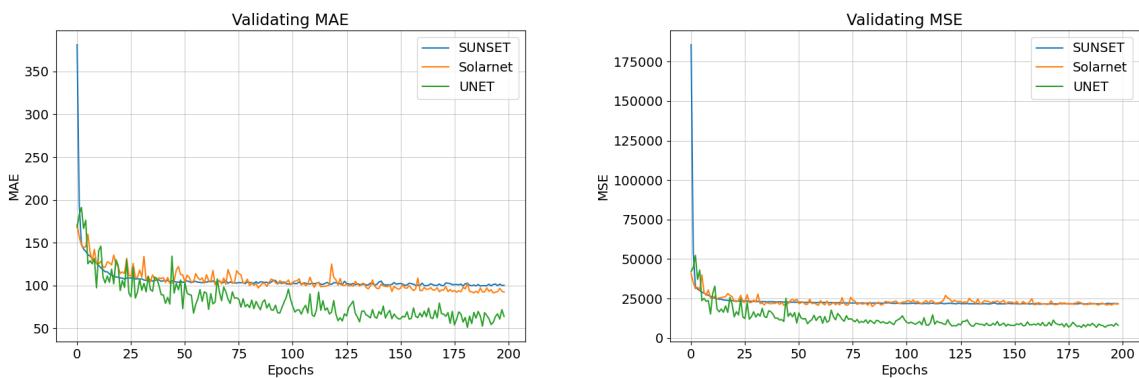
รูปที่ 27: รูปตัวอย่างการคำนวณภาพถ่ายท้องฟ้าของโมเดล PhyDNet

5.5.2 ผลลัพธ์จากการทดสอบโมเดลทำนายรังสีแสงอาทิตย์

สำหรับโมเดลทำนายค่ารังสีแสงอาทิตย์ที่มีประสิทธิภาพจะต้องให้ผลลัพธ์ที่มีความแม่นยำและใกล้เคียงกับค่ารังสีแสงอาทิตย์จริงมากที่สุด โดยสามารถวัดประสิทธิภาพได้จากค่าเฉลี่ยผลต่างกำลังสอง (MSE) และ ค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE)

ตารางที่ 3: ตารางผลการตรวจสอบและทดสอบโมเดลทำนายรังสีแสงอาทิตย์

Model (Technique)	Validating MSE	Validating MAE	Testing MSE	Testing MAE	Epoch
SUNSET (Forecasting)	21,620.99	99.23	22,819.60	103.56	187
Solarnet (Forecasting)	20,818.76	91.12	21,020.62	91.40	193
UNET (Regression)	6,678.61	50.88	6,132.15	49.42	181

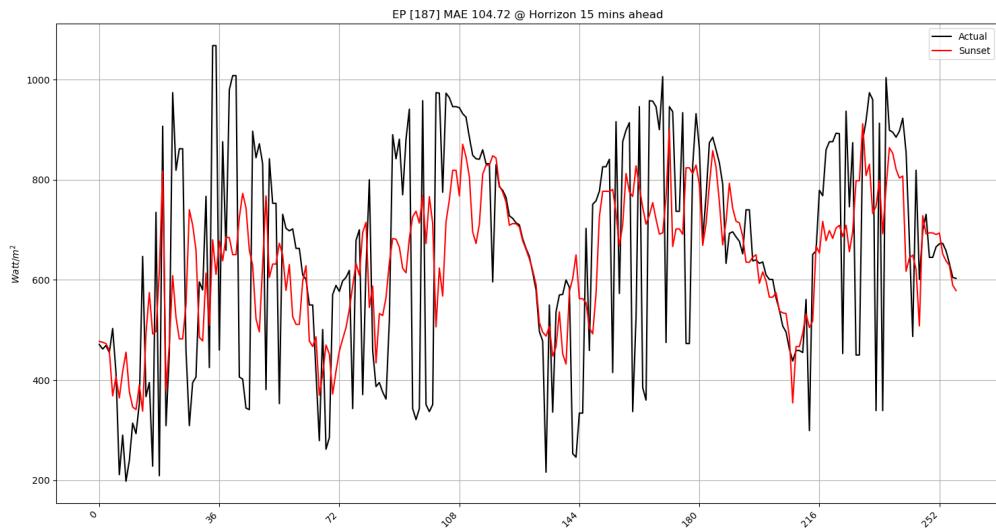


รูปที่ 28: กราฟแสดงค่า MAE และ MSE ในกระบวนการตรวจสอบเทียบกับแต่ละโมเดลทำนายค่ารังสีแสงอาทิตย์

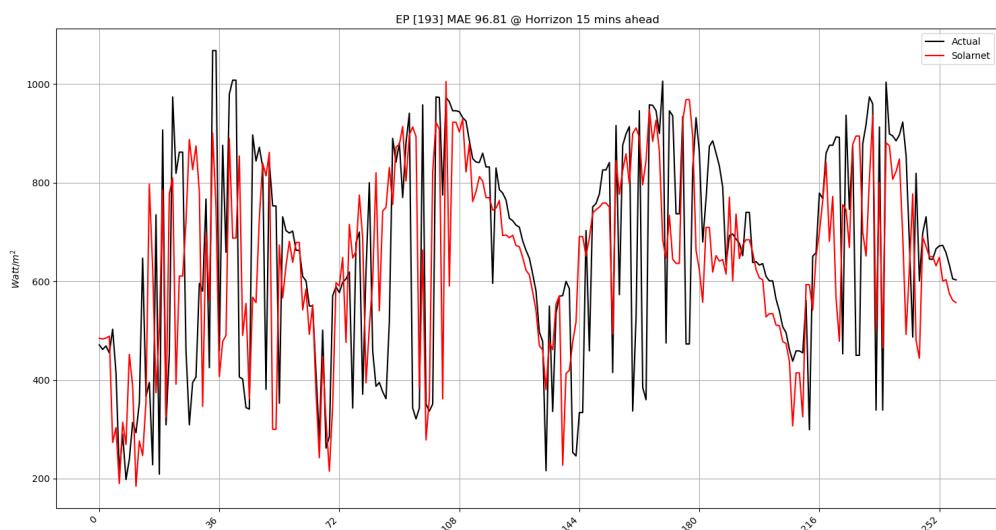
จากตารางที่ 3 จะเห็นว่าโมเดล UNET มีค่าเฉลี่ยผลต่างกำลังสอง (MSE) และ ค่าเฉลี่ยของผลต่างสัมบูรณ์ (MAE) ที่น้อยกว่าโมเดล SUNSET และ Solarnet เป็นอย่างมาก และยังเห็นได้จากรูปที่ 29 ว่าโมเดล UNET มีการทำนาย

ค่ารังสีแสงอาทิตย์ได้ด้วยความแม่นยำสูงและ เป็นโมเดล มีประสิทธิภาพที่ดีที่สุดโดยสามารถเอาชนะ (Outperformed) โมเดล SUNSET และ Solarnet ได้ และเมื่อพิจารณาเป็นประเภทในการทำนายจากโมเดล Solarnet และ SUNSET ที่มี การทำนายแบบ Forecasting และ UNET ที่มีการทำนายแบบ Regression จะพบว่าการทำ Regression มีประสิทธิภาพ ในการแก้ปัญหาความไม่แน่นอนได้ดีกว่าการทำ Forecasting

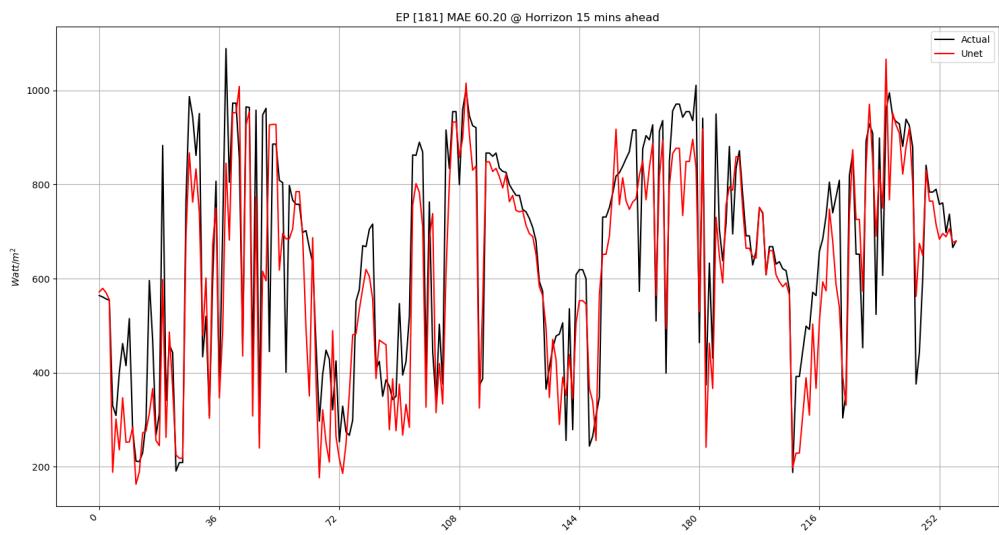
เมื่อเราเปรียบเทียบเฉพาะโมเดล Solarnet และ SUNSET จะพบว่าโมเดล Solarnet มีประสิทธิภาพดีกว่า SUNSET ที่ มีข้อดีใหญ่สุดและโดยที่รับทั้งภาพและค่ารังสีแสงอาทิตย์เป็นข้อมูลเข้ามา นั้นอาจจะแปลได้ว่า ค่ารังสีแสงอาทิตย์มีความ ผันผวนค่อนข้างสูง หรือ ค่ารังสีแสงอาทิตย์ในอดีตอาจจะไม่ได้มีความเกี่ยวข้องกับค่ารังสีแสงอาทิตย์ในปัจจุบันมากพอที่จะ ช่วยเพิ่มความแม่นยำการพยากรณ์ได้



รูปที่ 29: ตัวอย่างผลลัพธ์การทดสอบโมเดล SUNSET



รูปที่ 30: ตัวอย่างผลลัพธ์การทดสอบโมเดล Solarnet



รูปที่ 31: ตัวอย่างผลลัพธ์การทดสอบโมเดล UNET

6 บทสรุป

6.1 สรุปผลการดำเนินการ

จากการทดสอบของโมเดลที่นำมายกมาพิจารณาท้องฟ้าพบว่า PhyDNet + GAN สามารถเอาชนะโมเดล SkyUNet และ ConvLSTM โดยเห็นได้ชัดเจนจากภาพของผลลัพธ์ที่มีความคมชัดและสมจริงของภาพถ่ายท้องฟ้ามากกว่า และ ค่าที่ใช้วัดประสิทธิภาพ ในส่วนของการทดสอบโมเดลสำหรับทำนายค่ารังสีแสงอาทิตย์ SUNSET, Solarnet และ UNET พบร่วมกับโมเดล UNET เป็นโมเมลที่มีประสิทธิภาพที่ดีที่สุด โดยมีค่า MSE ที่ต่ำกว่าอีกสองโมเดล กว่า 3 เท่า และมีค่า MAE ต่ำกว่าประมาณ 1 เท่าของโมเดล SUNSET และสามารถสังเกตุได้จากผลลัพธ์ตัวอย่างที่มีความแม่นยำในการทำนายค่ารังสีแสงอาทิตย์

จากการพิจารณาตามประเภทในการทำนายจะพบว่าการทำ Regression มีประสิทธิภาพในการแก้ปัญหาความไม่แน่นอนได้ดีกว่าการทำ Forecasting และเมื่อเราเปรียบเทียบเฉพาะโมเดล Solarnet และ SUNSET จะพบว่าโมเดล Solarnet มีประสิทธิภาพดีกว่า SUNSET นั้นอาจจะแปลได้ว่า ค่ารังสีแสงอาทิตย์มีความผันผวนค่อนข้างสูง หรือ ค่ารังสีแสงอาทิตย์ในอดีตอาจจะไม่ได้มีความเกี่ยวข้องกับค่ารังสีแสงอาทิตย์ในปัจจุบันมากพอที่จะช่วยเพิ่มความแม่นยำการพยายามได้

โครงการนี้ยังสามารถที่จะต่อยอดได้โดย การนำหลักการการปรับจูนโมเดล (Fine-Tuning Model) เพื่อทำให้โมเดลมีความแม่นยำ นอกจากนี้ยังสามารถที่จะปรับปรุงองค์ประกอบต่างๆของโมเดล หรือ การนำโมเดลอื่นที่สามารถช่วยในการแก้ไขปัญหาต่างๆที่พบในโครงการนี้ได้เพื่อทำให้โมเดลมีประสิทธิภาพมากขึ้นและได้ผลลัพธ์ตามที่คาดหวัง เช่น ใช้เทคนิคอื่นๆในการทำนายการเคลื่อนตัวของเมฆ หรือ ปรับปรุงโมเดลให้มีความคมชัดของภาพที่ทำนาย เป็นต้น

6.2 สรุปองค์ความรู้ที่ได้จากการศึกษา

1. เรียนรู้และเข้าใจการทำงานของหลักการเรียนรู้เชิงลึก (Deep Learning) และวิธีการฝึกฝนและทดสอบโมเดล
2. เข้าใจหลักการทำงานและประโยชน์ โครงสร้างสถาปัตยกรรม U-Net, VQ-VAE, PhyDNet, GANs, LSTM, CNN เป็นต้น
3. ได้เรียนรู้ในการจัดการกับข้อมูลเพื่อให้เหมาะสมกับการนำไปใช้งานและการป้อนเข้าให้กับโมเดล

6.3 ปัญหา อุปสรรค และแนวทางการแก้ไข

1. เนื่องจากการโครงการนี้มีการใช้โมเดลและข้อมูลที่ใช้ทรัพยากรในระดับหนึ่ง จึงทำให้ในช่วงแรกจึงเกิดปัญหาด้านอุปกรณ์ โดยภัยหลังทางกลุ่มได้จัดหาอุปกรณ์และมีทรัพยากรที่พร้อมให้ใช้งาน
2. ในช่วงแรกยังขาดองค์ความรู้ในด้านการเรียนรู้เชิงลึก จึงต้องมีการศึกษาและค้นคว้าความรู้จากอินเตอร์เน็ต และงานวิจัย

6.4 กิติกรรมประกาศ

ขอขอบคุณ อ.ดร. สุวิชญา สุวรรณวิมลกุล อาจารย์ที่ปรึกษาโครงการ ที่ค่อยนำแนวและชี้แนะแนวทางให้ ทั้งรับฟังและช่วยแก้ไขปัญหานาฬาสิงที่นิสิตได้พับเจอเสนอ ตลอดจนให้คำปรึกษาและแนะนำนิสิต อาจารย์คอยตอบคำถามพร้อม

แนะนำวิธีแก้ไขและพัฒนาโครงการให้นิสิตอยู่เสมอ โดยโครงการนี้สำเร็จลุล่วงได้จากความช่วยเหลือและคำแนะนำของอาจารย์ทั้งสิ้น

ເອກສາຣອ້າງອີງ

- [1] Mohd Rizwan Sirajuddin Shaikh, Santosh B. Waghmare , Suvarna Shankar Labade , Pooja Vittal Fuke, Anil Tekale, “A review paper on electricity generation from solar energy,” Sep, 2017.
- [2] Leron Julian, Aswin C. Sankaranarayanan, “Precise forecasting of sky images using spatial warping,” 2021.
- [3] Anas Al-lahham, Obaidah Theeb, Khaled Elalem, Tariq A. Alshawi, and Saleh A. Alshebeili, “Sky imager-based forecast of solar irradiance using machine learning,” 2022.
- [4] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, “Estimation of solar irradiance using ground-based whole sky imagers,” *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016.
- [5] Talha Ahmad Siddiqui, Samarth Bharadwaj, and S. Kalyanaraman, “A deep learning approach to solar-irradiance forecasting in sky-videos,” IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 2166–2174.
- [6] Q. Paletta, G. Arbod, and J. Lasenby, “Omnivision forecasting: Combining satellite and sky images for improved deterministic and probabilistic intra-hour solar energy predictions,” *Applied Energy*, vol. 336, p. 120818, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306261923001824>
- [7] Alexandre Xavier, “An introduction to ConvLSTM,” [Accessed 18 Apr 2024]. [Online]. Available: <https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7>
- [8] Vincent Le Guen, Nicolas Thome, “Disentangling Physical Dynamics from Unknown Factors for Unsupervised Video Prediction,” In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020, 2020.
- [9] Y. Nie, E. Zelikman, A. Scott, Q. Paletta, and A. Brandt, “Skygpt: Probabilistic ultra-short-term solar forecasting using synthetic sky images from physics-constrained videogpt,” *Advances in Applied Energy*, vol. 14, p. 100172, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666792424000106>
- [10] Y. Sun, G. Székely, and A. R. Brandt, “Solar pv output prediction from video streams using convolutional neural networks,” *Energy Environ. Sci.*, vol. 11, pp. 1811–1818, 2018. [Online]. Available: <http://dx.doi.org/10.1039/C7EE03420B>
- [11] C. Feng, J. Zhang, W. Zhang, and B.-M. Hodge, “Convolutional neural networks for intra-hour solar forecasting based on sky image sequences,” *Applied Energy*, vol. 310, p. 118438, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306261921016639>

- [12] Dharmaraj, “Convolutional Neural Networks (CNN) — Architecture Explained,” [Accessed 20 Apr 2024]. [Online]. Available: <https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243>
- [13] Ottavio Calzone, “An Intuitive Explanation of LSTM),” [Accessed 19 Apr 2024]. [Online]. Available: <https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c>
- [14] Sirinart Tangruamsub, “Long Short-Term Memory (LSTM),” [Accessed 19 Apr 2024]. [Online]. Available: <https://medium.com/@sinart.t/long-short-term-memory-lstm-e6cb23b494c6>
- [15] O. Ronneberger, P.Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation.” In Medical Image Computing and Computer-Assisted Intervention (MICCAI): Springer, 2015, pp. 234–241.
- [16] Nikhil Tomar, “What is UNET?” [Accessed 19 Nov 2023]. [Online]. Available: <https://medium.com/analytics-vidhya/what-is-unet-157314c87634>
- [17] O. Ronneberger, P.Fischer, and T. Brox, “A comparison of satellite cloud motion vectors techniques to forecast intra-day hourly solar global horizontal irradiation, solar energy,” vol. 233, 2022, pp. 46–60.
- [18] Rohith Gandhi, “Generative Adversarial Networks — Explained,” [Accessed 20 Apr 2024]. [Online]. Available: <https://towardsdatascience.com/generative-adversarial-networks-explained-34472718707a>
- [19] Jonathan Hui, “GAN — What is Generative Adversarial Networks GAN?” [Accessed 20 Apr 2024]. [Online]. Available: <https://jonathan-hui.medium.com/gan-whats-generative-adversarial-networks-and-its-application-f39ed278ef09>
- [20] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition.” Computational and Biological Learning Society, 2015, pp. 1–14.
- [21] NI Emerson Website, “Peak signal-to-noise ratio as an image quality metric,” 23 Mar 2023, [Accessed 19 Nov 2023]. [Online]. Available: <https://www.ni.com/en/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module/peak-signal-to-noise-ratio-as-an-image-quality-metric.html>