

การใช้ปัญญาประดิษฐ์ในการสร้างภาพจำลองเพื่อทดแทนภาพถ่ายอวัยวะบนใบหน้าของบุคคลจริงสำหรับประยุกต์ใช้ในงานด้านการวิเคราะห์ใบหน้า



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชานวัตกรรม วิศวกรรมแพทย์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2566

USING ARTIFICIAL INTELLIGENCE TO GENERATE SYNTHETIC
IMAGES TO REPLACE ORGAN PHOTOGRAPHS ON
REAL HUMAN FACES FOR FACIAL ANALYSIS APPLICATIONS



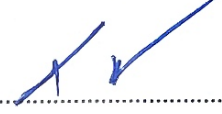
A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Biomedical Innovation Engineering
Suranaree University of Technology

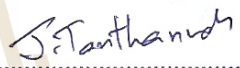
Academic Year 2023


การใช้ปัญญาประดิษฐ์ในการสร้างภาพจำลองเพื่อทดแทนภาพถ่ายอวัยวะ
บนใบหน้าของบุคคลจริงสำหรับประยุกต์ใช้ในงานด้านการวิเคราะห์ที่ใบหน้า

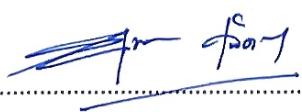
มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรปริญญามหาบัณฑิต


คณะกรรมการสอบวิทยานิพนธ์


.....
(ผู้ช่วยศาสตราจารย์ ดร.สุขเกษม วัชรชัยสกุล)
ประธานกรรมการ


.....
(ผู้ช่วยศาสตราจารย์ ดร.เจษฎา ตันทนงช)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)


.....
(รองศาสตราจารย์ ดร.พิสมัย กิตติภูมิ)
กรรมการ


.....
(รองศาสตราจารย์ ดร.ยุพาพร รักสกุลพิวัฒน์)
รองอธิการบดีฝ่ายวิชาการและประกันคุณภาพ


.....
(รองศาสตราจารย์ ดร.พรศิริ จงกล)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

จักรภัทร เวทศิลป์ : การใช้ปัญญาประดิษฐ์ในการสร้างภาพจำลองเพื่อทดแทนภาพถ่าย
อวัยวะบนใบหน้าของบุคคลจริงสำหรับประยุกต์ใช้ในงานด้านการวิเคราะห์ใบหน้า (USING
ARTIFICIAL INTELLIGENCE TO GENERATE SYNTHETIC IMAGES TO REPLACE
ORGAN PHOTOGRAPHS ON REAL HUMAN FACES FOR FACIAL ANALYSIS
APPLICATIONS)

อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.เจษฎา ตันทนุช, 55 หน้า.

คำสำคัญ: ปัญญาประดิษฐ์, การตรวจหาวัตถุ, YoloV5, การจำลองดวงตา จมูก ปาก ของมนุษย์

งานวิจัยนี้มีวัตถุประสงค์เพื่อสร้างฐานข้อมูลภาพของอวัยวะบนใบหน้า ได้แก่ ดวงตา จมูก และปาก โดยใช้วิธีการตรวจหาวัตถุเพื่อนำไปใช้ในงานด้านการเก็บข้อมูลทางการแพทย์ในอนาคต สำหรับผู้ป่วยที่มีปัญหาเกี่ยวกับใบหน้า งานวิจัยนี้ได้ใช้ซอฟต์แวร์ปัญญาประดิษฐ์สองประเภท ได้แก่ Midjourney และ Stylegan2-ADA-Pytorch เพื่อสร้างภาพจำลองของดวงตา จมูก และปากของมนุษย์ ผลการศึกษาพบว่าสามารถสร้างภาพจำลองของดวงตา จมูก และปากของมนุษย์ที่มีลักษณะแตกต่างกันได้ โดยมีจำนวนภาพที่สร้างขึ้นได้แก่ ดวงตา 459 ภาพ จมูก 483 ภาพ และปาก 462 ภาพ จากนั้นนำภาพเหล่านี้ไปสร้างเทมเพลตการฝึกสอนสำหรับการตรวจหาวัตถุโดยใช้ YOLOv5 ขั้นตอนถัดไปของการวิจัยคือการทดสอบการตรวจหาดวงตา จมูก และปากจากชุดข้อมูลภาพใบหน้า จำลองจำนวน 3,161 ภาพที่จัดเตรียมล่วงหน้าจาก <https://generated.photos/> ผลการวิจัยพบว่า การตรวจหาดวงตาจากภาพใบหน้าที่มีความถูกต้อง ความแม่นยำ และค่าเรียกคืนเท่ากับร้อยละ 98.58 การตรวจหาจมูกจากภาพใบหน้าที่มีความถูกต้องและค่าเรียกคืนเท่ากับร้อยละ 91.93 และค่าความแม่นยำเท่ากับร้อยละ 100 ในทำนองเดียวกัน การตรวจหาปากจากภาพใบหน้าที่มีความถูกต้องและค่าเรียกคืนเท่ากับร้อยละ 97.34 และค่าความแม่นยำเท่ากับร้อยละ 100 ตามลำดับ

สาขาวิชานวัตกรรม วิศวกรรมแพทย์
ปีการศึกษา 2566

ลายมือชื่อนักศึกษาจักรภัทร.....
ลายมือชื่ออาจารย์ที่ปรึกษาJ. Tanthunth.....

JAKAPAT WETTHASIN : USING ARTIFICIAL INTELLIGENCE TO GENERATE
SYNTHETIC IMAGES TO REPLACE ORGAN PHOTOGRAPHS ON REAL HUMAN
FACES FOR FACIAL ANALYSIS APPLICATIONS

THESIS ADVISOR: ASST. PROF. JESSADA TANTHANUCH, Ph.D., 55 PP.

Keywords: Artificial intelligence, simulated facial organ images, object detection,
YOLOv5

The objective of this research is to establish a database of facial organ images, including eyes, nose, and mouth, using object detection methods for future applications in medical data collection for patients with facial-related issues. Two types of artificial intelligence software, Midjourney and Stylegan2-ADA-Pytorch, were employed to generate synthetic images of human eyes, nose, and mouth. The study found that synthetic images of human eyes, nose, and mouth with varying characteristics could be generated, comprising 459, 483, and 462 images, respectively. Subsequently, these images were used to create training templates for object detection using YOLOv5. The next stage of the research involved testing the detection of eyes, nose, and mouth from a dataset of 3,161 artificial facial images prepared in advance from <https://generated.photos/>. The results revealed that the detection of eyes from facial images had an accuracy, precision, and recall of 98.58%. The detection of the nose from facial images had an accuracy and recall of 91.93% and a precision of 100%, respectively. Similarly, the detection of the mouth from facial images had an accuracy and recall of 97.34% and a precision of 100%, respectively.

School of Biomedical Innovation Engineering
Academic Year 2023

Student's Signature
Advisor's Signature
J. Tanthanuch

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จลุล่วงอย่างยอดเยี่ยม เนื่องจากได้รับคำปรึกษาและคำแนะนำอย่างดีเยี่ยมจากบุคคลต่อไปนี้

ผู้ช่วยศาสตราจารย์ ดร.สุขเกษม วัชรมัยสกุล ประธานกรรมการที่ปรึกษาวิทยานิพนธ์ ซึ่งได้ให้คำปรึกษาและชี้แนะเกี่ยวกับการศึกษาและการดำเนินการวิจัย

ผู้ช่วยศาสตราจารย์ ดร.เจษฎา ตัณฑนุช อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้ให้คำแนะนำช่วยเหลือ และ แก้ไขปรับปรุงข้อบกพร่องด้านการดำเนินการวิจัย แนะนำการเขียน การตรวจแก้ไข วิทยานิพนธ์เล่มนี้จนเสร็จสมบูรณ์ ตลอดจนคำแนะนำในการรับทุนการศึกษาในระดับบัณฑิตศึกษา

คณาจารย์ทุกท่านในหลักสูตรสาขาวิชาวิศวกรรมวิศวกรรมชีวการแพทย์ มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ให้คำแนะนำและสนับสนุนในการศึกษาและการวิจัย

ผู้วิจัยขอขอบคุณความกรุณาและการสนับสนุนจากทุกท่านที่ได้ช่วยเหลือจนเสร็จสมบูรณ์และประสบความสำเร็จในวิทยานิพนธ์นี้ ขอขอบคุณอย่างยิ่งด้วยความจริงใจ

จักรภัทร เวทศิลป์

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่	
1 บทนำ	2
1.1 ที่มาและความสำคัญของปัญหาการวิจัย	2
1.2 วัตถุประสงค์ของการศึกษา.....	3
1.3 ขอบเขตของงานวิจัย.....	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2 ปรัชณววรรณกรรมและงานวิจัยที่เกี่ยวข้อง	4
2.1 ปัญญาประดิษฐ์ (Artificial Intelligence: AI).....	5
2.2 การเรียนรู้ของเครื่อง (Machine learning).....	6
2.2.1 การเรียนรู้จากข้อมูล (Learning From Data).....	6
2.2.2 ประเภทของการเรียนรู้.....	6
2.3 การประยุกต์ใช้งาน.....	6
2.4 การเรียนรู้เชิงลึก (Deep learning).....	7
2.4.1 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network).....	8
2.5 ปัญญาประดิษฐ์สำหรับการสร้างภาพเสมือนจริง (Artificial Intelligence For Photorealistic Image Generation).....	11

สารบัญ (ต่อ)

หน้า

2.5.1 การประยุกต์ของปัญญาประดิษฐ์ในการสร้างภาพที่เหมือนจริง (photorealistic image generation).....	11
2.5.2 Generative Adversarial Networks: GANs.....	11
2.5.3 Variational Autoencoders (VAEs): VAEs	11
2.5.4 การโอนรูปแบบสไตล์และการเพิ่มความละเอียด	11
2.5.5 Conditional GANs	12
2.5.6 การประยุกต์ใช้	12
2.6 Generative Adversarial Networks (GANs).....	12
2.7 ภาษาไพธอน (Python) และคลังโปรแกรม (Libraries).....	14
2.7.1 ไพธอน (Python).....	14
2.7.2 Pytorch.....	14
2.7.3 TensorFlow.....	14
2.8 การตรวจหาวัตถุ (Object Detection).....	14
2.8.1 ขั้นตอนวิธีเบื้องต้นการตรวจจับวัตถุ.....	15
2.8.2 R-CNN	15
2.8.3 Region-based convolutional neural networks	15
2.8.4 Faster R-CNNs.....	15
2.8.5 Yolo (You Only Look Once).....	16
2.9 YoloV5 (You Only Look Once).....	16
2.9.1 การแบ่งภาพออกเป็นส่วนย่อย	16
2.9.2 การสกัดคุณลักษณะของวัตถุ	16
2.9.3 การคาดเดาประเภทของวัตถุ.....	16
2.9.4 การคาดเดาดำแหน่งของวัตถุ	17
2.9.5 การปรับปรุงผลลัพธ์	17

สารบัญ (ต่อ)

	หน้า
2.9.6 การค้นหาวัตถุในภาพ	17
2.10 เครื่องมือในการวิเคราะห์ประสิทธิภาพ	17
2.11 แนวคิดและงานวิจัยที่เกี่ยวข้อง.....	19
3 วิธีดำเนินการวิจัย.....	22
3.1 เครื่องมือที่ใช้ในการทำวิจัย.....	23
3.2 การศึกษาข้อมูลที่เกี่ยวข้อง.....	23
3.3 FlowChart การทำงาน.....	24
3.4 ขั้นตอนการทำงานวิจัย.....	25
4 ผลการวิจัยและอภิปรายผล	27
4.1 ผลการทดลองการสร้างภาพดวงตา จมูก และ ปาก.....	27
4.2 คำนวณประสิทธิภาพของ ปัญญาประดิษฐ์ ด้วย เมทริกซ์ความสับสน	32
4.3 ทดลองใช้กับรูปภาพที่ถูกสร้างขึ้นด้วยปัญญาประดิษฐ์.....	36
4.4 วิเคราะห์และอภิปรายผล	37
5 สรุปและข้อเสนอแนะ.....	39
5.1 สรุปผลการวิจัย.....	39
5.2 ข้อเสนอแนะ.....	40
รายการอ้างอิง	41
ภาคผนวก ก.....	44
ภาคผนวก ข.....	53
ประวัติผู้เขียน.....	55

สารบัญตาราง

ตารางที่	หน้า
2.1 เมทริกซ์ความสับสน (Confusion Matrix)	17
3.1 เครื่องมือ วัสดุ และอุปกรณ์ที่ใช้ในงานวิจัย.....	22
4.1 เมทริกซ์ความสับสนของดวงตา.....	33
4.2 ค่าเมทริกซ์ความสับสนของดวงตาเมื่อคิดเป็นร้อยละ	34
4.3 เมทริกซ์ความสับสนของจมูก	34
4.4 ค่าเมทริกซ์ความสับสนของจมูกเมื่อคิดเป็นร้อยละ.....	35
4.5 เมทริกซ์ความสับสนของปาก	35
4.6 ค่าเมทริกซ์ความสับสนของปากเมื่อคิดเป็นร้อยละ.....	36

สารบัญรูป

รูปที่	หน้า
2.1 ภาพแสดงประเภทของ Machine Learning	5
2.2 ภาพเปรียบเทียบระหว่าง Neural Network แบบธรรมดา	7
2.3 ตัวอย่างตัวกรองเพื่อหาเส้นตรงแนวทแยงสีขาว	8
2.4 การทำงานของ Padding	9
2.5 การทำงานของ Max Pooling	9
2.6 ภาพการทำงานของ StyleGAN	13
2.7 ภาพขั้นตอนการผสมภาพของ StyleGAN.....	13
4.1 ตัวอย่างภาพของดวงตาที่สร้างขึ้นโดย Midjourney	28
4.2 ตัวอย่างภาพของจมูกที่สร้างขึ้นโดย Midjourney.....	28
4.3 ตัวอย่างภาพของปากที่สร้างขึ้นโดย Midjourney.....	29
4.4 ตัวอย่างภาพของดวงตาที่ถูกปรับปรุงขึ้นใหม่โดย Stylegan2-Ada-Pytorch.....	30
4.5 ตัวอย่างภาพของจมูกที่ถูกปรับปรุงขึ้นใหม่โดย Stylegan2-Ada-Pytorch	30
4.6 ตัวอย่างภาพของปากที่ถูกปรับปรุงขึ้นใหม่โดย Stylegan2-Ada-Pytorch	31
4.7 ตัวอย่างข้อมูลที่ได้จากการระบุป้ายโดยใช้ เว็บไซต์ https://www.makesense.ai	31
4.8 ตัวอย่างการระบุป้ายโดยใช้ เว็บไซต์ https://www.makesense.ai	32
4.9 ภาพตัวอย่างการประมวลผลการฝึกการตรวจหาดวงตาของ YOLOv5.....	32
4.10 ภาพตัวอย่างที่ใช้ทดสอบ	33
4.11 แสดงตัวอย่างการตรวจหาดวงตาที่ถูกต้อง.....	36
4.12 แสดงตัวอย่างการตรวจหาดวงตาที่ไม่ถูกต้อง.....	37
5.1 ตรวจหาดวงตา จมูก และ ปาก จากภาพถ่ายดิจิทัลที่ถ่ายจากบุคคลจริง.....	39

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหาการวิจัย

การวิเคราะห์โรคที่แสดงออกบนใบหน้าเป็นเทคโนโลยีทางการแพทย์ที่ใช้ปัญญาประดิษฐ์ (AI) ในการวิเคราะห์ลักษณะของใบหน้าเพื่อค้นหาโรคต่าง ๆ โดยพิจารณาความสัมพันธ์ระหว่างลักษณะใบหน้ากับการเปลี่ยนแปลงในร่างกายที่เกี่ยวข้องกับโรค เทคโนโลยีนี้ยังอยู่ในขั้นพัฒนาเริ่มต้น แต่มีงานวิจัยหลายชิ้นที่แสดงให้เห็นว่าสามารถตรวจหาโรคได้อย่างแม่นยำ เทคโนโลยีนี้ใช้การถ่ายภาพใบหน้าของผู้ป่วย แล้วให้ AI ประมวลผลเพื่อหาลักษณะผิดปกติ เช่น รอยย่น รอยคล้ำ การเปลี่ยนสีผิว และการเปลี่ยนแปลงโครงสร้างใบหน้า ลักษณะเหล่านี้อาจเป็นตัวบ่งชี้โรค เช่น รอยย่นรอบดวงตา อาจบ่งชี้ถึงโรคเบาหวาน รอยคล้ำใต้ตาอาจบ่งชี้ถึงโรคหัวใจ การเปลี่ยนสีผิวอาจบ่งชี้ถึงโรคมะเร็ง และการเปลี่ยนแปลงโครงสร้างใบหน้าอาจบ่งชี้ถึงโรคติดเชื้อ (Qiang et al., 2022; Sriwichai, Tanthanuch & Yimmuang, 2023) การพัฒนาเทคโนโลยีนี้จำเป็นต้องมีการออกแบบฐานข้อมูลที่มีประสิทธิภาพสำหรับการทำวิศวกรรมข้อมูล เพื่อสนับสนุนการทำวิทยาการข้อมูลที่มีประสิทธิภาพ ส่งผลให้ AI มีศักยภาพสูงในการวิเคราะห์โรค

เพื่อให้การออกแบบระบบวิศวกรรมข้อมูลสำหรับการทำงานกับข้อมูลขนาดใหญ่เป็นไปอย่างราบรื่น จำเป็นต้องใช้ระบบ AI ในการเก็บข้อมูลใบหน้า เนื่องจากข้อกำหนดด้านจริยธรรมในการวิจัยในมนุษย์ที่ให้ความสำคัญกับสิทธิ ศักดิ์ศรี ความปลอดภัย และความเป็นอยู่ที่ดีของผู้เข้าร่วมการวิจัย การวิจัยในมนุษย์ต้องใช้ระเบียบวิธีที่ถูกต้องตามหลักวิทยาศาสตร์และจริยธรรม (คณะกรรมการจริยธรรมการวิจัยในมนุษย์, ม.ป.ป.) ทำให้การรวบรวมข้อมูลใบหน้าต้องผ่านกระบวนการที่ซับซ้อน และมีความเสี่ยงต่อความผิดพลาดในการเก็บข้อมูล วิทยานิพนธ์นี้มีวัตถุประสงค์ในการออกแบบการใช้ AI เพื่อสร้างภาพจำลองแทนภาพถ่ายใบหน้าของบุคคลจริง เมื่อได้ข้อมูลจำลองเหล่านี้เป็นจำนวนมากจะช่วยให้การนำไปใช้ในการวิเคราะห์ใบหน้าเป็นไปได้อย่างสะดวกยิ่งขึ้น

วิทยานิพนธ์นี้ประกอบด้วยขั้นตอนการใช้ AI สร้างอวัยวะหลักบนใบหน้า ได้แก่ ตา จมูก และปาก จากนั้นฝึกฝน AI สำหรับการตรวจหาวัตถุ เมื่อฝึกฝนเสร็จจะวัดประสิทธิภาพของระบบตรวจหา

วัตถุที่สร้างขึ้น ระบบที่พัฒนาขึ้นนี้จะช่วยให้การวิเคราะห์ใบหน้าในอนาคตสามารถใช้ภาพถ่ายของคนจริงในการวิเคราะห์ได้อย่างรวดเร็วยิ่งขึ้น

1.2 วัตถุประสงค์ของการศึกษา

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อศึกษาการทำงานของ StyleGAN และ การสร้างแบบจำลองสำหรับปัญญาประดิษฐ์ ในการตรวจหาวัตถุบนใบหน้าและสามารถจำแนกได้ว่า สิ่งที่ตรวจจับได้นั้นคือ ดวงตา จมูก และ ปาก

1.3 ขอบเขตของงานวิจัย

- 1.3.1 StyleGAN เป็นโปรแกรมคอมพิวเตอร์หลักที่ใช้ในการศึกษาและทำวิจัยในครั้งนี้
- 1.3.2 การสร้างแบบจำลองสำหรับปัญญาประดิษฐ์ ใช้ StyleGAN ชนิด StyleGAN2-ADA-Pytorch เป็นฐานสำหรับการสร้างแบบจำลองต่าง ๆ ในวิทยานิพนธ์นี้
- 1.3.3 กำหนดเงื่อนไขค่าความถูกต้องของในการตรวจจับอวัยวะบนใบหน้าได้แก่ ดวงตา จมูก และ ปาก ด้วยปัญญาประดิษฐ์ ต้องมีค่ามากกว่าร้อยละ 80

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ผลลัพธ์ของแบบจำลองที่สร้างขึ้น สามารถนำไปตรวจจับอวัยวะบนใบหน้าได้ และคาดหวังในอนาคตว่าจะสามารถนำสิ่งเหล่านี้ไปประยุกต์ใช้กับงานอื่น ๆ ได้ เช่น ความผิดปกติของใบหน้า การจับการโกหก เป็นต้น

บทที่ 2

ปรัทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

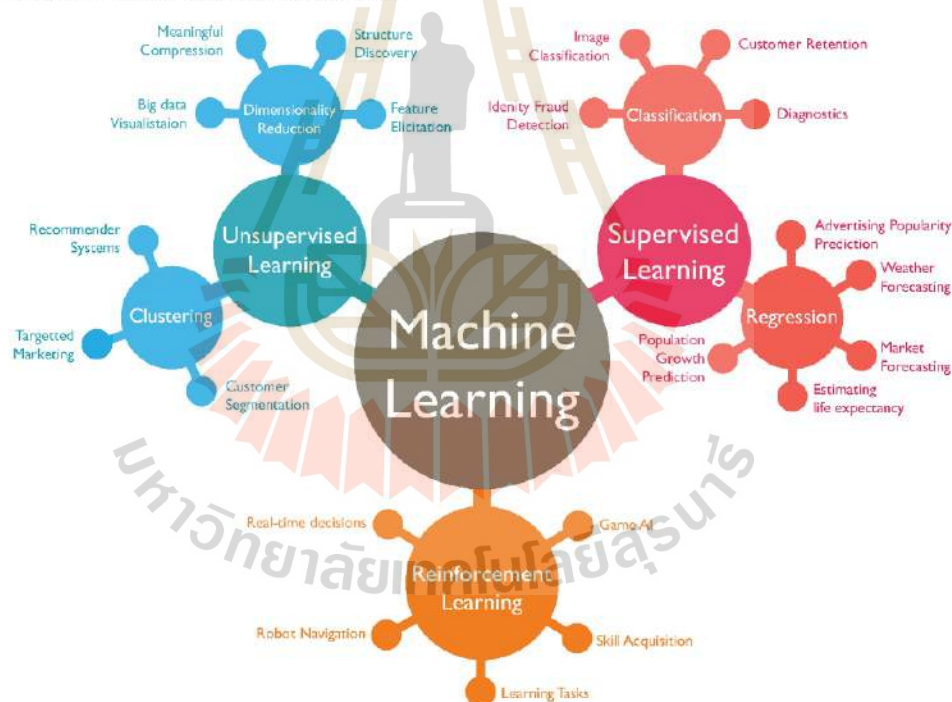
ในยุคดิจิทัลที่เทคโนโลยีเข้ามามีบทบาทสำคัญในหลายด้านของชีวิตประจำวัน ปัญญาประดิษฐ์ (AI) ได้กลายเป็นเครื่องมือที่สำคัญในการพัฒนานวัตกรรมและการแก้ปัญหาที่ซับซ้อน ด้วยความสามารถในการประมวลผลข้อมูลจำนวนมาก และการเรียนรู้จากข้อมูลเหล่านั้น ปัญญาประดิษฐ์ถูกนำมาใช้ในหลายด้าน ไม่ว่าจะเป็นการตรวจจับวัตถุ (Object Detection) และการตรวจโรค (Disease Detection) ซึ่งเป็นหัวข้อสำคัญในงานวิจัยและพัฒนาในปัจจุบัน

ปัญญาประดิษฐ์ หมายถึงความสามารถของระบบคอมพิวเตอร์ในการทำงาน ที่โดยปกติแล้วต้องอาศัยความฉลาดของมนุษย์ เช่น การเรียนรู้ การให้เหตุผล การแก้ปัญหา และการทำความเข้าใจ ภาษา การประยุกต์ใช้ ปัญญาประดิษฐ์ครอบคลุมถึงหลากหลายสาขา เช่น การแพทย์ การเกษตร การพัฒนาเมือง และอื่น ๆ ซึ่งทำให้ ปัญญาประดิษฐ์ เป็นหัวข้อที่ได้รับความสนใจจากนักวิจัยทั่วโลก การตรวจจับวัตถุเป็นหนึ่งในงานที่สำคัญที่สุดในด้านวิทยาการคอมพิวเตอร์และการประมวลผลภาพ (Computer Vision) โดยการตรวจจับวัตถุเป็นกระบวนการที่ทำให้ระบบสามารถระบุตำแหน่งของวัตถุต่าง ๆ ในภาพหรือวิดีโอได้ เทคนิคที่ใช้ในการตรวจจับวัตถุมีหลายประเภท เช่น การใช้เครือข่ายประสาทเทียมแบบลึก (Deep Neural Networks) และการเรียนรู้เชิงลึก (Deep Learning) ซึ่งสามารถเพิ่มความแม่นยำและความรวดเร็วในการตรวจจับโรค (Disease Detection) ในด้านการแพทย์ ปัญญาประดิษฐ์ได้มีบทบาทสำคัญในการตรวจโรคและวินิจฉัยโรคโดยอาศัยการประมวลผลข้อมูลทางการแพทย์ การวิเคราะห์ภาพทางการแพทย์ เช่น ภาพเอกซเรย์ ภาพ MRI และภาพจากกล้องจุลทรรศน์ การเรียนรู้เชิงลึกได้ถูกนำมาใช้เพื่อช่วยในการวินิจฉัยโรคอย่างแม่นยำมากขึ้น รวมถึงการตรวจหาโรคในระยะแรกเริ่ม ซึ่งมีผลต่อการรักษาที่มีประสิทธิภาพสูงขึ้น ตัวอย่างเช่น ปัญญาประดิษฐ์สามารถตรวจจับมะเร็งเต้านมจากภาพแมมโมแกรม หรือการตรวจหาโรคปอดจากภาพเอกซเรย์

2.1 ปัญญาประดิษฐ์ (Artificial Intelligence-AI)

ปัญญาประดิษฐ์คือระบบหรือเครือข่ายคอมพิวเตอร์ที่มนุษย์สร้างและพัฒนาขึ้น โดยมีวัตถุประสงค์เพื่อสร้างระบบที่สามารถคิด แยกแยะ และวิเคราะห์ข้อมูลได้อย่างมีประสิทธิภาพ ปัญญาประดิษฐ์จำลองวิธีการคิดของมนุษย์และใช้คอมพิวเตอร์ในการประมวลผลและตัดสินใจ การพัฒนาปัญญาประดิษฐ์ใช้ข้อมูลปริมาณมากจากแหล่งต่าง ๆ ในกระบวนการฝึกฝนและเรียนรู้ของระบบ เช่น การใช้ข้อมูลจากการสังเกตพฤติกรรมของมนุษย์หรือการรับรู้จากเซ็นเซอร์ต่าง ๆ เมื่อระบบได้รับข้อมูลเหล่านี้ จะนำมาเรียนรู้และสร้างความเข้าใจเกี่ยวกับข้อมูลนั้น

ปัญญาประดิษฐ์ถูกนำไปใช้ในหลายด้านของชีวิต เช่น การพัฒนารถยนต์ที่สามารถขับเคลื่อนได้เอง การปรับอารมณ์ของหุ่นยนต์ หรือการแก้ปัญหาทางการแพทย์ ความสามารถของปัญญาประดิษฐ์ในการเรียนรู้และปรับตัวได้เปิดโอกาสให้เกิดนวัตกรรมใหม่ ๆ และช่วยในการแก้ปัญหาที่ซับซ้อนในหลายสาขาของวิทยาศาสตร์และเทคโนโลยี



รูปที่ 2.1 ภาพแสดงประเภทของ Machine Learning

(ที่มา: Minaphinant, 2018)

2.2 การเรียนรู้ของเครื่อง (Machine Learning)

การเรียนรู้ของเครื่องเป็นส่วนหนึ่งของปัญญาประดิษฐ์ ที่มุ่งเน้นการพัฒนาขั้นตอนวิธีและโมเดลคอมพิวเตอร์ที่สามารถเรียนรู้ และการพยากรณ์หรือตัดสินใจได้โดยไม่ต้องมีการโปรแกรมอย่างชัดเจน หรืออีกนัยหนึ่งคือ เป็นการสอนคอมพิวเตอร์ให้สามารถเรียนรู้และพัฒนาตนเองจากประสบการณ์ที่ได้รับ

2.2.1 การเรียนรู้จากข้อมูล (Learning from Data)

อัลกอริทึม (Algorithm) การเรียนรู้ของเครื่องต้องการข้อมูลเพื่อค้นหารูปแบบความสัมพันธ์ และข้อมูลที่สำคัญ อย่างไรก็ตาม การเรียนรู้ของเครื่องสามารถฝึกด้วยชุดข้อมูลขนาดใหญ่ ทำให้สามารถสรุปและทำนายข้อมูลใหม่ที่ไม่เคยเห็นมาก่อนได้

2.2.2 ประเภทของการเรียนรู้

2.2.2.1 การเรียนรู้แบบมีผู้สอน (Supervised Learning) ในการเรียนรู้นี้ อัลกอริทึมจะทำการเรียนรู้จากข้อมูลที่มีป้ายกำกับ (Label) โดยข้อมูลนำเข้าจะถูกจับคู่กับผลลัพธ์หรือเป้าหมายที่ถูกต้อง เป้าหมายคือการเรียนรู้ จากข้อมูลนำเข้าไปสู่ผลลัพธ์ เพื่อให้อัลกอริทึมสามารถทำนายข้อมูลใหม่ที่ไม่เคยเห็นมาก่อน

2.2.2.2 การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) ในการเรียนรู้นี้ อัลกอริทึมจะทำการเรียนรู้จากข้อมูลที่ไม่มีป้ายกำกับ โดยมีวัตถุประสงค์เพื่อค้นพบรูปแบบหรือโครงสร้างที่ซ่อนอยู่ในข้อมูล การจัดกลุ่มและการลดมิติ เป็นงานที่มักพบในการเรียนรู้แบบไม่มีผู้สอน

2.2.2.3 การเรียนรู้แบบการกระทำ (Reinforcement Learning) ในการเรียนรู้นี้ เอเจนต์ (Agent) จะปฏิสัมพันธ์กับสภาพแวดล้อมและเรียนรู้ การตัดสินใจเพื่อให้ได้รางวัลสูงสุด มักถูกนำมาใช้ในหุ่นยนต์และการเล่นเกม

2.3 การประยุกต์ใช้งาน

การเรียนรู้ของเครื่อง (Machine Learning) ถูกนำมาใช้ในหลายด้านของชีวิตประจำวัน โดยมี การประยุกต์ใช้ที่หลากหลาย เช่น การระบุและตรวจจับภาพและเสียง การประมวลผลภาษาธรรมชาติ และการแปลภาษา ยานพาหนะอัตโนมัติและหุ่นยนต์ การวิเคราะห์ข้อมูลเพื่อการพยากรณ์ในด้าน การเงินและสุขภาพ ระบบแนะนำเนื้อหาที่ปรับให้เป็นส่วนบุคคล การตรวจสอบความปลอดภัยของ ข้อมูล และการวิเคราะห์ความคิดเห็นในสื่อสังคมออนไลน์ นอกจากนี้ การเรียนรู้ของเครื่องยังถูก นำมาใช้ในการวินิจฉัยทางการแพทย์ การตรวจสอบการทุจริตทางการเงิน การบริหารจัดการห่วงโซ่

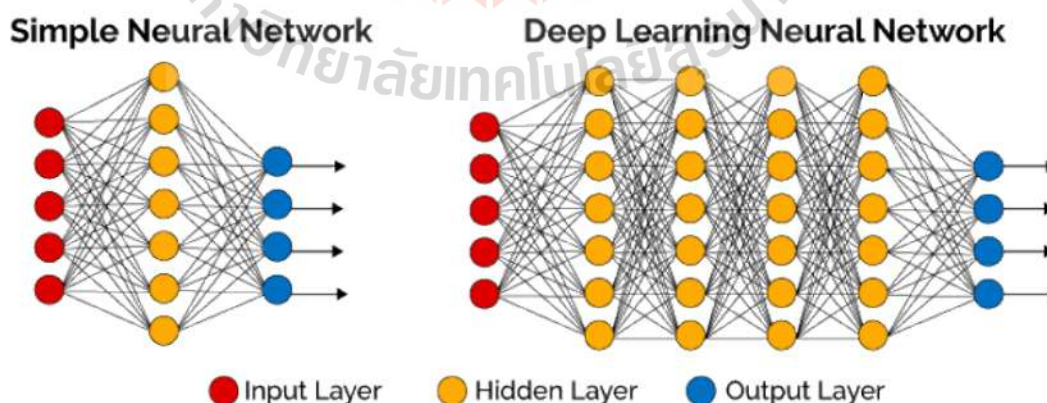
อุปทาน และการเพิ่มประสิทธิภาพในการผลิตอุตสาหกรรม ด้วยการใช้เทคโนโลยีนี้ ธุรกิจและองค์กรต่าง ๆ สามารถปรับปรุงกระบวนการทำงาน เพิ่มความแม่นยำในการตัดสินใจ และสร้างนวัตกรรมใหม่ๆ ที่ตอบสนองต่อความต้องการที่เปลี่ยนแปลงไปอย่างรวดเร็วของตลาดและผู้บริโภค

2.4 การเรียนรู้เชิงลึก (Deep Learning)

การเรียนรู้เชิงลึกเป็นสาขาหนึ่งของวิทยาศาสตร์ข้อมูลและปัญญาประดิษฐ์ที่มีความสำคัญมากในปัจจุบัน ทำงานโดยระบบคอมพิวเตอร์ที่มีความสามารถในการเรียนรู้ตามหลักการของการทำงานของระบบประสาทในสมองมนุษย์ การเรียนรู้เชิงลึกใช้โครงข่ายประสาทเทียมที่มีหลายชั้น เรียกว่าโครงข่ายประสาทเทียมระดับลึก (Deep Neural Networks) ซึ่งประกอบด้วยโหนดหรือเซลล์ประสาท (Neurons) หลายโหนดในแต่ละชั้น เปรียบได้กับเซลล์ประสาทในสมองมนุษย์

การเรียนรู้เชิงลึกจะฝึกโครงข่ายประสาทให้เรียนรู้และทำนายข้อมูลจากตัวอย่างที่มีอยู่ โดยปรับค่าพารามิเตอร์ภายในโครงข่ายเพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพ ข้อมูลตัวอย่างใช้ในการฝึกโครงข่ายให้รู้จักรูปแบบที่ซับซ้อน เช่น การจำแนกภาพถ่ายของสัตว์หรือการแบ่งหมวดหมู่ของข้อมูลต่าง ๆ

การเรียนรู้เชิงลึกมีการประยุกต์ใช้ในหลายด้านของชีวิตประจำวัน เช่น การจดจำใบหน้าในระบบรักษาความปลอดภัย การเข้าใจเสียงในระบบสมาร์ตโฮม หรือการพยากรณ์สภาพอากาศ ทั้งหมดนี้ทำให้การเรียนรู้เชิงลึกเป็นเครื่องมือที่สำคัญในการแก้ปัญหาและพัฒนาเทคโนโลยีให้มีประสิทธิภาพมากขึ้นในอนาคต และมีศักยภาพในการเปลี่ยนแปลงวิถีชีวิตของเราอย่างมีนัยสำคัญ



รูปที่ 2.2 ภาพเปรียบเทียบระหว่าง Neural Network แบบธรรมดา

กับ โครงข่ายประสาทการเรียนรู้เชิงลึก

(ที่มา: PasuJ, 2019)

2.4.1 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Networks)

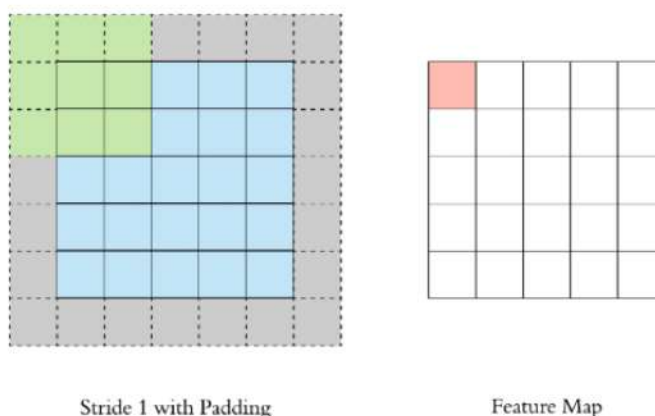
การศึกษาด้านการจำแนกข้อมูลภาพเป็นสิ่งสำคัญในวิทยาการนี้ โครงข่ายประสาทเทียมมีความสามารถในการจัดการและสกัดคุณลักษณะที่สำคัญจากข้อมูลภาพอย่างมีประสิทธิภาพ เหมือนการจำลองกระบวนการการมองเห็นของมนุษย์ที่สามารถมองรายละเอียดของพื้นที่เล็กๆ ๗ และรวมกลุ่มของพื้นที่เหล่านั้นเข้าด้วยกัน เพื่อสร้างความเข้าใจเกี่ยวกับวัตถุหรือรูปแบบที่มองเห็น

ขั้นตอนของ CNNs เริ่มต้นด้วยการกำหนดค่าในตัวกรองหรือเคอร์เนลซึ่งเป็นส่วนสำคัญในการสกัดคุณลักษณะโดยทั่วไปแล้วตัวกรองของภาพดิจิทัลจะเป็นตารางสองมิติที่มีขนาดเท่ากับพื้นที่ย่อยที่เราต้องการสกัดคุณลักษณะ เช่นถ้าต้องการหาเส้นตรงแนวทแยงสีขาว ตัวกรองอาจจะอยู่ในรูปแบบเป็นตารางขนาดเล็กที่นำมาสกัดคุณลักษณะเหล่านี้จากภาพ

1	-1	-1
-1	1	-1
-1	-1	1

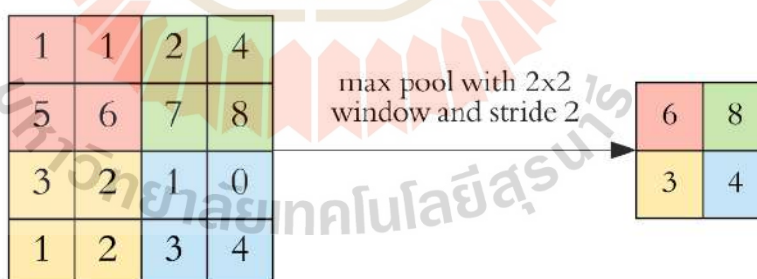
รูปที่ 2.3 ตัวอย่างตัวกรองเพื่อหาเส้นตรงแนวทแยงสีขาว

ในกระบวนการทำซ้ำบนพิกเซลของภาพข้อมูลเข้าใช้หลัก (Anchor) ที่ตั้งอยู่ตรงกลางและมีกรอบสี่เหลี่ยมเป็นจุดเริ่มต้น ขั้นแรกเริ่มต้นด้วยการทำซ้ำข้อมูลเข้าลงในตัวกรองที่ตั้งอยู่ในพิกเซลแรกของภาพข้อมูลเข้า ต่อมาตัวกรองจะเลื่อนไปหาพิกเซลอื่นในภาพทีละพิกเซลจนครบทุกพิกเซลในภาพ ขั้นตอนต่อไปคือการกำหนดขนาดของก้าวย่าง (Stride) ซึ่งบอกถึงการเลื่อนตัวกรองไปหาพิกเซลใดพิกเซลหนึ่ง หลังจากนั้นก็การทำ Padding โดยการเพิ่มกรอบล้อมรอบเมทริกซ์และเติมค่าเข้าไปค่าที่เติมเข้าไปมีค่าเท่ากับ 0 เพื่อให้แผนที่คุณลักษณะ (Feature Map) ที่ได้ยังคงมีขนาดเดิมเช่นเดิมของข้อมูลที่ป้อนเข้ามาใน CNNs โดยไม่เปลี่ยนแปลงขนาดลงหรือขนาดออกนอกเหนือจากที่ต้องการ



รูปที่ 2.4 การทำงานของ Pading

การทำแมกซ์พูลลิ่ง (Max Pooling) เป็นขั้นตอนสำคัญในการประมวลผลข้อมูลภาพของโมเดล CNNs เพื่อลดมิติของข้อมูลและเน้นคุณลักษณะที่สำคัญ การทำแมกซ์พูลลิ่งช่วยลดขนาดของแผนที่คุณลักษณะ (Feature Map) เพื่อให้การประมวลผลเร็วขึ้นและลดความซับซ้อนของโมเดล ขั้นตอนการทำแมกซ์พูลลิ่งคล้ายกับการสกัดคุณลักษณะของ CNNs โดยใช้ตัวกรองที่มีขนาดเหมือนกับการสกัดคุณลักษณะเหล่านั้น แต่เลือกค่าสูงสุดในบริเวณที่ตัวกรองทำการทาบบนข้อมูลภาพ การทำแมกซ์พูลลิ่งเป็นการเน้นคุณลักษณะที่สำคัญที่สุดในบริเวณนั้น ซึ่งช่วยลดการแปรผันของข้อมูลที่ไม่สำคัญและเพิ่มความเสถียรในการจำแนกหรือตรวจจับภาพ การเลือกค่าสูงสุดยังช่วยลดการสูญเสียข้อมูลที่เป็นไปได้ในกระบวนการทำแมกซ์พูลลิ่ง พูลไซส์ (Pool size) มีบทบาทในการกำหนดขนาดของบริเวณที่จะทำการสกัดคุณลักษณะได้ดีที่สุด



รูปที่ 2.5 การทำงานของ Max Pooling

การทำแมกซ์พูลลิ่งเป็นขั้นตอนสำคัญในการพัฒนาโมเดล CNNs ที่มีผลกระทบในการลดมิติข้อมูล การเน้นคุณลักษณะที่สำคัญ และเพิ่มความเร็วในการประมวลผลข้อมูลภาพ การเรียนรู้เพิ่มเติมเกี่ยวกับการใช้แมกซ์พูลลิ่ง และการปรับแต่งพารามิเตอร์สำหรับงานที่แตกต่างกัน สามารถพัฒนาโมเดล CNNs ในงานวิจัยในอนาคตได้อย่างมีประสิทธิภาพ (Natthawat, 2018)

ข้อดีของ CNNs เป็นการเรียนรู้คุณลักษณะขั้นสูงแบบอัตโนมัติจากข้อมูลดิจิทัล ซึ่งทำให้มีประสิทธิภาพสูงในการจำแนกภาพและวิดีโอที่ซับซ้อน สามารถเรียนรู้และจำแนกในส่วนต่าง ๆ ของภาพได้

การประยุกต์ใช้ของ CNNs ในการจำแนกภาพมีความสามารถในการจัดประเภทภาพเข้ากลุ่มตามคลาสที่กำหนดไว้ และนำไปใช้ในงานต่าง ๆ เช่น การระบุวัตถุในรูปภาพ การจัดประเภทโรคจากรูปภาพทางการแพทย์



2.5 ปัญญาประดิษฐ์สำหรับการสร้างภาพเสมือนจริง (Artificial Intelligence for Photorealistic Image Generation)

2.5.1 การประยุกต์ของปัญญาประดิษฐ์ในการสร้างภาพที่เหมือนจริง (photorealistic image generation)

หมายถึงการใช้เทคนิคปัญญาประดิษฐ์ขั้นสูงเพื่อสร้างภาพที่มีความสมจริงและรายละเอียดสูงที่เข้ากันได้กับรูปร่างกายจริง งานวิจัยนี้มีจุดประสงค์ คือ พัฒนาปัญญาประดิษฐ์ อย่างมีนัยสำคัญ ในปีหลังนี้โดยส่วนใหญ่ผ่านการพัฒนาระบบ Generative Adversarial Networks (GANs) และโมเดลปัญญาประดิษฐ์อื่น ๆ

2.5.2 Generative Adversarial Networks (GANs): GANs

เป็นกลุ่มของโมเดลปัญญาประดิษฐ์ที่ประกอบด้วยโครงข่ายประสาทเทียมสองตัว คือ ตัวสร้างภาพ (generator) และตัวคัดแยก (discriminator) ซึ่งทำงานในลักษณะการแข่งขันกัน ตัวสร้างจะพยายามสร้างภาพที่มีความเหมือนกับรูปภาพจริงจากข้อมูลที่จะอยู่ในรูปแบบของเวกเตอร์ที่มีค่าเป็นเลขสุ่ม โดยทั่วไปจะมีการสุ่มค่าในช่วงระหว่าง -1 ถึง 1 หรือ 0 ถึง 1 ในขณะที่ตัวคัดแยกจะพยายามแยกแยะระหว่างภาพจริงและภาพที่สร้างโดยตัวสร้าง ผ่านการฝึกฝนแบบกำหนดรอบ GANs กลายเป็นผู้ชำนาญมากขึ้นในการสร้างภาพที่เหมือนจริงจนถูกจำแนกได้ยากจากภาพจริง

2.5.3 Variational Autoencoders (VAEs): VAEs

เป็นโครงสร้างของระบบประสาทเทียมอีกประเภทหนึ่งที่ใช้ในการสร้างภาพ เรียนรู้ที่จะเข้ารหัสและถอดรหัสภาพในลักษณะที่กระตุ้นการสร้างภาพใหม่ที่เหมือนจริง VAEs มักถูกใช้ในการสร้างภาพที่ควบคุมและจัดการคุณลักษณะเฉพาะ เช่น การเปลี่ยนแปลงท่าทางใบหน้าหรือการแก้ไขวัตถุในภาพ

2.5.4 การโอนรูปแบบสไตล์และการเพิ่มความละเอียด

เทคนิคการโอนรูปแบบสไตล์ที่ใช้ปัญญาประดิษฐ์ ช่วยให้ภาพเปลี่ยนแปลงให้คล้ายกับสไตล์ศิลปิน หรือ ภาพที่ถูกถ่ายขึ้นโดยช่างภาพ การเพิ่มความละเอียดจะเพิ่มระดับรายละเอียดในภาพ ทำให้ภาพดูเหมือนจริงมากขึ้น

2.5.5 Conditional GANs

โมเดลรูปแบบเหล่านี้ให้ความสำคัญกับเงื่อนไขหรือข้อมูลที่ระบุได้ โดยใช้ข้อมูลเฉพาะเจาะจง เช่น ข้อกำหนดหรือรูปร่าง เมื่อสร้างภาพ ตัวสร้างภาพสามารถใช้ข้อมูลนี้เพื่อสร้างภาพที่สอดคล้องกับเงื่อนไขที่กำหนดไว้ ตัวอย่างเช่น สามารถสร้างภาพจากคำอธิบายจากข้อความได้ ซึ่งมีประโยชน์ในการใช้งานด้านกราฟิกคอมพิวเตอร์ การออกแบบ และอื่น ๆ

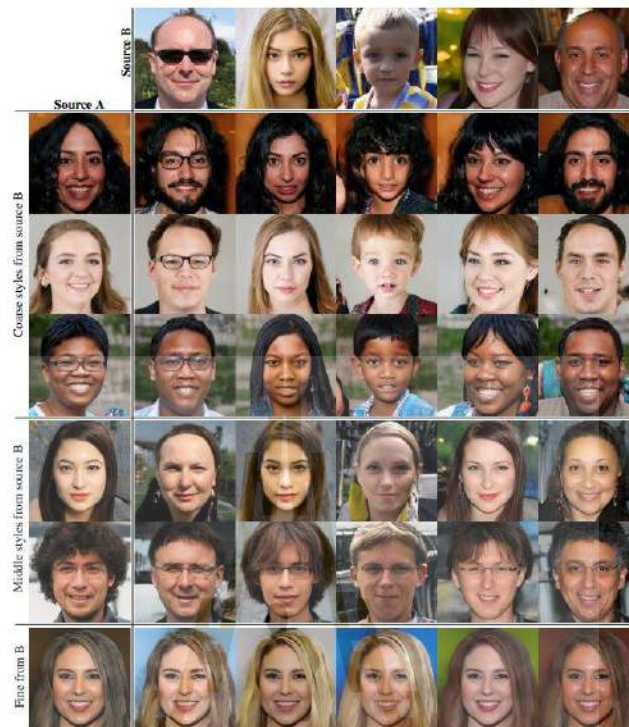
2.5.6 การประยุกต์ใช้

ปัญญาประดิษฐ์สำหรับการสร้างภาพที่เหมือนจริงมีการประยุกต์ใช้ในหลายสาขา เช่น บันเทิงและเกม การสร้างตัวละครที่เหมือนจริง สภาพแวดล้อม และฉากพิเศษในวิดีโอเกมและภาพยนตร์ ออกแบบและแฟชั่น การสร้างภาพผลิตภัณฑ์ที่คล้ายกับภาพของอุตสาหกรรมออกแบบและแฟชั่น สถาปัตยกรรม การสร้างภาพที่เหมือนจริงของแบบแผนออกแบบสถาปัตยกรรมและพื้นที่ภายใน ศิลปะและความคิดสร้างสรรค์ การช่วยศิลปินในการสร้างศิลปะดิจิทัลและช่วยในการสร้างฉากทางภาพในอุตสาหกรรมภาพยนตร์ ความจริงเสมือน (VR) และ ความเป็นจริงเสริม (AR) การเพิ่มความเรียบเรียงในสิ่งแวดล้อมเสมือน การตรวจวัดทางการแพทย์ การสร้างภาพที่มีความจริงเสมือนเหมือนในรูปแบบ 3 มิติ ของระบบร่างกายเพื่อการฝึกอบรมและการวินิจฉัย

ปัญญาประดิษฐ์สำหรับการสร้างภาพที่เหมือนจริงมากใช้ โมเดลการเรียนรู้เชิงลึกอย่าง GANs และ VAEs เพื่อสร้างภาพที่มีรายละเอียดสูงและสมจริง โดยมักมีการประยุกต์ใช้ในงานบันเทิง ออกแบบ สถาปัตยกรรม และอุตสาหกรรมอื่น ๆ นี่เป็นสาขาที่กำลังพัฒนาอย่างรวดเร็วและมีศักยภาพในการเปลี่ยนแปลงหลายสาขา

2.6 Generative Adversarial Networks (GANs)

GANs เป็นขั้นตอนวิธีที่เป็นที่นิยมที่สุดในปัจจุบันในการสร้างภาพเหมือนจริงด้วยปัญญาประดิษฐ์ GANs ซึ่งเป็น deep learning ชนิดหนึ่งที่มีความสามารถในการสร้างภาพหน้าคน ในช่วงเริ่มต้นของการพัฒนา GANs สามารถสร้างภาพใบหน้าได้แต่ยังไม่มีรายละเอียดที่ชัดเจนนักและยังเป็นเพียงแค่ภาพขาวดำ หลังจากนั้นในปี 2018 บริษัท Nvidia ได้ทำการพัฒนา GANs เพื่อสร้างหน้าคนขึ้นมาโดยใช้วิธีการที่เรียกว่า Style Transfer หรือต่อมาจะถูกเรียกรวมเป็นชื่อ StyleGAN และยังมีโปรแกรมให้ใช้อีกมากมาย เช่น StyleGAN2 StyleGAN2-ADA และ StyleGAN2-ADA-Pytorch โดยวิธีการของ StyleGAN คือการผสมลักษณะเฉพาะของภาพคนแรกเข้าไปร่วมกับบุคลิกที่ได้จากภาพของคนที่สอง



รูปที่ 2.6 ภาพการทำงานของ StyleGAN
(ที่มา: techguy, 2020)

ขั้นตอนการดำเนินการ พิจารณาจาก 3 องค์ประกอบต่อไปนี้

1. Coarse Style เป็นภาพโดยรวมแบบหยาบ ๆ เช่น โครงหน้า ท่าทางของใบหน้า
2. Middle styles เป็นการเก็บรายละเอียดรูปลักษณะของใบหน้าและตา
3. Fine Styles คือ โทนสีของภาพ



รูปที่ 2.7 ภาพขั้นตอนการผสมภาพของ StyleGAN
(ที่มา: techguy, 2020)

2.7 ภาษาไพธอน (Python) และคลังโปรแกรม (Libraries)

2.7.1 ภาษาไพธอน (Python) เป็นภาษาโปรแกรมมิ่งระดับสูงที่ถูกสร้างขึ้นในช่วงสุดท้ายของยุค 1980 โดย Guido van Rossum และเป็นหนึ่งในภาษาโปรแกรมที่ได้รับความนิยมสูงสุดทั่วโลก เนื่องจากความเรียบง่ายในการใช้งานและการอ่านโค้ดที่ง่ายต่อผู้ใช้งาน ภาษาไพธอนถูกออกแบบให้ไม่ซับซ้อนด้วยโครงสร้างและไวยากรณ์ที่เข้าใจง่าย การทำงานของโปรแกรมภาษาไพธอนเป็นแบบอินเทอร์พรีเตอร์ (Interpreter) ซึ่งหมายความว่าสามารถแปลและประมวลผลชุดคำสั่งได้ที่ละบรรทัด โดยทำให้คอมพิวเตอร์ทำงานตามที่ต้องการได้โดยตรง นอกจากนี้โปรแกรมภาษาไพธอนยังมีความสามารถในการนำไปใช้ในการเขียนโปรแกรมหลากหลายประเภทโดยไม่มีข้อจำกัดเฉพาะในงานที่เฉพาะเจาะจงเท่านั้น ซึ่งทำให้มีการนำไปใช้ในหลายหน่วยงานและองค์กรต่าง ๆ ทั่วโลก (Guido, 2009)

2.7.2 PyTorch เป็นโอเพนซอร์สที่ใช้สำหรับการพัฒนาและการฝึกฝนระบบประสาทเทียม ถูกพัฒนาโดย Facebook's AI Research lab (FAIR) และเป็นเครื่องมือที่นิยมใช้ในงานด้านการเรียนรู้เชิงลึก คุณสมบัติที่โดดเด่นของ PyTorch คือความสามารถในการกำหนดให้หน่วยประมวลผลกราฟิก (GPU - Graphics Processing Unit) มีส่วนรับผิดชอบในการคำนวณต่าง ๆ แทนหน่วยประมวลผลกลาง (CPU - Central Processing Unit) ซึ่งช่วยเพิ่มประสิทธิภาพและความเร็วในการคำนวณอย่างมาก

2.7.3 Tensorflow เป็นอีกคลังโปรแกรมหนึ่งของโปรแกรมภาษาไพธอน ซึ่งมีชุดคำสั่งที่เกี่ยวกับการเรียนรู้ของเครื่องเช่นกัน Tensorflow ถูกพัฒนาขึ้นโดยบริษัท Google คลังโปรแกรม Tensorflow นั้นอยู่ในรูปแบบโปรแกรมโอเพนซอร์ส (Open Source Software) ซึ่งเปิดโอกาสให้ผู้ใช้สามารถแก้ไขโปรแกรมได้ด้วยตนเอง ซึ่งช่วยให้ผู้ใช้สามารถพัฒนาต่อยอดเพื่อช่วยในการเขียนโปรแกรมด้านการเรียนรู้ของเครื่องได้สะดวกเป็นอย่างมาก

2.8 การตรวจหาวัตถุ (Object Detection)

การตรวจหาวัตถุเป็นเทคโนโลยีที่มีหลักการเกี่ยวกับวิสัยทัศน์คอมพิวเตอร์ (Computer Vision) และการประมวลผลภาพ (Image Processing) สามารถนำไปประยุกต์ใช้งานได้หลากหลายรูปแบบ เช่น การตรวจจับใบหน้า ระบบรักษาความปลอดภัยในอาคาร รถยนต์ไร้คนขับ และการจำแนกภาพหรือวิดีโอดิจิทัล เป็นต้น โดยการตรวจหาวัตถุนี้มีบทบาทสำคัญในการพัฒนาเทคโนโลยีและสร้างความปลอดภัย ตลอดจนความสะดวกสบายในชีวิตประจำวัน

2.8.1 ขั้นตอนวิธีเบื้องต้นการตรวจจับวัตถุ

2.8.1.1 แบ่งภาพหรือวิดีโอที่ค้นคืนที่ออกมาเป็นส่วนย่อย ๆ

2.8.1.2 ใช้เทคนิคการจดจำภาพเพื่อระบุวัตถุในแต่ละส่วนย่อย

2.8.1.3 รวบรวมผลลัพธ์จากแต่ละส่วนย่อยเพื่อระบุวัตถุทั้งหมดในภาพหรือวิดีโอที่ค้นคืน

2.8.2 R-CNN

ความสำคัญของงานวิจัยที่เขียนโดย Girshick et al. คือการนำเสนอโครงข่ายเพื่อสร้างพื้นที่เสนอ (Region Proposal Network) ซึ่งเป็นการลงมือสร้างทฤษฎีและเทคโนโลยี โดยโครงข่ายของ R-CNN ไม่เพียงแต่ประกอบด้วยองค์ประกอบหลักต่าง ๆ แต่ยังมีผลในการเปลี่ยนแปลงวิธีการใช้งานในสาขาคอมพิวเตอร์วิชั่นอย่างแท้จริง ด้วยความสามารถในการสร้างพื้นที่แบบอัตโนมัติที่มีประสิทธิภาพ ทำให้เห็นมุมมองใหม่ในการแก้ปัญหาทางด้านการรับรู้คอมพิวเตอร์และการประมวลผลภาพ นอกจากนี้ งานวิจัยนี้ยังเปิดโอกาสให้ผู้วิจัยอื่นสามารถพัฒนาและปรับปรุงโครงข่ายเหล่านี้ให้มีประสิทธิภาพมากยิ่งขึ้น และใช้งานได้กับหลากหลายสถานการณ์ที่แตกต่างกัน

โดยรวมแล้ว R-CNN จะประกอบด้วยกัน 4 ขั้นตอนดังนี้

2.8.2.1 การเสนอพื้นที่ในภาพที่อาจจะมีวัตถุที่สนใจด้วย Selective Search

2.8.2.2 การฝึกและปรับแต่งอย่างละเอียดบนตัวแบบ CNN

2.8.2.3 การฝึกตัวจำแนกประเภทด้วย SVM แบบแยกทีละคลาส

2.8.2.4 การฝึกการตีกรอบวัตถุให้แม่นยำ

2.8.3 Region-based convolutional neural networks (R-CNNs)

ในปี 2014 Ross Girshick และเครือข่ายของเขาได้พัฒนาขั้นตอนวิธีนี้ขึ้น โดยมีขั้นตอนการแบ่งภาพออกเป็นส่วนย่อยก่อน จากนั้นจึงใช้ CNNs เพื่อวิเคราะห์ภาพและระบุวัตถุในแต่ละส่วนย่อย (Girshick et al., 2014) การนำเสนอนี้เป็นที่เชื่อถือและมีผลสำคัญในการเปิดทางให้กับการพัฒนาวิจัยในด้านการตรวจจับวัตถุและการประมวลผลภาพในปีต่อไป

2.8.4 Faster R-CNNs

ในปี 2015 Ross Girshick และเครือข่ายของเขาได้พัฒนาขั้นตอนวิธีการตรวจจับวัตถุโดยการปรับปรุง R-CNNs โดยใช้เทคนิคที่ชื่อว่า region proposal network (RPN) เพื่อระบุส่วนย่อยที่มีแนวโน้มที่จะมีส่วนประกอบของวัตถุ หลังจากนั้นก็ยังคงใช้ CNNs เพื่อการวิเคราะห์ภาพและระบุวัตถุในแต่ละส่วนย่อยเช่นเดิม (Girshick, He, & Dollar, 2015) การปรับปรุงนี้เสริมสร้างเสถียรภาพของวิธีการตรวจจับวัตถุ

2.8.5 Yolo (You Only Look Once)

ขั้นตอนวิธีนี้ เป็นเทคนิคการตรวจจับวัตถุในภาพ (Modern Convolutional Detection) ซึ่งถูกพัฒนาด้วยโปรแกรมภาษา C++ และสามารถทำงานบนหน่วยประมวลผล CUDA ของหน่วยประมวลผลภาพกราฟิก (Graphics Processing Unit: GPU) เพื่อให้สามารถทำงานได้แบบทันที (Real Time) ถูกพัฒนาโดย Joseph Redmon และ Farhadi ซึ่งเป็นขั้นตอนวิธีการตรวจจับวัตถุที่ได้รับความนิยมมากและมีการพัฒนาอย่างต่อเนื่อง โดย YOLO เป็นระบบตรวจจับวัตถุที่เป็นที่นิยมมากในปัจจุบัน โดยรุ่นที่ได้รับความนิยมมากที่สุดในปัจจุบันคือ YOLOv5

2.9 YoloV5 (You Only Look Once)

YOLOv5 พัฒนาขึ้นในปี 2020 โดยต่อยอดจาก YOLO ต้นฉบับ โดย YOLOv5 ที่ถูกพัฒนาขึ้นมาใหม่มีประสิทธิภาพที่ดีกว่าเวอร์ชันก่อนหน้า ตัวอย่างการใช้ YOLOv5 ได้แก่การติดตั้งในรถยนต์ไร้คนขับของ Tesla และร้านค้าปลอดแคชเชียร์ของ Amazon ซึ่งเป็นการนำมาใช้ที่ได้รับความนิยมในการตรวจจับวัตถุจากภาพถ่ายและภาพวิดีโอที่บันทึกดิจิทัล

กระบวนการทำงานของ YOLOv5 ประกอบด้วยขั้นตอนต่าง ๆ ดังนี้

2.9.1 การแบ่งภาพออกเป็นส่วนย่อย (Image Subdivision)

ในขั้นตอนนี้ YOLOv5 จะแบ่งภาพออกเป็นส่วนเล็ก ๆ ที่เรียกว่า grid cells แต่ละ grid cell จะเป็นพื้นที่ที่ YOLOv5 จะตรวจจับวัตถุในภาพ เพื่อให้สามารถพิจารณาพื้นที่ของภาพอย่างละเอียดและมีประสิทธิภาพในการตรวจจับวัตถุ

2.9.2 การสกัดคุณลักษณะของวัตถุ (Feature Extraction)

ใช้ Convolutional Neural Network (CNN) เพื่อสกัดคุณลักษณะของวัตถุจากแต่ละ grid cell ที่ได้จากขั้นตอนที่ 1 โดย CNN ทำหน้าที่สกัดคุณลักษณะที่สำคัญของวัตถุ เช่น ลักษณะทางเชิงรูปร่าง และลักษณะเฉพาะของวัตถุ

2.9.3 การคาดเดาประเภทของวัตถุ (Object Category Prediction)

ในขั้นตอนนี้ YOLOv5 ใช้ชั้นเชื่อมโยงแบบสมบูรณ์ (fully-connected layer) ใน CNN เพื่อสร้างเวกเตอร์ที่บ่งชี้ประเภทของวัตถุที่ตรวจจับได้ โดยจำนวนมิติของเวกเตอร์นี้จะเท่ากับจำนวนประเภทของวัตถุที่ต้องการทำนาย เช่น รถยนต์ มนุษย์ และสัตว์ เป็นต้น

2.9.4 การคาดเดาตำแหน่งของวัตถุ (Object Localization)

ใช้ fully-connected layer เพื่อทำนายตำแหน่งของวัตถุในภาพ ซึ่งจะระบุตำแหน่งของวัตถุใน grid cell ที่สมบูรณ์ เช่น พิกเซลที่ตำแหน่งซ้ายบนของวัตถุ และพิกเซลที่ตำแหน่งขวาล่างของวัตถุ

2.9.5 ขั้นตอนการปรับปรุงผลลัพธ์ หลังการคาดเดาตำแหน่งของวัตถุ ขั้นตอนนี้เกี่ยวกับการปรับปรุงผลลัพธ์ เช่น ลบข้อมูลไม่จำเป็นที่มีความน่าจะเป็นต่ำ ลบข้อมูลรบกวน เช่น ตัวอักษร และรวมกลุ่มวัตถุที่ใกล้เคียงกันเข้าเป็นวัตถุเดียว

2.9.6 ขั้นตอนการค้นหาวัตถุในภาพ ในขั้นตอนสุดท้ายนี้ระบบจะค้นหาวัตถุและสร้างกรอบสี่เหลี่ยมรอบวัตถุที่ตรวจพบ รวมถึงแสดงข้อมูลเพิ่มเติมเกี่ยวกับวัตถุ เช่น ชื่อและคะแนนความเชื่อมั่นว่าเป็นวัตถุนั้น

2.10 เครื่องมือในการวิเคราะห์ประสิทธิภาพ

ในการวิเคราะห์ประสิทธิภาพของระบบปัญญาประดิษฐ์ที่พัฒนาขึ้นมักใช้เครื่องมือทางสถิติที่เรียกว่า เมทริกซ์ความสับสน (Confusion Matrix) เพื่อประเมินความแม่นยำในการทำนายผลของระบบ โดยเป็นตารางขนาด 2x2 ที่ประกอบด้วยหมวดหมู่ที่ระบบทำนาย (Predicted) และหมวดหมู่จริง (Actual)

เมทริกซ์ความสับสน ช่วยวิเคราะห์ความสามารถของระบบในการจำแนกหมวดหมู่ต่าง ๆ ได้อย่างชัดเจน โดยสามารถคำนวณอัตราส่วนความถูกต้อง (Accuracy) ความแม่นยำของการทำนาย (Precision) และค่าเรียกคืน (Recall) ได้จากข้อมูลในเมทริกซ์ความสับสนนี้

ตารางที่ 2.1 เมทริกซ์ความสับสน (Confusion Matrix)

		Actual Values		
			Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP	
	Negative (0)	FN	TN	

โดยค่าต่าง ๆ ที่ปรากฏในตารางมีความหมายดังนี้

- 1 True Positive (TP) คือ จำนวนของการทำนายที่ระบบ AI ทำนายว่าเป็น จริง (Positive) และผลการทดสอบยืนยันว่าเป็น จริง (Positive)
- 2 True Negative (TN) คือ จำนวนของการทำนายที่ระบบ AI ทำนายว่าเป็น ไม่จริง (Negative) และผลการทดสอบยืนยันว่าเป็น ไม่จริง (Negative)
- 3 False Positive (FP) คือ จำนวนของการทำนายที่ระบบ AI ทำนายว่าเป็น จริง (Positive) แต่ผลการทดสอบแสดงให้เป็น ไม่จริง (Negative)
- 4 False Negative (FN) คือ จำนวนของการทำนายที่ระบบ AI ทำนายว่าเป็น ไม่จริง (Negative) แต่ผลการทดสอบแสดงให้เป็น จริง (Positive)

จากข้อมูลข้างต้นเราสามารถนำมาประมวลผลได้ค่าต่าง ๆ โดยผลการประมวลที่เป็นที่นิยมใช้ในการวิเคราะห์ประสิทธิภาพของระบบปัญญาประดิษฐ์ มีดังนี้

ความถูกต้อง (Accuracy) เป็นค่าที่ใช้วัดประสิทธิภาพของระบบประมวลผลข้อมูลหรือปัญญาประดิษฐ์ในการทำนายผลลัพธ์ตามข้อมูลจริง โดยแสดงให้เห็นว่าระบบสามารถทำนายได้ถูกต้องในสัดส่วนเท่าใดจากทั้งหมด คำนวณได้โดย

$$\frac{TP+TN}{TP+TN+FP+FN}$$

ความแม่นยำ (Precision) เป็นค่าที่ใช้วัดความสามารถของระบบประมวลผลข้อมูลหรือปัญญาประดิษฐ์ในการทำนายสิ่งที่สนใจว่าจริงหรือไม่ คำนวณโดยการหารจำนวนของข้อมูลที่ทำนายถูกต้องในกลุ่มที่สนใจด้วยจำนวนของข้อมูลทั้งหมดที่ทำนายว่าเป็นกลุ่มนั้น คำนวณโดย

$$\frac{TP}{TP+FP}$$

ค่าเรียกคืน (Recall) หรือบางครั้งอาจเรียกว่าค่าความไว (Sensitivity) เป็นค่าที่ใช้วัดความสามารถของระบบประมวลผลข้อมูลหรือปัญญาประดิษฐ์ในการตรวจจับสิ่งที่สนใจว่าจริง ๆ มีอยู่แค่ไหน ในกรณีที่สิ่งที่สนใจเป็นบวก (positive) เช่นการตรวจจับโรคหรือธุรกรรมที่ผิดปกติ คำนวณโดยการหารจำนวนของข้อมูลที่ทำนายถูกต้องในกลุ่มที่สนใจด้วยจำนวนของข้อมูลที่เป็นจริงในกลุ่มนั้น คำนวณโดย

$$\frac{TP}{TP+FN}$$

ค่าทั้งสามที่กล่าวมาข้างต้นมีค่าเป็นจำนวนจริงระหว่าง 0 ถึง 1 โดยหากค่าเข้าใกล้ 0 แสดงว่าระบบประมวลผลข้อมูลหรือปัญญาประดิษฐ์มีประสิทธิภาพต่ำ และหากค่าเข้าใกล้ 1 แสดงว่าระบบประมวลผลข้อมูลหรือปัญญาประดิษฐ์มีประสิทธิภาพสูง

2.11 แนวคิดและงานวิจัยที่เกี่ยวข้อง

ปัจจุบันมีการใช้ระบบปัญญาประดิษฐ์ในงานต่าง ๆ มากมาย รวมถึงใช้ในการสร้างภาพเสมือนจริง ตัวอย่างเช่น โปรแกรม DALL-E 2 ที่ถูกพัฒนาโดย OpenAI สามารถเปลี่ยนข้อความให้เป็นภาพได้อย่างสะดวกรวดเร็ว นอกจากนี้ยังมี Imagen จาก Google และ Adobe Photoshop (Firefly) ที่สามารถแปลงข้อความให้เป็นภาพได้เช่นกัน รวมทั้ง Midjourney ที่สามารถสร้างภาพเสมือนจริงได้ใกล้เคียงกับภาพถ่ายมาก โดยสามารถให้รายละเอียดของแสง สี เงา ลายผิวได้อย่างเสมือนจริงมาก เหล่านี้เป็นตัวอย่างของโปรแกรมที่ใช้เทคโนโลยีปัญญาประดิษฐ์ในการสร้างภาพเสมือนจริง

การทำงานของ การใช้ AI ในการสร้างภาพเสมือนจริงมักใช้หลักการของการเรียนรู้เชิงลึก เช่น Convolutional Neural Networks (CNNs) ซึ่งเป็นระบบโครงข่ายประสาทเทียมที่ใช้สำหรับการวิเคราะห์ภาพ นอกจากนี้ยังมี Variational Autoencoders (VAEs) และ Generative Adversarial Networks (GANs) ที่เป็นขั้นตอนวิธีที่นิยมใช้ในการสร้างภาพเสมือนจริงด้วย AI โดยแต่ละขั้นตอนมีหลักการงานเฉพาะ เช่น CNNs เรียนรู้คุณสมบัติของภาพจากข้อมูล ส่วน VAEs สามารถถอดรหัสอัตโนมัติแบบแปรผันได้ เมื่อได้รับข้อมูล AI จะสามารถสร้างภาพใหม่โดยใช้คุณสมบัติที่เรียนรู้ได้ ส่วน GANs ประกอบด้วยตัวแบบ Generator และ Discriminator ซึ่งทำหน้าที่สร้างภาพใหม่และแยกแยะภาพจริงจากภาพปลอม การใช้ GANs เป็นที่นิยมมากที่สุดในปัจจุบันในการสร้างภาพเสมือนจริงด้วย AI

งานวิจัยเกี่ยวกับการตรวจจับวัตถุ (Object Detection) ได้รับความสนใจอย่างมากในช่วงหลายปีที่ผ่านมา ด้วยการประยุกต์ใช้เทคโนโลยี AI และ Machine Learning เพื่อพัฒนาความแม่นยำและประสิทธิภาพในการตรวจจับวัตถุต่าง ๆ

งานวิจัยของ Redmon et al. (2016) นำเสนอ YOLO (You Only Look Once) ซึ่งเป็นอัลกอริทึมที่สามารถตรวจจับวัตถุแบบเรียลไทม์ได้อย่างรวดเร็ว โดยใช้การเรียนรู้เชิงลึกแบบ Convolutional Neural Network (CNN) อัลกอริทึมนี้สามารถประมวลผลภาพได้ถึง 45 เฟรมต่อวินาที และมีความแม่นยำในการตรวจจับวัตถุสูงเมื่อเทียบกับอัลกอริทึมอื่น ๆ ที่มีอยู่ในขณะนั้น

งานวิจัยของ Liu et al. (2016) เสนอ SSD (Single Shot MultiBox Detector) ซึ่งเป็นอีกหนึ่งเทคนิคในการตรวจจับวัตถุที่ใช้ CNN อัลกอริทึมนี้สามารถตรวจจับวัตถุในภาพโดยไม่ต้องใช้การ

แบ่งภาพออกเป็นกริดเหมือน YOLO ทำให้มีความแม่นยำสูงขึ้นในการตรวจจับวัตถุขนาดเล็กและวัตถุที่มีลักษณะหลากหลาย

งานวิจัยของ Ren et al. (2015) พัฒนา Faster R-CNN (Region-based Convolutional Neural Network) ที่รวมเอาขั้นตอนการสร้าง Region Proposal เข้าไปในเครือข่าย CNN ทำให้การตรวจจับวัตถุเร็วขึ้นและมีประสิทธิภาพสูงกว่า R-CNN และ Fast R-CNN ในการตรวจจับวัตถุในภาพนิ่งและวิดีโอ

งานวิจัยของ He et al. (2017) นำเสนอ Mask R-CNN ซึ่งเป็นการขยายตัวของ Faster R-CNN โดยเพิ่มความสามารถในการทำ Segmentation หรือการแยกส่วนภาพ อัลกอริทึมนี้ไม่เพียงแต่สามารถตรวจจับวัตถุได้เท่านั้น แต่ยังสามารถแยกส่วนวัตถุในภาพออกมาได้อย่างละเอียดและแม่นยำ

งานวิจัยของ Lin et al. (2017) เสนอ Focal Loss สำหรับ Dense Object Detection ซึ่งพัฒนา RetinaNet อัลกอริทึมนี้ใช้ Focal Loss เพื่อแก้ปัญหาความไม่สมดุลระหว่างวัตถุที่มีอยู่มากและน้อยในชุดข้อมูล ทำให้มีความแม่นยำสูงขึ้นในการตรวจจับวัตถุที่มีความหนาแน่นสูง

งานวิจัยของ Alexey Bochkovskiy et al. (2020) นำเสนอ YOLOv4 ซึ่งเป็นการปรับปรุงจาก YOLO รุ่นก่อนหน้านี้ ด้วยการใช้เทคนิคหลายอย่างเช่น CSPDarknet53 Mish activation และ PANet เพื่อเพิ่มประสิทธิภาพในการตรวจจับวัตถุและทำให้การเรียนรู้เชิงลึกสามารถทำงานได้ดีขึ้นในสถานการณ์ต่าง ๆ

งานวิจัยของ Dai et al. (2016) เสนอ R-FCN (Region-based Fully Convolutional Networks) ที่ออกแบบมาเพื่อเพิ่มประสิทธิภาพในการตรวจจับวัตถุโดยใช้โครงข่ายประสาทแบบ Fully Convolutional ซึ่งช่วยลดเวลาในการประมวลผลและเพิ่มความแม่นยำในการตรวจจับวัตถุในภาพนิ่งและวิดีโอ

งานวิจัยของ Ma et al. (2018) นำเสนอการประยุกต์ใช้เทคโนโลยี AI ในการตรวจจับวัตถุในภาพทางการแพทย์ โดยใช้อัลกอริทึม Mask R-CNN ที่พัฒนาจาก Faster R-CNN ซึ่งนอกจากจะตรวจจับวัตถุแล้ว ยังสามารถระบุขอบเขตของวัตถุที่ตรวจพบได้อย่างแม่นยำ อัลกอริทึมนี้ถูกนำไปใช้ในการตรวจจับเนื้องอกในภาพถ่าย MRI และ CT scans โดยมีความแม่นยำสูงและสามารถระบุขอบเขตของเนื้องอกได้อย่างชัดเจน ทำให้แพทย์สามารถวินิจฉัยและวางแผนการรักษาได้เร็วขึ้น

งานวิจัยของ Jiang et al. (2019) พัฒนารูปแบบการตรวจจับและจำแนกประเภทของโรคในภาพถ่ายรังสีทรวงอกด้วยการใช้อัลกอริทึม YOLOv3 ซึ่งเป็นการพัฒนาต่อจาก YOLO โดยมีการเพิ่มประสิทธิภาพในการตรวจจับวัตถุขนาดเล็กและการจัดการภาพที่มีความละเอียดสูง งานวิจัยนี้แสดงให้เห็น

เห็นว่า YOLOv3 สามารถตรวจจับและจำแนกประเภทของโรค เช่น ปอดบวมและมะเร็งปอด ได้อย่างแม่นยำและรวดเร็ว ช่วยเพิ่มประสิทธิภาพในการวินิจฉัยและลดเวลาในการตรวจสอบภาพรังสี

งานวิจัยของ Li et al. (2020) เสนอการใช้เทคโนโลยี AI ในการตรวจจับและจำแนกเซลล์มะเร็งเรียงในภาพถ่ายจากกล้องจุลทรรศน์ โดยใช้อัลกอริทึม Faster R-CNN ที่มีการปรับปรุงให้เหมาะสมกับภาพทางการแพทย์ อัลกอริทึมนี้สามารถตรวจจับเซลล์มะเร็งได้อย่างแม่นยำและรวดเร็ว ทำให้สามารถวิเคราะห์ตัวอย่างทางพยาธิวิทยาได้อย่างมีประสิทธิภาพ งานวิจัยนี้ยังแสดงให้เห็นว่าอัลกอริทึมที่พัฒนาขึ้นสามารถนำไปใช้ในระบบการวินิจฉัยอัตโนมัติ ช่วยลดภาระงานของพยาธิแพทย์และเพิ่มความแม่นยำ

งานวิจัยเหล่านี้ตอกย้ำถึงความก้าวหน้าที่มีอยู่ในการพัฒนาเทคโนโลยีการตรวจจับวัตถุ โดยการแสดงให้เห็นถึงความแม่นยำและประสิทธิภาพที่มีค่าสูงของระบบปัญญาประดิษฐ์ในการทำนายและตรวจจับวัตถุต่าง ๆ ในภาพและวิดีโอที่ซับซ้อน การประยุกต์ใช้เทคโนโลยีนี้ได้รวมถึงหลากหลายสาขา ทั้งการแพทย์ที่ใช้ในการวินิจฉัยโรคและตรวจสอบภาพการแพทย์ การรักษาความปลอดภัยที่ใช้ในระบบตรวจจับการกระทำที่เป็นอันตราย และการพัฒนายานพาหนะไร้คนขับที่ใช้ในการระบุวัตถุและสภาพแวดล้อมรอบตัวเพื่อการนำทางและควบคุมระบบของรถอัจฉริยะ การวิจัยเหล่านี้ไม่เพียงแต่จะเสริมสร้างความเป็นไปได้ในการประยุกต์ใช้ในชีวิตประจำวันแล้ว แต่ยังเปิดโอกาสให้เทคโนโลยีนี้มีผลกระทบในด้านการเศรษฐกิจและสังคมทั่วไปได้อย่างกว้างขวาง

ในสาขาการแพทย์ก็เช่นเดียวกัน การใช้เทคโนโลยีปัญญาประดิษฐ์สามารถช่วยให้การวินิจฉัยโรคเป็นไปอย่างรวดเร็วและแม่นยำมากขึ้น ตัวอย่างเช่น การตรวจจับเนื้องอกในภาพถ่าย MRI และ CT scans หรือภาพทางการแพทย์อื่น ๆ ระบบปัญญาประดิษฐ์สามารถช่วยแยกแยะโรคและความผิดปกติที่ยากต่อการตรวจสอบด้วยตาเปล่าได้ โดยลดเวลาในการตรวจวินิจฉัยและช่วยให้การรักษาผู้ป่วยเป็นไปอย่างมีประสิทธิภาพมากขึ้น

นอกจากนี้ เทคโนโลยีนี้ยังมีการประยุกต์ใช้ในงานด้านการรักษาความปลอดภัย โดยใช้ระบบตรวจจับการกระทำที่เป็นอันตรายซึ่งสามารถตรวจจับเหตุการณ์ที่เป็นไปได้เช่น การขโมยหรือการละเมิดความปลอดภัยในพื้นที่ต่าง ๆ ได้อย่างรวดเร็วและมีประสิทธิภาพ

อีกทั้งในด้านการพัฒนายานพาหนะไร้คนขับ เทคโนโลยีนี้ช่วยในการระบุวัตถุและสภาพแวดล้อมรอบตัว เพื่อช่วยในการนำทางและควบคุมระบบของรถได้อย่างแม่นยำและปลอดภัยมากยิ่งขึ้น

บทที่ 3

วิธีดำเนินการวิจัย

ในบทนี้จะกล่าวถึงกระบวนการสร้างภาพของดวงตา จมูก และปาก โดยใช้เทคโนโลยี Midjourney และ Stylegan2-ADA-Pytorch รวมถึงการตรวจจับวัตถุเพื่อให้ได้ภาพที่แม่นยำและสวยงามในรูปแบบที่สอดคล้องกับความเป็นจริงของชีวิตประจำวันของมนุษย์ โดยในขั้นตอนแรกจะใช้เทคโนโลยี Midjourney เพื่อสร้างภาพที่สมจริงและสวยงามของดวงตา จมูก และปาก จากนั้นจะใช้ Stylegan2-ADA-Pytorch เพื่อปรับปรุงรายละเอียดและลวดลายของภาพให้มีความสวยงามและเหมือนจริงมากยิ่งขึ้น นอกจากนี้ยังมีการใช้เทคนิคการตรวจจับวัตถุเพื่อช่วยให้ระบบสามารถตรวจจับและปรับปรุงภาพได้อย่างถูกต้องและมีประสิทธิภาพ ในการสร้างภาพดวงตา จมูก และปากในที่สุดผ่านการปรับแต่งและการประมวลผลที่เหมาะสม ระบบจะสามารถสร้างภาพที่สวยงามและสมจริงได้อย่างมีประสิทธิภาพสูงสุด ซึ่งสามารถนำไปใช้งานได้หลากหลาย การประยุกต์ใช้ทางด้านศิลปะและการสร้างภาพถ่าย

3.1 เครื่องมือที่ใช้ในการทำวิจัย

ตารางที่ 3.1 เครื่องมือ วัสดุ และอุปกรณ์ที่ใช้ในงานวิจัย

เครื่องมือ
1. เครื่องคอมพิวเตอร์ตั้งโต๊ะ (desktop) <ul style="list-style-type: none">- มีหน่วยประมวลผลกลาง CPU Intel Core I7 8700K ใช้แผงวงจรหลัก (mainboard) ในคอมพิวเตอร์รุ่น Gigabyte Z370 AORUS Gaming 5-CF- หน่วยความจำ RAM ความจุ 32GB- หน่วยประมวลผลกราฟิก GPU Nvidia GeForce GTX 1080 Ti หน่วยความจำ RAM ของหน่วยประมวลผลภาพความจุ 11GB- ทำงานบนระบบปฏิบัติการ Microsoft Windows 11
ซอฟต์แวร์
2. ปัญญาประดิษฐ์เพื่อสร้างภาพเหมือนจริง Midjourney รุ่น Model V5.0

ตารางที่ 3.2 เครื่องมือ วัสดุ และอุปกรณ์ที่ใช้ในงานวิจัย (ต่อ)

ซอฟต์แวร์
3. ปัญญาประดิษฐ์ปรับปรุงภาพ Stylegan2-ADA-Pytorch (Style Generative Adversarial Networks 2 with Adaptive Discriminator Augmentation) รุ่นปี 2021 ทำงานภายใต้ภาษา Python ซึ่งเป็นซอฟต์แวร์สร้างภาพเสมือนจริงชนิด GANs
4. เว็บไซต์ https://www.makesense.ai เพื่อใช้ซอฟต์แวร์ในเว็บไซต์ทำการระบุป้าย (Labeling) วัตถุที่สนใจ
5. ซอฟต์แวร์ตรวจหาวัตถุ YOLOv5

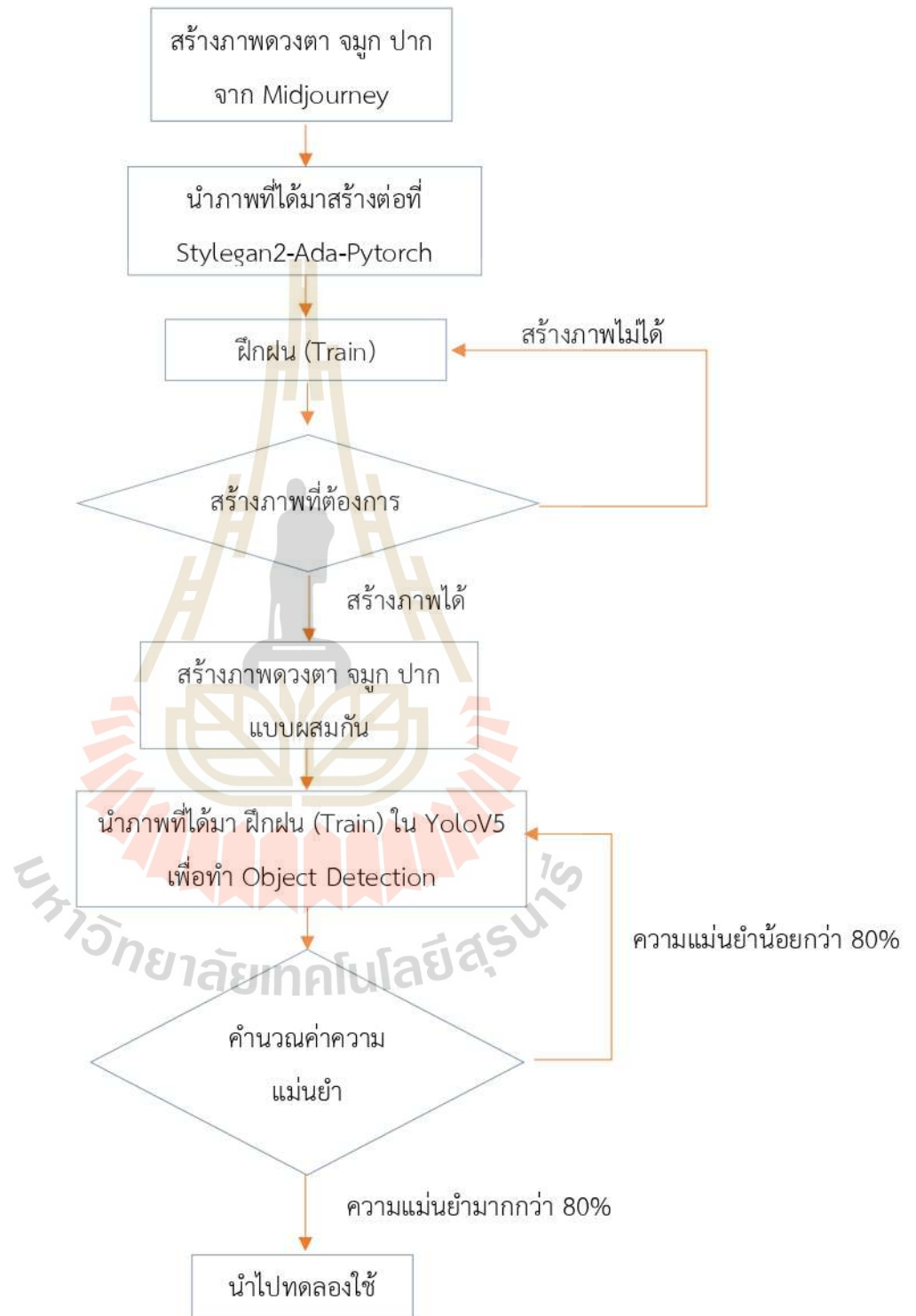
3.2 การศึกษาข้อมูลที่เกี่ยวข้อง

3.2.1 การทบทวนวรรณกรรมหรือผลงานวิจัยที่เกี่ยวข้อง มีความสำคัญเพื่อให้เป็นแนวทางในการดำเนินงานวิจัยต่อไป

3.2.2 การทำความเข้าใจเกี่ยวกับ StyleGAN เป็นการศึกษาเกี่ยวกับวิธีการทำงานของโมเดลปัญญาประดิษฐ์ที่ใช้ในการสร้างภาพเสมือนจริง

3.2.3 การทำความเข้าใจเกี่ยวกับซอฟต์แวร์ที่นำมาใช้ในงานวิจัย เป็นการศึกษาเกี่ยวกับเครื่องมือที่จำเป็นต้องใช้ในการดำเนินงานวิจัย

3.3 FlowChart การทำงาน



3.4 ขั้นตอนการทำงานวิจัย

3.4.1 สร้างแบบจำลองภาพของ ดวงตา จมูก และปาก

สร้างภาพดวงตา จมูก และ ปาก เสมือนจริงโดยใช้โปรแกรม Midjourney ซึ่งเป็นโปรแกรมที่สามารถสร้างภาพเสมือนจริงโดยสร้างภาพจำนวนทั้งหมด 53 ภาพ 50 ภาพ และ 55 ภาพตามลำดับ การสร้างภาพดวงตา จมูก และ ปาก ในขั้นตอนนี้ทำได้โดยกำหนดคำสั่งที่ Midjourney ให้สร้างภาพดวงตา จมูก และปาก รูปแบบต่าง ๆ ขึ้นมา

3.4.2 นำภาพทั้งหมดที่ถูกสร้างขึ้นด้วยโปรแกรม Midjourney ไปสร้างภาพเพิ่มเติม

จากการนำภาพทั้งหมดที่ถูกสร้างขึ้นด้วยโปรแกรม Midjourney มาทำการฝึกด้วยซอฟต์แวร์ Stylegan2-Ada-Pytorch เพื่อสร้างภาพเพิ่มเติมให้แตกต่างจากภาพเดิม โดยการผสมระหว่างภาพเหล่านั้นเพื่อสร้างภาพใหม่ที่มีการเปลี่ยนแปลงของดวงตา จมูก และ ปาก ซึ่งได้ผลลัพธ์เป็นภาพจำนวน 459 ภาพ 483 ภาพ และ 462 ภาพตามลำดับ การสร้างภาพในขั้นตอนนี้ถูกดำเนินการโดยการกำหนดพารามิเตอร์ (parameters) เพื่อทำการผสมระหว่างภาพแหล่งข้อมูล A และ แหล่งข้อมูล B ในกระบวนการที่เรียกว่าการสร้างภาพแบบผสม (Mixing)

3.4.3 ฝึกฝน ปัญญาประดิษฐ์ สำหรับการตรวจหาวัตถุ

ในขั้นตอนถัดมาเป็นขั้นตอนการฝึกการตรวจจับวัตถุโดยใช้ซอฟต์แวร์ YOLOv5 ในขั้นตอนนี้จะมีขั้นตอนย่อยในการดำเนินการ คือ

3.4.3.1 การระบุป้าย (labeling)

ในขั้นตอนนี้เราจะใช้เว็บไซต์ <https://www.makesense.ai> เพื่อระบุข้อมูลของวัตถุที่ปรากฏในภาพ เป็นส่วนของดวงตาที่ปรากฏ โดยหลังจากการระบุป้ายแล้ว เราจะได้รับข้อมูลไฟล์ออกมาในรูปแบบ .txt ซึ่งเป็นข้อมูลที่จำเป็นสำหรับการฝึกระบบปัญญาประดิษฐ์ในขั้นตอนถัดไป

3.4.3.2 การฝึก (training)

ขั้นตอนนี้จะนำข้อมูลไฟล์ภาพพร้อมกับข้อมูลไฟล์ระบุป้ายที่ได้ในขั้นตอนก่อนหน้าไปฝึกกระบวนการตรวจหาวัตถุ YOLOv5 โดยกำหนดค่าพารามิเตอร์ต่าง ๆ ดังนี้ ตั้งค่าจำนวนรอบของการเทรน (Epochs) ที่ 15,000 รอบ และตั้งค่าจำนวนรายการข้อมูลที่จะให้ Optimize จำนวนในหนึ่งครั้ง (Batch Size) ที่ 64 โดยค่าเหล่านี้จะมีผลต่อความเร็วและความแม่นยำในการคำนวณการฝึกในระบบในขั้นตอนนี้จะนำข้อมูลไฟล์ภาพพร้อมกับข้อมูลไฟล์ระบุป้ายที่ได้รับมาไปใช้ในการฝึก YOLOv5 โดยกำหนดค่าพารามิเตอร์ต่าง ๆ ดังนี้ ตั้งค่าจำนวนรอบการฝึก (Epochs) ที่ 500,000 รอบ และตั้งค่าจำนวนรายการข้อมูลที่จะให้ Optimize จำนวนในหนึ่งครั้ง (Batch Size) ที่ 64 โดยค่าเหล่านี้จะมีผลต่อความเร็วและความแม่นยำในการคำนวณ

3.4.3.3 การตรวจหาวัตถุและการวัดประสิทธิภาพ

ในขั้นตอนสุดท้ายนี้ จะเป็นการนำเข้าภาพวีดิทัศน์ดิจิทัลและภาพถ่ายเพื่อให้ระบบตรวจหาวัตถุที่ได้ฝึกไว้ตรวจหาดวงตาที่ปรากฏในภาพ และจะทำการวัดประสิทธิภาพของระบบการตรวจหาวัตถุนั้น การตรวจหาวัตถุในขั้นตอนสุดท้ายนี้ เราจะนำเข้าภาพวีดิทัศน์ดิจิทัลและภาพถ่ายเพื่อให้ระบบตรวจหาวัตถุที่ได้ฝึกไว้ตรวจหาดวงตาที่ปรากฏในภาพ พร้อมกับการวัดประสิทธิภาพของระบบการตรวจหาวัตถุนั้น



บทที่ 4

ผลการวิจัยและอภิปรายผล

ในบทนี้ เราจะนำเสนอผลการทดลอง การวิเคราะห์ และการอภิปรายผลการสร้างภาพโดยใช้ Midjourney ร่วมกับ Stylegan2-ADA-Pytorch ซึ่งเป็นการผสมผสานเทคโนโลยีปัญญาประดิษฐ์ในการสร้างภาพเสมือนจริง การทดลองนี้ให้ผลลัพธ์ที่น่าสนใจ ซึ่งสามารถนำไปใช้ในการวิเคราะห์คุณภาพของภาพที่สร้างขึ้นได้ สามารถนำภาพเหล่านี้ไปตรวจหาวัตถุได้ และใช้เมตริกซ์ความสับสนเพื่อวัดประสิทธิภาพของระบบปัญญาประดิษฐ์ที่ใช้ในการตรวจหาวัตถุ ผลลัพธ์จะช่วยให้ทราบถึงประสิทธิภาพและคุณภาพของการตรวจหาวัตถุด้วยปัญญาประดิษฐ์ ผลลัพธ์ที่ได้จะให้ข้อมูลและข้อสรุปที่เป็นประโยชน์ในการปรับปรุงและพัฒนาระบบในอนาคต

4.1 ผลการทดลองการสร้างภาพดวงตา จมูก และ ปาก

Midjourney ใช้โมเดลการเรียนรู้เชิงลึก (Deep Learning) เพื่อสร้างภาพใบหน้าที่สมจริง การทำงานของ Midjourney เริ่มจากการวิเคราะห์และแยกแยะลักษณะเฉพาะของภาพดวงตา จมูก และปาก จากนั้นจะสร้างภาพใหม่โดยอิงจากข้อมูลที่ได้เรียนรู้มา ซึ่งกระบวนการนี้เรียกว่า Generative Adversarial Networks (GANs) โดย GANs ประกอบด้วยสองโมเดลหลักคือ Generator และ Discriminator ที่ทำงานร่วมกันเพื่อปรับปรุงคุณภาพของภาพที่สร้างขึ้น

การสร้างภาพดวงตา ดวงตาเป็นองค์ประกอบที่สำคัญและซับซ้อนของใบหน้า AI ต้องสามารถสร้างดวงตาที่ดูมีชีวิตและสมจริงได้ โดย Midjourney จะใช้ข้อมูลจากภาพดวงตาจำนวนมากเพื่อเรียนรู้ลักษณะเฉพาะ เช่น รูปร่าง ขนาด สีของดวงตา และลักษณะของการสะท้อนแสงในดวงตา เพื่อให้ภาพที่สร้างขึ้นมามีความสมจริงมากที่สุด



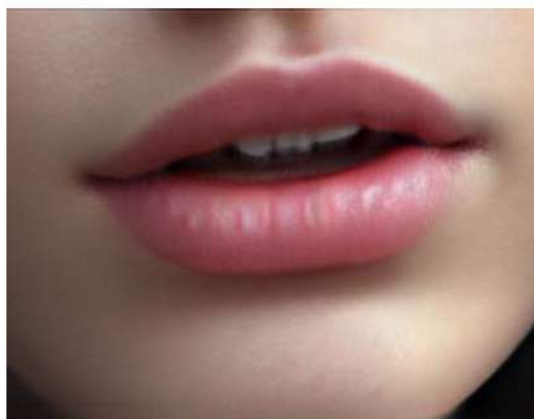
รูปที่ 4.1 ตัวอย่างภาพของดวงตาที่สร้างขึ้นโดย Midjourney

การสร้างภาพจมูก จมูกเป็นส่วนกลางของใบหน้าและมีลักษณะที่หลากหลาย ทั้งในแง่ของรูปร่าง ขนาด และรายละเอียด การสร้างภาพจมูกด้วย AI จำเป็นต้องมีความแม่นยำสูงเพื่อให้เข้ากับลักษณะเฉพาะของใบหน้าแต่ละคน ใน Midjourney ใช้การประมวลผลภาพและการวิเคราะห์เชิงลึกเพื่อสร้างจมูกที่ดูสมจริงและสอดคล้องกับองค์ประกอบอื่น ๆ ของใบหน้า



รูปที่ 4.2 ตัวอย่างภาพของจมูกที่สร้างขึ้นโดย Midjourney

การสร้างภาพปาก ปากเป็นส่วนที่มีการแสดงอารมณ์และความรู้สึกได้มากที่สุดในใบหน้า การสร้างภาพปากที่สมจริงจึงต้องสามารถแสดงลักษณะต่างๆ เช่น รอยยิ้ม ความโกรธ ความเศร้า ได้อย่างชัดเจน Midjourney ใช้ข้อมูลภาพปากจากแหล่งข้อมูลต่าง ๆ เพื่อเรียนรู้และสร้างภาพปากที่ดูสมจริงและมีชีวิตชีวา



รูปที่ 4.3 ตัวอย่างภาพของปากที่สร้างขึ้นโดย Midjourney

การใช้ Midjourney ในการสร้างภาพดวงตา จมูก และปากเป็นตัวอย่างของการนำเทคโนโลยี AI มาประยุกต์ใช้ในการสร้างสรรค์ภาพดิจิทัล ความสามารถในการสร้างภาพที่สมจริงและรายละเอียดสูง ช่วยให้ผู้ใช้งานสามารถสร้างภาพใบหน้าที่มีคุณภาพและความสมจริงสูงได้อย่างรวดเร็วและง่ายดาย เทคโนโลยีนี้ยังเปิดโอกาสให้เกิดนวัตกรรมใหม่ ๆ ในหลายสาขา เช่น การออกแบบตัวละครในเกม การสร้างภาพในอุตสาหกรรมภาพยนตร์ และการพัฒนาโปรแกรมการเรียนรู้เชิงลึกในด้านต่าง ๆ

StyleGAN2-ADA (Adaptive Discriminator Augmentation) เป็นเทคโนโลยีที่พัฒนาโดย NVIDIA ซึ่งปรับปรุงจาก StyleGAN2 เดิม โดยเพิ่มกลไกการปรับตัวของ Discriminator เพื่อให้สามารถเรียนรู้ได้ดีแม้ในกรณีที่มีข้อมูลจำกัด นอกจากนี้ การใช้งาน PyTorch สามารถนำไปใช้ในโปรเจกต์ต่าง ๆ ง่ายขึ้น

การสร้างภาพดวงตา ดวงตาเป็นส่วนที่ซับซ้อนและสำคัญในการสร้างภาพใบหน้า AI ต้องสามารถสร้างดวงตาที่มีชีวิตชีวาและสมจริง StyleGAN2-ADA-PyTorch ใช้เทคนิคการแยกและปรับแต่งคุณลักษณะเฉพาะของดวงตา เช่น รูปร่างของตา สีตา และการสะท้อนแสง โดยโมเดลนี้สามารถสร้างดวงตาที่มีรายละเอียดสูงและสอดคล้องกับลักษณะใบหน้าทั้งหมดได้อย่างแม่นยำ



รูปที่ 4.4 ตัวอย่างภาพของดวงตาที่ถูกปรับปรุงขึ้นใหม่โดย StyleGAN2-Ada-PyTorch

การสร้างภาพจมูก จมูกเป็นศูนย์กลางของใบหน้าและมีลักษณะเฉพาะที่หลากหลาย การสร้างจมูกที่สมจริงต้องอาศัยการเรียนรู้จากข้อมูลจำนวนมาก StyleGAN2-ADA-PyTorch สามารถปรับตัวเพื่อเรียนรู้ลักษณะของจมูกที่แตกต่างกัน เช่น ขนาด รูปร่าง และมุมของจมูก ทำให้ภาพจมูกที่สร้างขึ้นมีความสมจริงและสอดคล้องกับภาพใบหน้าอื่น ๆ



รูปที่ 4.5 ตัวอย่างภาพของจมูกที่ถูกปรับปรุงขึ้นใหม่โดย StyleGAN2-Ada-PyTorch

การสร้างภาพปาก ปากเป็นส่วนที่แสดงอารมณ์และความรู้สึกได้มากที่สุด การสร้างภาพปากที่สมจริงจึงต้องสามารถแสดงรายละเอียดต่าง ๆ เช่น รอยยิ้ม เส้นริมฝีปาก และลักษณะการเปิด-ปิด

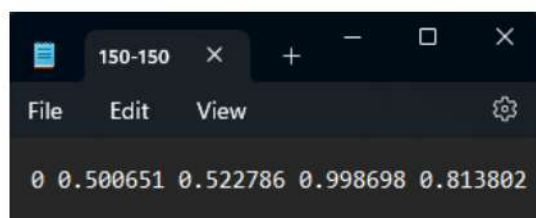
ปากได้อย่างชัดเจน StyleGAN2-ADA-PyTorch ใช้การเรียนรู้เชิงลึกเพื่อสร้างภาพปากที่สามารถแสดงอารมณ์ได้หลากหลายและมีความสมจริงสูง



รูปที่ 4.6 ตัวอย่างภาพของปากที่ถูกปรับปรุงขึ้นใหม่โดย Stylegan2-Ada-Pytorch

การใช้ StyleGAN2-ADA-PyTorch ในการสร้างภาพใบหน้ามีประโยชน์ในหลายสาขา ไม่ว่าจะเป็นการออกแบบตัวละครในเกม การสร้างภาพในอุตสาหกรรมภาพยนตร์ การพัฒนาตัวแบบในการศึกษาทางการแพทย์ การสร้างภาพใบหน้าที่สมจริงและรายละเอียดสูง โดยเฉพาะในส่วนของดวงตา จมูก และปาก การใช้งานเทคโนโลยีนี้ไม่เพียงแต่ช่วยเพิ่มประสิทธิภาพในการสร้างภาพดิจิทัล แต่ยังเปิดโอกาสให้เกิดนวัตกรรมและการพัฒนาในหลายสาขา ทำให้ผู้ใช้งานสามารถสร้างสรรค์งานที่มีคุณภาพสูงและตอบสนองความต้องการของตลาดได้อย่างมีประสิทธิภาพ

การระบุป้าย (Labeling) คือกระบวนการที่มนุษย์หรือเครื่องมืออัตโนมัติทำการกำหนดป้ายกำกับ (labels) ให้กับข้อมูล เช่น ภาพ ข้อความ หรือเสียง เพื่อให้โมเดลปัญญาประดิษฐ์ สามารถเรียนรู้และทำความเข้าใจข้อมูลเหล่านั้นได้อย่างถูกต้อง ตัวอย่างเช่น ในการฝึกโมเดลตรวจจับวัตถุ (Object Detection) อาจต้องระบุป้ายกำกับวัตถุต่าง ๆ ในภาพ เช่น ดวงตา จมูก และ ปาก เป็นต้น



รูปที่ 4.7 ตัวอย่างข้อมูลที่ได้จากการระบุป้ายโดยใช้ เว็บไซต์ <https://www.makesense.ai>

ในรูปที่ 4.7 ค่าแต่ละตัวมีความหมายดังนี้: ตัวเลขตัวแรกสุดนับจากด้านซ้าย (เลข 0) แสดงถึงประเภทของวัตถุ โดยที่เลข 0 ในที่นี้หมายถึง วัตถุที่เป็นดวงตา ตัวเลขตัวที่ 2 และ 3 ถัดมา (เลข

0.500651 และ 0.522786) หมายถึงตำแหน่งพิกัดตรงกลางของวัตถุในแนวแกน X และแกน Y โดยการเทียบตามสัดส่วนในแนวความกว้างและสูงของรูปภาพเป็น 1 ส่วนเลขตัวที่ 4 และ 5 ถัดมา (เลข 0.998698 และ 0.813802) หมายถึงความกว้างและความสูงของวัตถุที่ปรากฏในรูปภาพ เมื่อเทียบตามสัดส่วนในแนวความกว้างและสูงของรูปภาพเป็น 1



รูปที่ 4.8 ตัวอย่างการระบุป้ายโดยใช้ เว็บไซต์ <https://www.makesense.ai>

การเทรนการตรวจหาวัตถุโดยใช้ YOLOv5 เป็นหนึ่งในเทคโนโลยีการตรวจจับวัตถุที่มีประสิทธิภาพสูงและเป็นที่ยอมรับอย่างมาก YOLOv5 ได้รับการพัฒนาเพื่อให้ความรวดเร็วและแม่นยำมากขึ้นเมื่อเปรียบเทียบกับเวอร์ชันก่อนหน้า ด้วยโครงสร้างที่มีประสิทธิภาพและการใช้เทคนิคการเรียนรู้เชิงลึกที่ทันสมัย

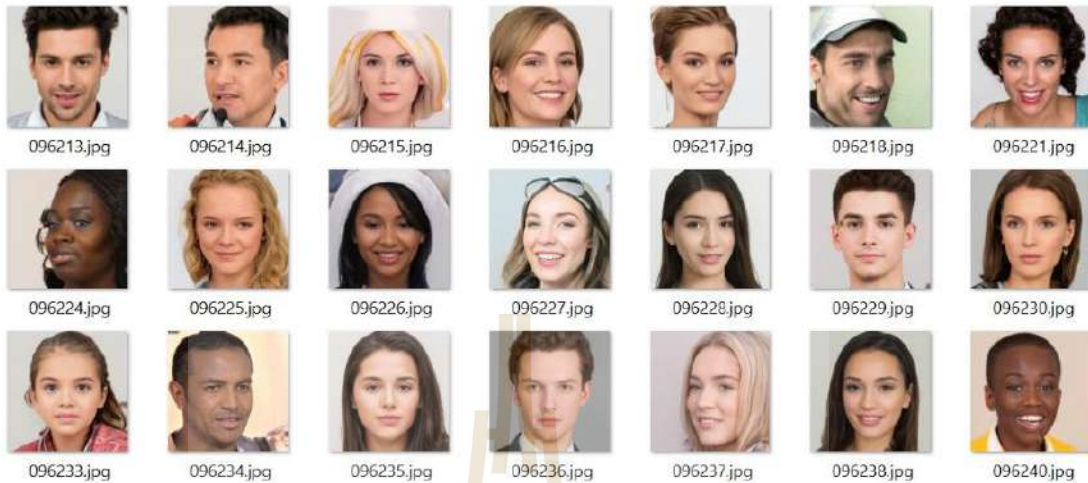
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size			
27961/499999	11.1G	0.005431	0.002571	0.0003929	116	640: 100%	27/27	[00:34<00:00, 1.28s/it]	
Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1	[00:01<00:00, 1.13s/it]	
all	79	226	0.942	0.879	0.917	0.566			
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size			
27962/499999	11.1G	0.005616	0.002522	0.0004652	106	640: 100%	27/27	[00:33<00:00, 1.26s/it]	
Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1	[00:01<00:00, 1.01s/it]	
all	79	226	0.943	0.879	0.917	0.566			
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size			
27963/499999	11.1G	0.005394	0.002003	0.000439	116	640: 100%	27/27	[00:27<00:00, 1.01s/it]	
Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1	[00:01<00:00, 1.05s/it]	
all	79	226	0.943	0.879	0.917	0.565			
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size			
27964/499999	11.1G	0.0055	0.002047	0.0004251	129	640: 100%	27/27	[00:30<00:00, 1.12s/it]	
Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1	[00:01<00:00, 1.03s/it]	
all	79	226	0.945	0.879	0.917	0.568			
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size			
27965/499999	11.1G	0.005414	0.002402	0.0004547	215	640: 48%	13/27	[00:16<00:14, 1.05s/it]	

รูปที่ 4.9 ภาพตัวอย่างการประมวลผลการฝึกการตรวจหาดวงตาของ YOLOv5

4.2 คำนวณประสิทธิภาพของ ปัญญาประดิษฐ์ ด้วย เมตริกซ์ความสับสน

จากเว็บไซต์ <https://generated.photos/> ได้คัดเลือกภาพหน้าคนที่มีคุณภาพและมีดวงตาชัดเจนโดยไม่ใส่แว่นตา และไม่มีภาพที่ตาปิดหรือก้มหน้าเกินไป หรือเอียงหน้ามากเกินไป ซึ่งเป็น

เงื่อนไขที่จำเป็นสำหรับการทดสอบประสิทธิภาพของระบบตรวจหาวัตถุ ซึ่งผลลัพธ์ที่ได้คือภาพทั้งหมดจำนวน 3,161 ภาพ



รูปที่ 4.10 ภาพตัวอย่างที่ใช้ทดสอบ

4.2.1 คำนวณค่าของ ดวงตา โดยใช้เมทริกซ์ความสับสน

ตารางที่ 4.1 เมทริกซ์ความสับสนของดวงตา

Confusion Matrix		
EYE	3116	0
Background	45	0
	EYE	Background

ความถูกต้อง ของ ดวงตา หาได้จาก

$$\frac{3116 + 0}{3116 + 0 + 45 + 0} = 0.99$$

ความแม่นยำ ของ ดวงตา หาได้จาก

$$\frac{3116}{3116 + 0} = 1$$

ค่าเรียกคืน ของ ดวงตา หาได้จาก

$$\frac{3116}{3116 + 45} = 0.99$$

จากการคำนวณเมทริกซ์ความสับสนของดวงตาของพบว่ามีค่าความถูกต้องอยู่ที่ 0.99 ค่าความแม่นยำ 1 ค่าเรียกคืน 0.99 ซึ่งค่ายิ่งเข้าใกล้ 1 มากเท่าไรหมายความว่า เป็นค่าที่ดี ซึ่งสามารถเปลี่ยนเป็นร้อยละโดยการคูณ 100 จะได้ ค่าความถูกต้องร้อยละ 98.58 ค่าความแม่นยำร้อยละ 100.00 ค่าการเรียกคืนร้อยละ 98.58

ตารางที่ 4.2 ค่าเมทริกซ์ความสับสนของดวงตาเมื่อคิดเป็นร้อยละ

ค่าเมทริกซ์ความสับสน		ร้อยละ
ค่าความถูกต้อง	0.99×100	98.58
ค่าความแม่นยำ	1×100	100
ค่าเรียกคืน	0.99×100	98.58

ตารางที่ 4.3 เมทริกซ์ความสับสนของจมูก

Confusion Matrix		
NOSE	2906	0
Background	255	0
	NOSE	Background

$$\frac{\text{ความถูกต้อง ของ จมูก หาได้จาก } 2906 + 0}{2906 + 0 + 255 + 0} = 0.92$$

ความแม่นยำ ของ จมูก หาได้จาก

$$\frac{2906}{2906 + 0} = 1$$

ค่าเรียกคืน ของ จมูก หาได้จาก

$$\frac{2906}{2906 + 255} = 0.92$$

จากการคำนวณเมทริกซ์ความสับสนของจุมกพบว่ามีค่าความถูกต้องอยู่ที่ 0.92 ค่าความแม่นยำ 1 ค่าเรียกคืน 0.92 ซึ่งค่ายิ่งเข้าใกล้ 1 มากเท่าไรหมายความว่าค่าที่ดี ซึ่งสามารถเปลี่ยนเป็นร้อยละโดยการคูณ 100 จะได้ ค่าความถูกต้องร้อยละ 91.98 ค่าความแม่นยำร้อยละ 100.00 ค่าการเรียกคืนร้อยละ 91.93

ตารางที่ 4.4 ค่าเมทริกซ์ความสับสนของจุมกเมื่อคิดเป็นร้อยละ

ค่าเมทริกซ์ความสับสน		ร้อยละ
ค่าความถูกต้อง	0.92×100	91.93
ค่าความแม่นยำ	1×100	100
ค่าเรียกคืน	0.92×100	91.93

ตารางที่ 4.5 เมทริกซ์ความสับสนของปาก

Confusion Matrix		
MOUTH	3074	0
Background	87	0
	MOUTH	Background

$$\frac{\text{ความถูกต้อง ของ ปาก หาได้จาก}}{3074 + 0} = 0.97$$

$$\frac{\text{ความแม่นยำ ของ ปาก หาได้จาก}}{3074} = 1$$

$$\frac{\text{ค่าเรียกคืน ของ ปาก หาได้จาก}}{3074 + 87} = 0.97$$

จากการคำนวณเมตริกซ์ความสับสนของดวงตาของพบว่ามีค่าความถูกต้องอยู่ที่ 0.97 ค่าความแม่นยำ 1 ค่าเรียกคืน 0.97 ซึ่งค่ายิ่งเข้าใกล้ 1 มากเท่าไรหมายความว่า เป็นค่าที่ดี ซึ่งสามารถเปลี่ยนเป็นร้อยละโดยการคูณ 100 จะได้ ค่าความถูกต้องร้อยละ 97.34 ค่าความแม่นยำร้อยละ 100.00 ค่าการเรียกคืนร้อยละ 97.34

ตารางที่ 4.6 ค่าเมตริกซ์ความสับสนของปากเมื่อคิดเป็นร้อยละ

ค่าเมตริกซ์ความสับสน		ร้อยละ
ค่าความถูกต้อง	0.97×100	97.34
ค่าความแม่นยำ	1×100	100
ค่าเรียกคืน	0.97×100	97.34

4.3 ทดลองใช้กับรูปภาพที่ถูกสร้างขึ้นด้วยปัญญาประดิษฐ์

จากค่าของ เมตริกซ์ความสับสน ที่ทราบนั้นพบว่ามีค่าความถูกต้อง ค่าความแม่นยำ และ ค่าเรียกคืน มากกว่าร้อยละ 80 จึงนำไปทดลองใช้กับรูปภาพที่ถูกสร้างขึ้นด้วยปัญญาประดิษฐ์



(a)

(b)

รูปที่ 4.11 แสดงตัวอย่างการการตรวจหาดวงตาที่ถูกต้อง

(a) สามารถระบุได้ว่าในรูปภาพนั้นมี ดวงตา จมูก และ ปาก

(b) สามารถระบุได้ว่าในรูปภาพนั้นมี ดวงตา จมูก และ ปาก



รูปที่ 4.12 แสดงตัวอย่างการตรวจหาดวงตาที่ไม่ถูกต้อง

4.4 วิเคราะห์และอภิปรายผล

จากผลลัพธ์ที่ได้แสดงให้เห็นว่าการสร้างเทมเพลตการฝึกสอนสำหรับการตรวจหาดวงตาจากภาพถ่ายประกอบด้วย 3 ส่วนหลัก ได้แก่

4.3.1 ส่วนการสร้างภาพจำลองดวงตา จมูก และ ปากของมนุษย์ด้วย Midjourney การใช้เทคโนโลยีนี้ช่วยให้สามารถสร้างภาพจำลองที่มีความเหมือนจริงของดวงตา จมูก และ ปากของมนุษย์ได้อย่างมีประสิทธิภาพ ภาพที่สร้างขึ้นมีความละเอียดสูงและสามารถปรับแต่งได้ตามต้องการ ซึ่งเป็นการสร้างฐานข้อมูลภาพที่มีคุณภาพสำหรับการฝึกสอนโมเดล

4.3.2 ส่วนการสร้างภาพเพิ่มเติมให้แตกต่างจากเดิมโดยใช้ซอฟต์แวร์ Stylegan2-ADA-Pytorch การใช้ Stylegan2-ADA-Pytorch ช่วยในการสร้างภาพเพิ่มเติมที่มีความหลากหลายและแตกต่างจากภาพเดิม ซึ่งเป็นการเพิ่มความหลากหลายให้กับฐานข้อมูลภาพ ทำให้โมเดลที่ฝึกสอนสามารถเรียนรู้จากภาพที่มีลักษณะแตกต่างกันได้ดีขึ้น ช่วยลดปัญหา overfitting และเพิ่มความสามารถในการตรวจจับวัตถุในสภาพแวดล้อมที่แตกต่างกัน

4.3.3 ส่วนการระบุข้อมูลของวัตถุที่ปรากฏในภาพโดยเว็บไซต์ <https://www.makesense.ai> การใช้เว็บไซต์นี้ช่วยในการระบุและทำ annotation ข้อมูลในภาพ ทำให้สามารถสร้างฐานข้อมูลที่มีข้อมูลเชิงบรรยายของวัตถุที่ต้องการตรวจจับได้อย่างแม่นยำและเป็นระบบ ซึ่งข้อมูลที่ได้จะนำไปใช้ในการฝึกสอนโมเดล YOLOv5 ต่อไป

ผลการทดลองพบว่า เทมเพลตการฝึกสอนที่สร้างขึ้นสามารถนำมาใช้ฝึก YOLOv5 ในการตรวจหาดวงตา จมูก และ ปากของมนุษย์จากภาพดิจิทัลได้จริงและมีประสิทธิภาพที่ดี ซึ่งหมายความว่า YOLOv5 สามารถตรวจจับและระบุวัตถุที่ต้องการได้อย่างแม่นยำและรวดเร็ว เมื่อทดสอบกับภาพจริง พบว่าโมเดลสามารถตรวจจับดวงตา จมูก และ ปากได้อย่างมีประสิทธิภาพ ไม่เพียงแค่นั้นภาพจำลองแต่ยังสามารถนำไปใช้กับภาพจริงได้ดี

จากการวิเคราะห์นี้ สามารถอนุมานได้ว่าเทมเพลตการฝึกสอนที่สร้างขึ้น มีความสามารถในการสร้างภาพจำลองที่มีคุณภาพสูงและสามารถใช้ในการฝึกสอนระบบตรวจจับวัตถุได้อย่างมีประสิทธิภาพ เทมเพลตนี้สามารถใช้แทนที่รูปภาพดวงตา จมูก และ ปากของมนุษย์



บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากผลลัพธ์ที่ได้แสดงให้เห็นว่า การสร้างเทมเพลตการฝึกสอนสำหรับการตรวจหาดวงตาจากภาพถ่ายประกอบด้วย 3 ส่วน ได้แก่ ส่วนการสร้างภาพจำลองดวงตา จมูก และ ปากของมนุษย์ด้วย Midjourne ส่วนการสร้างภาพเพิ่มเติมให้แตกต่างจากเดิมโดยใช้ซอฟต์แวร์ Stylegan2-ADA-Pytorch และส่วนการระบุข้อมูลของวัตถุที่ปรากฏในภาพโดยเว็บไซต์ <https://www.makesense.ai> สามารถนำมาใช้ฝึก YOLOv5 ในการตรวจหาดวงตา จมูก และ ปากของมนุษย์จากภาพดิจิทัลได้จริง และสามารถตรวจหาได้อย่างมีประสิทธิภาพ ดังนั้นสามารถอนุมานได้ว่าเทมเพลตการฝึกสอนที่สร้างขึ้นสามารถนำมาใช้ประโยชน์ในสร้างภาพจำลองดวงตา จมูก และปากของมนุษย์ทดแทนรูปภาพดวงตา จมูก และ ปากของมนุษย์จริงได้ และเมื่อนำภาพคนจริง ๆ มาใช้ทดสอบในการตรวจหาดวงตา จมูก และ ปาก เบื้องต้นก็พบว่าสามารถตรวจหาได้ดังแสดงในรูปที่ 5.1



รูปที่ 5.1 ตรวจหาดวงตา จมูก และ ปาก จากภาพถ่ายดิจิทัลที่ถ่ายจากบุคคลจริง

จากที่ได้สร้างภาพต้นแบบด้วย Midjourney และสร้างภาพเพิ่มเติม ดวงตา จมูก และ ปาก โดยใช้ซอฟต์แวร์ Stylegan2-ADA-Pytorch พบว่าปัญญาประดิษฐ์สามารถสร้างภาพ ดวงตา จมูก และ ปากได้ได้อย่างมีประสิทธิภาพ จึงนำภาพเหล่านี้ที่ ปัญญาประดิษฐ์ได้สร้างขึ้นไปทำการฝึกสอนเพื่อ

ตรวจหาวัตถุ ว่าสิ่งที่เห็นอยู่นั้นคือ ดวงตา จมูก และ ปาก หรือไม่ โดยกำหนดรอบในการฝึกสอนอยู่ที่ 500,000 รอบ โดยรอบที่มีผลการใช้งานได้ดีที่สุดจะอยู่ 14,759 เมื่อนำปัญญาประดิษฐ์ รอบที่ 14,759 ไปทดสอบด้วยเมทริกซ์ความสับสน โดยทำการคำนวณ และ ทดสอบจากภาพที่ถูกสร้างขึ้น ด้วยปัญญาประดิษฐ์ จาก <https://generated.photos/> จำนวน 3,161 จะพบว่ามีความต่าง ๆ ดังนี้ การตรวจหาดวงตาจากภาพใบหน้ามีความถูกต้องในการตรวจหาเท่ากับร้อยละ 98.58 ค่าความแม่นยำเท่ากับร้อยละ 98.58 และค่าเรียกคืนเท่ากับร้อยละ 100.00 การตรวจหาจมูกจากภาพ ใบหน้ามีความถูกต้องในการตรวจหาเท่ากับร้อยละ 91.93 ค่าความแม่นยำเท่ากับร้อยละ 100.00 และค่าเรียกคืนเท่ากับร้อยละ 91.93 และ การตรวจหาปากจากภาพใบหน้ามีความถูกต้องในการ ตรวจหาเท่ากับร้อยละ 97.34 ค่าความแม่นยำเท่ากับร้อยละ 100.00 และค่าเรียกคืนเท่ากับร้อยละ 97.34

จากการคำนวณ และทดสอบ พบว่ามีค่าความถูกต้อง ค่าความแม่นยำ และ ค่าเรียกคืน มีค่า มากกว่าร้อยละ 85 และ เมื่อนำไปทดสอบกับภาพที่สร้างขึ้นจากปัญญาประดิษฐ์ ก็พบว่าสามารถ ตรวจหา ดวงตา จมูก และ ปาก ได้มีประสิทธิภาพ จึงนำไปทดสอบกับภาพบุคคลจริงพบว่า เมื่อนำ ปัญญาประดิษฐ์ไปตรวจหาวัตถุจากภาพบุคคลจริงผลที่ได้คือ ปัญญาประดิษฐ์สามารถตรวจหาวัตถุ ดวงตา จมูก และ ปาก ได้อย่างมีประสิทธิภาพ จึงสามารถอนุมานได้ว่า สามารถใช้ภาพสังเคราะห์ หรือ ภาพที่ปัญญาประดิษฐ์สร้างขึ้นเป็นข้อมูลในการทำการตรวจหาวัตถุ แทนภาพจากบุคคลจริงได้

5.2 ข้อเสนอแนะ

จากผลวิจัยที่ได้ ทำให้ได้ต้นแบบการสร้างเทมเพลตการฝึกสอนสำหรับการตรวจหาดวงตา จมูก และ ปากจากภาพถ่ายได้ ซึ่งสามารถนำภาพเหล่านี้มาใช้ทดแทนภาพจากบุคคลจริงได้ โดยไม่ จำเป็นต้องนำภาพมนุษย์จริงมาเป็นส่วนหนึ่งของการทดลอง นอกจากนี้ผลจากการฝึกสอนสำหรับการ ตรวจหาดวงตา จมูก และ ปาก จากภาพถ่ายที่ได้ในงานวิจัยนี้ จะถูกนำไปประยุกต์ใช้ในการพิจารณา ตรวจหาวัตถุสำหรับทารกในครรภ์ด้วย ภาพคลื่นเสียงความถี่สูง (ultrasound) ว่าทารกที่อยู่ในครรภ์ นั้นมีโรคปากแหว่งเพดานโหว่ (Cleft Lip and Cleft Palate) หรือไม่ และ อาจจะสามารถนำไป ประยุกต์ใช้ในการพิจารณาในการตรวจหาวัตถุสำหรับ โรคไส้เลื่อนกระบังลมของทารกในครรภ์ (Diaphragmatic Hernia) ซึ่งโรคเหล่านี้สามารถทำให้ทารกที่อยู่ในครรภ์ เมื่อคลอดออกมาแล้วจะ เสียชีวิตภายใน 20 นาที ถ้าไม่มีทีมแพทย์ที่ 2 เตรียมรับช่วงไปผ่าตัดต่อหลังจากคลอดทันที การที่มี การตรวจหาวัตถุสำหรับโรคไส้เลื่อนกระบังลมของทารกในครรภ์ โดยใช้ปัญญาประดิษฐ์ มีความ

เป็นไปได้ที่จะลดความเสี่ยงของความผิดพลาดในการวินิจฉัยการตรวจพบโรคในระยะเริ่มต้น (Early Detection) และ ลดอัตราการเสียชีวิตของทารกหลังคลอด





รายการอ้างอิง

รายการอ้างอิง

- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint. <https://doi.org/10.48550/arXiv.2004.10934>
- Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. *Advances in Neural Information Processing Systems (NIPS)*, 29, 379-387. <https://doi.org/10.48550/arXiv.1605.06409>
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 580-587.
- Girshick, R., He, K., & Dollar, P. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 1440-1448.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems (Vol. 27)*. Curran Associates, Inc.
- Guinness, H. (2023, July 17). The best AI image generators in 2023. Zapier Blog. Retrieved from <https://zapier.com/blog/best-ai-image-generator/>
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2961-2969. <https://doi.org/10.1109/ICCV.2017.322>
- Jiang, H., Shen, Y., Wang, Y., Yue, Y., & Liu, Y. (2019). Detection and classification of diseases in chest X-ray images using YOLOv3. *Journal of Biomedical Informatics*, 93, 103183. <https://doi.org/10.1016/j.jbi.2019.103183>

- Li, R., Wang, H., & Li, X. (2020). Faster R-CNN for cancer cell detection and classification in microscopic images. *Computer Methods and Programs in Biomedicine*, 187, 105142. <https://doi.org/10.1016/j.cmpb.2019.105142>
- Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2980-2988. <https://doi.org/10.1109/ICCV.2017.324>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *European Conference on Computer Vision (ECCV)*, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
- Ma, N., Zhang, J., Liang, W., & Li, X. (2018). Application of Mask R-CNN in medical image object detection and segmentation. *IEEE Transactions on Medical Imaging*, 37(7), 1482-1492. <https://doi.org/10.1109/TMI.2018.2814590>
- Natthawatphongchit, N. (n.d.). มาลองดูวิธีการคิดของ CNN กัน. Medium. Retrieved from <https://medium.com/@natthawatphongchit/มาลองดูวิธีการคิดของ-cnn-กัน-e3f5d73eebaa>
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Machine Learning*, 105-114.
- Redmon, J., & Farnadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779-788.
- Redmon, J., Divvala, S., Girshick, R., & Farnadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788. <https://doi.org/10.1109/CVPR.2016.91>

- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NIPS)*, 28, 91-99. <https://doi.org/10.48550/arXiv.1506.01497>
- Thuan, D. (2021). Evolution of YOLO algorithm and YOLOv5: The state-of-the-art object detection algorithm. Bachelor's Thesis, Information Technology, Oulu University of Applied Sciences.



The logo of Sakon Nakhon Rajabhat University is a circular emblem. It features a central figure of a person sitting on a throne, surrounded by a decorative border. The text 'มหาวิทยาลัยเทคโนโลยีสุรนารี' is written in a semi-circle at the bottom of the emblem.

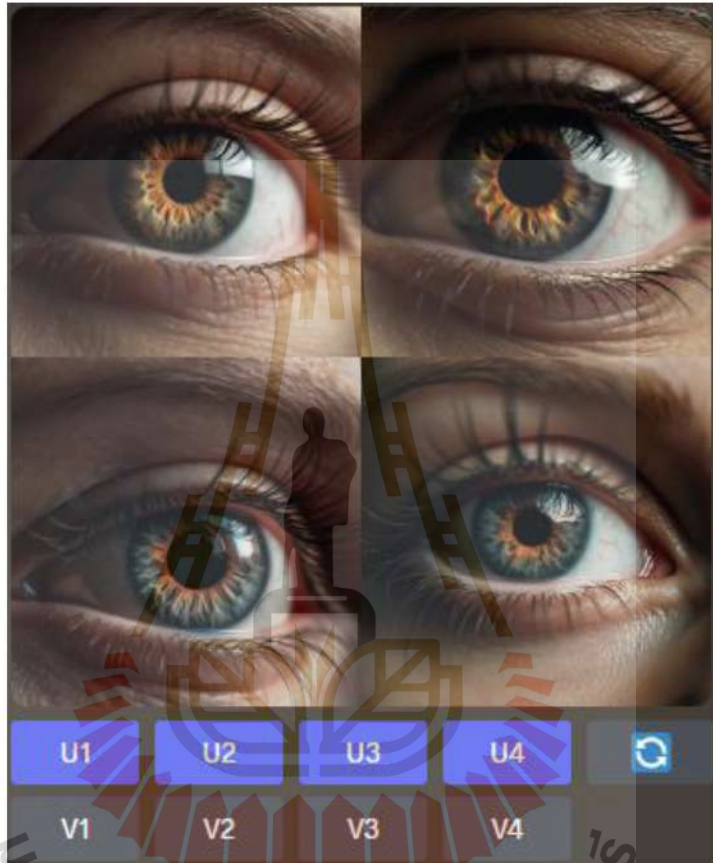
ภาคผนวก ก

การสร้างภาพโดย Midjourney โดยใช้คำสั่งในการสร้าง การสร้างภาพ
เพิ่มเติมโดยใช้ Stylegan2-ADA-Pytorch และการตรวจหาวัตถุ

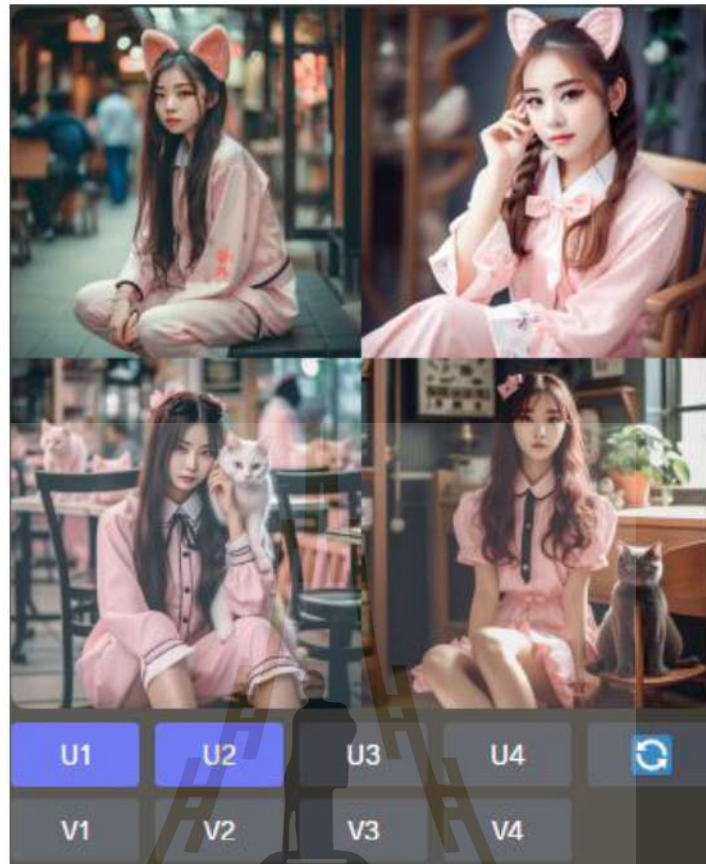
มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ก

1. การสร้างภาพโดย Midjourney โดยใช้คำสั่งในการสร้าง การสร้างภาพเพิ่มเติมโดยใช้ Stylegan2-ADA-Pytorch และการตรวจหาวัตถุ

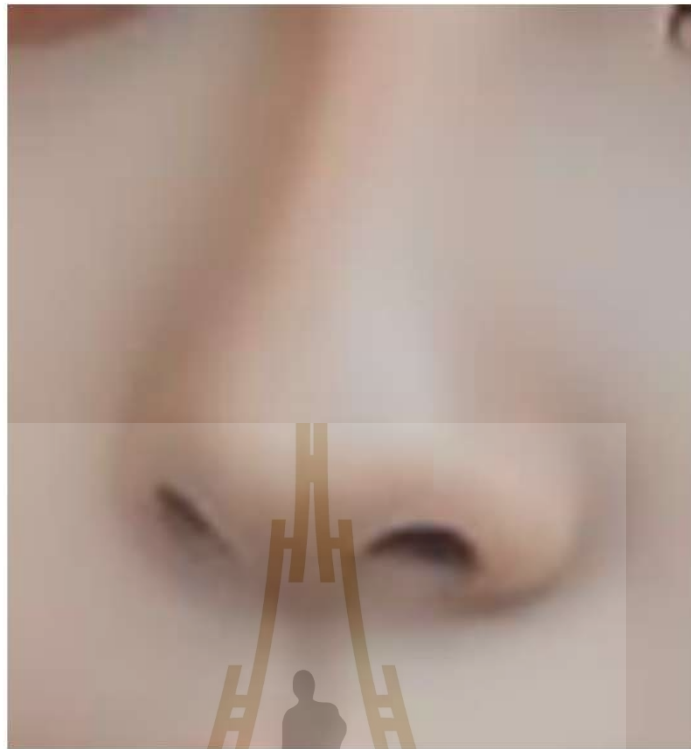


คำสั่งที่ใช้ในการสร้าง create new picture of close-up eye while lying, Ultra Realistic , best quality , 8K , High Resolution
create new picture of close-up black eye , Ultra Realistic , best quality , 8K , High Resolution --v 5 --s 750 -

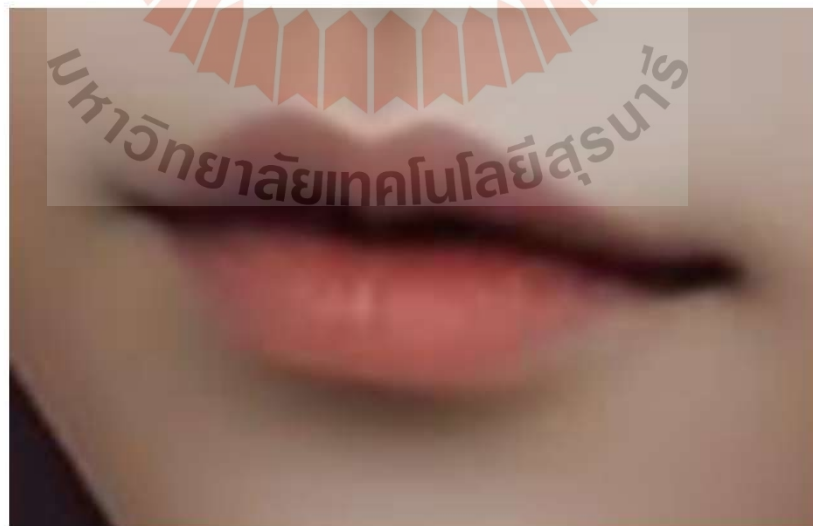


คำสั่งที่ใช้ในการสร้าง full body create Beautiful charm face like Yeji ITZY Teenage Female , long hair, makeup with Glow Glossy Pink lip , wear Japanese maid suit has cat ears ,use Mirrorless Camera Sony A7 Mark 4 with 50mm Lens Photography F/1.4 IOS is 200 SpeedShutter is 1/500 , sharp focus , set against the backdrop of lunchtime cafe, Ultra Realistic , best quality , model pose --v 5 --s 750

(ซึ่งจะนำภาพที่สร้างเหล่านี้ไป ตัด ในส่วนของ จมูก และ ปาก เนื่องจาก Midjourney ที่ใช้แบบฟรีนั้น ไม่สามารถใช้สร้างภาพเพิ่มเติมได้อีก)



ภาพที่จมูกถูกตัดจากภาพการสร้างภาพคนด้วย Midjourney



ภาพที่ปากถูกตัดจากภาพการสร้างภาพคนด้วย Midjourney

2. การสร้างภาพเพิ่มเติมโดยใช้ Stylegan2-ADA-Pytorch

```

import os
import click
import re
import json
import tempfile
import torch
import dnnlib

from training import training_loop
from metrics import metric_main
from torch_utils import training_stats
from torch_utils import custom_ops

#-----

class UserError(Exception):
    pass

#-----

def setup_training_loop_kwargs():
    # General options (not included in desc).
    gpus = None, # Number of GPUs: <int>, default = 1 gpu
    snap = None, # Snapshot interval: <int>, default = 50 ticks
    metrics = None, # List of metric names: [], ['fid50k_full'] (default), ...
    seed = None, # Random seed: <int>, default = 0

    # Dataset.
    data = None, # Training dataset (required): <path>
    cond = None, # Train conditional model based on dataset labels: <bool>, default = False
    subset = None, # Train with only N images: <int>, default = all
    mirror = None, # Augment dataset with x-flips: <bool>, default = False

    # Base config.
    cfg = None, # Base config: 'auto' (default), 'stylegan2', 'paper256', 'paper512', 'paper1024', 'cifar'
    gamma = None, # Override R1 gamma: <float>
    kimg = None, # Override training duration: <int>
    batch = None, # Override batch size: <int>

    # Discriminator augmentation.
    aug = None, # Augmentation mode: 'ada' (default), 'noaug', 'fixed'
    p = None, # Specify p for 'fixed' (required): <float>
    target = None, # Override ADA target from 'ada': <float>, default = depends on aug
    augpipe = None, # Augmentation pipeline: 'blit', 'gaorn', 'color', 'filter', 'noise', 'cutout', 'bg', 'bgc' (default), ..., 'bgcfn'

    # Transfer learning.
    resume = None, # Load previous networks: 'noresume' (default), 'ffhq256', 'ffhq512', 'ffhq1024', 'celebahq256', 'lsundog256', <file>, <url>

```

ตัวอย่างภาษาไพธอนที่ใช้ในการเพิ่มจำนวนภาพ จะใช้คำสั่ง

python train.py --outdir=C:/Project/MachineLearningForArtists/MyProject --
 data=./datasets/stylegan_dataset.zip --gpus=1 --cfg=paper512 --mirror=1 --snap=10 --
 metrics=None ในการเพิ่มจำนวนภาพ

```

C:\WINDOWS\System32\Win... x + v - □ x
298 king 1261.6 time 3d 09h 59m sec/tick 992.9 sec/king 246.27 maintenance 8.0 cpuser 3.85 gpuset 8.19 augment 0.604
299 king 1265.6 time 3d 10h 15m sec/tick 993.3 sec/king 246.34 maintenance 0.3 cpuser 3.85 gpuset 8.19 augment 0.606
300 king 1269.7 time 3d 10h 32m sec/tick 993.2 sec/king 246.33 maintenance 0.3 cpuser 3.85 gpuset 8.19 augment 0.608
301 king 1273.7 time 3d 10h 49m sec/tick 988.9 sec/king 246.27 maintenance 13.1 cpuser 3.86 gpuset 8.22 augment 0.651
302 king 1277.7 time 3d 11h 05m sec/tick 993.2 sec/king 246.32 maintenance 0.0 cpuser 3.86 gpuset 8.19 augment 0.653
303 king 1281.8 time 3d 11h 22m sec/tick 993.3 sec/king 246.34 maintenance 0.3 cpuser 3.86 gpuset 8.25 augment 0.655
304 king 1285.8 time 3d 11h 38m sec/tick 993.1 sec/king 246.31 maintenance 0.3 cpuser 3.86 gpuset 8.27 augment 0.654
305 king 1289.8 time 3d 11h 55m sec/tick 982.3 sec/king 243.63 maintenance 0.3 cpuser 3.86 gpuset 8.22 augment 0.657
306 king 1293.9 time 3d 12h 11m sec/tick 993.3 sec/king 246.35 maintenance 0.0 cpuser 3.86 gpuset 8.18 augment 0.660
307 king 1297.9 time 3d 12h 28m sec/tick 993.5 sec/king 246.41 maintenance 0.3 cpuser 3.86 gpuset 8.28 augment 0.663
308 king 1301.9 time 3d 12h 44m sec/tick 993.5 sec/king 246.40 maintenance 0.3 cpuser 3.86 gpuset 8.24 augment 0.663
309 king 1306.0 time 3d 13h 01m sec/tick 988.8 sec/king 245.24 maintenance 0.3 cpuser 3.86 gpuset 8.28 augment 0.666
310 king 1310.0 time 3d 13h 18m sec/tick 993.2 sec/king 246.32 maintenance 0.0 cpuser 3.86 gpuset 8.23 augment 0.664
311 king 1314.0 time 3d 13h 34m sec/tick 993.4 sec/king 246.37 maintenance 15.3 cpuser 3.86 gpuset 8.21 augment 0.666
312 king 1318.0 time 3d 13h 51m sec/tick 993.1 sec/king 246.31 maintenance 0.3 cpuser 3.86 gpuset 8.17 augment 0.666
313 king 1322.1 time 3d 14h 07m sec/tick 988.5 sec/king 245.17 maintenance 0.3 cpuser 3.86 gpuset 8.23 augment 0.667
314 king 1326.1 time 3d 14h 24m sec/tick 993.3 sec/king 246.35 maintenance 0.0 cpuser 3.86 gpuset 8.20 augment 0.669
315 king 1329.1 time 3d 14h 40m sec/tick 993.5 sec/king 246.40 maintenance 0.3 cpuser 3.86 gpuset 8.31 augment 0.670
316 king 1332.2 time 3d 14h 57m sec/tick 993.8 sec/king 246.48 maintenance 0.3 cpuser 3.86 gpuset 8.26 augment 0.672
317 king 1335.2 time 3d 15h 14m sec/tick 989.2 sec/king 245.33 maintenance 0.3 cpuser 3.86 gpuset 8.25 augment 0.673
318 king 1338.2 time 3d 15h 30m sec/tick 993.4 sec/king 246.37 maintenance 0.0 cpuser 3.86 gpuset 8.26 augment 0.675
319 king 1341.3 time 3d 15h 47m sec/tick 993.4 sec/king 246.38 maintenance 0.3 cpuser 3.86 gpuset 8.27 augment 0.677
320 king 1344.3 time 3d 16h 03m sec/tick 993.6 sec/king 246.43 maintenance 0.3 cpuser 3.86 gpuset 8.21 augment 0.677
321 king 1347.3 time 3d 16h 20m sec/tick 982.5 sec/king 243.67 maintenance 15.2 cpuser 3.86 gpuset 8.22 augment 0.676
322 king 1350.4 time 3d 16h 36m sec/tick 993.6 sec/king 246.43 maintenance 0.0 cpuser 3.86 gpuset 8.26 augment 0.677
323 king 1353.4 time 3d 16h 53m sec/tick 993.6 sec/king 246.42 maintenance 0.3 cpuser 3.86 gpuset 8.21 augment 0.680
324 king 1356.4 time 3d 17h 10m sec/tick 993.7 sec/king 246.45 maintenance 0.3 cpuser 3.86 gpuset 8.30 augment 0.685
325 king 1359.5 time 3d 17h 26m sec/tick 991.9 sec/king 246.80 maintenance 0.3 cpuser 3.86 gpuset 8.23 augment 0.688
326 king 1362.5 time 3d 17h 43m sec/tick 1001.8 sec/king 248.47 maintenance 0.0 cpuser 3.86 gpuset 8.23 augment 0.692

```

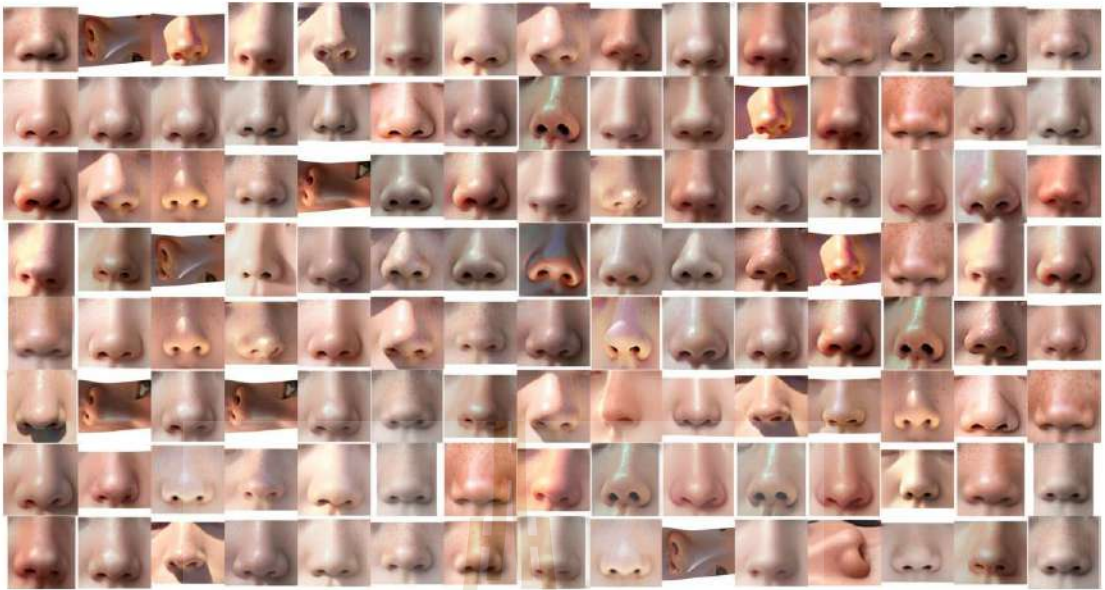
ตัวอย่างขณะที่คอมพิวเตอร์กำลังฝึกสอนเพื่อสร้างภาพเพิ่มเติม



ตัวอย่างภาพดวงตาที่สร้างเพิ่มเติม



ตัวอย่างภาพปากที่สร้างเพิ่มเติม



ตัวอย่างภาพจมูกที่สร้างเพิ่มเติม



3. การตรวจหาวัตถุ

```

import argparse
import math
import os
import random
import subprocess
import sys
import time
from copy import deepcopy
from datetime import datetime, timedelta
from pathlib import Path

try:
    import comet_ml # must be imported before torch (if installed)
except ImportError:
    comet_ml = None

import numpy as np
import torch
import torch.distributed as dist
import torch.nn as nn
import yaml
from torch.optim import lr_scheduler
from tqdm import tqdm

FILE = Path(__file__).resolve()
ROOT = FILE.parents[0] # YOLOv5 root directory
if str(ROOT) not in sys.path:
    sys.path.append(str(ROOT)) # add ROOT to PATH
ROOT = Path(os.path.relpath(ROOT, Path.cwd())) # relative

import val as validate # for end-of-epoch MAP
from models.experimental import attempt_load
from models.yolo import Model
from utils.autoanchor import check_anchors
from utils autobatch import check_train_batch_size
from utils.callbacks import Callbacks
from utils.dataloaders import create_dataloader
from utils.downloads import attempt_download, is_url
from utils.general import (LOGGER, TQDM_BAR_FORMAT, check_amp, check_dataset, check_file, check_git_info,
                           check_git_status, check_img_size, check_requirements, check_suffix, check_yaml, colorstr,
                           get_latest_run, increment_path, init_seeds, intersect_dicts, labels_to_class_weights,
                           labels_to_image_weights, methods, one_cycle, print_args, print_mutation, strip_optimizer,
                           yaml_save)
from utils.loggers import Loggers
from utils.loggers.comet.comet_utils import check_comet_resume
from utils.loss import ComputeLoss

```

ตัวอย่างภาษาไพธอนที่ใช้ในการเพิ่มจำนวนภาพ จะใช้คำสั่ง

```
python train.py --img 640 --batch 32 --epochs 500000 --patience 500000 --data
face.yaml --cfg models/yolov5s.yaml
```

	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1 [00:00<00:00, 1.00it/s]
	all	79	226	0.869	0.856	0.903	0.54	
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size		
66598/499999	11G	0.004318	0.001777	0.0004409	113	640: 100%	27/27 [00:29<00:00, 1.09s/it]	
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1 [00:01<00:00, 1.00s/it]
	all	79	226	0.869	0.856	0.904	0.541	
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size		
66599/499999	11G	0.00432	0.001906	0.0004393	123	640: 100%	27/27 [00:30<00:00, 1.12s/it]	
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1 [00:01<00:00, 1.05s/it]
	all	79	226	0.871	0.856	0.904	0.54	
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size		
66600/499999	11G	0.004174	0.001878	0.0004348	133	640: 100%	27/27 [00:27<00:00, 1.03s/it]	
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1 [00:00<00:00, 1.02it/s]
	all	79	226	0.871	0.856	0.904	0.537	
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size		
66601/499999	11G	0.004122	0.001798	0.000379	119	640: 100%	27/27 [00:29<00:00, 1.09s/it]	
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1 [00:01<00:00, 1.10s/it]
	all	79	226	0.872	0.856	0.904	0.539	
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size		
66602/499999	11G	0.004222	0.001853	0.0004396	120	640: 100%	27/27 [00:31<00:00, 1.16s/it]	
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1 [00:01<00:00, 1.02s/it]
	all	79	226	0.871	0.856	0.904	0.542	
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size		
66603/499999	11G	0.004344	0.001801	0.0004943	113	640: 100%	27/27 [00:26<00:00, 1.04it/s]	
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1 [00:01<00:00, 1.01s/it]
	all	79	226	0.869	0.856	0.903	0.541	
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size		
66604/499999	11G	0.004113	0.00175	0.0004243	149	640: 100%	27/27 [00:32<00:00, 1.22s/it]	
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1 [00:00<00:00, 1.02it/s]
	all	79	226	0.869	0.856	0.904	0.537	
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size		
66605/499999	11G	0.00421	0.001745	0.0003995	107	640: 100%	27/27 [00:31<00:00, 1.16s/it]	
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1 [00:01<00:00, 1.07s/it]
	all	79	226	0.866	0.856	0.904	0.541	
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size		
66606/499999	11G	0.004153	0.001773	0.0004215	137	640: 100%	27/27 [00:30<00:00, 1.12s/it]	
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	1/1 [00:01<00:00, 1.24s/it]
	all	79	226	0.872	0.856	0.904	0.538	
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size		
66607/499999	11G	0.004023	0.001606	0.0003646	220	640: 19%	5/27 [00:04<00:20, 1.05it/s]	

ตัวอย่างระหว่างที่คอมพิวเตอร์ทำการฝึกสอนการตรวจหาวัตถุ

มหาวิทยาลัยเทคโนโลยีสุรนารี



ภาคผนวก ข

บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

มหาวิทยาลัยเทคโนโลยีสุนาร์

ภาคผนวก ข

บทความทางวิชาการที่ได้นำเสนอในระหว่างการศึกษา

- จักรภัทร เวทศิลป์ และเจษฎา ตัณทนุช. (2566). การใช้ปัญญาประดิษฐ์พัฒนาภาพจำลองดวงตาของมนุษย์เพื่อการประยุกต์ใช้ในการสร้างเทมเพลตการฝึกสอนสำหรับการตรวจหาวัตถุ (Utilizing Artificial Intelligence to Develop Simulated Human Eye Images for Practical Application in Creating Training Templates for Object Detection). ในหนังสือประมวลบทความ (Proceedings) การประชุมวิชาการระดับชาติ ครั้งที่ 18 และการประชุมวิชาการระดับนานาชาติ ครั้งที่ 8 มหาวิทยาลัยศรีปทุม ออนไลน์ เรื่อง วิจัยและนวัตกรรมสู่การพัฒนาที่ยั่งยืน (The Proceedings of the 18th National and the 8th International Sripatum University Conference: Research and Innovations to Sustainable Development) (หน้า 2695-2705). 27 ตุลาคม 2566.

SPU

SRIPATUM
UNIVERSITY

มหาวิทยาลัยศรีปทุม

เกียรติบัตรฉบับนี้ให้ไว้เพื่อแสดงว่า

จักรภัทร เวทศิลป์, เจษฎา ตัณทนุช

ได้เข้าร่วมนำเสนอบทความ
ในการประชุมวิชาการระดับชาติ ครั้งที่ ๑๘ และการประชุมวิชาการระดับนานาชาติ ครั้งที่ ๘
มหาวิทยาลัยศรีปทุม ออนไลน์ ประจำปี ๒๕๖๖
เรื่อง วิจัยและนวัตกรรมสู่การพัฒนาที่ยั่งยืน



ขอให้ความสุข และความจำเริญตลอดไป

วันศุกร์ที่ ๒๗ ตุลาคม ๒๕๖๖

(ดร.รัชนิพร พุคยาภรณ์ พุกกะมาน)

อธิการบดี

ประวัติผู้เขียน

นายจักรภัทร เวทศิลป์ เกิดวันที่ 23 เดือน พฤศจิกายน พ.ศ. 2540 ณ จังหวัดขอนแก่น สำเร็จการศึกษาระดับมัธยมจากโรงเรียนสาธิตมหาวิทยาลัยขอนแก่น (ศึกษาศาสตร์) ถนนมะลิวัลย์ ตำบลในเมือง อำเภอเมืองขอนแก่น จังหวัดขอนแก่น 40002 ในปีการศึกษา 2558 และเข้ารับการศึกษ ต่อในระดับอุดมศึกษา ณ มหาวิทยาลัยเทคโนโลยีสุรนารี 111 ถนนมหาวิทยาลัย ตำบลสุรนารี อำเภอเมือง จังหวัดนครราชสีมา 30000 ปีการศึกษา 2560 จนสำเร็จการศึกษาระดับปริญญาตรีจากคณะ วิศวกรรมศาสตร์ สาขาวิชา วิศวกรรมอากาศยาน ณ มหาวิทยาลัยเทคโนโลยีสุรนารี วันที่ 17 เดือน กรกฎาคม พ.ศ. 2564

เข้ารับการศึกษต่อ ระดับปริญญาโท วันที่ 16 เดือน สิงหาคม พ.ศ. 2564 ในสาขาวิชา วิศวกรรมเครื่องกล และ ปี พ.ศ.2565 ได้ย้ายสาขาวิชา จาก วิศวกรรมเครื่องกล มาเป็น นวัตกรรม วิศวกรรมแพทย์ โดยระหว่างที่ศึกษาได้มีโอกาสทำงานวิจัย และ นำเสนอผลงานในงาน การประชุม วิชาการระดับชาติครั้งที่ 18 และการประชุมวิชาการระดับนานาชาติครั้งที่ 8 มหาวิทยาลัยศรีปทุม ประจำปี 2566 (SPUCON2023) ดังที่ปรากฏในภาคผนวก ข

มหาวิทยาลัยเทคโนโลยีสุรนารี