การใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันสำหรับสร้างโมเดลในการแท็กภาพอนิเมะ

อดิเทพ พรหมพา

งานนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการข้อมูล

คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

2567

การใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันสำหรับสร้างโมเดลในการแท็กภาพอนิเมะ

อดิเทพ พรหมพา

งานนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการข้อมูล
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
2567

Using Convolutional Neural Networks for Creating Models to Tag Anime Images

ADITHEP PHOMPHA

AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR MASTER DEGREE OF SCIENCE

IN DATA SCIENCE

FACULTY OF INFORMATICS

BURAPHA UNIVERSITY

2024

คณะกรรมการควบคุมงานนิพนธ์และคณะกรรมการสอบงานนิพนธ์ได้พิจารณางานนิพนธ์ของ อดิเทพ พรหมพา ฉบับนี้แล้ว เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล ของมหาวิทยาลัยบูรพาได้

คณะกรรมการควบคุมงานนิพนธ์
อาจารย์ที่ปรึกษาหลัก

.....
(รองศาสตราจารย์ ดร.สุนิสา ริมเจริญ)

คณะกรรมการสอบงานนิพนธ์

..... ประธาน
(ผู้ช่วยศาสตราจารย์ ดร.อุรีรัฐ สุขสวัสดิ์ชน)

..... กรรมการ
(ดร.คนึงนิจ กุโบลา)

..... กรรมการ
(รองศาสตราจารย์ ดร.สุนิสา ริมเจริญ)

..... คณบดีคณะวิทยาการสารสนเทศ
(ผู้ช่วยศาสตราจารย์ ภูสิต กุลเกษม)
วันที่.....เดือน.....พ.ศ.....

บัณฑิตวิทยาลัย มหาวิทยาลัยบูรพา อนุมัติให้รับงานนิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล ของมหาวิทยาลัยบูรพา

..... คณบดีบัณฑิตวิทยาลัย
(รองศาสตราจารย์ ดร.วิทวัส แจ้งเอี่ยม)
วันที่.....เดือน.....พ.ศ.....

65910099: สาขาวิชา: วิทยาการข้อมูล; วท.ม. (วิทยาการข้อมูล)
คำสำคัญ: อนิเมะ, แท็ก, CNN, GCN, ResNET, ResNeXT, EfficientNet, Multi-label

อดิเทพ พรหมพา : การใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันสำหรับสร้างโมเดลในการแท็กภาพอนิเมะ. (Using Convolutional Neural Networks for Creating Models to Tag Anime Images) คณะกรรมการควบคุมงานนิพนธ์: สุนิสา ริมเจริญ, วศ.ด. ปี พ.ศ. 2567.

ในปัจจุบันมีเว็บไซต์ซึ่งเป็นแหล่งรวมภาพผลงานอนิเมะให้นักวาดภาพหาแรงบันดาลใจในการสร้างผลงานของตน แต่แท็กของภาพอนิมะไม่มีรูปแบบที่ชัดเจน บางภาพอนิเมะถูกกำหนดแท็กไม่เพียงพอ ทำให้นักวาดภาพที่เข้ามาหาแรงบันดาลใจมีโอกาสค้นหาเจอภาพที่ต้องการน้อยลง พวกเขาอาจสูญเสียโอกาสในการสร้างผลงานให้ออกมาดีที่สุด ผู้วิจัยจึงนำเสนอการนำโมเดล ML-GCN ซึ่งเป็นโมเดลสำหรับการทำ Muti-label เพื่อช่วยในการกำหนดแท็กของภาพอนิเมะให้มีความถูกต้องและครบถ้วนมากขึ้น โครงสร้างหลักของโมเดลนี้ประกอบด้วยโครงข่ายคอนโวลูชันแบบกราฟ (Graph Convolutional Networks: GCN) และโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network: CNN) ในงานวิจัยนี้ผู้วิจัยเปรียบเทียบอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชัน 3 แบบ ได้แก่ ResNET, ResNeXT และ EfficientNet เพื่อวิเคราะห์ว่าอัลกอริทึมใดจำแนกแท็กของภาพอนิเมะได้ถูกต้องมากกว่า จากผลการทดลองพบว่า ResNeXT มีค่าความแม่นยำเฉลี่ย (mAP) สูงกว่า ResNET และ EfficientNet ทำให้ ResNeXT เหมาะนำมาใช้ร่วมกับโมเดล ML-GCN ในการจำแนกแท็กของภาพอนิเมะ

65910099: MAJOR: DATA SCIENCE; M.Sc. (DATA SCIENCE)

KEYWORDS: Anime, Tag, CNN, GCN, ResNET, ResNeXT, EfficientNet, Multi-label

ADITHEP PHOMPHA : USING CONVOLUTIONAL NEURAL NETWORKS FOR CREATING MODELS TO TAG ANIME IMAGES. ADVISORY COMMITTEE: SUNISA RIMCHAROEN, Ph.D. 2024.

Recently, there are websites that provide anime artworks for artists who search for inspiration to create their own works. Unfortunately, tags of anime images lack clear structures, and some images are insufficiently tagged. This reduces the chances of the artists to find the specific images they need and they might lose their chances of producing their best work. To solve this problem, we present using the ML-GCN model, a multi-label classification, to improve the correctness and completeness of anime image tagging. The core structure of this model consists of Graph Convolutional Networks (GCN) and Convolutional Neural Networks (CNN). In this study, we compared three convolutional neural network algorithms (ResNET, ResNeXT, and EfficientNet) to determine which algorithm more accurately classifies anime image tags. The experimental results show that ResNeXT yields a higher mean average precision (mAP) than ResNET and EfficientNet, indicating that ResNeXT is better suited for apply with the ML-GCN model in classifying anime image tags.

กิตติกรรมประกาศ

งานวิจัยนี้สามารถดำเนินการจนประสบความสำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความอนุเคราะห์และสนับสนุนเป็นอย่างดียิ่งจาก รศ. ดร. สุนิสา ริมเจริญ ผู้เป็นอาจารย์ที่ปรึกษาที่กรุณาให้คำปรึกษา ความรู้ ข้อแนะนำ และปรับปรุงแก้ไขข้อบกพร่องต่าง ๆ จนกระทั่งงานวิจัยครั้งนี้สำเร็จเรียบร้อยด้วยดี ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบคุณบุคคลท่านอื่นที่ไม่ได้เอ่ยนาม ที่คอยเป็นกำลังใจ ให้ความช่วยเหลือ คอยให้คำปรึกษาแก่ผู้วิจัยตลอดระยะเวลาการดำเนินงานวิจัยในครั้งนี้

สุดท้ายนี้ผู้วิจัยหวังว่างานวิจัยฉบับนี้คงเป็นประโยชน์สำหรับผู้ที่สนใจศึกษาต่อไป

อดิเทพ พรหมพา

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1
บทนำ

ในปัจจุบันเทคโนโลยีในการวิเคราะห์ข้อมูลมีความก้าวหน้ามากขึ้น เทคนิคการเรียนรู้เชิงลึก (Deep Learning) ถูกใช้เพื่อสอนให้คอมพิวเตอร์วิเคราะห์และเรียนรู้ชุดข้อมูลเพื่อให้คอมพิวเตอร์สามารถตัดสินใจและทำงานแทนมนุษย์ได้ เทคนิคนี้สามารถนำมาประยุกต์เพื่อสอนให้คอมพิวเตอร์ช่วยกำหนดแท็กให้แก่ภาพอนิเมะให้มีความถูกต้องและครบถ้วนมากขึ้น โครงข่ายประสาทเทียมแบบคอนโวลูชันเป็นหนึ่งในเทคนิคการเรียนรู้เชิงลึกที่ใช้สำหรับการเรียนรู้ข้อมูลประเภทรูปภาพโดยเฉพาะ ผู้วิจัยศึกษาโมเดล ML-GCN ซึ่งเป็นโมเดลสำหรับเรียนรู้รูปภาพที่ประกอบด้วยหลายแท็กเพื่อใช้กำหนดแท็กของภาพอนิเมะ โครงสร้างหลักของโมเดลนี้ประกอบด้วยโครงข่ายประสาทเทียมแบบคอนโวลูชันและโครงข่ายคอนโวลูชันแบบกราฟ โดยเนื้อหาในบทนี้จะกล่าวเกี่ยวกับที่มาและความสำคัญของงานวิจัย วัตถุประสงค์ของงานวิจัย ประโยชน์ของงานวิจัย ขอบเขตและรายละเอียดของงานวิจัย เครื่องมือที่ใช้ในงานวิจัย ขั้นตอนในการดำเนินงานวิจัย และแผนการดำเนินงานวิจัย

1.1 ที่มาและความสำคัญของงานวิจัย

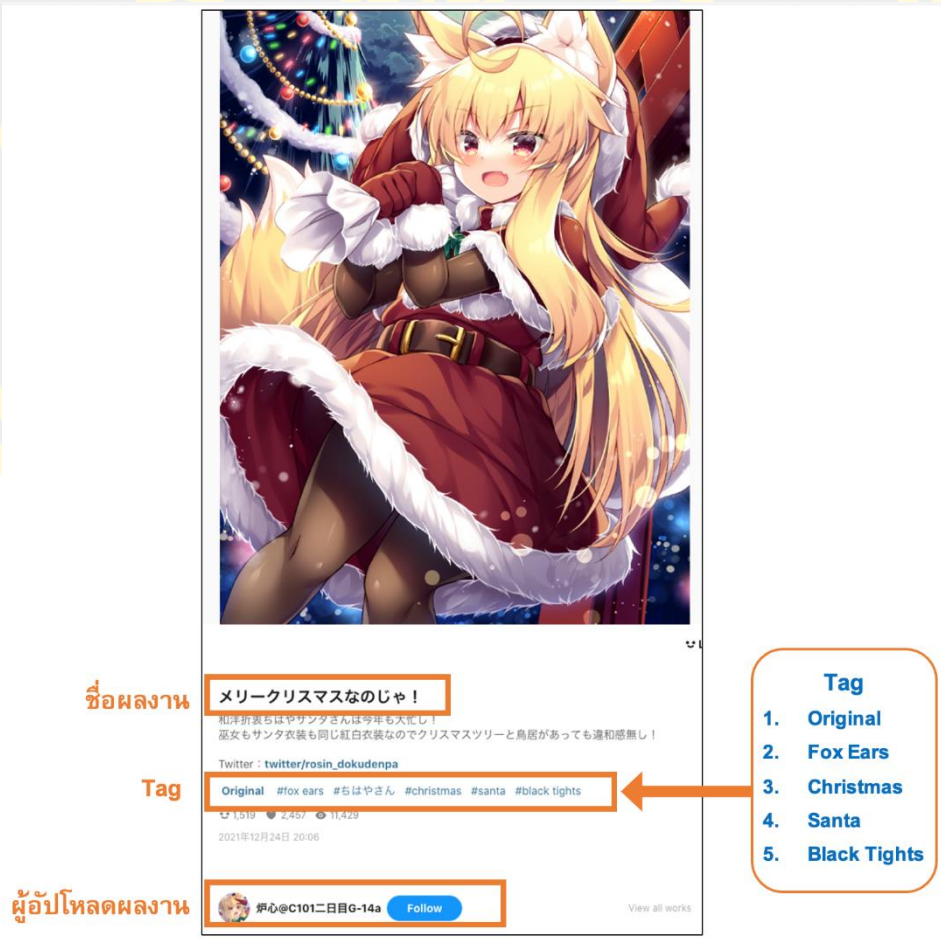
ภาพอนิเมะเป็นรูปแบบการวาดการ์ตูนที่ถูกแพร่หลายมาจากประเทศญี่ปุ่น บนอินเตอร์เน็ตมีเว็บไซต์สาธารณะที่เปิดให้ผู้ใช้ทั่วไปสามารถนำภาพอนิเมะที่ตนวาดมาเผยแพร่เพื่อเป็นแรงบันดาลใจให้แก่ผู้วาดภาพอนิเมะ โดยเว็บไซต์สาธารณะซึ่งเป็นแหล่งรวมภาพอนิเมะและเป็นที่นิยม คือ Pixiv.net นอกจากนี้การจัดอันดับเว็บไซต์ด้านทัศนศิลป์และการออกแบบ (Visual Arts and Design) ที่ผู้คนทั่วโลกเข้าใช้มากที่สุดในปี 2023 แสดงให้เห็นว่า เว็บไซต์ที่ผู้คนทั่วโลกเข้าใช้มากที่สุดในปี 2023 คือ Pixiv.net ตามด้วย Deviantart.com และ Behance.net ตามลำดับดังภาพที่ 1-1

Rank	Website	Category
1	pixiv.net	Arts & Entertainment > Visual Arts and Design
2	deviantart.com	Arts & Entertainment > Visual Arts and Design
3	behance.net	Arts & Entertainment > Visual Arts and Design

ภาพที่ 1-1 การจัดอันดับด้านทัศนศิลป์และการออกแบบ ณ เดือนธันวาคม 2023
อ้างอิง: https://www.similarweb.com/top-websites/arts-and-entertainment/visual-arts-and-design/

เว็บไซต์ Pixiv.net เป็นแหล่งอัปโหลดผลงานภาพอนิเมะและเป็นแหล่งค้นหาแรงบันดาลใจสำหรับผู้วาดภาพอนิเมะ เว็บไซต์นี้เปิดให้บุคคลทั่วไปสามารถเข้าลงทะเบียนใช้งานได้ การค้นหาภาพผลงานบนเว็บไซต์ Pixiv.net มีลักษณะการค้นหา 3 รูปแบบ ได้แก่ การค้นหาจากชื่อผลงาน การค้นหาจากชื่อผู้อัปโหลดผลงาน และการค้นหาจากแท็กต่าง ๆ (Tag) ของภาพอนิเมะ โดยแท็กเหล่านี้ถูกกำหนดให้แก่ภาพอนิเมะในแต่ละภาพเพื่อบ่งบอกลักษณะของภาพนั้นและใช้เป็นคีย์เวิร์ดเพื่อให้ผู้อื่นสามารถค้นหาภาพอนิเมะตรงตามลักษณะที่ต้องการได้ง่ายขึ้น

ภาพที่ 1-2 คือ ตัวอย่างผลงานภาพอนิเมะบนเว็บไซต์ Pixiv.net ภาพผลงานนี้มีแท็กต่าง ๆ กำกับ ได้แก่ "Original" (ภาพนี้เป็นต้นฉบับ) "Fox Ears" (ตัวละครบนภาพมีหูสุนัขจิ้งจอก) "Christmas" (ภาพนี้เป็นธีมวันคริสต์มาส) "Santa" (ตัวละครบนภาพสวมชุดซานต้าคลอส) และ "Black Tights" (ตัวละครสวมถุงน่องสีดำ)



ภาพที่ 1-2 ตัวอย่างผลงานภาพอนิเมะบนเว็บไซต์ Pixiv.net

อ้างอิง: https://www.pixiv.net/en/artworks/94987223

การกำหนดแท็กให้แก่ภาพผลงานถูกดำเนินการในขั้นตอนของการอัปโหลดภาพผลงานบนเว็บไซต์ บางเว็บไซต์ไม่มีข้อกำหนดว่าแต่ละภาพต้องกำหนดแท็กด้านใดบ้าง เช่น ธีมของภาพผลงาน ลักษณะของตัวละครบนภาพผลงาน ภาพฉากหลัง เป็นต้น ผู้อัปโหลดผลงานสามารถกำหนดเองว่าระบุแท็กด้านใดบนภาพรวมทั้งสามารถสร้างแท็กใหม่ขึ้นเองได้ ทำให้บางผลงานถูกกำหนดแท็กไม่เพียงพอ เกิดแท็กใหม่ที่มีหลายความหมายในแท็กเดียวและเกิดแท็กใหม่ที่มีความหมายซ้ำกับแท็กที่มีอยู่ในระบบ อาจส่งผลให้นักวาดภาพที่เข้ามาในเว็บไซต์เพื่อหาแรงบันดาลใจในการวาดภาพอนิเมะมีโอกาสค้นหาเจอภาพที่ต้องการน้อยลงหรืออาจค้นหาไม่เจอ พวกเขาอาจสูญเสียโอกาสในการสร้างผลงานให้ออกมาดีที่สุด โดยเฉพาะผู้ที่ลงแข่งขันประกวดวาดภาพหรือผู้ที่ทำงานเกี่ยวกับการสร้างสรรค์ผลงานศิลปะเป็นอาชีพ พวกเขาอาจเสียโอกาสการสร้างชื่อเสียงหรือโอกาสการได้รับงาน สำหรับนักวาดภาพการหาแรงบันดาลใจและคิดไอเดียเป็นขั้นตอนที่ยาก ดังนั้นหากสามารถค้นหาภาพผลงานที่ตรงใจได้ตั้งแต่ต้นและเป็นจำนวนมากจะช่วยให้สร้างสรรค์ผลงานเร็วขึ้นและสมบูรณ์แบบมากขึ้น

ตารางที่ 1-1 แสดงตัวอย่างปัญหาการอัปโหลดภาพอนิเมะไปยังเว็บไซต์ที่เผยแพร่ภาพอนิเมะ ไม่มีข้อกำหนดว่าแต่ละภาพอนิเมะต้องกำหนดแท็กด้านใดบ้าง จึงทำให้เกิดปัญหาเกี่ยวกับการค้นหารูปภาพดังนี้

- หากผู้ที่ต้องการค้นหาภาพอนิเมะใช้คีย์เวิร์ดในการค้นหาภาพว่า “Fox Ears (หูสุนัขจิ้งจอก)” เขาจะเจอเพียงภาพในลำดับที่ 1 เพียงภาพเดียว แม้ว่าภาพในลำดับที่ 1 และ 2 มีสุนัขจิ้งจอกเหมือนกัน
- ภาพในลำดับที่ 2 ผู้อัปโหลดภาพสร้างแท็ก “Cat Ears Maid" ขึ้นเอง ซึ่งแท็กนี้มีการระบุคุณลักษณะหลายสิ่งในแท็กเดียว แต่ควรแยกเป็น 2 แท็ก คือ "Cat Ears" และ "Maid" เพื่อให้สามารถค้นหาภาพอนิเมะได้ง่ายขึ้น

ตารางที่ 1-1 การเปรียบเทียบแท็กของภาพอนิเมะจากเว็บไซต์ Pixiv

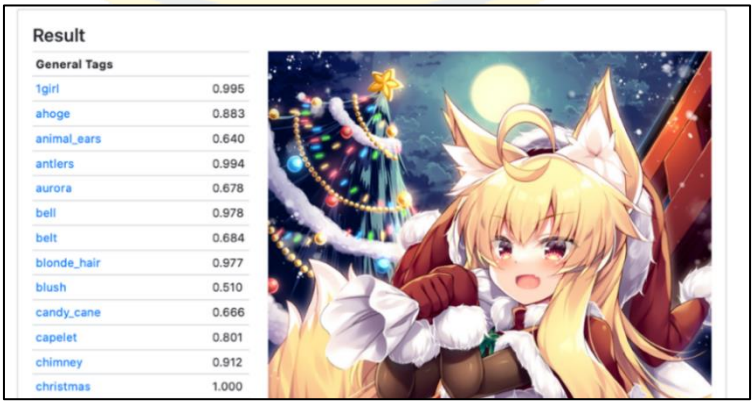
ลำดับที่	ภาพ	แท็ก
1.	 https://www.pixiv.net/en/artworks/94987223	1. Original 2. Fox Ears 3. Christmas 4. Santa 5. Black Tights

ตารางที่ 1 การเปรียบเทียบแท็กของภาพอนิเมะจากเว็บไซต์ Pixiv (ต่อ)

ลำดับที่	ภาพ	แท็ก
2.	 https://www.pixiv.net/en/artworks/110725274	1. Original 1. Skeb (ชื่อเว็บไซต์รับจ้างวาดรูป) 2. Girl 3. Tail
3.	 https://www.pixiv.net/en/artworks/102630131	1. Original 2. Cat Ears Maid 3. Original 1000+ Bookmarks

ปัญหาการกำหนดแท็กมีสาเหตุจากผู้อัปโหลดกำหนดแท็กให้แก่รูปภาพด้วยตนเองโดยไม่มีรูปแบบที่ชัดเจน ทำให้เกิดปัญหาการกำหนดแท็กไม่เพียงพอและปัญหาจากการสร้างแท็กใหม่แต่การบังคับให้ผู้อัปโหลดภาพกำหนดแท็กด้านต่าง ๆ ให้ครบถ้วนจะทำให้เกิดความไม่สะดวกสบายในการอัปโหลดภาพผลงาน ผู้วิจัยจึงเสนอว่า ในขั้นตอนที่ผู้อัปโหลดภาพ ระบบควรแนะนำแท็กให้แก่ภาพผลงานเพื่อไม่เพิ่มภาระให้ผู้อัปโหลดภาพและเพิ่มโอกาสในการหาภาพเหล่านี้พบมากขึ้น ผู้วิจัยต้องการสร้างโมเดลในการแท็กภาพอนิเมะเพื่อสร้างแท็กพื้นฐานให้แก่ภาพอนิเมะ

ในส่วนของการสร้างแท็ก ผู้วิจัยศึกษาโมเดล DeepDanbooru ของ KichangKim ซึ่งเป็นโมเดลสำหรับแท็กภาพอนิเมะ ผู้วิจัยทดสอบสร้างแท็กจากภาพอนิเมะโดยใช้โมเดลดังกล่าวดังภาพที่ 1-3 ซึ่งแท็กที่ถูกสร้างออกมามีจำนวน 71 แท็ก



ภาพที่ 1-3 การทดสอบการสร้างแท็กโดยใช้โมเดล DeepDanbooru ของ KichangKim

รูปแบบแท็กที่สร้างจากโมเดล DeepDanbooru หลัก ๆ แบ่งออกเป็น 13 หมวดหมู่ ดังต่อไปนี้

1) จำนวนตัวละครบนภาพ
2) บทบาทตัวละคร เผ่าพันธุ์ตัวละคร และประเภทตัวละคร
3) เครื่องแต่งกายตัวละครโดยละเอียด
4) ส่วนของร่างกายตัวละครและสิ่งที่ผลิตจากร่างกาย เช่น เลือด เหงื่อ น้ำตา
5) สีตา
6) สีเส้นผม
7) ทรงผม
8) สีหน้าและท่าทางตัวละคร
9) ฉากหลัง เช่น ลวดลายตกแต่ง สถานที่ ธรรมชาติ ดอกไม้ ธาตุต่าง ๆ (ดิน น้ำ ลม ไฟ)
10) สิ่งของบนภาพ เช่น อาหาร เครื่องดื่ม เฟอร์นิเจอร์ เครื่องมือ อุปกรณ์ อาวุธ ยานพาหนะ ฯลฯ
11) ธีม สไตล์ อีเว็นต์ และงานเทศกาล
12) ประเภทของภาพผลงาน
13) เรทความปลอดภัย

หากเปรียบเทียบแท็กที่สร้างโดยโมเดล DeepDanbooru กับการกำหนดแท็กบนเว็บ Pixiv.net พบว่า แท็กที่สร้างจากโมเดลมีรายละเอียดและมีจำนวนมากเกินไป บางภาพมีแท็กที่ถูกสร้างออกมามากถึง 70 แท็ก ทำให้ผู้วาดภาพอนิเมะที่เข้ามาหาแรงบันดาลใจยากต่อการพิจารณาลักษณะของภาพที่ตนต้องการ

ผู้วิจัยจึงเลือกใช้แท็กเพียง 8 หมวดหมู่จากทั้งหมด 13 หมวดหมู่ ได้แก่ บทบาทตัวละคร เครื่องแต่งกายตัวละคร ส่วนของร่างกายและสิ่งที่ผลิตจากร่างกาย ทรงผม สีหน้าและท่าทางตัวละคร ฉากหลัง สิ่งของบนภาพ และธีม นอกจากนี้ผู้วิจัยเสนอว่าควรมีแท็กเพิ่มอีก 2 หมวดหมู่ซึ่งไม่มีในโมเดล DeepDanbooru คือ แท็กที่ระบุโทนสีโดยรวมของภาพ เพื่อให้ผู้ที่เข้ามาหาแรงบันดาลใจสามารถค้นหาภาพที่ใช้โทนสีในการสะท้อนอารมณ์ด้านต่าง ๆ ได้ง่ายขึ้น และแท็กลักษณะการเน้น เพื่อให้ผู้ที่เข้ามาหาแรงบันดาลใจสามารถเลือกค้นหาภาพที่เน้นตัวละครหรือเน้นฉากหลังเป็นหลักได้

แท็กที่ผู้วิจัยใช้จึงมีทั้งหมด 10 หมวดหมู่ ซึ่งประกอบด้วยหมวดหมู่ที่มีในโมเดล DeepDanbooru จำนวน 8 หมวดหมู่ และหมวดหมู่ที่ผู้วิจัยเพิ่มใหม่จำนวน 2 หมวดหมู่

ดังตารางที่ 1-2 ผู้วิจัยเรียกแท็ก 10 หมวดหมู่นี้ว่า “แท็กพื้นฐาน” โดยสาเหตุที่ไม่ใช้แท็ก 5 หมวดหมู่ที่เหลือและบางแท็กสร้างแค่แบบคร่าว ๆ เพราะจากการตรวจสอบแท็กของภาพอนิเมะบนเว็บ Pixiv.net พบว่าภาพอนิเมะมีแท็กจำนวนไม่มาก แต่แท็กที่ถูกสร้างโดยโมเดล DeepDanbooru มีจำนวนมากเกินไป ดังนั้นผู้วิจัยจึงเลือกเฉพาะหมวดหมู่แท็กที่จำเป็นและไม่ใช้แท็กที่ใช้อธิบายรายละเอียดเสริมของภาพ นอกจากนี้การมีแท็กจำนวนน้อยจะช่วยลดระยะเวลาที่ระบบใช้ค้นหาภาพ

ตารางที่ 1-2 การเปรียบเทียบแท็กของโมเดล DeepDanbooru กับ แท็กพื้นฐานที่ผู้วิจัยเลือกใช้

ลำดับที่	**หมวดหมู่**	**แท็กจากโมเดล DeepDanbooru**	**แท็กพื้นฐาน**
1.	จำนวนตัวละครบนภาพ	✓	
2.	บทบาทตัวละคร	✓	✓
3.	เครื่องแต่งกายตัวละคร	✓ (โดยละเอียด)	✓ (คร่าว ๆ)
4.	ส่วนของร่างกายและสิ่งที่ผลิตจากร่างกาย	✓	✓
5.	สีตา	✓	
6.	สีเส้นผม	✓	
7.	ทรงผม	✓	✓
8.	สีหน้าและท่าทางตัวละคร	✓	✓
9.	ฉากหลัง	✓	✓
10.	สิ่งของบนภาพ	✓ (โดยละเอียด)	✓ (คร่าว ๆ)
11.	ธีม	✓	✓
12.	ประเภทของภาพผลงาน	✓	
13.	เรทความปลอดภัย	✓	
14.	ลักษณะการเน้น		✓
15.	โทนสีโดยรวม		✓

ทั้งนี้ “โดยละเอียด” และ “คร่าว ๆ” ของหมวดหมู่เครื่องแต่งกายตัวละครและหมวดหมู่สิ่งของบนภาพบนตารางที่ 1-2 มีความหมายดังต่อไปนี้

- หมวดหมู่เครื่องแต่งกายตัวละครคร่าว ๆ ประกอบด้วย แท็กที่บ่งบอกเครื่องแต่งกายที่ตัวละครใส่แบบภาพรวมหรือประเภทชุดที่ใส่ เช่น ชุดพยาบาล เครื่องแบบ ชุดเกราะ ชุดว่ายน้ำ ชุดราตรี เป็นต้น

- หมวดหมู่เครื่องแต่งกายตัวละครโดยละเอียด ประกอบด้วย แท็กที่บ่งบอกเครื่องแต่งกายที่ตัวละครใส่ทั้งแบบภาพรวมและแบบเจาะจงเป็นชิ้น เช่น เสื้อเชิ้ต เสื้อแจ็คเก็ต กระโปงขาสั้น ถุงเท้า หมวก เข็มกลัด แหวน สร้อยคอ เป็นต้น
- หมวดหมู่สิ่งของบนภาพคร่าว ๆ ประกอบด้วย แท็กที่บ่งบอกวัตถุหรืออุปกรณ์ที่ตัวละครถืออยู่หรือมีบทบาทในภาพมาก เช่น อาวุธ หนังสือ หุ่นยนตร์ โทรศัพท์ อาหาร ขนม ยานพาหนะ เป็นต้น
- หมวดหมู่สิ่งของบนภาพโดยละเอียด ประกอบด้วย แท็กที่บ่งบอกวัตถุหรืออุปกรณ์ที่อยู่บนภาพทั้งหมด เช่น โต๊ะ เก้าอี้ กระถางต้นไม้ โคมไฟ พัดลม เฟอร์นิเจอร์ เป็นต้น

ผู้วิจัยไม่ใช้แท็กจำนวนตัวละครบนภาพเพราะผู้วิจัยมีจุดประสงค์เพื่อใช้โมเดลเพื่อแท็กภาพอนิเมะที่มีตัวละครเพียง 1 คนเท่านั้น ผู้วิจัยไม่ใช้แท็กที่เกี่ยวกับสีตาและสีเส้นผมเพราะเป็นเพียงรายละเอียดประกอบของภาพ ผู้วิจัยไม่ใช้แท็กประเภทของภาพผลงานเพราะผู้วิจัยใช้ภาพอนิเมะที่เป็นภาพวาดแบบปกติเท่านั้นจึงไม่จำเป็นต้องระบุประเภทของภาพผลงาน ไม่ใช้ภาพอนิเมะที่ต่างจากปกติ เช่น มังงะ ภาพสเก็ต ภาพแสดงสีหน้าตัวละครหลายหน้า ภาพออกแบบ รูปถ่าย และภาพซ้อนแบบซูม ผู้วิจัยไม่ใช้แท็กเรทความปลอดภัยเพราะผู้วิจัยไม่ใช่ภาพอนิเมะที่มีเนื้อหาสำหรับผู้ใหญ่

โมเดลที่ผู้วิจัยประยุกต์ใช้ในงานวิจัยนี้เพื่อแท็กภาพอนิเมะ คือ โมเดล ML-GCN ซึ่งเป็นโมเดลที่เปิดให้สามารถดาวน์โหลดได้ฟรีบนเว็บไซต์ Github โมเดลนี้ถูกเผยแพร่โดย Zhao-Min Chen และคณะ(Z.-M. Chen, Wei, X.-S., Wang, P., & Guo, Y., 2019) โมเดลนี้ใช้เทคนิคการเรียนรู้เชิงลึกเพื่อสอนให้คอมพิวเตอร์จดจำและจำแนกลักษณะต่าง ๆ ของรูปภาพที่ประกอบด้วยมากกว่า 1 แท็ก

ผู้วิจัยเลือกใช้โมเดล ML-GCN เพราะมีแนวคิดคล้ายกับโมเดลแท็กภาพอนิเมะในงานวิจัยของ Pengfei Deng และคณะ(P. Deng, Ren, J., Lv, S., Feng, J., & Kang, H., 2020) และงานวิจัยของ Ziwen Lan และคณะ (Z. Lan, Maeda, K., Ogawa, T., & Haseyama, M., 2023) ซึ่งทั้ง 2 งานวิจัยนี้ใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันและโครงข่ายคอนโวลูชันแบบกราฟเป็นส่วนประกอบในโมเดล ส่วนโมเดล DeepDanbooru ใช้เพียงโครงข่ายประสาทเทียมแบบคอนโวลูชันไม่มีโครงข่ายคอนโวลูชันแบบกราฟ

โมเดล ML-GCN ประกอบด้วยเทคนิคการเรียนรู้เชิงลึก 2 เทคนิค ได้แก่ โครงข่ายคอนโวลูชันแบบกราฟและโครงข่ายประสาทเทียมแบบคอนโวลูชัน โดยผู้วิจัยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน 3 แบบ ได้แก่ ResNET (K. He, Zhang, X., Ren, S., & Sun, J. , 2016)

ซึ่งเป็นอัลกอริทึมที่ใช้ในโมเดล ML-GCN และใช้ ResNeXT (S. Xie, Girshick, R., Tu, Z., He, K., & Dollar, P., 2017) กับ EfficientNet (M. Tan, & Le, Q. V. , 2020) ซึ่งเป็นอัลกอริทึมโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ใหม่กว่า ResNET เพื่อวิเคราะห์ว่าอัลกอริทึมใดให้ผลการทำนายแท็กของภาพอนิเมะได้ถูกต้องมากกว่า

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อสร้างโมเดลในการแท็กภาพอนิเมะโดยสร้างแท็กพื้นฐานให้แก่ภาพอนิเมะ
2. เพื่อวิเคราะห์และเปรียบเทียบอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชัน 3 แบบ ได้แก่ ResNET, ResNeXT และ EfficientNet ว่าอัลกอริทึมใดจำแนกแท็กของภาพอนิเมะร่วมกับโมเดล ML-GCN ได้ถูกต้องมากกว่า

1.3 ประโยชน์ของงานวิจัย

1. ช่วยให้ภาพผลงานภาพอนิเมะถูกกำหนดแท็กพื้นฐานโดยอัตโนมัติ
2. ลดปัญหาการสร้างแท็กใหม่ที่มีหลายความหมายหรือความหมายซ้ำกับแท็กอื่นที่มีอยู่ในระบบ
3. ช่วยเพิ่มโอกาสให้ผู้ที่ต้องการหาแรงบันดาลใจในการวาดภาพอนิเมะสามารถหาภาพที่ตรงใจพบมากขึ้น

1.4 ขอบเขตและรายละเอียดของงานวิจัย

1. ผู้วิจัยใช้ข้อมูลรูปภาพอนิเมะ danbooru2021 ซึ่งสามารถดาวน์โหลดข้อมูลได้ฟรีบนเว็บไซต์ Kaggle เป็นข้อมูลสอน (Training Data) และข้อมูลทดสอบ (Testing Data)
2. ผู้วิจัยประยุกต์ใช้โมเดล ML-GCN เพื่อแท็กภาพอนิเมะ
3. ผู้วิจัยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน 3 แบบร่วมกับโมเดล ML-GCN ได้แก่ ResNET, ResNeXT และ EfficientNet
4. โมเดลที่พัฒนาใช้สร้างแท็กให้แก่ภาพอนิเมะที่เป็นภาพวาดแบบปกติ ไม่ใช้ภาพอนิเมะที่ต่างจากปกติ เช่น มังงะ ภาพสเก็ต ภาพแสดงสีหน้าตัวละครหลายหน้า ภาพออกแบบ รูปถ่าย และภาพซ้อนแบบซูม
5. โมเดลที่พัฒนาใช้สำหรับสร้างแท็กให้กับภาพสีและขาวดำ
6. โมเดลที่พัฒนาใช้สำหรับภาพอนิเมะที่มีตัวละครหลักเป็นมนุษย์ชายหรือหญิง 1 คน เท่านั้น

7. โมเดลที่พัฒนาใช้สำหรับสร้างแท็กพื้นฐานเท่านั้น โดยแท็กพื้นฐาน คือ แท็กที่จำเป็นต่อการอธิบายภาพ และไม่ใช่แท็กที่อธิบายรายละเอียดเสริมของภาพ ซึ่งแท็กพื้นฐานที่ผู้วิจัยใช้มี 10 หมวดหมู่ ได้แก่ บทบาทตัวละคร เครื่องแต่งกายตัวละคร(คร่าว ๆ) ส่วนของร่างกายและสิ่งที่ผลิตจากร่างกาย ทรงผม สีหน้าและท่าทางตัวละคร ฉากหลัง สิ่งของบนภาพ(คร่าว ๆ) ธีม ลักษณะการเน้น และโทนสีโดยรวม
8. ผู้วิจัยสร้างแท็ก 2 หมวดหมู่เพิ่มเติม คือ แท็กลักษณะการเน้น และแท็กโทนสีโดยรวม

1.5 เครื่องมือที่ใช้ในงานวิจัย

1.5.1 ซอฟต์แวร์ที่ใช้ในการพัฒนา

ผู้วิจัยใช้โปรแกรม Jupyter Notebook และภาษา Python ช่วยตรวจสอบความถูกต้องและกลั่นกรองข้อมูลภาพอนิเมะ ส่วนขั้นตอนการรันโมเดลผู้วิจัยเปลี่ยนจาก Jupyter Notebook เป็น Google Colab Pro+ เพราะต้องใช้การประมวลผลจำนวนมากและรวดเร็ว นอกจากนี้ Google Colab Pro+ สามารถรันโมเดลในขณะที่ปิดบราวเซอร์ได้ โดย Runtime Type ของ Google Colab Pro+ ผู้วิจัยใช้เป็น A100 GPU ผู้วิจัยใช้ Google Colab Pro+ แค่ในขั้นตอนรันโมเดลเพราะ Google Colab Pro+ มีค่าใช้จ่ายสูง

1.5.2 ไลบรารี่ที่ใช้ในการพัฒนา

โมเดลที่ผู้วิจัยใช้ประกอบด้วยไลบรารีหลัก ดังต่อไปนี้ python 3.10.12, numpy 1.26.4, torch 2.4.1, torchnet 0.0.4, torchvision 0.19.1, PIL 10.4.0 และ tqdm 4.66.5

1.6 ขั้นตอนในการดำเนินงานวิจัย

1. ศึกษาข้อมูลและงานวิจัย
2. จัดหาข้อมูลและเตรียมเครื่องมือต่าง ๆ
3. เตรียมข้อมูล
4. เตรียมโมเดล
5. รันและทดสอบโมเดล
6. วิเคราะห์ผลลัพธ์และแก้ไขจนกว่าได้รับผลลัพธ์ที่น่าพึงพอใจ
7. สรุปผลการวิจัย
8. จัดทำเอกสารให้สมบูรณ์

1.7 แผนการดำเนินงานวิจัย

ผู้วิจัยประยุกต์ใช้โมเดล ML-GCN และเปรียบเทียบอัลกอริทึมโครงข่ายประสาทเทียมแบบคอนโวลูชันทั้ง 3 แบบ มีแผนเริ่มดำเนินงานตั้งแต่วันที่ 1 กรกฎาคม 2566 ถึงวันที่ 29 พฤศจิกายน พ.ศ. 2567 รวมทั้งสิ้น 17 สัปดาห์ หรือ 518 วัน โดยมีรายละเอียดดำเนินงานวิจัยดังตารางที่ 1-3

ตารางที่ 1-3 แผนการดำเนินงานวิจัย

แผนการดำเนินงานวิจัย					
งาน		**คาดว่าจะเริ่ม**	**คาดว่าจะเสร็จ**	**วันที่เริ่ม**	**วันที่เสร็จ**
ตั้งแต่วันที่ 1 กรกฎาคม 2566 ถึงวันที่ 31 พฤศจิกายน พ.ศ. 2567					
1.	ศึกษาข้อมูลและงานวิจัย	ก.ค. 66	พ.ย. 66	ก.ค. 66	พ.ย. 66
2.	จัดหาข้อมูลและเตรียมเครื่องมือต่าง ๆ	ก.ย. 66	พ.ย. 66	ก.ย. 66	พ.ย. 66
3.	เตรียมข้อมูล				
	3.1 ทำความสะอาดข้อมูลและคัดแยกแท็ก	ธ.ค. 66	ม.ค. 67	ธ.ค. 66	ม.ค. 67
	3.2 สร้างแท็กใหม่และคัดเลือกแท็กที่สำคัญ	ม.ค. 67	ม.ค. 67	ม.ค. 67	ม.ค. 67
4.	เตรียมโมเดล	ก.พ. 67	ก.พ. 67	ก.พ. 67	ก.พ. 67
5.	รันและทดสอบโมเดล	ก.พ. 67	ก.พ. 67	ก.พ. 67	มิ.ย. 67
6.	เตรียมข้อมูล ครั้งที่ 2				
	6.1 ตรวจสอบแท็กของรูปภาพด้วยตนเอง	ก.ค. 67	ส.ค. 67	ก.ค. 67	ส.ค. 67
5.	รันและทดสอบโมเดล ครั้งที่ 2	ก.ย. 67	ต.ค. 67	ก.ย. 67	ต.ค. 67
6.	สรุปผลการวิจัยและจัดทำเอกสารให้สมบูรณ์	ต.ค. 67	ต.ค. 67	ต.ค. 67	ต.ค. 67
7.	เผยแพร่งานวิจัย	ต.ค. 67	ต.ค. 67	ต.ค. 67	ต.ค. 67
8.	นำเสนองานวิจัย	พ.ย. 67	พ.ย. 67	พ.ย. 67	พ.ย. 67

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ผู้วิจัยต้องการประยุกต์ใช้โมเดล ML-GCN ในการแท็กภาพอนิเมะโดยใช้อัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันทั้ง 3 แบบ ผู้วิจัยจึงศึกษาข้อมูลรวมทั้งทฤษฎีที่เกี่ยวข้องเพื่อให้มีความรู้ความเข้าใจและประยุกต์ใช้โมเดลได้ถูกต้องตรงตามเป้าหมาย ซึ่งผู้วิจัยได้ค้นคว้าข้อมูลได้แก่ นิยามคำศัพท์เฉพาะ โครงข่ายประสาทเทียมแบบคอนโวลูชัน โครงข่ายคอนโวลูชันแบบกราฟ และงานวิจัยหรือบทความที่เกี่ยวข้อง ทั้งนี้ผู้วิจัยใช้อัลกอริทึม 3 แบบของโครงข่ายประสาทเทียมแบบคอนโวลูชันในการพัฒนาโมเดล ได้แก่ ResNET ซึ่งเป็นอัลกอริทึมที่ใช้ในโมเดล ML-GCN และใช้ ResNeXT กับ EfficientNet ซึ่งทั้ง 2 เป็นอัลกอริทึมใหม่ของโครงข่ายประสาทเทียมแบบคอนโวลูชันเพื่อวิเคราะห์ว่าอัลกอริทึมใดจำแนกแท็กของภาพอนิเมะร่วมกับโมเดล ML-GCN ได้ถูกต้องมากกว่า

2.1 นิยามคำศัพท์เฉพาะ

ภาพอนิเมะ คือ ภาพการ์ตูนซึ่งใช้รูปแบบการวาดการ์ตูนที่ถูกแพร่หลายมาจากประเทศญี่ปุ่น

แท็ก คือ สิ่งที่บ่งบอกลักษณะของภาพอนิเมะ (Label) และมักถูกใช้เป็นคีย์เวิร์ดเพื่อให้สามารถค้นหาภาพอนิเมะต่าง ๆ ตรงตามลักษณะที่ต้องการได้ง่ายขึ้น ในแต่ละภาพอนิเมะจึงมักมีแท็กมากกว่า 1 แท็ก

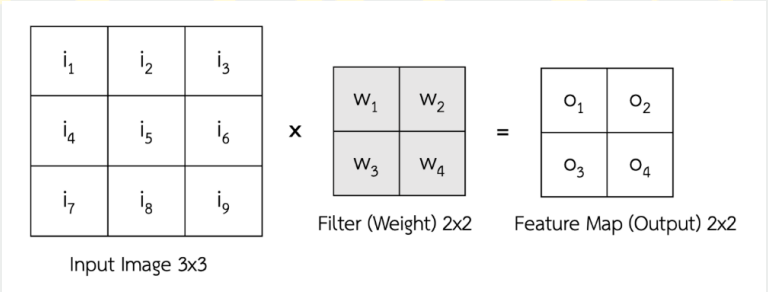
2.2 โครงข่ายประสาทเทียมแบบคอนโวลูชัน

โครงข่ายประสาทเทียมเป็นอัลกอริทึมที่ใช้เทคนิคการเรียนรู้เชิงลึกจำลองการทำงานของโครงข่ายประสาทในสมอง เพื่อให้คอมพิวเตอร์สามารถเรียนรู้และวิเคราะห์ได้แบบมนุษย์ โครงข่ายประสาทเทียมทั่วไปจะนำข้อมูลฟีเจอร์ (Feature) ของข้อมูลนำเข้ามาประมวลผล ซึ่งในกรณีที่ใช้ข้อมูลนำเข้าเป็นรูปภาพ ทุกข้อมูลพิกเซลของรูปภาพ (Pixel) จะถูกใช้เป็นฟีเจอร์ ดังนั้นจึงไม่เหมาะแก่การใช้วิธีการของโครงข่ายประสาทเทียมทั่วไปมาประมวลผล เพราะฟีเจอร์จะมีจำนวนมาก จึงมีการคิดค้นวิธีประมวลผลใหม่เพื่อใช้สำหรับประมวลผลข้อมูลนำเข้าที่เป็นรูปภาพโดยเฉพาะ โครงข่ายประสาทเทียมแบบคอนโวลูชันจึงถูกสร้างขึ้น

โครงข่ายประสาทเทียมแบบคอนโวลูชันเป็นสถาปัตยกรรมโครงข่ายประสาทเทียมพื้นฐานสำหรับใช้ในการจำแนกข้อมูล (Classification) โดยมีข้อมูลนำเข้าเป็นรูปภาพ (S. Weidman,

2019) โครงข่ายประสาทเทียมแบบคอนโวลูชันเป็นโครงข่ายประสาทเทียมประเภทหนึ่งที่ถูกพัฒนาขึ้นเพื่อใช้จำแนกข้อมูลประเภทรูปภาพโดยเฉพาะ

กระบวนการคอนโวลูชัน (Convolution Operation) เป็นกระบวนการสำหรับประมวลผลข้อมูลนำเข้าที่เป็นรูปภาพ (Input Image) ซึ่งจะถูกมองว่าเป็นข้อมูลเมทริกซ์ (Matrix) ขั้นตอนนี้ต้องใช้ค่าน้ำหนักที่มีรูปแบบเป็นเมทริกซ์เรียกว่า ฟิวเตอร์ (Convolutional Filter) หรือ เคอร์เนล (Kernel) เพื่อใช้หารูปแบบที่เหมือนกัน (Pattern) บนรูปภาพที่นำเข้ามาในโมเดล โดยการนำเมทริกซ์ของฟิวเตอร์ไปคูณกับส่วนต่าง ๆ ของเมทริกซ์ของรูปภาพจนครบทุกส่วน ซึ่งเมทริกซ์ผลลัพธ์ที่เกิดจากการคูณนี้เรียกว่า ฟีเจอร์แมพ (Feature Map) ขนาดของฟีเจอร์แมพจะมีขนาดเล็กกว่ารูปภาพที่นำไปคูณกับฟิวเตอร์เสมอ ดังตัวอย่างในภาพที่ 2-1

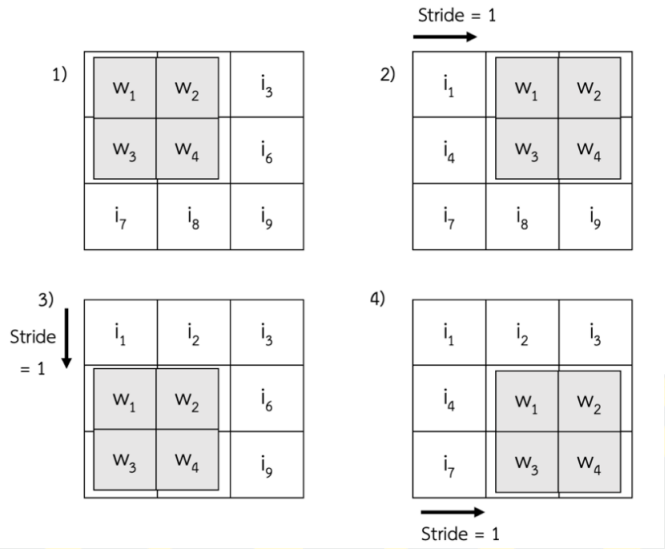


ภาพที่ 2-1 กระบวนการคอนโวลูชัน

จากภาพที่ 2-1 ค่าต่าง ๆ ในฟีเจอร์แมพได้จากการนำฟิวเตอร์ไปคูณกับส่วนต่าง ๆ ของรูปภาพจนครบทุกส่วนดังนี้

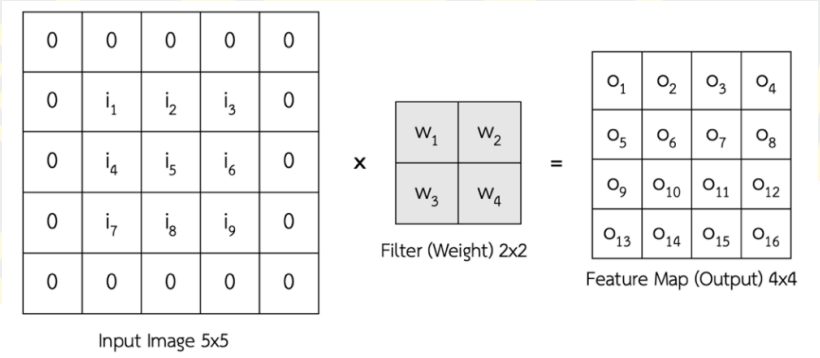
- $o_1 = (i_1 \times w_1) + (i_2 \times w_2) + (i_4 \times w_3) + (i_5 \times w_4)$
- $o_2 = (i_2 \times w_1) + (i_3 \times w_2) + (i_5 \times w_3) + (i_6 \times w_4)$
- $o_3 = (i_4 \times w_1) + (i_5 \times w_2) + (i_7 \times w_3) + (i_8 \times w_4)$
- $o_4 = (i_5 \times w_1) + (i_6 \times w_2) + (i_8 \times w_3) + (i_9 \times w_4)$

ขั้นตอนการนำเมทริกซ์ของฟิวเตอร์ไปคูณกับส่วนต่าง ๆ ของเมทริกซ์ของรูปภาพที่นำเข้ามาในโมเดลในภาพที่ 2-1 ถูกแสดงบนภาพที่ 2-2 โดยแสดงลำดับการเลื่อนฟิวเตอร์ไปคูณกับส่วนต่าง ๆ ของรูปภาพจนครบทุกส่วน ฟิวเตอร์ต้องเริ่มเลื่อนจากซ้ายไปขวาเสมอและเมื่อคูณเสร็จก็จะเลื่อนลงล่างเพื่อคูณแถวใหม่ ซึ่งระยะการเลื่อนฟิวเตอร์จะถูกเรียกว่า สไตค์ (Stride) โดยภาพที่ 2-2 มีค่าสไตค์คือ 1 ซึ่งยิ่งค่าสไตค์มีมาก ขนาดของฟีเจอร์แมพที่ได้จะยิ่งเล็ก และมีการสูญหายของข้อมูลมากขึ้นเช่นกัน ดังนั้นจึงต้องกำหนดค่าระยะการเลื่อนให้เหมาะสม



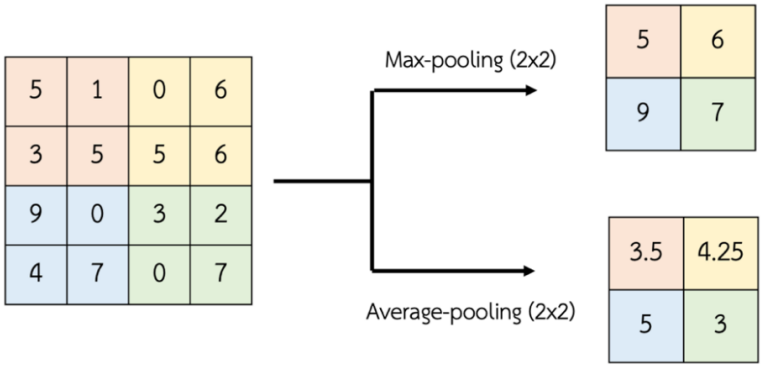
ภาพที่ 2-2 ลำดับการเลื่อนฟิวเตอร์ในกระบวนการคอนโวลูชันโดยค่าสไตค์คือ 1

การลดปัญหาไม่ให้ฟีเจอร์แมพที่ได้ออกมามีขนาดเล็กเกินไปและลดการสูญเสียข้อมูลบริเวณขอบภาพ สามารถทำได้โดยการเติมระยะขอบให้แก่รูปภาพ เรียกว่า แพดดิ่ง (Padding) โดยระยะขอบจะมีค่าเมทริกซ์คือ 0 ดังตัวอย่างในภาพที่ 2-3



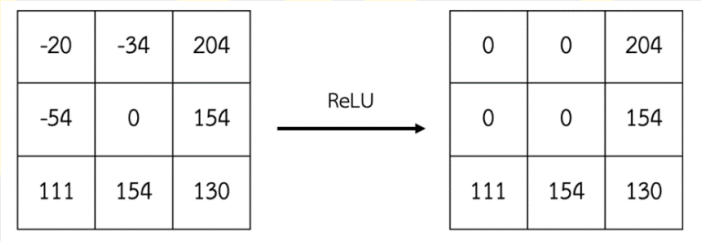
ภาพที่ 2-3 กระบวนการคอนโวลูชันที่มีการใช้แพดดิ่งและมีค่าสไตค์คือ 1

วิธีลดขนาดรูปภาพหรือฟีเจอร์แมพในโครงข่ายประสาทเทียมแบบคอนโวลูชันมีหลัก ๆ 2 วิธี คือ การกำหนดค่าสไตค์ให้สูงหรือใช้พูลลิ่งเลเยอร์ (Pooling Layer) โดยพูลลิ่งเลเยอร์ที่นิยมใช้ ได้แก่ พูลลิ่งค่าสูงสุด (Max-pooling) และพูลลิ่งค่าเฉลี่ย (Average-pooling) พูลลิ่งเลเยอร์จะทำการลดขนาด (Downsample) ให้แก่รูปภาพหรือฟีเจอร์แมพที่ส่งเข้ามาในพูลลิ่งเลเยอร์ โดยการแบ่งรูปภาพหรือฟีเจอร์แมพเหล่านั้นออกเป็นส่วน ๆ แล้วหาค่าสูงสุดหรือค่าเฉลี่ยในแต่ละส่วนแล้วนำมาต่อกัน ทำให้ได้ผลลัพธ์ออกมาเป็นแบ่งรูปภาพหรือฟีเจอร์แมพที่มีขนาดเล็กลง หากตอนแบ่งส่วนใช้ค่าสูงจะทำให้มีการสูญหายของข้อมูลมาก ส่วนใหญ่จึงแบ่งส่วนแบบ 2x2 ดังตัวอย่างในภาพที่ 2-4



ภาพที่ 2-4 การทำพูลลิ่งค่าสูงสุดและพูลลิ่งค่าเฉลี่ยแบบ 2x2

โครงข่ายประสาทเทียมแบบคอนโวลูชันนิยมนิยมใช้ฟังก์ชันลีลู (Rectified Linear Unit : ReLU) เพื่อกำจัดค่าตัวเลขติดลบและแทนที่ด้วย 0 ในกรณีที่ค่าในฟีเจอร์แมพมีค่าน้อยกว่า 0 หลังคูณกับฟิวเตอร์ ดังตัวอย่างในภาพที่ 2-5

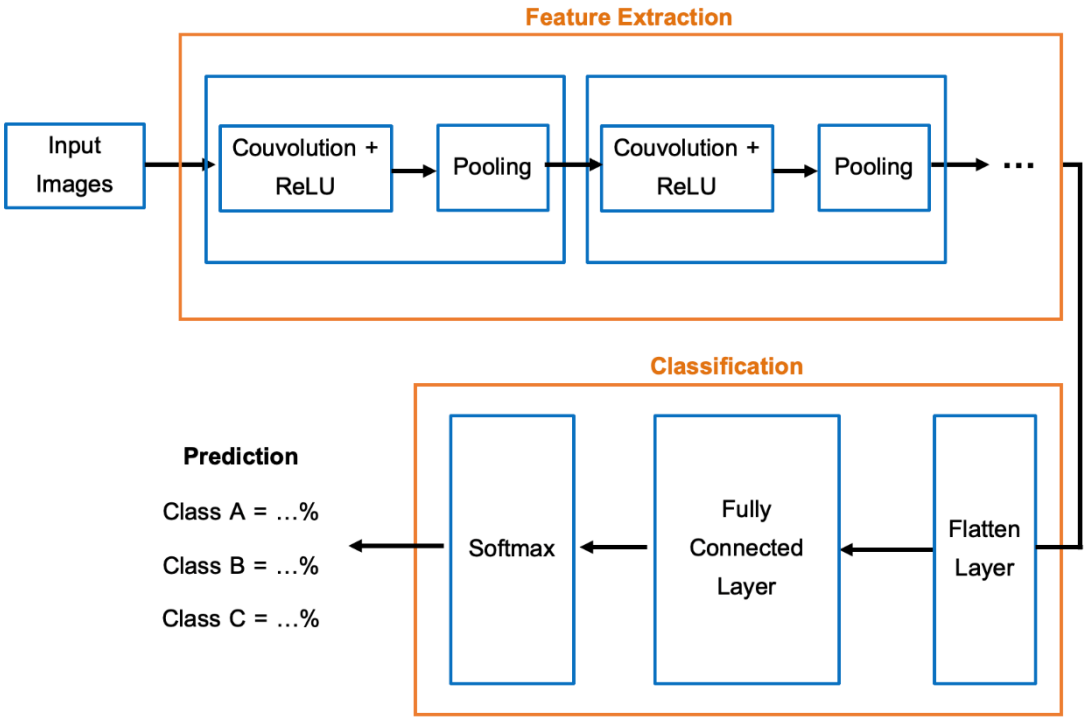


ภาพที่ 2-5 การนำฟีเจอร์แมพผ่านฟังก์ชันลีลู

โดยทั่วไปโครงข่ายประสาทเทียมแบบคอนโวลูชันจะแบ่งเป็น 2 ส่วนหลัก ๆ คือ ขั้นตอนสกัดฟีเจอร์และขั้นตอนจำแนกรูปภาพดังตัวอย่างในภาพที่ 2-6

- ขั้นตอนสกัดฟีเจอร์ (Feature Extraction) คือ ขั้นตอนในการทำกระบวนการคอนโวลูชันซึ่งเป็นขั้นตอนในการนำภาพข้อมูลนำเข้าและฟิวเตอร์มาสร้างฟีเจอร์แมพ ซึ่งกระบวนการคอนโวลูชันสามารถมีได้มากกว่า 1 ชั้น
- ขั้นตอนจำแนกภาพ (Classification) คือ ขั้นตอนสำหรับจำแนกรูปภาพที่นำเข้ามาว่ามีลักษณะคล้ายกับคลาสใด โดยจะนำฟีเจอร์แมพสุดท้ายที่ได้จากกระบวนการคอนโวลูชันมาแปลงเป็นเลเยอร์ที่มีข้อมูลเป็น 1 มิติ เรียกว่า แฟ็ทเท็นเลเยอร์ (Fatten Layer) และนำเข้าสู่เลเยอร์ที่เชื่อมต่อทุกส่วนเข้าด้วยกัน (Fully Connected Layer) และจากนั้นใช้ซอฟต์แม็คฟังก์ชัน (Softmax Function) เพื่อ

แปลงค่าที่ได้เป็นตัวเลขความน่าจะเป็นว่าในแต่ละคลาส คลาสใดมีความน่าจะเป็นมากที่สุด



ภาพที่ 2-6 โครงสร้างพื้นฐานของโครงข่ายประสาทเทียมแบบคอนโวลูชัน

ฟิวเตอร์ที่ใช้ในกระบวนการคอนโวลูชันสามารถสร้างจากการสุ่มค่าน้ำหนักหรือใช้ฟิวเตอร์แบบเฉพาะที่มีการกำหนดค่าน้ำหนักอย่างชัดเจน ยกตัวอย่างเช่น

- ฟิวเตอร์ตรวจจับเส้นขอบ (Edge Detection) ซึ่งมีค่าเมทริกซ์ดังตัวอย่างในภาพที่ 2-7

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{Or} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

ภาพที่ 2-7 ฟิวเตอร์สำหรับตรวจจับเส้นขอบ

- ฟิวเตอร์ปรับความคมชัด (Sharpen) ซึ่งมีค่าเมทริกซ์ดังตัวอย่างในภาพที่ 2-8

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

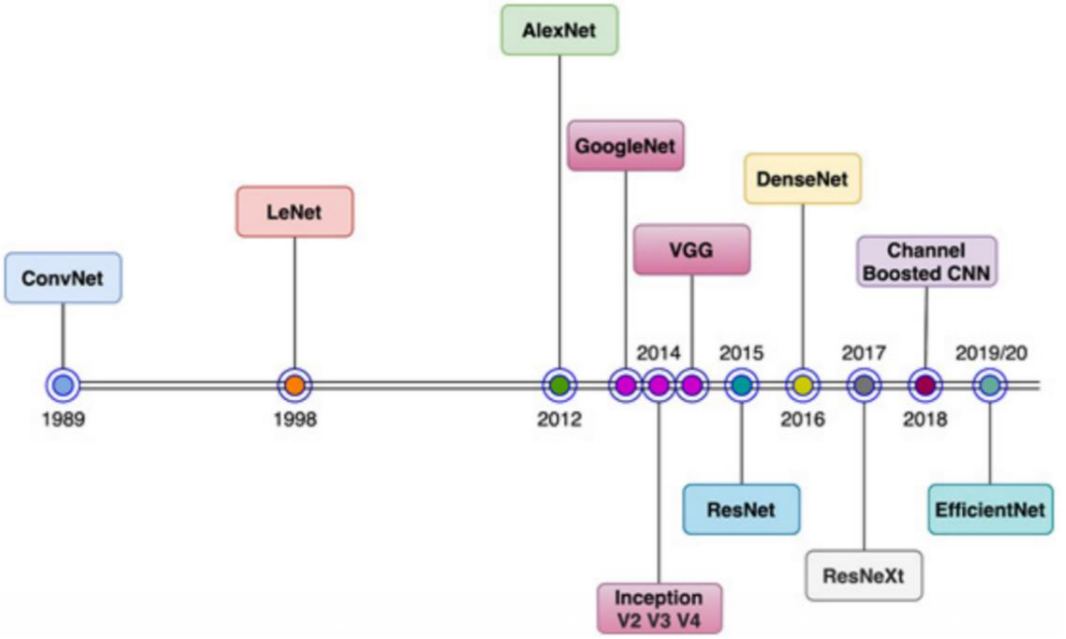
ภาพที่ 2-8 ฟิวเตอร์สำหรับปรับให้ภาพมีความคมชัด

- ฟิวเตอร์ปรับภาพให้เบลอ (Box Blur) ซึ่งมีค่าเมทริกซ์ดังตัวอย่างในภาพที่ 2-9

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

ภาพที่ 2-9 ฟิวเตอร์สำหรับทำให้ภาพเบลอ

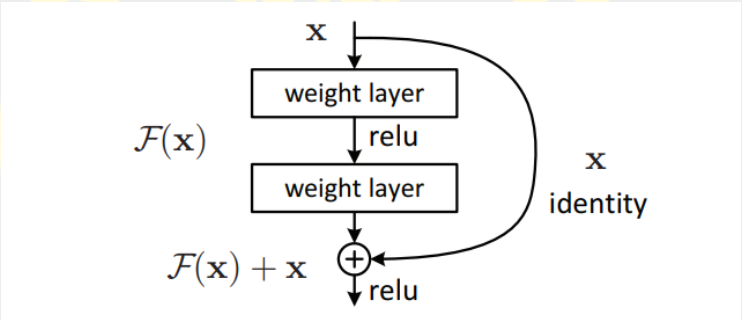
งานวิจัยนี้ผู้วิจัยสร้างโมเดลโดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน 3 แบบ ได้แก่ ResNET ซึ่งเป็นอัลกอริทึมที่ใช้ในโมเดล ML-GCN และใช้ ResNeXT กับ EfficientNet ซึ่งเป็นอัลกอริทึมใหม่ของโครงข่ายประสาทเทียมแบบคอนโวลูชันดังตัวอย่างในภาพที่ 2-10 ผู้วิจัยตัดสินใจไม่เลือกใช้อัลกอริทึม Channel Boosted เนื่องจากมีข้อมูลในการศึกษาค้นคว้าน้อย



ภาพที่ 2-10 ทามไลน์ของอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชัน
อ้างอิง: (R. A. Jha, 2021)

2.2.1 ResNET (Residual Network)

ResNET เป็นอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ถูกคิดค้นในปี ค.ศ. 2015 เพื่อแก้ปัญหาอัตราการเกิดข้อผิดพลาด (Error Rate) ที่เพิ่มขึ้นจากการใช้เลเยอร์จำนวนมาก ซึ่งโดยทั่วไปการเพิ่มเลเยอร์จะช่วยลดอัตราการเกิดข้อผิดพลาดได้ แต่ถ้าจำนวนเลยอร์ที่ใช้มีจำนวนมากจะกลายเป็นการเพิ่มอัตราการเกิดข้อผิดพลาดเนื่องจากค่า Gradient กลายเป็น 0 หรือมีค่ามากเกินไป ปัญหานี้เรียกว่า Vanishing/Exploding Gradient อัลกอริทึมนี้ใช้เทคนิคการข้ามการเลเยอร์ (Skip Connection/Shortcut Connection) เพื่อแก้ปัญหาดังกล่าว โดยการข้ามเลเยอร์ที่ส่งผลให้ประสิทธิภาพของอัลกอริทึมแย่ลง ซึ่งชุดของเลเยอร์ที่ใช้เทคนิคการข้ามการเลเยอร์จะถูกเรียกว่า Residual Block (K. He, Zhang, X., Ren, S., & Sun, J. , 2016) ภาพที่ 2-11 คือตัวอย่างของ Residual Block



ภาพที่ 2-11 เทคนิคการข้ามการเลเยอร์ใน ResNET
อ้างอิง: (K. He, Zhang, X., Ren, S., & Sun, J. , 2016)

ภาพที่ 2-11 สามารถเขียนเป็นสมการดังสมการที่ (1) F(x) คือ ฟังก์ชันเลเยอร์ โดยในกรณีที่ F(x) มีค่า Gradient ไม่ดี จะเกิดการข้ามการเลเยอร์ ส่งผลให้F(x) มีค่าเป็น 0 ดังนั้น y จะจึงมีค่าเท่ากับ x เพื่อป้องกัน Vanishing/Exploding Gradient ส่วนกรณีที่ F(x) มีค่า Gradient ปกติ จะไม่เกิดการข้ามการเลเยอร์ และ y จะจึงมีค่าเท่ากับ F(x) + x

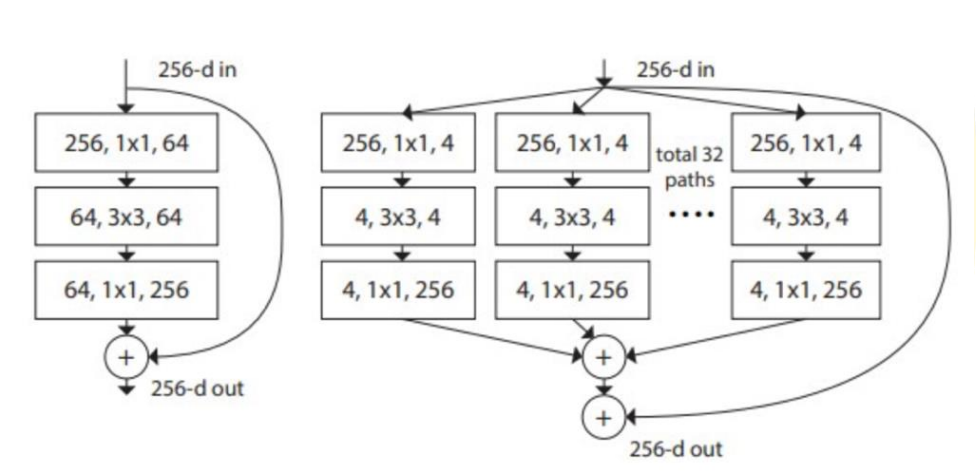
$$y = F(x) + x \tag{1}$$

ResNET ที่ใช้ในงานวิจัยนี้คือ ResNET-101 ซึ่งถูกใช้ในโมเดล ML-GCN เพื่อเรียนรู้ข้อมูลนำเข้าที่เป็นรูปภาพ ฟิวเตอร์ที่ใช้ในโมเดลนี้จะถูกสร้างโดยการสุ่มค่าและจะค่อย ๆ ถูกปรับค่าในระหว่างการเรียนรู้ของโมเดลจนได้ค่าที่เหมาะสม โมเดล ResNET-101 ประกอบด้วยเลเยอร์หลัก 101 เลเยอร์ ซึ่งแบ่งโครงสร้างหลักเป็น 6 ส่วน ได้แก่

1) ส่วนหัว ซึ่งประกอบด้วย 2 เลเยอร์ ได้แก่
   - Conv1 ซึ่งใช้ฟิวเตอร์ขนาด 7x7 จำนวน 64 ฟิวเตอร์ และใช้ สไตค์ 2
   - Max Pooling ใช้ฟิวเตอร์ขนาด 3x3 และ สไตค์ 2
2) Conv2_x เป็น Residual Block จำนวน 3 ชุด แต่ละชุดด้วย 3 เลเยอร์ รวมทั้งหมดเป็น 9 เลเยอร์ สไตค์ 1 และให้ผลลัพธ์เป็นฟีเจอร์แมพขนาด 56x56 โดยแต่ละชุดของ Residual Block ประกอบด้วยเลเยอร์ดังต่อไปนี้
   2.1) เลเยอร์ที่ 1 ใช้ฟิวเตอร์ขนาด 1x1 จำนวน 64 แผ่น
   2.2) เลเยอร์ที่ 2 ใช้ฟิวเตอร์ขนาด 3x3 จำนวน 64 แผ่น
   2.3) เลเยอร์ที่ 3 ใช้ฟิวเตอร์ขนาด 1x1 จำนวน 256 แผ่น
3) Conv3_x เป็น Residual Block จำนวน 4 ชุด แต่ละชุดด้วย 3 เลเยอร์ รวมทั้งหมดเป็น 12 เลเยอร์ สไตค์ 2 และให้ผลลัพธ์เป็นฟีเจอร์แมพขนาด 28x28 โดยแต่ละชุดของ Residual Block ประกอบด้วยเลเยอร์ดังต่อไปนี้
   3.1) เลเยอร์ที่ 1 ใช้ฟิวเตอร์ขนาด 1x1 จำนวน 128 แผ่น
   3.2) เลเยอร์ที่ 2 ใช้ฟิวเตอร์ขนาด 3x3 จำนวน 128 แผ่น
   3.3) เลเยอร์ที่ 3 ใช้ฟิวเตอร์ขนาด 1x1 จำนวน 512 แผ่น
4) Conv4_x เป็น Residual Block จำนวน 23 ชุด แต่ละชุดด้วย 3 เลเยอร์ รวมทั้งหมดเป็น 69 เลเยอร์ สไตค์ 2 และให้ผลลัพธ์เป็นฟีเจอร์แมพขนาด 14x14 โดยแต่ละชุดของ Residual Block ประกอบด้วยเลเยอร์ดังต่อไปนี้
   4.1) เลเยอร์ที่ 1 ใช้ฟิวเตอร์ขนาด 1x1 จำนวน 256 แผ่น
   4.2) เลเยอร์ที่ 2 ใช้ฟิวเตอร์ขนาด 3x3 จำนวน 256 แผ่น
   4.3) เลเยอร์ที่ 3 ใช้ฟิวเตอร์ขนาด 1x1 จำนวน 1024 แผ่น
5) Conv5_x เป็น Residual Block จำนวน 3 ชุด แต่ละชุดด้วย 3 เลเยอร์ รวมทั้งหมดเป็น 9 เลเยอร์ สไตค์ 2 และให้ผลลัพธ์เป็นฟีเจอร์แมพขนาด 7x7 โดยแต่ละชุดของ Residual Block ประกอบด้วยเลเยอร์ดังต่อไปนี้
   5.1) เลเยอร์ที่ 1 ใช้ฟิวเตอร์ขนาด 1x1 จำนวน 512 แผ่น
   5.2) เลเยอร์ที่ 2 ใช้ฟิวเตอร์ขนาด 3x3 จำนวน 512 แผ่น
   5.3) เลเยอร์ที่ 3 ใช้ฟิวเตอร์ขนาด 1x1 จำนวน 2048 แผ่น
6) พูลลิ่ง และ Softmax Function

2.2.2 ResNeXT

ResNeXT เป็นอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ถูกคิดค้นในปี ค.ศ. 2017 เพื่อเพิ่มทางเลือกในการเพิ่มประสิทธิภาพของโมเดลโดยพยายามไม่ใช้การเพิ่มความลึกหรือความกว้างของโมเดล อัลกอริทึมนี้ใช้การเพิ่มกิ่ง (Branch) แทนการเพิ่มความลึกหรือความกว้างของโมเดล เรียกว่า Cardinality ใช้หลักการแยกส่วนข้อมูลนำเข้าที่เป็นรูปภาพเพื่อประมวลผล จากนั้นนำมารวมกันในตอนท้าย จุดประสงค์หลักของอัลกอริทึมคือใช้เพิ่มประสิทธิภาพให้กับโมเดลที่ใช้ข้อมูลนำเข้าจำนวนมากโดยพยายามหลีกเลี่ยงการเพิ่มความลึกหรือความกว้างของโมเดล (S. Xie, Girshick, R., Tu, Z., He, K., & Dollar, P., 2017) ภาพที่ 2-12 แสดงการเปรียบเทียบระหว่างอัลกอริทึม ResNET และอัลกอริทึม ResNeXT



ภาพที่ 2-12 การเปรียบเทียบระหว่างอัลกอริทึม ResNET (ภาพซ้าย) และ ResNeXT (ภาพขวา)
อ้างอิง: (S. Xie, Girshick, R., Tu, Z., He, K., & Dollar, P., 2017)

รูปขวาในภาพที่ 2-12 สามารถเขียนเป็นสมการดังสมการที่ (2) ซึ่งสมการนี้จะคล้ายกับสมการของ ResNET แต่เปลี่ยนจาก F(x) เป็น T(x) และเพิ่มเครื่องหมาย $\sum$ เพราะ ResNeXT มีการแยกประมวลผลแบบขนาน (parallel) และต้องนำมารวมกันในตอนท้าย จำนวนของ $\sum$ คือ C (Cardinality) หมายถึงจำนวนกิ่ง

$$y = \sum_{i=1}^{C} T_i(x) + x \tag{2}$$

ResNeXT ที่ใช้ในงานวิจัยนี้คือ ResNeXT-101 ซึ่งมีโครงสร้างเหมือนกับ ResNET-101 แต่แตกต่างในส่วน Residual Block เพราะในเลเยอร์ที่ 2 ของทุก Residual Block จะถูกแบ่งเป็น Cardinality จำนวน 32 กิ่ง

2.2.3 EfficientNet

EfficientNet เป็นอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ถูกคิดค้นในปี ค.ศ. 2019 เพื่อแก้ปัญหาการกำหนดค่าอัตราส่วน (Scaling) ของโมเดลโครงข่ายประสาทเทียมแบบคอนโวลูชันรุ่นก่อนหน้าเพราะการปรับอัตราส่วนของโมเดลสามารถเพิ่มประสิทธิภาพให้แก่โมเดลได้ แต่ยังไม่มีรูปแบบการปรับค่าอัตราส่วนโมเดลที่ชัดเจน ทำให้ต้องใช้การลองผิดลองถูกแก้ไขซ้ำหลายครั้งจนกระทั่งได้ค่าที่เหมาะสม วิธีการปรับอัตราส่วนของโมเดลมี 3 แบบ ได้แก่ (1) การปรับอัตราส่วนความกว้างของโมเดลโดยการเพิ่มจำนวนนิวตรอนในชั้นเลยอร์ (2) การปรับอัตราส่วนความลึกของโมเดลโดยการเพิ่มจำนวนชั้นเลเยอร์ และ (3) การปรับอัตราส่วนความละเอียด (ความกว้างและความสูง) ของข้อมูลนำเข้าที่เป็นรูปภาพ อัลกอริทึมนี้จึงถูกคิดค้นเพื่อเสนอวิธีการปรับค่าอัตราส่วนทั้ง 3 ด้าน (ความกว้าง ความลึก และความละเอียด) ให้เกิดความสมดุลเรียกว่า Compound Scaling โดยใช้อัตราส่วนคงที่ (Fixed Ratio) เพื่อในปรับค่าอัตราส่วนทั้ง 3 ด้าน (M. Tan, & Le, Q. V. , 2020)

ภาพที่ 2-13 แสดงการเปรียบเทียบโมเดลโครงข่ายประสาทเทียมแบบคอนโวลูชัน 5 แบบที่มีการปรับอัตราส่วน ดังต่อไปนี้

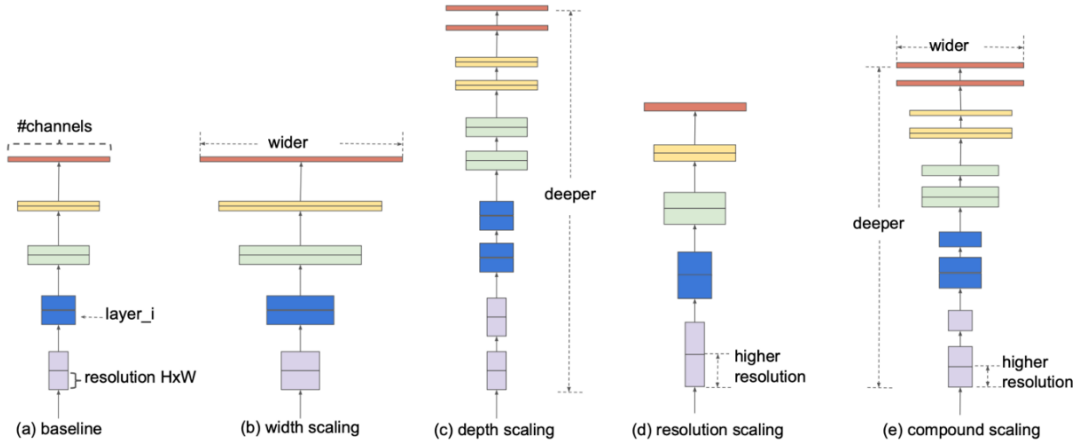
(a) คือ โมเดลโครงข่ายประสาทเทียมแบบพื้นฐาน

(b) คือ โมเดลโครงข่ายประสาทเทียมที่ถูกปรับแค่อัตราส่วนความกว้าง

(c) คือ โครงสร้างของโมเดลโครงข่ายประสาทเทียมที่ถูกปรับแค่อัตราส่วนลึก

(d) คือ โครงสร้างของโมเดลโครงข่ายประสาทเทียมที่ถูกปรับแค่อัตราส่วนความละเอียดของรูปภาพที่ใช้ในการสอนโมเดล

(e) คือ โครงสร้างของโมเดลแบบ Compound Scaling ที่ถูกเสนอใน EfficientNet ซึ่งมีการปรับอัตราส่วนทั้ง 3 ด้านโดยใช้อัตราส่วนคงที่



ภาพที่ 2-13 การเปรียบเทียบการปรับอัตราส่วนในโมเดลโครงข่ายประสาทเทียมแบบคอนโวลูชัน
อ้างอิง: (M. Tan, & Le, Q. V. , 2020)

สมการของ Compound Scaling มีดังต่อไปนี้

$$depth: d = \alpha^{\Phi}$$

$$width: w = \beta^{\Phi}$$

$$resolution: r = \gamma^{\Phi}$$

$$s.t. \alpha \times \beta^2 \times \gamma^2 \approx 2$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

φ เป็นค่าสัมประสิทธิ์ที่ผู้ใช้กำหนดขึ้นเองเพื่อควบคุมจำนวนทรัพยากรสำหรับการปรับขนาดโมเดล (model scaling) ส่วน α β และ γ เป็นค่าคงที่ที่สามารถกำหนดได้โดยการค้นหาแบบกริดขนาดเล็ก (Small Grid Search) ซึ่งใช้เป็นค่าคงที่ของความลึกของโมเดล ความกว้างของโมเดล และความละเอียดของรูปภาพที่ใช้ในการสอนโมเดล

มาตรวัดความเร็วในการคำนวณ (Floating-point operations per second: FLOPS) ของกระบวนการคอนโวลูชันโดยทั่วไปจะมีสัดส่วนแปรผันต่อค่าความลึกของโมเดล ความกว้างของโมเดล และความละเอียดของรูปภาพที่ใช้ในการสอนโมเดลดังนี้ d, w^2 และ r^2 เช่น หากเพิ่มความลึกเป็น 2 เท่า FLOPS จะเพิ่ม 2 เท่าเช่นกัน แต่ถ้าหากเพิ่มเพิ่มความกว้างเป็น 2 เท่า FLOPS

จะเพิ่ม 4 เท่า เป็นต้น นอกจากนี้มีการจำกัดค่า (constraint) โดยผลคูณของ α , β^2 และ γ^2 จะมีค่าประมาน 2 โดยซึ่งค่า α, β และ γ แต่ละค่าจะมีค่ามากกว่า 1

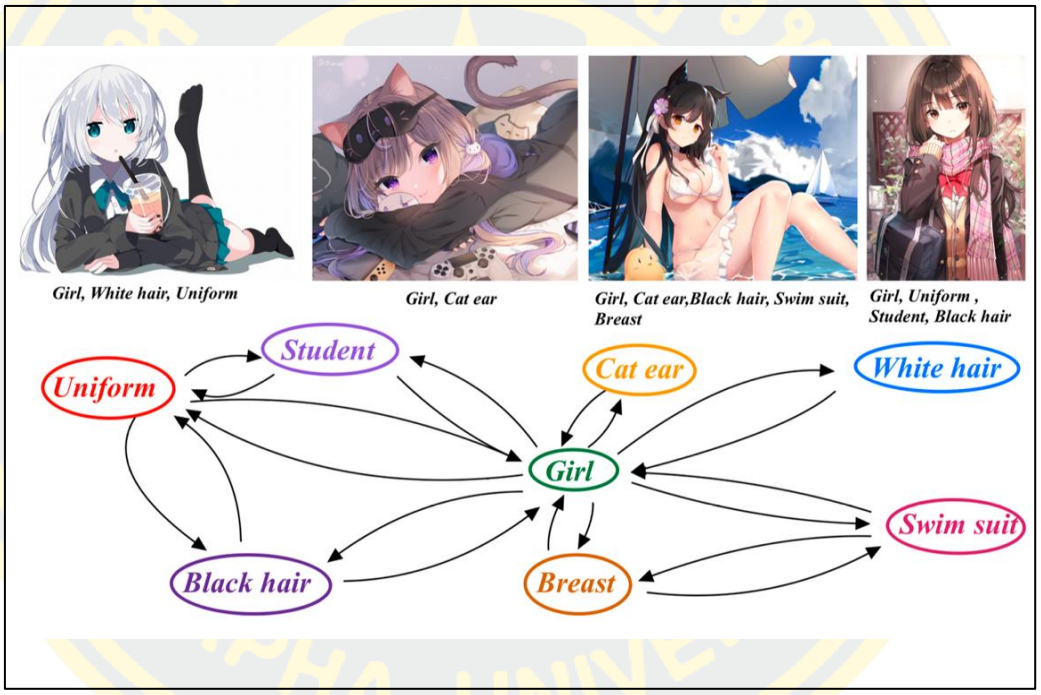
EfficientNet ใช้โครงสร้างแบบ MBConv Blocks ซึ่งมีต้นแบบมาจาก Mobile Inverted Bottleneck Convolution ของโมเดล MobileNetV2 เพื่อให้สามารถใช้ Compound Scaling ปรับความกว้างและความลึกของโมเดลโดยรักษาประสิทธิภาพของโมเดลไว้

EfficientNet ที่ใช้ในงานวิจัยนี้คือ EfficientNetB5 เพราะขนาดของข้อมูลเอาท์พุทของ EfficientNetB5 (ก่อนผ่าน Fully Connected Layer) มีขนาด 2048 แบบเดียวกับ ResNET-101 และ ResNeXT-101 โดย EfficientNetB5 มีโครงสร้างหลัก 9 ส่วน ได้แก่

1) Stem หรือ Convolutional Layer แรก รูปภาพนำเข้าต้องมีขนาด 456 x 456 x 3 ใช้ฟิวเตอร์ขนาด 3x3 จำนวน 48 แผ่น และสไตค์ 2
2) MBConv Blocks ที่ 1 มีค่า Expansion Factor คือ 1 มีฟิวเตอร์ขนาด 3x3 จำนวน 24 แผ่น ทำซ้ำ 3 รอบ และสไตค์ 1
3) MBConv Blocks ที่ 2 มีค่า Expansion Factor คือ 6 มีฟิวเตอร์ขนาด 3x3 จำนวน 40 แผ่น ทำซ้ำ 4 รอบ และสไตค์ 2
4) MBConv Blocks ที่ 3 มีค่า Expansion Factor คือ 6 มีฟิวเตอร์ขนาด 5x5 จำนวน 80 แผ่น ทำซ้ำ 4 รอบ และสไตค์ 2
5) MBConv Blocks ที่ 4 มีค่า Expansion Factor คือ 6 มีฟิวเตอร์ขนาด 3x3 จำนวน 112 แผ่น ทำซ้ำ 6 รอบ และสไตค์ 1
6) MBConv Blocks ที่ 5 มีค่า Expansion Factor คือ 6 มีฟิวเตอร์ขนาด 5x5 จำนวน 192 แผ่น ทำซ้ำ 6 รอบ และสไตค์ 2
7) MBConv Blocks ที่ 6 มีค่า Expansion Factor คือ 6 มีฟิวเตอร์ขนาด 3x3 จำนวน 320 แผ่น ทำซ้ำ 3 รอบ และสไตค์ 1
8) เลยอร์สุดท้าย เป็น Convolutional Layer มีฟิวเตอร์ขนาด 1x1 จำนวน 1280 แผ่น และสไตค์ 1
9) พูลลิ่ง

2.3 โครงข่ายคอนโวลูชันแบบกราฟ

การทำนายแท็กของภาพอนิเมะ แท็กบางแท็กมักมีความสัมพันธ์ซึ่งกันและกัน เช่น บางแท็กมักเกิดขึ้นพร้อมกัน ดังนั้นหากโมเดลสามารถเรียนรู้ความสัมพันธ์ของแท็กเหล่านี้จะสามารถเพิ่มความแม่นยำในการแท็กภาพได้มากขึ้น ภาพที่ 2-14 แสดงตัวอย่างความสัมพันธ์ของแท็กบนภาพอนิเมะที่สร้างโดยกราฟแบบมีทิศทาง (Directed Graph) โดยแท็ก A -> แท็ก B หมายถึง หากมีแท็ก A ก็จะมีโอกาสที่จะมีแท็ก B ด้วย เช่น หากภาพอนิเมะใดมีแท็กนักเรียน (Student) จะมีโอกาสที่มีแท็กเครื่องแบบ (Uniform) และแท็กเด็กผู้หญิง (Girl) ด้วยเช่นกัน

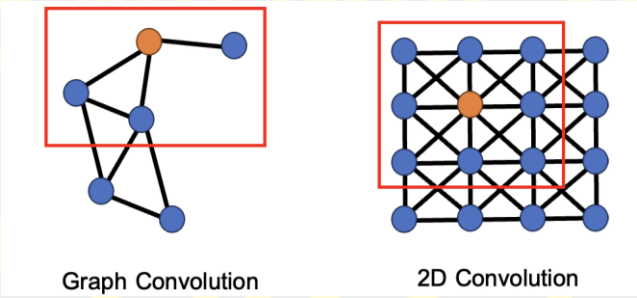


ภาพที่ 2-14 ตัวอย่างความสัมพันธ์ของแท็กบนภาพอนิเมะที่สร้างโดยกราฟแบบมีทิศทาง
อ้างอิง: (P. Deng, Ren, J., Lv, S., Feng, J., & Kang, H., 2020)

โครงข่ายคอนโวลูชันแบบกราฟเป็นโครงข่ายประสาทเทียมชนิดหนึ่งที่สามารถใช้ในการเรียนรู้ความสัมพันธ์ระหว่างข้อมูลนำเข้าบนโครงสร้างที่มีลักษณะเป็นกราฟ แตกต่างจากโครงข่ายประสาทเทียมแบบคอนโวลูชันทั่วไปเพราะข้อมูลนำเข้าไม่ใช่เป็นรูปภาพแต่ใช้เป็นกราฟ โดยกราฟประกอบด้วยโหนด (Node) และเส้นเชื่อมความสัมพันธ์ (T. N. Kipf, & Welling, M., 2017)

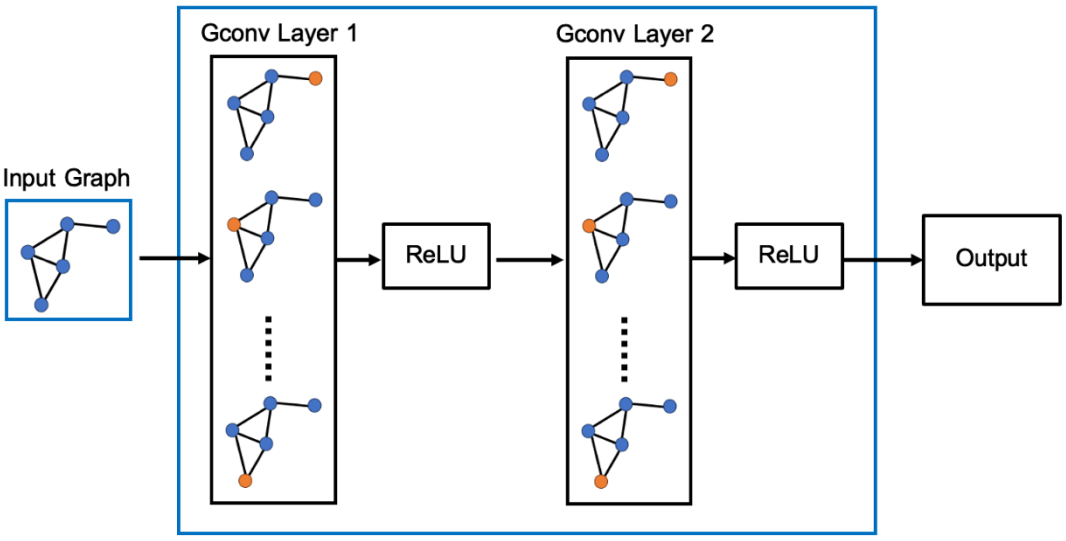
ภาพที่ 2-15 แสดงการเปรียบเทียบกระบวนการคอนโวลูชันของกราฟและรูปภาพ 2D กระบวนการคอนโวลูชันสำหรับรูปภาพและกราฟมีความคล้ายกันคือ คอนโวลูชันสำหรับกราฟจะมองโหนดอื่นที่เชื่อมต่อกับตนเป็นโหนดเพื่อนบ้าน (Neighbor Node) และใช้ค่าเฉลี่ยของโหนดตัวเอง

(โหนดสีส้ม) และโหนดเพื่อนบ้านในการคำนวณ ส่วนคอนโวลูชันสำหรับรูปภาพจะมองทุกพิกเซลเป็นโหนด โหนดเพื่อนบ้านจะถูกกำหนดตามขนาดของฟิวเตอร์ ค่าน้ำหนักเฉลี่ยของโหนดสีส้มและโหนดเพื่อนบ้านจะถูกนำไปคำนวณ กระบวนการคอนโวลูชันสำหรับรูปภาพและกราฟมีความแตกต่างกันคือ โหนดเพื่อนบ้านในกราฟจะมีลำดับและจำนวนที่ไม่แน่นอน แต่โหนดเพื่อนบ้านในรูปภาพจะมีลำดับและจำนวนที่แน่นอน (Z. Wu, Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S., 2019)



ภาพที่ 2-15 การเปรียบเทียบกระบวนการคอนโวลูชันของกราฟและรูปภาพ 2D

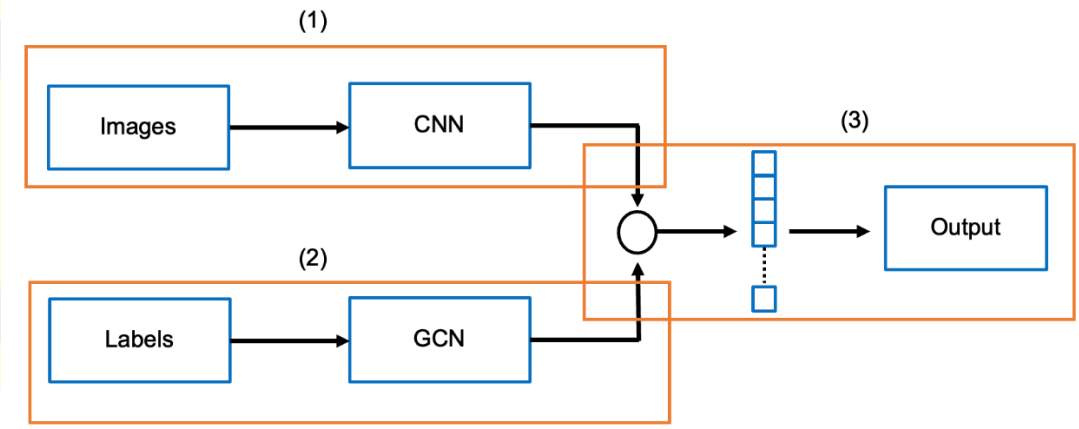
ภาพที่ 2-16 แสดงโครงสร้างของโครงข่ายคอนโวลูชันแบบกราฟ โดยโครงข่ายคอนโวลูชันแบบกราฟเริ่มต้นโดยการนำข้อมูลนำเข้าที่เป็นกราฟเข้าสู่ชั้นของกระบวนการคอนโวลูชันแบบกราฟ (Graph Convolution Layer: Gconv) โดยอาจมีชั้น Gconv ได้มากกว่า 1 ชั้น โดยในแต่ละชั้น Gconv จะมีการคำนวณทุกโหนดโดยในแต่ละโหลดจะมีการคำนวณค่าเฉลี่ยของโหนดตัวเองและโหนดเพื่อนบ้าน สามารถใส่ฟังก์ชันลีลูหรือฟังก์ชันอื่นต่อจากชั้น Gconv ได้



ภาพที่ 2-16 ตัวอย่างโครงสร้างของโครงข่ายคอนโวลูชันแบบกราฟ

2.4 งานวิจัยหรือบทความที่เกี่ยวข้อง

ผู้วิจัยศึกษาบทความทางวิชาการต่าง ๆ เพื่อนำความรู้มาประยุกต์ใช้สำหรับโมเดลสำหรับแท็กภาพอนิเมะ โดยงานวิจัยของ Pengfei Deng และคณะ (P. Deng, Ren, J., Lv, S., Feng, J., & Kang, H., 2020) ได้นำเสนอการพัฒนาโมเดลสำหรับแท็กภาพอนิเมะโดยใช้โครงข่ายคอนโวลูชันแบบกราฟ โดยโมเดลดังกล่าวถูกแบ่งออกเป็น 3 ส่วนหลัก ๆ คือ (1) ส่วนการเรียนรู้ภาพ ซึ่งเป็นส่วนที่ใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันในการเรียนรู้คุณลักษณะของภาพ (2) ส่วนการเรียนรู้แท็กของภาพ ซึ่งเป็นส่วนที่ใช้โครงข่ายคอนโวลูชันแบบกราฟในการเรียนรู้ความสัมพันธ์และความเกี่ยวข้องกันของแท็กต่าง ๆ (3) ส่วนที่นำผลลัพธ์ของส่วนการเรียนรู้ภาพและส่วนการเรียนรู้แท็กของภาพมารวมเข้าด้วยกัน ดังตัวอย่างในภาพที่ 2-17



ภาพที่ 2-17 โครงสร้างหลักของโมเดลแท็กภาพอนิเมะ

สามปีหลังจากงานวิจัยของ Pengfei Deng และคณะ มีงานวิจัยที่นำเสนอโดย Ziwen Lan และคณะ (Z. Lan, Maeda, K., Ogawa, T., & Haseyama, M., 2023) นำเสนอวิธีการเพิ่มประสิทธิภาพโมเดลในการแท็กภาพอนิเมะที่ใช้โครงข่ายคอนโวลูชันแบบกราฟดังต่อไปนี้ (1) ใช้ข้อมูลสอนจากทั้งภาพการ์ตูนสไตล์อนิเมะและภาพจริง เพื่อให้โมเดลมีขอบเขตการเรียนรู้ที่กว้างขึ้น โดยภาพจริงจะถูกนำมาเปลี่ยนสไตล์ให้เป็นแบบการ์ตูนสไตล์อนิเมะก่อนนำเข้าโมเดล (2) กระบวนการของโครงข่ายคอนโวลูชันแบบกราฟมีการเชื่อมโยงความสัมพันธ์ของแท็กต่าง ๆ แท็กเหล่านี้ถูกมองว่าอยู่ในระดับที่เท่าเทียมกัน แต่แท็กเหล่าสามารถนำมาจัดเป็นลำดับชั้นได้ (Hierarchy) ว่า แท็กใดเป็นพ่อแม่ (Parent) แท็กใดเป็นลูก (Child) เพื่อเพิ่มความแม่นยำให้แก่โมเดล

นอกจากการนี้ยังมีงานวิจัยอื่นที่พัฒนาโมเดลในการแท็กภาพอนิเมะโดยไม่ใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันและโครงข่ายคอนโวลูชันแบบกราฟ เช่น งานวิจัยของ Fan Yi

และคณะ (F. Yi, Wu, J. Zhao, M., & Zhou, S., 2023) โมเดลในการแท็กภาพอนิเมะถูกแบ่งออกเป็น 3 ส่วนหลัก ๆ เช่นกัน แต่ส่วนการเรียนรู้ภาพใช้ ViT-L แทนการใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน และส่วนการเรียนรู้แท็กของภาพใช้ BERT แทนการใช้โครงข่ายคอนโวลูชันแบบกราฟ นอกจากนี้งานวิจัยดังกล่าวใช้สมการ TF-IDF (Term Frequency - Inverse Document Frequency) เพื่อใช้พิจารณาความสำคัญของแท็ก โดยสมการนี้ผู้วิจัยสามารถนำมาประยุกต์ใช้ได้

อย่างไรก็ตามจากการศึกษาค้นคว้าข้อมูลเกี่ยวกับการพัฒนาโมเดลในการแท็กภาพอนิเมะ พบว่าโมเดลในการแท็กภาพอนิเมะมักประกอบด้วย 2 ส่วนหลัก ๆ คือ ส่วนการเรียนรู้ภาพและส่วนการเรียนรู้แท็กของภาพ แต่ยังไม่มีงานวิจัยใดที่ใช้อัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันหลายแบบเพื่อเปรียบเทียบว่าอัลกอริทึมใดมีความเหมาะสำหรับการพัฒนาโมเดลสำหรับแท็กภาพอนิเมะ

ดังนั้นในงานวิจัยนี้ผู้วิจัยสร้างโมเดลโดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน 3 แบบ ได้แก่ ResNET ซึ่งเป็นอัลกอริทึมที่ใช้ในโมเดล ML-GCN และใช้ ResNeXT กับ EfficientNet ซึ่งเป็นอัลกอริทึมโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ใหม่กว่า ResNET เพื่อวิเคราะห์ว่าอัลกอริทึมใดให้ผลการทำนายแท็กของภาพอนิเมะได้ถูกต้องมากกว่า

โมเดลที่ผู้วิจัยประยุกต์ใช้ในงานวิจัยนี้เพื่อแท็กภาพอนิเมะ คือ โมเดล ML-GCN ซึ่งเป็นโมเดลที่เปิดให้สามารถดาวน์โหลดได้ฟรีบนเว็บไซต์ Github และถูกเผยแพร่โดย Zhao-Min Chen และคณะ (Z.-M. Chen, Wei, X.-S., Wang, P., & Guo, Y., 2019) โมเดลนี้ใช้เทคนิคการเรียนรู้เชิงลึกเพื่อสอนให้คอมพิวเตอร์จดจำและจำแนกลักษณะต่าง ๆ ของรูปภาพที่ประกอบด้วยมากกว่า 1 แท็ก ซึ่งโมเดลนี้มีโครงสร้างคล้ายคลึงกับโมเดลแท็กภาพอนิเมะที่นำเสนอในงานวิจัยของ Pengfei Deng และคณะ (P. Deng, Ren, J., Lv, S., Feng, J., & Kang, H., 2020)

บทที่ 3
รายละเอียดของการดำเนินงานวิจัย

งานวิจัยนี้มีจุดประสงค์เพื่อประยุกต์ใช้โมเดล ML-GCN และเปรียบเทียบอัลกอริทึมโครงข่ายประสาทเทียมแบบคอนโวลูชันทั้ง 3 แบบ ได้แก่ ResNET ResNeXT และ EfficientNet เพื่อวิเคราะห์ว่าอัลกอริทึมใดเหมาะแก่การใช้ร่วมกับโมเดล ML-GCN ในการแท็กภาพอนิเมะ โดยเนื้อหาในบทนี้จะกล่าวเกี่ยวกับ (1) แหล่งข้อมูลที่ผู้วิจัยดาวน์โหลดข้อมูลภาพอนิเมะ (2) การเตรียมข้อมูลโดยการทำความสะอาดข้อมูล การคัดแยกแท็ก การสร้างแท็กใหม่ การคัดเลือกแท็กที่สำคัญ และการตรวจสอบรูปภาพด้วยตนเอง (3) การเตรียมโมเดลและการแปลงข้อมูลเพื่อใช้ในโมเดล และ (4) วิธีการทดลอง

3.1 การค้นหาแหล่งข้อมูลภาพอนิเมะ

ผู้วิจัยใช้ข้อมูลรูปภาพอนิเมะ danbooru2021 ซึ่งเป็นข้อมูลรูปภาพอนิเมะจากเว็บไซต์ danbooru ซึ่งเป็นเว็บไซต์ที่เปิดให้ผู้ใช้ทั่วไปนำภาพอนิเมะอัปโหลดลงเว็บไซต์ได้ ข้อมูลนี้เปิดให้สามารถดาวน์โหลดข้อมูลได้ฟรีบนเว็บไซต์ Kaggle ผ่าน https://www.kaggle.com/datasets/alamson/safebooru ซึ่งข้อมูลที่ดาวน์โหลดมาอยู่ในรูปแบบไฟล์ CSV และมีข้อมูลทั้งหมด 3,020,460 แถว และ 9 คอลัมน์ ดังตัวอย่างในภาพที่ 3-1 และ 3-22 ข้อมูลภาพอนิมะเหล่านี้มีแท็กที่ไม่ซ้ำกันทั้งหมด 427,578 แท็ก

	id	created_at	rating	score	sample_url	sample_width
0	1	1264803292	s	37	//safebooru.org/samples/1/sample_e7b3dc281d431...	850
1	2	1264803292	s	12	//safebooru.org/samples/1/sample_27ff11b17a2c3...	850
2	3	1264803298	s	8	//safebooru.org/samples/1/sample_ebd16eb1d1547...	850
3	4	1264803299	s	5	//safebooru.org/samples/1/sample_6fbb9a4b9099e...	850

ภาพที่ 3-1 ข้อมูลภาพอนิเมะในไฟล์ CSV (ครึ่งซ้าย)

sample_height	preview_url	tags
638	//safebooru.org/thumbnails/1/thumbnail_e7b3dc2...	1girl bag black_hair blush bob_cut bowieknife ...
1208	//safebooru.org/thumbnails/1/thumbnail_27ff11b...	barding black cape celty_sturluson dress dulla...
599	//safebooru.org/thumbnails/1/thumbnail_ebd16eb...	blue_eyes blush brown_hair original scan takoy...
519	//safebooru.org/thumbnails/1/thumbnail_6fbb9a4...	game_cg hagall_valkyr mecha_musume shirogane_n...

ภาพที่ 3-2 ข้อมูลภาพอนิเมะในไฟล์ CSV (ครึ่งขวา)

ภาพที่ 3-1 และ 3-2 มีรายละเอียดคอลัมน์ดังต่อไปนี้

- id คือ รหัสภาพอนิเมะ
- created_at คือ เวลาที่อัปโหลดภาพอนิเมะ
- rating คือ การจำกัดอายุ ได้แก่
  - s (Safe) คือ ภาพที่ไม่มีเนื้อหา 18+
  - e (Explicit) คือ ภาพที่มีเนื้อหา 18+
  - q (Questionable) คือภาพที่อาจเป็นได้ทั้ง Safe และ Explicit
- score คือ คะแนน ซึ่งสามารถมีคะแนนติดลบได้
- sample_url คือ URL ภาพอนิเมะ
- sample_width คือ ความกว้างภาพอนิเมะ
- sample_height คือ ความสูงภาพอนิเมะ
- preview_url คือ URL ภาพตัวอย่าง (มีขนาดเล็กกว่าภาพจาก sample_url)
- tag คือ แท็กของภาพอนิเมะ ซึ่งในกรณีที่มีหลายแท็กจะใช้พื้นที่ว่าง (Whitespace) ในการแยกแท็กแต่ละแท็กออกจากกัน

3.2 การเตรียมข้อมูล (Data Preparation)

ผู้วิจัยใช้ข้อมูล danbooru2021 ซึ่งเป็นข้อมูลภาพอนิเมะจากเว็บไซต์ซึ่งผู้อัปโหลดภาพอนิเมะสามารถกำหนดแท็กให้แก่รูปภาพเองได้ ดังนั้นแท็กเหล่านี้จึงมีบางแท็กที่สะกดผิด

ไม่สื่อความหมาย ความหมายซ้ำกับแท็กอื่น กำหนดแท็กไม่ครบถ้วน ฯลฯ ผู้วิจัยต้องตรวจสอบความถูกต้องของแท็กของภาพอนิเมะเหล่านี้เพื่อให้มั่นใจว่าแท็กมีความถูกต้องจริงก่อนนำเข้าโมเดล

ผู้วิจัยใช้โปรแกรม Jupyter Notebook และภาษา Python ช่วยตรวจสอบความถูกต้องและกลั่นกรองข้อมูลภาพอนิเมะและข้อมูลแท็กในเบื้องต้นเพื่อแบ่งเบาภาระของผู้วิจัย ในขั้นตอนสุดท้ายผู้วิจัยตรวจสอบภาพทีละภาพด้วยตนเองเพื่อให้มั่นใจในความถูกต้องของแท็ก โดยมีรายละเอียดขั้นตอนดังต่อไปนี้

3.2.1 การทำความสะอาดข้อมูลเบื้องต้น

- ลบแถวที่ภาพอนิเมะได้คะแนนติดลบ
- ลบแถวที่มีค่าหาย (Missing Data) ปรากฏในคอลัมน์แท็กและคอลัมน์ URL
- แก้ไขข้อมูล URL ภาพอนิเมะ เพราะบาง URL ไม่ขึ้นต้นด้วย “https://” แต่ขึ้นต้นด้วย “//”
- ลบแถวที่ข้อมูลในในคอลัมน์ URL ไม่ลงท้ายด้วย .png .jpg หรือ .jpeg

3.2.2 การคัดแยกแท็ก

- ลบแถวที่มีแท็กบ่งบอกว่ามีตัวละครหญิงหรือชายมากกว่า 1 คนออก เพราะผู้วิจัยสร้างโมเดลเพื่อแท็กภาพอนิเมะที่มีตัวละครเพียง 1 คนเท่านั้น
- ลบแถวที่มีแท็กบ่งบอกว่าไม่ใช่ภาพอนิเมะปกติ เช่น มังงะ (Manga/Comic) ภาพสเก็ต (Sketch) รูปถ่าย (Photo) ภาพซ้อนแบบซูม (Zoom-layer) ภาพออกแบบ (Design/Reference) และ ภาพแสดงสีหน้าตัวละครหลายหน้า (Expression)
- นำแท็กที่เป็นสัญลักษณ์ออก เช่น >_< , =_= , ^3^ เป็นต้น
- นำแท็กที่บ่งบอกลักษณะสีออก เช่น หมวกสีแดง (red-hat) และหมวกสีเหลือง (yellow-hat) ให้เหลือเพียงคำว่า หมวก (hat)
- ลบแถวที่มีจำนวนแท็กตั้งแต่ 20 แท็กขึ้นไปออก เพราะภาพที่มีจำนวนแท็กมากมักเป็นภาพที่มีหลายตัวละคร

3.2.3 การดาวน์โหลดภาพอนิเมะ

หลังจากขั้นตอนการกลั่นกรองแท็ก ภาพอนิเมะลดจาก 3,020,460 ภาพเหลือเพียง 1,060,144 ภาพ ผู้วิจัยสุ่มเลือกภาพอนิเมะ 100,000 ภาพเพื่อใช้ในงานวิจัยและเริ่มดาวน์โหลดภาพ

หลังจากการดาวน์โหลดภาพอนิเมะ ผู้วิจัยต้องการแปลงขนาดของภาพอนิเมะทุกภาพเป็นชนิด jpg และปรับขนาดเป็น 500x500 พิกเซล ผู้วิจัยจึงตรวจสอบขนาดของภาพและลบภาพที่อัตราส่วน สูง/กว้าง ไม่อยู่ระหว่าง 1.8 (1270/720) และ 0.5 (720/1270) เพราะภาพเหล่านี้จะเสียหายหลังแปลงขนาดเป็น 500x500 พิกเซล หลังจากการปรับขนาดภาพและการลบภาพที่ไม่เหมาะแก่การปรับภาพออก ภาพอนิเมะเหลือ 68,940 ภาพ

3.2.4 การสร้างแท็กใหม่

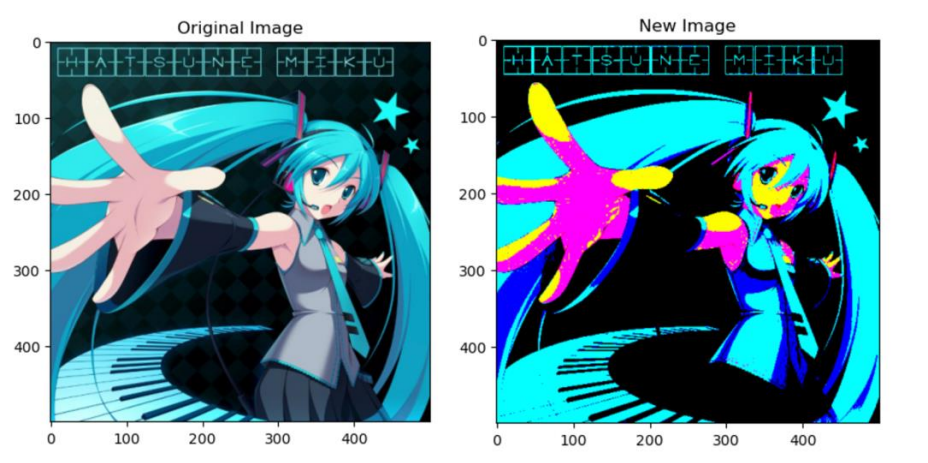
ผู้วิจัยสร้างแท็กเพิ่มอีก 2 หมวดหมู่ คือ แท็กที่ระบุโทนสีโดยรวมของภาพ เพื่อให้ผู้ที่เข้ามาหาแรงบันดาลใจสามารถค้นหาภาพที่ใช้โทนสีในการสะท้อนอารมณ์ด้านต่าง ๆ ได้ง่ายขึ้น และแท็กลักษณะการเน้น เพื่อให้ผู้ที่เข้ามาหาแรงบันดาลใจสามารถเลือกค้นหาภาพที่เน้นตัวละครหรือเน้นฉากหลังเป็นหลักได้ โดยมีรายละเอียดการสร้างดังนี้

- แท็กที่ระบุโทนสีโดยรวมของภาพ
  - แท็กนี้สร้างโดยแปลงสีบนภาพอนิเมะให้เป็นสีพื้นฐานเพื่อลดความหลากหลายของสีและทำให้เห็นสัดส่วนของสีได้ชัดเจนมากขึ้น กรณีที่บนภาพอนิเมะใดมีสีพื้นฐานที่รวมกันได้สัดส่วนบนภาพตั้งแต่ 40% ขึ้นไป หมายความว่าสีดังกล่าวเป็นโทนสีโดยรวมของภาพอนิเมะนั้น
  - สีพื้นฐานที่ผู้วิจัยกำหนดประกอบด้วย สีปฐมภูมิ (Primary Colors) สีทุติยภูมิ (Secondary Colors) และสีดำ รวมทั้งหมด 7 สี ได้แก่ สีแดง เขียว น้ำเงิน สีฟ้า สีม่วงแดง สีเหลือง และสีดำ
  - ในขั้นตอนการแปลงสีภาพ สีบนภาพอนิเมะจะถูกแปลงเป็นสีพื้นฐานที่มีค่าสีใกล้เคียงกับค่าสีเดิมที่สุด โดยใช้สมการ Euclidean Distance เพื่อหาว่าสีบนภาพอนิเมะใกล้เคียงกับสีพื้นฐานใด โดยคำนวณได้ดังสมการที่ (3)

$$d(p,q) = \sqrt{(R_q - R_p)^2 + (G_q - G_p)^2 + (B_q - B_p)^2} \qquad (3)$$

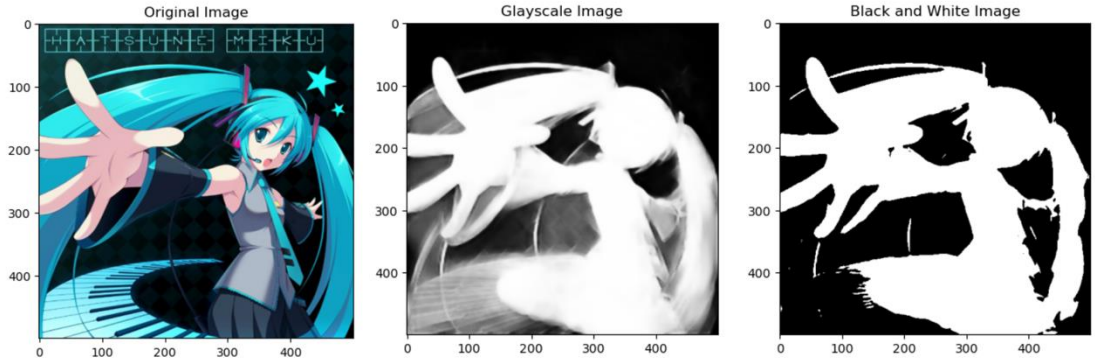
- d คือ ระยะทางระหว่างค่าสี
- p คือ สีพื้นฐาน
- q คือ สีของ Pixel บนภาพอนิเมะ
- R คือ ค่าสีแดง เป็นหนึ่งในค่าของ RGB
- G คือ ค่าสีเขียว เป็นหนึ่งในค่าของ RGB
- B คือ ค่าสีฟ้า เป็นหนึ่งในค่าของ RGB

- ภาพที่ 3-3 คือ ตัวอย่างภาพอนิเมะที่ถูกแปลงสีเป็นสีพื้นฐาน ซึ่งประกอบด้วยสัดส่วนสีพื้นฐานดังนี้
  - black: 52.20%
  - cyan: 30.04%
  - magenta: 7.29%
  - blue: 6.19%
  - yellow: 4.10%
  - red: 0.16%
  - green: 0.02%



ภาพที่ 3-3 ตัวอย่างภาพอนิเมะที่ถูกแปลงสีเป็นสีพื้นฐาน

- แท็กลักษณะการเน้น
  - แท็กนี้สร้างโดยการเทียบสัดส่วนตัวละครกับพื้นที่ทั้งหมดบนภาพ ผู้วิจัยใช้โมเดล Anime Segmentation ของ SkyTNT เพื่อใช้แยกฉากหลังและตัวละครบนภาพอนิเมะออกจากกัน โมเดลนี้จะแปลงภาพอนิเมะเป็นภาพ Greyscale ซึ่งตัวละครจะมีสีขาวและเทาในขณะที่ฉากหลังมีสีดำ ผู้วิจัยนำภาพ Greyscale ที่ได้จากโมเดล Anime Segmentation แปลงเป็นภาพขาวดำ โดยกำหนดค่า Threshold เป็น 200 ดังภาพที่ 3-4 ซึ่งภาพนี้ประกอบด้วยสัดส่วนสีขาว 47.80% และสีดำ 52.20% ของพื้นที่บนภาพ



ภาพที่ 3-4 ตัวอย่างภาพอนิเมะที่ถูกแปลงเป็นภาพ Greyscale และภาพขาวดำ

- ในกรณีภาพอนิเมะมีสัดส่วนสีขาวครอบคลุมพื้นที่ตั้งแต่ 20% ของภาพลงไป หมายความว่าภาพนั้นเน้นฉากหลังและถูกเพิ่มแท็ก “focus_background” ส่วนภาพอนิเมะที่ไม่มีแท็ก “focus_background” หมายความว่าภาพนั้นเน้นตัวละคร
- โมเดล Anime Segmentation ของ SkyTNT สามารถดาว์โหลดได้ฟรีจาก https://github.com/SkyTNT/anime-segmentation/tree/main?tab=readme-ov-file

ผู้วิจัยตั้งใจสร้างแท็กเพิ่ม 8 แท็ก ประกอบด้วยแท็กจากหมวดหมู่แท็กที่ระบุโทนสีโดยรวม จำนวน 7 แท็ก และ แท็กจากหมวดหมู่แท็กลักษณะการเน้น จำนวน 1 แท็ก ซึ่งประกอบด้วยแท็กดังนี้ red_main green_main, blue_main, cyan_main, magenta_main, yellow_main, black_main และ focus_background แต่จากการตรวจสอบภาพอนิเมะที่ผู้วิจัยใช้พบว่าภาพอนิเมะที่สีโดยรวมเป็นสีเขียวและสีน้ำเงินมีจำนวนน้อย ผู้วิจัยจึงไม่ใช้แท็ก green_main และ blue_main ทำให้แท็กที่ผู้วิจัยสร้างเหลือเพียงแค่ 6 แท็ก

3.2.5 การคัดเลือกแท็กที่สำคัญ

เนื่องจากจำนวนแท็กที่ไม่ซ้ำกันของภาพอนิเมะมีจำนวนมาก ผู้วิจัยจึงกลั่นกรองเลือกเฉพาะแท็กที่สำคัญ ในขั้นตอนแรกผู้วิจัยคัดเลือกเฉพาะแท็กที่พบในภาพอนิมะตั้งแต่ 1,000 ภาพเป็นต้นไป และใช้สมการ TF-IDF เพื่อหาค่าความสำคัญของแท็กของที่คัดเลือกไว้

สมการ TF-IDF เป็นสมการที่นิยมใช้ในการหาค่าความสำคัญของคำบนเอกสาร ซึ่งงานวิจัยของ Fan Yi และคณะ (F. Yi, Wu, J. Zhao, M., & Zhou, S., 2023) ได้ประยุกต์ใช้สมการดังกล่าวในการหาค่าความสำคัญของแท็กบนภาพอนิเมะ โดยคำนวณได้ดังสมการที่ (4)

$$\sum^{c_i} TF \times IDF_i = \left(\sum^{c_i} \frac{1}{k_i}\right) \times log\frac{C}{c_i} \quad (4)$$

- i คือ แท็กที่สนใจ
- c_i คือ จำนวนของภาพอนิเมะที่มีแท็ก i
- k_i คือ จำนวนของแท็กทั้งหมดของภาพอนิเมะที่มีแท็ก i
- C คือ จำนวนของภาพอนิเมะทั้งหมด
- TF (Term Frequency) คือ ค่าความถี่ของแท็ก i ที่พบในภาพอนิเมะ ซึ่งคำนวณได้จากการนำจำนวนแท็ก i ที่พบในภาพอนิมะ (มีค่าเป็น 1 เสมอ) หารด้วย k_i ซึ่งเป็นจำนวนของแท็กทั้งหมดของภาพอนิเมะที่มีแท็ก i
- ∑TF คือ ผลรวมของค่า TF ที่คำนวณจากภาพอนิเมะที่พบแท็ก i ทีละภาพรวมกัน ยิ่งมีค่ามากแปลว่าแท็กนั้นมีความสำคัญ
- DF_i (Document Frequency) คือ ค่าความถี่ของภาพอนิเมะที่มีแท็ก i ซึ่งยิ่งค่าน้อยแปลว่าแท็กนั้นมีความสำคัญ ซึ่งคำนวณได้จากการนำ C ซึ่งเป็นจำนวนของภาพอนิเมะทั้งหมดหารด้วย c_i ซึ่งเป็นจำนวนของภาพอนิเมะที่มีแท็ก i
- IDF_i (Inverse Document Frequency) คือ ค่าผกผันของ DF_i ซึ่งเหตุผลที่ใช้ค่าผกผันเพื่อเปลี่ยนจาก “ยิ่งมีค่าน้อยแปลว่าแท็กนั้นมีความสำคัญ” เป็น “ยิ่งมีค่ามากแปลว่าแท็กนั้นมีความสำคัญ”

หลังจากใช้สมการ TF-IDF ผู้วิจัยคัดเลือกแท็กที่สำคัญออกมา 10 แท็ก เมื่อรวมกับแท็กที่ผู้วิจัยสร้างเพิ่มในขั้นตอนที่ 3.2.4 จำนวน 6 แท็ก แท็กทั้งหมดที่ผู้วิจัยใช้จึงมีทั้งหมด 16 แท็ก ผู้วิจัยลบแท็กอื่นนอกเหลือจากแท็กที่คัดเลือกไว้ออกจากภาพอนิเมะทั้งหมดและลบภาพอนิเมะที่ไม่เหลือแท็กออก จากนั้นผู้วิจัยนำภาพอนิเมะที่เหลือผ่านกระบวนการ Domnsampling เพื่อลดความต่างของจำนวนแท็กทั้ง 16 แท็ก และได้ผลลัพธ์เป็นภาพอนิเมะทั้งหมด 9,904 ภาพ

3.2.6 การตรวจสอบรูปภาพด้วยต้นเอง

หลังจากการใช้ภาษา Python ช่วยตรวจสอบและกลั่นกรองข้อมูลภาพอนิเมะจนเหลือภาพอนิเมะทั้งหมด 9,904 ภาพ และ 16 แท็ก ผู้วิจัยไล่ตรวจสอบภาพเหล่านี้ทีละภาพด้วยตนเองเพื่อให้มั่นใจในความถูกต้องและครบถ้วนของแท็กที่ภาพอนิเมะเหล่านี้มี และลบภาพอนิเมะที่มีหลายตัวละครหรือภาพอนิเมะที่ไม่ปกติออก ท้ายที่สุดเหลือภาพอนิเมะทั้งหมด 9,392 ภาพ

3.3 การเตรียมโมเดลและการแปลงข้อมูลเพื่อใช้ในโมเดล

3.3.1 การเตรียมโมเดล

ผู้วิจัยปรับปรุงโมเดล ML-GCN ให้สามารถใช้งานบน Google Colab Pro+ และสามารถใช้งานร่วมกับอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันทั้ง 3 แบบ ได้แก่ ResNET ResNeXT และ EfficientNet โดย Runtime Type ของ Google Colab Pro+ ผู้วิจัยใช้เป็น A100 GPU

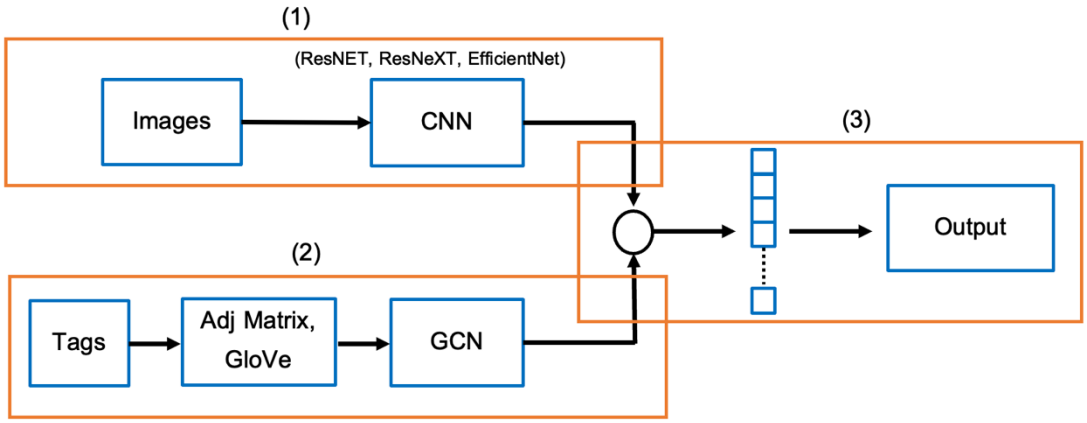
ผู้วิจัยใช้โมเดล ML-GCN ซึ่งถูกนำเสนอโดย Zhao-Min Chen และคณะ (2019) โมเดลนี้ใช้สอนคอมพิวเตอร์ให้เรียนรู้การจำแนกข้อมูลรูปภาพที่ประกอบด้วยมากกว่า 1 แท็ก (Multi-label Classification) โมเดลนี้เปิดให้สามารถดาวน์โหลดได้ฟรีบนเว็บไซต์ Github ผ่าน https://github.com/megvii-research/ML-GCN

โมเดล ML-GCN มีโครงสร้างแบ่งออกเป็น 3 ส่วนหลัก ได้แก่

1) ส่วนการเรียนรู้ภาพ ซึ่งเป็นส่วนที่ใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันเพื่อเรียนรู้คุณลักษณะของภาพ โดยอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ผู้วิจัยใช้ ได้แก่ ResNET-101 ResNeXT-101 และ EfficientNetB5 โดย ResNET-101 เป็นอัลกอริทึมที่ถูกใช้ในโมเดล ML-GCN ผู้วิจัยเลือกใช้ ResNeXT-101 และ EfficientNetB5 เพราะขนาดของข้อมูลเอาท์พุท (ก่อนผ่าน Fully Connected Layer) มีขนาด 2048 แบบเดียวกับ ResNET-101
2) ส่วนการเรียนรู้แท็กของภาพ ซึ่งเป็นส่วนที่ใช้โครงข่ายคอนโวลูชันแบบกราฟเพื่อเรียนรู้ความสัมพันธ์และความเกี่ยวข้องกันของแท็กต่าง ๆ ซึ่งแท็กของภาพต้องแปลงเป็นค่าเวกเตอร์ความสัมพันธ์โดยใช้อัลกอริทึม GloVe และแปลงเป็นกราฟรูปแบบ Adj Matrix ก่อนนำเข้าใช้ในโมเดล เพราะโครงข่ายคอนโวลูชันแบบกราฟต้องใช้ข้อมูลนำเข้าในรูปแบบกราฟ

3) ส่วนที่นำผลลัพธ์ของส่วนการเรียนรู้ภาพและส่วนการเรียนรู้แท็กของภาพมารวมเข้าด้วยกันโดยใช้การคูณ เมื่อใช้โมเดลทำนายแท็กให้แก่รูปภาพ โมเดลจะทำนายแท็กของรูปภาพดังกล่าวว่ามีความน่าจะเป็นที่จะประกอบด้วยแท็กเหล่านี้เท่าใดแบบรายแท็ก

ภาพรวมของโมเดล ML-GCN ที่ใช้ในงานวิจัยแสดงในรูปภาพที่ 3-5



ภาพที่ 3-5 ภาพรวมของโมเดล ML-GCN ที่ใช้ในงานวิจัย

3.3.2 การแปลงข้อมูลแท็กเพื่อใช้ในโมเดล

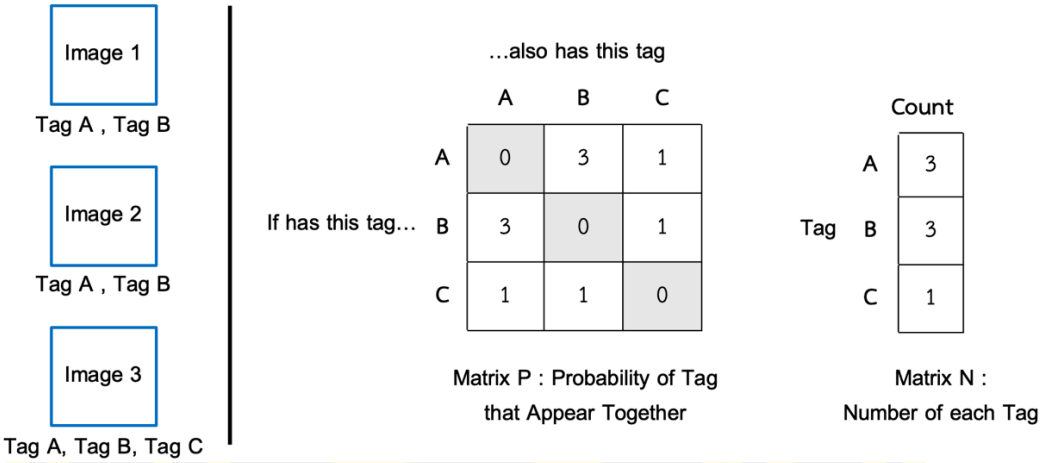
การใช้โครงข่ายคอนโวลูชันแบบกราฟเพื่อเรียนรู้ความสัมพันธ์และความเกี่ยวข้องกันของแท็กต้องแปลงแท็กเป็นค่าเวกเตอร์ความสัมพันธ์โดยใช้อัลกอริทึม GloVe และแปลงเป็นกราฟรูปแบบ Adj Matrix โดย GloVe และ Adj Matrix มีความหมายดังต่อไปนี้

- GloVe (Global Vectors for Word Representation) คืออัลกอริทึมที่สามารถใช้หาค่าความสัมพันธ์ของคำ (Words) โดยการแปลงค่าเหล่านั้นเป็นค่าเวกเตอร์ (Vector) หากค่าเวกเตอร์ของคำใกล้กันมากแสดงว่าคำเหล่านั้นมีความสัมพันธ์เกี่ยวข้องกันมาก อัลกอริทึมนี้สามารถใช้หาความสัมพันธ์ระหว่างแท็กของภาพอนิเมะได้ เช่น แท็กเหล่านี้มีโอกาสเกิดขึ้นพร้อมกัน เป็นต้น
- Adj Matrix (Adjacency Matrix) คือเมทริกซ์ที่ใช้แทนค่ารูปแบบกราฟความสัมพันธ์ได้ ผู้วิจัยแปลงแท็กของภาพอนิเมะเป็นเมทริกซ์นี้เพื่อใช้เป็นข้อมูลนำเข้าของโครงข่ายคอนโวลูชันแบบกราฟ เมทริกซ์นี้ประกอบด้วยเลข 1 และ 0 โดยเลข 1 หมายถึงแท็กมีความสัมพันธ์กัน และเลข 0 หมายถึงแท็กไม่ความสัมพันธ์กัน

การแปลงแท็กเป็นค่าเวกเตอร์ความสัมพันธ์สามารถทำได้โดยการนำแท็กของภาพอนิเมะทั้งหมดใส่ลงใน อัลกอริทึม GloVe แต่การสร้าง Adj Matrix จากแท็กของภาพอนิเมะมีขั้นตอนที่ซับซ้อนกว่าเพราะต้องสร้างเมทริกซ์ P และเมทริกซ์ N และนำมาหารกันเพื่อสร้าง Adj Matrix

ภาพที่ 3-6 แสดงตัวอย่างเมทริกซ์ P และเมทริกซ์ N ที่ถูกสร้างจากแท็กของภาพอนิเมะโดยสมมติว่ามีภาพอนิเมะทั้งหมด 3 ภาพ และมีแท็กทั้งหมด 3 แบบ ได้แก่ แท็ก A แท็ก B และแท็ก C ซึ่งภาพอนิเมะทั้ง 3 มีรายละเอียดดังต่อไปนี้

- ภาพอนิเมะที่ 1 ประกอบด้วย 2 แท็ก ได้แก่ แท็ก A และ แท็ก B
- ภาพอนิเมะที่ 2 ประกอบด้วย 2 แท็ก ได้แก่ แท็ก A และ แท็ก B
- ภาพอนิเมะที่ 3 ประกอบด้วย 3 แท็ก ได้แก่ แท็ก A แท็ก B และ แท็ก C



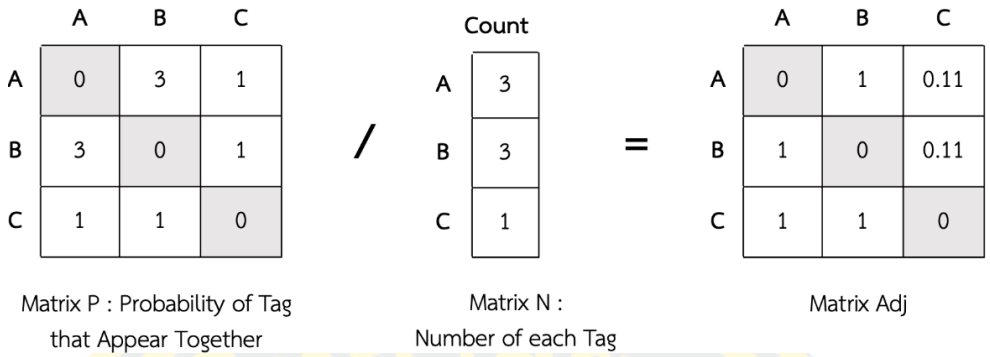
ภาพที่ 3-6 แสดงตัวอย่างเมทริกซ์ P และเมทริกซ์ N ที่ถูกสร้างจากแท็กของภาพอนิเมะ

เมทริกซ์ P คือ เมทริกซ์ความน่าจะเป็นที่แท็กจะเกิดขึ้นพร้อมกัน เส้นทแยงมุมหลักของเมทริกซ์นี้มีค่าเท่ากับ 0 ทั้งหมด ส่วนช่องอื่นมีตัวอย่างการคำนวณดังนี้

- ความน่าจะเป็นหากมีแท็ก A แล้วจะมีแท็ก B คือ 3 เพราะจำนวนภาพอนิเมะที่มีแท็ก A และแท็ก B มีทั้งหมด 3 ภาพ
- ความน่าจะเป็นหากมีแท็ก A แล้วจะมีแท็ก C คือ 1 เพราะจำนวนภาพอนิเมะที่มีแท็ก A และแท็ก C มีทั้งหมด 1 ภาพ

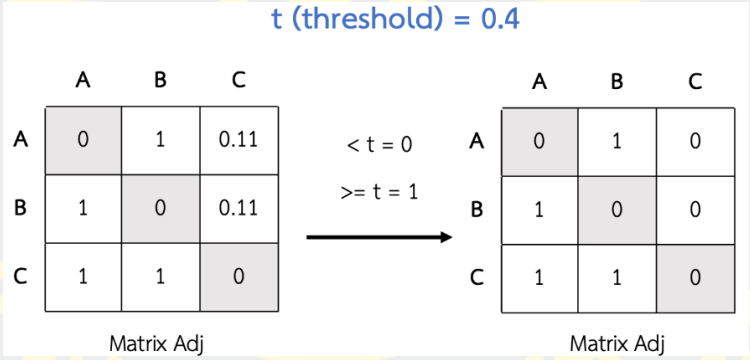
เมทริกซ์ N คือ เมทริกซ์จำนวนภาพอนิเมะที่แต่ละแท็กมี โดยแท็ก A มี 3 ภาพ แท็ก B มี 3 ภาพ และแท็ก C มี 1 ภาพ

Adj Matrix สร้างจากการนำเมทริกซ์ N หารทุกคอลัมน์ของเมทริกซ์ P ซึ่งมีผลลัพธ์ดังภาพที่ 3-7



ภาพที่ 3-7 การสร้าง Adj Matrix โดยการนำเมทริกซ์ N หารเมทริกซ์ P

หลังจากการหารเมทริกซ์ในภาพที่ 3-7 จะต้องนำเมทริกซ์ผลลัพธ์ที่ได้มาแปลงเป็นค่า 0 กับ 1 โดยใช้กำหนดค่า t (threshold) เป็น 0.4 ในกรณีที่ค่าในเมทริกซ์มีน้อยกว่าค่า t จะถูกเปลี่ยนเป็น 0 แต่กรณีที่ค่ามีมากกว่า t จะถูกเปลี่ยนเป็น 1 ดังภาพที่ 3-8



ภาพที่ 3-8 การปรับค่าใน Adj Matrix โดยใช้ค่า Threshold

3.4 วิธีการทดลอง

ผู้วิจัยใช้ภาพอนิเมะทั้งหมด 9,392 ภาพซึ่งประกอบด้วยแท็กที่ไม่ซ้ำกันทั้งหมด 16 แท็กในการรันโมเดล ML-GCN แต่ละแท็กมีภาพอนิเมะไม่น้อยกว่า 700 ภาพ โดยผู้วิจัยแบ่งข้อมูลภาพอนิเมะ 80% เป็นข้อมูลฝึกสอน (7,514 ภาพ) และ 20% เป็นข้อมูลทดสอบ (1,878 ภาพ) ขนาดของภาพอนิเมะที่ใช้สำหรับอัลกอริทึม ResNET-101 และResNeXT-101 คือ 500x500 แต่ EfficientNetB5 ใช้ขนาด 456x456 เพราะเป็นข้อกำหนดของ EfficientNetB5

ผู้วิจัยวัดประสิทธิภาพของโมเดล ML-GCN ที่ใช้ร่วมกับอัลกอริทึมทั้ง 3 แบบของโครงข่ายประสาทเทียมแบบคอนโวลูชันโดยใช้สมการดังต่อไปนี้

3.4.1 สมการ P@k (Precision@k)

สมการ P@k เป็นสมการที่ผู้วิจัยใช้ประเมินค่าความแม่นยำของแท็ก (หรือ Class) ที่โมเดลทายให้กับภาพอนิมะ 1 ภาพว่าถูกต้องเพียงใด โดยสมการนี้จะแตกต่างจากสมการ Precision แบบธรรมดา เพราะสมการนี้ต้องเรียงข้อมูลและคำนวณค่าความแม่นยำจากตำแหน่งของตน (k) และตำแหน่งก่อนหน้าทั้งหมด โดยคำนวณได้ดังสมการที่ (5)

$$\mathrm{P}(k) = \frac{TP@k}{TP@k + FP@k} \tag{5}$$

- *@k* คือ ลำดับของภาพอนิเมะซึ่งถูกเรียงตามค่าความน่าจะเป็น (Probability) ที่โมเดลทายเป็นบวก
- *TP@k* (True Positive@*k*) คือ จำนวนของภาพอนิเมะที่โมเดลทายว่าเป็นบวกและทายถูกโดยนับตั้งแต่ภาพลำดับที่ *k* และภาพลำดับก่อนหน้า *k* ทั้งหมด
- *FP@k* (False Positive@*k*) คือ จำนวนของภาพอนิเมะที่โมเดลทายว่าเป็นบวกแต่ทายผิด โดยนับตั้งแต่ภาพลำดับที่ *k* และภาพลำดับก่อนหน้า *k* ทั้งหมด

3.4.2 สมการ AP (Average Precision)

สมการ AP เป็นสมการที่ผู้วิจัยใช้ประเมินค่าความแม่นยำเฉลี่ยของ 1 แท็ก (หรือ Class) ที่โมเดลทายให้แก่ภาพอนิเมะที่ใช้เป็นข้อมูลทดสอบทั้งหมด (Testing Data) ดังนั้นทุกแท็กจะมีค่า AP เป็นของตัวเอง โดยคำนวณได้ดังสมการที่ (6)

$$\mathrm{AP}_i = \frac{\sum_{k=1}^{n}(\mathrm{P}(k) * rel(k))}{n} \tag{6}$$

- *i* คือ ลำดับของแท็ก (ทุกแท็กต้องคำนวณค่า *AP*)
- *n* คือ จำนวนภาพอนิเมะที่ใช้เป็นข้อมูลทดสอบ
- P(*k*) คือ คำนวณค่า P@*k* ของภาพอนิเมะทีละภาพตามลำดับที่ *k*
- *rel*(*k*) คือ ตัวแปรเงื่อนไข ซึ่งเป็นได้ทั้ง 1 หรือ 0 ในกรณีที่ภาพอนิเมะในเฉลยมีแท็ก ค่าของ *rel*(*k*) จะมีค่าเป็น 1 แต่กรณีในเฉลยไม่มีแท็ก จะมีค่าเป็น 0 ซึ่งหากค่าเป็น 0 จะหมายความว่าไม่ต้องบวกสะสมค่า P(*k*) ของภาพอนิเมะภาพปัจจุบัน เพราะ P(*k*) * 0 จะได้ผลลัพธ์เป็น 0

3.4.3 สมการ mAP (Mean Average Precision)

สมการ mAP เป็นสมการที่ผู้วิจัยใช้ประเมินค่าความแม่นยำเฉลี่ยของโมเดลโดยพิจารณาจากค่าความแม่นยำของทุกแท็กที่ทำนายภาพอนิเมะที่เป็นข้อมูลทดสอบ (นำค่า AP ของทุกแท็กรวมกันและหาค่าเฉลี่ย) โดยคำนวณได้ดังสมการที่ (7)

$$\text{mAP} = \frac{\sum_{i=1}^{j} \text{AP}_i}{j} \tag{7}$$

- j คือ จำนวนแท็กทั้งหมด
- AP_i คือ ค่า AP ของแท็กลำดับที่ i

บทที่ 4

ผลการดำเนินงานวิจัย

ผู้วิจัยประยุกต์ใช้โมเดล ML-GCN และเปรียบเทียบอัลกอริทึมโครงข่ายประสาทเทียมแบบคอนโวลูชันทั้ง 3 แบบ ได้แก่ ResNET-101 ResNeXT-101 และ EfficientNetB5 เพื่อวิเคราะห์ว่าอัลกอริทึมใดเหมาะแก่การใช้ร่วมกับโมเดล ML-GCN ในการแท็กภาพอนิเมะ ผู้วิจัยดำเนินการทดสอบอัลกอริทึมละ 4 ครั้ง ซึ่งแต่ละครั้งมีการปรับเปลี่ยนค่าพารามิเตอร์เพื่อสังเกตผลลัพธ์และการเปลี่ยนแปลงของค่า mAP เนื้อหาในบทนี้จะกล่าวเกี่ยวกับผลการทดสอบปรับค่าอัตราการเรียนรู้ ผลการทดสอบเพิ่มจำนวนรอบ และผลการตรวจสอบค่า AP ของแต่ละแท็ก

4.1 ผลการทดสอบปรับค่าอัตราการเรียนรู้

ผู้วิจัยทดสอบโมเดล ML-GCN โดยใช้อัลกอริทึมทั้ง 3 ได้แก่ ResNET-101 ResNeXT-101 และ EfficientNetB5 ผู้วิจัยรันอัลกอริทึมละ 3 ครั้งและมีการปรับค่าอัตราการเรียนรู้ในระหว่างการรันโมเดล ผู้วิจัยบันทึกเวลาที่ใช้ในการรันโมเดลและจำนวน Compute Unit ที่ใช้ โดยมีพารามิเตอร์ต่าง ๆ ดังนี้

1) Epoch คือ จำนวนรอบ กำหนดเป็น 300 รอบเสมอ
2) LR (Learning Rate) คือ ค่าอัตราการเรียนรู้ กำหนดค่าเริ่มต้นเป็น 0.1 เสมอ
3) Epoch Step คือ การปรับค่าอัตราการเรียนรู้ทุก x รอบ โดยทุก ๆ x รอบที่โมเดลรันอัตราการเรียนรู้จะลดลง 0.1 เท่า เพื่อให้โมเดลเรียนรู้ลึกขึ้นเมื่อผ่านไป x รอบ โดยการทดสอบทั้ง 3 ครั้งของแต่ละอัลกอริทึม กำหนดให้ค่า x มีค่าเป็น 160, 110 และ 80 ตามลำดับ ซึ่งมีจำนวนครั้งในการปรับค่า LR ในการรัน 300 รอบดังต่อไปนี้

    - Epoch Step = 160 มีจำนวนการปรับค่า LR ทั้งหมด 1 ครั้ง คือ รอบที่ 160
    - Epoch Step = 110 มีจำนวนการปรับค่า LR ทั้งหมด 2 ครั้ง ได้แก่ รอบที่ 110 และ 220
    - Epoch Step = 80 มีจำนวนการปรับค่า LR ทั้งหมด 3 ครั้ง ได้แก่ รอบที่ 80, 160 และ 240

Compute Unit คือ จำนวนทรัพยากรที่ใช้ในการประมวลผลเมื่อสั่งรันโมเดลใน Google Colab ซึ่งราคา ณ เดือน ตุลาคม 2014 ของ 100 Compute Unit คือ 343.47 บ. และ 500

Compute Unit คือ 1,677.76 บ. ส่วน Runtime Type ของ Google Colab Pro+ ผู้วิจัยใช้เป็น A100 GPU

ผลการทดลองแสดงในตารางที่ 4-1

ตารางที่ 4-1 การเปรียบเทียบการทดสอบ 3 อัลกอริทึมโดยการรันโมเดล 300 รอบ

ลำดับที่	อัลกอริทึม	Epoch Step	จำนวนการปรับค่า LR	เวลาที่ใช้ (นาที)	Compute Unit	mAP
1	ResNET	160	1	471	92.13	81.70
2		110	2	467	90.41	81.91
3		80	3	468	87.78	**82.47**
4	ResNeXT	160	1	558	107.9	**83.54**
5		110	2	561	109.37	82.88
6		80	3	561	118.32	82.63
7	EfficientNet	160	1	851	163.7	69.47
8		110	2	845	166.43	71.37
9		80	3	852	164.67	**74.34**

ตารางที่ 4-1 แสดงให้เห็นว่าในการรันทั้ง 3 ครั้ง ResNeXT ได้ค่า mAP มากที่สุด ตามด้วย ResNET และ EfficientNet ตามลำดับ ในด้าน Compute Unit และเวลาที่ใช้ในการรันโมเดล EfficientNet ใช้มากที่สุดเมื่อเทียบกับ ResNET และ ResNeXT

การทดสอบปรับค่า LR ในระหว่างการรันโมเดล ผลปรากฏว่าการเพิ่มจำนวนการปรับค่า LR สามารถเพิ่มค่า mAP เล็กน้อย ให้แก่ ResNET และ EfficientNet ส่วน ResNeXT ได้ผลตรงข้าม คือ ค่า mAP ลดลงจาก 83.54 เป็น 82.88 และ 82.63 ตามลำดับ อย่างไรก็ตามไม่ควรปรับค่า LR ในระหว่างการรันหลายครั้ง เพราะค่า LR คูณ 0.1 ทุกครั้ง

4.2 ผลการทดสอบเพิ่มจำนวนรอบ

ผู้วิจัยทดสอบเพิ่มจำนวนรอบการรันเพื่อทดสอบว่าสามารถเพิ่มค่า mAP ให้อัลกอริทึมเหล่านี้ได้หรือไม่ โดยผู้วิจัยเลือก Epoch Step ของแต่ละอัลกอริทึมที่ให้ค่า mAP สูงสุด เพิ่มจำนวน Epoch จาก 300 เป็น 400 และเพิ่มค่า Epoch Step เพื่อรักษาจำนวนการปรับค่า LR ไว้ โดยมีรายละเอียดดังนี้

- ในกรณีที่ Epoch Step = 160 เปลี่ยนเป็น Epoch Step = 220 เพื่อให้มีจำนวนการปรับค่า LR ทั้งหมด 1 ครั้ง คือ รอบที่ 220
- ในกรณีที่ Epoch Step = 110 เปลี่ยนเป็น Epoch Step = 150 เพื่อให้มีจำนวนการปรับค่า LR ทั้งหมด 2 ครั้ง ได้แก่ รอบที่ 150 และ 300
- ในกรณีที่ Epoch Step = 80 เปลี่ยนเป็น Epoch Step = 110 เพื่อให้มีจำนวนการปรับค่า LR ทั้งหมด 3 ครั้ง ได้แก่ รอบที่ 110, 220 และ 330

ผลการทดลองแสดงในตารางที่ 4-2

ตารางที่ 4-2 การทดสอบเพิ่มจำนวนรอบการรันจาก 300 รอบเป็น 400 รอบ

ลำดับที่	อัลกอริทึม	Epoch	Epoch Step	เวลาที่ใช้ (นาที)	Compute Unit	mAP
1	ResNET	300 400	80 110	468 623	87.78 121.94	**82.47** 82.07
2	ResNeXT	300 400	160 220	558 747	107.9 143.78	83.54 **84.07**
3	EfficientNet	300 400	80 110	852 1114	164.67 217.52	74.34 **76.50**

ตารางที่ 4-2 แสดงให้เห็นว่าหลังการเพิ่มจำนวนรอบการารันของทั้ง 3 อัลกอริทึมจาก 300 เป็น 400 รอบ ค่า mAP ใหม่ของ ResNeXT และ EfficientNet เพิ่มขึ้น ส่วน ResNET ลดลง แสดงให้เห็นว่า ResNeXT และ EfficientNet สามารถเพิ่มจำนวนรอบให้ค่า mAP เพิ่มขึ้นได้ อย่างไรก็ตามการเพิ่มจำนวนรอบจำนวนมากไม่สามารถเพิ่มค่า mAP ได้มากเสมอไปเพราะ mAP จะหยุดเพิ่มเมื่อถึงรอบที่ z ซึ่งรอบที่ z มีค่าหลากหลายตามชนิดของอัลกอริทึม พารามิเตอร์ที่ใช้และข้อมูลฝึกสอน

4.3 ผลการตรวจสอบค่า AP ของแต่ละแท็ก

หลังจากทดสอบเพิ่มจำนวนรอบ ผู้วิจัยเลือกพารามิเตอร์ที่ให้ค่า mAP มากที่สุดของแต่ละอัลกอริทึมเพื่อใช้ตรวจสอบ ได้แก่

1. ResNET-101: Epoch=300, Epoch Step=80, mAP=82.47
2. ResNeXT-101: Epoch=400, Epoch Step=220, mAP=84.07
3. EfficientNetB5: Epoch=400, Epoch Step=110, mAP=76.50

ผู้วิจัยตรวจสอบค่า AP ของแต่ละแท็ก (คลาส) ของแต่ละอัลกอริทึมที่เลือกไว้เพื่อเปรียบเทียบความแม่นยำของแท็ก ซึ่งได้ผลลัพธ์ดังตารางที่ 4-3

ตารางที่ 4-3 การตรวจสอบค่า AP ของแต่ละแท็กของทั้ง 3 อัลกอริทึม

ID	แท็ก	ResNET	ResNeXT	EfficientNet
0	bikini	0.71	0.71	0.63
1	black_main	0.93	0.93	0.93
2	chibi	0.77	0.79	0.70
3	closed_eyes	0.80	0.83	0.67
4	cyan_main	0.88	0.90	0.90
5	focus_background	0.63	0.69	0.55
6	long_hair	0.90	0.92	0.86
7	magenta_main	0.87	0.79	0.85
8	navel	0.75	0.81	0.64
9	red_main	0.88	0.89	0.87
10	serafuku	0.68	0.76	0.46
11	short_hair	0.85	0.86	0.79
12	sky	0.81	0.80	0.76
13	white_background	0.96	0.97	0.96
14	wink	0.81	0.84	0.69
15	yellow_main	0.99	0.99	0.98

ตารางที่ 4-3 แสดงให้เห็นว่าค่า AP ของแท็กใน ResNET และ ResNeXT มีค่าใกล้เคียงกัน แต่ส่วนใหญ่ ResNeXT มีค่าสูงกว่า ส่วน EfficientNet แท็กส่วนใหญ่มีค่าน้อยกว่า ResNET และ ResNeXT ยกเว้นแท็กที่เกี่ยวข้องกับสีเพราะมีค่าใกล้เคียงกับค่าของ ResNET และ ResNeXT มาก แสดงให้เห็นว่าความแม่นยำในการทำนายแท็กที่เกี่ยวกับสีของทั้ง 3 อัลกอริทึมมีค่าใกล้เคียงกัน

ข้อสังเกตที่น่าสนใจ คือ การทำนายแท็กโทนสีหลักม่วงแดง (magenta_main) ของ ResNeXT แม่นยำน้อยกว่า ResNET และ EfficientNet ผู้วิจัยตรวจสอบภาพอนิเมะที่โมเดลทายผิดว่ามีความน่าจะเป็นที่มีแท็กโทนสีหลักม่วงแดงสูง พบว่าภาพที่ทายผิดส่วนใหญ่ (ภาพที่ทายผิดซึ่งพบใน ResNeXT แต่ไม่พบใน ResNET และ EfficientNet) เป็นภาพที่สัดส่วนสีขาวมากและสีม่วงแดงที่มีสีอ่อน อาจทำให้โมเดลสับส่วน ดังนั้นในกรณีที่ใช้โมเดล ResNeXT อาจต้องเพิ่มข้อมูลสอนของแท็กสีหลักม่วงแดงให้มากขึ้น

นอกจากนี้ 3 แท็กที่ทั้ง 3 อัลกอรึทึมทายแม่นยำมากที่สุดหรือมีค่า AP มากที่สุด (ไฮไลท์สีเขียว) ได้แก่ yellow_main, black_main และ white_background ส่วน 3 แท็กที่ทายแม่นยำน้อยที่สุดหรือมีค่า AP น้อยที่สุด (ไฮไลท์สีเหลือง) ได้แก่ focus_background, serafuku และ bikini

ผู้วิจัยตรวจสอบภาพอนิเมะที่ทายผิดของแท็ก focus_background, serafuku และ bikini โดยตรวจสอบ 15 ภาพอนิเมะที่โมเดลทายผิดว่ามีความน่าจะเป็นที่มีแท็กเหล่านี้สูงที่สุด ซึ่งคาดว่ามีสาเหตุดังต่อไปนี้

- สาเหตุที่แท็ก bikini (ชุดบิกินี) และ serafuku (ชุดนักเรียนกะลาสี) มีค่า AP น้อย อาจเนื่องมาจากไม่มีแท็กอื่นที่ลักษณะคล้ายแท็กเหล่านี้ในการสอนโมเดล เช่น โมเดลมักทายภาพอนิเมะที่ตัวละครสวมชุดชั้นในว่าเป็นสวมชุดบิกินี และโมเดลมักทายภาพอนิเมะที่ตัวละครสวมชุดนักเรียนธรรมดาว่าเป็นชุดนักเรียนกะลาสี เป็นต้น ผู้วิจัยคาดว่าหากเพิ่มแท็กที่ลักษณะคล้ายแท็กเหล่านี้ในการสอนโมเดล เช่น แท็กชุดชั้นใน (Underwear) และแท็กชุดนักเรียนธรรมดา (Normal School Uniform) อาจช่วยให้ค่าความแม่นยำ หรือ AP ของแท็ก bikini และ serafuku เพิ่มมากขึ้น

- สาเหตุที่แท็ก focus_background มีค่า AP น้อย อาจเนื่องมาจากในขั้นตอนการสร้างแท็ก ผู้วิจัยกำหนดให้ภาพอนิเมะมีสัดส่วนสีขาวครอบคลุมพื้นที่ตั้งแต่ 20% ของภาพลงไป มีแท็ก “focus_background” ซึ่ง 20% อาจยังไม่ชัดเจน ทำให้โมเดลเรียนรู้ยาก หากเปลี่ยนจาก 20% เป็น 10% อาจช่วยให้ค่าความแม่นยำ หรือ AP เพิ่มมากขึ้น

บทที่ 5
สรุปผลดำเนินงานวิจัย

จากการดำเนินงานวิจัยระยะเวลา 17 สัปดาห์ โดยเริ่มตั้งแต่นที่ 1 กรกฎาคม 2566 ผู้วิจัยประยุกต์ใช้โมเดล ML-GCN เพื่อจำแนกแท็กของภาพอนิเมะและเปรียบเทียบอัลกอริทึมโครงข่ายประสาทเทียมแบบคอนโวลูชันทั้ง 3 แบบ ได้แก่ ResNET-101 ResNeXT-101 และ EfficientNetB5 เพื่อวิเคราะห์ว่าอัลกอริทึมใดเหมาะแก่การใช้ร่วมกับโมเดล ML-GCN ในการแท็กภาพอนิเมะ เนื้อหาในบทนี้จะกล่าวเกี่ยวกับสรุปผลการดำเนินงานวิจัย ข้อจำกัดของงานวิจัย ปัญหาและอุปสรรค การนำผลการวิจัยไปใช้ ข้อเสนอแนะ และแนวโน้มหรือทิศทางการพัฒนาในอนาคต

5.1 สรุปผลการดำเนินงานวิจัย

ผู้วิจัยใช้ภาพอนิเมะทั้งหมด 9,392 ภาพซึ่งประกอบด้วยแท็กที่ไม่ซ้ำกันทั้งหมด 16 แท็กในการรันโมเดล ML-GCN โดยผู้วิจัยแบ่งข้อมูลภาพอนิเมะ 80% เป็นข้อมูลฝึกสอน (7,514 ภาพ) และ 20% เป็นข้อมูลทดสอบ (1,878 ภาพ)

จากการทดสอบอัลกอริทึมโครงข่ายประสาทเทียมแบบคอนโวลูชันทั้ง 3 แบบ ได้แก่ ResNET-101 ResNeXT-101 และ EfficientNetB5 ผลปรากฏว่า ResNeXT มีค่า mAP หรือค่าความแม่นยำเฉลี่ยมากที่สุด

ข้อสังเกตที่ ResNeXT ให้ผลลัพธ์ดีที่สุดอาจเนื่องมาจากโครงสร้างของ ResNeXT มีการใช้ Cardinality ซึ่งไม่มีใน ResNET และ EfficientNet โดย Cardinality คือการแยกส่วนข้อมูลเป็นหลายกิ่งเพื่อประมวลผลและนำมารวมเข้าด้วยกันในภายหลัง แต่ละกิ่งสามารถเรียนรู้คุณลักษณะของภาพอนิเมะ (Feature) ที่แตกต่างกัน ส่งผลให้โมเดลสามารถเรียนรู้ข้อมูลได้หลากหลายมากขึ้น ส่วน EfficientNet ไม่เหมาะใช้ร่วมกับโมเดล ML-GCN ในการจำแนกแท็กของภาพอนิเมะเพราะได้ค่า mAP น้อยและใช้เวลามาก

ผู้วิจัยทดสอบปรับค่า LR ในระหว่างการรันโมเดลเพื่อสังเกตความเปลี่ยนแปลงของค่า mAP ซึ่งผลปรากฏว่าการเพิ่มจำนวนการปรับ LR สามารถเพิ่มค่า mAP เล็กน้อย ให้แก่ ResNET และ EfficientNet ส่วน ResNeXT ได้ผลตรงข้าม คือ ค่า mAP ลดลง

ผู้วิจัยทดสอบเพิ่มจำนวนรอบของทั้ง 3 อัลกอริทึมจาก 300 เป็น 400 รอบ ซึ่งผลปรากฏว่าค่า mAP ใหม่ของ ResNeXT และ EfficientNet เพิ่มขึ้น ส่วน ResNET ลดลง แสดงให้เห็นว่า

ResNeXT และ EfficientNet สามารถเพิ่มจำนวนรอบให้ค่า mAP เพิ่มขึ้นได้ อย่างไรก็ตามการเพิ่มจำนวนรอบจำนวนมากไม่สามารถเพิ่มค่า mAP ได้มากเสมอไปเพราะ mAP จะหยุดเพิ่มเมื่อถึงรอบที่ z ซึ่งรอบที่ z มีค่าหลากหลายตามชนิดของอัลกอริทึม พารามิเตอร์ที่ใช้และข้อมูลฝึกสอน

ผู้วิจัยตรวจสอบค่า AP ของแต่ละแท็กของแต่ละอัลกอริทึม เพื่อเปรียบเทียบความแม่นยำของแท็ก และได้ข้อสังเกตดังต่อไปนี้

- แท็กที่สร้างจาก EfficientNet ส่วนใหญ่มีค่าน้อยกว่า ResNET และ ResNeXT ยกเว้นแท็กที่เกี่ยวข้องกับสีเพราะมีค่าใกล้เคียงกับค่าของ ResNET และ ResNeXT มาก แสดงให้เห็นว่าความแม่นยำในการทำนายแท็กที่เกี่ยวกับสีของทั้ง 3 อัลกอริทึมมีค่าใกล้เคียงกัน

- การทำนายแท็กโทนสีหลักม่วงแดง (magenta_main) ของ ResNeXT แม่นยำน้อยกว่า ResNET และ EfficientNet ดังนั้นในกรณีที่ใช้โมเดล ResNeXT อาจต้องเพิ่มข้อมูลสอนของแท็กสีหลักม่วงแดงให้มากขึ้น

- 3 แท็กที่ทั้ง 3 อัลกอรึทึมทายแม่นยำมากที่สุดหรือมีค่า AP มากที่สุดได้แก่ yellow_main, black_main และ white_background

- 3 แท็กที่ทายแม่นยำน้อยที่สุดหรือมีค่า AP น้อยที่สุด ได้แก่ focus_background, serafuku และ bikini ซึ่งผู้วิจัยตรวจสอบภาพอนิเมะที่ทายผิดของแท็ก serafuku, bikini, และ focus_background ซึ่งผู้วิจัยคาดว่าหากเพิ่มแท็กที่ลักษณะคล้ายแท็ก serafuku และ bikini ในการสอนโมเดล จะสามารถเพิ่มค่าความแม่นยำของแท็ก bikini และ serafuku ให้มากขึ้น ส่วนแท็ก focus_background ควรแก้ไขในขั้นตอนการสร้างแท็ก เพื่อให้โมเดลเรียนรู้ได้ง่ายขึ้น และสุดท้ายอาจเพิ่มจำนวนข้อมูลฝึกสอนเพื่อให้โมเดลเรียนรู้รูปแบบที่หลากหลายมากขึ้นและมีความแม่นยำมากขึ้น

5.2 ข้อจำกัดของงานวิจัย

ในขั้นตอนการสร้างแท็ก ผู้วิจัยสร้างแท็กที่ระบุโทนสีโดยรวมของภาพไว้ทั้งหมด 7 แท็ก ได้แก่ red_main green_main, blue_main, cyan_main, magenta_main, yellow_main และ black_main ซึ่งแท็ก yellow_main มองว่าภาพที่มีสีขาวเป็นหลักเป็น yellow_main เพราะสีขาวและสีเหลืองเป็นสีที่ใกล้เคียงกัน

5.3 ปัญหาและอุปสรรค

ขั้นตอนเตรียมข้อมูลภาพอนิเมะเป็นขั้นตอนที่ยากและใช้เวลามากที่สุด เพราะต้องตรวจสอบภาพอนิเมะทีละภาพเพื่อให้มั่นใจว่าแต่ละภาพมีแท็กที่ถูกต้องก่อนใช้ในโมเดล ดังนั้นงานวิจัยในอนาคตควรเลือกชุดข้อมูลสอนที่มั้นใจว่ามีการแท็กภาพอย่างถูกต้อง

5.4 การนำผลการวิจัยไปใช้

โมเดลแท็กภาพอนิเมะนี้สามารถนำไปใช้บนเว็บไซต์ภาพอนิเมะที่เปิดให้บุคคลทั่วไปสามารถอัปโหลดภาพอนิเมะลงบนเว็บไซต์ โดยในขั้นตอนการอัปโหลดภาพระบบสามารถแนะนำแท็กพื้นฐานให้แก่ผู้อัปโหลดภาพแบบอัตโนมัติ ซึ่งผู้อัปโหลดภาพสามารถนำออกได้หากไม่ถูกใจ วิธีนี้สามารถช่วยอำนวยความสะดวกให้แก่ผู้อัปโหลดรูปภาพ ทำให้รูปภาพถูกกำหนดแท็กอย่างเป็นระบบมากขึ้น ช่วยให้นักวาดภาพหรือผู้ที่เข้ามาค้นหาภาพบนเว็บไซต์สามารถค้นหาภาพได้ง่ายและตรงใจมากขึ้น

5.5 ข้อเสนอแนะ

1) ควรเลือกชุดข้อมูลสอนที่มั้นใจว่ามีการแท็กภาพอย่างถูกต้องเพื่อประหยัดเวลา

2) ขั้นตอนการเตรียมข้อมูลควรทำบนเครื่องคอมพิวเตอร์ของผู้วิจัย และย้ายไปใช้ Google Colab Pro+ เฉพาะขั้นตอนการรันโมเดลที่ใช้ข้อมูลสอนจำนวนมากเพื่อประหยัดค่าใช้จ่าย

5.6 แนวโน้มหรือทิศทางการพัฒนาในอนาคต

การพัฒนาในอนาคตอาจเพิ่มจำนวนแท็กที่ใช้ฝึกสอนโมเดล ใช้ภาพอนิเมะที่มีความหลากหลายและจำนวนมากขึ้นเพื่อใช้สร้างโมเดลที่มีความแม่นยำมากขึ้น อาจเพิ่มแท็กที่ระบุโทนสีโดยรวมของภาพอีกแท็กคือ white_main

บรรณานุกรม

Z.-M. Chen, Wei, X.-S., Wang, P., & Guo, Y. (2019). Multi-label image recognition with graph convolutional networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

P. Deng, Ren, J., Lv, S., Feng, J., & Kang, H. (2020). Multi-Label Image Recognition in Anime Illustration with Graph Convolutional Networks.

K. He, Zhang, X., Ren, S., & Sun, J. . (2016). Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

R. A. Jha. (2021). *Mastering PyTorch*.

T. N. Kipf, & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Z. Lan, Maeda, K., Ogawa, T., & Haseyama, M. (2023). Hierarchical Multi-Label Attribute Classification With Graph Convolutional Networks on Anime Illustration.

M. Tan, & Le, Q. V. . (2020). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*.

S. Weidman. (2019). *Deep Learning from Scratch*(First Editions ed.).

Z. Wu, Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2019). A Comprehensive Survey on Graph Neural Networks.

S. Xie, Girshick, R., Tu, Z., He, K., & Dollar, P. (2017). Aggregated Residual Transformations for Deep Neural Networks. *International Conference on Learning Representations (ICLR)*.

F. Yi, Wu, J. Zhao, M., & Zhou, S. (2023). Anime Character Identification and Tag Prediction by Multimodality Modeling: Dataset and Model. *International Joint Conference on Neural Networks (IJCNN)*.

ภาคผนวก

ผลงานที่เผยแพร่


SPU 5th
NATIONAL AND INTERNATIONAL
SRIPATUM UNIVERSITY
CONFERENCE 2024
The 19th National and
The 9th International Sripatum University Conference
RESEARCH AND INNOVATIONS TO
SUSTAINABLE DEVELOPMENT
การประชุมวิชาการระดับชาติ ครั้งที่ 19
และการประชุมวิชาการระดับนานาชาติ ครั้งที่ 9
"วิจัยและนวัตกรรมสู่การพัฒนาที่ยั่งยืน"
29 OCT 2024
Submit Research or Academic Articles to Participate in Oral Presentation
(Via Online System)
9.00 AM. - 4.00 PM.
zoom
LIVE
SDGs SPU
4 QUALITY EDUCATION
17 PARTNERSHIPS FOR THE GOALS
Organized by Sripatum University
in cooperation with
• The University of Palermo, Italy
• Sholokhov Moscow State University for the Humanities,Russia
• Universidad de Colima, Mexico
• University of Taipei, Taiwan
• Institut Teknologi Sepuluh Nopember, Indonesia
• The Joint Graduate School of Energy and Environment
• The Social Science Research Association of Thailand
• Lawyers Council Under the Royal Patronage
• Thai Federation on Logistics
• The Institute of Internal Auditors of Thailand
• Prachachuen Research Network
• Journal Network of Social Sciences and Humanities
• Program Management Unit for Human Resources &
Institutional Development, Research and Innovation (PMU-B)
Register Now Until
13 September 2024
http://spucon.spu.ac.th
spucon@spu.ac.th
spuresearch
SCAN ME
+66-2579-1111 ext. 1331 (Miss Nadcha Sudyajai)
+66-2579-1111 ext. 1239 (Mr. Aphimuk Sadompruek)
+66-2579-1111 ext. 1336 (Dr. Siriporn Thongkaew)
+66-2579-1111 ext. 1155 (Assoc. Prof. Dr. Subin Yurarach)

SPU
SRIPATUM UNIVERSITY

BANGKHEN
2410/2
PHAHOLYOTHIN RD.,
JATUJAK, BANGKOK
10900
TEL. 0 2579 1111
FAX. 0 2561 1721
www.spu.ac.th

CHONBURI CAMPUS
79 BANGNA-TRAD RD.,
KLONGTAMRU, MUANG,
CHONBURI 20000
TEL. 0 3874 3690-9
FAX. 0 3874 3700
www.east.spu.ac.th

KHON KAEN
182/12 MOO 4,
SRICHAN RD.,
NAIMUANG DISTRICT,
AMPHUR MUANG,
KHON KAEN 40000
TEL. 0 4322 4111
FAX. 0 4322 4119
www.khonkaen.spu.ac.th

ที่ มศป.0203/4594

22 ตุลาคม 2567

เรื่อง ตอบรับการนำเสนอบทความในการประชุมวิชาการ

เรียน คุณอดิเทพ พรหมพา, คุณสุนิสา ริมเจริญ

ตามที่ท่านได้ส่งบทความเรื่อง "การใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันสำหรับสร้างโมเดลในการแท็กภาพอนิเมะ" เพื่อนำเสนอในงานประชุมวิชาการระดับชาติ ครั้งที่ 19 และงานประชุมวิชาการระดับนานาชาติ ครั้งที่ 9 มหาวิทยาลัยศรีปทุม ประจำปี 2567 เรื่อง การวิจัยและนวัตกรรมสู่การพัฒนาที่ยั่งยืน (The 19th National and The 9th International Sripatum University Conference on Research and Innovations to Sustainable Development: SPUCON2024) แบบออนไลน์ (Virtual conference) ในวันอังคารที่ 29 ตุลาคม 2567 เวลา 09.00 - 16.00 น. นั้น

ในการนี้ ผู้ทรงคุณวุฒิ (Peer reviewers) ได้ประเมินบทความเรื่องดังกล่าวเป็นที่เรียบร้อยแล้ว และคณะกรรมการพิจารณาผลงาน มีมติเห็นชอบให้นำเสนอบทความในการประชุมวิชาการฯ ตามวัน เวลา ดังกล่าวข้างต้น และจะตีพิมพ์ในหนังสือประมวลบทความการประชุมวิชาการฯ ในรูปแบบอิเล็กทรอนิกส์ (e-Proceedings) ต่อไป

จึงเรียนมาเพื่อโปรดทราบ

(รองศาสตราจารย์ ดร.ปิยากร หวังมหาพร)
ประธานคณะกรรมการพิจารณาผลงานการประชุมวิชาการระดับชาติ ครั้งที่ 19
และการประชุมวิชาการระดับนานาชาติ ครั้งที่ 9
มหาวิทยาลัยศรีปทุม ประจำปี 2567

ฝ่ายเลขานุการคณะกรรมการพิจารณาผลงาน SPU Conference 2024
ศูนย์ส่งเสริมการวิจัยและการประกันคุณภาพการศึกษา มหาวิทยาลัยศรีปทุม
โทรศัพท์ 02 579 1111 ต่อ 1331 1336 1155
ไปรษณีย์อิเล็กทรอนิกส์ spucon@spu.ac.th

การใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันสำหรับสร้างโมเดลในการแท็กภาพอนิเมะ

Using Convolutional Neural Networks for Creating Models to Tag Anime Images

นายอดิเทพ พรหมพา

คณะวิทยาการสนเทศ มหาวิทยาลัยบูรพา

E-mail: adithep344@gmail.com

รศ. ดร. สุนิสา ริมเจริญ

คณะวิทยาการสนเทศ มหาวิทยาลัยบูรพา

E-mail: rsunisa@buu.ac.th

บทคัดย่อ

ในปัจจุบันมีเว็บไซต์ซึ่งเป็นแหล่งรวมภาพผลงานอนิเมะให้นักวาดภาพหาแรงบันดาลใจในการสร้างผลงานของตน แต่แท็กของภาพอนิมะไม่มีรูปแบบที่ชัดเจน บางภาพอนิเมะถูกกำหนดแท็กไม่เพียงพอ ทำให้นักวาดภาพที่เข้ามาหาแรงบันดาลใจมีโอกาสค้นหาเจอภาพที่ต้องการน้อยลง พวกเขาอาจสูญเสียโอกาสในการสร้างผลงานให้ออกมาดีที่สุด ผู้วิจัยจึงนำเสนอการนำโมเดล ML-GCN ซึ่งเป็นโมเดลสำหรับการทำ Muti-label เพื่อช่วยในการกำหนดแท็กของภาพอนิเมะให้มีความถูกต้องและครบถ้วนมากขึ้น โครงสร้างหลักของโมเดลนี้ประกอบด้วยโครงข่ายคอนโวลูชันแบบกราฟ (Graph Convolutional Networks: GCN) และโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network: CNN) ในงานวิจัยนี้ผู้วิจัยเปรียบเทียบอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชัน 3 แบบ ได้แก่ ResNET, ResNeXT และ EfficientNet เพื่อวิเคราะห์ว่าอัลกอริทึมใดจำแนกแท็กของภาพอนิเมะได้ถูกต้องมากกว่า จากผลการทดลองพบว่า ResNeXT มีค่าความแม่นยำเฉลี่ย (mAP) สูงกว่า ResNET และ EfficientNet ทำให้ ResNeXT เหมาะนำมาใช้ร่วมกับโมเดล ML-GCN ในการจำแนกแท็กของภาพอนิเมะ

คำสำคัญ: อนิเมะ, แท็ก, CNN, GCN, ResNET, ResNeXT, EfficientNet, Muti-label

ABSTRACT

Recently, there are websites that provide anime artworks for artists who search for inspiration to create their own works. Unfortunately, tags of anime images lack clear structures, and some images are insufficiently tagged. This reduces the chances of the artists to find the specific images they need and they might lose their chances of producing their best work. To solve this problem, we present using the ML-GCN model, a multi-label classification, to improve the correctness and completeness of anime image tagging. The core structure of this model consists of Graph Convolutional Networks (GCN) and Convolutional Neural Networks (CNN). In this study, we compared three convolutional neural network algorithms (ResNET, ResNeXT, and EfficientNet) to determine which algorithm more accurately classifies anime image tags. The experimental results show that

ResNeXT yields a higher mean average precision (mAP) than ResNET and EfficientNet, indicating that ResNeXT is better suited for apply with the ML-GCN model in classifying anime image tags.

KEYWORDS: Anime, Tag, CNN, GCN, ResNET, ResNeXT, EfficientNet, Muti-label

1. ความสำคัญและที่มาของปัญหาวิจัย

ภาพอนิเมะเป็นรูปแบบการวาดการ์ตูนที่ถูกแพร่หลายมาจากประเทศญี่ปุ่น บนอินเตอร์เน็ตมีเว็บไซต์สาธารณะที่เปิดให้ผู้ใช้ทั่วไปสามารถนำภาพอนิเมะที่ตนวาดมาเผยแพร่เพื่อเป็นแรงบันดาลใจให้แก่ผู้วาดภาพอนิเมะ เช่น เว็บไซต์ Pixiv.net เป็นต้น ในขั้นตอนการอัปโหลดรูปภาพลงบนเว็บไซต์ ผู้อัปโหลดภาพอนิเมะต้องกำหนดแท็ก (Tag) หรือลักษณะของตัวละครบนรูปภาพให้แก่ภาพของตนเพื่อใช้เป็นคีย์เวิร์ดให้ผู้อื่นสามารถค้นหาภาพง่ายขึ้น โดยตารางที่ 1 แสดงตัวอย่างของภาพอนิเมะและแท็กของแต่ละภาพซึ่งถูกเผยแพร่บนเว็บไซต์ Pixiv.net

ตารางที่ 1 ตัวอย่างแท็กของภาพอนิเมะจากเว็บไซต์ Pixiv.net

ลำดับที่	ภาพ	แท็ก
1.	 https://www.pixiv.net/en/artworks/94987223	1. Original 2. Fox Ears 3. Christmas 4. Santa 5. Black Tights
2.	 https://www.pixiv.net/en/artworks/110725274	1. Skeb (ชื่อเว็บไซต์รับจ้างวาดรูป) 2. Girl 3. Tail
3.	 https://www.pixiv.net/en/artworks/102630131	1. Original 2. Cat Ears Maid 3. Original 1000+ Bookmarks

บางครั้งการอัปโหลดภาพอนิเมะไปยังเว็บไซต์ที่เผยแพร่ภาพอนิเมะ ไม่มีข้อกำหนดว่าแต่ละภาพอนิเมะต้องกำหนดแท็กด้านใดบ้าง จึงทำให้เกิดปัญหาเกี่ยวกับการค้นหารูปภาพ สามารถสังเกตได้จากตารางที่ 1 ดังนี้

- หากผู้ที่ต้องการค้นหาภาพอนิเมะใช้คีย์เวิร์ดในการค้นหาภาพว่า “Fox Ears (หูสุนัขจิ้งจอก)” เขาจะเจอเพียงภาพในลำดับที่ 1 เพียงภาพเดียว แม้ว่าภาพในลำดับที่ 1 และ 2 มีสุนัขจิ้งจอกเหมือนกัน

- ภาพในลำดับที่ 2 ผู้อัปโหลดภาพสร้างแท็ก “Cat Ears Maid" ขึ้นเอง ซึ่งแท็กนี้มีการระบุคุณลักษณะหลายสิ่งในแท็กเดียว แต่ควรแยกเป็น 2 แท็ก คือ "Cat Ears" และ "Maid" เพื่อให้สามารถค้นหาภาพอนิเมะได้ง่ายขึ้น

การที่ผู้อัปโหลดภาพอนิเมะสามารถกำหนดแท็กของภาพได้อย่างอิสระ ทำให้บางภาพถูกกำหนดแท็กไม่เพียงพอ รวมทั้งเกิดแท็กที่มีความหมายซ้ำกันแต่ชื่อต่างกัน อาจส่งผลให้นักวาดภาพที่เข้ามาในเว็บไซต์เพื่อหาแรงบันดาลใจในการวาดภาพอนิเมะ มีโอกาสค้นหาเจอภาพที่ต้องการน้อยลงหรืออาจค้นหาไม่เจอ พวกเขาอาจสูญเสียโอกาสในการสร้างผลงานให้ออกมาดีที่สุด โดยเฉพาะผู้ที่ลงแข่งขันประกวดวาดภาพหรือผู้ที่ทำงานเกี่ยวกับการสร้างสรรค์ผลงานศิลปะเป็นอาชีพ พวกเขาอาจเสียโอกาสการสร้างชื่อเสียงหรือโอกาสการได้รับงาน สำหรับนักวาดภาพการหาแรงบันดาลใจและคิดไอเดียเป็นขั้นตอนที่ยาก ดังนั้นหากสามารถค้นหาภาพผลงานที่ตรงใจได้ตั้งแต่ต้นและเป็นจำนวนมากจะช่วยให้สร้างสรรค์ผลงานเร็วขึ้นและสมบูรณ์แบบมากขึ้น

ปัญหาการกำหนดแท็กมีสาเหตุจากผู้อัปโหลดกำหนดแท็กให้แก่รูปภาพด้วยตนเองโดยไม่มีรูปแบบที่ชัดเจน ทำให้เกิดปัญหาการกำหนดแท็กไม่เพียงพอและเกิดแท็กใหม่ที่มีความหมายคล้ายกับแท็กที่มีอยู่ แต่การบังคับให้ผู้อัปโหลดภาพกำหนดแท็กด้านต่าง ๆ ให้ครบถ้วนจะทำให้เกิดความไม่สะดวกสบายในการอัปโหลดภาพผลงาน ผู้วิจัยจึงเสนอว่า ในขั้นตอนที่ผู้อัปโหลดภาพ ระบบควรแนะนำแท็กให้แก่ภาพผลงานเพื่อไม่เพิ่มภาระให้ผู้อัปโหลดภาพและเพิ่มโอกาสในการหาภาพเหล่านี้พบมากขึ้น

ผู้วิจัยศึกษาโมเดล ML-GCN ซึ่งเผยแพร่โดย Zhao-Min Chen และคณะ (2019) โมเดลนี้ใช้เทคนิคการเรียนรู้เชิงลึกเพื่อสอนให้คอมพิวเตอร์จดจำและจำแนกลักษณะต่าง ๆ ของภาพถ่าย โมเดลนี้สามารถประยุกต์ใช้เพื่อจำแนกแท็กของภาพอนิเมะและใช้โมเดลนี้แนะนำแท็กของภาพอนิเมะให้แก่ผู้อัปโหลดภาพอนิเมะในขั้นตอนการกำหนดแท็กได้

โมเดล ML-GCN ประกอบด้วยเทคนิคการเรียนรู้เชิงลึก 2 เทคนิค ได้แก่ โครงข่ายคอนโวลูชันแบบกราฟและโครงข่ายประสาทเทียมแบบคอนโวลูชัน โดยผู้วิจัยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน 3 แบบ ได้แก่ ResNET (Kaiming He และคณะ, 2016) ซึ่งเป็นอัลกอริทึมที่ใช้ในโมเดล ML-GCN และใช้ ResNeXT (Saining Xie และคณะ, 2017) และ EfficientNet (Mingxing Tan และ Quoc V. Le, 2020) ซึ่งเป็นอัลกอริทึมใหม่ของโครงข่ายประสาทเทียมแบบคอนโวลูชันเพื่อวิเคราะห์ว่าอัลกอริทึมใดให้ผลการทำนายแท็กของภาพอนิเมะได้ถูกต้องมากกว่า

2. วัตถุประสงค์ของการวิจัย

เพื่อสร้างโมเดลการเรียนรู้เชิงลึกในการแท็กภาพอนิเมะ

3. เอกสารและงานวิจัยที่เกี่ยวข้อง

3.1 โครงข่ายประสาทเทียมแบบคอนโวลูชัน

โครงข่ายประสาทเทียมแบบคอนโวลูชันเป็นอัลกอริทึมที่ใช้เทคนิคการเรียนรู้เชิงลึกจำลองการทำงานของโครงข่ายประสาทในสมอง เพื่อให้คอมพิวเตอร์สามารถเรียนรู้และวิเคราะห์ได้แบบมนุษย์ ซึ่งโครงข่ายประสาทเทียมแบบคอนโวลูชันเป็นโครงข่ายประสาทเทียมที่ออกแบบสำหรับการเรียนรู้ข้อมูลรูปภาพโดยเฉพาะ

งานวิจัยนี้ผู้วิจัยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน 3 แบบ ได้แก่ ResNET ซึ่งเป็นอัลกอริทึมที่ใช้ในโมเดล ML-GCN และใช้ ResNeXT กับ EfficientNet ซึ่งเป็นอัลกอริทึมใหม่ของโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Ashish Ranjan Jha, 2021)

1) ResNET (Residual Network) เป็นอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ถูกคิดค้นเพื่อแก้ปัญหาอัตราการเกิดข้อผิดพลาด (Error Rate) ที่เพิ่มขึ้นจากการใช้เลเยอร์จำนวนมาก ซึ่งโดยทั่วไปการเพิ่มเลเยอร์จะช่วยลดอัตราการเกิดข้อผิดพลาดได้ แต่ถ้าจำนวนเลยอร์ใช้มีจำนวนมากจะกลายเป็นการเพิ่มอัตราการเกิดข้อผิดพลาด ปัญหานี้เรียกว่า Vanishing/Exploding Gradient อัลกอริทึมนี้ใช้เทคนิคการข้ามเลเยอร์ (Skip Connection/Shortcut Connection) เพื่อแก้ปัญหาดังกล่าวโดยการข้ามเลเยอร์ที่ส่งผลให้ประสิทธิภาพของอัลกอริทึมแย่ลง (Kaiming He และคณะ, 2016)

2) ResNeXT เป็นอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ถูกคิดค้นเพื่อเพิ่มทางเลือกในการเพิ่มประสิทธิภาพของโมเดลโดยพยายามไม่ใช้การเพิ่มความลึกหรือความกว้างของโมเดล อัลกอริทึมนี้ใช้การเพิ่มกิ่ง (Branch) แทนการเพิ่มความลึกหรือความกว้างของโมเดล เรียกว่า Cardinality ใช้หลักการแยกส่วนข้อมูลนำเข้าที่เป็นรูปภาพเพื่อประมวลผล จากนั้นนำมารวมกันในตอนท้าย จุดประสงค์หลักของอัลกอริทึมคือใช้เพิ่มประสิทธิภาพให้กับโมเดลที่ใช้ข้อมูลนำเข้าจำนวนมากโดยพยายามหลีกเลี่ยงการเพิ่มความลึกหรือความกว้างของโมเดล (Saining Xie และคณะ, 2017)

3) EfficientNet เป็นอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ถูกคิดค้นเพื่อแก้ปัญหาการกำหนดค่าอัตราส่วน (Scaling) ของโมเดลโครงข่ายประสาทเทียมแบบคอนโวลูชันรุ่นก่อนหน้า เพราะการปรับอัตราส่วนของโมเดลสามารถเพิ่มประสิทธิภาพให้แก่โมเดลได้ แต่ยังไม่มีรูปแบบการปรับค่าอัตราส่วนโมเดลที่ชัดเจน ทำให้ต้องใช้การลองผิดลองถูกแก้ไขซ้ำหลายครั้งจนกระทั่งได้ค่าที่เหมาะสม วิธีการปรับอัตราส่วนของโมเดลมี 3 แบบ ได้แก่ (1) การปรับอัตราส่วนความกว้างของโมเดลโดยการเพิ่มจำนวนนิวตรอนในชั้นเลยอร์ (2) การปรับอัตราส่วนความลึกของโมเดลโดยการเพิ่มจำนวนชั้นเลเยอร์ และ (3) การปรับอัตราส่วนความละเอียด (ความกว้างและความสูง) ของข้อมูลนำเข้าที่เป็นรูปภาพ อัลกอริทึมนี้จึงถูกคิดค้นเพื่อเสนอวิธีการปรับค่าอัตราส่วนทั้ง 3 ด้าน (ความกว้าง ความลึก และความละเอียด) ให้เกิดความสมดุลเรียกว่า Compound Scaling โดยใช้อัตราส่วนคงที่ (Fixed Ratio) เพื่อในปรับค่าอัตราส่วนทั้ง 3 ด้าน (Mingxing Tan และ Quoc V. Le, 2020)

3.2 โครงข่ายคอนโวลูชันแบบกราฟ

โครงข่ายคอนโวลูชันแบบกราฟ เป็นโครงข่ายประสาทเทียมชนิดหนึ่งที่สามารถใช้เรียนรู้ความสัมพันธ์ระหว่างข้อมูลนำเข้าบนโครงสร้างที่มีลักษณะเป็นกราฟ แตกต่างจากโครงข่ายประสาทเทียมแบบคอนโวลูชันทั่วไปเพราะข้อมูลนำเข้าไม่ใช้เป็นรูปภาพ แต่ใช้เป็นกราฟที่สร้างขึ้นจากการโยงเส้นความสัมพันธ์ระหว่างแท็กของภาพอนิเมะ (Thomas N. Kipf และ Max Welling, 2017) ตัวอย่างความสัมพันธ์ของแท็ก เช่น หากภาพอนิเมะใดมีแท็ก “นักเรียน” ก็มีโอกาสมีแท็ก “เครื่องแบบ” และแท็ก “เด็กผู้หญิง” ด้วยเช่นกัน เป็นต้น

3.3 Adj Matrix (Adjacency Matrix)

Adjacency Matrix คือเมทริกซ์ที่ใช้แทนค่ารูปแบบกราฟความสัมพันธ์ได้ ผู้วิจัยแปลงแท็กของภาพอนิเมะเป็นเมทริกซ์นี้เพื่อใช้เป็นข้อมูลนำเข้าของโครงข่ายคอนโวลูชันแบบกราฟ เมทริกซ์นี้ประกอบด้วยเลข 1 และ 0 โดยเลข 1 หมายถึงแท็กมีความสัมพันธ์กัน และเลข 0 หมายถึงแท็กไม่ความสัมพันธ์กัน

3.4 GloVe (Global Vectors for Word Representation)

GloVe คืออัลกอริทึมที่สามารถใช้หาค่าความสัมพันธ์ของคำ (Words) โดยการแปลงค่าเหล่านั้นเป็นค่าเวกเตอร์ (Vector) หากค่าเวกเตอร์ของคำใกล้กันมากแสดงว่าคำเหล่านั้นมีความสัมพันธ์เกี่ยวข้องกันมาก

อัลกอริทึมนี้สามารถใช้หาความสัมพันธ์ระหว่างแท็กของภาพอนิเมะได้ เช่น แท็กเหล่านี้มีโอกาสเกิดขึ้นพร้อมกัน เป็นต้น

3.5 โมเดล ML-GCN

งานวิจัยของ Pengfei Deng และคณะ (2020) และงานวิจัยของ Ziwen Lan และคณะ (2023) ได้นำเสนอแนวคิดการพัฒนาโมเดลสำหรับแท็กภาพอนิเมะโดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันและโครงข่ายคอนโวลูชันแบบกราฟ

ผู้วิจัยประยุกต์ใช้โมเดล ML-GCN เพื่อใช้แท็กภาพอนิเมะ เพราะโมเดลนี้ใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันและโครงข่ายคอนโวลูชันแบบกราฟเป็นส่วนประกอบหลักในโมเดลและเป็นโมเดลที่เปิดให้สามารถดาวน์โหลดได้ฟรีบนเว็บไซต์ Github

โมเดล ML-GCN ถูกนำเสนอโดย Zhao-Min Chen และคณะ (2019) เพื่อใช้สอนให้คอมพิวเตอร์เรียนรู้การจำแนกข้อมูลรูปภาพที่ประกอบด้วยมากกว่า 1 แท็ก (Multi-label Classification)

3.6 การหาค่าความสำคัญของแท็ก

งานวิจัยนี้ใช้สมการ TF-IDF (Term Frequency - Inverse Document Frequency) สำหรับการหาค่าความสำคัญของคำบนเอกสาร ซึ่งงานวิจัยของ Fan Yi และคณะ (2023) ประยุกต์ใช้สมการนี้ในการหาค่าความสำคัญของแท็กบนภาพอนิเมะ เนื่องจากแท็กของภาพอนิเมะมีจำนวนมาก ผู้วิจัยจึงใช้สมการนี้เพื่อคัดเลือกเฉพาะแท็กที่มีความสำคัญให้คอมพิวเตอร์เรียนรู้ โดยคำนวณได้ดังสมการที่ (1)

$$\sum^{c_i} \mathrm{TF} \times \mathrm{IDF}_i = \left(\sum^{c_i} \frac{1}{k_i} \right) \times log \frac{C}{c_i} \quad (1)$$

- i คือ แท็กที่สนใจ
- c_i คือ จำนวนของภาพอนิเมะที่มีแท็ก i
- k_i คือ จำนวนของแท็กทั้งหมดของภาพอนิเมะที่มีแท็ก i
- C คือ จำนวนของภาพอนิเมะทั้งหมด
- TF (Term Frequency) คือ ค่าความถี่ของแท็ก i ที่พบในภาพอนิเมะ ซึ่งคำนวณได้จากการนำจำนวนแท็ก i ที่พบในภาพอนิมะ (มีค่าเป็น 1 เสมอ) หารด้วย k_i ซึ่งเป็นจำนวนของแท็กทั้งหมดของภาพอนิเมะที่มีแท็ก i
- $\sum$TF คือ ผลรวมของค่า TF ที่คำนวณจากภาพอนิเมะที่พบแท็ก i ทีละภาพรวมกัน ยิ่งมีค่ามากแปลว่าแท็กนั้นมีความสำคัญ
- DF_i (Document Frequency) คือ ค่าความถี่ของภาพอนิเมะที่มีแท็ก i ซึ่งยิ่งค่าน้อยแปลว่าแท็กนั้นมีความสำคัญ ซึ่งคำนวณได้จากการนำ C ซึ่งเป็นจำนวนของภาพอนิเมะทั้งหมดหารด้วย c_i ซึ่งเป็นจำนวนของภาพอนิเมะที่มีแท็ก i
- IDF_i (Inverse Document Frequency) คือ ค่าผกผันของ DF_i ซึ่งเหตุผลที่ใช้ค่าผกผันเพื่อเปลี่ยนจาก “ยิ่งมีค่าน้อยแปลว่าแท็กนั้นมีความสำคัญ” เป็น “ยิ่งมีค่ามากแปลว่าแท็กนั้นมีความสำคัญ”

3.7 สมการในการวิเคราะห์ประสิทธิภาพ

3.7.1 สมการ P@*k* (Precision@*k*) เป็นสมการที่ผู้วิจัยใช้ประเมินค่าความแม่นยำของแท็ก (หรือ Class) ที่โมเดลทายให้กับภาพอนิมะ 1 ภาพว่าถูกต้องเพียงใด โดยสมการนี้จะแตกต่างจากสมการ Precision แบบธรรมดา เพราะสมการนี้ต้องเรียงข้อมูลและคำนวณค่าความแม่นยำจากตำแหน่งของตน (*k*) และตำแหน่งก่อนหน้าทั้งหมด โดยคำนวณได้ดังสมการที่ (2)

$$\mathrm{P}(k) = \frac{TP@k}{TP@k + FP@k} \tag{2}$$

- *@k* คือ ลำดับของภาพอนิเมะซึ่งถูกเรียงตามค่าความน่าจะเป็น (Probability) ที่โมเดลทายเป็นบวก
- *TP@k* (True Positive@*k*) คือ จำนวนของภาพอนิเมะที่โมเดลทายว่าเป็นบวกและทายถูกโดยนับตั้งแต่ภาพลำดับที่ *k* และภาพลำดับก่อนหน้า *k* ทั้งหมด
- *FP@k* (False Positive@*k*) คือ จำนวนของภาพอนิเมะที่โมเดลทายว่าเป็นบวก แต่ทายผิด โดยนับตั้งแต่ภาพลำดับที่ *k* และภาพลำดับก่อนหน้า *k* ทั้งหมด

3.7.2 สมการ AP (Average Precision) เป็นสมการที่ผู้วิจัยใช้ประเมินค่าความแม่นยำเฉลี่ยของ 1 แท็ก (หรือ Class) ที่โมเดลทายให้แก่ภาพอนิเมะที่ใช้เป็นข้อมูลทดสอบทั้งหมด (Testing Data) ดังนั้นทุกแท็กจะมีค่า AP เป็นของตัวเอง โดยคำนวณได้ดังสมการที่ (3)

$$\mathrm{AP}_i = \frac{\sum_{k=1}^{n}(\mathrm{P}(k) * rel(k))}{n} \tag{3}$$

- *i* คือ ลำดับของแท็ก (ทุกแท็กต้องคำนวณค่า *AP*)
- *n* คือ จำนวนภาพอนิเมะที่ใช้เป็นข้อมูลทดสอบ
- P(*k*) คือ คำนวณค่า P@*k* ของภาพอนิเมะทีละภาพตามลำดับที่ *k*
- *rel*(*k*) คือ ตัวแปรเงื่อนไข ซึ่งเป็นได้ทั้ง 1 หรือ 0 ในกรณีที่ภาพอนิเมะนี้ในเฉลยมีแท็ก *rel*(*k*) จะมีค่าเป็น 1 แต่กรณีที่ในเฉลยไม่มีแท็ก จะมีค่าเป็น 0 ซึ่งหากค่าเป็น 0 จะหมายความว่าไม่ต้องบวกสะสมค่า P(*k*) ของภาพอนิเมะภาพปัจจุบัน เพราะ P(*k*) * 0 จะได้ผลลัพธ์เป็น 0

3.7.3 สมการ mAP (Mean Average Precision) เป็นสมการที่ผู้วิจัยใช้ประเมินค่าความแม่นยำเฉลี่ยของโมเดลโดยพิจารณาจากค่าความแม่นยำของทุกแท็กที่ทำนายภาพอนิเมะที่เป็นข้อมูลทดสอบ (นำค่า AP ของทุกแท็กรวมกันและหาค่าเฉลี่ย) โดยคำนวณได้ดังสมการที่ (4)

$$\mathrm{mAP} = \frac{\sum_{i=1}^{j}\mathrm{AP}_i}{j} \tag{4}$$

- j คือ จำนวนแท็กทั้งหมด
- AP_i คือ ค่า AP ของแท็กลำดับที่ i

4. วิธีดำเนินการวิจัย

4.1 ค้นหาแหล่งข้อมูลภาพอนิเมะ

ผู้วิจัยใช้ข้อมูลรูปภาพอนิเมะ danbooru2021 ซึ่งเปิดสามารถดาวน์โหลดข้อมูลได้ฟรีบนเว็บไซต์ Kaggle ในรูปแบบไฟล์ CSV มีข้อมูล URL ของภาพอนิเมะทั้งหมด 3,020,460 ภาพและแท็กที่ไม่ซ้ำกัน 427,578 แท็ก ข้อมูล danbooru2021 เป็นข้อมูลรูปภาพอนิเมะจากเว็บไซต์ danbooru ซึ่งเป็นเว็บไซต์ที่เปิดให้ผู้ใช้ทั่วไปนำภาพอนิเมะอัปโหลดลงเว็บไซต์ได้

4.2 การเตรียมข้อมูล (Data Preparation)

ข้อมูล danbooru2021 เป็นข้อมูลภาพอนิเมะจากเว็บไซต์ซึ่งผู้อัปโหลดภาพอนิเมะสามารถกำหนดแท็กให้แก่รูปภาพเองได้ ดังนั้นแท็กเหล่านี้จึงมีบางแท็กที่สะกดผิด ไม่สื่อความหมาย ความหมายคล้ายคลึงกับแท็กอื่น กำหนดแท็กไม่ครบถ้วน ฯลฯ ผู้วิจัยจึงต้องตรวจสอบความถูกต้องของแท็กของภาพอนิเมะเหล่านี้เพื่อให้มั่นใจว่าแท็กมีความถูกต้องจริงก่อนนำเข้าโมเดล

ผู้วิจัยใช้โปรแกรม Jupyter Notebook และภาษา Python ช่วยตรวจสอบความถูกต้องและกลั่นกรองข้อมูลภาพอนิเมะและข้อมูลแท็กในเบื้องต้นเพื่อแบ่งเบาภาระของผู้วิจัย และในขั้นตอนสุดท้ายผู้วิจัยตรวจสอบภาพทีละภาพด้วยตนเองเพื่อให้มั่นใจในความถูกต้องของแท็ก โดยมีขั้นตอนดังต่อไปนี้

4.2.1 การทำความสะอาดข้อมูลในเบื้องต้น

- ลบข้อมูลภาพอนิเมะที่ไม่มีการกำหนดแท็กหรือไม่มี URL
- แก้ไขข้อมูล URL ภาพอนิเมะ เพราะบาง URL ไม่ขึ้นต้นด้วย “https://” แต่ขึ้นต้นด้วย “//”
- ลบข้อมูลภาพอนิเมะที่ URL ไม่ลงท้ายด้วย .png .jpg หรือ .jpeg

4.2.2 การกลั่นกรองแท็ก

- ลบข้อมูลภาพอนิเมะที่มีแท็กบ่งบอกว่ามีตัวละครหญิงหรือชายมากกว่า 1 คนออก เพราะผู้วิจัยสร้างโมเดลเพื่อแท็กภาพอนิเมะที่มีตัวละครเพียง 1 คนเท่านั้น
- ลบข้อมูลภาพอนิเมะที่มีแท็กบ่งบอกว่าไม่ใช่ภาพอนิเมะปกติ เช่น มังงะ ภาพสเก็ต ภาพแสดงสีหน้าตัวละครหลายหน้า ภาพออกแบบ รูปถ่าย และภาพซ้อนแบบซูม (Zoom-layer)
- นำแท็กที่เป็นสัญลักษณ์ออก เช่น >_<, =_=, ^3^ เป็นต้น
- นำแท็กที่บ่งบอกลักษณะสีออก เช่น หมวกสีแดง (red-hat) และหมวกสีเหลือง (yellow-hat) ให้เหลือเพียงคำว่า หมวก (hat)
- ลบข้อมูลภาพอนิเมะที่มีจำนวนแท็กตั้งแต่ 20 แท็กขึ้นไปออก เพราะภาพที่มีจำนวนแท็กมากมักเป็นภาพที่มีหลายตัวละคร

4.2.3 การดาวน์โหลดภาพอนิเมะ

หลังจากขั้นตอนการกลั่นกรองแท็ก ภาพอนิเมะลดจาก 3,020,460 ภาพเหลือเพียง 1,060,144 ภาพ ผู้วิจัยสุ่มเลือกภาพอนิเมะ 100,000 ภาพเพื่อใช้ในงานวิจัยและเริ่มดาวน์โหลดภาพ

หลังจากการดาวน์โหลดภาพอนิเมะ ผู้วิจัยต้องการแปลงขนาดของภาพอนิเมะทุกภาพเป็นชนิด jpg และปรับขนาดเป็น 500x500 พิกเซล ผู้วิจัยจึงตรวจสอบขนาดของภาพและลบภาพที่อัตราส่วน สูง/กว้าง ไม่อยู่ระหว่าง 1.8 (1270/720) และ 0.5 (720/1270) เพราะภาพเหล่านี้จะเสียหายหลังแปลงขนาดเป็น 500x500 พิกเซล

4.2.4 การคัดเลือกแท็กที่สำคัญ

เนื่องจากจำนวนแท็กที่ไม่ซ้ำกันของภาพอนิเมะมีจำนวนมาก ผู้วิจัยจึงกลั่นกรองเลือกเฉพาะแท็กที่สำคัญ ในขั้นตอนแรกผู้วิจัยคัดเลือกเฉพาะแท็กที่พบในภาพอนิมะตั้งแต่ 1,000 ภาพเป็นต้นไป และใช้สมการ TF-IDF เพื่อหาค่าความสำคัญของแท็กของที่คัดเลือกไว้ ผู้วิจัยคัดเลือกแท็กที่สำคัญออกมา 16 แท็ก ผู้วิจัยลบแท็กอื่นนอกเหลือจากแท็กที่คัดเลือกไว้ออกจากภาพอนิเมะทั้งหมดและลบภาพอนิเมะที่ไม่เหลือแท็กออก จากนั้นผู้วิจัยนำภาพอนิเมะที่เหลือผ่านกระบวนการ Domnsampling เพื่อลดความต่างของจำนวนแท็กทั้ง 16 แท็ก และได้ผลลัพธ์เป็นภาพอนิเมะทั้งหมด 9,904 ภาพ

4.2.5 การตรวจสอบภาพด้วยตนเอง

หลังจากการใช้ภาษา Python ช่วยตรวจสอบและกลั่นกรองข้อมูลภาพอนิเมะจนเหลือภาพอนิเมะทั้งหมด 9,904 ภาพ และ 16 แท็ก ผู้วิจัยไล่ตรวจสอบภาพเหล่านี้ทีละภาพด้วยตนเองเพื่อให้มั่นใจในความถูกต้องและครบถ้วนของแท็กที่ภาพอนิเมะเหล่านี้มี และลบภาพอนิเมะที่มีหลายตัวละครหรือภาพอนิเมะที่ไม่ปกติออก ท้ายที่สุดเหลือภาพอนิเมะทั้งหมด 9,392 ภาพ

4.3 การเตรียมโมเดล

ผู้วิจัยปรับปรุงโมเดล ML-GCN ให้สามารถใช้งานบน Google Colab Pro+ และสามารถใช้งานร่วมกับอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันทั้ง 3 แบบ ได้แก่ ResNET ResNeXT และ EfficientNet โดย Runtime Type ของ Google Colab Pro+ ผู้วิจัยใช้เป็น A100 GPU

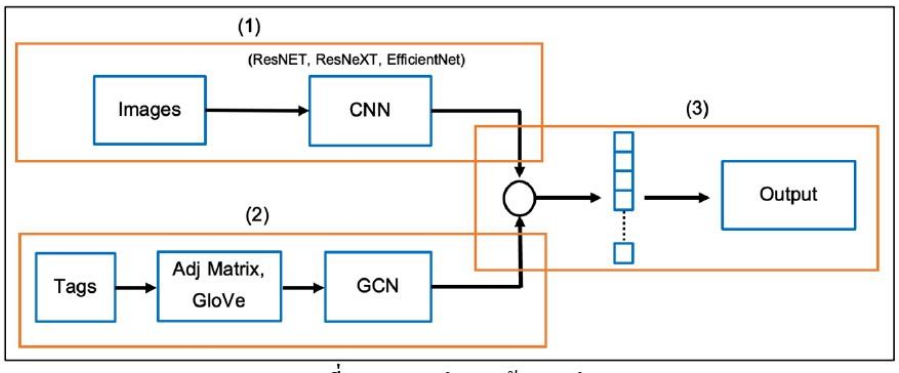
โมเดลที่ผู้วิจัยใช้ประกอบด้วยไลบรารีหลัก ดังต่อไปนี้ python 3.10.12, numpy 1.26.4, torch 2.4.1, torchnet 0.0.4, torchvision 0.19.1, PIL 10.4.0 และ tqdm 4.66.5

แผนภาพที่ 1 แสดงโครงสร้างหลักของโมเดล ML-GCN ที่ผู้วิจัยใช้ ซึ่งมีโครงสร้างแบ่งออกเป็น 3 ส่วนหลัก ได้แก่

1) ส่วนการเรียนรู้ภาพ ซึ่งเป็นส่วนที่ใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันในการเรียนรู้คุณลักษณะของภาพ โดยอัลกอริทึมของโครงข่ายประสาทเทียมแบบคอนโวลูชันที่ที่ผู้วิจัยใช้ ได้แก่ ResNET-101 ResNeXT-101 และ EfficientNetB5 โดย ResNET-101 เป็นอัลกอริทึมที่ถูกใช้ในโมเดล ML-GCN ผู้วิจัยเลือกใช้ ResNeXT-101 และ EfficientNetB5 เพราะขนาดของข้อมูลเอาท์พุท (ก่อนผ่าน Fully Connected Layer) มีขนาด 2048 แบบเดียวกับ ResNET-101

2) ส่วนการเรียนรู้แท็กของภาพ ซึ่งเป็นส่วนที่ใช้โครงข่ายคอนโวลูชันแบบกราฟในการเรียนรู้ความสัมพันธ์และความเกี่ยวข้องกันของแท็กต่าง ๆ ซึ่งแท็กของภาพต้องแปลงเป็นค่าเวกเตอร์ความสัมพันธ์โดยใช้อัลกอริทึม GloVe และแปลงเป็นกราฟรูปแบบ Adj Matrix ก่อนนำเข้าใช้ในโมเดล เพราะโครงข่ายคอนโวลูชันแบบกราฟไม่ใช้ข้อมูลรูปภาพเป็นข้อมูลนำเข้า

3) ส่วนที่นำผลลัพธ์ของส่วนการเรียนรู้ภาพและส่วนการเรียนรู้แท็กของภาพมารวมเข้าด้วยกันโดยใช้การคูณ เมื่อใช้โมเดลกำหนดแท็กให้แก่รูปภาพ โมเดลจะทำนายแท็กของรูปภาพดังกล่าวว่ามีความน่าจะเป็นที่จะประกอบด้วยแท็กเหล่านั้นเท่าใด



แผนภาพที่ 1 ภาพรวมโครงสร้างของโมเดล ML-GCN

4.4 วิธีการทดลอง

ผู้วิจัยใช้ภาพอนิเมะทั้งหมด 9,392 ภาพซึ่งประกอบด้วยแท็กที่ไม่ซ้ำกันทั้งหมด 16 แท็กในการรันโมเดล ML-GCN แต่ละแท็กมีภาพอนิเมะไม่น้อยกว่า 700 ภาพโดยผู้วิจัยแบ่งข้อมูลภาพอนิเมะ 80% เป็นข้อมูลฝึกสอน และ 20% เป็นข้อมูลทดสอบ ขนาดภาพอนิเมะที่ใช้สำหรับอัลกอริทึม ResNET-101 และResNeXT-101 คือ 500x500 แต่ EfficientNetB5 ใช้ขนาด 456x456 เพราะเป็นข้อกำหนดของ EfficientNetB5

ผู้วิจัยวัดประสิทธิภาพของโมเดล ML-GCN ที่ใช้ร่วมกับอัลกอริทึมทั้ง 3 แบบของโครงข่ายประสาทเทียมแบบคอนโวลูชันโดยใช้สมการ AP และ mAP

5. ผลการวิจัย

ผู้วิจัยทดสอบโมเดล ML-GCN ทั้ง 3 ที่ใช้อัลกอริทึม ResNET ResNeXT และ EfficientNet โดยดำเนินการทดสอบอัลกอริทึมละ 3 ครั้ง และกำหนดพารามิเตอร์ต่าง ๆ ดังนี้

1) จำนวนรอบ (Epoch) กำหนดเป็น 300 รอบเสมอ
2) ค่าอัตราการเรียนรู้ (Learning Rate) กำหนดค่าเริ่มต้นเป็น 0.1 เสมอ
3) การปรับค่าอัตราการเรียนรู้ (Epoch Step) โดยทุก ๆ x รอบที่โมเดลรันอัตราการเรียนรู้จะลดลง 0.1 เท่า เพื่อให้โมเดลเรียนรู้ลึกขึ้นเมื่อผ่านไป x รอบ โดยการทดสอบทั้ง 3 ครั้งของแต่ละอัลกอริทึมกำหนดให้ค่า x มีค่าเป็น 60, 80 และ 100 ตามลำดับ

ผลการทดลองแสดงในตารางที่ 1

ตารางที่ 1 การเปรียบเทียบการทดสอบ 3 อัลกอริทึมโดยการรันโมเดล 300 รอบ

ลำดับที่	อัลกอริทึม	ปรับค่าอัตราการเรียนรู้ ทุก x รอบ	เวลาที่ใช้ (นาที)	mAP
1	ResNET-101	80	468	82.47
2		110	467	81.91
3		160	471	81.70

ลำดับที่	อัลกอริทึม	ปรับค่าอัตราการเรียนรู้ ทุก x รอบ	เวลาที่ใช้ (นาที)	mAP
4	ResNeXT-101	80	561	82.63
5		110	561	82.88
6		160	558	83.54
7	EfficientNetB5	80	852	74.34
8		110	845	71.37
9		160	851	69.47

6. อภิปรายผล

จากการทดสอบอัลกอริทึมโครงข่ายประสาทเทียมแบบคอนโวลูชันทั้ง 3 แบบ ได้แก่ ResNET ResNeXT และ EfficientNet ผลปรากฏว่า ResNeXT มีค่า mAP หรือค่าความแม่นยำเฉลี่ยมากที่สุด ข้อสังเกตที่ ResNeXT ให้ผลลัพธ์ดีที่สุดอาจเนื่องมาจากโครงสร้างของResNeXT มีการใช้ Cardinality ซึ่งไม่มีใน ResNET และ EfficientNet โดย Cardinality คือการแยกส่วนข้อมูลเป็นหลายกิ่งเพื่อประมวลผลและนำมารวมเข้าด้วยกันในภายหลัง แต่ละกิ่งสามารถเรียนรู้คุณลักษณะของภาพอนิเมะ (Feature) ที่แตกต่างกัน ส่งผลให้โมเดลสามารถเรียนรู้ข้อมูลได้หลากหลายมากขึ้น ส่วน EfficientNet ไม่เหมาะใช้ร่วมกับโมเดล ML-GCN ในการจำแนกแท็กของภาพอนิเมะเพราะได้ค่า mAP น้อยและใช้เวลามาก

7. ข้อเสนอแนะ

7.1 ข้อเสนอแนะในการนำผลวิจัยไปใช้

งานวิจัยนี้มีจุดประสงค์เพื่อเปรียบเทียบอัลกอริทึมโครงข่ายประสาทเทียมแบบคอนโวลูชันทั้ง 3 แบบ ได้แก่ ResNET ResNeXT และ EfficientNet ผู้วิจัยคาดหวังว่าขั้นตอนวิธีดำเนินการวิจัยและผลลัพธ์ของงานวิจัยนี้จะเป็นประโยชน์แก่ผู้ที่ต้องการศึกษาเกี่ยวกับโมเดลแท็กภาพอนิเมะ

7.2 ข้อเสนอแนะในการวิจัยครั้งต่อไป

ขั้นตอนเตรียมข้อมูลภาพอนิเมะเป็นขั้นตอนที่ยากและใช้เวลามากที่สุด เพราะต้องตรวจสอบภาพอนิเมะทีละภาพเพื่อให้มั่นใจว่าแต่ละภาพมีแท็กที่ถูกต้องก่อนใช้ในโมเดล ดังนั้นงานวิจัยในอนาคตควรเลือกชุดข้อมูลสอนที่มั่นใจว่ามีการแท็กภาพอย่างถูกต้อง และอาจใช้ภาพอนิเมะที่มีความหลากหลายและจำนวนมากขึ้นเพื่อใช้สร้างโมเดลที่มีความแม่นยำ

8. เอกสารอ้างอิง

K. He, Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

S. Xie, Girshick, R., Tu, Z., He, K., & Dollar, P. (2017). Aggregated Residual Transformations for Deep Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

T. N. Kipf, & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. International Conference on Learning Representations (ICLR).

Chen, Z.-M., Wei, X.-S., Wang, P., & Guo, Y. (2019). Multi-label image recognition with graph convolutional networks. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

M. Tan, & Le, Q. V. (2020). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning.

P. Deng, Ren, J., Lv, S., Feng, J., & Kang, H. (2020). Multi-Label Image Recognition in Anime Illustration with Graph Convolutional Networks.

R. A. Jha. (2021). Mastering PyTorch. Packt Publishing Ltd. Birmingham. Livery Place.

F. Yi, Wu, J. Zhao, M., & Zhou, S. (2023). Anime Character Identification and Tag Prediction by Multimodality Modeling: Dataset and Model. International Joint Conference on Neural Networks (IJCNN).

Z. Lan, Maeda, K., Ogawa, T., & Haseyama, M. (2023). Hierarchical Multi-Label Attribute Classification With Graph Convolutional Networks on Anime Illustration.

ประวัติย่อของผู้วิจัย

ชื่อ-สกุล	อดิเทพ พรหมพา
วัน เดือน ปี เกิด	26 สิงหาคม 2542
สถานที่เกิด	จังหวัดชลบุรี
สถานที่อยู่ปัจจุบัน	99/36 หมู่ 9 ถ.สุขุมวิท ต.บางพระ อ.ศรีราชา จ.ชลบุรี
ตำแหน่งและประวัติการทำงาน	นักศึกษา
ประวัติการศึกษา	ปริญญาตรี สาขาวิศวกรรมซอฟต์แวร์ มหาวิทยาลัยบูรพา