



เว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย
A Web-based Visual Search for Ancient Artifacts

นายธนาชนนท์ คำวัน

นายสหวัชร รอดกลาง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2562

ประโยชน์ของเรื่อง : เว็บไซต์ค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย
 A Web-based Visual Search for Ancient Artifacts
 ผู้เสนอโครงการ : 1. นายธนากร คำวัน
 2. นายสหัส รอดกลาง
 สาขา : วิทยาการคอมพิวเตอร์
 ภาควิชา : วิทยาการคอมพิวเตอร์และสารสนเทศ
 คณะ : วิทยาศาสตร์ประยุกต์
 อาจารย์ที่ปรึกษา : 1. ผู้ช่วยศาสตราจารย์ สถิตย์ ประสมพันธ์
 2. ผู้ช่วยศาสตราจารย์ ดร.ลือพล พิพานเมฆาภรณ์
 ปีการศึกษา : 2562

คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ อนุมัติให้
 ประโยชน์ของเรื่องนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชา
 วิทยาการคอมพิวเตอร์ประยุกต์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ สถิตย์ ประสมพันธ์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ลือพล พิพานเมฆาภรณ์)

.....กรรมการ

(อาจารย์ ดร.สรวิชัย รัตนสัญญา)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ธนภัทร อนุศาสน์อมรกุล)

.....หัวหน้าภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ

(ผู้ช่วยศาสตราจารย์ ดร.อัศรา ประโยชน์)

ลิขสิทธิ์ของภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์
 มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2562

(ก)

บทคัดย่อ

ปัจจุบันกรมศิลปากรมีโบราณวัตถุมากมายหลายล้านชิ้น การที่ผู้คนทั่วไปอยากทราบถึงข้อมูลของวัตถุนั้น ๆ เป็นเรื่องที่ยาก หากจะต้องสอบถามไปยังผู้รู้ จะใช้เวลาเป็นอย่างมาก นอกจากนี้ทางกรมศิลปากรเองก็ยังไม่มียระบบค้นหาข้อมูลที่รองรับข้อมูลมากมายหลายล้านข้อมูล จึงทำให้เกิดระบบเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่ายขึ้นมานั่นเอง

ระบบเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย สร้างขึ้นเพื่ออำนวยความสะดวกแก่ผู้ที่ต้องการค้นหาข้อมูลของโบราณวัตถุนั้น ๆ เป็นเว็บการค้นหารูปด้วยรูป (Input นำเข้าเป็นรูปภาพ และแสดงผลออกมาเป็นรูปภาพที่ใกล้เคียง 3 อันดับ) ที่มีข้อมูลจากกรมศิลปากรโดยตรง คาดว่า จะสามารถเผยแพร่ความรู้ ข้อมูลให้แก่ผู้ใช้ได้อย่างมากมาย

ผลการทดลองพบว่าอัลกอริทึมที่นำเสนอสามารถให้ผลลัพธ์ในการค้นหาข้อมูลด้วยค่า Accuracy 80% โดยคาดว่าจะจะเป็นเครื่องมือสำหรับผู้ใช้ในการค้นหาข้อมูลรูปภาพที่เกิดประโยชน์ต่อผู้ใช้ได้เป็นอย่างดี

กิตติกรรมประกาศ

การจัดทำปฏิญานพนธ์เรื่องเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย (A Web-based Visual Search for Ancient Artifacts) สำเร็จลุล่วงได้เนื่องจากความอนุเคราะห์ และคำแนะนำจากอาจารย์ที่ปรึกษาปฏิญานพนธ์ ผู้ช่วยศาสตราจารย์ สติย์ ประสมพันธ์ ผู้ช่วยศาสตราจารย์ ดร.ลือพล พิพานเมฆาภรณ์ และผู้ช่วยศาสตราจารย์ ดร.เนียบวุฒิ รัตนวิไลสกุล ที่ได้สละเวลาให้คำปรึกษา แนะนำ ตรวจสอบและแก้ไขปัญหาตลอดการดำเนินงาน อันเป็นประโยชน์อย่างยิ่งในการทำงาน คณะผู้จัดทำรู้สึกซาบซึ้งในความกรุณาขอบคุณทุกท่านเป็นอย่างยิ่ง จึงขอกราบขอบพระคุณไว้ ณ โอกาสนี้

ขอกราบขอบพระคุณ กรมศิลปากร ที่อำนวยความสะดวกโดยได้ส่งข้อมูลโบราณวัตถุและภาพถ่าย ให้คณะผู้จัดทำได้นำข้อมูลมาใช้งาน ซึ่งเป็นประโยชน์ต่อการทำปฏิญานพนธ์ให้ประสบความสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้คณะผู้จัดทำขอขอบพระคุณบุคคลในครอบครัวทุกท่าน ที่เปิดโอกาสให้ได้รับการศึกษาเล่าเรียน ตลอดจนคอยช่วยเหลือและให้กำลังใจคณะผู้จัดทำเสมอมาจนสำเร็จการศึกษา คณะผู้จัดทำรู้สึกซาบซึ้งในความปรารถนาดีของทุกท่านเป็นอย่างยิ่ง จึงขอกราบขอบพระคุณไว้ในโอกาสนี้

นายธนันท์ คำวัน

นายสหวัชร รอดกลาง

2 เมษายน 2563

สารบัญ

	หน้า
บทคัดย่อ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญ (ต่อ)	จ
สารบัญตาราง	ฉ
สารบัญภาพ	ช
สารบัญภาพ (ต่อ)	ซ
สารบัญภาพ (ต่อ)	ฌ
สารบัญภาพ (ต่อ)	ญ
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตและข้อจำกัดของโครงการ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 แผนการดำเนินงาน	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 การประมวลผลภาพ (Image Processing)	3
2.2 การประมาณค่าความเข้มของแสง	3
2.3 การประมวลผลภาพเพื่อการค้นคืนภาพถ่าย	4
2.4 โครงข่ายประสาทเทียม (Artificial Neural Network)	5
2.5 อัลกอริทึมการเรียนรู้ของโครงข่ายประสาทเทียม	8
2.6 โครงข่ายประสาทเทียมเชิงลึก (Deep Neural Network)	9
2.7 กระบวนการฝึกฝนโครงข่ายประสาทเทียม	11
2.8 การทำดัชนีภาพ (Image Indexing)	26
บทที่ 3 ขั้นตอนการดำเนินงาน	
3.1 System Architecture	29
3.2 ขั้นตอนการดำเนินงาน	30

สารบัญ (ต่อ)

	หน้า
3.3 Use Case Diagram	33
3.4 Use Case Description	34
3.5 Activity Diagram	37
3.6 Sequence Diagram	38
3.7 การออกแบบการทดลอง	39
บทที่ 4 การพัฒนาระบบ	
4.1 เครื่องมือที่ใช้ในการพัฒนาระบบ	43
4.2 การพัฒนาระบบ	44
บทที่ 5 ผลการดำเนินงาน	
5.1 ผลการดำเนินงาน	72
5.2 การแสดงผลหน้าเว็บเพจ	77
บทที่ 6 บทสรุปและแนวทางพัฒนาต่อ	
6.1 สรุปผลการดำเนินงาน	81
6.2 ปัญหาและข้อจำกัดในการดำเนินงาน	87
6.3 แนวทางในการพัฒนาต่อในอนาคต	88
บรรณานุกรม	89

สารบัญตาราง

	หน้า
ตารางที่ 2-1 แสดงจำนวนพารามิเตอร์และความแม่นยำบน Top-1 และ Top-5	14
ตารางที่ 2-2 รายการประเภทวัสดุจำแนกตามฐานข้อมูลกรมศิลปากร	15
ตารางที่ 2-2 รายการประเภทวัสดุจำแนกตามฐานข้อมูลกรมศิลปากร (ต่อ)	16
ตารางที่ 2-2 รายการประเภทวัสดุจำแนกตามฐานข้อมูลกรมศิลปากร (ต่อ)	17
ตารางที่ 2-2 รายการประเภทวัสดุจำแนกตามฐานข้อมูลกรมศิลปากร (ต่อ)	18
ตารางที่ 2-2 รายการประเภทวัสดุจำแนกตามฐานข้อมูลกรมศิลปากร (ต่อ)	19
ตารางที่ 2-3 รายการประเภทศิลปะจำแนกตามฐานข้อมูลกรมศิลปากร	20
ตารางที่ 2-3 รายการประเภทศิลปะจำแนกตามฐานข้อมูลกรมศิลปากร (ต่อ)	21
ตารางที่ 2-3 รายการประเภทศิลปะจำแนกตามฐานข้อมูลกรมศิลปากร (ต่อ)	22
ตารางที่ 3-1 ภาพรวมของการดำเนินงาน	30
ตารางที่ 3-1 ภาพรวมของการดำเนินงาน (ต่อ)	31
ตารางที่ 3-2 Use case Description : ค้นหาข้อมูลรูปภาพโบราณวัตถุด้วยภาพ	34
ตารางที่ 3-2 Use case Description : ค้นหาข้อมูลรูปภาพโบราณวัตถุด้วยภาพ (ต่อ)	35
ตารางที่ 3-3 Use case Description : ค้นหาข้อมูลรูปภาพโบราณวัตถุจากวัสดุในการสร้าง	35
ตารางที่ 3-4 Use case Description : ค้นหาข้อมูลรูปภาพโบราณวัตถุจากประเภทของศิลปะ	36
ตารางที่ 3-5 แสดงประเภทชุดข้อมูลของภาพทดลอง	39
ตารางที่ 3-5 แสดงประเภทชุดข้อมูลของภาพทดลอง (ต่อ)	40
ตารางที่ 5-1 แสดงผลลัพธ์ 3 อันดับ จากการนำภาพเข้าทดสอบ	73
ตารางที่ 5-1 แสดงผลลัพธ์ 3 อันดับ จากการนำภาพเข้าทดสอบ (ต่อ)	74

สารบัญภาพ

	หน้า
ภาพที่ 2-1 แสดงการไล่เนตสี	3
ภาพที่ 2-2 ลำดับการนับของบิตในเลขฐานสอง	4
ภาพที่ 2-3 จุดภาพที่ถูกแทนที่ด้วยตัวเลขที่บอกถึงค่าสีตั้งแต่ 0-255	4
ภาพที่ 2-4 โครงสร้างข่ายงานประสาทเทียม	6
ภาพที่ 2-5 สถาปัตยกรรมของข่ายงาน (Network Architecture)	6
ภาพที่ 2-6 โครงข่ายแบบหลายชั้น	7
ภาพที่ 2-7 วิธีค้นหาในทิศทางที่ลาดชันที่สุด (Steepest Descent Method)	9
ภาพที่ 2-8 โครงข่ายประสาทเทียมเชิงลึก (Deep Neural Network)	10
ภาพที่ 2-9 Deep Learning สำหรับการเรียนรู้จดจำใบหน้า	10
ภาพที่ 2-10 แสดงกระบวนการสร้างโมเดล AI เพื่อใช้ในการสืบค้นข้อมูลโบราณวัตถุ	11
ภาพที่ 2-11 ขั้นตอนการสร้างโมเดล AI	12
ภาพที่ 2-12 โครงสร้างทั่วไปของโครงข่ายประสาทเทียมแบบ efficientnet	12
ภาพที่ 2-13 แสดงโครงสร้างและจำนวนพารามิเตอร์ของโมเดล B0	13
ภาพที่ 2-14 แสดงการผลการเปรียบเทียบความเร็วและความแม่นยำของโมเดล efficientnet B0 ถึง B6 และโครงข่ายประสาทเทียมคอนโวลูชันแบบอื่นบนฐานข้อมูลภาพ ImageNet	13
ภาพที่ 2-15 แสดงการทำงานของฟังก์ชัน SoftMax	23
ภาพที่ 2-16 หน้าจอการฝึกฝนโครงข่ายประสาทเทียม efficientnet บนเครื่องแม่ข่าย	25
ภาพที่ 2-17 การเข้ารหัสภาพโบราณวัตถุเพื่อการทำดัชนีภาพ	26
ภาพที่ 2-18 แสดงตัวอย่างการสร้าง Binary Code เพื่อแบ่งกลุ่มข้อมูล	27
ภาพที่ 2-19 การแบ่งกลุ่มข้อมูลโดยใช้ Binary Code ความยาว 15 บิต	28
ภาพที่ 3-1 System Architecture	29
ภาพที่ 3-2 ขั้นตอนการสร้างโมเดล AI	31
ภาพที่ 3-3 การเข้ารหัสภาพโบราณวัตถุเพื่อการทำดัชนีภาพ	32
ภาพที่ 3-4 Use Case Diagram ของเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย	33
ภาพที่ 3-5 Activity Diagram	37
ภาพที่ 3-6 Sequence Diagram	38

สารบัญภาพ (ต่อ)

	หน้า
ภาพที่ 3-7 ภาพรวมของกระบวนการทดสอบระบบ	41
ภาพที่ 3-8 Network Architecture โครงสร้าง Neural Network	42
ภาพที่ 4-1 โค้ดแสดงการสร้างข้อมูล Output	44
ภาพที่ 4-2 โค้ดแสดงการสร้างข้อมูล Input	45
ภาพที่ 4-3 โค้ดแสดงการแปลงเป็น one hot vector	45
ภาพที่ 4-4 โค้ดแสดงการเตรียมข้อมูลสำหรับนำไปเทรน	46
ภาพที่ 4-5 โค้ดแสดงการเตรียมข้อมูลสำหรับนำไปเทรน และการแบ่งข้อมูล train และ test	47
ภาพที่ 4-6 โค้ดแสดงการคำนวณความแม่นยำสำหรับการเทรน	48
ภาพที่ 4-7 โค้ดแสดงการปรับส่วนของโมเดล VGG16	49
ภาพที่ 4-8 โค้ดแสดงการเทรนโมเดล	50
ภาพที่ 4-9 โค้ดแสดงการสร้าง Argument	50
ภาพที่ 4-10 โค้ดแสดงการคำนวณความแม่นยำสำหรับการทำดัชนีภาพ	51
ภาพที่ 4-11 โค้ดแสดงการโหลดข้อมูลจากฐานข้อมูล	52
ภาพที่ 4-12 โค้ดแสดงการโหลดข้อมูลรูปภาพ	52
ภาพที่ 4-13 โค้ดแสดงการ Normalize	53
ภาพที่ 4-14 โค้ดแสดงการสร้างดัชนีภาพ	53
ภาพที่ 4-14 โค้ดแสดงการสร้างดัชนีภาพ (ต่อ)	54
ภาพที่ 4-15 โค้ดแสดงการเรียกฟังก์ชันสำหรับสร้างดัชนีภาพ	54
ภาพที่ 4-16 โค้ดแสดงการเริ่มการทำงานของตัวหน้าเว็บเพจ	55
ภาพที่ 4-17 โค้ดแสดงการโหลดโมเดลจาก Argument มาใช้	55
ภาพที่ 4-18 โค้ดแสดงการโหลดฐานข้อมูล	56
ภาพที่ 4-19 โค้ดแสดงการทำงานส่วนการรับข้อมูลรูปภาพ	57
ภาพที่ 4-20 โค้ดแสดงการโหลดข้อมูลรูปภาพ	57
ภาพที่ 4-21 โค้ดแสดงการ Normalize	58
ภาพที่ 4-22 โค้ดแสดงการค้นหา feature ในดัชนีภาพ	58
ภาพที่ 4-23 โค้ดแสดงการสร้างไฟล์ Json ที่มีผลการทำนาย 3 อันดับ	59
ภาพที่ 4-24 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (1)	59

สารบัญภาพ (ต่อ)

	หน้า
ภาพที่ 4-24 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (1) (ต่อ)	60
ภาพที่ 4-25 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (2)	61
ภาพที่ 4-26 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (3)	61
ภาพที่ 4-27 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (4)	62
ภาพที่ 4-28 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (5)	63
ภาพที่ 4-29 โค้ดแสดงการทำงานของส่วนหน้าเว็บเพจ	64
ภาพที่ 4-30 โค้ดแสดงการทำงานของส่วนหลังจากรับข้อมูลจาก backend	65
ภาพที่ 4-31 โค้ดการสร้างตารางแสดงผล	66
ภาพที่ 4-32 โค้ดการแสดงผลข้อมูลในตารางแสดงข้อมูล	66
ภาพที่ 4-32 โค้ดการแสดงผลข้อมูลในตารางแสดงข้อมูล (ต่อ)	67
ภาพที่ 4-33 โค้ดการแสดงผลข้อมูลของหน้า culture.html	68
ภาพที่ 4-34 โค้ดการสร้างตารางแสดงผลของข้อมูลของหน้า culture.html	68
ภาพที่ 4-34 โค้ดการสร้างตารางแสดงผลของข้อมูลของหน้า culture.html (ต่อ)	69
ภาพที่ 4-35 โค้ดการแสดงผลข้อมูลของหน้า material.html	70
ภาพที่ 4-36 โค้ดการสร้างตารางแสดงผลของข้อมูลของหน้า material.html	70
ภาพที่ 5-1 แสดงผลการนำเข้าชุดภาพทดลอง (1)	75
ภาพที่ 5-2 แสดงผลการนำเข้าชุดภาพทดลอง (2)	76
ภาพที่ 5-3 หน้าเริ่มต้นเว็บเพจ	77
ภาพที่ 5-4 หน้าแสดงผลผลการทำนายรูปภาพใกล้เคียง (1)	78
ภาพที่ 5-5 หน้าแสดงผลผลการทำนายรูปภาพใกล้เคียง (2)	79
ภาพที่ 5-6 หน้าแสดงผลผลการค้นหารูปภาพจากประเภทศิลปะ และวัสดุในการสร้าง	80
ภาพที่ 6-1 ตัวอย่างแบบประเมินความพึงพอใจในการใช้งานระบบ	82
ภาพที่ 6-2 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (1)	82
ภาพที่ 6-3 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (2)	83
ภาพที่ 6-4 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (3)	83
ภาพที่ 6-5 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (4)	84
ภาพที่ 6-6 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (5)	84

สารบัญภาพ (ต่อ)

	หน้า
ภาพที่ 6-7 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (6)	85
ภาพที่ 6-8 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (7)	86
ภาพที่ 6-9 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (8)	87
ภาพที่ 6-10 ตัวอย่างรูปภาพมีปัญหา (1)	87
ภาพที่ 6-11 ตัวอย่างรูปภาพมีปัญหา (2)	88
ภาพที่ 6-12 ตัวอย่างรูปภาพมีปัญหา (3)	88

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ในอดีตก่อน พ.ศ. ๒๕๕๔ มรดกศิลปวัฒนธรรมไทย โบราณสถาน โบราณวัตถุ ศิลปวัตถุ ประวัติศาสตร์ วรรณกรรม การละคร ดนตรี ฯลฯ อยู่ในความรับผิดชอบของหน่วยงาน กรม กระทรวง ต่าง ๆ อย่างหลากหลาย ไม่มีการรวบรวม จัดไว้ใน ความรับผิดชอบ ของหน่วยงานใด

จนกระทั่งในรัชสมัยของ พระบาทสมเด็จพระมงกุฎเกล้าเจ้าอยู่หัว พระองค์ ทรงเล็งเห็น ความสำคัญ "มรดกศิลปวัฒนธรรม" จึงทรงมีพระราชดำริ ให้โอนกิจการ จากกระทรวงวัง และกรม พิพิธภัณฑ์ จากกระทรวงธรรมการ มาจัดตั้งเป็น "กรมศิลปากร"

ปัจจุบันกรมศิลปากรมีโบราณวัตถุมากมายหลายล้านชิ้น การที่ผู้คนอยากทราบข้อมูลของวัตถุ นั้น ๆ เป็นเรื่องที่ยาก หากจะต้องสอบถามไปยังผู้รู้ ก็เสียเวลา และทางกรมศิลปากรเองก็ยังไม่ มีระบบค้นหาข้อมูลที่รองรับข้อมูลมากมายหลายล้าน จึงทำให้เกิดระบบเว็บสืบค้นข้อมูลโบราณวัตถุ ผ่านการใช้ภาพถ่ายชิ้นนั้นนั่นเอง

ระบบเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย สร้างขึ้นเพื่ออำนวยความสะดวกแก่ผู้ที่ ต้องการค้นหาข้อมูลของโบราณวัตถุนั้น ๆ เป็นเว็บการค้นหารูปด้วยรูป (Input นำเข้าเป็นรูปภาพ และแสดงผลออกมาเป็นรูปภาพที่ใกล้เคียง 3 อันดับ) ที่มีข้อมูลจากทางกรมศิลปากรโดยตรง คาดว่า จะสามารถเผยแพร่ความรู้ ข้อมูลให้แก่ผู้ใช้ได้อย่างมากมาย

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อพัฒนาเว็บแอปพลิเคชัน สำหรับการสืบค้นข้อมูล โดยอาศัยเทคนิคทางด้าน Machine Learning

1.2.2 เพื่อศึกษาโบราณวัตถุ สำหรับตรวจสอบว่าวัตถุนั้น ๆ เกิดขึ้นในยุคสมัยใด สร้างจากวัสดุ ใด หรือมีที่มาอย่างไร

1.2.3 เพื่อให้ผู้ใช้งานจะสามารถนำสิ่งที่พบเจอ (เช่น พบโบราณวัตถุ หรือพบภาพถ่าย โบราณวัตถุ) นำมาใช้ศึกษาและตรวจสอบได้จากเว็บแอปพลิเคชันที่จัดทำได้อย่างสะดวกมากกว่าเดิม

1.3 ขอบเขตและข้อจำกัดของโครงการ

1.3.1 หน้าเว็บเพจสำหรับแสดงผลจากการค้นหาข้อมูล โดยกระบวนการ Machine Learning

1.3.1.1 สร้างหน้าเว็บเพจ สำหรับการ Browse Image เพื่อนำภาพไปสู่กระบวนการ Machine Learning

1.3.1.2 เชื่อมต่อเว็บเพจกับฐานข้อมูล เพื่อนำข้อมูลออกมาแสดงผลได้

1.3.1.3 สามารถแสดงผลข้อมูล (เช่น รูปภาพ และรายละเอียด) ผ่านหน้าเว็บเพจได้

1.3.2 โมเดล AI สำหรับการ Train รูปภาพตัวอย่าง และเป็นโมเดลที่ใช้ในการตรวจสอบภาพที่นำเข้ามาค้นหา

1.3.2.1 สร้างโมเดล AI สำหรับการ Train รูปภาพตัวอย่างได้

1.3.2.2 สามารถทำการประมวลผลภาพ (Image Processing) จาก Input และ Output ที่นำเข้า ได้อย่างสมบูรณ์

1.3.2.3 สามารถทำการค้นหารูปภาพที่มีความใกล้เคียงกับรูปภาพที่ถูกนำเข้า (Input) ได้อย่างรวดเร็วและแม่นยำโดยการทำดัชนีภาพ (image indexing) ได้

1.3.3 จัดทำเอกสารสำหรับส่งต่อนักพัฒนาคนอื่น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ผู้ใช้สามารถใช้ภาพถ่ายโบราณวัตถุในการค้นหาข้อมูลของภาพถ่ายชิ้นนั้น ๆ ได้

1.4.2 เพื่อเป็นการลดระยะเวลาในการสืบค้นข้อมูลจากภาพถ่าย

1.4.3 โครงการนี้จะช่วยแก้ปัญหาให้แก่ ผู้ที่ต้องการจะศึกษา เรียนรู้เกี่ยวกับโบราณวัตถุ

1.5 แผนการดำเนินงาน

1.5.1 ศึกษาหลักการทำงานของ Machine Learning

1.5.2 ศึกษาการทำงาน python โดยใช้ Flask Framework

1.5.3 สร้างตัวโมเดล จากการศึกษา Machine Learning

1.5.4 เทรนโมเดลให้มีความแม่นยำ

1.5.5 ทดลองและปรับปรุงตัวโมเดลให้แม่นยำมากขึ้น

1.5.6 จัดทำเอกสารสำหรับส่งต่อนักพัฒนาคนอื่น

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

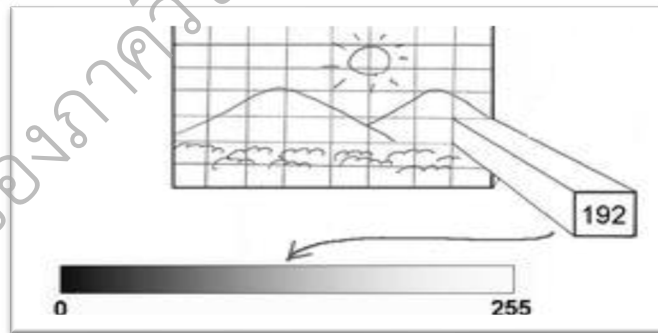
2.1 การประมวลผลภาพ (Image Processing)

การประมวลผลภาพ (Image Processing) คือ การทำงานอย่างใดอย่างหนึ่งกับภาพต้นฉบับ (Input Image) เพื่อให้ได้ภาพผลลัพธ์ (Output Image) มีลักษณะของภาพที่เป็นไปตามที่ต้องการซึ่งการกระทำกับภาพที่ใช้ในการประมวลผลภาพดิจิทัลมีอยู่มากมายหลายแบบ ความเข้าใจเกี่ยวกับคุณลักษณะและการแยกแยะประเภทของการกระทำกับภาพ จะช่วยให้เราสามารถคาดคะเนภาพผลลัพธ์ที่ได้จากการกระทำแต่ละแบบ หรือประมาณความซับซ้อนของการกระทำกับภาพที่น่าจะนำไปใช้ได้ การกระทำกับภาพในการประมวลผลภาพดิจิทัลสามารถแบ่งออกได้สองประเภทคือ

- การประมวลผลภาพแบบจุด (Point Image Processing)
- การประมวลผลภาพแบบบริเวณ (Local Image Processing)

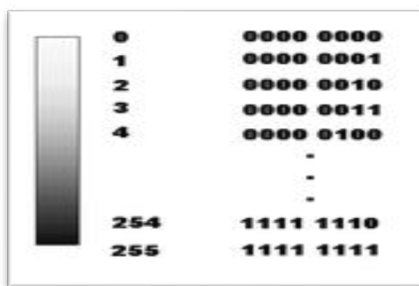
2.2 การประมาณค่าความเข้มของแสง

เมื่อได้ภาพจากการ Sampling มาแล้ว แต่ละจุดในภาพจะถูกแทนด้วยสีภาพในโทนสีเทา หรือ Grayscale จะประกอบไปด้วยสีดำ และไล่เฉดสีจางลงไปจนถึงสีขาวดังภาพที่ 2-1



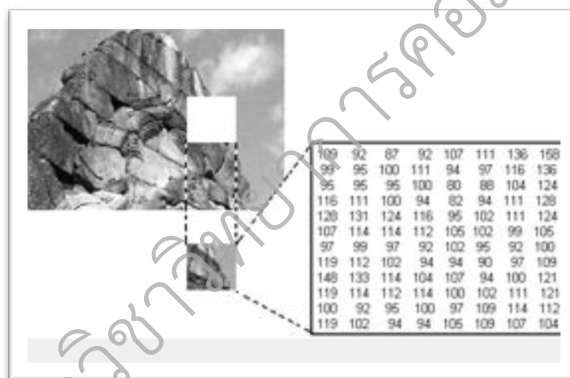
ภาพที่ 2-1 แสดงการไล่เฉดสี

สีดำจะแทนด้วยค่าตัวเลข 255 สีขาวจะแทนด้วยค่าตัวเลขคือ 0 รวมทั้งสิ้น 256 ระดับสี (0-255) หรือ 2 กำลัง 8 โดยที่ 8 ก็คือ จำนวนบิตในหน่วยความจำที่ใช้ในการเก็บค่านี้หนึ่งค่า เพราะฉะนั้น สีดำจะถูกแทนด้วยรหัสในเลขฐานสองคือ 00000000 และสีขาวก็就会被แทนด้วยรหัส 11111111 และสีที่อยู่ตรงกลางระหว่างสีดำกับสีขาวก็จะไล่ไปตามลำดับการนับของบิตในเลขฐานสองดังภาพที่ 2-2



ภาพที่ 2-2 ลำดับการนับของบิตในเลขฐานสอง

ถ้าภาพเป็นแบบโทนขาวดำ (Grayscale) แต่ละจุดภาพก็จะถูกแทนที่ด้วยตัวเลขที่บอกถึงค่าสี ตั้งแต่ 0-255 ดังภาพที่ 2-3



ภาพที่ 2-3 จุดภาพที่ถูกแทนที่ด้วยตัวเลขที่บอกถึงค่าสีตั้งแต่ 0-255

จะเห็นได้ว่า แต่ละจุด ๆ จะถูกแทนที่ด้วยตัวเลข ซึ่งตัวเลขเหล่านี้ก็อยู่ระหว่าง 0-255 คือตั้งแต่ 0,1,2,3,4 255 เป็นโทนสีเทา แต่ถ้าเป็นภาพขาวดำจะมีอยู่ด้วยกันแค่ 2 สีคือ สีดำ แทนด้วยเลข 0 กับสีขาวแทนด้วยเลข 255

2.3 ประมวลผลภาพเพื่อการค้นคืนภาพถ่าย

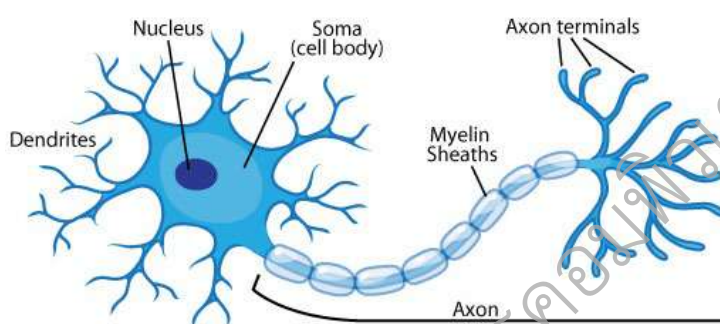
ลักษณะเฉพาะของภาพเป็นคุณสมบัติที่สามารถหาได้ด้วยขั้นตอนวิธีทางการประมวลผลภาพ (Image Processing Algorithm) ลักษณะเฉพาะพื้นฐานของภาพ ประกอบไปด้วย 3 ส่วนได้แก่

1. สี (Color) เป็นลักษณะเฉพาะที่มีบทบาทสำคัญในระบบค้นคืนภาพ เนื่องจากสีเป็นสิ่งที่โดดเด่นสามารถสังเกตเห็นได้ชัดเจนจากการมองภาพ นอกจากนี้ยังสามารถช่วยในการแยกแยะสิ่งต่าง ๆ ภายในภาพออกจากกันได้ดี เช่น สีฟ้าของท้องฟ้า สีเขียวของต้นไม้หรือใบไม้ เป็นต้น
2. รูปร่าง (Shape) เป็นลักษณะเฉพาะของภาพที่ใช้อธิบายถึงรูปร่างและลักษณะรวมถึงขนาดของวัตถุภายในภาพ ซึ่งทำให้สามารถแยกแยะระหว่างวัตถุที่มีรูปร่างแตกต่างกันออกจากกันได้
3. พื้นผิว (Texture) เป็นลักษณะเฉพาะที่ใช้อธิบายความหยาบ ความละเอียด หรือความซับซ้อนของวัตถุภายในภาพ ซึ่งภาพแต่ละภาพอาจจะประกอบไปด้วยวัตถุที่มีลักษณะพื้นผิวที่แตกต่างกันออกไป การวิเคราะห์พื้นผิวจะช่วยให้สามารถแยกแยะความแตกต่างของวัตถุได้ดียิ่งขึ้น เช่น การแยกแยะความแตกต่างระหว่างภาพเสือดาว ซึ่งมีลายเป็นจุด กับเสือชนิดอื่นซึ่งมีลายเป็นริ้วหรือแถบยาว ออกจากกันได้ เป็นต้น

2.4 โครงข่ายประสาทเทียม (Artificial Neural Network)

โครงข่ายประสาทเทียม (Artificial Neural Network) หรือที่มักจะเรียกสั้น ๆ ว่า ข่ายงานประสาท (Neural Network หรือ Neural Net) คือโมเดลทางคณิตศาสตร์ สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบ Connectionist เพื่อจำลองการทำงานของเครือข่ายประสาทในสมองมนุษย์ ด้วยวัตถุประสงค์ที่จะสร้างเครื่องมือซึ่งมีความสามารถในการเรียนรู้การจดจำแบบรูป (Pattern Recognition) และการอุปมาความรู้เช่นเดียวกับความสามารถที่มีในสมองมนุษย์ แนวคิดเริ่มต้นของเทคนิคนี้ได้มาจากการศึกษาข่ายงานไฟฟ้าชีวภาพ (Bioelectric Network) ในสมอง ซึ่งประกอบด้วย เซลล์ประสาท หรือ นิวรอน (Neurons) และ จุดประสานประสาท (Synapses) แต่ละเซลล์ประสาทประกอบด้วยปลายในการรับกระแสประสาท เรียกว่า เดนไดรต์ (Dendrite) ซึ่งเป็น input และปลายในการส่งกระแสประสาทเรียกว่า แอกซอน (Axon) ซึ่งเป็นเหมือน output ของเซลล์ เซลล์เหล่านี้ทำงานด้วยปฏิกิริยาไฟฟ้าเคมี เมื่อมีการกระตุ้นด้วยสิ่งเร้าภายนอกหรือกระตุ้นด้วยเซลล์ด้วยกัน กระแสประสาทจะวิ่งผ่านเดนไดรต์เข้าสู่นิวเคลียสซึ่งจะเป็นตัวตัดสินใจว่าต้องกระตุ้นเซลล์อื่น ๆ ต่อหรือไม่ ถ้ากระแสประสาทแรงพอ นิวเคลียสก็จะกระตุ้นเซลล์อื่น ๆ ต่อไปผ่านทางแอกซอนของมันตามโมเดลนี้ข่ายงานประสาทเกิดจากการเชื่อมต่อระหว่างเซลล์ประสาท จนเป็นเครือข่ายที่ทำงานร่วมกัน

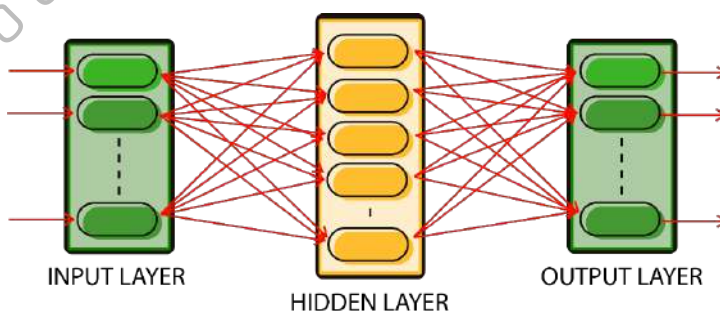
หลักการสำคัญของโครงข่ายประสาทเทียม คือ ความพยายามที่จะลอกเลียนแบบการทำงานของเซลล์ประสาทในสมองมนุษย์เพื่อทำงานได้อย่างมีประสิทธิภาพลักษณะทั่วไปของโครงข่ายประสาทเทียม คือ การที่โหนดต่าง ๆ จำลองมาจากไซแนปของเซลล์ประสาทระหว่างเดนไดรต์ทำหน้าที่รับสัญญาณประสาทจากเซลล์ข้างเคียงแอกซอน ประมวลผลผลลัพธ์ส่งไปยังเซลล์อื่น มีฟังก์ชันเป็นตัวกำหนดสัญญาณส่งออก (Activation Functions or Transfer Function)



ภาพที่ 2-4 โครงสร้างข่ายงานประสาทเทียม

สถาปัตยกรรมของข่ายงาน (Network Architecture) โครงข่ายประสาทเทียม ประกอบด้วยชั้นประมวลผลข้อมูลอย่างน้อย 3 ชั้น ได้แก่

- ชั้นอินพุต (Input layer)
- ชั้นซ่อน (Hidden layer)
- ชั้นเอาต์พุต (Output layer)



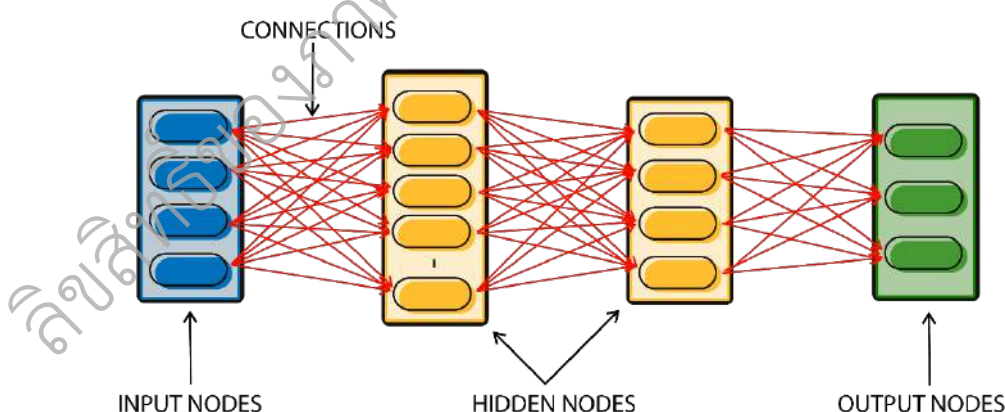
ภาพที่ 2-5 สถาปัตยกรรมของข่ายงาน (Network Architecture)

สถาปัตยกรรมของข่ายงาน (Network architecture) แบ่งประเภทของโครงข่ายประสาทเทียมตามจำนวนชั้นของโครงข่ายได้ 2 แบบ ได้แก่

- โครงข่ายแบบชั้นเดียว (Single layer)
- โครงข่ายแบบหลายชั้น (Multi layer)

1. โครงข่ายแบบชั้นเดียว (Single layer) เป็นโครงข่ายประสาทเทียมอย่างง่ายที่มีเพียงชั้นรับข้อมูลป้อนเข้าและชั้นส่งข้อมูลออกเท่านั้น โหนดในชั้นรับข้อมูลป้อนเข้าทำหน้าที่รับข้อมูลเข้า (Input value) แล้วส่งข้อมูลผ่านเส้นเชื่อมโยงต่าง ๆ ไปให้โหนดในชั้นส่งข้อมูลออก ความเข้มของสัญญาณหรือปริมาณข้อมูลที่นำเข้าสู่โหนดในชั้นส่งข้อมูลออกจะขึ้นอยู่กับค่าน้ำหนักที่อยู่บนเส้นเชื่อมโยง โหนดในชั้นส่งข้อมูลออกจะนำข้อมูลที่ได้รับมาคำนวณโดยใช้ฟังก์ชันทางคณิตศาสตร์ที่เรียกว่า ฟังก์ชันการแปลง (Transfer function) ที่เหมาะสมกับปัญหา แล้วส่งผลลัพธ์ที่ได้ออกมาเป็นข้อมูลส่งออก เช่น โครงข่ายแบบชั้นเดียวแบบเพอเซปตรอนอย่างง่าย (Simple perceptron) โครงข่ายโฮปฟิลด์ (Hopfield networks)

2. โครงข่ายแบบหลายชั้น (Multi layer) เป็นโครงข่ายที่มีชั้นแอบแฝงตั้งแต่ 1 ชั้นขึ้นไป โครงข่ายแบบหลายชั้นจะใช้ในกรณีที่ปัญหาที่มีความซับซ้อน ซึ่งโครงข่ายแบบชั้นเดียวไม่สามารถแก้ปัญหาได้ จึงเพิ่มจำนวนโหนดที่มีการคำนวณ หรือชั้นแอบแฝงให้กับโครงข่าย ตัวอย่างของโครงข่ายแบบหลายชั้น เช่น การแพร่ย้อนกลับ (Back propagation) เซลฟออร์แกนไนซิงแมปส์ (Self-organizing maps)



ภาพที่ 2-6 โครงข่ายแบบหลายชั้น

การปรับค่าน้ำหนัก (Adjusting weight) การประมวลผลของข่ายงานประสาทเทียมยอมให้มีกระบวนการปรับค่าน้ำหนักในระหว่างการเรียนรู้ของระบบ เพื่อให้ข่ายงานสามารถเรียนรู้พฤติกรรมของข้อมูลใช้ฝึกสอน (Training data) จนบรรลุวัตถุประสงค์ มี 3 แนวทาง

- การฝึกสอนแบบมีผู้สอน (Supervised Training)
- การฝึกสอนแบบไม่มีผู้สอน (Unsupervised Training)
- การฝึกสอนแบบมีการสนับสนุน (Reinforcement Training)

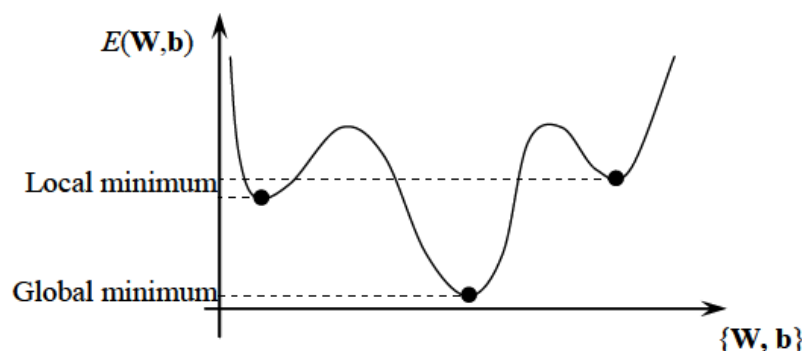
ฟังก์ชันกระตุ้น (Activation Function) การทำงานของโครงข่ายประสาทเทียมจะมีฟังก์ชันสำหรับการแปลงผลลัพธ์ที่ได้ให้อยู่ในรูปของกลุ่มข้อมูลหรือข้อมูลส่งออก

2.5 อัลกอริทึมการเรียนรู้ของโครงข่ายประสาทเทียม

Hebbian Learning Rule เป็นกฎการเรียนรู้ที่เก่าแก่ที่สุดและใช้งานอย่างแพร่หลายจนถึงปัจจุบันนี้ มีจุดประสงค์ในการใช้เหมือนกับสหสัมพันธ์ (Correlation) อาจจะเรียกว่าการเรียนรู้แบบสหสัมพันธ์ก็ได้ ซึ่งมีกฎเกณฑ์ที่สำคัญ ดังนี้ กรณีที่ค่าอินพุตและเอาต์พุต ซึ่งมีค่าที่เป็นไปในทางเดียวกัน มีผลให้ค่าน้ำหนักตรงการเชื่อมต่อระหว่างโหนดถูกเพิ่มขึ้น ทำให้การส่งเสริมการกระตุ้นนิวรอนให้เพิ่มมากขึ้น กรณีที่ค่าอินพุตและเอาต์พุต ไม่เป็นไปในทางเดียวกัน มีผลให้ค่าน้ำหนักตรงการเชื่อมต่อระหว่างโหนดถูกลดลง ทำให้ลดทอนการกระตุ้นนิวรอนลง จากกฎเกณฑ์ 2 ข้อดังกล่าวสามารถสร้างเป็นสมการทั่วไปของค่าเปลี่ยนแปลงค่าน้ำหนักอย่างง่าย ๆ ได้ดังนี้ $\Delta w = x_i y_i$ เมื่อ x_i หมายถึงค่าอินพุต y_i หมายถึงค่าเอาต์พุต Δw หมายถึงค่าเปลี่ยนแปลงค่าน้ำหนัก

Delta Learning Rule (Widrow-Hoff Rule or Least Mean Square (LMS)) การเรียนรู้แบบค่าผิดพลาดกำลังสองเฉลี่ยน้อยที่สุด ในโครงข่ายประสาทเทียมแบบเปอร์เซ็ปตรอนใช้ฟังก์ชันถ่ายโอนจำกัดเชิง ซึ่งเป็นชนิดสองกลุ่ม ($t, a \in \{0,1\}$ หรือ $t, a \in \{-1,1\}$) ถ้าหากต้องการเอาต์พุตเป็นสัญญาณต่อเนื่อง ทำได้โดยเปลี่ยนฟังก์ชันถ่ายโอนเป็นชนิดเชิงเส้น และใช้การเรียนรู้แบบค่าผิดพลาดกำลังสองเฉลี่ยน้อยที่สุด (Least Mean Square Error (LMS) Learning) ในการฝึกฝนโครงข่ายประสาทเทียมแบบเชิงเส้นโครงข่ายประสาทเทียมแบบเชิงเส้น

ตัวชี้การทำให้บรรลุผลสำเร็จของการเรียนรู้ด้วยค่าผิดพลาดกำลังสองเฉลี่ย โครงข่ายประสาทเทียมแบบเชิงเส้นมีคู่อินพุตและเป้าหมายและเมทริกซ์น้ำหนัก W และไบแอส b เมื่อป้อนอินพุตเข้าโครงข่าย สามารถคำนวณเอาต์พุตได้ตามสมการ ค่าผิดพลาดของโครงข่ายเปรียบเทียบกับค่าเอาต์พุตของโครงข่าย และค่าเป้าหมาย สามารถคำนวณได้จาก อัลกอริทึมค่าผิดพลาดกำลังสองเฉลี่ยน้อยที่สุด (LMS) จะทำการปรับค่าน้ำหนักและไบแอส ของโครงข่ายเชิงเส้นตรงราบเท่าที่ค่าผิดพลาดกำลังสองเฉลี่ยยังถูกลดค่าลง



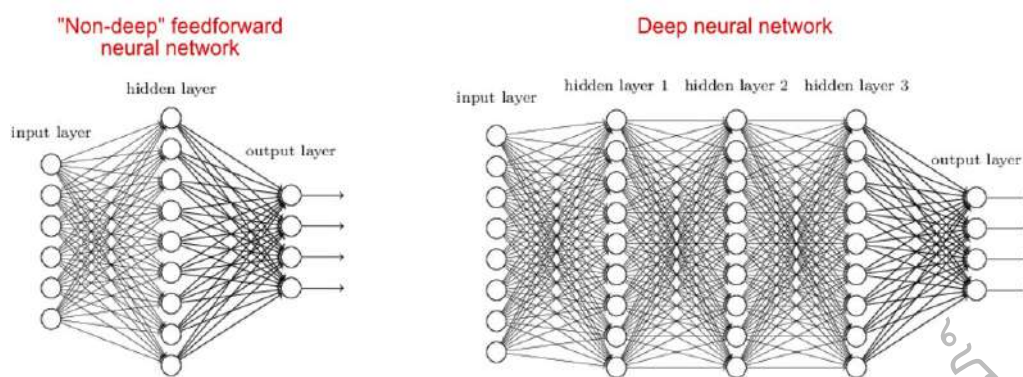
ภาพที่ 2-7 วิธีค้นหาในทิศทางที่ลาดชันที่สุด (Steepest Descent Method)

สมการนี้เป็นวิธีค้นหาในทิศทางที่ลาดชันที่สุด เป็นวิธีการหาค่าต่ำสุดหรือสูงสุดแบบหนึ่งที่เราจะรู้จักกันดี ในการค้นหาค่าผิดพลาดต่ำสุดด้วยวิธีทิศทางที่ลาดชัน ที่สุดนั้น ทุก ๆ รอบถัดไปของการค้น ค่าผิดพลาดจะต้องลดลงเรื่อย ๆ ในวิธีการเรียนรู้ด้วยการหาค่าผิดพลาดให้ถูกต้อง (Error Correction Learning) ในแต่ละรอบของวิธีลาดชันที่สุดจะคำนวณในทิศทางลาดลง ซึ่งตรงกันข้ามกับค่าอนุพันธ์ที่ค่าน้ำหนัก w_m นั่นคือ พิจารณาค่าผิดพลาดของข้อมูลแต่ละเรคคอร์ดค่าอนุพันธ์ของ $E(x)$ เป็นค่าความลาดชันอย่างประมาณเทียบกับค่าน้ำหนักและเทียบกับไบแอสจะได้

Back-propagation เป็นอัลกอริทึมที่ใช้ในการเรียนรู้ของเครือข่ายประสาทวิธีหนึ่งที่นิยมใช้ใน Multilayer perceptron เพื่อปรับค่าน้ำหนักในเส้นเชื่อมต่อระหว่างโหนดให้เหมาะสม โดยการปรับค่านี้จะขึ้นกับความแตกต่างของค่าเอาต์พุตที่คำนวณได้กับค่าเอาต์พุตที่ต้องการ

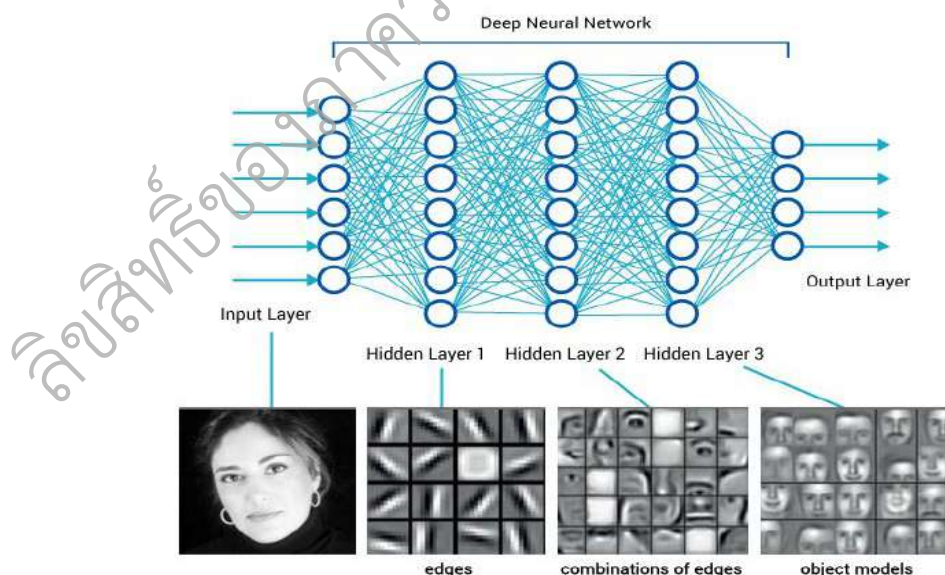
2.6 โครงข่ายประสาทเทียมเชิงลึก (Deep Neural Network)

ความท้าทายอย่างหนึ่งในการพัฒนาระบบปัญญาประดิษฐ์ (AI : Artificial Intelligence) คือ การพัฒนาให้เครื่องจักรสามารถเรียนรู้และทำนายหรือสร้างองค์ความรู้ได้ ด้วยเหตุนี้จึงเป็นที่มาของสาขา Machine Learning แต่เทคนิคที่จะสอนให้เครื่องคอมพิวเตอร์สามารถเรียนรู้ได้นั้นก็มีเทคนิคมากมาย เช่น Supervised Learning, Unsupervised Learning, Reinforcement Learning หรือ Artificial Neuron Networks เป็นต้น ซึ่งจริง ๆ แล้ว Deep Learning ก็คือ ANN : Artificial Neuron Networks นั่นเอง โดย Deep Learning และ ANN เป็นอัลกอริทึมที่ถูกสร้างขึ้นมาเพื่อการเรียนรู้ของเครื่อง แต่ความแตกต่างระหว่าง Deep Learning กับ ANN ก็คือระดับ Hidden layer ที่ใน Deep Learning มี Hidden layer มากกว่าใน ANN



ภาพที่ 2-8 โครงข่ายประสาทเทียมเชิงลึก (Deep Neural Network)

โครงข่ายประสาทเทียม (Artificial Neural Networks) นั้นอาศัยแนวคิดและเทคนิคจากการทำงานของระบบโครงข่ายประสาทในระบบประสาทของมนุษย์ โดยจำลองการทำงานเหมือนกับกลุ่มเซลล์ประสาทที่เชื่อมโยงกันเป็นระบบประสาทที่สามารถรับรู้หลายๆ สิ่งในเวลาเดียวกัน ด้วยการประมวลผลแบบขนาน (Parallel Network) ทำให้ระบบสามารถตัดสินใจได้ใกล้เคียงกับมนุษย์ในการที่เครื่องจะสามารถเข้าใจสิ่งต่าง ๆ ได้ก็จำเป็นที่จะต้องมีความรู้ (Knowledge) เสียก่อน [Input layer] จากนั้นก็จะประเมินชุดข้อมูล [Hidden layer] และนำเสนอหรือแทนองค์ความรู้ [Output layer]

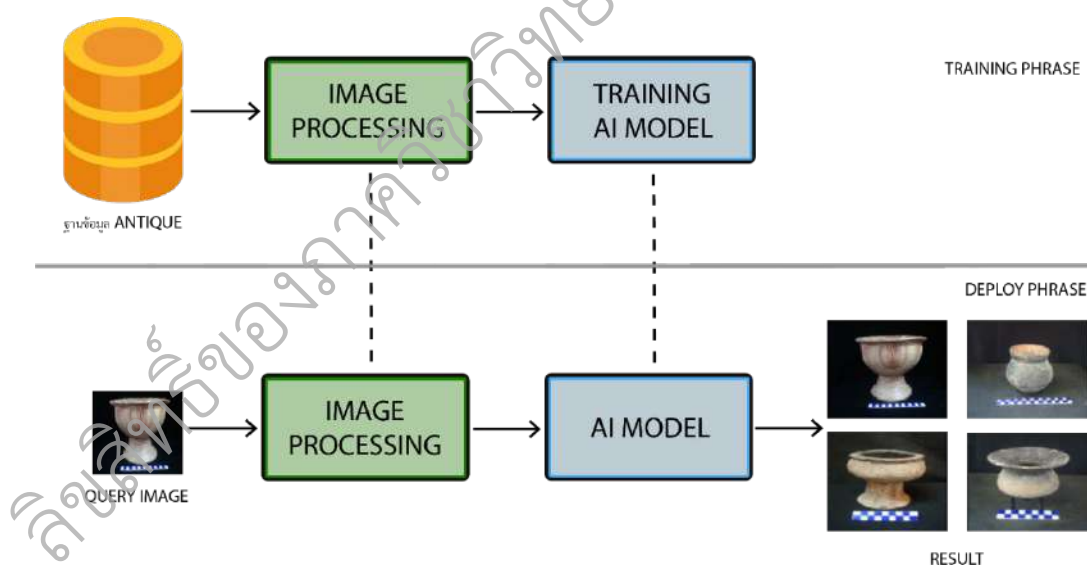


ภาพที่ 2-9 Deep Learning สำหรับการเรียนรู้จดจำใบหน้า

Deep Neural Network สำหรับการเรียนรู้จดจำใบหน้า Deep Learning ถูกนำมาประยุกต์ใช้ในงานต่าง ๆ มากมาย เช่น การแยกแยะใบหน้าแต่ละคน ตัวอย่างเช่นในการติดแท็กรูปภาพเพื่อนใน Facebook หรือการแยกวัตถุที่ไม่ใช่คน หรือใช้เป็นส่วนหนึ่งในระบบรถยนต์ไร้คนขับ เป็นต้น

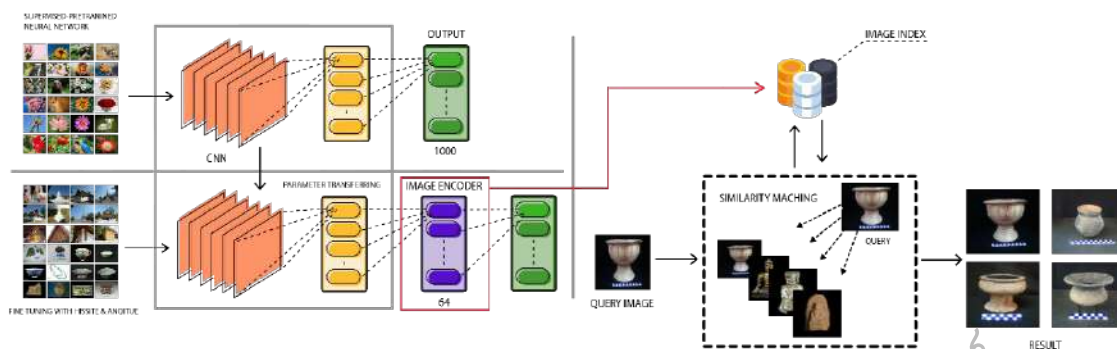
2.7 กระบวนการฝึกฝนโครงข่ายประสาทเทียม

ภาพที่ 2-10 แสดงขั้นตอนการสร้างโมเดล AI เพื่อใช้ในการสืบค้นภาพ โดยเริ่มจากการนำชุดข้อมูลภาพในฐานข้อมูล Antique เข้าสู่กระบวนการประมวลผลภาพเพื่อปรับปรุงคุณภาพของภาพที่ใช้ในการเรียนรู้โดยโครงข่ายประสาทเทียม หลังจากขั้นตอนการประมวลผลภาพ ภาพแต่ละภาพจะถูกปรับขนาดให้เท่ากันคือ 150×150 พิกเซล ซึ่งเป็นขนาดที่เหมาะสมในการเรียนรู้ของโครงข่ายประสาทเทียม โดยการกำหนดวัตถุประสงค์การเรียนรู้ (Objective Function) ซึ่งประกอบไปด้วย การเรียนรู้ภาพเพื่อทำนายสี วัสดุ และศิลปะ ของภาพโบราณวัตถุให้สอดคล้องกับฐานข้อมูลที่จัดเก็บ ซึ่งในฐานข้อมูลจะแสดงรายการวัสดุและศิลปะซึ่งจะถูกกำหนดให้เป็นวัตถุประสงค์ในการสอนโครงข่ายประสาทเทียมเพื่อแยกแยะภาพ



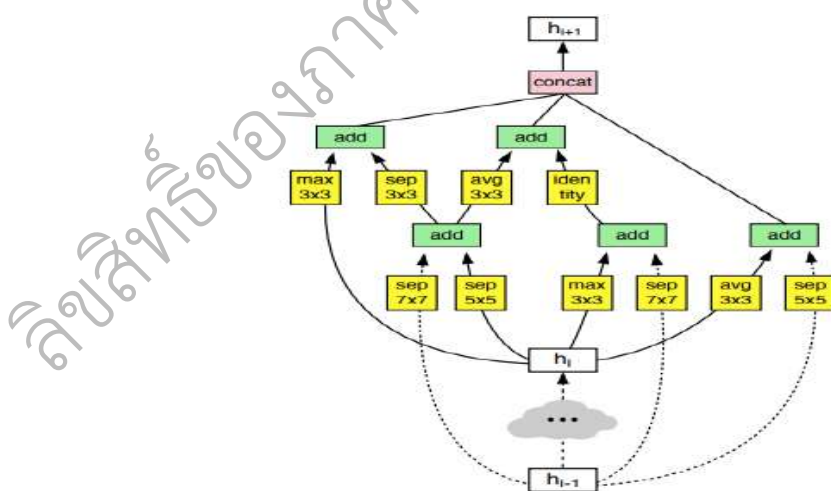
ภาพที่ 2-10 แสดงกระบวนการสร้างโมเดล AI เพื่อใช้ในการสืบค้นข้อมูลโบราณวัตถุ

ภาพที่ 2-11 แสดงขั้นตอนการสร้างโมเดล AI โดยจะเริ่มจากโครงข่ายประสาทเทียมซึ่งถูกฝึกฝนมาแล้วด้วยข้อมูลภาพจำนวนมาก ในโครงการนี้เลือกใช้โครงข่ายประสาทเทียม efficientnet ซึ่งเป็นโครงข่ายประสาทเทียม แบบคอนโวลูชัน (Convolution Neural Network)

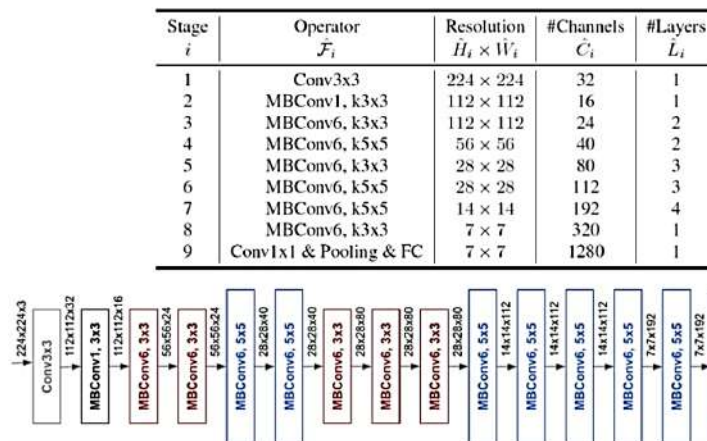


ภาพที่ 2-11 ขั้นตอนการสร้างโมเดล AI

Convolution Neural Network ถูกออกแบบโดย Google มีข้อดีคือให้ความรวดเร็วและความแม่นยำในการวิเคราะห์ภาพสูง แนวคิดในการออกแบบ efficient net คือเทคนิคการผสมผสานระหว่างโมเดลโครงข่ายประสาทเทียมแบบลึก (Deep Neural Network) ซึ่งมีข้อดีคือให้ความแม่นยำในการจำแนกสูง ขณะที่การทำงานค่อนข้างช้าและมีโอกาสเกิด Overfit ได้ง่ายเนื่องจากปัญหาของ Gradient Vanishing และโครงข่ายประสาทเทียมแบบกว้าง (Wide Neural Network) ที่มีความเร็วในการทำงานสูงแต่มีข้อจำกัดในเรื่องความแม่นยำ โดย efficientnet จะวิเคราะห์ชุดข้อมูลฝึกฝนเพื่อหาจำนวนพารามิเตอร์ที่มีความเหมาะสมทั้งในแง่ของความเร็วและความแม่นยำ ดังภาพที่ 2-12 และภาพที่ 2-13 ซึ่งเป็นโมเดลแรกที่ถูกพัฒนาขึ้นโดย Google

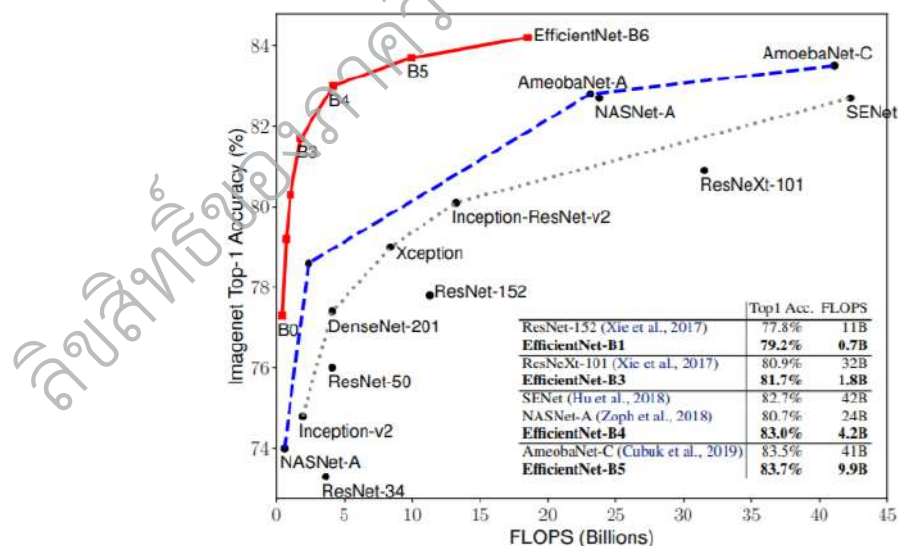


ภาพที่ 2-12 โครงสร้างทั่วไปของโครงข่ายประสาทเทียมแบบ efficientnet



ภาพที่ 2-13 แสดงโครงสร้างและจำนวนพารามิเตอร์ของโมเดล B0

ปัจจุบันโมเดล efficientnet ถูกพัฒนาขึ้นมาหลายโมเดลให้ชื่อว่า B0 B1 B2 ... B6 แต่ละโมเดลแตกต่างกันที่จำนวนพารามิเตอร์ ความกว้างและความลึกของโครงข่ายประสาทเทียม ปัจจุบันโครงข่ายประสาทเทียม efficientnet ถือเป็นโครงข่ายประสาทเทียมที่ทำงานได้รวดเร็วที่สุดและแม่นยำที่สุดเมื่อเปรียบเทียบกับโครงข่ายประสาทเทียมคอนโวลูชันแบบอื่น ดังภาพที่ 2-14 และตารางที่ 2-1 แสดงผลการเปรียบเทียบความเร็วและความแม่นยำของ efficientnet ในแต่ละโมเดล



ภาพที่ 2-14 แสดงผลการเปรียบเทียบความเร็วและความแม่นยำของโมเดล efficientnet B0 ถึง B6 และโครงข่ายประสาทเทียมคอนโวลูชันแบบอื่นบนฐานข้อมูลภาพ ImageNet

ตารางที่ 2-1 แสดงจำนวนพารามิเตอร์และความแม่นยำบน Top-1 และ Top-5

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPS	Ratio-to-EfficientNet
EfficientNet-B0	76.3%	93.2%	5.3M	1x	0.39B	1x
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	78.8%	94.4%	7.8M	1x	0.70B	1x
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	79.8%	94.9%	9.2M	1x	1.0B	1x
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	81.1%	95.5%	12M	1x	1.8B	1x
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	82.6%	96.3%	19M	1x	4.2B	1x
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	83.3%	96.7%	30M	1x	9.9B	1x
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	84.0%	96.9%	43M	1x	19B	1x
EfficientNet-B7	84.4%	97.1%	66M	1x	37B	1x
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

ที่มา : <https://www.profillic.com/paper/arxiv:1905.11946>

เนื่องจากโมเดล efficientnet ถูกฝึกฝนมาให้ความแม่นยำในการจำแนกภาพทั่วไป (1000 Label) เช่น สุนัข แมว เครื่องบิน รถยนต์ เป็นต้น จึงเป็นไปได้ที่จะใช้โมเดล efficientnet โดยตรงเพื่อแยกแยะภาพโบราณวัตถุ ดังนั้นก่อนที่จะนำไปใช้งานจะต้องมีการฝึกฝนโครงข่ายประสาทเทียมเพิ่มเติม เริ่มต้นจะต้องมีการกำหนดวัตถุประสงค์ในการฝึกฝนโครงข่ายประสาทเทียม efficientnet โดยจะทำการปรับเปลี่ยนจำนวนเวอร์เทกซ์ในชั้นผลลัพธ์ (Output layer) อย่างไรก็ดีเนื่องจากจำนวนของโบราณวัตถุมีมาก และภาพที่ใช้ในการฝึกฝนของแต่ละโบราณวัตถุมีน้อยเกินไป จึงเป็นไปได้ยากที่จะฝึกฝนโครงข่ายประสาทเทียมให้มีความแม่นยำในการทำนายชื่อโบราณวัตถุโดยตรง

เพื่อที่จะแก้ปัญหาดังกล่าวจึงต้องทำการดัดแปลงโดยแทนที่ฝึกฝนให้โครงข่ายประสาทเทียมทำนายชื่อโบราณวัตถุโดยตรง ก็จะทำให้โครงข่ายประสาทเทียมทำนาย feature ของแต่ละภาพแทน จากนั้นใช้ feature ที่โครงข่ายประสาทเทียมทำนายได้ เพื่อถอดรหัสว่าเป็นโบราณวัตถุใดในฐานข้อมูลกรมศิลปากร นั่นเอง

ข้อดีของเทคนิคเหล่านี้คือ สามารถเพิ่มรายการโบราณวัตถุเข้าไปได้เรื่อย ๆ อย่างไม่จำกัด ยิ่งไปกว่านั้นการฝึกฝนโมเดลก็สามารถทำได้รวดเร็วกว่าการเรียนรู้โดยตรง เริ่มต้นจะต้องมีการกำหนดวัตถุประสงค์การฝึกฝน โดยการใช้ข้อมูลรายการวัตถุและรายการประเภทของศิลปะของโบราณวัตถุ ซึ่งทางกรมศิลปากรได้มีการกำหนดไว้ในฐานข้อมูลแล้ว ดังตารางที่ 2-2 และตารางที่ 2-3








ตารางที่ 2-2 รายการประเภทวัสดุจำแนกตามฐานข้อมูลกรมศิลปากร

รายการที่	ประเภทของวัสดุ	จำนวนภาพ	ตัวอย่างรูปภาพ
1	หิน	2,741	
2	ดินเผา	321	
3	ภาพพิมพ์บนไม้	464	
4	เหล็ก	56	
5	ไม้	331	
6	ดินเผาเคลือบ	4,171	
7	ภาพพิมพ์หิน	658	

ตารางที่ 2-2 (ต่อ) รายการประเภทวัสดุจำแนกตามฐานข้อมูลกรมศิลปากร

รายการที่	ประเภทของวัสดุ	จำนวนภาพ	ตัวอย่างรูปภาพ
8	โลหะผสมของเหล็ก	1,331	
9	กระดาษ	23	
10	จิตรกรรมบนพื้นผ้าใบ	4,413	
11	ภาพพิมพ์โลหะ	10,213	
12	ทองแดง	4,680	
13	ผ้า	55	
14	จิตรกรรมสีน้ำ	183	

ตารางที่ 2-2 (ต่อ) รายการประเภทวัสดุจำแนกตามฐานข้อมูลกรมศิลปากร

รายการที่	ประเภทของวัสดุ	จำนวนภาพ	ตัวอย่างรูปภาพ
15	สำริด	413	
16	ดีบุก	433	
17	งา	10	
18	จิตรกรรมสีฝุ่น	223	
19	หนังสัตว์	25	
20	โลหะผสมทองแดง	85	
21	กระดูกสัตว์	50	







ตารางที่ 2-2 (ต่อ) รายการประเภทวัสดุจำแนกตามฐานข้อมูลกรมศิลปากร

รายการที่	ประเภทของวัสดุ	จำนวนภาพ	ตัวอย่างรูปภาพ
22	จิตรกรรมบนกระดาน	3	
23	หินทราย	48	
24	ตะกั่ว	1,543	
25	โลหะ	2,998	
26	จิตรกรรมบนไม้	168	
27	หินชนวน	387	
28	จีน	8	
29	ปูนปั้น	21	

ตารางที่ 2-2 (ต่อ) รายการประเภทวัสดุจำแนกตามฐานข้อมูลกรมศิลปากร

รายการที่	ประเภทของวัสดุ	จำนวนภาพ	ตัวอย่างรูปภาพ
30	จิตรกรรมสีน้ำมันบนพื้นผ้าใบ	722	
31	หินอ่อน	607	
32	เงิน	391	
33	ปูนปลาสเตอร์	2,114	
34	จิตรกรรมสีน้ำมันบนไม้	148	
35	ทองคำ	63	
36	จิตรกรรมสีอะคริลิก	111	
37	อื่น ๆ	283	

ตารางที่ 2-3 รายการประเภทศิลปะจำแนกตามฐานข้อมูลกรมศิลปากร

รายการที่	ประเภทของศิลปะ	จำนวนภาพ	ตัวอย่างรูปภาพ
1	ก่อนประวัติศาสตร์	39,929	
2	ทวารวดี	13,848	
3	ศรีวิชัย	958	
4	หริภุญไชย	686	
5	ลพบุรี	14,162	
6	ล้านนา	13,279	

ตารางที่ 2-3 (ต่อ) รายการประเภทศิลปะจำแนกตามฐานข้อมูลกรมศิลปากร

รายการที่	ประเภทของศิลปะ	จำนวนภาพ	ตัวอย่างรูปภาพ
7	สุโขทัย	30,705	
8	อุททอง	385	
9	ล้านช้าง	1349	
10	อยุธยา	37,123	
11	ธนบุรี	39	
12	รัตนโกสินทร์	44,395	
13	ต่างประเทศ	14,452	

ตารางที่ 2-3 (ต่อ) รายการประเภทศิลปะจำแนกตามฐานข้อมูลกรมศิลปากร

รายการที่	ประเภทของศิลปะ	จำนวนภาพ	ตัวอย่างรูปภาพ
14	ระหว่างการศึกษา	2,996	
15	อื่น ๆ	598	

ในการฝึกฝนให้โมเดล AI มีความสามารถในการทำนายวัสดุและประเภทของศิลปะจากภาพถ่าย จะเริ่มจากการปรับจำนวนค่าข้อมูลในชั้นเอาต์พุต (Output layer) ซึ่งปกติแล้ว จำนวนค่าข้อมูลในชั้นเอาต์พุตของโครงข่ายประสาทเทียม efficientnet ซึ่งถูกฝึกฝนบนชุดข้อมูลภาพ ImageNet จะมีค่าเท่ากับ 1,000 เท่ากับจำนวนประเภทวัตถุที่ชุดข้อมูล ImageNet ได้ทำการแบ่งกลุ่มไว้ โดยจะแทนที่ด้วยจำนวนค่าข้อมูลเท่ากับ 52 ซึ่งแบ่งออกเป็นจำนวนรายการวัสดุ 37 รายการ และจำนวนรายการประเภทศิลปะวัตถุ 15 รายการ โดยโครงข่าย

ประสาทเทียมจะถูกฝึกฝนให้สามารถทำนายค่าข้อมูลเหล่านี้จากภาพถ่ายที่ป้อนเข้ามา อย่างไรก็ตามเนื่องจากภาพถ่ายโบราณ 1 ภาพ อาจจะประกอบไปด้วยวัสดุและประเภทของศิลปะมากกว่า 1 รายการ ดังนั้นโครงข่ายประสาทเทียม efficientnet จึงต้องมีความสามารถในการทำนายค่าเหล่านี้ได้พร้อมกันมากกว่า 1 รายการ เรียกปัญหานี้ว่า Multi-label Classification ซึ่งจะมีวิธีการฝึกฝนโครงข่ายประสาทเทียมที่แตกต่างจากการฝึกฝนทั่วไป

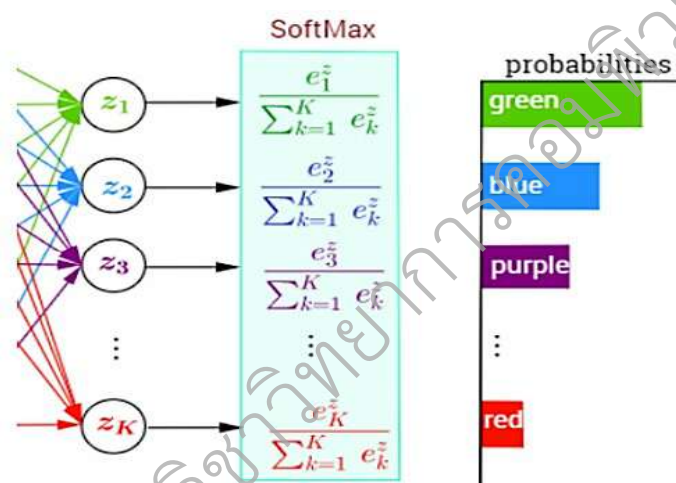
กล่าวคือในการฝึกฝนโครงข่ายประสาทเทียมทั่วไป ซึ่งจะทำนายค่าข้อมูลได้เพียง 1 ค่าในเวลาเดียวกันจะใช้ฟังก์ชัน SoftMax เพื่อคำนวณค่าความน่าจะเป็นที่มากที่สุดที่ชั้นเอาต์พุตเป็นคำตอบของโครงข่ายประสาทเทียม โดยมีนิยามดังนี้

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$$

โดยที่ $\sigma(\mathbf{z})_i$ หมายถึงค่าผลลัพธ์ของ SoftMax ที่ชั้นเอาต์พุตลำดับที่ i และ

K หมายถึงจำนวนค่าเอาต์พุตทั้งหมดของโครงข่ายประสาทเทียม

จะเห็นว่าฟังก์ชัน SoftMax จะทำการนอร์มัลไลซ์ (Normalize) ค่าเอาต์พุตแต่ละค่าที่โครงข่ายประสาทเทียมคำนวณออกมา โดยจะนำค่าเอาต์พุตแต่ละตัวหารด้วยผลรวมของค่าเอาต์พุตทั้งหมดที่เป็นไปได้ ดังนั้นผลลัพธ์ที่ได้จะถูกปรับให้อยู่ระหว่าง 0 ถึง 1 ซึ่งก็จะหมายถึงค่าความน่าจะเป็นนั่นเอง ค่าผลลัพธ์ใดที่มีค่าสูงที่สุดก็就会被เลือกเป็นคำตอบนั่นเอง ภาพที่ 2-15 แสดงการทำงานของฟังก์ชัน SoftMax เพื่อการทำนายเอาต์พุตของโครงข่ายประสาทเทียม จากภาพจะเห็นว่าหลังจากที่ค่าเอาต์พุตผ่านฟังก์ชัน SoftMax ค่าความน่าจะเป็นของ green จะมีค่าสูงที่สุดและจะถูกเลือกเป็นคำตอบของโครงข่ายประสาทเทียม



ภาพที่ 2-15 แสดงการทำงานของฟังก์ชัน SoftMax

สำหรับปัญหาของ Multi-label Classification จะมีการแทนที่ฟังก์ชัน SoftMax ด้วยฟังก์ชัน Binary Cross-entropy

$$Cost(Y, Z) = \frac{1}{m} \sum \max(Z, 0) - ZY + \log(1 + e^{-|Z|})$$

โดยที่ Y, Z หมายถึงเอาต์พุตจริง (Actual output) และเอาต์พุตจากการทำนาย (Predict Output) ของโครงข่ายประสาทเทียม โดย m หมายถึงจำนวนข้อมูลฝึกฝนของโครงข่ายประสาทเทียม นั่นเอง จะเห็นว่า Binary Cross-entropy จะอนุญาตให้โครงข่ายประสาทเทียมสามารถทำนายเอาต์พุตได้มากกว่า 1 ค่าพร้อมกัน โดยจะพยายามฝึกฝนเพื่อให้ค่า Cost ระหว่างเอาต์พุตจริงแต่ละค่าและเอาต์พุตจากการทำนายมีค่าต่ำที่สุดเท่าที่จะเป็นไปได้

นอกจากการทำนายค่าเอาต์พุต ระบบ Visual Search จะต้องมีความสามารถในการดึงภาพโบราณวัตถุที่เกี่ยวข้องเพื่อนำมาแสดงบนหน้าเว็บเพจได้อีกด้วย โดยในการทำงานจะมีการเพิ่มชั้นข้อมูลเข้าไป 1 ชั้น เพื่อทำหน้าที่ในการเข้ารหัสภาพ (Image encoding) ในรูปแบบของเวกเตอร์ (Vector) โดยขนาดของเวกเตอร์จะขึ้นอยู่กับชุดข้อมูลภาพที่นำมาฝึกฝน หากกำหนดให้มีขนาดใหญ่เกินไปจะส่งผลให้การทำงานช้าลง และหากกำหนดให้มีขนาดเล็กเกินไปก็จะส่งผลให้การดึงข้อมูลภาพที่เกี่ยวข้องไม่แม่นยำเท่าที่ควร จากการศึกษาทดลองชุดข้อมูลดังกล่าวพบว่าขนาดของเวกเตอร์สำหรับการเข้ารหัสภาพที่เหมาะสมที่สุดเท่ากับ 64 โดยเมื่อทำการออกแบบโครงสร้างของโครงข่ายประสาทเทียมเรียบร้อยแล้ว ก็จะมีการฝึกฝนโครงข่ายประสาทเทียมเพื่อให้สามารถสร้างโมเดล AI

```

Downloading data from https://github.com/fchollet/deep-learning-models/releases/download/v0.1/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58892288/58889256 [=====] - 341s 6us/step
Train on 179963 samples, validate on 44991 samples
Epoch 1/30
179963/179963 [=====] - 19527s 109ms/step - loss: 0.0887 - fbeta: 0.4518 - val_loss: 0.0
702 - val_fbata: 0.5834
Epoch 2/30
179963/179963 [=====] - 18109s 101ms/step - loss: 0.0626 - fbeta: 0.6397 - val_loss: 0.0
595 - val_fbata: 0.6747
Epoch 3/30
179963/179963 [=====] - 18476s 103ms/step - loss: 0.0522 - fbeta: 0.7130 - val_loss: 0.0
547 - val_fbata: 0.7019
Epoch 4/30
179963/179963 [=====] - 18889s 105ms/step - loss: 0.0450 - fbeta: 0.7609 - val_loss: 0.0
533 - val_fbata: 0.7243
Epoch 5/30
179963/179963 [=====] - 19025s 106ms/step - loss: 0.0388 - fbeta: 0.7991 - val_loss: 0.0
511 - val_fbata: 0.7562
Epoch 6/30
179963/179963 [=====] - 18954s 105ms/step - loss: 0.0332 - fbeta: 0.8322 - val_loss: 0.0
504 - val_fbata: 0.7688
Epoch 7/30
179963/179963 [=====] - 19176s 107ms/step - loss: 0.0284 - fbeta: 0.8600 - val_loss: 0.0
507 - val_fbata: 0.7770
Epoch 8/30
179963/179963 [=====] - 19083s 106ms/step - loss: 0.0239 - fbeta: 0.8836 - val_loss: 0.0
537 - val_fbata: 0.7831
Epoch 9/30
179963/179963 [=====] - 19179s 107ms/step - loss: 0.0202 - fbeta: 0.9039 - val_loss: 0.0
573 - val_fbata: 0.7818
Epoch 10/30
179963/179963 [=====] - 18871s 105ms/step - loss: 0.0171 - fbeta: 0.9198 - val_loss: 0.0
582 - val_fbata: 0.7849
Epoch 11/30
179963/179963 [=====] - 19083s 106ms/step - loss: 0.0146 - fbeta: 0.9331 - val_loss: 0.0
623 - val_fbata: 0.7984
Epoch 12/30
179963/179963 [=====] - 19029s 106ms/step - loss: 0.0125 - fbeta: 0.9431 - val_loss: 0.0
637 - val_fbata: 0.8014
Epoch 13/30
179963/179963 [=====] - 18849s 105ms/step - loss: 0.0107 - fbeta: 0.9521 - val_loss: 0.0
674 - val_fbata: 0.7992
Epoch 14/30
179963/179963 [=====] - 19020s 106ms/step - loss: 0.0094 - fbeta: 0.9582 - val_loss: 0.0
702 - val_fbata: 0.8050
Epoch 15/30
179963/179963 [=====] - 19395s 108ms/step - loss: 0.0082 - fbeta: 0.9645 - val_loss: 0.0
753 - val_fbata: 0.8052
Epoch 16/30
179963/179963 [=====] - 19474s 108ms/step - loss: 0.0072 - fbeta: 0.9693 - val_loss: 0.0
786 - val_fbata: 0.8065
Epoch 17/30
179963/179963 [=====] - 19352s 108ms/step - loss: 0.0064 - fbeta: 0.9727 - val_loss: 0.0
799 - val_fbata: 0.7997
Epoch 18/30
179963/179963 [=====] - 19201s 107ms/step - loss: 0.0059 - fbeta: 0.9752 - val_loss: 0.0
866 - val_fbata: 0.8052
Epoch 19/30
179963/179963 [=====] - 19248s 107ms/step - loss: 0.0055 - fbeta: 0.9771 - val_loss: 0.0
833 - val_fbata: 0.8102
Epoch 20/30
179963/179963 [=====] - 19100s 106ms/step - loss: 0.0048 - fbeta: 0.9802 - val_loss: 0.0
874 - val_fbata: 0.8090
Epoch 21/30
179963/179963 [=====] - 19436s 108ms/step - loss: 0.0045 - fbeta: 0.9818 - val_loss: 0.0
911 - val_fbata: 0.8108
Epoch 22/30
179963/179963 [=====] - 19092s 106ms/step - loss: 0.0039 - fbeta: 0.9843 - val_loss: 0.0
921 - val_fbata: 0.8074
Epoch 23/30
179963/179963 [=====] - 19111s 106ms/step - loss: 0.0037 - fbeta: 0.9851 - val_loss: 0.0
983 - val_fbata: 0.8114
Epoch 24/30
179963/179963 [=====] - 18861s 105ms/step - loss: 0.0034 - fbeta: 0.9864 - val_loss: 0.0
956 - val_fbata: 0.8157
Epoch 25/30
179963/179963 [=====] - 19127s 106ms/step - loss: 0.0029 - fbeta: 0.9887 - val_loss: 0.1
004 - val_fbata: 0.8164
Epoch 26/30
179963/179963 [=====] - 19065s 106ms/step - loss: 0.0030 - fbeta: 0.9882 - val_loss: 0.1
016 - val_fbata: 0.8136
Epoch 27/30
179963/179963 [=====] - 19432s 108ms/step - loss: 0.0029 - fbeta: 0.9889 - val_loss: 0.1
008 - val_fbata: 0.8120
Epoch 28/30
179963/179963 [=====] - 19212s 107ms/step - loss: 0.0026 - fbeta: 0.9901 - val_loss: 0.1
019 - val_fbata: 0.8151
Epoch 29/30
179963/179963 [=====] - 19130s 106ms/step - loss: 0.0023 - fbeta: 0.9910 - val_loss: 0.1
078 - val_fbata: 0.8153
Epoch 30/30
179963/179963 [=====] - 19417s 108ms/step - loss: 0.0022 - fbeta: 0.9914 - val_loss: 0.1
120 - val_fbata: 0.8184
44991/44991 [=====] - 2759s 61ms/step
> loss=0.112, fbata=0.818

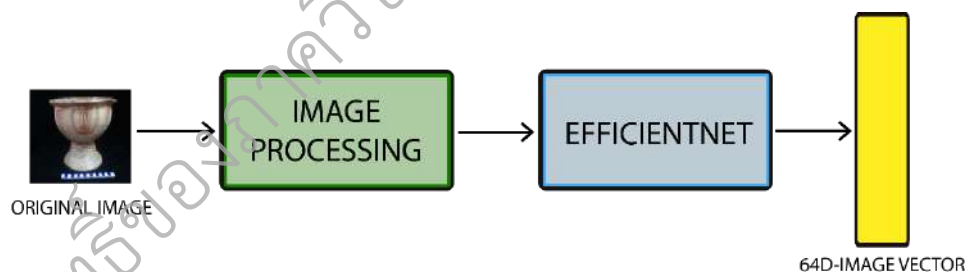
```

ภาพที่ 2-16 หน้าจอการฝึกฝนโครงข่ายประสาทเทียม efficientnet บนเครื่องแม่ข่าย

2.8 การทำดัชนีภาพ (Image Indexing)

การทำดัชนีภาพ คือ การนำเอาลักษณะเฉพาะของภาพ เช่น ฮิสโตแกรมของสีมาสร้างเป็นเวกเตอร์ลักษณะเฉพาะของภาพแต่ละภาพในฐานข้อมูลและจัดเก็บไว้ ลักษณะเฉพาะของภาพแต่ละภาพที่แยกออกมาจะอยู่ในรูปของค่าที่เป็นตัวเลขจำนวน n ค่า (ซึ่งขึ้นอยู่กับเงื่อนไขของแต่ละวิธี) หรือเวกเตอร์ขนาด n มิติ ซึ่งก็คือเวกเตอร์ลักษณะเฉพาะของภาพนั้น ๆ โดยเวกเตอร์ลักษณะเฉพาะเหล่านี้จะนำมาใช้เป็นดัชนีภาพซึ่งแทนด้วยจุดในปริภูมิ n มิติ

หลังจากฝึกฝนโครงข่ายประสาทเทียมเรียบร้อยแล้ว ขั้นตอนถัดไปคือการนำชุดภาพโบราณวัตถุทั้งหมดไปทำดัชนีภาพ (Image Indexing) เพื่อวัตถุประสงค์ในการสืบค้นที่รวดเร็ว เนื่องจากข้อมูลภาพมีขนาดใหญ่จะเป็นการไม่เหมาะสมที่จะนำเอาภาพต้นฉบับไปทำดัชนีภาพโดยตรง สำหรับระบบ Visual Search จะใช้ข้อมูลเวกเตอร์เข้ารหัสภาพที่ได้ทำการดัดแปลงโครงข่ายประสาทเทียมเพิ่มเติมเพื่อนำไปทำดัชนีภาพแทนที่ข้อมูล ภาพต้นฉบับ โดยแต่ละภาพในฐานข้อมูลจะถูกนำไปเข้ารหัสภาพเพื่อให้อยู่ในรูปของเวกเตอร์ความยาว 64 จากนั้นจะทำการคำนวณ Hash Table เพื่อนำข้อมูลภาพที่เข้ารหัสแล้วไปจัดเก็บเพื่อใช้ในการสืบค้น ภาพที่ 22 แสดงตัวอย่างการเข้ารหัสภาพโบราณวัตถุให้อยู่ในรูปของเวกเตอร์ขนาด 64 มิติ เพื่อนำไปสร้างดัชนีภาพ

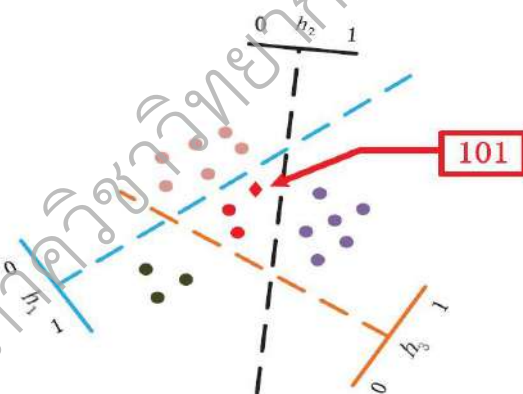


ภาพที่ 2-17 การเข้ารหัสภาพโบราณวัตถุเพื่อการทำดัชนีภาพ

เทคนิคในการสร้างดัชนีปัจจุบันมีมากมายหลายเทคนิค โดยเทคนิคที่ถูกนำมาประยุกต์ใช้ในการสร้างดัชนีเพื่อการสืบค้นภาพจะใช้เทคนิค Locality-Sensitive Hashing (LSH) ซึ่งมีข้อดีในแง่ของความเร็วในการสืบค้นภาพที่มีจำนวนมาก LSH เป็นเทคนิคการทำดัชนีข้อมูลซึ่งอาศัยความใกล้เคียง (Similarity) ของข้อมูลในการสร้างดัชนีเพื่อการสืบค้น มีแนวคิดคือข้อมูลที่มีลักษณะใกล้เคียงกันมักจะถูกมีค่าข้อมูลไม่แตกต่างกันมาก

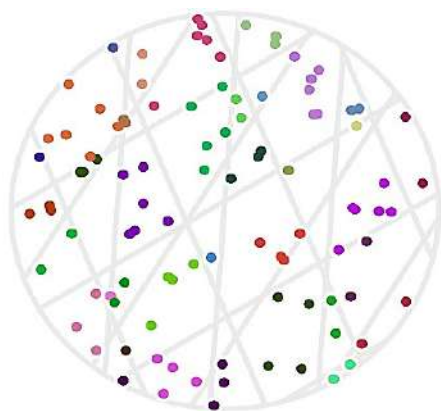
สามารถแบ่งกลุ่มข้อมูลตามข้อมูลที่มีลักษณะคล้ายคลึงกันได้ก็จะช่วยจำกัดพื้นที่และจำนวนในการสืบค้นข้อมูล ในการแบ่งกลุ่มข้อมูลจะมีการกำหนด Binary Code เพื่อเข้ารหัสข้อมูลที่มีความใกล้เคียงกัน ขนาดของ Binary Code จะขึ้นอยู่กับความซับซ้อนของข้อมูลเป็นหลัก แต่ก็มีหลักคิดคือหากกำหนด Binary Code ให้มีความยาวมาก ก็จะมีแบ่งกลุ่มข้อมูลมากจะช่วยสืบค้นได้แม่นยำมากขึ้น ขณะที่หากกำหนด Binary Code น้อยก็จะมีความเร็วในการสืบค้นนั่นเอง

ภาพที่ 2-18 แสดงตัวอย่างการสร้าง Binary Code สำหรับการแบ่งกลุ่มข้อมูลโดยกำหนดความยาวของ Binary Code เท่ากับ 3 ซึ่งได้แก่ $h_1h_2h_3$ โดยที่แต่ละ h_i จะมีค่าเป็นไปได้อีกคือ 0 และ 1 นั่นเอง ในการสร้าง Binary Code จะใช้เทคนิค Random Sampling เพื่อทำการสุ่ม Binary Code เพื่อลองแบ่งกลุ่มข้อมูลจากนั้นอัลกอริทึมจะเริ่มทำการปรับ Binary Code ตามโครงสร้างของข้อมูลที่น่ามาทำดัชนี โดยข้อมูลที่ค่าแตกต่างกันไม่มากจะต้องถูกจัดกลุ่มไว้ในกลุ่มเดียวกัน ซึ่งก็หมายถึงการได้รับ Binary Code เดียวกัน ตัวอย่างในภาพที่ 2-18 คือ ข้อมูลในกลุ่มสีแดงทั้งหมดจะถูกเข้ารหัสด้วย Binary Code เดียวกันคือ 101 ตามลำดับ



ภาพที่ 2-18 แสดงตัวอย่างการสร้าง Binary Code เพื่อแบ่งกลุ่มข้อมูล

ดังนั้นหากข้อมูลอินพุตที่รับเข้ามาถูกเข้ารหัสด้วย Code 101 ก็จะทำให้การค้นหาข้อมูลที่ต้องการโดยการเปรียบเทียบกับข้อมูลในกลุ่มสีแดงเท่านั้น ดังภาพที่ 2-19 แสดงการแบ่งกลุ่มข้อมูล Binary Code ขนาดความยาวเท่ากับ 15 บิต



ภาพที่ 2-19 การแบ่งกลุ่มข้อมูลโดยใช้ Binary Code ความยาว 15 บิต

จากการทดลองพบว่าจำนวน Binary Code ที่เหมาะสมสำหรับการเข้ารหัสข้อมูลภาพขนาด 64 บิตเท่ากับ 128 บิต ดังนั้นแต่ละภาพจะถูกนำไปทำดัชนีภาพด้วยจำนวนบิตข้อมูลขนาด 128 บิตนั่นเอง ผลลัพธ์ที่ได้จากการทำดัชนีก็จะถูกบันทึกในรูปแบบของไฟล์ .ann

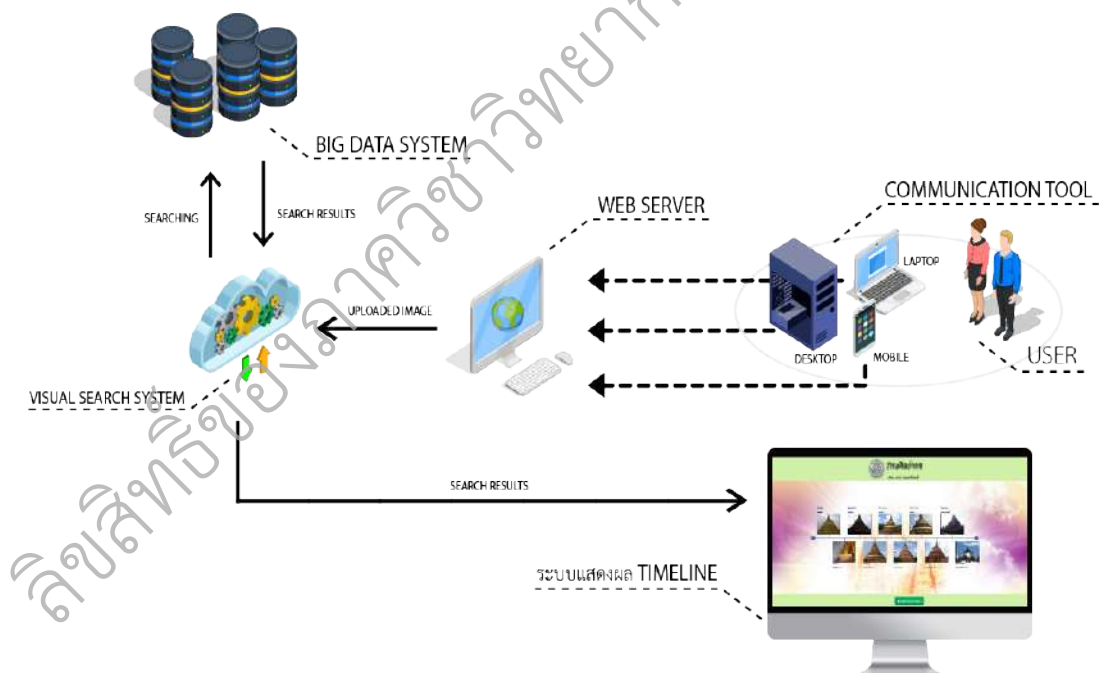
บทที่ 3

ขั้นตอนการดำเนินงาน

ขั้นตอนในการออกแบบพัฒนาระบบ

1. System Architecture
2. ขั้นตอนการดำเนินงาน
3. Use case Diagram
4. Use case Description
5. Activity Diagram
6. Sequence Diagram

3.1 System Architecture



ภาพที่ 3-1 System Architecture

อธิบายภาพที่ 3-1

- 3.1.1 ผู้ใช้งานทำการเปิดหน้าเว็บ
- 3.1.2 ผู้ใช้งานเลือกรูปที่ต้องการ จากนั้นระบบจะส่งข้อมูลรูปไปยังระบบ Machine Learning
- 3.1.3 ระบบ Machine Learning ประมวลผลและส่งข้อมูลรูปภาพที่มีความคล้ายกันออกมา
- 3.1.4 ระบบนำข้อมูลรูปภาพที่มีความคล้ายกันจาก Database มาแสดงผลผ่านทางหน้าเว็บ

3.2 ขั้นตอนการดำเนินงาน

ตารางที่ 3-1 ภาพรวมของการดำเนินงาน

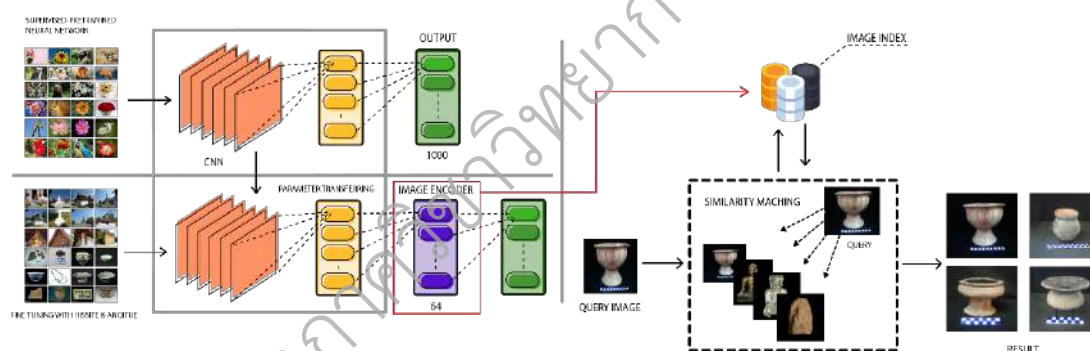
วันที่	สิ่งที่ทำ
22 ก.ค. 2562	- การเสนอหัวข้อโครงการ
23 ก.ค. – 27 ก.ค. 2562	- เก็บและตรวจสอบข้อมูลรูปภาพจากกรมศิลปากร - วางแผนและแบ่งสัดส่วนการทำงาน - ศึกษาการทำหน้าเว็บเพจ - ศึกษาเทคโนโลยีที่เกี่ยวข้อง
28 ก.ค. – 4 ส.ค. 2562	- ติดตั้งโปรแกรมที่เกี่ยวข้อง - ศึกษาการทำงานของ Neural Network
5 ส.ค. – 2 ก.ย. 2562	- สร้างโมเดลสำหรับเรียนรู้รูปภาพ - ทำ User Interface (UI) สำหรับหน้าเว็บเพจ
3 ก.ย. – 17 ก.ย. 2562	- ทดสอบผลการทำนายของโมเดล - นำข้อมูลรูปภาพไปทำดัชนีภาพ (image indexing) - เชื่อมต่อฐานข้อมูล (Database) กับหน้าเว็บเพจ - ทดสอบการดึงข้อมูลรูปภาพและรายละเอียดจากฐานข้อมูล (Database) ให้แสดงผลลัพธ์ไปยังหน้าเว็บเพจ
18 ก.ย. – 21 ต.ค. 2562	- ปรับแก้ไขโมเดลให้เกิดความแม่นยำสูงสุด - เชื่อมต่อหน้าเว็บเพจกับโมเดลทดลอง
22 ต.ค. 2562	- การสอบโครงการพิเศษ 1
23 ต.ค. – 31 ธ.ค. 2562	- เชื่อมต่อหน้าเว็บเพจกับโมเดลจริง - เพิ่มเติมและแก้ไขระบบให้มีความสมบูรณ์

ตารางที่ 3-1 (ต่อ) ภาพรวมของการดำเนินงาน

วันที่	สิ่งที่ทำ
1 ม.ค. – 21 ก.พ. 2563	- ทดสอบการทำงานของหน้าเว็บเพจและโมเดล
22 ก.พ. – 22 มี.ค. 2563	- ทดสอบและแก้ไขระบบ - จัดทำเอกสารระบบ
2 เม.ย. 2563	- การสอบโครงการพิเศษ 2

ขั้นตอนและการดำเนินการสร้าง เว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย (A Web-based Visual Search for Ancient Artifacts) นั้นแบ่งการทำงานออกเป็น 3 ส่วน

3.2.1 การใช้ Machine Learning สร้างโมเดล AI



ภาพที่ 3-2 ขั้นตอนการสร้างโมเดล AI

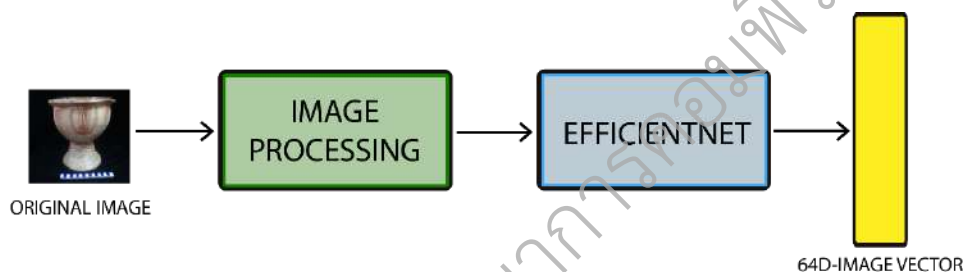
อธิบายภาพที่ 3-2

- 3.2.1.1 เริ่มจากนำโครงข่ายประสาทเทียม ซึ่งถูกฝึกฝนมาแล้วด้วยข้อมูลภาพจำนวนมาก ในโครงข่ายนี้เลือกใช้โครงข่ายประสาทเทียม efficient net ซึ่งเป็นโครงข่ายประสาทเทียม แบบคอนโวลูชัน (Convolution neural network)
- 3.2.1.2 ก่อนที่จะนำโมเดล efficient net ไปใช้งาน จะต้องมีการฝึกฝนโครงข่ายประสาทเทียมเพิ่มเติม เพราะโมเดล efficient net ถูกฝึกฝนมาให้มีความแม่นยำในการจำแนกภาพทั่วไป (1000 Label) เช่น สุนัข แมว เครื่องบิน รถยนต์ เป็นต้น จึงเป็นไปได้ที่จะใช้โมเดล efficient net โดยตรงเพื่อแยกแยะภาพโบราณวัตถุ

อธิบายภาพที่ 3-2 (ต่อ)

- 3.2.1.3 ทำการดัดแปลงโมเดล โดยแทนที่การฝึกฝน จากการให้โครงข่ายประสาทเทียมทำนายชื่อโบราณวัตถุโดยตรง จะให้โครงข่ายประสาทเทียมทำนาย feature ของแต่ละภาพแทน
- 3.2.1.4 จากนั้นใช้ feature ที่โครงข่ายประสาทเทียมทำนายได้ เพื่อถอดรหัสว่าเป็นโบราณวัตถุใดในฐานข้อมูลกรมศิลปากร

3.2.2 การสร้างดัชนีภาพ (image indexing)



ภาพที่ 3-3 การเข้ารหัสภาพโบราณวัตถุเพื่อการทำดัชนีภาพ

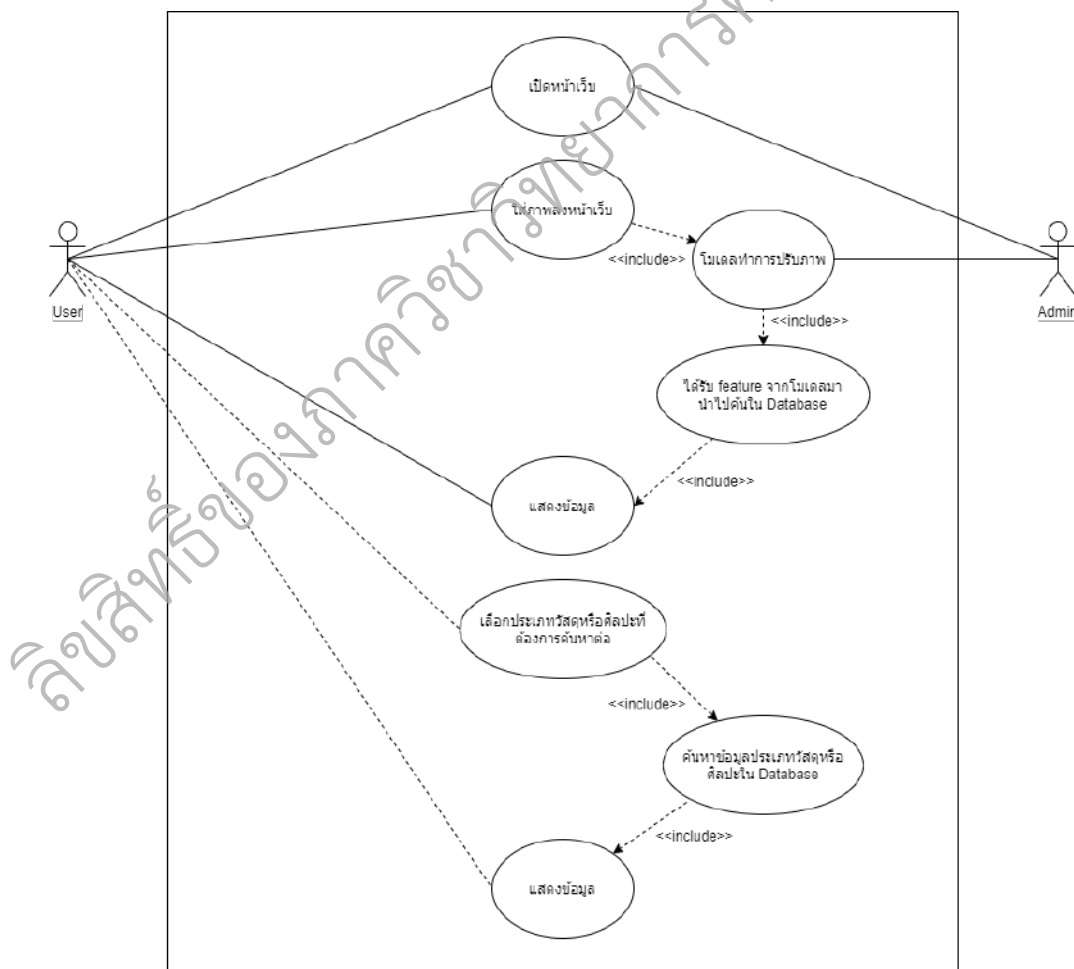
อธิบายภาพที่ 3-3

- 3.2.2.1 เทคนิคที่ถูกนำมาประยุกต์ใช้ในการสร้างดัชนีเพื่อการสืบค้นภาพจะใช้เทคนิค Locality-Sensitive Hashing (LSH)
- 3.2.2.2 แบ่งกลุ่มข้อมูลตามข้อมูลที่มีลักษณะคล้ายคลึงกัน ช่วยจำกัดพื้นที่และจำนวนในการสืบค้นข้อมูล
- 3.2.2.3 ในการแบ่งกลุ่มข้อมูลจะมีการกำหนด binary code เพื่อเข้ารหัสข้อมูลที่มีความใกล้เคียงกัน ขนาดของ binary code จะขึ้นอยู่กับความซับซ้อนของข้อมูลเป็นหลัก
- 3.2.2.4 หากกำหนด binary code ให้มีความยาวมาก ก็จะมีแบ่งกลุ่มข้อมูลมากจะช่วยสืบค้นได้แม่นยำมากขึ้น ขณะที่หากกำหนด binary code น้อยก็จะมีความเร็วในการสืบค้น

3.2.3 การสร้างเว็บไซต์

- 3.2.3.1 ใช้ภาษา Python ในการสร้างเว็บเพจโดยมีการใช้ Flask Framework มาช่วยในการสร้างเว็บเพจ
- 3.2.3.2 สร้างเว็บไซต์สำหรับการนำเข้าข้อมูลรูปภาพ
- 3.2.3.3 ทำการเชื่อมต่อกับโมเดลที่ได้รับการเรนจากรูปโบราณวัตถุแล้ว
- 3.2.3.4 ทำการเชื่อมต่อกับฐานข้อมูลที่มีไฟล์รูปภาพ และรายละเอียดของโบราณวัตถุ
- 3.2.3.5 นำ ID ของภาพที่ได้จากการค้นหาในการทำดัชนีภาพ ไปค้นหาต่อในฐานข้อมูล เพื่อนำข้อมูลรูปภาพ และรายละเอียดมาแสดง
- 3.2.3.6 แสดงผลการทำนายข้อมูลรูปภาพ 3 อันดับ

3.3 Use Case Diagram



ภาพที่ 3-4 Use Case Diagram ของเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย

อธิบายภาพที่ 3-4

- 3.3.1 ผู้ใช้งานทำการเปิดหน้าเว็บ
- 3.3.2 ผู้ใช้งานเพิ่มรูปภาพเข้าไปในระบบ และระบบทำการแปลงรูปภาพเป็นอาเรย์เพื่อนำเข้าไปประมวลผลในโมเดล
- 3.3.3 โมเดลนำภาพมาประมวลผลโดยผลลัพธ์ที่ได้จะเป็นตัว feature แล้วนำไปค้นหารูปภาพใกล้เคียง 3 อันดับใน database
- 3.3.4 จากนั้นตัวโมเดลจะส่งไฟล์ json ออกมา
- 3.3.5 ระบบจะนำไฟล์ json ไปดึงข้อมูลจาก database เพื่อนำไปประมวลผลบนหน้าเว็บ
- 3.3.6 นำผลลัพธ์ที่ได้จากระบบมาแสดง
- 3.3.7 หากผู้ใช้ต้องการค้นหาประเภทวัตถุหรือประเภทศิลปะเพิ่มเติมก็สามารถค้นหาได้โดยเลือกประเภทวัตถุหรือศิลปะนั้น ๆ
- 3.3.8 ระบบจะทำการค้นหาข้อมูลใน database แล้วนำผลลัพธ์ที่ได้จากระบบมาแสดง

3.4 Use case Description

ตารางที่ 3-2 Use case Description : ค้นหาข้อมูลรูปภาพโบราณวัตถุด้วยภาพ

Use Case Name :	ค้นหาข้อมูลภาพโบราณวัตถุด้วยภาพ	
Actor :	ผู้ใช้งานเว็บเพจ	
Purpose :	เพื่อให้ผู้ใช้งานค้นหาข้อมูลของภาพที่ต้องการค้นหา	
Related Use Cases :	-	
Pre – Conditions :	<ul style="list-style-type: none"> - เข้าสู่หน้าหลักของเว็บเพจ - อัปโหลดรูปภาพที่ต้องการค้นหา - กดยืนยัน 	
Post – Conditions :	- แสดงรูปภาพและรายละเอียดของโบราณวัตถุที่มีความใกล้เคียงกับรูปภาพที่ได้ทำการอัปโหลดไป 3 อันดับ	
Flow of Event :	User	System
	1. ผู้ใช้อัปโหลดรูปภาพที่ต้องการค้นหา	

ตารางที่ 3-2 (ต่อ) Use case Description : ค้นหาข้อมูลรูปภาพโบราณวัตถุด้วยภาพ

Flow of Event :	User	System
		2. ระบบทำการแปลงรูปภาพเป็น อาเรย์ แล้วนำส่งรูปภาพเข้าสู่ กระบวนการทำงานของโมเดล
Exception :	1. ประเภทของรูปภาพที่ทำการอัปโหลด ไม่ตรงกับรูปภาพ เป้าหมาย	

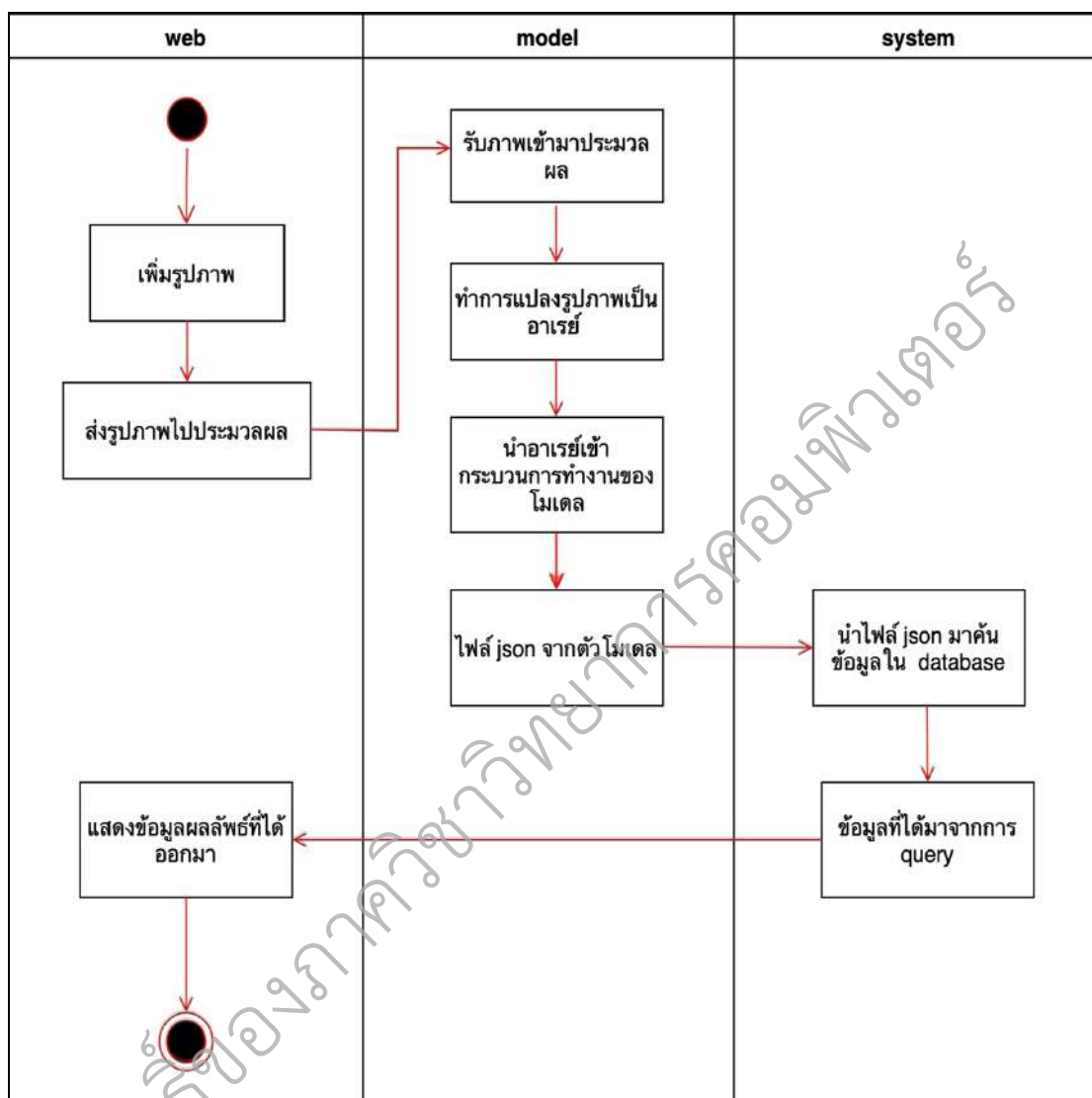
ตารางที่ 3-3 Use case Description : ค้นหาข้อมูลรูปภาพโบราณวัตถุจากวัสดุในการสร้าง

Use Case Name :	ค้นหาข้อมูลภาพโบราณวัตถุจากวัสดุในการสร้าง	
Actor :	ผู้ใช้งานเว็บเพจ	
Purpose :	เพื่อให้ผู้ใช้งานค้นหาข้อมูลของภาพที่ใช้วัสดุในการสร้างเดียวกับ ภาพก่อนหน้า	
Related Use Cases :	<ul style="list-style-type: none"> - ค้นหาข้อมูลภาพโบราณวัตถุด้วยภาพ - ค้นหาข้อมูลภาพโบราณวัตถุจากประเภทของศิลปะ 	
Pre – Conditions :	- กดปุ่มวัสดุที่ต้องการค้นหา	
Post – Conditions :	- แสดงรูปภาพและรายละเอียดของโบราณวัตถุที่ใช้วัสดุในการสร้าง เดียวกับรูปภาพก่อนหน้า 10 อันดับ	
Flow of Event :	User	System
	1. ผู้ใช้กดปุ่มวัสดุที่ต้องการ ค้นหา	2. ระบบทำการเก็บชื่อวัสดุที่ผู้ใช้กด ปุ่มไว้ 3. ระบบนำชื่อไปค้นหาในฐานข้อมูล 4. ระบบนำรูปภาพและรายละเอียด ของโบราณวัตถุ 10 อันดับมาแสดง
Exception :	1. รูปภาพบางภาพ ไม่มีข้อมูลวัสดุในฐานข้อมูล จึงไม่สามารถนำมา แสดงได้	

ตารางที่ 3-4 Use case Description : ค้นหาข้อมูลรูปภาพโบราณวัตถุจากประเภทของศิลปะ

Use Case Name :	ค้นหาข้อมูลภาพโบราณวัตถุจากประเภทของศิลปะ	
Actor :	ผู้ใช้งานเว็บเพจ	
Purpose :	เพื่อให้ผู้ใช้งานค้นหาข้อมูลของภาพที่มีประเภทศิลปะเดียวกับภาพก่อนหน้า	
Related Use Cases :	<ul style="list-style-type: none"> - ค้นหาข้อมูลภาพโบราณวัตถุด้วยภาพ - ค้นหาข้อมูลภาพโบราณวัตถุจากวัสดุในการสร้าง 	
Pre – Conditions :	- กดปุ่มศิลปะที่ต้องการค้นหา	
Post – Conditions :	- แสดงรูปภาพและรายละเอียดของโบราณวัตถุที่มีประเภทศิลปะเดียวกับรูปภาพก่อนหน้า 10 อันดับ	
Flow of Event :	User	System
	1. ผู้ใช้กดปุ่มศิลปะต้องการค้นหา	2. ระบบทำการเก็บชื่อศิลปะที่ผู้ใช้กดปุ่มไว้ 3. ระบบนำชื่อไปค้นหาในฐานข้อมูล 4. ระบบนำรูปภาพและรายละเอียดของโบราณวัตถุ 10 อันดับมาแสดง
Exception :	1. รูปภาพบางภาพ ไม่มีข้อมูลศิลปะในฐานข้อมูล จึงไม่สามารถนำมาแสดงได้	

3.5 Activity Diagram

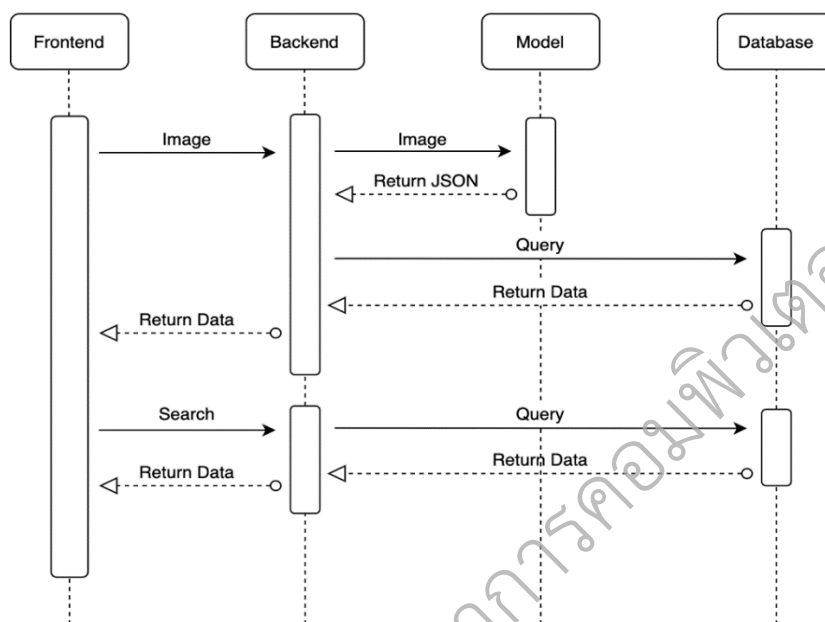


ภาพที่ 3-5 Activity Diagram

อธิบายภาพที่ 3-5

- 3.5.1 ผู้ใช้งานเพิ่มรูปภาพเข้าไปในระบบ และระบบทำการส่งไปรูปภาพประมวลผล (web)
- 3.5.2 ระบบรับรูปภาพเข้ามาประมวลผล จากนั้นทำการแปลงรูปภาพเป็นอาเรย์เพื่อนำเข้าไปประมวลผลในโมเดล จากนั้นตัวโมเดลจะส่งไฟล์ json ออกมา (model)
- 3.5.3 ระบบจะนำไฟล์ json ไปดึงข้อมูลจาก database เพื่อนำไปประมวลผลบนหน้าเว็บ (system)
- 3.5.4 นำผลลัพธ์ที่ได้จาก ระบบมาประมวลผล (web)

3.6 Sequence Diagram



ภาพที่ 3-6 Sequence Diagram

อธิบายภาพที่ 3-6

- 3.6.1 การทำงานของระบบ เริ่มจากผู้ใช้ทำการส่งรูป เข้าไปในระบบผ่านทางเว็บไซต์ (Frontend)
- 3.6.2 หลังจากนั้น ระบบจะนำรูปดังกล่าวส่งผ่านไปยังโมเดล (Backend)
- 3.6.3 ตัวโมเดลจะทำการประมวลผล แล้วส่งไฟล์ json กลับมายัง backend (Model)
- 3.6.4 หลังจากนั้น นำไฟล์ json ที่ได้จากโมเดลไป query ใน database (Backend)
- 3.6.5 Database ดึงข้อมูลและส่งกลับไปให้ backend (Database)
- 3.6.6 จากนั้นจะส่งข้อมูลหลังจากการประมวลผลไปให้หน้าเว็บ (Backend)
- 3.6.7 แสดงผลข้อมูลที่ได้จาก backend (Frontend)
- 3.6.8 ผู้ใช้งานต้องการค้นหาข้อมูลเพิ่มเติม (Frontend)
- 3.6.9 จากนั้น ระบบจะนำข้อมูลที่ต้องการค้นหาไป query ใน database (Backend)
- 3.6.10 Database ดึงข้อมูลและส่งกลับไปให้ backend (Database)
- 3.6.11 จากนั้นจะส่งข้อมูลหลังจากการประมวลผลไปให้หน้าเว็บ (Backend)
- 3.6.12 แสดงผลข้อมูลที่ได้จาก backend (Frontend)

3.7 การออกแบบการทดลอง

ขั้นตอนการออกแบบการทดลองเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย (A Web-based Visual Search for Ancient Artifacts)

3.7.1 ชุดข้อมูลทดลอง (Dataset) ที่นำมาใช้สำหรับการทดลองเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่ายนั้น ถูกแบ่งออกเป็น 2 ประเภทดังตารางที่ 3-5

ตารางที่ 3-5 แสดงประเภทชุดข้อมูลภาพทดลอง

ประเภทชุดข้อมูลทดลอง	จำนวนภาพ	แหล่งที่มา	ตัวอย่างข้อมูลรูปภาพ
ชุดข้อมูลภาพทดลองที่มีในฐานข้อมูล	20	ฐานข้อมูลจากทางกรมศิลปากร	

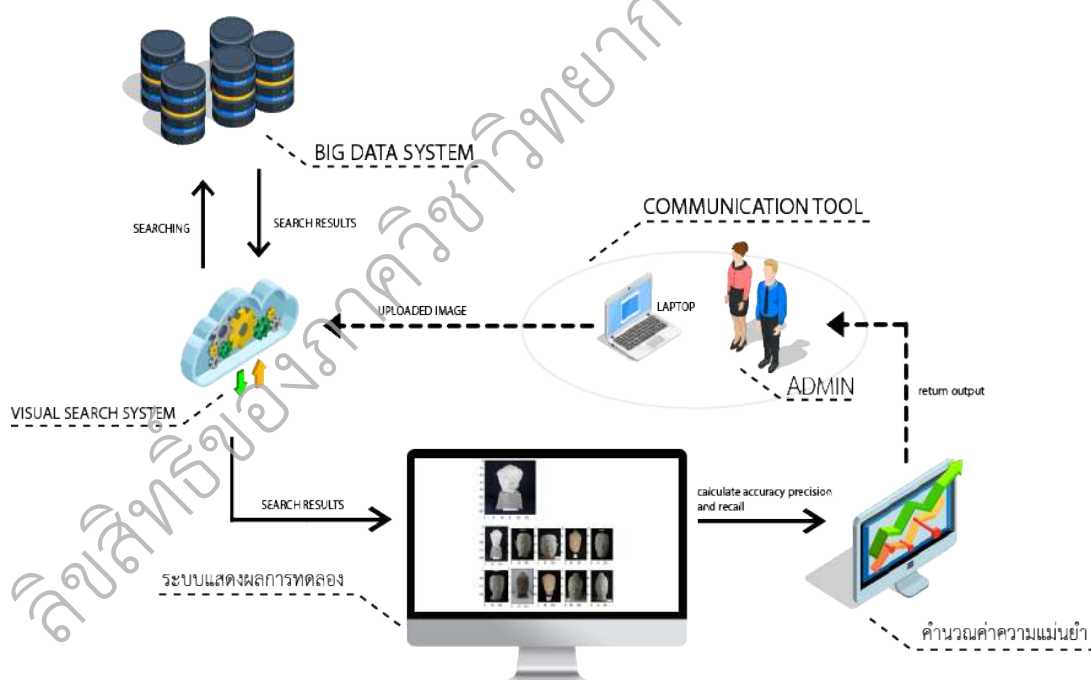
ตารางที่ 3-5 (ต่อ) แสดงประเภทชุดข้อมูลภาพทดลอง

ประเภทชุดข้อมูลทดลอง	จำนวนภาพ	แหล่งที่มา	ตัวอย่างข้อมูลรูปภาพ
ชุดข้อมูลภาพทดลองที่ไม่มีในฐานข้อมูล	37	ภายในเว็บไซต์ของกรมศิลปากรและอื่น ๆ	 72/17g  4/15  © 2016 The National Storage Museum 33/4/11  16/25/2542

3.7.2 การประเมินผลการปฏิบัติงาน (Performance Evaluation)

- 3.7.2.1 นำชุดภาพทดลองทั้ง 2 ประเภท เข้าสู่กระบวนการประมวลผลของโมเดล เพื่อให้โมเดลทำนายและส่ง feature ออกมา เพื่อไปค้นหาในฐานข้อมูล
- 3.7.2.2 ผลลัพธ์จะถูกส่งออกมาเป็นภาพที่มีความใกล้เคียงกับภาพนำเข้า โดยทางผู้จัดทำกำหนดให้แสดงผลออกมา 10 อันดับ
- 3.7.2.3 เช็กผลลัพธ์กับชุดภาพทดลองที่ได้นำเข้าไว้ ว่ามีภาพใกล้เคียงอยู่ในอันดับที่เท่าไร และทำการบันทึกผลการทดลอง
- 3.7.2.4 เมื่อนำเข้าชุดภาพทดลองครบทั้ง 57 ภาพแล้ว นำเปอร์เซ็นต์จากการบันทึกผลการทดลอง มาคำนวณ Precision และ Accuracy นำออกมาเป็นเปอร์เซ็นต์

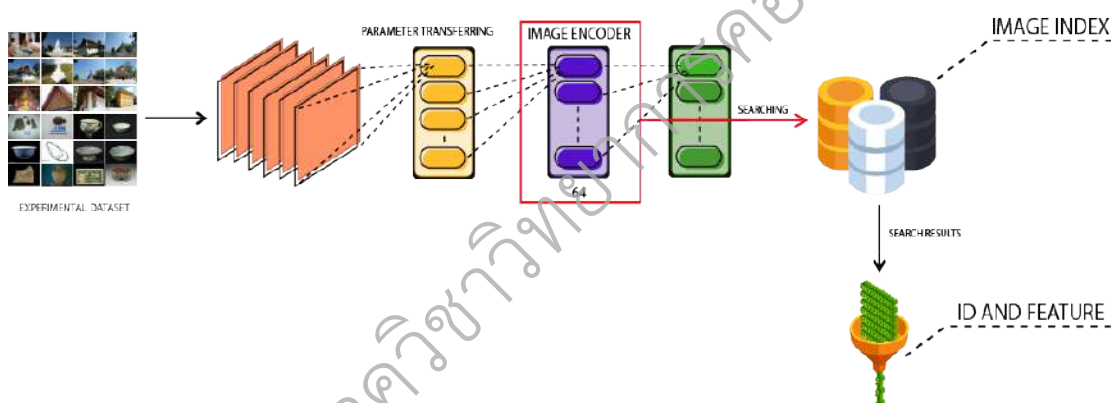
3.7.3 Network Architecture



ภาพที่ 3-7 ภาพรวมของกระบวนการทดสอบระบบ

อธิบายภาพที่ 3-7

- 3.7.3.1 ผู้จัดทำ หรือผู้ทำการทดลองระบบ นำชุดภาพทดลองเข้าสู่กระบวนการประมวลผลของโมเดล เพื่อให้โมเดลทำนาย
- 3.7.3.2 ผลลัพธ์ที่ได้ออกมาเป็น feature และทำการค้นหาใน Database เพื่อดึงภาพใกล้เคียงออกมาแสดง 10 อันดับ
- 3.7.3.3 เช็คผลลัพธ์กับชุดภาพทดลองที่ได้นำเข้าไว้ ว่ามีภาพใกล้เคียงอยู่ในอันดับที่เท่าไร และทำการบันทึกผลการทดลอง
- 3.7.3.4 เมื่อสิ้นสุดการนำชุดภาพทดลองเข้าทดลอง จะนำผลบันทึกผลการทดลอง มาคำนวณ Precision และ Accuracy นำออกมาเป็นเปอร์เซ็นต์



ภาพที่ 3-8 Network Architecture โครงสร้าง Neural Network

อธิบายภาพที่ 3-8

- 3.7.3.5 เมื่อชุดภาพทดลองถูกนำเข้าสู่กระบวนการประมวลผลของโมเดล โมเดลจะเริ่มทำนาย โดยการปรับขนาดภาพ และแปลงภาพเป็นรูปแบบของอาเรย์
- 3.7.3.6 จากนั้นจะนำเข้าสู่กระบวนการทำนายผลที่ได้รับ และแสดงผลลัพธ์
- 3.7.3.7 ผลลัพธ์ที่ได้จะออกมาเป็น feature และ feature นี้จะถูกนำไปทำการค้นหาใน Database ที่มี feature อื่น ๆ ที่เคยได้รับการ Train มาแล้ว เพื่อดึง Id และ feature ใกล้เคียงออกมาแสดง 10 อันดับ
- 3.7.3.8 และนำ feature เหล่านี้ไปดึงข้อมูลภาพ เพื่อแสดงผลผ่านทางหน้าจอแสดงผล

บทที่ 4

การพัฒนาระบบ

จากการศึกษาค้นคว้าข้อมูลที่เกี่ยวข้องกับการพัฒนาระบบ รวมทั้งศึกษาการออกแบบระบบการทำงานของระบบเพื่อเป็นแนวทางในการพัฒนาระบบเพื่อให้เป็นไปอย่างมีประสิทธิภาพ ในบทนี้จะอธิบายถึงการพัฒนาระบบ โดยแสดงถึงรายละเอียดของสิ่งแวดล้อม และเครื่องมือต่าง ๆ ที่ใช้ในการพัฒนาระบบ เพื่อให้สามารถทำงานได้ตามที่ได้ออกแบบไว้

4.1 เครื่องมือที่ใช้ในการพัฒนาระบบ

4.1.1 เครื่องมือ

- ภาษา Python
- Visual Studio Code 2019
- Jupyter Notebook

4.1.2 ระบบปฏิบัติการที่ใช้ในการพัฒนาระบบ

- Windows 10 Pro
- macOS Catalina version 10.15.3

4.1.3 ระบบปฏิบัติการที่ใช้ในการพัฒนาระบบ

4.1.3.1 สำหรับสร้างหน้าเว็บเพจ

- MacBook Pro 16 inch
- Processor : Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz
- RAM : 16.0 GB
- System type : 64-bit operating system, x64-based

4.1.3.2 สำหรับสร้างโมเดล

- Notebook Acer Nitro AN515-52
- Processor : Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz
- RAM : 8.00 GB (7.85 GB usable)
- System type : 64-bit operating system, x64-based processor

4.2 การพัฒนาระบบ

4.2.1 การพัฒนาโมเดล Machine Learning

```

10. def create_label_mapping(mapping_csv):
11.     labels = set()
12.     for i in range(len(mapping_csv)):
13.         x = [mapping_csv['antique_type_id'][i]]
14.         y = [mapping_csv['style_id'][i]]
15.         labels.update(x)
16.         labels.update(y)
17.     labels = list(labels)
18.     labels.sort()
19.     labels_map = {labels[i]:i for i in range(len(labels))}
20.     inv_labels_map = {i:labels[i] for i in range(len(labels))}
21.     return labels_map, inv_labels_map

```

ภาพที่ 4-1 โค้ดแสดงการสร้างข้อมูล Output

อธิบายภาพที่ 4-1

บรรทัดที่ 13-16 กำหนดตัวแปร x และ y สำหรับเตรียมข้อมูล Output โดย x คือ จำนวนคลาสของวัสดุ และ y คือ จำนวนคลาสของศิลปะ จากนั้นนำ x และ y update เข้า labels

บรรทัดที่ 18-21 ทำการ sort ข้อมูล Output จากนั้นทำการแบ่งข้อมูลออกตามลำดับ แล้ว return

```

23. def create_file_mapping(mapping_csv):
24.     mapping = dict()
25.     for i in range(len(mapping_csv)):
26.         name = mapping_csv['file_name'][i]
27.         type_name = mapping_csv['antique_type_id'][i]
28.         style = mapping_csv['style_id'][i]
29.         mapping[name] = [type_name, style]
30.     return mapping

```

ภาพที่ 4-2 โค้ดแสดงการสร้างข้อมูล Input

อธิบายภาพที่ 4-2

บรรทัดที่ 24-30 สร้าง mapping แล้วนำเข้าสู่ข้อมูลของชื่อไฟล์รูปภาพ และมีจำนวนคลาสของวัตถุกับศิลปะ ที่เป็นของชื่อไฟล์นั้น ๆ จากนั้น return

```

32. def one_hot_encode(tags, mapping):
33.     encoding = zeros(len(mapping), dtype = 'uint8')
34.     for tag in tags:
35.         encoding[mapping[tag]] = 1
36.     return encoding

```

ภาพที่ 4-3 โค้ดแสดงการแปลงเป็น one hot vector

อธิบายภาพที่ 4-3

บรรทัดที่ 32-36 ทำการแปลงเป็น one hot vector เพื่อนำไปเทรนได้


```

38. def load_dataset(path, file_mapping, tag_mapping):
39.     photos, targets = list(), list()
40.     for filename in listdir(folder):
41.         if filename in file_mapping:
42.             photo = load_img(path + filename, target_size = (150, 150))
43.             photo = img_to_array(photo, dtype = 'uint8')
44.             labels = file_mapping[filename]
45.             target = one_hot_encode(labels, tag_mapping)
46.             photos.append(photo)
47.             targets.append(target)
48.     X = asarray(photos, dtype = 'uint8')
49.     y = asarray(targets, dtype = 'uint8')
50.     return X, y

```

ภาพที่ 4-4 โค้ดแสดงการเตรียมข้อมูลสำหรับนำไปเทรน

อธิบายภาพที่ 4-4

- บรรทัดที่ 42 ทำการโหลดข้อมูลรูปภาพ แล้วทำการ resize เป็น 150*150 เพื่อความรวดเร็ว และประหยัดพื้นที่ในการเทรน
- บรรทัดที่ 43 แปลงรูปภาพเป็น array
- บรรทัดที่ 44 ให้ labels เป็นข้อมูลนำเข้า (Input)
- บรรทัดที่ 45 ให้ target เป็นข้อมูลส่งออก (Output) โดยส่งไปยังฟังก์ชัน one_hot_encode ดังภาพที่ 4-3
- บรรทัดที่ 46-50 นำเข้าข้อมูลรูปภาพ, Input และ Output แล้ว return

```

52. filename = 'antiquitiess.csv'
53. csv_data = read_csv(filename)
54. label_mapping, _ = create_label_mapping(csv_data )
55. file_mapping = create_file_mapping(csv_data )
56. print('..... load_data .....')
57. X, y = load_dataset(folder, file_mapping, label_mapping)
58.
59. print('read image data 150x150')
60. trainX, testX, trainY, testY = train_test_split(X, y, test_size = 0.2,
        random_state = 1)
61. print(trainX.shape, trainY.shape, testX.shape, testY.shape)

```

ภาพที่ 4-5 โค้ดแสดงการเตรียมข้อมูลสำหรับนำไปเทรน และการแบ่งข้อมูล train และ test

อธิบายภาพที่ 4-5

- บรรทัดที่ 52-53 การเรียกไฟล์ที่มีข้อมูลสำหรับนำมาใช้ในการเทรน
- บรรทัดที่ 54-59 การเตรียมข้อมูลสำหรับนำไปเทรน โดยส่งไปยังฟังก์ชันต่าง ๆ
- บรรทัดที่ 60 การแบ่งข้อมูลสำหรับ train และ test

```

13. def fbeta(y_true, y_pred, beta=2):
14.     y_pred = backend.clip(y_pred, 0, 1)
15.     tp = backend.sum(backend.round(backend.clip(y_true * y_pred, 0, 1)),
16.                       axis = 1)
17.     fp = backend.sum(backend.round(backend.clip(y_pred - y_true, 0, 1)),
18.                       axis = 1)
19.     fn = backend.sum(backend.round(backend.clip(y_true - y_pred, 0, 1)),
20.                       axis = 1)
21.     p = tp / (tp + fp + backend.epsilon())
22.     r = tp / (tp + fn + backend.epsilon())
23.     bb = beta ** 2
24.     fbeta_score = backend.mean((1 + bb) * (p * r) / (bb * p + r +
25.                                                         backend.epsilon()))
26.     return fbeta_score

```

ภาพที่ 4-6 โค้ดแสดงการคำนวณความแม่นยำสำหรับการเทรน

อธิบายภาพที่ 4-6

บรรทัดที่ 13-22 ฟังก์ชันการคำนวณความแม่นยำสำหรับการเทรน

```

24. def define_modelVGG(in_shape=(150, 150, 3), out_shape = 10):
25.     model = VGG16(include_top = False, weights = 'imagenet', input_shape
        =
        in_shape)
26.
27.     for layer in model.layers:
28.         layer.trainable = False
29.
30.     model.get_layer('block5_conv1').trainable = True
31.     model.get_layer('block5_conv2').trainable = True
32.     model.get_layer('block5_conv3').trainable = True
33.     model.get_layer('block5_pool').trainable = True
34.
35.     flat1 = Flatten()(model.layers[-1].output)
36.     class1 = Dense(128, activation = 'relu', kernel_initializer =
        'he_uniform')(flat1)
37.     output = Dense(out_shape, activation = 'sigmoid')(class1)
38.     model = Model(inputs = model.inputs, outputs = output)
39.
40.     opt = SGD(lr = 0.005, momentum = 0.9)
41.     model.compile(optimizer = opt, loss = 'binary_crossentropy',
        metrics = [fbeta])
42.     return model

```

ภาพที่ 4-7 โค้ดแสดงการปรับส่วนของโมเดล VGG16

อธิบายภาพที่ 4-7

บรรทัดที่ 24-33 การเรียกใช้โมเดล VGG16 โดยตัดส่วน Output ของโมเดลออก

บรรทัดที่ 35-42 เพิ่มเติม hidden layer ส่วนต่าง ๆ ที่จำเป็นเข้าไปในโมเดล แล้ว return

```

44. model = define_modelVGG()
45. history = model.fit(trainX, trainY, validation_data = (testX, testY), batch_size
    = 16, epochs = 20, verbose = 1)
46. loss, fbeta = model.evaluate(testX, testY)
47. print('> loss=%.3f, fbeta=%.3f' % (loss, fbeta))
48. model.save('final_antiques_model.h5')

```

ภาพที่ 4-8 โค้ดแสดงการเทรนโมเดล

อธิบายภาพที่ 4-8

- บรรทัดที่ 44 เรียกใช้โมเดลโดยส่งไปที่ฟังก์ชัน `define_modelVGG`
- บรรทัดที่ 45-48 ทำการเทรนโมเดลโดยใช้ข้อมูลที่ได้มาเข้ามาแล้ว พร้อมทั้งแสดงความแม่นยำ และข้อมูล loss แล้ว save โมเดล

```

13. parser = argparse.ArgumentParser()
14. parser.add_argument('--model', default = 'final_antiques_model.h5', type =
    str, help = 'weight model')
15. parser.add_argument('--index', default = 'antiques_idx.ann', type = str, help
    = 'The filename of image index')
16. parser.add_argument('--path', default = 'thumb/', type = str, help = 'path of
    images')

```

ภาพที่ 4-9 โค้ดแสดงการสร้าง Argument

อธิบายภาพที่ 4-9

- บรรทัดที่ 13-16 สร้าง Argument โดยตั้ง default ของโมเดล, Index และ path ของรูปภาพไว้

```

18. def fbeta(y_true, y_pred, beta=2):
19.     y_pred = backend.clip(y_pred, 0, 1)
20.     tp = backend.sum(backend.round(backend.clip(y_true * y_pred, 0, 1)),
21.                       axis = 1)
21.     fp = backend.sum(backend.round(backend.clip(y_pred - y_true, 0, 1)),
22.                       axis = 1)
22.     fn = backend.sum(backend.round(backend.clip(y_true - y_pred, 0, 1)),
23.                       axis = 1)
23.     p = tp / (tp + fp + backend.epsilon())
24.     r = tp / (tp + fn + backend.epsilon())
25.     bb = beta ** 2
26.     fbeta_score = backend.mean((1 + bb) * (p * r) / (bb * p + r +
27.                                                       backend.epsilon()))
27.     return fbeta_score

```

ภาพที่ 4-10 โค้ดแสดงการคำนวณความแม่นยำสำหรับการทำดัชนีภาพ

อธิบายภาพที่ 4-10

บรรทัดที่ 18-27 ฟังก์ชันการคำนวณความแม่นยำสำหรับการทำดัชนีภาพ

```

29. def load_database(user1 = 'postgres', password1 = 'b', host1 = '127.0.0.1',
    database1 = 'anti'):
30.     con = mysql.connector.connect(user = user1, password = password1,
        host= host1, database = database1, auth_plugin =
        'mysql_native_password')
31.     cursor = con.cursor()
32.     sql = ("select file_name, id from files where file_name is not null")
33.     cursor.execute(sql)
34.     p = cursor.fetchall()
35.     data = dict(p)
36.     return data

```

ภาพที่ 4-11 โค้ดแสดงการโหลดข้อมูลจากฐานข้อมูล

อธิบายภาพที่ 4-11

บรรทัดที่ 29-36 การเชื่อมต่อกับฐานข้อมูล และ select ชื่อไฟล์ของรูปภาพ แล้ว return

```

38. def load_image(filename):
39.     img = load_img(filename, target_size = (150, 150))
40.     img = img_to_array(img)
41.     img = img.reshape(1, 150, 150, 3)
42.     img = img.astype('uint8')
43.     return img

```

ภาพที่ 4-12 โค้ดแสดงการโหลดข้อมูลรูปภาพ

อธิบายภาพที่ 4-12

บรรทัดที่ 39 ทำการโหลดข้อมูลรูปภาพ แล้วทำการ resize เป็น 150*150 เพื่อความรวดเร็ว และประหยัดพื้นที่ในการทำดัชนีภาพ

บรรทัดที่ 40-43 แปลงรูปภาพเป็น array และ reshape ให้เท่ากับขนาดต่อนเทรน แล้ว return

```

45. def normalize(x):
46.     normalized_values = list()
47.     maximum = np.max(x)
48.     minimum = np.min(x)
49.     for y in x:
50.         x_normalized = (y - minimum) / (maximum - minimum)
51.         normalized_values.append(x_normalized)
52.     n_array = np.array(normalized_values)
53.     return n_array

```

ภาพที่ 4-13 โค้ดแสดงการ Normalize

อธิบายภาพที่ 4-13

บรรทัดที่ 45-53 ทำการ Normalize ข้อมูล

```

55. def build_index(VEC_LENGTH = 64, NUM_TREES = 30):
56.     model = load_model(args.model, custom_objects={'fbeta': fbeta})
57.     intermediate_layer_model = Model(input = model.input, output =
        model.get_layer('dense_3').output)
58.     print('load model is okay')
59.
60.     t = AnnoyIndex(VEC_LENGTH, 'angular')
61.     folder = args.path
62.     db = load_database()
63.     print('load database is okay')
64.
65.     for filename in listdir(folder):
66.         if filename in db.keys():

```

ภาพที่ 4-14 โค้ดแสดงการสร้างดัชนีภาพ


```

67.         print('load ', filename)
68.         img = load_image(os.path.join(folder, filename))
69.         feature = intermediate_layer_model.predict(img)
70.         data = np.reshape(feature, (-1, ))
71.         x = normalize(data)
72.         t.add_item(db[filename], x)
73.     t.build(NUM_TREES)

t.save(args.index)

```

ภาพที่ 4-14 (ต่อ) โค้ดแสดงการสร้างดัชนีภาพ

อธิบายภาพที่ 4-14

- บรรทัดที่ 56-57 โหลดโมเดลจาก Argument และนำส่วน hidden layer ก่อนที่จะถึง Output ออกมาใช้
- บรรทัดที่ 60-62 ทำการสร้างดัชนีภาพ และกำหนด path ที่ใช้เรียกข้อมูลรูปภาพ จากนั้นทำการโหลดฐานข้อมูล โดยส่งไปที่ฟังก์ชัน load_database
- บรรทัดที่ 65-74 อ่านชื่อไฟล์ แล้วนำไปโหลดข้อมูลรูปภาพที่ฟังก์ชัน load_image จากนั้นนำไปใส่ไว้ในต้นไม้ที่ได้ทำการสร้างดัชนีภาพเอาไว้ และ save ต้นไม้นี้ จะได้ไฟล์เป็น .ann

```

76. if __name__ == '__main__':
77.     args = parser.parse_args()
78.     build_index(64, 30)
79.     print('build index is okay...')

```

ภาพที่ 4-15 โค้ดแสดงการเรียกฟังก์ชันสำหรับสร้างดัชนีภาพ

อธิบายภาพที่ 4-15

- บรรทัดที่ 77-78 เรียก Argument ที่ได้ตั้ง default ไว้ในตัวแปร args จากนั้นส่งไปที่ฟังก์ชัน build_index เพื่อทำการสร้างดัชนีภาพ

4.2.2 การทำงานในส่วน backend ของเว็บเพจในการพัฒนาใช้ภาษา Python

```
44. @app.route("/")
45. def hello():
46.     return render_template("index.html")
```

ภาพที่ 4-16 โค้ดแสดงการเริ่มการทำงานของตัวหน้าเว็บเพจ

อธิบายภาพที่ 4-16

บรรทัดที่ 44	เป็นการกำหนดว่า url path ไหนจะเรียกฟังก์ชันไหน '/' ก็จะเรียกฟังก์ชัน Home มา render ตัว html
บรรทัดที่ 45	สร้างฟังก์ชันขึ้นมา 1 ฟังก์ชัน (hello)
บรรทัดที่ 46	การไปเรียกหน้า index.html มาใช้

```
48. def load_m(model_file):
49.     model = load_model(model_file, compile=False)
50.     intermediate_layer_model = Model(inputs = model.input, outputs =
        model.get_layer('dense_3').output)
51.     return intermediate_layer_model
```

ภาพที่ 4-17 โค้ดแสดงการโหลดโมเดลจาก Argument มาใช้

อธิบายภาพที่ 4-17

บรรทัดที่ 48-51	โหลดโมเดลจาก Argument และนำส่วน hidden layer ก่อนที่จะถึง Output ออกมาใช้ แล้ว return
-----------------	---

```

53. def load_database(user1='root', password1='b12345678b', host1 =
    '202.44.40.189', database1='antique'):
54.     con = mysql.connector.connect(user=user1, password=password1,
        host=host1, database=database1,
        auth_plugin='mysql_native_password')
55.     cursor = con.cursor()
56.     sql = ("select id, file_name from files where file_name is not null")
57.     cursor.execute(sql)
58.     p = cursor.fetchall()
59.     data1 = dict(p)
60.     sql = ("select id, antique_id from files where file_name is not null")
61.     cursor.execute(sql)
62.     p = cursor.fetchall()
63.     data2 = dict(p)
64.     return data1, data2

```

ภาพที่ 4-18 โค้ดแสดงการโหลดฐานข้อมูล

อธิบายภาพที่ 4-18

บรรทัดที่ 53-64 ทำการโหลดฐานข้อมูล โดยแยกเป็น 2 ส่วน ส่วนแรกไปเรียกข้อมูล Id และ file_name มาใช้ และส่วนที่สอง ไปเรียก Id และ antique_id มาใช้

```

66. @app.route("/upload", methods=["POST", "GET"])
67. def upload():
68.     if request.method == "POST":
69.         if request.files:
70.             file = request.files["image"]
71.             filepath = "./static/img/image.jpg"
72.             file.save(filepath)
73.     return redirect(url_for('upload1'))

```

ภาพที่ 4-19 โค้ดแสดงการทำงานส่วนการรับข้อมูลรูปภาพ

อธิบายภาพที่ 4-19

บรรทัดที่ 28 การรับ url จากหน้าเว็บ
 บรรทัดที่ 34-36 รับ ไฟล์รูปภาพ มาจากหน้าเว็บ สร้าง path
 สำหรับเก็บรูปและเซฟไฟล์ ชื่อ image.jpg

```

78. @app.route("/upload1")
79. def upload1():
80.     img = load_img("./static/img/image.jpg", target_size = (150, 150))
81.     img = img_to_array(img)
82.     img = img.reshape(1, 150, 150, 3)
83.     img = img.astype('uint8')
84.     result = model.predict(img)
85.     data = np.reshape( result , (-1 , ))

```

ภาพที่ 4-20 โค้ดแสดงการโหลดข้อมูลรูปภาพ

อธิบายภาพที่ 4-20

บรรทัดที่ 80 ทำการโหลดข้อมูลรูปภาพ แล้วทำการ resize เป็น 150*150 เพื่อความ
 รวดเร็ว และประหยัดพื้นที่ในการนำเข้าโมเดล

อธิบายภาพที่ 4-20 (ต่อ)

บรรทัดที่ 81-83 แปลงรูปภาพเป็น array และ reshape ให้เท่ากับขนาดตอนเทรน

บรรทัดที่ 84-85 นำเข้าโมเดล เพื่อให้ได้ผลการทำนาย

```
87. normalized_values = list()
88.     maximum = np.max(data)
89.     minimum = np.min(data)
90.     for y in data:
91.         x_normalized = (y - minimum) / (maximum - minimum)
92.         normalized_values.append(x_normalized)
93.     n_array = np.array(normalized_values)
```

ภาพที่ 4-21 โค้ดแสดงการ Normalize

อธิบายภาพที่ 4-21

บรรทัดที่ 87-93 ทำการ Normalize ข้อมูล

```
95. t = AnnoyIndex(128, 'angular')
96.     t.load(args.index)
97.     img_ids = t.get_nns_by_vector(n_array, 10, include_distances = True)
```

ภาพที่ 4-22 โค้ดแสดงการค้นหา feature ในดัชนีภาพ

อธิบายภาพที่ 4-22

บรรทัดที่ 95-97 ทำการโหลดต้นไม้ของดัชนีภาพ จาก Argument และส่ง feature ที่ต้องการค้นหาเข้าไปค้นหา และให้แสดงผล

```

99. result = {}
100. result['result'] = []
101. for j in range(len(img_ids[0])):
102.     if img_ids[1][j] <= 1:
103.         result['result'].append({'obj_id': db2[img_ids[0][j]],
                                   'img_file': db1[img_ids[0][j]]})
104.     x = db2[img_ids[0][0]]
105.     y = db2[img_ids[0][1]]
106.     z = db2[img_ids[0][2]]
107. with open('result.json', 'w') as outfile:
108.     json.dump(result, outfile)

```

ภาพที่ 4-23 โค้ดแสดงการสร้างไฟล์ Json ที่มีผลการทำนาย 3 อันดับ

อธิบายภาพที่ 4-23

บรรทัดที่ 99-108 ทำการสร้างไฟล์ result.json เพื่อเก็บค่าผลการทำนายรูปภาพใกล้เคียง 3 อันดับ ภายในไฟล์ประกอบไปด้วย Id รูปภาพ, ชื่อไฟล์รูปภาพ

```

110. cursor = connection.cursor()
111. sql_query = "SELECT a.id, f.file_name, a.property, a.culture,
               a.antique_type_other from antiquities a, files f where
               f.antique_id = a.id and a.id=%s"
112. cursor.execute(sql_query, (x,))
113. record = cursor.fetchall()
114. cursor.execute(sql_query, (y,))
115. record1 = cursor.fetchall()
116. cursor.execute(sql_query, (z,))

```

ภาพที่ 4-24 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (1)

```

117.         record2 = cursor.fetchall()
118.         cursor.close()
119.         return render_template("showing.html", rec = record, rec1 =
            record1, rec2 = record2)

```

ภาพที่ 4-24 (ต่อ) โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (1)

อธิบายภาพที่ 4-24

บรรทัดที่ 110-118 การ query ข้อมูลโดยรับ ตัวแปร x, y, z มาทำการ query

บรรทัดที่ 119 การไปเรียกหน้า showing.html ละส่งข้อมูลที่เก็บอยู่ในตัวแปร rec, rec1, rec2 ไปยังหน้าเว็บเพจ

```

121. @app.route('/newpage/<string:culture>', methods=["POST", "GET"])
122. def newpage(culture):
123.     cursor = connection.cursor()
124.     sql_query = "SELECT antiquities.id, files.file_name
        , antiquities.property, antiquities.culture
        , antiquities.antique_type_other FROM antiquities INNER JOIN files ON
        antiquities.id = files.antique_id AND antiquities.culture = %s ORDER
        BY antiquities.id limit 10"
125.     sql_query = sql_query.encode('utf-8')
126.     cursor.execute(sql_query, (culture,))
127.     record = cursor.fetchall()
128.     connection.commit()
129.     return render_template("culture.html", record=record)

```

ภาพที่ 4-25 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (2)

อธิบายภาพที่ 4-25

- บรรทัดที่ 121 รับ url จากหน้าเว็บเพจโดยมีการส่ง parameter culture มาจากหน้าเว็บเพจ
- บรรทัดที่ 122 เป็นการรับ parameter culture จากหน้าเว็บเพจ
- บรรทัดที่ 123-127 การ query ข้อมูลจาก Database โดยเอาเอาข้อมูลจาก parameter มา query และดึงข้อมูลออกมาเป็น list
- บรรทัดที่ 129 การไปเรียกหน้า culture.html มาใช้ และส่งข้อมูลจากตัวแปร record ไปแสดงในหน้าเว็บเพจ

```

131. @app.route('/newpage/<string:culture>/<string:material>',
      methods=["POST", "GET"])
132. def newpage1(culture, material):
133.     cursor = connection.cursor()
134.     sql_query = "SELECT antiquities.id, files.file_name
                   , antiquities.property, antiquities.culture
                   , antiquities.antique_type_other FROM antiquities INNER JOIN files ON
                   antiquities.id = files.antique_id AND antiquities.culture = %s AND
                   antiquities.antique_type_other = %s ORDER BY antiquities.id limit 10"
135.     sql_query = sql_query.encode('utf-8')
136.     cursor.execute(sql_query, (culture, material,))
137.     record = cursor.fetchall()
138.     connection.commit()
139.     return render_template("culture.html", record=record)

```

ภาพที่ 4-26 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (3)

อธิบายภาพที่ 4-26

- บรรทัดที่ 131 รับ url จากหน้าเว็บเพจโดยมีการส่ง parameter culture และ material มาจากหน้าเว็บเพจ

อธิบายภาพที่ 4-26 (ต่อ)

- บรรทัดที่ 132 เป็นการรับ parameter culture และ material จากหน้าเว็บเพจ
- บรรทัดที่ 133-137 การ query ข้อมูลจาก Database โดยเอาเอาข้อมูลจาก parameter มา query และดึงข้อมูลออกมาเป็น list
- บรรทัดที่ 139 การไปเรียกหน้า culture.html มาใช้ และส่งข้อมูลจากตัวแปร record ไปแสดงในหน้าเว็บเพจ

```

141. @app.route('/newmaterial/<string:material>', methods=["POST", "GET"])
142. def newmaterial(material):
143.     cursor = connection.cursor()
144.     sql_query = "SELECT antiquities.id, files.file_name
                    , antiquities.property, antiquities.culture
                    , antiquities.antique_type_other FROM antiquities INNER JOIN files ON
                    antiquities.id = files.antique_id AND
                    antiquities.antique_type_other = %s ORDER BY antiquities.id limit 10"
145.     sql_query = sql_query.encode('utf-8')
146.     cursor.execute(sql_query, (material,))
147.     record = cursor.fetchall()
148.     connection.commit()
149.     return render_template("material.html", record=record)

```

ภาพที่ 4-27 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (4)

อธิบายภาพที่ 4-27

- บรรทัดที่ 141 รับ url จากหน้าเว็บเพจโดยมีการส่ง parameter material มาจากหน้าเว็บเพจ
- บรรทัดที่ 142 เป็นการรับ parameter material จากหน้าเว็บเพจ
- บรรทัดที่ 143-147 การ query ข้อมูลจาก Database โดยเอาเอาข้อมูลจาก parameter มา query และดึงข้อมูลออกมาเป็น list
- บรรทัดที่ 149 การไปเรียกหน้า material.html มาใช้ และส่งข้อมูลจากตัวแปร record ไปแสดงในหน้าเว็บเพจ

```

151. @app.route('/newmaterial/<string:material>/<string:culture>',
      methods=["POST", "GET"])
152. def newmaterial1(material, culture):
153.     cursor = connection.cursor()
154.     sql_query = "SELECT antiquities.id, files.file_name ,antiquities.property
                    ,antiquities.culture ,antiquities.antique_type_other FROM antiquities INNER
                    JOIN files ON antiquities.id=files.antique_id AND
                    antiquities.antique_type_other=%s AND antiquities.culture=%s ORDER BY
                    antiquities.id limit 10 "
155.     sql_query = sql_query.encode('utf-8')
156.     cursor.execute(sql_query, (material, culture,))
157.     record = cursor.fetchall()
158.     connection.commit()
159.     return render_template("material.html", record=record)

```

ภาพที่ 4-28 โค้ดแสดงการ query ข้อมูลจากฐานข้อมูล (5)

อธิบายภาพที่ 4-28

- บรรทัดที่ 151 รับ url จากหน้าเว็บเพจโดยมีการส่ง parameter material และ culture มาจากหน้าเว็บเพจ
- บรรทัดที่ 152 เป็นการรับ parameter material และ culture จากหน้าเว็บ เพจ
- บรรทัดที่ 153-157 การ query ข้อมูลจาก Database โดยเอาเอาข้อมูลจาก parameter มา query และดึงข้อมูลออกมาเป็น list
- บรรทัดที่ 159 การไปเรียกหน้า material.html มาใช้ และส่งข้อมูลจากตัวแปร record ไปแสดงในหน้าเว็บเพจ

4.2.3 การทำงานในส่วนของเว็บส่วน frontend ของเว็บเพจในการพัฒนาใช้ภาษา Python

```

10. <form class="md-form" action="{{url_for('upload')}}" method="POST"
    enctype="multipart/form-data">
11.     <div id="body">
12.         <div align="center">
13.             
15.         </div>
16.         <div class="out-wrap">
17.             <p class="head">File Upload</p>
18.             <div class="in-wrap">
19.                 <label for="file" class="ui icon button">Open File</label>
20.                 <p class="f-name"></p>
21.                 <input type="file" name="image" id="file" accept=".jpg,
                    .jpeg" required>
22.             </div>
23.             <button class="upload btn" type="submit"><i class="fa fa-
                cloud-upload"></i></button>
24.         </div>
25.     </div>
26. </form>

```

ภาพที่ 4-29 โค้ดแสดงการทำงานส่วนหน้าเว็บเพจ

อธิบายภาพที่ 4-29

- | | |
|-----------------|--|
| บรรทัดที่ 10 | การส่งไฟล์ที่ได้รับไปยัง Flask โดยส่งไปยัง url upload |
| บรรทัดที่ 13-14 | การดึงรูปจาก folder มาแสดงบนหน้าเว็บ |
| บรรทัดที่ 19-23 | การอัปโหลดไฟล์ โดยจะให้รับเฉพาะไฟล์ สกุล .jpg และ .jpeg เท่านั้น |

```

7. <section>
8.     <nav class="navbar navbar-dark bg-dark">
9.         <a class="navbar-brand" href="{{ url_for('hello') }}">HOME</a>
10.    </nav>
11. </section>
12.
13. <section>
14.     <div class="container">
15.         <div class="text-center">
16.             <h1>รูปภาพ</h1>
17.             
20.         </div>
21.     </div>
22. </section>

```

ภาพที่ 4-30 โค้ดแสดงการทำงานหลังจากได้รับข้อมูลจาก backend

อธิบายภาพที่ 4-30

บรรทัดที่ 7-11 การสร้างปุ่มกดกลับไปหน้าโฮมดูลรูปภาพ

บรรทัดที่ 13-21 การแสดงรูปหลังจากอัปโหลดรูปภาพ

```

23. <section>
24.   <div class="container">
25.     </br>
26.     <h2 align="center">รูปที่ใกล้เคียงกับรูปที่ค้นหา</h2>
27.     <table class="table">
28.       <thead class="thead-dark">
29.         <tr>
30.           <th style="width:5%" scope="col">เลขที่</th>
31.           <th style="width:10%" scope="col">รูปภาพ</th>
32.           <th style="width:30%" scope="col">รายละเอียด</th>
33.           <th style="width:30%" scope="col">ศิลปะ</th>
34.           <th style="width:30%" scope="col">วัสดุ</th>
35.         </tr>
36.       </thead>

```

ภาพที่ 4-31 โค้ดการสร้างตารางแสดงผล

อธิบายภาพที่ 4-31

บรรทัดที่ 23-36 การสร้างหัวตารางแสดงผลข้อมูล

```

38. <tbody>
39.   {% for record in rec %}
40.   <!-- row1 -->
41.   <tr>
42.     <th scope="row">{{ record[0] }}</th>
43.     <td>
44.       

```

ภาพที่ 4-32 โค้ดการแสดงผลข้อมูลในตารางแสดงผลข้อมูล

```

45.         </td>
46.         <td class="col-md-2">
47.             <div class="tab">
48.                 {{ record[2] }}
49.             </div>
50.         </td>
51.         <td class="col-md-2">
52.             <div class="tab">
53.                 <a href="/newpage/{{ record[3] }}/{{ record[4]
54.                     }}">{{ record[3] }}</a>
55.             </div>
56.         </td>
57.         <td class="col-md-2">
58.             <div class="tab">
59.                 <a href="/newmaterial/{{ record[4] }}/{{
60.                     record[3] }}">{{ record[4] }}</a>
61.             </div>
        </td>
    </tr>
    {% endfor %}

```

ภาพที่ 4-32 (ต่อ) โค้ดการแสดงผลข้อมูลในตารางแสดงข้อมูล

อธิบายภาพที่ 4-32

- บรรทัดที่ 38-63 รับข้อมูล List มาจาก backend และทำการแสดงผลข้อมูลแต่ละ record โดยจะใช้ for loop
- บรรทัดที่ 54 การค้นหาภาพจาก ศิลปะ และวัสดุ ที่แสดงผลออกมาในตาราง โดยจะส่ง parameter ศิลปะ และวัสดุ เพื่อไปทำการ query ต่อไป
- บรรทัดที่ 59 การค้นหาภาพจาก ศิลปะ และวัสดุ ที่แสดงผลออกมาในตาราง โดยจะส่ง parameter ศิลปะ และวัสดุ เพื่อไปทำการ query ต่อไป

```

13. <section>
14.   <div class="container">
15.     </br>
16.     <h2 align="center">รูปที่แบ่งตามประเภทศิลปะ</h2>
17.     <table class="table">
18.       <thead class="thead-dark">
19.         <tr>
20.           <th style="width:5%" scope="col">เลขที่</th>
21.           <th style="width:10%" scope="col">รูปภาพ</th>
22.           <th style="width:30%" scope="col">รายละเอียด</th>
23.           <th style="width:30%" scope="col">ศิลปะ</th>
24.           <th style="width:30%" scope="col">วัสดุ</th>
25.         </tr>
26.       </thead>

```

ภาพที่ 4-33 โค้ดการแสดงผลข้อมูลของหน้า culture.html

อธิบายรูปที่ 4-33

บรรทัดที่ 13-26 การสร้างหัวตารางแสดงผลข้อมูล

```

27. <tbody>
28.   {% for record in record %}
29.     <tr>
30.       <th scope="row">{{ record[0] }}</th>
31.       <td>
32.         

```

ภาพที่ 4-34 โค้ดการสร้างตารางแสดงผลของข้อมูลของหน้า culture.html

```

34.         <td class="col-md-2">
35.             <div class="tab">
36.                 {{ record[2] }}
37.             </div>
38.         </td>
39.         <td class="col-md-2">
40.             <div class="tab">
41. ....<a href="/newpage/{{ record[3] }}/{{ record[4] }}">{{ record[3] }}</a>
42.             </div>
43.         </td>
44.         <td class="col-md-2">
45.             <div class="tab">
46. <a href="/newmaterial/{{ record[4] }}/{{ record[3] }}">{{ record[4] }}</a>
47.             </div>
48.         </td>
49.         {% endfor %}
50.     </tbody>
51. </table>
52. </div>

</section>

```

ภาพที่ 4-34 (ต่อ) โค้ดการสร้างตารางแสดงผลของข้อมูลของหน้า culture.html

อธิบายภาพที่ 4-34

- บรรทัดที่ 27-55 รับข้อมูล List มาจาก backend และทำการแสดงข้อมูลแต่ละ record โดยจะใช้ for loop
- บรรทัดที่ 43 การค้นหาภาพจาก ศิลปะ และวัสดุ ที่แสดงผลออกมาในตาราง โดยจะส่ง parameter ศิลปะ และวัสดุ เพื่อไปทำการ query ต่อไป
- บรรทัดที่ 48 การค้นหาภาพจาก ศิลปะ และวัสดุ ที่แสดงผลออกมาในตาราง โดยจะส่ง parameter ศิลปะ และวัสดุ เพื่อไปทำการ query ต่อไป


```

13. <section>
14.   <div class="container">
15.     <br>
16.     <h2 align="center">รูปที่แบ่งตามวัสดุ</h2>
17.     <table class="table">
18.       <thead class="thead-dark">
19.         <tr>
20.           <th style="width:5%" scope="col">เลขที่</th>
21.           <th style="width:10%" scope="col">รูปภาพ</th>
22.           <th style="width:30%" scope="col">รายละเอียด</th>
23.           <th style="width:30%" scope="col">ศิลปะ</th>
24.           <th style="width:30%" scope="col">วัสดุ</th>
25.         </tr>
26.       </thead>

```

ภาพที่ 4-35 โค้ดการแสดงผลข้อมูลของหน้า material.html

อธิบายรูปที่ 4-35

บรรทัดที่ 13-26 การสร้างหัวตารางแสดงผลข้อมูล

```

27. <tbody>
28.   {% for record in record %}
29.     <tr>
30.       <th scope="row">{{ record[0] }}</th>
31.       <td>
32.         

```

ภาพที่ 4-36 โค้ดการสร้างตารางแสดงผลของข้อมูลของหน้า material.html

```

34.         <td class="col-md-2">
35.             <div class="tab">
36.                 {{ record[2] }}
37.             </div>
38.         </td>
39.         <td class="col-md-2">
40.             <div class="tab">
41.                 <a href="/newpage/{{ record[3] }}/{{ record[4] }}">{{ record[3] }}</a>
42.             </div>
43.         </td>
44.         <td class="col-md-2">
45.             <div class="tab">
46.                 <a href="/newmaterial/{{ record[4] }}/{{ record[3] }}">{{ record[4] }}</a>
47.             </div>
48.         </td>
49.     {% endfor %}
50. </tbody>
51. </table>
52. </div>

</section>

```

ภาพที่ 4-36 (ต่อ) โค้ดการสร้างตารางแสดงผลของข้อมูลของหน้า material.html

อธิบายภาพที่ 4-36

- บรรทัดที่ 27-55 รับข้อมูล List มาจาก backend และทำการแสดงข้อมูลแต่ละ record โดยจะใช้ for loop
- บรรทัดที่ 43 การค้นหาภาพจาก ศิลปะ และวัสดุ ที่แสดงผลออกมาในตาราง โดยจะส่ง parameter ศิลปะ และวัสดุ เพื่อไปทำการ query ต่อไป
- บรรทัดที่ 48 การค้นหาภาพจาก ศิลปะ และวัสดุ ที่แสดงผลออกมาในตาราง โดยจะส่ง parameter ศิลปะ และวัสดุ เพื่อไปทำการ query ต่อไป

บทที่ 5

ผลของการดำเนินงาน

ผลการดำเนินงานพัฒนาเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย (A Web-based Visual Search for Ancient Artifacts) สามารถสรุปผลการดำเนินงานได้ ดังนี้

5.1 ผลการดำเนินงาน

พัฒนาเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย (A Web-based Visual Search for Ancient Artifacts) สามารถใส่รูปภาพเพื่อค้นหาข้อมูลรูปภาพได้ใกล้เคียง และสามารถนำไปเป็นต้นแบบในการพัฒนาระบบต่อไปได้ ทำการวัดผลความแม่นยำโดยใช้ชุดภาพทดสอบจำนวน 57 ภาพ โดยแบ่งเป็น 2 ชุดย่อย































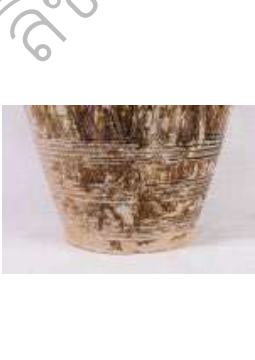









1. ภาพโบราณวัตถุที่มีอยู่ในฐานข้อมูล 20 ภาพ
2. ภาพโบราณวัตถุที่ไม่มีอยู่ในฐานข้อมูล 37 ภาพ

การทดสอบความแม่นยำในการค้นหาภาพโบราณวัตถุ แบ่งเป็น 3 ระดับ ดังนี้


1. ความแม่นยำของโมเดลในการค้นหาภาพโบราณวัตถุที่อันดับ 1 Precision@1 (top-1) เท่ากับ 0.719
2. ความแม่นยำของโมเดลในการค้นหาภาพโบราณวัตถุที่อันดับ 3 Precision@3 (top-3) เท่ากับ 0.824
3. ความแม่นยำของโมเดลในการค้นหาภาพโบราณวัตถุที่อันดับ 5 Precision@5 (top-5) เท่ากับ 0.877

ดังนั้นสรุปได้ว่าค่าเฉลี่ยความแม่นยำในการค้นหาภาพโบราณวัตถุ เท่ากับ Average Precision = 0.807 หรือ 80.70%

ตารางที่ 5-1 แสดงผลลัพธ์ 3 อันดับ จากการนำภาพเข้าทดสอบ

ภาพทดสอบ	อันดับ	ผลลัพธ์ 3 อันดับ																				
	1	<table><tr><th>เลขที่</th><th>รูปภาพ</th><th>รายละเอียด</th><th>ศิลปะ</th><th>วัสดุ</th></tr><tr><td>378076</td><td></td><td>ค่านพม่าหรือพระองค์การพระเป็นเจ้าเสด็จตามนางของเคตต์เป็นเจ้าหญิงเสด็จขึ้นเป็นนางโดยตลอดเหนือจากการพระมีพิธีของพระเศวตมาลาพระนางเป็นสันตกับเป็นแนวโค้งพระเศวตเสด็จขึ้นเป็นแนวพระบาทเสด็จขึ้นเป็นแนวพระบาท</td><td>รัตนโกสินทร์</td><td>ดินเผา</td></tr><tr><td>380314</td><td></td><td>เศียรพระเป็นเจ้าเสด็จตามนางของเคตต์เป็นเจ้าหญิงเสด็จขึ้นเป็นนางโดยตลอดเหนือจากการพระมีพิธีของพระเศวตมาลาพระนางเป็นสันตกับเป็นแนวโค้งพระเศวตเสด็จขึ้นเป็นแนวพระบาทเสด็จขึ้นเป็นแนวพระบาท</td><td>รัตนโกสินทร์</td><td>ดินเผา</td></tr><tr><td>380335</td><td></td><td>เศียรพระเป็นเจ้าเสด็จตามนางของเคตต์เป็นเจ้าหญิงเสด็จขึ้นเป็นนางโดยตลอดเหนือจากการพระมีพิธีของพระเศวตมาลาพระนางเป็นสันตกับเป็นแนวโค้งพระเศวตเสด็จขึ้นเป็นแนวพระบาทเสด็จขึ้นเป็นแนวพระบาท</td><td>รัตนโกสินทร์</td><td>ดินเผา</td></tr></table>	เลขที่	รูปภาพ	รายละเอียด	ศิลปะ	วัสดุ	378076		ค่านพม่าหรือพระองค์การพระเป็นเจ้าเสด็จตามนางของเคตต์เป็นเจ้าหญิงเสด็จขึ้นเป็นนางโดยตลอดเหนือจากการพระมีพิธีของพระเศวตมาลาพระนางเป็นสันตกับเป็นแนวโค้งพระเศวตเสด็จขึ้นเป็นแนวพระบาทเสด็จขึ้นเป็นแนวพระบาท	รัตนโกสินทร์	ดินเผา	380314		เศียรพระเป็นเจ้าเสด็จตามนางของเคตต์เป็นเจ้าหญิงเสด็จขึ้นเป็นนางโดยตลอดเหนือจากการพระมีพิธีของพระเศวตมาลาพระนางเป็นสันตกับเป็นแนวโค้งพระเศวตเสด็จขึ้นเป็นแนวพระบาทเสด็จขึ้นเป็นแนวพระบาท	รัตนโกสินทร์	ดินเผา	380335		เศียรพระเป็นเจ้าเสด็จตามนางของเคตต์เป็นเจ้าหญิงเสด็จขึ้นเป็นนางโดยตลอดเหนือจากการพระมีพิธีของพระเศวตมาลาพระนางเป็นสันตกับเป็นแนวโค้งพระเศวตเสด็จขึ้นเป็นแนวพระบาทเสด็จขึ้นเป็นแนวพระบาท	รัตนโกสินทร์	ดินเผา
เลขที่	รูปภาพ	รายละเอียด	ศิลปะ	วัสดุ																		
378076		ค่านพม่าหรือพระองค์การพระเป็นเจ้าเสด็จตามนางของเคตต์เป็นเจ้าหญิงเสด็จขึ้นเป็นนางโดยตลอดเหนือจากการพระมีพิธีของพระเศวตมาลาพระนางเป็นสันตกับเป็นแนวโค้งพระเศวตเสด็จขึ้นเป็นแนวพระบาทเสด็จขึ้นเป็นแนวพระบาท	รัตนโกสินทร์	ดินเผา																		
380314		เศียรพระเป็นเจ้าเสด็จตามนางของเคตต์เป็นเจ้าหญิงเสด็จขึ้นเป็นนางโดยตลอดเหนือจากการพระมีพิธีของพระเศวตมาลาพระนางเป็นสันตกับเป็นแนวโค้งพระเศวตเสด็จขึ้นเป็นแนวพระบาทเสด็จขึ้นเป็นแนวพระบาท	รัตนโกสินทร์	ดินเผา																		
380335		เศียรพระเป็นเจ้าเสด็จตามนางของเคตต์เป็นเจ้าหญิงเสด็จขึ้นเป็นนางโดยตลอดเหนือจากการพระมีพิธีของพระเศวตมาลาพระนางเป็นสันตกับเป็นแนวโค้งพระเศวตเสด็จขึ้นเป็นแนวพระบาทเสด็จขึ้นเป็นแนวพระบาท	รัตนโกสินทร์	ดินเผา																		
	1	<table><tr><th>เลขที่</th><th>รูปภาพ</th><th>รายละเอียด</th><th>ศิลปะ</th><th>วัสดุ</th></tr><tr><td>124301</td><td></td><td>เก้าอี้ไม้สีน้ำตาล 2 ขา 4 ขา มีลายสลักไม้เป็นรูปดอกไม้ลาย</td><td>รัตนโกสินทร์</td><td>ไม้</td></tr><tr><td>124302</td><td></td><td>มีลายสลักไม้ลาย</td><td>รัตนโกสินทร์</td><td>ไม้</td></tr><tr><td>124298</td><td></td><td>เก้าอี้ไม้สีน้ำตาล 2 ขา 4 ขา มีลายสลักไม้เป็นรูปดอกไม้ลาย</td><td>รัตนโกสินทร์</td><td>ไม้</td></tr></table>	เลขที่	รูปภาพ	รายละเอียด	ศิลปะ	วัสดุ	124301		เก้าอี้ไม้สีน้ำตาล 2 ขา 4 ขา มีลายสลักไม้เป็นรูปดอกไม้ลาย	รัตนโกสินทร์	ไม้	124302		มีลายสลักไม้ลาย	รัตนโกสินทร์	ไม้	124298		เก้าอี้ไม้สีน้ำตาล 2 ขา 4 ขา มีลายสลักไม้เป็นรูปดอกไม้ลาย	รัตนโกสินทร์	ไม้
เลขที่	รูปภาพ	รายละเอียด	ศิลปะ	วัสดุ																		
124301		เก้าอี้ไม้สีน้ำตาล 2 ขา 4 ขา มีลายสลักไม้เป็นรูปดอกไม้ลาย	รัตนโกสินทร์	ไม้																		
124302		มีลายสลักไม้ลาย	รัตนโกสินทร์	ไม้																		
124298		เก้าอี้ไม้สีน้ำตาล 2 ขา 4 ขา มีลายสลักไม้เป็นรูปดอกไม้ลาย	รัตนโกสินทร์	ไม้																		
	2	<table><tr><th>เลขที่</th><th>รูปภาพ</th><th>รายละเอียด</th><th>ศิลปะ</th><th>วัสดุ</th></tr><tr><td>124083</td><td></td><td>ก้นลึก ขอบปากทึบ ลวดสลักไม้ทรงสี่เหลี่ยม ลวดลายของ จานรูป 6 ใบ</td><td>รัตนโกสินทร์</td><td>กระเบื้อง</td></tr><tr><td>124056</td><td></td><td>ขอบปากทึบ มีลวดลาย ลวดสลักไม้ทรงสี่เหลี่ยม ลวดลายของ</td><td>รัตนโกสินทร์</td><td>กระเบื้อง</td></tr><tr><td>124085</td><td></td><td>ก้นลึก ขอบปากทึบ ลวดสลักไม้ทรงสี่เหลี่ยม ลวดลายของ จานรูป 6 ใบ</td><td>รัตนโกสินทร์</td><td>กระเบื้อง</td></tr></table>	เลขที่	รูปภาพ	รายละเอียด	ศิลปะ	วัสดุ	124083		ก้นลึก ขอบปากทึบ ลวดสลักไม้ทรงสี่เหลี่ยม ลวดลายของ จานรูป 6 ใบ	รัตนโกสินทร์	กระเบื้อง	124056		ขอบปากทึบ มีลวดลาย ลวดสลักไม้ทรงสี่เหลี่ยม ลวดลายของ	รัตนโกสินทร์	กระเบื้อง	124085		ก้นลึก ขอบปากทึบ ลวดสลักไม้ทรงสี่เหลี่ยม ลวดลายของ จานรูป 6 ใบ	รัตนโกสินทร์	กระเบื้อง
เลขที่	รูปภาพ	รายละเอียด	ศิลปะ	วัสดุ																		
124083		ก้นลึก ขอบปากทึบ ลวดสลักไม้ทรงสี่เหลี่ยม ลวดลายของ จานรูป 6 ใบ	รัตนโกสินทร์	กระเบื้อง																		
124056		ขอบปากทึบ มีลวดลาย ลวดสลักไม้ทรงสี่เหลี่ยม ลวดลายของ	รัตนโกสินทร์	กระเบื้อง																		
124085		ก้นลึก ขอบปากทึบ ลวดสลักไม้ทรงสี่เหลี่ยม ลวดลายของ จานรูป 6 ใบ	รัตนโกสินทร์	กระเบื้อง																		
	-	<table><tr><th>เลขที่</th><th>รูปภาพ</th><th>รายละเอียด</th><th>ศิลปะ</th><th>วัสดุ</th></tr><tr><td>165703</td><td></td><td>เคลือบสีน้ำตาล มีลายเส้นแนวทแยงเส้นโค้งขึ้นเบื้อง ๑ รอบตัวไห</td><td></td><td></td></tr><tr><td>165767</td><td></td><td>เคลือบสีน้ำตาลเข้ม ไม่ลื่นเงา คือทำเป็นขอมปู่นี มีลายเส้นโค้งขึ้นเบื้อง ๑ รอบตัวไห</td><td></td><td></td></tr><tr><td>165767</td><td></td><td>เคลือบสีน้ำตาลเข้ม ไม่ลื่นเงา คือทำเป็นขอมปู่นี มีลายเส้นโค้งขึ้นเบื้อง ๑ รอบตัวไห</td><td></td><td></td></tr></table>	เลขที่	รูปภาพ	รายละเอียด	ศิลปะ	วัสดุ	165703		เคลือบสีน้ำตาล มีลายเส้นแนวทแยงเส้นโค้งขึ้นเบื้อง ๑ รอบตัวไห			165767		เคลือบสีน้ำตาลเข้ม ไม่ลื่นเงา คือทำเป็นขอมปู่นี มีลายเส้นโค้งขึ้นเบื้อง ๑ รอบตัวไห			165767		เคลือบสีน้ำตาลเข้ม ไม่ลื่นเงา คือทำเป็นขอมปู่นี มีลายเส้นโค้งขึ้นเบื้อง ๑ รอบตัวไห		
เลขที่	รูปภาพ	รายละเอียด	ศิลปะ	วัสดุ																		
165703		เคลือบสีน้ำตาล มีลายเส้นแนวทแยงเส้นโค้งขึ้นเบื้อง ๑ รอบตัวไห																				
165767		เคลือบสีน้ำตาลเข้ม ไม่ลื่นเงา คือทำเป็นขอมปู่นี มีลายเส้นโค้งขึ้นเบื้อง ๑ รอบตัวไห																				
165767		เคลือบสีน้ำตาลเข้ม ไม่ลื่นเงา คือทำเป็นขอมปู่นี มีลายเส้นโค้งขึ้นเบื้อง ๑ รอบตัวไห																				

ตารางที่ 5-1 (ต่อ) แสดงผลลัพธ์ 3 อันดับ จากการนำภาพเข้าทดสอบ

ภาพทดสอบ	อันดับ	ผลลัพธ์ 3 อันดับ
	-	<p>ไม่พบภาพที่ใกล้เคียงกับที่ค้นหา</p>

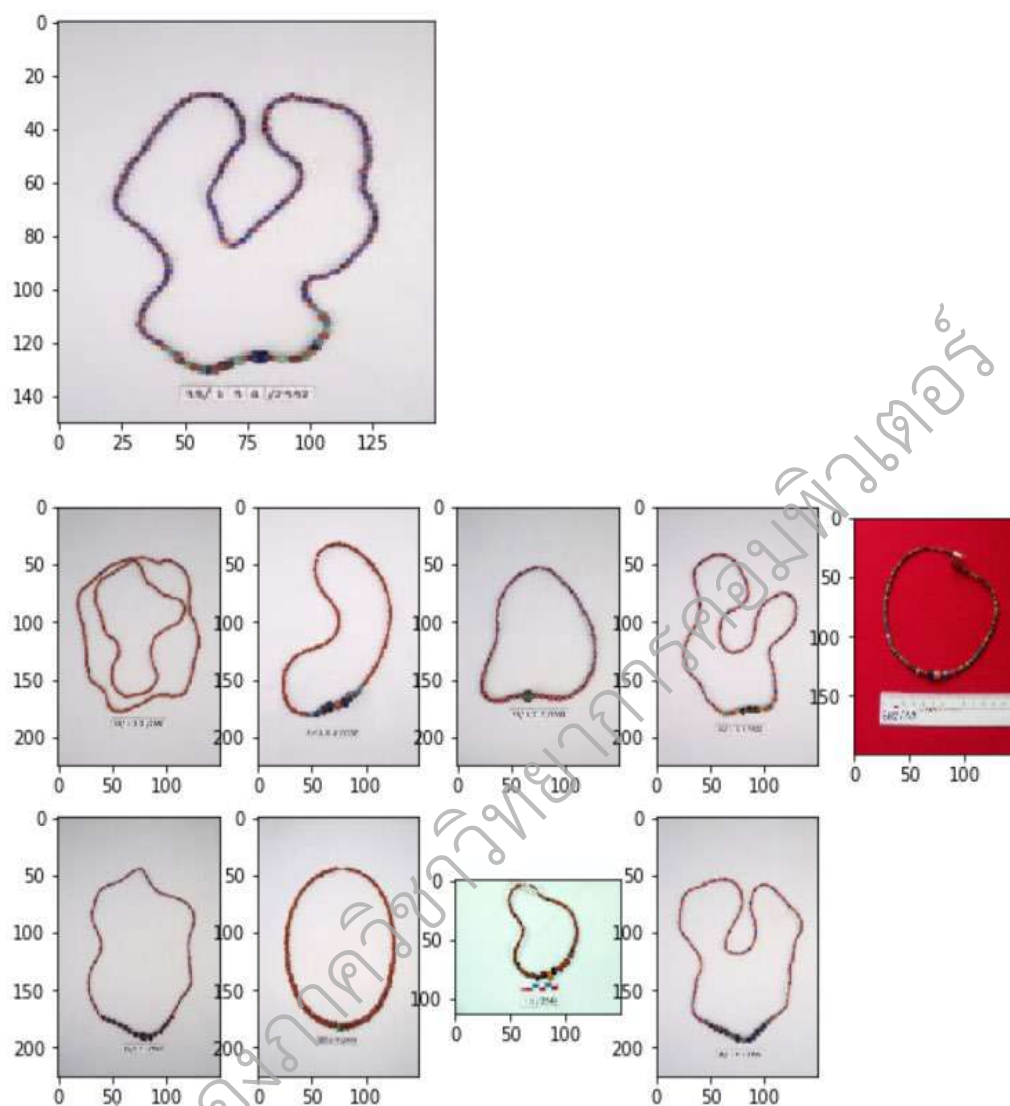
ลิขสิทธิ์ของภาควิชาวิทยาการคอมพิวเตอร์



ภาพที่ 5-1 แสดงผลการนำเข้าชุดภาพทดลอง (1)

อธิบายภาพที่ 5-1

ภาพแสดงผลการนำเข้าชุดภาพทดลอง โดยภาพที่นำเข้าเป็นภาพที่มีอยู่ในฐานข้อมูล ซึ่งผลลัพธ์ออกมาอยู่ที่อันดับ 1 หรือก็คือภาพที่เคยได้รับการเทรนแล้วนั่นเอง



ภาพที่ 5-2 แสดงผลการนำเข้าชุดภาพทดลอง (2)

อธิบายภาพที่ 5-2

ภาพแสดงผลการนำเข้าชุดภาพทดลอง โดยภาพที่นำเข้าเป็นภาพที่ไม่มีอยู่ในฐานข้อมูล ซึ่งผลลัพธ์ออกมาอยู่ที่อันดับ 9 ซึ่งอ้างอิงจากข้อมูลของทางกรมศิลปากร พบว่าเป็นวัตถุชนิดเดียวกัน

5.2 การแสดงผลหน้าเว็บเพจ

5.2.1 หน้าเริ่มต้นเว็บเพจ



ภาพที่ 5-3 หน้าเริ่มต้นเว็บเพจ

อธิบายภาพที่ 5-3

เป็นหน้าเริ่มต้นเว็บเพจ โดยจะมีปุ่มเลือกไฟล์ **OPEN FILE** และปุ่มอัปโหลดไฟล์  เพื่อส่งรูปเข้าไปประมวลผลในโมเดล

5.2.2 หน้าแสดงข้อมูลผลการทำนายรูปภาพใกล้เคียง




ภาพที่ 5-4 หน้าแสดงข้อมูลผลการทำนายรูปภาพใกล้เคียง (1)

อธิบายภาพที่ 5-4







หน้าแสดงข้อมูลที่ได้จากผลการทำนายรูปภาพ 3 อันดับ จะนำ id รูปภาพ รูปภาพ รายละเอียดประเภทศิลปะ และประเภทวัสดุที่สร้าง ออกมาแสดงในรูปแบบตารางแสดงผล และเมื่อผู้ใช้งานต้องการค้นหาประเภทศิลปะ หรือประเภทวัสดุ ก็สามารถคลิกที่ศิลปะ หรือวัสดุ นั้น ๆ ได้

HOME

รูปภาพ



รูปที่ใกล้เคียงกับรูปที่ค้นหา





เลขที่	รูปภาพ	รายละเอียด	ศิลปะ	วัสดุ
102277		ทำเป็นโลหะเคลือบ เรือนมาดอลกาโน มีเขียนและสีราวทำโดยมือคนคนทำชุด	รัตนโกสินทร์	เครื่องปั้นดินเผา
102277		ทำเป็นโลหะเคลือบ เรือนมาดอลกาโน มีเขียนและสีราวทำโดยมือคนคนทำชุด	รัตนโกสินทร์	เครื่องปั้นดินเผา
140690		ทำกรงทำจากสำริด กับเคลือบสี มีรูปสัตว์เป็น ลวดลายแบบขลุ่ยระนาด	รัตนโกสินทร์	เครื่องปั้นดินเผา
140690		ทำกรงทำจากสำริด กับเคลือบสี มีรูปสัตว์เป็น ลวดลายแบบขลุ่ยระนาด	รัตนโกสินทร์	เครื่องปั้นดินเผา
139351		ทำกรงทำจากสำริด กับเคลือบสี มีรูปสัตว์เป็น ลวดลายแบบขลุ่ยระนาด	รัตนโกสินทร์	เครื่องปั้นดินเผา
139351		ทำกรงทำจากสำริด กับเคลือบสี มีรูปสัตว์เป็น ลวดลายแบบขลุ่ยระนาด	รัตนโกสินทร์	เครื่องปั้นดินเผา

ภาพที่ 5-5 หน้าแสดงข้อมูลผลการทำนายรูปภาพใกล้เคียง (2)

อธิบายภาพที่ 5-5

หน้าแสดงข้อมูลที่ได้จากผลการทำนายรูปภาพ 3 อันดับ จะนำ id รูปภาพ รูปภาพ รายละเอียดประเภทศิลปะ และประเภทวัสดุที่สร้าง ออกมาแสดงในรูปแบบตารางแสดงผล และเมื่อผู้ใช้งานต้องการค้นหาประเภทศิลปะ หรือประเภทวัสดุ ก็สามารถคลิกที่ศิลปะ หรือวัสดุ นั้น ๆ ได้

5.2.3 หน้าแสดงข้อมูลการค้นหารูปภาพจากประเภทศิลปะ และวัสดุในการสร้าง

HOME				
รูปที่แบ่งตามยุคสมัย				
เลขที่	รูปภาพ	รายละเอียด	ศิลปะ	วัสดุ
102133		เขียนลายพรรณพฤษชาติ	รัตนโกสินทร์	เครื่องถ้วย
102133		เขียนลายพรรณพฤษชาติ	รัตนโกสินทร์	เครื่องถ้วย
102133		เขียนลายพรรณพฤษชาติ	รัตนโกสินทร์	เครื่องถ้วย
102210		เขียนลายพรรณพฤษชาติ	รัตนโกสินทร์	เครื่องถ้วย

ภาพที่ 5-6 หน้าแสดงข้อมูลการค้นหารูปภาพจากประเภทศิลปะ และวัสดุในการสร้าง

อธิบายภาพที่ 5-6

หน้าแสดงข้อมูลการค้นหารูปภาพจากประเภทศิลปะ และวัสดุในการสร้าง ต่อจากภาพที่ 5-4 และ 5-6 เมื่อผู้ใช้คลิกที่ศิลปะ หรือวัสดุ นั้น ๆ ระบบจะทำการค้นหาข้อมูลศิลปะ และวัสดุที่เหมือนกับที่ผู้ใช้เลือก และนำมาแสดงผล 10 อันดับ

บทที่ 6

บทสรุปและแนวทางการพัฒนาต่อ

หลังจากสิ้นสุดการสร้างเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย (A Web-based Visual Search for Ancient Artifacts) มีผลของการดำเนินงาน ปัญหาที่พบในขณะพัฒนาระบบ รวมถึงข้อเสนอแนะในการนำ เาาระบบไปพัฒนาต่อยอดในอนาคต สามารถสรุปได้ดังนี้

6.1 สรุปผลการดำเนินงาน

ผลจากการสร้างเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย (A Web-based Visual Search for Ancient Artifacts) โดยสามารถสรุปผลการดำเนินงานได้ ดังนี้

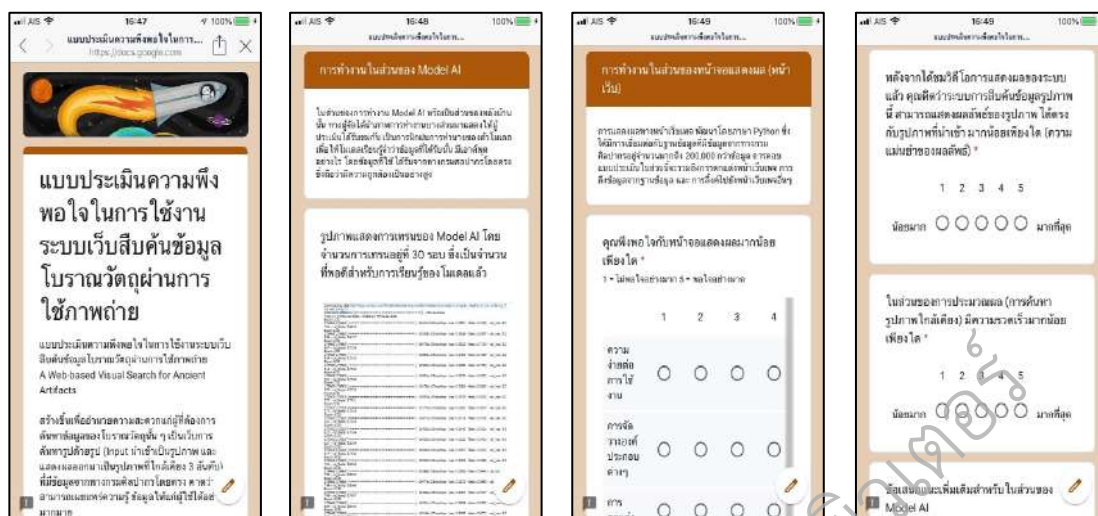
6.1.1 การแสดงผลการทำนายของโมเดล AI

- สามารถแสดงผลการทำนายได้อย่างแม่นยำ และมีประสิทธิภาพ
- สามารถแสดงผลการทำนายได้อย่างรวดเร็ว
- สามารถทำการค้นหาโดยใช้ดัชนีภาพได้อย่างรวดเร็ว และถูกต้อง

6.1.2 การแสดงผลของหน้าเว็บเพจ

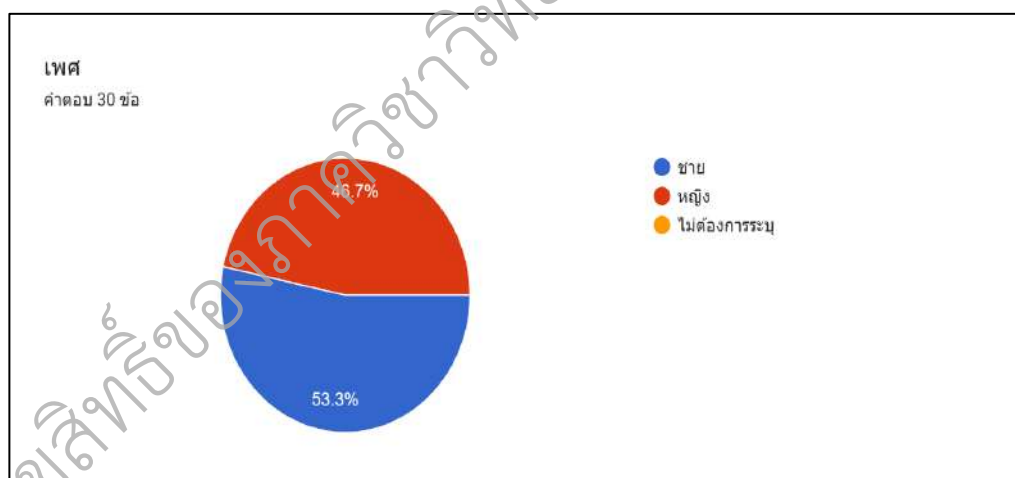
- สามารถทำการอัปโหลดรูปภาพเพื่อนำไปประมวลผลต่อได้
- สามารถแสดงผลข้อมูลจากการใส่ภาพเข้าไป ได้อย่างถูกต้องและแม่นยำ
- สามารถแสดงข้อมูล ของศิลปะ และวัสดุ ตามที่ต้องการค้นหาได้

ทั้งนี้ทางผู้จัดทำได้จัดทำแบบประเมิน สำหรับบุคคลทั่วไป ที่ได้ทดลองระบบเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย ดังภาพที่ 6-1 โดยมีผู้ทดลองระบบเป็นจำนวน 30 คน



ภาพที่ 6-1 ตัวอย่างแบบประเมินความพึงพอใจในการใช้งานระบบ

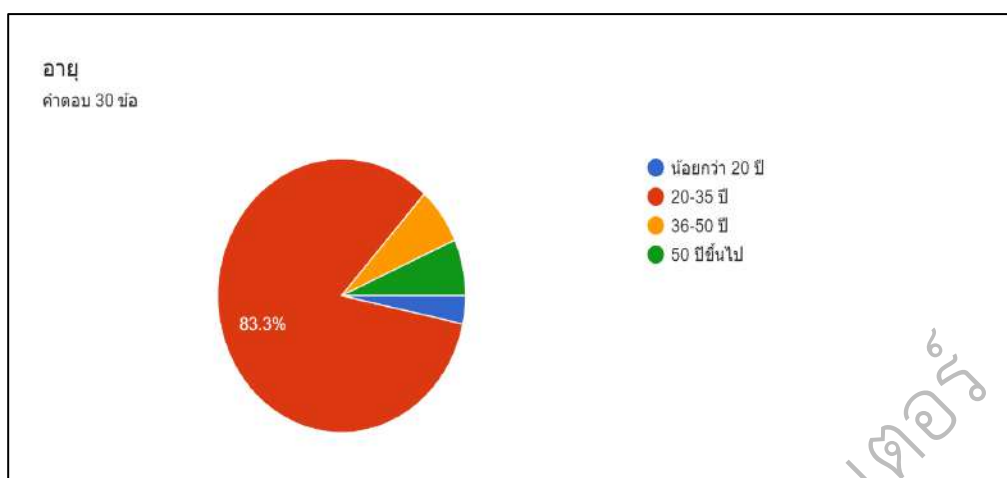
ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการถ่ายภาพ มีดังนี้



ภาพที่ 6-2 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (1)

อธิบายภาพที่ 6-2

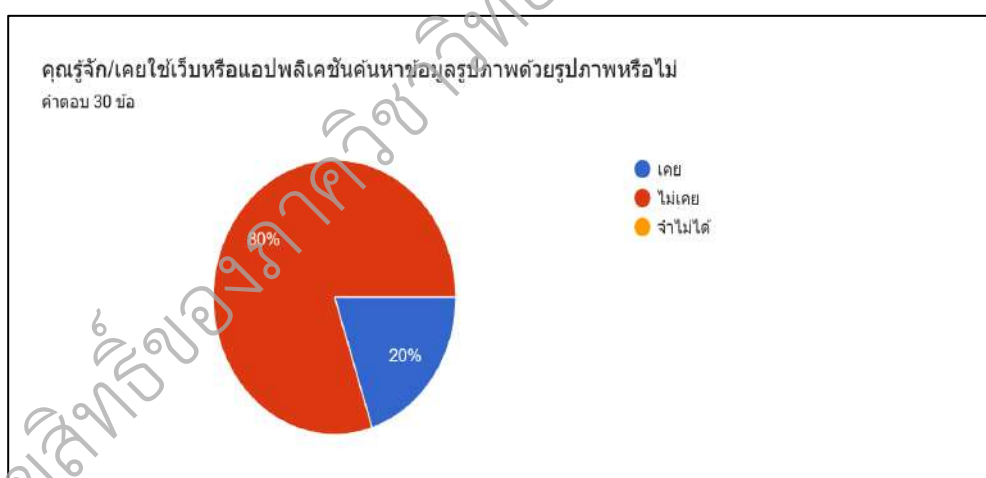
จากผู้ทดลองจำนวน 30 คน แบ่งเป็น ชาย 16 คน หรือ 53.3% และ หญิง 14 คน หรือ 46.7%



ภาพที่ 6-3 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (2)

อธิบายภาพที่ 6-3

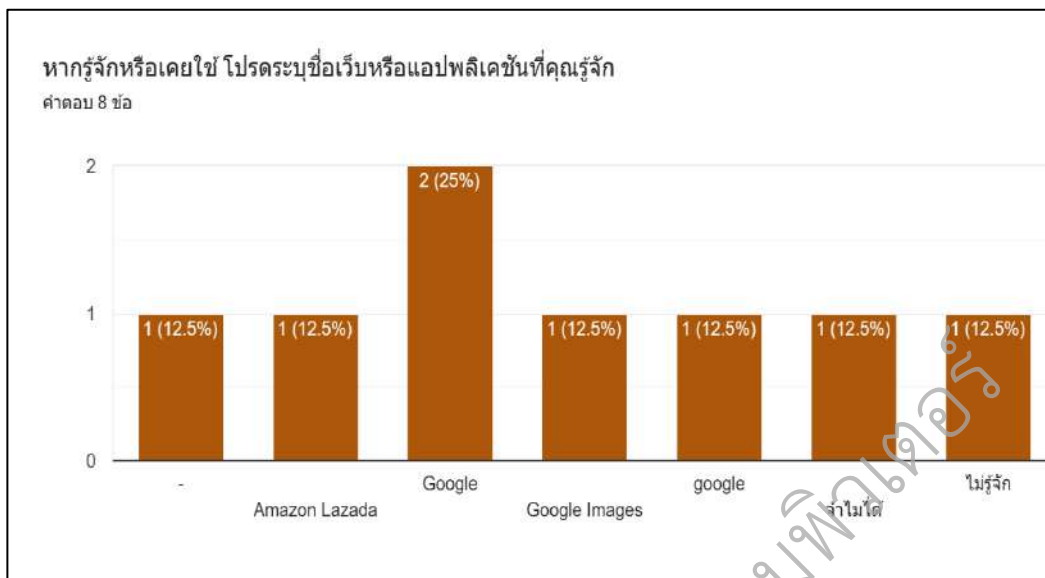
จากผู้ทดลองจำนวน 30 คน แบ่งตามช่วงอายุ ได้แก่ น้อยกว่า 20 ปี 1 คน หรือ 3.3%, 20-35 ปี 25 คน หรือ 83.3%, 36-50 ปี 2 คน หรือ 6.7% และ 50 ปีขึ้นไป 2 คน หรือ 6.7%



ภาพที่ 6-4 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (3)

อธิบายภาพที่ 6-4

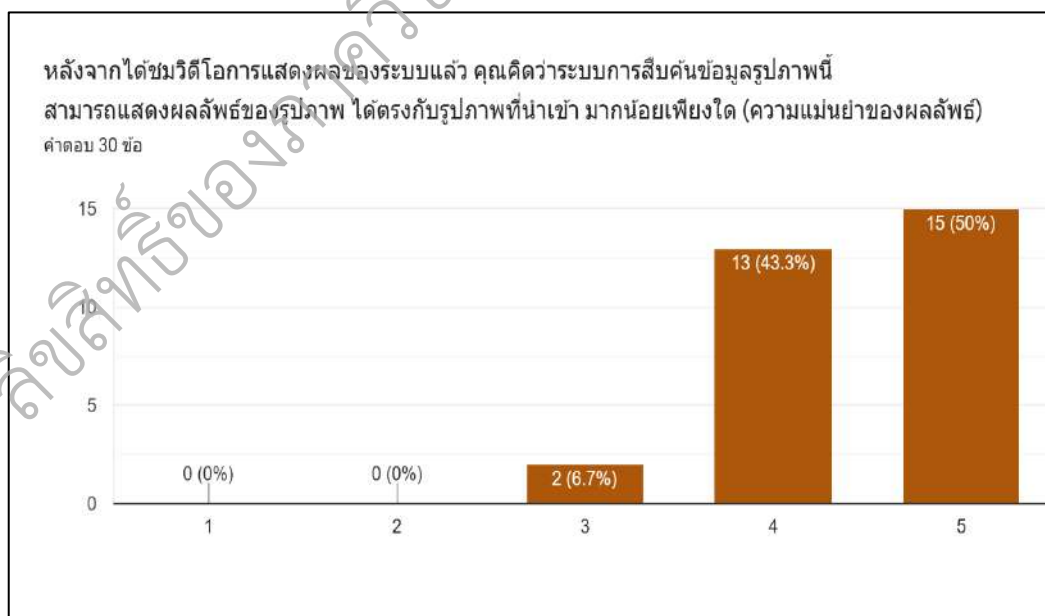
จากผู้ทดลองจำนวน 30 คน มีผู้ที่รู้จัก หรือเคยใช้เว็บไซต์หรือแอปพลิเคชันค้นหาข้อมูลรูปภาพด้วยรูปภาพ 6 คน หรือ 20% และ ไม่รู้จัก หรือไม่เคยใช้เว็บไซต์หรือแอปพลิเคชันค้นหาข้อมูลรูปภาพด้วยรูปภาพ 24 คน หรือ 80%



ภาพที่ 6-5 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (4)

อธิบายภาพที่ 6-5

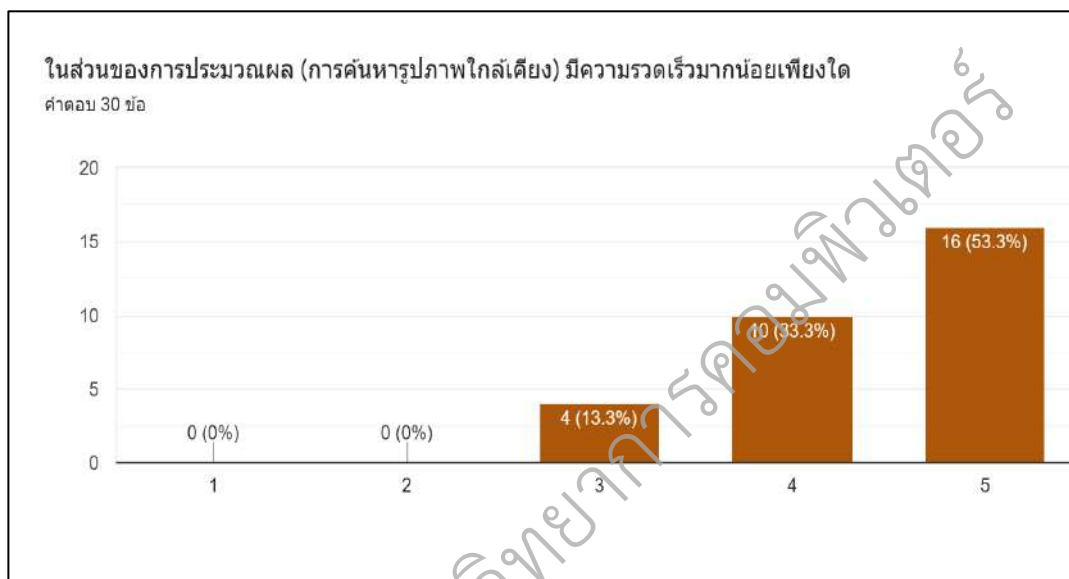
จากผู้ทดลองจำนวน 6 คน หรือ 20% ที่รู้จัก หรือเคยใช้เว็บ หรือแอปพลิเคชันค้นหาข้อมูลรูปภาพด้วยรูปภาพ เช่น Amazon, Lazada, Google และ Google Images



ภาพที่ 6-6 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (5)

อธิบายภาพที่ 6-6

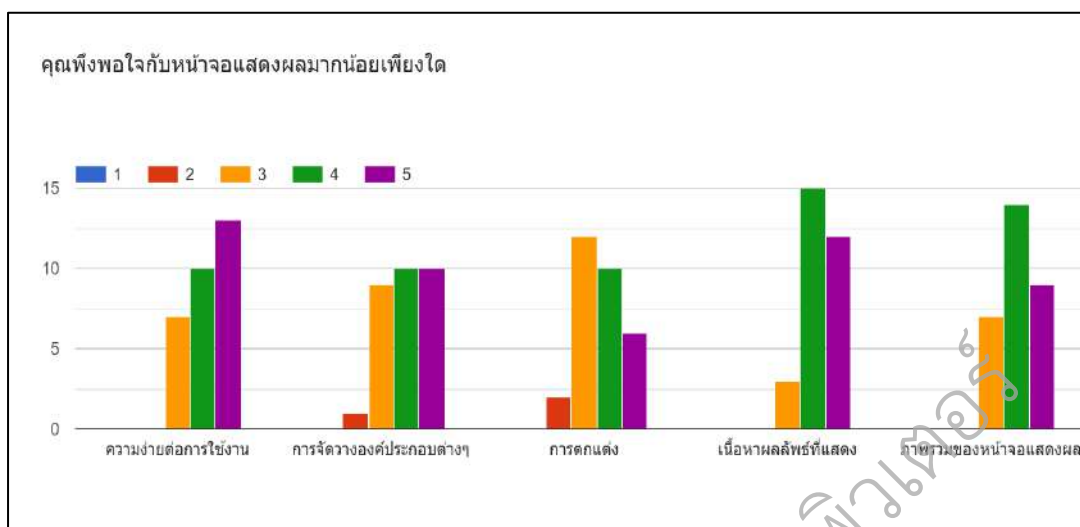
สำหรับในการประเมินการทำงานของโมเดล AI ส่วนของความแม่นยำ จากผู้ทดลองจำนวน 30 คน มีผู้พึงพอใจมากที่สุด (ระดับ 5) จำนวน 15 คน หรือ 50%, พึงพอใจ (ระดับ 4) จำนวน 13 คน หรือ 43.3% และ พึงพอใจปานกลาง (ระดับ 3) 2 คน หรือ 6.7%



ภาพที่ 6-7 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (6)

อธิบายภาพที่ 6-7

สำหรับในการประเมินการทำงานของโมเดล AI ส่วนของความเร็วในการค้นหา จากผู้ทดลองจำนวน 30 คน มีผู้พึงพอใจมากที่สุด (ระดับ 5) จำนวน 16 คน หรือ 53.3%, พึงพอใจ (ระดับ 4) จำนวน 10 คน หรือ 33.3% และ พึงพอใจปานกลาง (ระดับ 3) 4 คน หรือ 13.3%

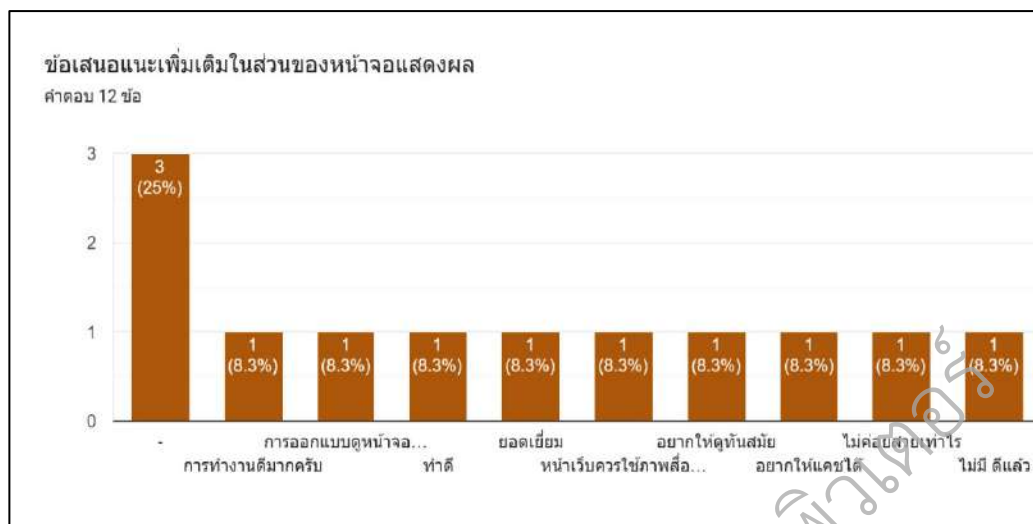


ภาพที่ 6-8 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (7)

อธิบายภาพที่ 6-8

สำหรับในการประเมินการทำงานของหน้าจอแสดงผล ถูกแบ่งออกเป็น 5 ส่วน จากผู้ทดลองจำนวน 30 คน ดังนี้

- ส่วนที่ 1 ความง่ายต่อการใช้งาน มีผู้พึงพอใจมากที่สุด (ระดับ 5) จำนวน 13 คน, พึงพอใจมาก (ระดับ 4) จำนวน 10 คน และ พึงพอใจปานกลาง (ระดับ 3) 7 คน
- ส่วนที่ 2 การจัดวางองค์ประกอบต่าง ๆ มีผู้พึงพอใจมากที่สุด (ระดับ 5) จำนวน 10 คน, พึงพอใจมาก (ระดับ 4) จำนวน 10 คน, พึงพอใจปานกลาง (ระดับ 3) 9 คน และ พึงพอใจน้อย (ระดับ 2) 1 คน
- ส่วนที่ 3 การตกแต่ง มีผู้พึงพอใจมากที่สุด (ระดับ 5) จำนวน 6 คน, พึงพอใจมาก (ระดับ 4) จำนวน 10 คน, พึงพอใจปานกลาง (ระดับ 3) 12 คน และ พึงพอใจน้อย (ระดับ 2) 2 คน
- ส่วนที่ 4 เนื้อหาผลลัพธ์ที่แสดง มีผู้พึงพอใจมากที่สุด (ระดับ 5) จำนวน 12 คน, พึงพอใจมาก (ระดับ 4) จำนวน 15 คน และ พึงพอใจปานกลาง (ระดับ 3) 3 คน
- ส่วนที่ 5 ภาพรวมของหน้าจอแสดงผล มีผู้พึงพอใจมากที่สุด (ระดับ 5) จำนวน 9 คน, พึงพอใจมาก (ระดับ 4) จำนวน 14 คน และ พึงพอใจปานกลาง (ระดับ 3) 7 คน



ภาพที่ 6-9 ผลลัพธ์จากแบบประเมินความพึงพอใจในการใช้งานระบบ (8)

อธิบายภาพที่ 6-9

สำหรับในการประเมินการทำงานของหน้าจอแสดงผล จากผู้ทดลองจำนวน 30 คน มีข้อเสนอแนะเพิ่มเติม ดังภาพที่ 6-9

6.2 ปัญหาและข้อจำกัดในการดำเนินงาน

6.2.1 ข้อมูลรูปภาพที่ได้รับ มีปัญหาเล็กน้อยบางภาพ ดังนี้



ภาพที่ 6-10 ตัวอย่างข้อมูลรูปภาพที่มีปัญหา (1)

อธิบายภาพที่ 6-10

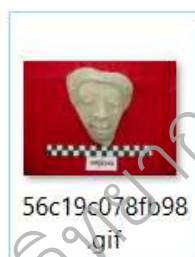
ข้อมูลรูปภาพที่ได้รับมีปัญหาคือ รูปภาพบางภาพมีขนาดเล็กเกินไป ดังภาพที่ 6-10 กล่าวคือขนาดของวัตถุในภาพมีขนาดเล็กเกินไปกว่าที่ตัวโมเดล AI จะประมวลผล หรือทำการทำนายผลลัพธ์ออกมา



ภาพที่ 6-11 ตัวอย่างข้อมูลรูปภาพที่มีปัญหา (2)

อธิบายภาพที่ 6-11

ข้อมูลรูปภาพที่ได้รับมีปัญหาคือ รูปภาพบางภาพมีขนาดเล็กเกินไป ดังภาพที่ 6-11 ในกรณีนี้กล่าวคือ ขนาดขององค์ประกอบรอบ ๆ วัตถุมีขนาดใหญ่กว่าขนาดของวัตถุในภาพซึ่ง มีขนาดเล็กเกินไปกว่าที่ตัวโมเดล AI จะประมวลผล หรือทำการทำนายผลลัพธ์ออกมา



ภาพที่ 6-12 ตัวอย่างข้อมูลรูปภาพที่มีปัญหา (3)

อธิบายภาพที่ 6-12

ข้อมูลรูปภาพที่ได้รับมีปัญหาคือ รูปภาพบางภาพมีไฟล์นามสกุลที่ไม่ตรงกับวัตถุประสงค์ของการประมวลผลของโมเดล AI ทำให้ไม่สามารถนำมาใช้ประมวลผลในโมเดล AI ได้

6.3 แนวทางในการพัฒนาต่อไปในอนาคต

- 6.3.1 เพิ่มข้อมูลรูปภาพนำเข้า สำหรับการเทรนอย่างมีประสิทธิภาพมากยิ่งขึ้น
- 6.3.2 สามารถต่อยอดโมเดล โดยการเพิ่ม Output เพื่อนำไปเทรนให้เกิดการเรียนรู้ และได้ผลลัพธ์ที่แม่นยำมากยิ่งขึ้น
- 6.3.3 สามารถนำโมเดลที่ใช้ไปดัดแปลงเพื่อต่อยอดกับระบบอื่น ๆ เช่น ระบบสืบค้นข้อมูล แยกประเภทสุนัข หรือแมว และระบบสืบค้นข้อมูลแยกประเภทมนุษย์ เป็นต้น
- 6.3.4 เพิ่มฟังก์ชันในหน้าเว็บเพจให้มีการแสดงผลอื่น ๆ เช่น Timeline ยุคสมัยของโบราณวัตถุ

บรรณานุกรม

- [1] คอมพิวเตอร์วิทัศน์. (2551). [ออนไลน์]. เข้าถึงได้จาก : <https://www.slideshare.net/guest492661/com-vision-presentation>. (วันที่ค้นข้อมูล : 4 พฤศจิกายน 2562).
- [2] STEM Learning & Teaching Approach. (2562). [ออนไลน์]. เข้าถึงได้จาก : http://doc.cdn.simplesite.com/d/fc/48/284852681709209852/53fc6587-948e-4dc3-b541-deb866f9edab/STEM_docs.pdf. (วันที่ค้นข้อมูล : 4 พฤศจิกายน 2562).
- [3] Deep Learning คืออะไร. (2560). [ออนไลน์]. เข้าถึงได้จาก : <https://medium.com/@athivvat/deep-learning-คืออะไร-785e16d01773>. (วันที่ค้นข้อมูล : 4 พฤศจิกายน 2562).
- [4] Articles Under. (2560). [ออนไลน์]. เข้าถึงได้จาก : <https://deepnotes.io/category/cnn-series>. (วันที่ค้นข้อมูล : 4 พฤศจิกายน 2562).
- [5] Locality-sensitive hashing (LSH). (2561). [ออนไลน์]. เข้าถึงได้จาก : https://www.researchgate.net/figure/Locality-sensitive-hashing-LSH_fig4_314300245. (วันที่ค้นข้อมูล : 4 พฤศจิกายน 2562).
- [6] Introduction to Locality-Sensitive Hashing. (2561). [ออนไลน์]. เข้าถึงได้จาก : https://unboxresearch.com/articles/lsh_post1.html. (วันที่ค้นข้อมูล : 4 พฤศจิกายน 2562).
- [7] EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. (2562). [ออนไลน์]. เข้าถึงได้จาก : <https://www.profillic.com/paper/arxiv:1905.11946>. (วันที่ค้นข้อมูล : 4 พฤศจิกายน 2562).



คพ. 04

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ

หนังสือรับรองการทดสอบโครงการพิเศษ

วันที่ 19 เดือน กุมภาพันธ์ พ.ศ. 25 63

เรื่อง หนังสือรับรองการทดสอบโครงการพิเศษ

เรียน หัวหน้าภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ

สิ่งที่แนบมาด้วย 1. เอกสารประกอบการสอบจำนวน 5 เล่ม

ตามที่ข้าพเจ้า (นาย,นาง,นางสาว) สนั่น นันท์ คำวัน รหัสประจำตัว

5	9	0	4	0	6	2	6	3	0	2	4	1
---	---	---	---	---	---	---	---	---	---	---	---	---

(นาย,นาง,นางสาว) สนธิ์ธร รอดกลาง รหัสประจำตัว

5	9	0	4	0	6	2	6	3	0	4	5	4
---	---	---	---	---	---	---	---	---	---	---	---	---

นักศึกษาสาขาวิชาวิทยาการคอมพิวเตอร์ (CS) ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์
ได้จัดทำโครงการพิเศษเรื่อง เว็บไซต์ค้นข้อมูลโบราณวัตถุสำหรับการใช้ภาพจากให้กับหน่วยงาน กรมศิลปากร ตั้งแต่ภาคเรียนที่ 1 นั้นบัดนี้โครงการดังกล่าวได้ดำเนินการเสร็จและพร้อมนำไปให้หน่วยงานทดสอบตั้งแต่วันที่ 22 กุมภาพันธ์ 2563
จึงเรียนมาเพื่อ โปรดพิจารณาอนุญาตลงชื่อ สนั่น นันท์ คำวัน ผู้เสนอโครงการ(นาย สนั่น นันท์ คำวัน)หมายเลขโทรศัพท์มือถือ 0924153555ลงชื่อ สนธิ์ธร รอดกลาง ผู้เสนอโครงการ(นาย สนธิ์ธร รอดกลาง)หมายเลขโทรศัพท์มือถือ 088 6073503

ความเห็นของอาจารย์ที่ปรึกษา

เห็นชอบในสิ่งที่แนบมาไปทดสอบ และแก้ไขตามความเห็นของอาจารย์ที่ปรึกษาลงชื่อ [Signature] อาจารย์ที่ปรึกษาโครงการพิเศษ(นาย 2/3-นายพร)ลงชื่อ [Signature] อาจารย์ที่ปรึกษาโครงการพิเศษ(นาย 2/3-นายพร)19 / 2 / 63

- หมายเหตุ 1. ให้ส่งหนังสือฉบับนี้เข้าแฟ้มไว้ที่ธุรการภาควิชาและให้นักศึกษาสำเนาเก็บไว้
2. ให้แนบสำเนาหนังสือฉบับนี้ประกอบการยื่นขอสอบโครงการพิเศษตามกำหนดเวลา
3. ในกรณีที่มียาจารย์ที่ปรึกษามากกว่า 1 คน ให้อาจารย์ทุกท่านลงนามร่วมกัน



หนังสือรับรองการใช้ประโยชน์จากผลงานวิจัยหรืองานสร้างสรรค์

คณะวิทยาศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

1) ข้อมูลโครงการวิจัยหรืองานสร้างสรรค์

ชื่อโครงการวิจัยหรืองานสร้างสรรค์

เว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย

A Web-based Visual Search for Ancient Artifacts

ระยะเวลาการดำเนินโครงการ 7 เดือน

วันเริ่มต้นโครงการ 22 / กรกฎาคม / 2562 วันสิ้นสุดโครงการ 25 / กุมภาพันธ์ / 2563

งบประมาณ

แหล่งทุน

2) วัตถุประสงค์ของโครงการ

.....เพื่อพัฒนาเว็บแอปพลิเคชัน สำหรับการสืบค้นข้อมูล โดยอาศัยเทคนิคทางด้าน Machine Learning.....

3) ข้อมูลคณะผู้วิจัย

หัวหน้าโครงการ

ชื่อหัวหน้าโครงการนาย สติณัฏ์ ประสมพันธ์.....

ตำแหน่งทางวิชาการผู้ช่วยศาสตราจารย์.....

หน่วยงานต้นสังกัด.....

(ภาควิชา)วิทยาการคอมพิวเตอร์และสารสนเทศ.....

(คณะ)วิทยาศาสตร์ประยุกต์.....

ผู้ร่วมโครงการ

1. นาย ธนานนท์ คำวัน

2. นาย สหวัชร รอดกลาง

4) ข้อมูลหน่วยงานที่นำผลงานวิจัยหรืองานสร้างสรรค์ไปใช้ประโยชน์

ชื่อหน่วยงานกรมศิลปากร.....

ชื่อผู้บริหารระดับสูงตำแหน่ง.....

สถานที่ตั้ง.....

โทรศัพท์โทรสาร.....

5) การนำผลงานวิจัยหรืองานสร้างสรรค์ไปใช้ประโยชน์ (กรุณาเลือกตอบเพียง 1 หัวข้อ)

- ☒ การใช้ประโยชน์ในเชิงสาธารณะ (การพัฒนาคุณภาพชีวิตและเศรษฐกิจของประชาชน)
- ☐ การใช้ประโยชน์ในเชิงนโยบาย (การประกาศใช้กฎหมาย กำหนดมาตรการ/กฎเกณฑ์ ขององค์กร)
- ☐ การใช้ประโยชน์ในเชิงพาณิชย์ (การพัฒนาซึ่งก่อให้เกิดรายได้ หรือการเพิ่มประสิทธิภาพการผลิต)
- ☐ การใช้ประโยชน์ทางอ้อม (สร้างคุณค่าทางจิตใจ)

6) ช่วงเวลาที่นำงานวิจัยไปใช้ประโยชน์ 25 / กุมภาพันธ์ / 2563

ขอรับรองว่า (หน่วยงานที่ใช้ประโยชน์)กรมศิลปากร..... ได้นำผลงานวิจัยหรือ งาน
สร้างสรรค์จากโครงการเว็บสืบค้นข้อมูลโบราณวัตถุผ่านการใช้ภาพถ่าย.....ไปใช้จนก่อให้เกิดประโยชน์ได้จริง
อย่างชัดเจนตามวัตถุประสงค์ของโครงการ

* แนบเอกสารหลักฐานที่แสดงการใช้ประโยชน์จากโครงการฯ ทำหนังสือรับรองการใช้ประโยชน์ เช่น
ภาพถ่าย รายงานผลการใช้ประโยชน์จากผลงานวิจัย สื่อประชาสัมพันธ์การใช้ประโยชน์ผลงานวิจัย *

ผู้ดำเนินโครงการ

ผู้ใช้ประโยชน์จากโครงการ

ลงชื่อ

ลงชื่อ

(ผู้ช่วยศาสตราจารย์ สถิตย์ ประสมพันธ์)

(ผู้ช่วยศาสตราจารย์ สถิตย์ ประสมพันธ์)

หัวหน้าโครงการ

ตำแหน่งผู้ช่วยศาสตราจารย์.....

ประทับตราของหน่วยงานด้วย