



การรู้จำลายมือเขียนอักษรล้านนาด้วยเทคนิคการรวมแบบจำลองโหวต

Lanna Handwriting Recognition
using Vote Ensemble Technique

ประกิต หมิ่นนามหน่อ

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีดิจิทัล

มหาวิทยาลัยราชภัฏเชียงใหม่

ปีการศึกษา 2566

(ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏเชียงราย)

การรู้จำลายมือเขียนอักษรล้านนาด้วยเทคนิคการรวมแบบจำลองโหวต



การรู้จำลายมือเขียนอักษรล้านนาด้วยเทคนิคการรวมแบบจำลองโหวต

Lanna Handwriting Recognition
using Vote Ensemble Technique

ประกิต หมิ่นนามหน่อ

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีดิจิทัล
มหาวิทยาลัยราชภัฏเชียงราย
ปีการศึกษา 2566

บทคัดย่อ

ชื่องานวิจัย : การรู้จำลายมือเขียนอักษรล้านนาด้วยเทคนิคการรวมแบบจำลองโหวต
ชื่อผู้วิจัย : นายประกิต หมิ่นนามหน่อ
สาขาวิชา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2566
อาจารย์ที่ปรึกษา : ผศ.ดร.ธนาวุฒิ ธนวาณิชย์

ภาษาล้านนาเป็นมรดกทางภาษาของชาวล้านนา บันทึกไว้ในสื่อหลายชนิด เช่น หนังสือใบลาน และศิลาจารึก ประกอบด้วยข้อมูลทางประวัติศาสตร์ และข้อมูลที่มีความสำคัญ ซึ่งหนังสือใบลาน และศิลาจารึกผู้ฟังได้ถ่ายเมื่อขาดการดูแลรักษาเป็นเวลานาน เอกสารจึงเกิดความเสียหาย ทำให้ส่งผลกระทบต่อลูกหลานชาวล้านนาในรุ่นหลังที่ยากต่อการศึกษาค้นคว้าข้อมูลทางประวัติศาสตร์ บทความนี้นำเสนอการใช้เทคนิคการรวมแบบจำลองโหวต ด้วยการพัฒนาแบบจำลองในการรู้จำลายมือเขียนอักษรล้านนา โดยทดสอบกับชุดข้อมูลภาพเอกสารจากเครื่องพิมพ์ ชุดข้อมูลภาพเอกสารลายมือเขียนบนกระดาษสีขาว และชุดข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลาน จำนวน 3,408 ภาพตัวอักษร แบบจำลองฮิสโตแกรมแมทชิ่ง (Histogram Matching 16x16) ได้ผลความแม่นยำแบบ F1-Score 87.59% ในขณะที่แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) ได้ผลความแม่นยำแบบ F1-Score 93.84% และแบบจำลองการถดถอยโลจิสติก (Logistic Regression) ได้ผลความแม่นยำแบบ F1-Score 94.53% เมื่อนำแบบจำลองแบบจำลองฮิสโตแกรมแมทชิ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติกรวมกันด้วยเทคนิคการรวมแบบจำลองโหวต ได้แบบจำลองการรวมแบบจำลองโหวต (Vote Ensemble 16x16) มีผลความแม่นยำแบบ F1-Score 95.31% ซึ่งมีค่าความแม่นยำแบบ F1-Score มากที่สุด แบบจำลองการรวมแบบจำลองโหวตทำให้มีผลความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนาเพิ่มขึ้น

กิตติกรรมประกาศ

การพัฒนาแบบจำลองการรู้จำลายมือเขียนอักษรล้านนาด้วยเทคนิคการรวมแบบจำลองโหวตจะสำเร็จลุล่วงไปด้วยดีไม่ได้ หากไม่ได้รับความกรุณาจากบุคคลหลายท่าน ข้าพเจ้าขอขอบคุณบุคคลสำคัญตั้งที่จะกล่าวถึง ดังต่อไปนี้เป็นอย่างยิ่งที่ช่วยให้คำแนะนำแก้ไขปัญหา จนช่วยให้ข้าพเจ้าทำการพัฒนาแบบจำลองการรู้จำลายมือเขียนอักษรล้านนาด้วยเทคนิคการรวมแบบจำลองโหวตสำเร็จลุล่วงไปด้วยดี ซึ่งได้แก่ บิดามารดา ที่ให้การสนับสนุน ตลอดจน ผศ.ดร.ธนาวุฒิ ธนวานิชย์ อาจารย์ที่ปรึกษา ผศ.ดร.ภูมิพงษ์ ดวงตั้ง ผศ.ดร.กฤตกรณ์ ศรีวันนา ผศ.ดร.มยุร ไยบัวเทศ ผศ.กมล บุญล้อม และอาจารย์ อธิคม ศิริ ที่คอยสั่งสอนวิชาเพื่อนำไปใช้งานการทำงาน ให้คำปรึกษา และคำแนะนำในการพัฒนาแบบจำลองการรู้จำลายมือเขียนอักษรล้านนาด้วยเทคนิคการรวมแบบจำลองโหวต ตั้งแต่เริ่มต้นจนเสร็จสิ้นกระบวนการทำงาน

สำหรับคุณงามความดีที่เกิดจากการพัฒนาแบบจำลองการรู้จำลายมือเขียนอักษรล้านนาด้วยเทคนิคการรวมแบบจำลองโหวต ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รัก และเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาท วิชาความรู้ และถ่ายทอดประสบการณ์ที่ดีต่อข้าพเจ้า ช่วยเหลือ และแนะนำสิ่งที่ดีให้กับข้าพเจ้า

ประกิต หมิ่นนามหน่อ

ตุลาคม 2566

สารบัญ

	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญตาราง	จ
สารบัญภาพ	ฉ
บทที่	
1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
1.4 ขอบเขตของการวิจัย	1
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	2
2.1 ตัวอักษรภาษาล้านนา	2
2.2 การประมวลผลภาพเอกสาร	3
2.3 ฮิสโตแกรม	7
2.4 การเรียนรู้ของเครื่อง	9
2.5 การเรียนรู้เชิงลึก	12
2.6 การประเมินประสิทธิภาพแบบจำลอง	18
2.7 งานวิจัยที่เกี่ยวข้อง	20
3 วิธีดำเนินการวิจัย	22
3.1 การเตรียมข้อมูล	23
3.2 การประมวลผลข้อมูลล่วงหน้า	25
3.3 การแบ่งชุดข้อมูล	29
3.4 การสร้างแบบจำลอง	30

สารบัญ (ต่อ)

บทที่	หน้า
3	
3.5	35
4	36
4.1	36
4.2	39
4.3	40
5	42
5.1	42
5.2	43
5.3	44
บรรณานุกรม	45
ภาคผนวก	47
ภาคผนวก ก	48
ประวัติผู้วิจัย	49

สารบัญตาราง

ตารางที่	หน้า	
3.1	รูปแบบตัวอักษรที่นำมาใช้และตัวอย่างรูปแบบตัวอักษร	23
3.2	การกำหนดเครื่องหมายให้กับตัวอักษรล้านนา	24
3.3	การกำหนดพารามิเตอร์ของสามเทคนิควิธีการแปลงภาพเป็นภาพแบบขาวดำ	26
3.4	รายละเอียดชุดข้อมูลภาพเอกสาร	30
4.1	การแบ่งส่วนภาพของแบบจำลองฮิสโตแกรมแมทซิง	37
4.2	ร้อยละของประสิทธิภาพความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ของแบบจำลองฮิสโตแกรมแมทซิง	37
4.3	ร้อยละของประสิทธิภาพความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ของแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน	38
4.4	ร้อยละของประสิทธิภาพความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ของแบบจำลองการถดถอยโลจิสติก	38
4.5	รายชื่อแบบจำลองที่ทำการรวมแบบจำลองโหวต	39
4.6	ร้อยละของประสิทธิภาพความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ของการรวมแบบจำลองโหวต	40
4.7	ร้อยละของประสิทธิภาพความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ของแบบจำลองฮิสโตแกรมแมทซิงร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน แบบจำลองการถดถอยโลจิสติก และการรวมแบบจำลองโหวต	41

สารบัญภาพ

ภาพที่		หน้า
2.1	ตัวอักษรภาษาไทยเทียบตัวอักษรภาษาล้านนา (ก) กลุ่มอักษรในวรรค (ข) กลุ่มอักษรนอกวรรค (ค) กลุ่มอักษรเพิ่มเติม	2
2.2	โครงสร้างการวาง และวิธีการเขียนตัวอักษรล้านนา	2
2.3	ลักษณะของภาพที่คอมพิวเตอร์มองเห็น (ก) ภาพสี (ข) ภาพสีเทาหรือขาวดำ	4
2.4	กราฟฮิสโตแกรมจำนวนผู้ป่วยกับช่วงเวลาเข้ารับบริการ	8
2.5	ฮิสโตแกรมแมทซ์ของภาพจุด (Cloud of points) ที่ทำการแบ่งส่วนภาพเป็น 9 ส่วน (Grid for counting) และแสดงกราฟฮิสโตแกรม (Histogram)	8
2.6	แผนภูมิแบบเส้นที่ผ่านการใช้ฟังก์ชันโลจิสติก หรือฟังก์ชันซิกมอยด์	10
2.7	กระบวนการสร้างแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน	12
2.8	ขั้นตอนการคำนวณของคอนโวลูชัน (ก) ผลลัพธ์ F_1 (ข) ผลลัพธ์ F_2 (ค) ผลลัพธ์ F_9	14
2.9	ขั้นตอนการคำนวณของการลดขนาด (ก) วิธีการ MaxPooling (ข) AvgPooling (ค) MinPooling	15
2.10	กระบวนการทำงานของชั้นที่เชื่อมต่อ	16
2.11	โครงสร้างการตั้งค่าพารามิเตอร์ของแบบจำลอง LeNet-5	17
3.1	ขั้นตอนการสร้างแบบจำลองและประเมินผลแบบจำลอง	22
3.2	ขั้นตอนการประมวลผลข้อมูลล่วงหน้า	26
3.3	(ก) ภาพต้นฉบับ ภาพที่ผ่านการแปลงภาพเป็นภาพขาวดำโดยใช้ (ข) วิธีการ Niblack's (ค) วิธีการ Nick และ (ง) วิธีการ Otsu thresholding	27
3.4	(ก) ภาพที่ผ่านการแปลงภาพขาวดำ (ข) ภาพที่ผ่านการตัดส่วนของภาพ	28
3.5	ภาพที่ผ่านการครอบตัดภาพ (ข) ภาพที่ผ่านการเปลี่ยนขนาดภาพเป็นขนาด 64x64 พิกเซล	28
3.6	(ก) ภาพลักษณะของข้อมูล (ข) ภาพการกำหนดเงื่อนไขของอาร์เรย์ที่มีค่า เป็น 0 เท่ากับสีขาว และมีค่าเป็น 1 เท่ากับสีดำ	29
3.7	กระบวนการสร้างแบบจำลองฮิสโตแกรมแมทซ์	31
3.8	กระบวนการสร้างแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน	32
3.9	กระบวนการสร้างแบบจำลองการถดถอยโลจิสติก	33

สารบัญภาพ (ต่อ)

ภาพที่	หน้า	
3.10	กระบวนการทำงานของการรวมแบบจำลองโหวต	34
ก.1	โค้ดการสร้างแบบจำลอง	47

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ภาษาล้านนาเป็นมรดกทางภาษาของชาวล้านนา บันทึกไว้ในสื่อหลายชนิด เช่น หนังสือใบลาน และศิลาจารึก ประกอบด้วย ข้อมูลทางประวัติศาสตร์ ประเพณี ศิลปวัฒนธรรม กฎหมาย วรรณกรรมพื้นบ้าน ตำรายาสมุนไพร และการแพทย์ ซึ่งหนังสือใบลาน และศิลาจารึกผู้ฟังได้ง่าย เมื่อขาดการดูแลรักษาเป็นเวลานาน เอกสารจึงเกิดความเสียหายทำให้ส่งผลกระทบต่อลูกหลานชาวล้านนาในรุ่นหลังที่ยากต่อการศึกษาข้อมูลทางประวัติศาสตร์

ปัญหาหนังสือใบลาน และศิลาจารึกที่ใช้ในการบันทึกข้อมูลภาษาล้านนาผู้ฟังได้ง่าย จึงเป็นปัญหาที่มีความจำเป็นต้องหาทางแก้ไข ด้วยการพัฒนาแบบจำลองที่สามารถรู้จำลายมือเขียนอักษรล้านนาด้วยเทคนิคการรวมแบบจำลองโหวตในการรู้จำลายมือเขียนอักษรล้านนา เพื่อแก้ไขปัญหาดังกล่าวด้วยการเก็บข้อมูลอักษรล้านนาให้อยู่ในรูปแบบดิจิทัล

ดังนั้นในงานวิจัยเล่มนี้ ผู้วิจัยต้องการศึกษาเทคนิคฮิสโตแกรมแมทซิง โครงข่ายประสาทเทียมแบบคอนโวลูชัน การถดถอยโลจิสติก และการรวมแบบจำลองโหวต เพื่อสร้างแบบจำลองการรู้จำลายมือเขียนอักษรล้านนาให้มีความแม่นยำสามารถนำไปใช้ในการเก็บข้อมูลตัวอักษรล้านนาให้อยู่ในรูปแบบดิจิทัล เพื่ออนุรักษ์วัฒนธรรม และประวัติศาสตร์ชาวล้านนา

1.2 วัตถุประสงค์

เพื่อประยุกต์ใช้เทคนิคการรวมแบบจำลองโหวตในการรวมแบบจำลองฮิสโตแกรมแมทซิง ร่วมกับโครงข่ายประสาทเทียมแบบคอนโวลูชัน และการถดถอยโลจิสติกให้มีความแม่นยำมากขึ้น

1.3 ประโยชน์ที่คาดว่าจะได้รับ

ได้ประยุกต์ใช้เทคนิคการรวมแบบจำลองโหวตในการรวมฮิสโตแกรมแมทซิงร่วมกับโครงข่ายประสาทเทียมแบบคอนโวลูชัน และการถดถอยโลจิสติกให้มีความแม่นยำมากขึ้น

1.4 ขอบเขตของการวิจัย

สามารถรู้จำ และวิเคราะห์ภาพตัวอักษรภาษาล้านนาที่เป็นกลุ่มอักษรในวรรค นอกวรรค และกลุ่มอักษรเพิ่มเติม ยกเว้นตัวอักษร (๕, ๗) รวมทั้งหมด 42 ตัวอักษร

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การศึกษาวิจัยเรื่อง “การรู้จำลายมือเขียนอักษรล้านนาด้วยเทคนิคการรวมแบบจำลองโหวต” ผู้วิจัยได้ศึกษาแนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้องเพื่อใช้ประกอบเป็นแนวทางในการทำวิจัยโดยแยกออกเป็น 7 หัวข้อดังต่อไปนี้

- 2.1 ตัวอักษรภาษาล้านนา
- 2.2 การประมวลผลภาพ
- 2.3 ฮิสโตแกรม
- 2.4 การเรียนรู้ของเครื่อง
- 2.5 การเรียนรู้เชิงลึก
- 2.6 การประเมินประสิทธิภาพแบบจำลอง
- 2.7 งานวิจัยที่เกี่ยวข้อง

2.1 ตัวอักษรภาษาล้านนา

ตัวอักษรภาษาล้านนา[1] เป็นอักษรที่ใช้เขียนภาษาล้านนา แบ่งออกเป็น 3 กลุ่ม กลุ่มอักษรใน วรรคแบ่งออกเป็น 5 ววรรค ววรรคละ 5 ตัว มี 25 ตัวอักษร กลุ่มอักษรนอกวรรคมี 8 ตัวอักษร กลุ่มอักษรเพิ่มเติมมี 11 ตัวอักษร และกลุ่มอักษรที่เป็นเสียงสระ 8 ตัวอักษร ดังภาพที่ 2.1

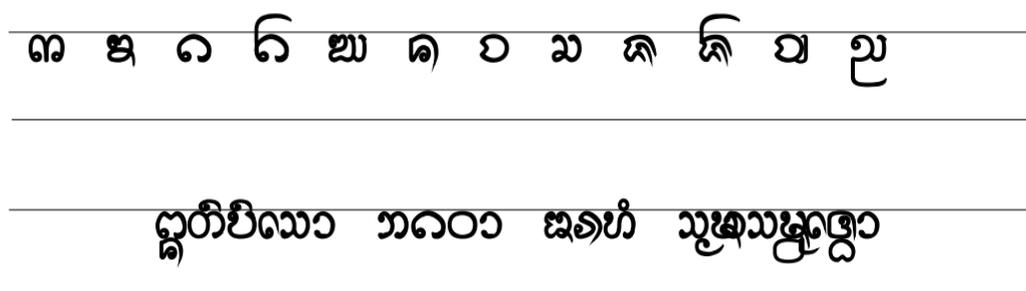
	ก	ข	ค	ฃ	ง
	ด	ธ	ด	ข	ค
	จ	ฉ	ช	ฌ	ญ
	อ	ว	ล	ว	บ
กลุ่มอักษรในวรรค	ฎ	ฐ	ด, ๓, ฎ	ฒ	ณ
	ย	ย	๕, ๖, ๗	๘	๙
	ต	ถ	ท	ธ	น
	ด	ด	ด	ด	ด
	ป	ผ	พ	ภ	ม
	บ	ฝ	ผ	ว	ฮ

(ก)

	ย	ร	ล	ว	ส
กลุ่มอักษรนอกรวรรค	ข	ฉ	ณ	อ	ง
	ห	ฬ	อ		
	บ	ป	ผ		
			(ข)		
กลุ่มอักษรเพิ่มเติม	ช	ค	ซ	บ	ป
	ฝ	ฉ	ฉ	บ	ฝ
	ฟ	ศ	ษ	อย, หย	ฮ
	ด	ต	ด	ค	ฏ
		(ค)			

ภาพที่ 2.1 ตัวอักษรภาษาไทยเทียบตัวอักษรภาษาล้านนา (ก) กลุ่มอักษรในวรรค (ข) กลุ่มอักษรนอกรวรรค (ค) กลุ่มอักษรเพิ่มเติม

โครงสร้างการวาง และวิธีการเขียนตัวอักษรล้านนา พยัญชนะล้านนาจะเขียนข้างล่างบรรทัดเท่านั้นดังภาพที่ 2.2 เพราะตัวอักษรล้านนามีสระ และพยัญชนะบางตัวที่มีรูปแบบพิเศษจำเพาะจึงทำให้ยากแก่การเขียนไว้บนบรรทัด ในกรณีที่มีการผสมคำจะยากแก่การบังคับให้สระและพยัญชนะให้มีระยะห่างของช่องไฟสม่ำเสมอ ซึ่งอาจทำให้ไม่เป็นระเบียบได้ โครงสร้างการวาง และวิธีการเขียนจึงเป็นส่วนสำคัญในการเขียนของตัวอักษรล้านนา



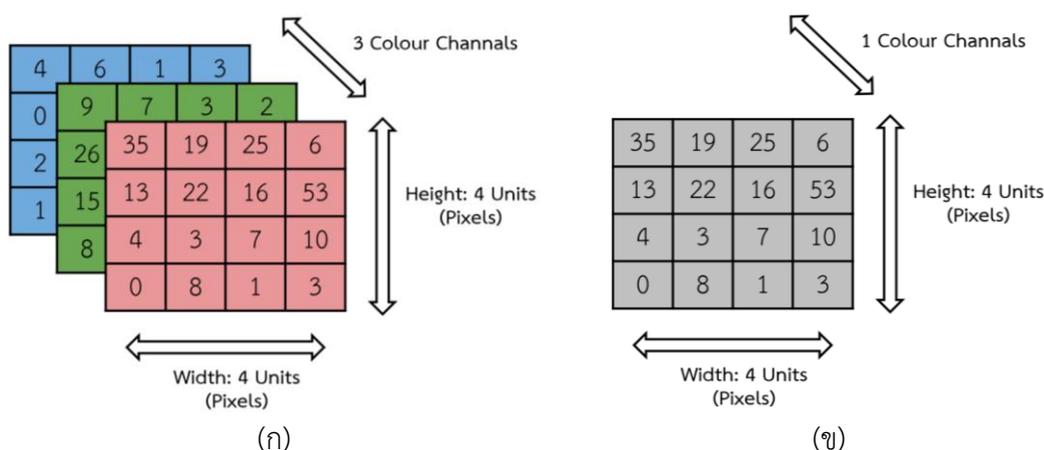
ภาพที่ 2.2 โครงสร้างการวาง และวิธีการเขียนตัวอักษรล้านนา

2.2 การประมวลผลภาพเอกสาร

การประมวลผลภาพเอกสาร (Document Image Processing) เป็นกระบวนการที่ใช้คอมพิวเตอร์เพื่อปรับปรุง แปลงรูปร่าง และวิเคราะห์ภาพดิจิทัล สามารถดำเนินการกับรูปภาพ 2 มิติหรือ 3 มิติ โดยมีวัตถุประสงค์หลัก เพื่อการดึงข้อมูลที่มีความสำคัญออกจากภาพหรือการเพิ่มความเข้าใจ และปรับปรุงคุณภาพของรูปภาพให้ดียิ่งขึ้น

เนื่องจากเอกสารอยู่ในรูปแบบของภาพดิจิทัล การประมวลผลภาพดิจิทัลเป็นสิ่งสำคัญ โดยเฉพาะในงานวิเคราะห์เอกสาร ก่อนที่จะสามารถรู้จำรูปภาพในเอกสารได้ จำเป็นต้องทำการแปลงภาพเอกสารให้อยู่ในรูปแบบที่เหมาะสม

ข้อมูลรูปภาพประกอบด้วยสี 3 สี แดง น้ำเงิน และเขียว ในคอมพิวเตอร์มองเป็น 3 ช่องสัญญาณ (Channel) หรืออาร์เรย์ (Array) 3 มิติ เรียกว่า RGB ดังภาพที่ 2.3 (ก) และภาพสีเทาหรือขาวดำมีเพียง 1 ช่องสัญญาณ หรืออาร์เรย์ 2 มิติ ดังภาพที่ 2.3 (ข) ในช่องสัญญาณประกอบด้วยพิกเซลมีค่าระหว่าง 0 ถึง 255



ภาพที่ 2.3 ลักษณะของภาพที่คอมพิวเตอร์มองเห็น (ก) ภาพสี (ข) ภาพสีเทาหรือขาวดำ

(ที่มา : <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-net-works-the-eli5-way/>, สืบค้นเมื่อวันที่ 14 ตุลาคม 2566)

2.2.1 การแปลงภาพเป็นภาพแบบขาวดำ

การแปลงภาพเป็นภาพแบบขาวดำ (Image Binarization) คือ ส่วนหนึ่งของการประมวลผลภาพ เป็นกระบวนการแปลงภาพระดับสีเทาให้เป็นภาพขาวดำ ซึ่งแต่ละพิกเซลจะมีค่าเป็น 0 หรือ 1 (Binary) เนื่องจากการประมวลผลภาพสีใช้เวลานาน และใช้พื้นที่ในการวิเคราะห์มาก ในการวิเคราะห์ภาพเอกสารจำเป็นต้องทำการแปลงภาพเป็นภาพแบบขาวดำในวัตถุที่พิจารณา

เช่น อักษร รูปภาพ และตาราง เป็นต้น วิธีการแปลงภาพเป็นภาพขาวดำขึ้นอยู่กับ ระดับสีเทา สามารถแบ่งออกเป็นสองประเภท[2] ได้แก่ การกำหนดค่าเทรชโฮลด์แบบทั้งภาพ และกำหนดค่าเทรชโฮลด์แบบเฉพาะที่

การกำหนดค่าเทรชโฮลด์แบบทั้งภาพ

การกำหนดค่าเทรชโฮลด์แบบทั้งภาพ (Global thresholding) ใช้ค่าเทรชโฮลด์ค่าเดียวสำหรับภาพทั้งหมด สามารถคำนวณโดยใช้สมการดังนี้

$$B(x, y) = \begin{cases} 1 & I(x, y) \geq T \\ 0 & \text{Otherwise} \end{cases} \quad (2.1)$$

เมื่อ

- B คือ ค่าพิกเซลที่ตำแหน่ง (x, y) ในภาพขาวดำ
- I คือ ภาพสีเทาที่ตำแหน่ง (x, y)
- T คือ ค่าเทรชโฮลด์ ที่ถูกกำหนดจากการกำหนดค่า เช่น การกำหนดค่าด้วยวิธีออตสึ

การกำหนดค่าด้วยวิธีออตสึ (Otsu thresholding)[3] อยู่ในหมวดวิธีการกำหนดค่าเทรชโฮลด์แบบทั้งภาพ เป็นวิธีการที่มีประสิทธิภาพ และรู้จักกันดี วิธีนี้ สมมติว่าฮิสโตแกรมค่าความเข้มของพิกเซลในภาพเอกสารที่มีลักษณะแบ่งแยกเป็นสองรูปแบบ คือ สีพื้นหลัง และสีพื้นหน้าที่มีความแตกต่างกัน ค่าเทรชโฮลด์ถูกกำหนดโดยการเพิ่มค่าความแปรปรวนระหว่างรูปแบบสีพื้นหลังและสีพื้นหน้าให้มากที่สุด เรียกว่าการเลือกค่าเทรชโฮลด์ สามารถคำนวณโดยใช้สมการดังนี้

$$T = \arg \max_{t \in [0, N-1]} (n_B(t)n_O(t)[\mu_B(t) - \mu_O(t)]^2) \quad (2.2)$$

เมื่อ

$$n_B(t) = \sum_{i=0}^{t-1} P(i)$$

$$n_O(t) = \sum_{i=t}^{N-1} P(i)$$

$$\mu_B(t) = \frac{\sum_{i=0}^{t-1} iP(i)}{n_B(t)}$$

$$\mu_0(t) = \frac{\sum_{i=t}^{N-1} iP(i)}{n_0(t)}$$

โดย P คือ ฮิสโตแกรมค่าความเข้มของพิกเซล ซึ่งอยู่ในช่วง (0, N-1)

การกำหนดค่าเทรชโฮลด์แบบเฉพาะที่

การกำหนดค่าเทรชโฮลด์แบบเฉพาะที่ (Local thresholding) ใช้ค่าเทรชโฮลด์หลายค่า มีวิธีการคำนวณค่าเทรชโฮลด์ที่เหมาะสมสำหรับแต่ละพิกเซล โดยใช้ข้อมูลพิกเซลในบริเวณโดยรอบเพื่อกำหนดค่าเทรชโฮลด์ให้เป็นที่ไปตามสภาพแวดล้อมของแต่ละพิกเซล นั้นหมายความว่าค่าเทรชโฮลด์สามารถปรับให้เหมาะกับพื้นที่ที่มีความแตกต่างกันได้

การกำหนดค่าด้วยวิธีนิแบล็ค (Niblack's method)[4] อยู่ในหมวดวิธีการกำหนดค่าเทรชโฮลด์แบบเฉพาะที่ วิธีที่ถูกกล่าวถึงในวรรณกรรมหลายแห่ง คือ วิธีการกำหนดค่าด้วยวิธีนิแบล็ค มีการกำหนดค่าเทรชโฮลด์ระดับท้องถิ่นด้วยการคำนวณจากค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐานของพิกเซลในบริเวณใกล้เคียง สามารถคำนวณโดยใช้สมการดังนี้

$$T(x,y) = m(x,y) + k \times \sigma(x,y) \quad (2.3)$$

เมื่อ

- T คือ ค่าเทรชโฮลด์ที่ตำแหน่ง (x, y)
- m คือ ค่าเฉลี่ยของพิกเซลในบริเวณใกล้เคียง
- σ คือ ค่าเบี่ยงเบนค่ามาตรฐานของพิกเซลในบริเวณใกล้เคียงทั้งหมด
- k คือ ค่าควบคุมความไวของค่าเทรชโฮลด์ต่อการเปลี่ยนแปลงความเข้ม ของแต่ละพื้นที่ใกล้เคียง โดยที่ค่า k เป็นค่า -0.2 ตามที่ อ้างอิงจาก Niblack[4]

การกำหนดค่าด้วยวิธีนิค (Nick method)[5] วิธีการกำหนดค่าด้วยวิธีนิคถูกพัฒนาขึ้นมาจากวิธีการกำหนดค่าด้วยวิธีนิแบล็ค ได้ปรับปรุงในด้านปัญหาความคมชัดต่ำ โดยการลดค่าเทรชโฮลด์ลง สามารถคำนวณโดยใช้สมการดังนี้

$$T(x,y) = m + k \sqrt{\frac{\sum I(x,y)^2 - m(x,y)^2}{NP}} \quad (2.4)$$

เมื่อ

- T คือ ค่าเทรซโฮลด์ที่ตำแหน่ง (x, y)
- m คือ ค่าเฉลี่ยของพิกเซลในบริเวณใกล้เคียง
- k คือ ค่าควบคุมความไวของค่าเทรซโฮลด์ต่อการเปลี่ยนแปลงความเข้ม ของแต่ละพื้นที่ใกล้เคียง โดยที่ค่า k เป็นค่า $(-0.1, -0.2)$
- I คือ ค่าความเข้มของพิกเซลที่ตำแหน่ง (x, y)
- NP คือ จำนวนรวมของพิกเซลทั้งหมดในภาพที่ตรวจจับหรือประมวลผล

2.2.2 การแปลงภาพ

การแปลงภาพ (Image Transformation) คือ กระบวนการที่ใช้เปลี่ยนแปลงรูปแบบหรือลักษณะของภาพดิจิทัล เพื่อให้ภาพมีลักษณะหรือรูปแบบที่เหมาะสมกับการใช้งาน การแปลงภาพที่ได้รับความนิยมมักเกี่ยวข้องกับการปรับปรุงคุณภาพของภาพ และการเปลี่ยนแปลงลักษณะของภาพ เช่น การหมุนภาพ การเปลี่ยนขนาดภาพ และการตัดส่วนของภาพที่ไม่ต้องการ

การตัดส่วนของภาพ

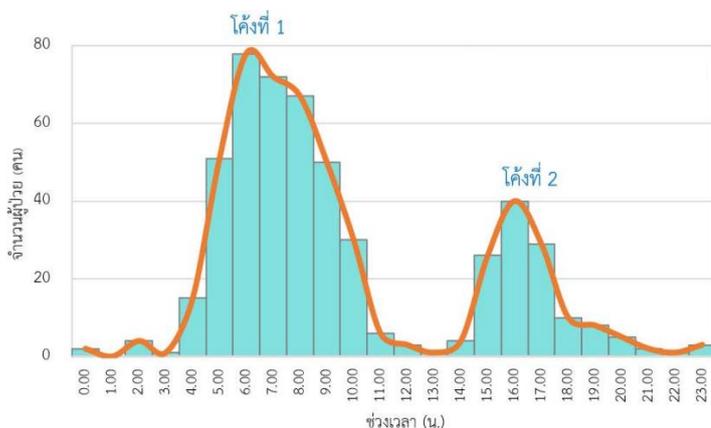
การตัดส่วนของภาพ (Cropping) คือ กระบวนการลบหรือตัดส่วนหนึ่งของภาพ เพื่อให้ได้ภาพใหม่ที่มีขนาดหรือส่วนที่ต้องการเฉพาะเจาะจง ในกระบวนการนี้สามารถเลือกเฉพาะส่วนที่สนใจ และตัดส่วนที่ไม่จำเป็นออก

การเปลี่ยนขนาดภาพ

การเปลี่ยนขนาดภาพ (Resizing) คือ กระบวนการปรับขนาดของภาพ โดยการเปลี่ยนความกว้างและความสูงของภาพให้มีขนาดใหม่ตามที่กำหนด โดยการเพิ่มหรือลดจำนวนพิกเซล ในภาพ เพื่อให้ภาพเหมาะสมกับการใช้งาน การเปลี่ยนขนาดภาพจะรักษาสัดส่วนของภาพเพื่อไม่ทำให้เกิดการบิดเบือนในภาพ

2.3 ฮิสโตแกรม

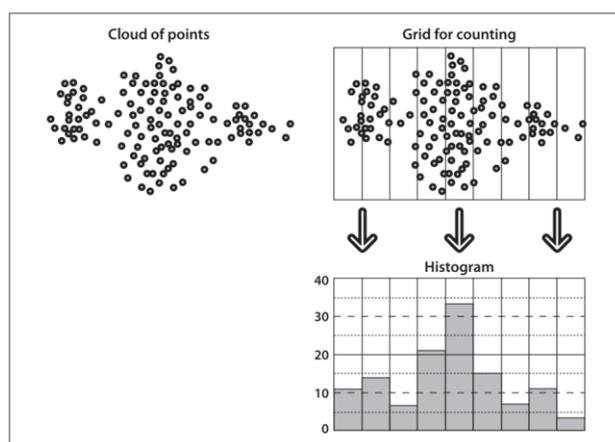
ฮิสโตแกรม (Histogram)[6] เป็นกราฟแท่งที่ใช้สำหรับการแสดงข้อมูล ที่มีความถี่หรือการกระจายของข้อมูล และจัดกลุ่มข้อมูล โดยใช้แกนแนวตั้ง (Y) แสดงข้อมูลตัวเลขความถี่ และแกนแนวนอน (X) แสดงข้อมูลที่ต้องการแสดง ความสูงของแกนแนวตั้งแสดงถึงความถี่หรือจำนวนครั้งของข้อมูลที่ตรงกับช่วงข้อมูลของแกนแนวนอน ช่วยให้เข้าใจถึงภาพรวมการแจกแจงข้อมูลดังภาพที่ 2.4



ภาพที่ 2.4 กราฟฮิสโตแกรมจำนวนผู้ป่วยกับช่วงเวลาเข้ารับบริการ

(ที่มา : https://www.si.mahidol.ac.th/th/division/um/admin/download_files/298_48_105Qe4S.pdf, สืบค้นเมื่อวันที่ 23 กันยายน 2566)

ฮิสโตแกรมแมทซิง (Histogram Matching) เป็นฟังก์ชันที่แสดงในแต่ละส่วนของภาพมีจำนวนจุด (พิกเซล) จำนวนเท่าใด โดยการแบ่งส่วนภาพแนวนอนหรือแนวตั้งตามต้องการเพื่อกำหนดขอบเขต และนับจำนวนพิกเซล ในภาพตามแต่ละส่วนมีจำนวนพิกเซลเท่าไร เมื่อนำมาแสดงเป็นกราฟฮิสโตแกรม แกนแนวนอน (X) คือ จำนวนส่วนของภาพ และแกนแนวตั้ง (Y) คือ จำนวนพิกเซล ดังภาพที่ 2.5 เป็นการทำฮิสโตแกรมแมทซิงของภาพกลุ่มจุด



ภาพที่ 2.5 ฮิสโตแกรมแมทซิงของภาพจุด (Cloud of points) ที่ทำการแบ่งส่วนภาพเป็น 9 ส่วน (Grid for counting) และแสดงกราฟฮิสโตแกรม (Histogram)

(ที่มา : Learning OpenCV: Gary Bradski and Adrian Kaehler. 2008 : 193-195, สืบค้นเมื่อวันที่ 12 ตุลาคม 2566)

2.4 การเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่อง (Machine Learning) เป็นกลุ่มหนึ่งของเทคโนโลยีที่เกี่ยวข้องกับการพัฒนาแบบจำลองคอมพิวเตอร์ที่สามารถเรียนรู้ และพัฒนาจากข้อมูล โดยไม่ต้องเขียนโปรแกรมเฉพาะ แบบจำลองที่สร้างขึ้นจะใช้ข้อมูลเป็นพื้นฐานในการเรียนรู้ สร้างความเข้าใจในแบบแผนหรือรูปแบบทางสถิติ และรูปแบบทางคณิตศาสตร์

การเรียนรู้ของเครื่อง ได้รับความนิยมนำไปใช้งานอย่างแพร่หลายในด้านชีวิตประจำวัน และธุรกิจ เช่น การรู้จำ และแนะนำสินค้าในอุตสาหกรรมอีคอมเมิร์ซ การวิเคราะห์ข้อมูลทางการแพทย์ และการจัดการสื่อสารทางภาษามธรรมชาติ เป็นต้น การใช้การเรียนรู้ของเครื่องสามารถช่วยลดเวลา และทรัพยากรในการทำงาน มีความสามารถในการตัดสินใจที่แม่นยำ ในหลายกรณี การเรียนรู้ของเครื่องมีหลายประเภทตามวิธีการเรียนรู้ และการใช้งานของอัลกอริทึมที่สามารถแบ่งออกได้เป็นสองประเภท คือ การเรียนรู้แบบมีผู้สอน และการเรียนรู้แบบไม่มีผู้สอน

2.4.1 การเรียนรู้แบบมีผู้สอน

การเรียนรู้แบบมีผู้สอน (Supervised Learning) เป็นหนึ่งในวิธีการเรียนรู้ของเครื่อง โดยแบบจำลองที่ถูกฝึกจะเรียนรู้จากชุดข้อมูลสำหรับฝึกที่มีคำตอบในข้อมูล แบบจำลองจะพยายามเรียนรู้ และสร้างความเข้าใจในความสัมพันธ์ระหว่างข้อมูลนำเข้ากับคำตอบเพื่อทำนายผลลัพธ์ต่อจากข้อมูลใหม่ที่ไม่เคยเห็นมาก่อน ตัวอย่างการใช้งานได้แก่ การจำแนกประเภทของอีเมลอย่างอัตโนมัติว่าเป็นอีเมลขยะหรือไม่ และการจำแนกภาพให้ถูกต้องว่าเป็นสัตว์เลี้ยงอะไร

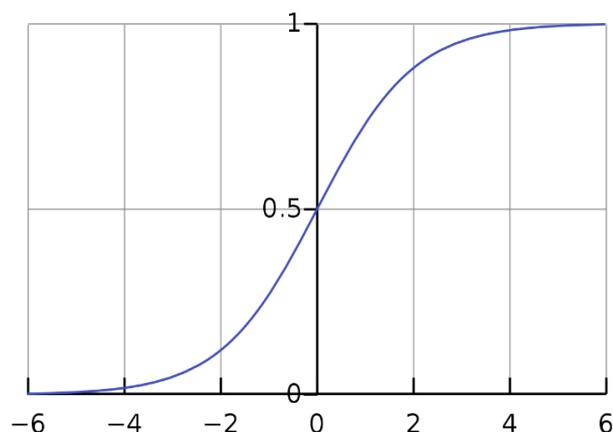
การถดถอยโลจิสติก

การถดถอยโลจิสติก (Logistic Regression) เป็นอัลกอริทึมในการเรียนรู้ของเครื่องแบบหนึ่งที่ใช้ในการจัดการกับปัญหาการจำแนกข้อมูลหรือการทำนายโอกาสเหตุการณ์สองค่าหรือมากกว่า (Binary Classification, Multi-Class Classification) โดยการใช้ฟังก์ชันโลจิสติก[7] (Logistic Function) หรือเรียกว่า ฟังก์ชันซิกมอยด์ (Sigmoid Function) เพื่อสร้างแบบจำลองที่สามารถคำนวณความน่าจะเป็น ซึ่งมักจะมีค่าระหว่าง 0 ถึง 1 โดยใช้ค่าคงที่ ที่เรียกว่าค่า พารามิเตอร์ (Parameter) ที่ได้จากระบวนการเรียนรู้จากข้อมูลนำเข้า เพื่อหาค่าที่ดีที่สุดในการจำแนกหรือการทำนาย สามารถคำนวณโดยใช้สมการดังนี้

$$\text{sigmoid function} = \frac{1}{1 + e^{-\text{value}}} \quad (2.5)$$

เมื่อ

e คือ ฐานของ natural logarithms
value คือ ตัวเลขที่ต้องการแปลง



ภาพที่ 2.6 แผนภูมิแบบเส้นที่ผ่านการใช้ฟังก์ชันโลจิสติก หรือฟังก์ชันซิกมอยด์

(ที่มา : <https://commons.wikimedia.org/wiki/File:Logistic-curve.svg>, สืบค้นเมื่อวันที่ 23 กันยายน 2566)

แผนภูมิแบบเส้นของเลขระหว่าง -6 และ 6 ที่ถูกเปลี่ยนแปลงให้อยู่ในช่วง 0 ถึง 1 ฟังก์ชันโลจิสติก หรือที่เรียกว่า ฟังก์ชันซิกมอยด์จากภาพที่ 2.6 ที่ใช้ในการถดถอยโลจิสติก โดยที่ตัวแปรต้นจะถูกรวมกันเพื่อใช้ในการเพิ่มน้ำหนัก และประเมินความสัมพันธ์กับตัวแปรตาม เพื่อคำนวณความน่าจะเป็นของเหตุการณ์เป็นบวก ตัวอย่างการใช้งานในการจำแนกอีเมลว่า เป็นอีเมลขยะหรือไม่ด้วยสมการดังนี้

$$P(Y = 1) = \frac{1}{1 + e^{(B_0 + B_1 \times X)}} \quad (2.6)$$

เมื่อ

- Y คือ ค่าตัวแปรตามที่ต้องการจำแนก (1 หรือ 0)
- B₀ คือ ค่าสัมบูรณ์ Intercept
- B₁ คือ ค่าสัมประสิทธิ์ Coefficient สำหรับตัวแปรต้น (X)
- X คือ ค่าตัวแปรต้น

โดย B จะได้จากการเรียนรู้ของแบบจำลองจากชุดข้อมูลสำหรับการเรียนรู้

ในการจำแนกอีเมลว่าเป็นอีเมลขยะหรือไม่ สมมติตัวแปรต้น (X) มีคุณลักษณะ ดังนี้ จำนวนคำ (Word Count) จำนวนคำที่มีลิงก์ (Link Count) 1 หน่วยต่อ 10 คำ สามารถคำนวณความน่าจะเป็นของอีเมลขยะ (Y=1) ด้วยการแทนค่าตัวแปรต้นในสมการดังนี้

$$P(Y = 1) = \frac{1}{1 + e^{-(B_0 + (B_1 \times \text{word count}) + (B_2 \times \text{link count}))}} \quad (2.7)$$

สำหรับการทำนายทำได้ด้วยการแทนค่าในสมการที่ 2.7 ตัวอย่างเช่น การจำแนกว่าอีเมลที่มีจำนวน 5 คำ และจำนวนคำที่มีลิงก์ 1 คำ เป็นอีเมลขยะหรือไม่ เมื่อมีการเรียนรู้และทราบค่า Coefficients ของ $B_0 = -0.1$ $B_1 = -0.4$ และ $B_2 = 1$ เมื่อใช้สมการข้างต้นจะสามารถคำนวณความเป็นไปได้ในการที่อีเมลที่มีจำนวน 50 คำ และจำนวนคำที่มีลิงก์ 10 คำ เป็นอีเมลขยะหรือไม่

$$P(Y = 1) = \frac{1}{1 + e^{-(-0.1 + (-0.4 \times 5) + (1 \times 1))}}$$

$$P(Y = 1) = 0.24974$$

เมื่อ P เข้าใกล้ศูนย์จะทำนายว่าความน่าจะเป็นอีเมลขยะต่ำ ซึ่งสามารถกำหนด P เพื่อแบ่งกลุ่มเป็นสองกลุ่มได้ดังนี้

$$\text{prediction} = \begin{cases} 1 & P(Y = 1) \geq 0.5 \\ 0 & P(Y = 1) < 0.5 \end{cases}$$

2.4.2 การเรียนรู้แบบไม่มีผู้สอน

การเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) เป็นหนึ่งในวิธีการเรียนรู้ของเครื่อง โดยแบบจำลองไม่ได้รับข้อมูลเริ่มต้นที่มีคำตอบหรือคำแนะนำ แต่จะหาโครงสร้างหรือรูปแบบในข้อมูลเพื่อจัดกลุ่มวัดความคล้ายคลึงหรือลดมิติข้อมูล ตัวอย่างการใช้งานได้แก่ การจัดกลุ่มข้อมูล (Clustering) เพื่อหากลุ่มคล้ายคลึงกัน และการลดมิติข้อมูล (Dimensionality Reduction) เพื่อลดขนาดของข้อมูลในทางคณิตศาสตร์

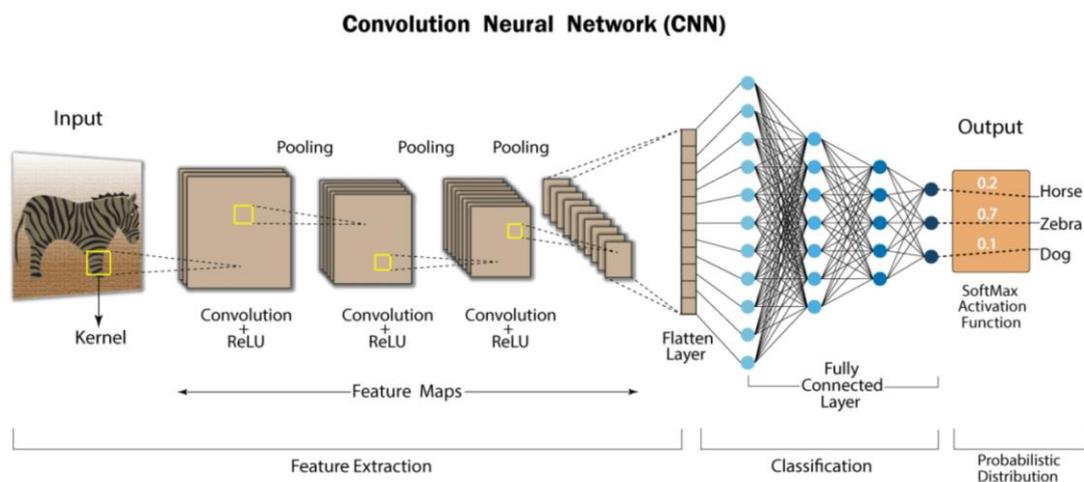
2.5 การเรียนรู้เชิงลึก

การเรียนรู้เชิงลึก (Deep Learning)[8] เป็นกลุ่มหนึ่งของเทคนิคการเรียนรู้ของเครื่อง ซึ่งใช้โครงข่ายประสาทเทียม (Artificial Neural Networks) ในการจำลองการเรียนรู้ของมนุษย์ และ

ใช้ในการแก้ไขงานที่ต้องการความเข้าใจ และการระบุรูปแบบข้อมูลที่ซับซ้อน โดยที่ไม่ต้องระบุกฎหรือสมการทางคณิตศาสตร์โดยชัดเจน ดังนั้นการเรียนรู้เชิงลึกมักจะเหมาะสำหรับการทำนาย และการจัดการข้อมูลที่มีความซับซ้อน เช่น การจดจำรูปภาพ (Image Recognition) การแปลภาษา (Machine Translation) และการระบุวัตถุในวิดีโอ (Object Detection in Videos) เป็นต้น

2.5.1 โครงข่ายประสาทเทียมแบบคอนโวลูชัน

โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network)[9] เป็นแบบหนึ่งของโครงข่ายประสาทเทียมที่ถูกออกแบบมาเพื่อการประมวลผลข้อมูลที่มีลักษณะเป็นภาพหรือวิดีโอ มีการใช้งานในหลายแอปพลิเคชัน และงานวิจัยด้านการรู้จำหรือตรวจจับวัตถุในรูปภาพ



ภาพที่ 2.7 กระบวนการสร้างแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน

(ที่มา : <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-understand-the-basics>, สืบค้นเมื่อวันที่ 18 ตุลาคม 2566)

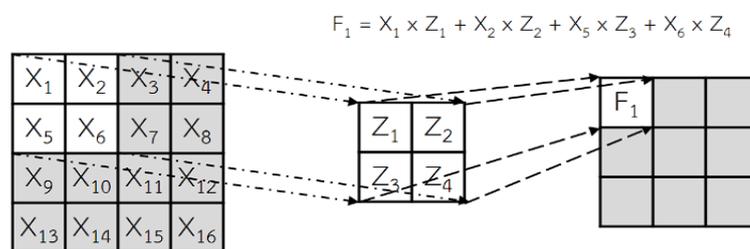
จากภาพที่ 2.7 เป็นกระบวนการสร้างแบบจำลองโครงข่ายประสาทเทียมคอนโวลูชัน โดยมีลำดับขั้นตอน 3 ขั้นตอน ได้แก่

1. Feature Extraction ประกอบด้วยข้อมูลนำเข้า (Input) และแผนที่คุณลักษณะ (Feature Maps) ซึ่งในภาพที่ 2.7 มี Input เป็นภาพสี มีลักษณะดังภาพที่ 2.3 (ก) เป็น Array 3 มิติ พิกเซลมีค่าระหว่าง 0 ถึง 255 เมื่อนำ Input เข้าสู่ขั้นตอนคอนโวลูชัน (Convolution) หรือการลดขนาด (Pooling) จะได้ผลลัพธ์เรียกว่า Feature Maps และนำ Feature Maps ผ่านชั้นเรียบ (Flatten Layer) จะมีลักษณะเป็น Array 1 มิติ เพื่อนำไปสู่ลำดับขั้นตอนถัดไป

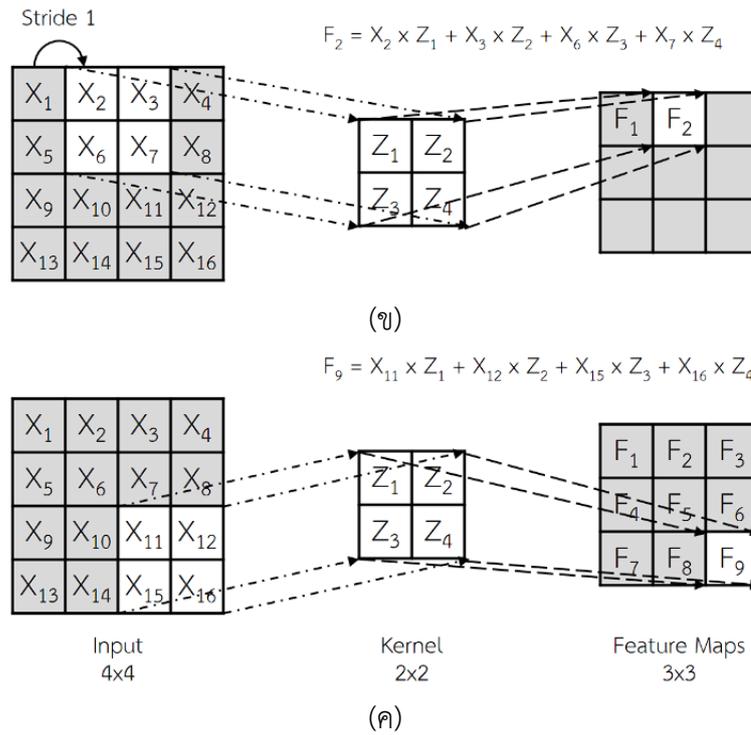
2. Classification เป็นขั้นตอนที่นำข้อมูลจากขั้นตอน Feature Extraction นำมาประมวลผล เรียกว่าชั้นที่เชื่อมต่อทั้งหมด (Fully Connected Layer) หรือโครงข่ายประสาทเทียม (Neural Network) ประกอบด้วย Input Layer คือ ชั้นการนำเข้าข้อมูลโดยมีโหนด (Node) เท่ากับจำนวนข้อมูลจากชั้น Flatten Layer ซึ่งในภาพที่ 2.7 มีจำนวน 11 โหนด Hidden Layer คือ ชั้นที่มีโหนดอยู่ภายใน และทำหน้าที่ประมวลผลข้อมูลที่ได้จากชั้น Input Layer โดยความลึกของโครงข่ายนั้นขึ้นอยู่กับจำนวนชั้นของ Hidden Layer ส่วนจำนวน Node สามารถกำหนดได้ ซึ่งในภาพที่ 2.7 มีชั้น Hidden Layer 2 ชั้น ชั้นที่ 1 มี 7 โหนด ชั้นที่ 2 มี 5 โหนด และ Output Layer คือ ชั้นผลลัพธ์ของการประมวลผล โดยจำนวนโหนดในชั้นนี้ขึ้นอยู่กับประเภทของการทำนาย ซึ่งในภาพที่ 2.7 มีจำนวน 3 โหนด
3. Probabilistic Distribution คือ ขั้นตอนการนำผลลัพธ์จากขั้นตอน Classification มาแสดงผลลัพธ์ พร้อมกับค่าจากฟังก์ชันการกระตุ้น (Activation Functions) ซึ่งในภาพที่ 2.7 มีจำนวนผลลัพธ์ 3 ผลลัพธ์ Horse 0.2 Zebra 0.7 และ Dog 0.1

หลักการทำงานของโครงข่ายประสาทเทียมแบบคอนโวลูชันเริ่มจากขั้นตอน Feature Extraction ที่มีการนำข้อมูลภาพหรือวิดีโอเข้ามาเพื่อสกัดคุณลักษณะ และลดขนาด มีการทำงานดังนี้

คอนโวลูชัน (Convolution)[10] คือ การสกัดคุณลักษณะของข้อมูลจากพื้นที่เฉพาะในรูปภาพ โดยการคูณระหว่าง Input กับตัวกรอง (Kernel) ดังภาพที่ 2.8 ซึ่งมีพารามิเตอร์ที่ต้องกำหนด คือ ขนาดของ Input ขนาดของ Kernel และ ค่า Stride



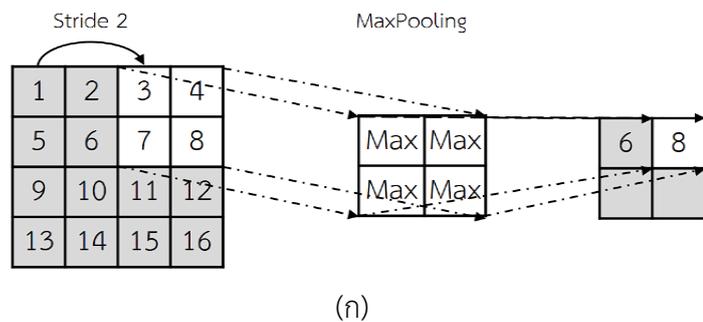
(ก)

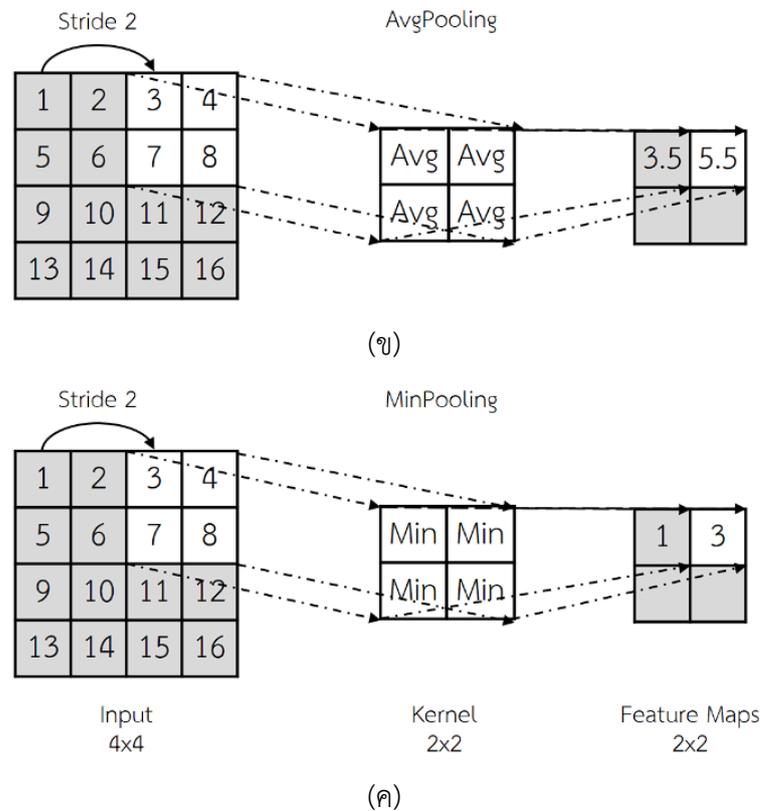


ภาพที่ 2.8 ขั้นตอนการคำนวณของคอนโวลูชัน (ก) ผลลัพธ์ F_1 (ข) ผลลัพธ์ F_2 (ค) ผลลัพธ์ F_9

จากภาพที่ 2.8 ขั้นตอนการคำนวณของคอนโวลูชัน คือ หาผลรวมของการคูณระหว่าง Input กับ Kernel โดยใช้ Kernel ขยับไปทั่วทั้ง Input จากซ้ายไปขวา และบนลงล่าง เมื่อทำขั้นตอนคอนโวลูชันเสร็จจะได้ Feature Maps และนำไปใช้ในขั้นตอนถัดไป

การลดขนาด (Pooling)[11] คือ การลดขนาดเพื่อลดขนาดของข้อมูลที่สกัดลักษณะเด่นแล้ว ซึ่งช่วยลดการคำนวณ และทำให้แบบจำลองมีความสามารถในการทำงานที่เร็วขึ้น มีหลากหลายวิธี เช่น วิธีการ MaxPooling AvgPooling และ MinPooling ดังภาพที่ 2.9 ซึ่งมีพารามิเตอร์ที่ต้องกำหนด คือ ขนาดของ Input ขนาดของ Kernel และ Stride





ภาพที่ 2.9 ขั้นตอนการคำนวณของการลดขนาด (ก) วิธีการ MaxPooling (ข) AvgPooling (ค) MinPooling

จากภาพที่ 2.9 ขั้นตอนการคำนวณของการลดขนาด คือ การเลือกค่าระหว่าง Input กับ Kernel ด้วยวิธีการ ดังนี้

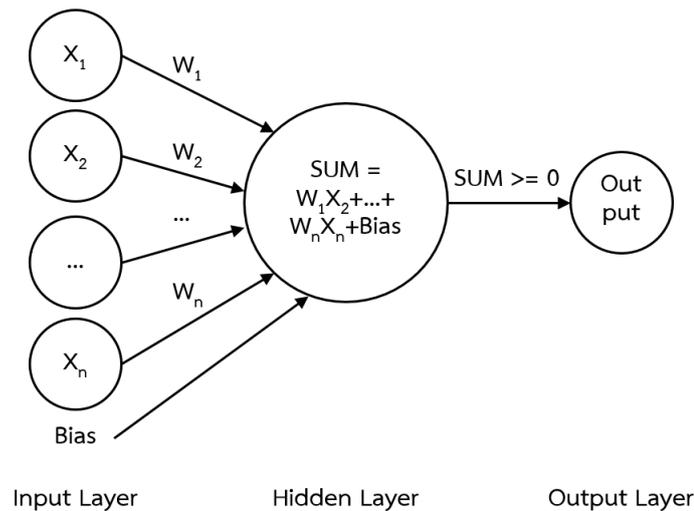
1. วิธีการ MaxPooling คือ การเลือกค่ามากที่สุด
2. วิธีการ AvgPooling คือ การเลือกค่าเฉลี่ย
3. วิธีการ MinPooling คือ การเลือกค่าน้อยที่สุด

โดยใช้ Kernel ขยับไปทั่วทั้ง Input จากซ้ายไปขวา และบนลงล่าง เมื่อทำขั้นตอนการลดขนาดเสร็จ จะได้ Feature Maps และนำไปใช้ในขั้นตอนถัดไป

ชั้นเรียบ (Flatten Layer) คือ การปรับลักษณะของข้อมูลให้อยู่ในรูปแบบที่พร้อมจะนำไปใช้ในขั้นตอน Classification ซึ่งข้อมูลที่ผ่านมาชั้นเรียบจะมีลักษณะเป็น Array 1 มิติ

หลังจากขั้นตอน Feature Extraction จะเข้าสู่ขั้นตอน Classification ที่มีขั้นตอนเพียงหนึ่งขั้นตอน คือ ชั้นที่เชื่อมต่อทั้งหมด

ชั้นที่เชื่อมต่อทั้งหมด (Fully Connected Layer)[12] คือ การนำข้อมูลเข้าสู่ชั้นที่เชื่อมต่อทั้งหมด ที่ประกอบด้วย Input Layer Hidden Layer และ Output Layer เริ่มจากชั้น Input Layer มีการนำเข้าสู่ข้อมูลเรียกว่า Node (X_n) โดยแต่ละ Node จะมีค่าน้ำหนักที่เป็นส่วนสำคัญในการทำนายผลลัพธ์ดังภาพที่ 2.10



ภาพที่ 2.10 กระบวนการทำงานของชั้นที่เชื่อมต่อ

จากภาพที่ 2.10 โดยในชั้น Hidden Layer จะทำการคำนวณ Node กับค่าน้ำหนัก สามารถคำนวณโดยใช้สมการดังนี้

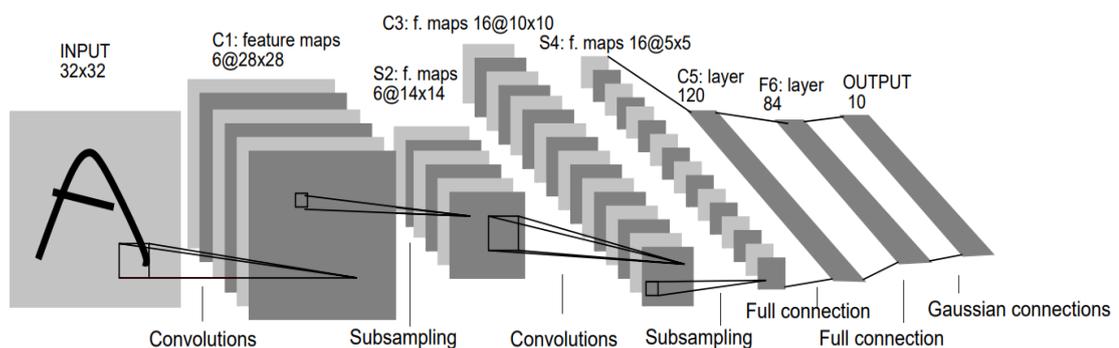
$$SUM = \sum_{i=1}^n (X_i \times W_i) + Bias \quad (2.8)$$

เมื่อ

SUM	คือ	ผลลัพธ์จากการคำนวณ
n	คือ	จำนวน Node ทั้งหมด
X_i	คือ	ค่าของ Node
W_i	คือ	ค่าน้ำหนักที่ต้องปรับให้เหมาะสมกับ Input Layer
Bias	คือ	ค่าพารามิเตอร์ที่ต้องปรับให้เหมาะสมกับ Input Layer

เมื่อ SUM มีค่ามากกว่า 0 จะส่งออกค่า 1 ไปยังชั้น Output Layer แต่ถ้า SUM น้อยกว่า 0 จะส่งออกค่า 0 ไปยังชั้น Output Layer ในกรณีที่มีคำตอบของ Output เป็น 0 หรือ 1 (Binary Classification) แต่ถ้าในกรณีที่มีคำตอบของหลายหมวดหมู่ (Multi Classification) การใช้ Multi-layer Perceptron (MLP) จะมีความยืดหยุ่นมากขึ้น

ขั้นตอนสุดท้าย Probabilistic Distribution คือ การแสดงผลลัพธ์ของการประมวลผล โดยก่อนที่จะแสดงผลลัพธ์ ต้องนำผลลัพธ์มาผ่านฟังก์ชันการกระตุ้น ประกอบด้วย ReLU Sigmoid และ Tanh ซึ่งส่วนใหญ่จะใช้ ReLU ซึ่งหมายถึงถ้าผลลัพธ์มีค่าน้อยกว่า 0 จะกลายเป็น 0 และมีค่าเป็นบวกจะคงไว้เป็นค่านั้น หลังจากนั้นจะแสดงผลลัพธ์พร้อมกับค่าที่ได้จากฟังก์ชันการกระตุ้น



ภาพที่ 2.11 โครงสร้างการตั้งค่าพารามิเตอร์ของแบบจำลอง LeNet-5

(ที่มา : Gradient-Based Learning Applied to Document Recognition : Yann LeCun, Leon Bottou, Yoshua Bengio and Patrick Haffner. 1998 : 6-7, สืบค้นเมื่อวันที่ 12 ตุลาคม 2566)

แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน LeNet-5 จากภาพที่ 2.11 เป็นที่รู้จักอย่างแพร่หลายที่ออกแบบโดย Yann LeCun, Leon Bottou, Yoshua Bengio และ Patrick Haffner เพื่อการรู้จำอักษรและตัวอักษรพิมพ์ในยุค 1990 สถาปัตยกรรม LeNet-5 ออกแบบเพื่อระบุตัวเลขที่เขียนด้วยมือ ในชุดข้อมูล MNIST[13] โครงสร้างมีความเรียบง่าย และง่ายต่อการเข้าใจ ข้อมูลนำเข้ารูปภาพสีเทาแบบกระดาดขนาด 32x32 และคอนโวลูชันกำหนดพารามิเตอร์ Kernel ขนาด 5x5 Stride 1 ก็กับการลดขนาดกำหนดพารามิเตอร์ Kernel ขนาด 2x2 Stride จำนวน 2 คู่ และชั้นที่เชื่อมต่อทั้งหมด 2 ชั้น โดยกำหนดจำนวน Node 120 และ 84 ตามลำดับ และในส่วนท้ายของโครงสร้างมีผลลัพธ์พร้อมกับค่าความมั่นใจ

2.6 การประเมินประสิทธิภาพแบบจำลอง

การประเมินประสิทธิภาพของแบบจำลอง เป็นขั้นตอนสำคัญในกระบวนการสร้าง และใช้แบบจำลองทางด้านการเรียนรู้ของเครื่อง และปัญญาประดิษฐ์ การประเมินช่วยให้ทราบว่าแบบจำลองทำงานอย่างไร และมีประสิทธิภาพในการแก้ไขปัญหาหรือการทำนายผล การประเมินแบบจำลองสามารถทำได้หลายวิธี ควรเลือกใช้ให้เหมาะสมกับวัตถุประสงค์ของแบบจำลองข้อมูล

2.6.1 เมทริกซ์ความสับสน

เมทริกซ์ความสับสน (Confusion Matrix) เป็นเครื่องมือที่ใช้ในการประเมินประสิทธิภาพของแบบจำลองในงานการจำแนก (Classification) โดยสร้างตารางเพื่อแสดงผลการทำนายของแบบจำลองเทียบกับค่าจริงของข้อมูลที่ใช้ในการฝึก และทดสอบโดยมีส่วนประกอบหลัก 4 อย่างดังนี้

1. True Positives (TP) หมายถึง ข้อมูลที่ทำนายถูกต้องว่าเป็น Positive
2. True Negatives (TN) หมายถึง ข้อมูลที่ทำนายถูกต้องว่าเป็น Negative
3. False Positives (FP) หมายถึง ข้อมูลที่ทำนายผิดว่าเป็น Positive ซึ่งเรียกว่า False Alarm
4. False Negatives (FN) หมายถึง ข้อมูลที่ทำนายผิดว่าเป็น Negative ซึ่งเรียกว่า Miss

เมื่อมี Confusion Matrix สามารถนำข้อมูลที่ได้ไปใช้ในการคำนวณการประเมินประสิทธิภาพแบบจำลอง เช่น Accuracy Precision Recall และ F1-Score

2.6.2 ความแม่นยำและความสามารถในการครอบคลุม

การประเมินประสิทธิภาพของแบบจำลองโดยใช้ ความแม่นยำและความสามารถในการครอบคลุม (Precision and Recall) เหมาะสมกับข้อมูลที่ไม่สมดุล (Imbalanced Data)

ความแม่นยำ (Precision) หมายถึง ความสามารถของแบบจำลองในการทำนายผลให้ถูกต้องเมื่อมีการทำนายว่าเป็น Positive class มีความสำคัญเมื่อต้องการให้ความเชื่อมั่นในความแม่นยำของการทำนาย สามารถคำนวณโดยใช้สมการดังนี้

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.9)$$

เมื่อ

Precision	คือ ความแม่นยำในการทำนายผลให้ถูกต้อง
TP	คือ จำนวนของตัวอย่างที่แบบจำลองทำนายว่าเป็น Positive และเป็นจริง
FP	คือ จำนวนของตัวอย่างที่แบบจำลองทำนายว่าเป็น Positive แต่เป็นเท็จ

ความสามารถในการครอบคลุม (Recall) หมายถึง ความสามารถของแบบจำลองในการครอบคลุมตัวอย่างที่เป็น Positive Class มีความสำคัญในงานที่ต้องการครอบคลุมหรือตรวจจับสิ่งที่สำคัญ เป็นการประเมินแบบจำลองว่าสามารถทำนายผลกลุ่ม Positive มีความแม่นยำเพียงใดสามารถคำนวณโดยใช้สมการดังนี้

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.10)$$

เมื่อ

Recall	คือ	ความสามารถในการครอบคลุมในการทำนายผลกลุ่ม positive
TP	คือ	จำนวนของตัวอย่างที่แบบจำลองทำนายว่าเป็น Positive และเป็นจริง
FN	คือ	จำนวนของตัวอย่างที่แบบจำลองทำนายว่าเป็น Negative แต่เป็นจริง

2.6.3 คะแนน F1

คะแนน F1 (F1-Score) เป็นค่าเฉลี่ยระหว่างความแม่นยำ และความสามารถในการครอบคลุม ค่า F1-Score มีความสำคัญเมื่อต้องการประเมินประสิทธิภาพของแบบจำลองในกรณีที่ความสำคัญของความแม่นยำ และความสามารถในการครอบคลุม เท่ากัน หรือเมื่อคุณไม่สามารถตัดสินใจว่าควรให้ความสำคัญกับความสำคัญของความแม่นยำหรือความสามารถในการครอบคลุมมากกว่าหรือน้อยกว่ากันได้อย่างชัดเจน ค่า F1-Score สามารถใช้เพื่อเปรียบเทียบประสิทธิภาพระหว่างแบบจำลองที่ใช้ในงานที่ความสำคัญของ Precision และ Recall เท่ากันหรือใกล้เคียงกัน ค่า F1-Score มีค่าระหว่าง 0 ถึง 1 โดยค่าเข้าใกล้ 1 ยิ่งแสดงถึงประสิทธิภาพที่ดีมากขึ้น สามารถคำนวณโดยใช้สมการดังนี้

$$\text{F1 Score} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.11)$$

เมื่อ

F1 Score	คือ	ค่าเฉลี่ยระหว่างความแม่นยำ และความสามารถในการครอบคลุม (ค่าความแม่นยำ)
Precision	คือ	ความแม่นยำในการทำนายผลให้ถูกต้อง
Recall	คือ	ความสามารถในการครอบคลุมในการทำนายผลกลุ่ม Positive

2.6.4 ค่าความผิดพลาด

ค่าความผิดพลาด (Root Mean Square Error หรือ RMSE) เป็นเครื่องมือทางสถิติที่ใช้ในการประเมินประสิทธิภาพของแบบจำลองในงานการทำนายต่อเนื่อง (Regression) โดยการวัดความคลาดเคลื่อนระหว่างค่าที่แบบจำลองทำนาย (Predicted) และค่าจริง (Actual) ของตัวแปรตาม (Dependent Variable) ในชุดข้อมูลสำหรับทดสอบค่า RMSE สามารถคำนวณโดยใช้สมการดังนี้

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (\text{Predicted}_i - \text{Actual}_i)^2}{N}} \quad (2.12)$$

เมื่อ

RMSE	คือ	ค่าความผิดพลาด
Predicted	คือ	ค่าที่ได้จากการทำนาย
Actual	คือ	ค่าที่ได้จากความจริง
N	คือ	จำนวนตัวอย่างในชุดการทดสอบ

2.7 งานวิจัยที่เกี่ยวข้อง

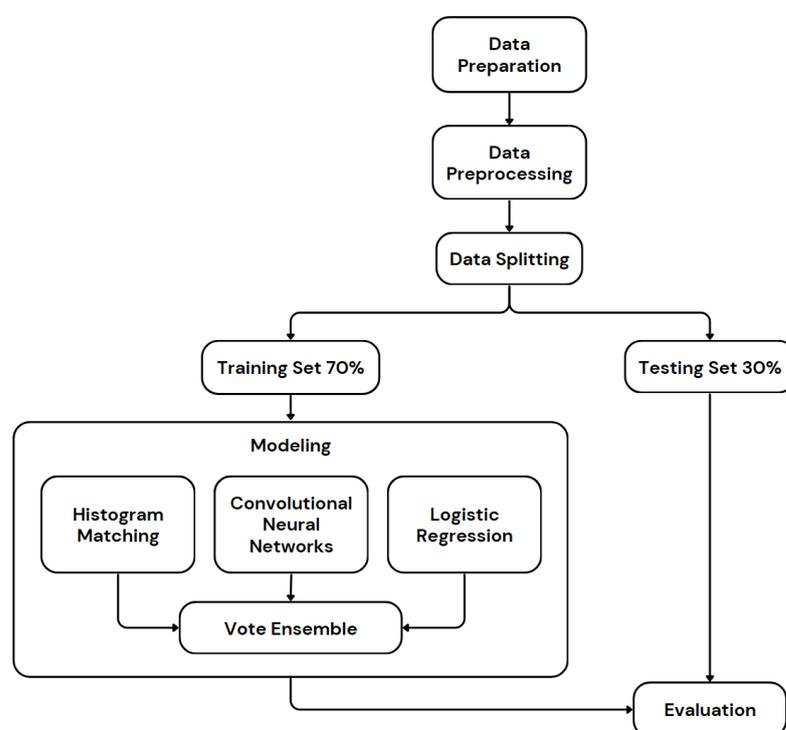
โอฬาริก สุรินตะ และกวีพจน์ บันลือวงศ์ [14] เรื่อง “การรู้จำตัวอักษรลายมือเขียนโดยใช้ตัวจำแนกข้อมูลด้วยวิธีการคำนวณเพื่อนบ้านใกล้ที่สุด k ตำแหน่ง” งานวิจัยนี้นำเสนอการนำวิธีการคำนวณเพื่อนบ้านใกล้ที่สุด k ตำแหน่ง (k-Nearest Neighbors: kNN) เพื่อใช้ในงานด้านการรู้จำตัวอักษรลายมือเขียน โดยมีขั้นตอนวิธี kNN ทำโดยนำข้อมูลมาเปรียบเทียบเพื่อหาค่าความใกล้เคียงระหว่างข้อมูลที่ต้องการจำแนก และนำข้อมูลที่มีค่าระยะทางน้อยที่สุดจำนวน k ข้อมูลมาพิจารณาจัดกลุ่ม มีอัตราการรู้จำ มากกว่า 90% ซึ่งชุดข้อมูลที่ใช้ในงานวิจัยทางด้านตัวอักษรลายมือเขียนได้เก็บรวบรวม ตัวอักษรลายมือเขียนที่มีลักษณะที่แตกต่างกัน เช่น โครงสร้างของตัวอักษรที่มีความคล้ายคลึงกัน การเขียนตัวอักษรที่มีความหลากหลาย ตัวอักษรที่เขียนไม่สมบูรณ์ หรือแม้กระทั่งข้อมูลมีจำนวนไม่เพียงพอ เป็นต้น โดยได้มีการทดสอบกับอักษร MNIST แบ่งเป็นชุดข้อมูลสำหรับการเรียนรู้ 60,000 อักษร และชุดทดสอบ 10,000 อักษร แบ่งรูปเป็น 32x32 พิกเซล มีอัตราการรู้จำ 87.80% อักษร Farsi แบ่งเป็นชุดข้อมูลสำหรับการเรียนรู้ 60,000 อักษร และชุดทดสอบ 20,000 อักษร แบ่งรูปเป็น 32x32 พิกเซล มีอัตราการรู้จำ 91.50% อักษร Gurmukhi แบ่งเป็นชุดข้อมูลสำหรับการเรียนรู้ 3,150 อักษร และชุดทดสอบ 350 อักษร แบ่งรูปเป็น 100x100 พิกเซล มีอัตราการรู้จำ 94.12% และอักษร Thai Character แบ่งเป็นชุดข้อมูลสำหรับการเรียนรู้ 5,310 อักษร และชุดทดสอบ 590 อักษร แบ่งรูปเป็น 40x40 พิกเซล มีอัตราการรู้จำ 88.00%

กันต์กมล วงศ์กา [15] เรื่อง “เทคนิคฮิสโตแกรมแมทชิงสำหรับการรู้จำและวิเคราะห์ตัวอักษรล้านนา” งานวิจัยนี้นำเสนอการนำวิธีเทคนิคฮิสโตแกรมแมทชิง (Histogram Matching) เพื่อใช้ในงานด้านการรู้จำและวิเคราะห์ตัวอักษรล้านนาด้วยกระบวนการประมวลผลภาพ โดยมีขั้นตอนการรู้จำพยัญชนะ โดยมีข้อมูลนำเข้าเป็นภาพถ่ายพยัญชนะล้านนาด้วยกล้องดิจิทัล โดยผ่านกระบวนการเบื้องต้นเพื่อทำการปรับปรุงคุณภาพของภาพด้วยการแยกพยัญชนะ ร่วมกับหลายเทคนิคดังนี้ การกำหนดค่าเทรชโฮลด์แบบทั้งภาพ (Global Thresholding) การหาขอบ (Edge Detection) การติดตามขอบวัตถุ (Contour Following) กำหนดค่าเทรชโฮลด์แบบเฉพาะที่ (Local Thresholding) การปรับขนาดพยัญชนะ (Resize Character) และการเปรียบเทียบภาพ (Histogram Matching) ซึ่งจากการทดลองอ่านพยัญชนะล้านนาด้วยเทคนิคฮิสโตแกรมแมทชิงพบว่ามีความแม่นยำเฉลี่ยเท่ากับ 80.95%

บทที่ 3

วิธีดำเนินการวิจัย

วิธีการวิจัยในการศึกษานี้นำเสนอการรู้จำลายมือเขียนอักษรล้านนา ด้วยเทคนิคการรวมแบบจำลองโหวต เป็นการศึกษาโดยใช้รูปแบบการวิจัยหลายรูปแบบมาประยุกต์เข้าด้วยกัน ประกอบด้วยการจัดเตรียมชุดข้อมูล การประมวลผลข้อมูลล่วงหน้า การแบ่งชุดข้อมูล การสร้างแบบจำลอง การรวมแบบจำลองโหวต และการประเมินผล



ภาพที่ 3.1 ขั้นตอนการสร้างแบบจำลองและประเมินผลแบบจำลอง

จากภาพที่ 3.1 เป็นขั้นตอนการสร้างแบบจำลอง โดยเริ่มจาก Data Preparation นำข้อมูลมาประมวลผลที่ Data Preprocessing และ Data Splitting แบ่งชุดข้อมูลที่ได้จาก Data Preprocessing เป็น Training Set 70% นำไป Modeling แบบจำลอง Histogram Matching Convolutional Neural Network และ Logistic Regression หลังจากนั้นนำ แบบจำลองทั้งสามรวมกันเป็นแบบจำลอง Vote Ensemble และแบ่งชุดข้อมูลที่เหลือเป็น Testing Set 30% นำไป Evaluation

3.1 การเตรียมข้อมูล

การเตรียมข้อมูล (Data Preparation) เป็นกระบวนการเตรียม การรวบรวมข้อมูลที่จะนำมาใช้ และจัดระเบียบข้อมูลให้เหมาะสมสำหรับการสร้างแบบจำลอง เพื่อให้แบบจำลองมีความแม่นยำมากขึ้น

จากภาพที่ 3.1 Data Preparation ได้ดำเนินการรวบรวมข้อมูล เนื่องจากชุดข้อมูลภาพเอกสารของอักษรล้านนาไม่เป็นที่นิยมจึงไม่มีชุดข้อมูลมาตรฐานสำหรับการทดลอง จึงต้องจัดเตรียมชุดข้อมูลเอง โดยการจัดเตรียมสร้างชุดข้อมูลภาพเอกสารอักษรล้านนาจากแบบอักษรล้านนา และแบบอักษรล้านนาลายมือเขียน ตัวอักษรที่ปรากฏบนชุดข้อมูลประกอบด้วยตัวอักษรภาษาล้านนารวม 42 ตัวอักษร

ตารางที่ 3.1 รูปแบบตัวอักษรที่นำมาใช้และตัวอย่างรูปแบบตัวอักษร

รูปแบบตัวอักษร	ตัวอย่าง
TH SarabunPSK	สวัสดีครับ
Cr-Father by Berm	ວໍວໍຊີວໍຣໍບ
Cr-Insom by Berm	ວໍວໍຊີວໍຣໍບ
Cr-Paitoon by Berm	ວໍວໍຊີວໍຣໍບ
LN TILOK	ວໍວໍຊີວໍຣໍບ
LN Mon Saen	ວໍວໍຊີວໍຣໍບ

ชุดข้อมูลแบบอักษรล้านนาจัดทำบนรูปแบบตัวอักษรหลากหลายลักษณะ จำนวน 5 รูปแบบตัวอักษรจากตารางที่ 3.1

ชุดข้อมูลแบบอักษรล้านนาลายมือเขียนจัดทำโดยนักเรียนมัธยมศึกษาที่มีลักษณะรูปแบบตัวอักษรแตกต่างกัน จำนวน 58 คน

ชุดข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลานนำมาจากฐานข้อมูลของ Digital Library of Northern Thai Manuscripts ที่เป็นฐานเก็บรวบรวมข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลาน โดยคัดกรองจากข้อกำหนด คือ ความครบถ้วนของชุดข้อมูลที่จะนำมาสอนแบบจำลองอ้างอิงความสมบูรณ์ของตัวอักษร จำนวน 7 ชุดข้อมูล

ตารางที่ 3.2 การกำหนดเครื่องหมายให้กับตัวอักษรล้านนา

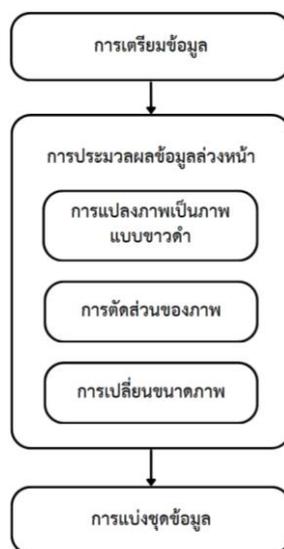
เครื่องหมาย	ตัวอักษรล้านนา
LN001	ဝ
LN002	ဗ
LN003	၎
LN004	ဃ
LN005	၎
LN006	ဝ
LN007	၎
LN008	၎
LN009	၎
LN010	ဝ
LN011	ဃ
LN012	၎
LN013	ဃ
LN014	၎
LN015	ဝ
LN016	၎
LN017	ဝ
LN018	၎
LN019	ဝ
LN020	၎
LN021	ဝ
LN022	ဝ
LN023	၎
LN024	ဝ
LN025	၎
LN026	ဝ
LN027	၎

เครื่องหมาย	ตัวอักษรล้านนา
LN028	๑
LN029	๒
LN030	๓
LN031	๔
LN032	๕
LN033	๖
LN034	๗
LN035	๘
LN036	๙
LN037	๐
LN038	๑
LN039	๒
LN040	๓
LN041	๔
LN042	๕

การจัดเตรียมชุดข้อมูลภาพเอกสาร จัดทำโดยการแปลงไฟล์เอกสารเป็นไฟล์ภาพ หน้ากระดาษตัวอักษรที่มีความละเอียดภาพ 300 dpi (Dot Per Inch) และถ่ายภาพชุดข้อมูลที่มีความละเอียดภาพ 5312x2988 พิกเซล ในพื้นที่ที่มีแสงอย่างเหมาะสม จากนั้นนำไฟล์ภาพ หน้ากระดาษตัวอักษรไปผ่านเครื่องมือทำเครื่องหมายภาพ โดยใช้ Web Application Roboflow นำมาใช้ในการกำหนดเครื่องหมายให้ตัวอักษรล้านนาในไฟล์ภาพหน้ากระดาษตัวอักษร 42 จากตารางที่ 3.2

3.2 การประมวลผลข้อมูลล่วงหน้า

การประมวลผลข้อมูลล่วงหน้า (Data Preprocessing) เป็นขั้นตอนที่สำคัญเพื่อให้ข้อมูลมีลักษณะที่เหมาะสมพร้อมสำหรับการสร้างแบบจำลอง และประกอบด้วยขั้นตอน การแปลงภาพเป็นภาพขาวดำ การตัดส่วนของภาพ และการเปลี่ยนขนาดภาพ



ภาพที่ 3.2 ขั้นตอนการประมวลผลข้อมูลล่วงหน้า

หลังจากได้ชุดข้อมูลรูปภาพที่มีการกำหนดเครื่องหมาย เพื่อนำชุดข้อมูลรูปภาพมาสร้างแบบจำลอง ต้องทำการแปลงชุดข้อมูลรูปภาพให้เป็นรูปแบบที่เหมาะสม ด้วยการประมวลผลข้อมูลล่วงหน้าประกอบด้วย 3 ขั้นตอน จากภาพที่ 3.2 ได้แก่ การแปลงภาพเป็นภาพแบบขาวดำ การตัดส่วนของภาพ และการเปลี่ยนขนาดภาพ อธิบายรายละเอียดของแต่ละขั้นตอนดังต่อไปนี้

3.2.1 การแปลงภาพเป็นภาพแบบขาวดำ

การแปลงภาพเป็นภาพแบบขาวดำ (Binarization) เป็นกระบวนการแปลงภาพระดับสีเทาให้เป็นภาพขาวดำ กระบวนการนี้ช่วยในการจำแนกวัตถุหรือลายเส้นในภาพที่ต้องการ โดยเฉพาะกรณีที่ข้อมูลสีไม่จำเป็นต่อการวิเคราะห์

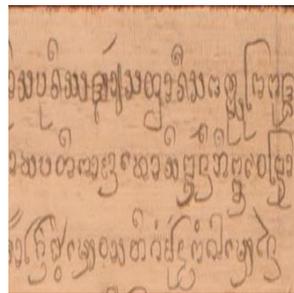
ตารางที่ 3.3 การกำหนดพารามิเตอร์ของสามเทคนิควิธีการแปลงภาพเป็นภาพแบบขาวดำ

วิธีการ	พารามิเตอร์
Niblack's	Window_size = 50, k = -0.2
Nick	Window_size = 50, k = -0.2
Otsu thresholding	ไม่มีพารามิเตอร์

พารามิเตอร์ Window_size คือ ขนาดของพื้นที่ใกล้เคียงแต่ละพิกเซลที่ใช้ในการคำนวณค่าเทรชโฮลด์ ซึ่งสามารถช่วยจัดการกับความแปรปรวนของแสง และสัญญาณรบกวนได้

พารามิเตอร์ k คือ พารามิเตอร์ควบคุมความไวของค่าเทรชโฮลด์ ต่อการเปลี่ยนแปลงความเข้มของแต่ละพื้นที่ใกล้เคียง ควรกำหนดค่า k เป็นบวกเมื่อต้องการกับภาพที่มีแสงสม่ำเสมอ และกำหนดที่ค่า k เป็นลบเมื่อต้องการกับภาพที่มีแสงสว่างต่างกัน

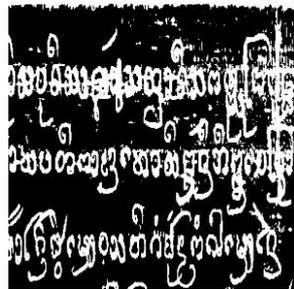
นำภาพเอกสารมาแปลงภาพเป็นภาพขาวดำเพื่อลดสัญญาณรบกวนที่ไม่ใช่ตัวอักษรออกจากภาพเอกสาร ในการศึกษาวิจัยนี้ได้ทดลองใช้เทคนิคการแปลงภาพเป็นภาพแบบขาวดำ 3 วิธี ได้แก่ วิธี Niblack's วิธี Nick และวิธี Otsu thresholding มีการกำหนดค่าจากตารางที่ 3.3



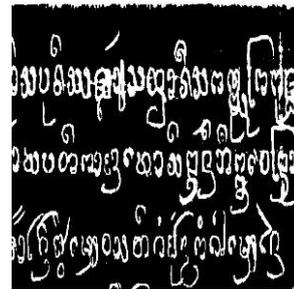
(ก)



(ข)



(ค)



(ง)

ภาพที่ 3.3 (ก) ภาพต้นฉบับ ภาพที่ผ่านการแปลงภาพเป็นภาพขาวดำโดยใช้ (ข) วิธีการ Niblack's (ค) วิธีการ Nick และ (ง) วิธีการ Otsu thresholding

จากการสังเกตภาพที่ผ่านการแปลงภาพเป็นภาพแบบขาวดำจากภาพที่ 3.3 วิธีการ Niblack's และ Otsu thresholding ดูเหมาะสมในการแปลงภาพเอกสารโบราณเป็นภาพแบบขาวดำ วิธีการทั้งสองมีประสิทธิภาพในแยกแยะตัวอักษรจากภาพพื้นหลัง และลดสัญญาณรบกวนจากภาพ แต่วิธีการ Otsu thresholding มีแนวโน้มที่จะแปลงภาพแบบขาวดำที่บางส่วนของภาพมีตัวอักษรขนาดบาง และขาดหายไม่ชัดเจน จึงตัดสินใจใช้วิธีการ Niblack's เพื่อแปลงภาพเอกสารโบราณเป็นภาพแบบขาวดำ

3.2.2 การตัดส่วนของภาพ

การตัดส่วนของภาพ (Cropping) เป็นกระบวนการที่ใช้ในการลดขนาดของภาพหรือลบส่วนที่ไม่จำเป็นออก เหลือเฉพาะส่วนที่สนใจ เพื่อลดความซับซ้อนของการวิเคราะห์ และช่วยเน้นข้อมูลที่สำคัญในการประมวลผล



(ก)



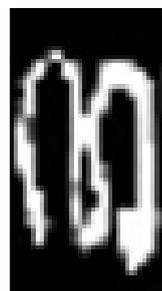
(ข)

ภาพที่ 3.4 (ก) ภาพที่ผ่านการแปลงภาพขาวดำ (ข) ภาพที่ผ่านการตัดส่วนของภาพ

หลังจากการนำภาพเอกสารภาพขาวดำ ผ่านการตัดส่วนของภาพจะได้ภาพตัวอักษรที่ลบส่วนที่ไม่จำเป็นออกเหลือเฉพาะส่วนที่สนใจจากภาพที่ 3.4 (ข)

3.2.3 การเปลี่ยนขนาดภาพ

การเปลี่ยนขนาดภาพ (Resizing) เป็นกระบวนการที่ใช้ในการปรับขนาดของภาพให้มีขนาดตามที่ต้องการ โดยการลด หรือขยายขนาดของภาพให้เป็นมิติที่ต้องการ การเปลี่ยนขนาดภาพเป็นกระบวนการสำคัญเนื่องจากมีผลต่อคุณภาพ และประสิทธิภาพของการประมวลผล



(ก)

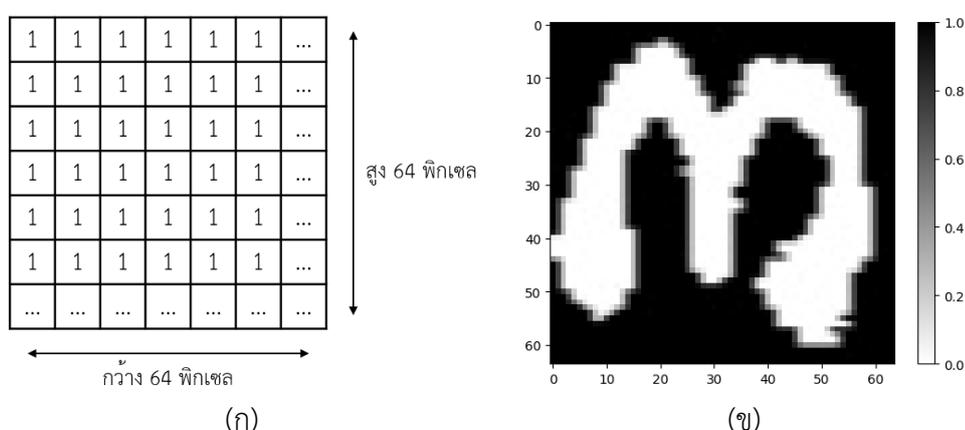


(ข)

ภาพที่ 3.5 (ก) ภาพที่ผ่านการครอบตัดภาพ (ข) ภาพที่ผ่านการเปลี่ยนขนาดภาพเป็นขนาด 64x64 พิกเซล

นำภาพที่ผ่านการตัดส่วนของภาพ ทำการเปลี่ยนขนาดภาพตัวอักษรเป็นขนาด 64x64 พิกเซลหรือขนาดเท่ากับ 4096 พิกเซล ให้เหมาะสมสำหรับการนำไปสร้างแบบจำลองจากภาพที่ 3.5 (ข)

เมื่อชุดข้อมูลรูปภาพผ่านการประมวลผลข้อมูลล่วงหน้าทั้ง 3 ขั้นตอนแล้ว ข้อมูลที่อยู่ในชุดข้อมูลจะมีลักษณะเป็นพิกเซล หรือที่เรียกว่าอาร์เรย์ 3 มิติ (Array 3D) ขนาด 64X64 และอาร์เรย์แต่ละช่องมีค่า (Value) เป็น 0 หรือ 1 ดังภาพ 3.6 (ก) เมื่อนำมาแสดงเป็นภาพด้วยการกำหนดเงื่อนไขของอาร์เรย์ที่มีค่าเป็น 0 เท่ากับสีขาว และมีค่าเป็น 1 เท่ากับสีดำ ได้ดังภาพ 3.6 (ข)



ภาพที่ 3.6 (ก) ภาพลักษณะของข้อมูล (ข) ภาพการกำหนดเงื่อนไขของอาร์เรย์ที่มีค่าเป็น 0 เท่ากับสีขาว และมีค่าเป็น 1 เท่ากับสีดำ

3.3 การแบ่งชุดข้อมูล

การแบ่งชุดข้อมูล (Data Splitting) เป็นกระบวนการที่สำคัญในการเตรียมข้อมูลสำหรับการฝึก และทดสอบแบบจำลอง เป้าหมายของการแบ่งชุดข้อมูล คือ การแยกข้อมูลออกเป็นส่วนย่อย เพื่อให้มีชุดข้อมูลสำหรับการฝึก (Training) และชุดข้อมูลสำหรับการทดสอบ (Testing) เพื่อประเมินแบบจำลอง (Evaluation) โดยนำข้อมูลที่ได้จากการประมวลผลข้อมูลล่วงหน้า มีข้อมูลภาพตัวอักษรจำนวน 3,408 ภาพตัวอักษร ประกอบด้วย ชุดข้อมูลภาพเอกสารจากเครื่องพิมพ์ ชุดข้อมูลภาพเอกสารลายมือเขียนบนกระดาษสีขาว และชุดข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลาน ดังตารางที่ 3.4 โดยแบ่งสัดส่วนชุดข้อมูลสำหรับการฝึก 70% และชุดข้อมูลสำหรับการทดสอบ 30% ด้วยการสุ่มข้อมูล

ตารางที่ 3.4 รายละเอียดชุดข้อมูลภาพเอกสาร

ชื่อชุดข้อมูล	จำนวนข้อมูล (ภาพตัวอักษร)
ชุดข้อมูลภาพเอกสารจากเครื่องพิมพ์	194
ชุดข้อมูลภาพเอกสารลายมือเขียนบนกระดาษสีขาว	2,980
ชุดข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลาน	234

จากตารางที่ 3.4 แสดงรายละเอียดชุดข้อมูลภาพเอกสารจากเครื่องพิมพ์มีจำนวน 194 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนบนกระดาษสีขาวมีจำนวน 2,980 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลานมีจำนวน 234 ภาพตัวอักษร

3.3.1 ชุดข้อมูลสำหรับฝึก

ชุดข้อมูลสำหรับฝึก (Training Set) เป็นชุดข้อมูลที่ใช้ในกระบวนการสร้าง หรือฝึกแบบจำลอง โดยนำชุดข้อมูลให้แบบจำลองเรียนรู้ ชุดข้อมูลสำหรับฝึกมีข้อมูลภาพตัวอักษรจำนวน 2,385 ภาพตัวอักษร ประกอบด้วย ชุดข้อมูลภาพเอกสารจากเครื่องพิมพ์มีจำนวน 136 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนบนกระดาษสีขาวมีจำนวน 2,086 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลานมีจำนวน 164 ภาพตัวอักษร

3.3.2 ชุดข้อมูลสำหรับทดสอบ

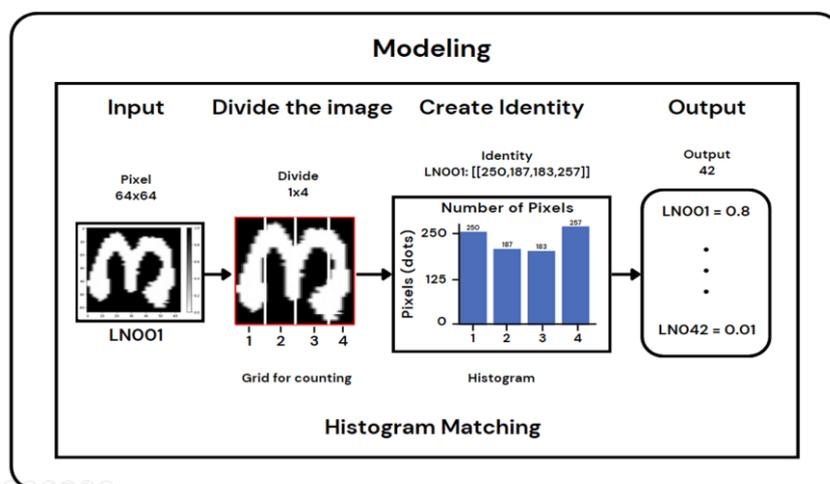
ชุดข้อมูลสำหรับทดสอบ (Testing Set) เป็นชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพแบบจำลองหลังจากแบบจำลองถูกฝึกเสร็จสิ้น ชุดข้อมูลนี้ไม่ได้ใช้ในกระบวนการฝึก จึงเป็นความลับต่อแบบจำลอง ชุดข้อมูลสำหรับทดสอบมีข้อมูลภาพตัวอักษรจำนวน 1,023 ภาพตัวอักษร ประกอบด้วย ชุดข้อมูลภาพเอกสารจากเครื่องพิมพ์มีจำนวน 58 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนบนกระดาษสีขาวมีจำนวน 894 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลานมีจำนวน 70 ภาพตัวอักษร

3.4 การสร้างแบบจำลอง

การสร้างแบบจำลอง (Modeling) เป็นกระบวนการสำคัญที่ทำให้แบบจำลองเรียนรู้ปรับพารามิเตอร์ เพื่อทำงานได้อย่างถูกต้อง และมีประสิทธิภาพในการรู้จำลายมือเขียนอักษรล้านนา

การศึกษาวิจัยนี้นำเสนอการสร้างแบบจำลองฮิสโตแกรมแมทซิง โครงข่ายประสาทเทียมแบบคอนโวลูชัน การถดถอยโลจิสติก และการรวมแบบจำลองโหวต

3.4.1 ฮิสโตแกรมแมทซิง

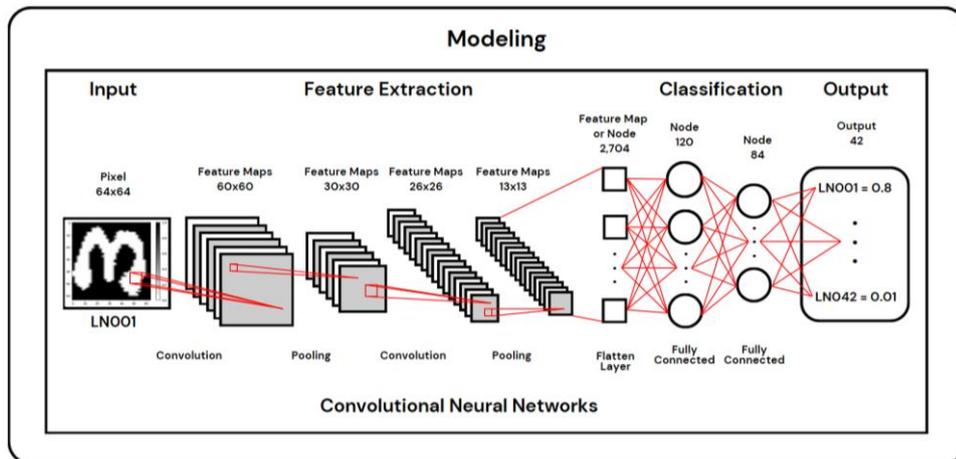


ภาพที่ 3.7 กระบวนการสร้างแบบจำลองฮิสโตแกรมแมทซิง

ฮิสโตแกรมแมทซิง (Histogram Matching : HM) จากภาพที่ 3.7 ประกอบด้วย การนำเข้าข้อมูล (Input) การแบ่งส่วนภาพ (Divide The Image) การสร้างเอกลักษณ์ (Create Identity) และการส่งออกข้อมูล (Output)

1. Input คือ การรับข้อมูลเข้ามา ซึ่งในภาพที่ 3.7 มี Input เป็นภาพขาวดำ มีลักษณะดังภาพที่ 3.6 (ก) เป็น Array 3 มิติ ขนาด 64x64 พิกเซล (64, 64, 1) แต่ละพิกเซลมีค่าระหว่าง 0 ถึง 1
2. Divide The Image คือ การแบ่งส่วนภาพแนวนอน (Row) หรือนอนตั้ง (Column) ซึ่งในภาพที่ 3.7 แบ่งส่วนภาพเป็นแนวนอน 1 แถวตั้ง 4 เพื่อกำหนดขอบเขตสำหรับการนับจำนวนพิกเซล
3. Create Identity คือ การนับจำนวนพิกเซลในแต่ละส่วนว่ามีจำนวนพิกเซลเท่าไร ซึ่งในภาพที่ 3.7 ส่วนที่ 1 มีจำนวนพิกเซล 250 ส่วนที่ 2 มีจำนวนพิกเซล 187 ส่วนที่ 3 มีจำนวนพิกเซล 183 และส่วนที่ 4 มีจำนวนพิกเซล 257 เมื่อนำมาแสดงเป็นกราฟฮิสโตแกรม แกนแนวนอน (X) คือ จำนวนส่วนของภาพ และแกนแนวตั้ง (Y) คือ จำนวนพิกเซล จากนั้นเก็บจำนวนพิกเซลของแต่ละส่วนไว้ใน Identity เพื่อกำหนดเอกลักษณ์ของตัวอักษร
4. Output คือ ขั้นตอนการนำผลลัพธ์จากการทำนายมาแสดงผลพร้อมกับค่าความถูกต้อง [เครื่องหมายจากตารางที่ 3.2, ค่าความถูกต้อง] ซึ่งในภาพที่ 3.7 มีจำนวนผลลัพธ์ 42 ผลลัพธ์ และค่าความถูกต้อง ได้ผลดังนี้ “[LN001,0.8], [..., ...], [LN042, 0.01]”

3.4.2 โครงข่ายประสาทเทียมแบบคอนโวลูชัน



ภาพที่ 3.8 กระบวนการสร้างแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน

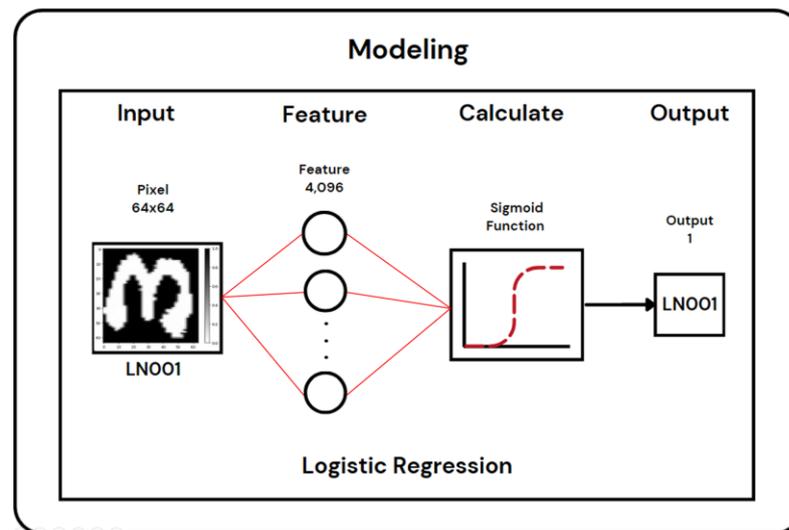
โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network : CNN) จากภาพที่ 3.8 ประกอบด้วย Input การสกัดคุณลักษณะ (Feature Extraction) การจัดหมวดหมู่ (Classification) และ Output ซึ่งมีการกำหนดพารามิเตอร์ตามแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน LeNet-5

1. Input คือ การรับข้อมูลเข้ามา ซึ่งในภาพที่ 3.8 มี Input เป็นภาพขาวดำ มีลักษณะดังภาพที่ 3.6 (ก) เป็น Array 3 มิติ ขนาด 64x64 พิกเซล (64, 64, 1) แต่ละพิกเซลมีค่าระหว่าง 0 ถึง 1
2. Feature Extraction คือ การสกัดคุณลักษณะ ประกอบด้วยขั้นตอน Convolution Pooling และ Flatten Layer ซึ่งในภาพที่ 3.8 มี Convolution ที่กำหนดพารามิเตอร์ Input 64x64 Kernel 5x5 Stride 1 ได้ Feature Maps ขนาด 60x60 มีขั้นตอนการคำนวณดังภาพที่ 2.8 Pooling แบบ AvgPooling ที่กำหนดพารามิเตอร์ Kernel 2x2 Stride 2 ได้ Feature Maps ขนาด 30x30 มีขั้นตอนการคำนวณดังภาพที่ 2.9 (ข) ทำขั้นตอน Convolution กับ Pooling อีกครั้งได้ Feature Maps ขนาด 13x13 และนำมาผ่าน Flatten Layer จะมีลักษณะเป็น Array 1 มิติ จะได้ Feature Maps ขนาด 2,704
3. Classification คือ การนำข้อมูลจากขั้นตอน Feature Extraction นำมาประมวลผลข้อมูล โดยใช้ Fully Connected Layer ประกอบด้วย Input Layer Hidden Layer และ Output Layer ซึ่งในภาพที่ 3.8 Input Layer มีโหนด จำนวน 2,704 โหนด เท่ากับจำนวนข้อมูลจาก Flatten Layer

มี Hidden Layer จำนวน 2 ชั้น โดยกำหนดชั้นที่ 1 มี 120 โหนด ชั้นที่ 2 มี 84 โหนด มีการเรียนรู้รูปแบบหรือความสัมพันธ์ระหว่างข้อมูล Input Layer กับข้อมูล Output Layer และประมวลผลข้อมูลโดยมีการคำนวณจากสมการที่ 2.8 ดังภาพที่ 2.10 เมื่อได้ผลลัพธ์มีจะส่งออกมา ไปยังชั้น Output Layer จำนวน 42 โหนด

4. Output คือ ขั้นตอนการนำผลลัพธ์จากขั้นตอน Classification มาแสดงผล พร้อมกับค่าจาก Activation Functions [เครื่องหมายจากตารางที่ 3.2, ค่าจาก Activation Functions] ซึ่งในภาพที่ 3.8 มีจำนวนผลลัพธ์ 42 ผลลัพธ์ และค่าจาก Activation Functions แบบ ReLU ได้ผลดังนี้ “[LN001,0.8], [..., ...], [LN042, 0.01]”

3.4.3 การถดถอยโลจิสติก



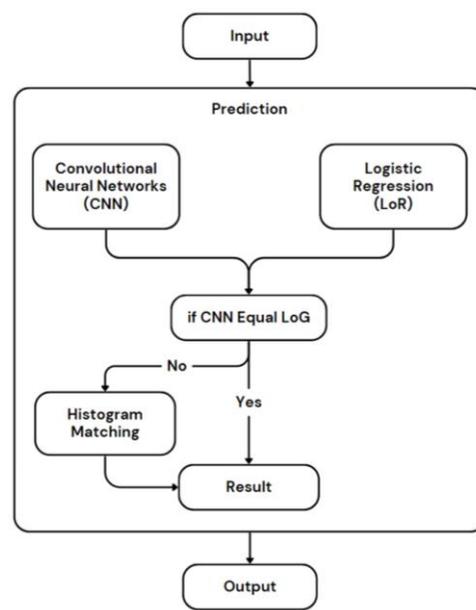
ภาพที่ 3.9 กระบวนการสร้างแบบจำลองการถดถอยโลจิสติก

การถดถอยโลจิสติก (Logistic Regression : LoR) จากภาพที่ 3.9 ประกอบด้วย Input คุณลักษณะ (Feature) ฟังก์ชันซิกมอยด์ (Sigmoid Function) และ Output

1. Input คือ การรับข้อมูลเข้ามา ซึ่งในภาพที่ 3.9 มี Input เป็นภาพขาวดำ มีลักษณะดังภาพที่ 3.6 (ก) เป็น Array 3 มิติ ขนาด 64x64 พิกเซล (64, 64, 1) แต่ละพิกเซลมีค่าระหว่าง 0 ถึง 1

2. Feature คือ การนำข้อมูลจาก Input ทำให้มีลักษณะเป็น Array 1 มิติ Feature เป็นส่วนสำคัญที่ทำให้แบบจำลองมีความแม่นยำ ซึ่งในภาพที่ 3.9 จะได้ Feature ขนาด 4,096
3. Calculate คือ การนำข้อมูล Feature มาประมวลผลข้อมูลด้วย Sigmoid Function เพื่อหาความน่าจะเป็นของข้อมูล โดยมีการคำนวณจากสมการที่ 2.5
4. Output คือ ขั้นตอนการนำผลลัพธ์จากขั้นตอน Calculate มาแสดงผลเครื่องหมายจากตารางที่ 3.2 ซึ่งในภาพที่ 3.9 มีจำนวนผลลัพธ์ 1 ผลลัพธ์ ได้ผลดังนี้ “LN001”

3.4.4 การรวมแบบจำลองโหวต



ภาพที่ 3.10 กระบวนการทำงานของการรวมแบบจำลองโหวต

การรวมแบบจำลองโหวต (Vote Ensemble) จากภาพที่ 3.10 ประกอบด้วย Input การทำนายผล (Prediction) และ Output

1. Input คือ การรับข้อมูลเข้ามา เป็นภาพขาวดำ มีลักษณะดังภาพที่ 3.6 (ก) เป็น Array 3 มิติ ขนาด 64x64 พิกเซล (64, 64, 1) แต่ละพิกเซลมีค่าระหว่าง 0 ถึง 1

2. Prediction คือ การนำข้อมูลจาก Input เข้าแบบจำลอง Convolutional Neural Network (CNN) และแบบจำลอง Logistic Regression (LoR) เมื่อสองแบบจำลองทำนายได้ผลลัพธ์ตรงกัน เช่น CNN ทำนาย LN001 และ LoR ทำนาย LN001 จะเข้าเงื่อนไข Yes จะส่งผลลัพธ์ไปยัง Output แต่เมื่อผลลัพธ์ไม่ตรงกัน เช่น CNN ทำนาย LN001 และ LoR ทำนาย LN002 จะเข้าเงื่อนไข No จะนำข้อมูลจาก Input เข้าแบบจำลอง Histogram Matching เมื่อแบบจำลอง Histogram Matching ทำนายได้ผลลัพธ์ และค่าความถูกต้อง เช่น [LN001,0.8], [LN002,0.1], [..., ...], [LN042, 0.01] นำผลลัพธ์ของ Histogram Matching ที่ตรงกับผลลัพธ์ของ CNN กับ LoR เปรียบเทียบ และทำการเลือกผลลัพธ์ที่มีค่าความถูกต้องมากกว่า คือ [LN001, 0.8] และส่งผลการทำนาย LN001 ไปยัง Output
3. Output คือ ขั้นตอนการนำผลลัพธ์จาก Prediction มาแสดงผล ได้ผลดังนี้ “LN001”

3.5 การประเมินผล

การประเมินผล (Evaluation) จากภาพที่ 3.1 เป็นขั้นตอนสำคัญในกระบวนการสร้างและพัฒนาประสิทธิภาพแบบจำลอง ด้วยการทดสอบคุณสมบัติ และประสิทธิภาพแบบจำลองในการทำนายข้อมูลที่ไม่เคยเห็นมาก่อนจากชุดข้อมูลสำหรับทดสอบ

เมื่อสร้างแบบจำลองจากชุดข้อมูลสำหรับฝึกจะนำแบบจำลองที่ได้มาทดสอบกับชุดข้อมูลสำหรับทดสอบ เพื่อประเมินผลประสิทธิภาพด้วยค่าความแม่นยำ F1-Score และ ค่าความผิดพลาด Root Mean Square Error (RMSE)

บทที่ 4

ผลการทดลอง

การศึกษาครั้งนี้ การรู้จำลายมือเขียนอักษรล้านนาด้วยแบบจำลองฮิสโตแกรมแมทซิ่ง ร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และการถดถอยโลจิสติก ผู้ศึกษาได้ดำเนินการวิเคราะห์ข้อมูล โดยนำเสนอเป็น 3 ตอนดังนี้

- 4.1 ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา ก่อนใช้แบบจำลองฮิสโตแกรมแมทซิ่ง ร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก
- 4.2 ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา หลังใช้แบบจำลองฮิสโตแกรมแมทซิ่ง ร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก
- 4.3 การพัฒนาประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา โดยใช้แบบจำลองฮิสโตแกรมร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก

4.1 ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา ก่อนใช้แบบจำลองฮิสโตแกรมแมทซิ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก

ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา ก่อนใช้แบบจำลองฮิสโตแกรมแมทซิ่ง ร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก เมื่อนำชุดข้อมูลสำหรับทดสอบ ประกอบด้วย ภาพเอกสารจากเครื่องพิมพ์มีจำนวน 58 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนบนกระดาษสีขาวมีจำนวน 894 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลานมีจำนวน 70 ภาพตัวอักษร เพื่อทดสอบแบบจำลองฮิสโตแกรมแมทซิ่ง และแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก

4.1.1 แบบจำลองฮิสโตแกรมแมทซิ่ง

ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา เมื่อประเมินผลการรู้จำด้วยแบบจำลองฮิสโตแกรมแมทซิ่ง (HM) จำนวน 5 แบบจำลอง โดยแต่ละแบบจำลองมีการแบ่งส่วนภาพแตกต่างกัน ดังตาราง 4.1 โดยใช้ชุดข้อมูลสำหรับทดสอบ ได้ผลประสิทธิภาพ ดังตารางที่ 4.2

ตารางที่ 4.1 การแบ่งส่วนภาพของแบบจำลองฮิสโตแกรมแมทซิ่ง

แบบจำลอง	การแบ่งส่วนภาพ (แนวนอน x แนวตั้ง)
HM	8x8
HM	12x12
HM	16x16
HM	20x20
HM	24x24

ตารางที่ 4.2 ร้อยละของประสิทธิภาพความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบของแบบจำลองฮิสโตแกรมแมทซิ่ง

แบบจำลอง	F1-Score (%)	RMSE (%)
HM (8x8)	81.52	42.98
HM (12x12)	87.59	35.23
HM (16x16)	87.59	35.23
HM (20x20)	86.61	36.60
HM (24x24)	78.40	46.48

ร้อยละของประสิทธิภาพความแม่นยำ (F1-Score) ในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาด (RMSE) ในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ด้วยแบบจำลองฮิสโตแกรมแมทซิ่ง จากตารางที่ 4.2 พบว่า ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง HM (8x8) มีค่าความแม่นยำแบบ F1-Score จากสมการที่ 2.11 81.52% และมีค่า RMSE จากสมการที่ 2.12 42.98% ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง HM (12x12) มีค่าความแม่นยำแบบ F1-Score 87.59% และมีค่า RMSE 35.23% ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง HM (16x16) มีค่าความแม่นยำแบบ F1-Score 87.59% และมีค่า RMSE 35.23% ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง HM (20x20) มีค่าความแม่นยำแบบ F1-Score 86.61% และมีค่า RMSE 36.60% ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง HM (24x24) มีค่าความแม่นยำแบบ F1-Score 78.40% และมีค่า RMSE 46.48%

4.1.2 แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน

ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา เมื่อประเมินผลการรู้จำด้วยแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) โดยใช้ชุดข้อมูลสำหรับทดสอบ ได้ผลประสิทธิภาพ ดังตารางที่ 4.3

ตารางที่ 4.3 ร้อยละของประสิทธิภาพความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ของแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน

แบบจำลอง	F1-Score (%)	RMSE (%)
CNN	93.84	24.82

ร้อยละของประสิทธิภาพความแม่นยำ ในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ด้วยแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน จากตารางที่ 4.3 พบว่า ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง CNN มีความแม่นยำแบบ F1-Score 93.84% และมีค่า RMSE 24.82%

4.1.3 แบบจำลองการถดถอยโลจิสติก

ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา เมื่อประเมินผลการรู้จำด้วยแบบจำลองการถดถอยโลจิสติก (LoR) โดยใช้ชุดข้อมูลสำหรับทดสอบ ได้ผลประสิทธิภาพ ดังตาราง ที่ 4.4

ตารางที่ 4.4 ร้อยละของประสิทธิภาพความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ของแบบจำลองการถดถอยโลจิสติก

แบบจำลอง	F1-Score (%)	RMSE (%)
LoR	94.53	23.40

ร้อยละของประสิทธิภาพความแม่นยำ ในการรู้จำลายมือเขียนอักษรล้านนา และ ร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ด้วยแบบจำลองการถดถอยโลจิสติก (LoR) จากตารางที่ 4.4 พบว่า ประสิทธิภาพการรู้จำลายมือเขียน อักษรล้านนาของแบบจำลอง LoR มีค่าความแม่นยำแบบ F1-Score 93.84% และมีค่า RMSE 24.82%

4.2 ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา หลังใช้แบบจำลองฮิสโตแกรมแมทซึ่งร่วมกับ แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก

ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา หลังใช้แบบจำลองฮิสโตแกรมแมทซึ่ง ร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก หรือหลังใช้การรวมแบบจำลองโหวต ดังหัวข้อการรวมแบบจำลองโหวตที่ 3.4.4 ภาพที่ 3.10 เมื่อนำ ชุดข้อมูลสำหรับทดสอบ ประกอบด้วย ภาพเอกสารจากเครื่องพิมพ์มีจำนวน 58 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนบนกระดาษสีขาวมีจำนวน 894 ภาพตัวอักษร ชุดข้อมูลภาพ เอกสารลายมือเขียนอักษรล้านนาบนใบลานมีจำนวน 70 ภาพตัวอักษร เพื่อทดสอบการรวม แบบจำลองโหวต

4.2.1 การรวมแบบจำลองโหวต

ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา เมื่อประเมินผลการรู้จำ ด้วยการรวมแบบจำลองโหวต (VE) จำนวน 5 แบบจำลอง โดยแต่ละแบบจำลองมีการรวมกับ แบบจำลองที่แตกต่างกัน ดังตาราง 4.5 โดยใช้ชุดข้อมูลสำหรับทดสอบ ได้ผลประสิทธิภาพ ดังตารางที่ 4.6

ตารางที่ 4.5 รายชื่อแบบจำลองที่ทำการรวมแบบจำลองโหวต

แบบจำลอง	รวมกับแบบจำลอง
VE (8x8)	HM (8x8), CNN, LoR
VE (12x12)	HM (12x12) , CNN, LoR
VE (16x16)	HM (16x16) , CNN, LoR
VE (20x20)	HM (20x20) , CNN, LoR
VE (24x24)	HM (24x24) , CNN, LoR

ตารางที่ 4.6 ร้อยละของประสิทธิภาพความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ของการรวมแบบจำลองโหวต

แบบจำลอง	F1-Score (%)	RMSE (%)
VE (8x8)	95.11	22.11
VE (12x12)	95.21	21.89
VE (16x16)	95.31	21.66
VE (20x20)	95.31	21.66
VE (24x24)	95.11	22.11

ร้อยละของประสิทธิภาพความแม่นยำ ในการรู้จำลายมือเขียนอักษรล้านนา และร้อยละของค่าความผิดพลาดในการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบ ด้วยการรวมแบบจำลองโหวต จากตารางที่ 4.6 พบว่า ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง VE (8x8) มีค่าความแม่นยำแบบ F1-Score 95.11% และมีค่า RMSE 22.11% ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง VE (12x12) มีค่าความแม่นยำแบบ F1-Score 95.21% และมีค่า RMSE 21.89% ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง VE (16x16) มีค่าความแม่นยำแบบ F1-Score 95.31% และมีค่า RMSE 21.66% ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง VE (20x20) มีค่าความแม่นยำแบบ F1-Score 95.31% และมีค่า RMSE 21.66% ประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลอง VE (24x24) มีค่าความแม่นยำแบบ F1-Score 95.11% และมีค่า RMSE 22.11%

4.3 การพัฒนาประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา โดยใช้แบบจำลองฮีโตนแกรมร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก

การพัฒนาประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนาก่อน และหลังใช้แบบจำลองฮีโตนแกรมร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก หรือก่อน และหลังการรวมแบบจำลองโหวต พัฒนาประสิทธิภาพการรู้จำลายมือเขียนอักษรล้านนา โดยการนำชุดข้อมูลสำหรับทดสอบ ประกอบด้วย ภาพเอกสารจากเครื่องพิมพ์มีจำนวน 58 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนบนกระดาษสีขาวมีจำนวน 894 ภาพตัวอักษร ชุดข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลานมีจำนวน 70 ภาพตัวอักษร เพื่อทดสอบแบบจำลองฮีโตนแกรมแมทซิง (HM) แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN)

แบบจำลองการถดถอยโลจิสติก (LoR) และการรวมแบบจำลองโหวต (VE) แสดงผลประสิทธิภาพดังตารางที่ 4.7

ตารางที่ 4.7 ร้อยละของประสิทธิภาพความแม่นยำในการรู้จำลายมือเขียนอักษร กับชุดข้อมูลสำหรับทดสอบ ของแบบจำลองฮิสโตแกรมแมทซึ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน แบบจำลองการถดถอยโลจิสติก และการรวมแบบจำลองโหวต

F1-Score ก่อนใช้ VE		F1-Score หลังใช้ VE		F1-Score เพิ่มขึ้น (%)
แบบจำลอง	F1-Score (%)	แบบจำลอง	F1-Score (%)	
HM (8x8)	81.52	VE (8x8)	95.11	13.59
HM (12x12)	87.59	VE (12x12)	95.21	7.62
HM (16x16)	87.59	VE (16x16)	95.31	7.72
HM (20x20)	86.61	VE (20x20)	95.31	8.7
HM (24x24)	78.40	VE (24x24)	95.11	16.71
CNN	93.84	VE (16x16)	95.31	1.47
LoR	94.53	VE (16x16)	95.31	0.78

จากตารางที่ 4.7 แบบจำลองก่อนใช้ VE ของแบบจำลอง CNN และ LoR เปรียบเทียบกับแบบจำลองหลังใช้ VE (16x16) เพราะแบบจำลอง CNN และ LoR ไม่มีการแบ่งส่วนภาพ จึงเปรียบเทียบกับแบบจำลอง VE (16x16) ที่มีค่า F1-Score มากที่สุด

พบว่าผลการรู้จำลายมือเขียนอักษรล้านนา กับชุดข้อมูลสำหรับทดสอบของแบบจำลองหลังใช้ VE เปรียบเทียบกับผลการรู้จำลายมือเขียนอักษรล้านนาของแบบจำลองก่อนใช้ VE โดยแบบจำลอง HM (8x8) มีค่า F1-Score เพิ่มขึ้น 13.59% แบบจำลอง HM (12x12) มีค่า F1-Score เพิ่มขึ้น 7.62% แบบจำลอง HM (16x16) มีค่า F1-Score เพิ่มขึ้น 7.72% แบบจำลอง HM (20x20) มีค่า F1-Score เพิ่มขึ้น 8.7% แบบจำลอง HM (24x24) มีค่า F1-Score เพิ่มขึ้น 16.71% มากที่สุด แบบจำลอง CNN มีค่า F1-Score เพิ่มขึ้น 1.47% และแบบจำลอง LoR มีค่า F1-Score เพิ่มขึ้น 0.78% น้อยที่สุด

บทที่ 5

สรุปผลการวิจัย

การศึกษานี้ แก้ไขปัญหาหนังสือใบลาน และศิลาจารึกที่ใช้ในการบันทึกข้อมูลภาษาล้านนาผู้ฟังได้ง่าย แก้ไขด้วยเทคนิคการรวมแบบจำลองโหวต เพื่อสร้างแบบจำลองที่มีความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนา โดยทดสอบกับชุดข้อมูลภาพเอกสารจากเครื่องพิมพ์ชุดข้อมูลภาพเอกสารลายมือเขียนบนกระดาษสีขาว และชุดข้อมูลภาพเอกสารลายมือเขียนอักษรล้านนาบนใบลาน จำนวน 3,408 ภาพตัวอักษร แบบจำลองฮีโตแกรมแมทซิ่ง (HM 16x16) ได้ผลความแม่นยำแบบ F1-Score 87.59 ในขณะที่แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) ได้ผลความแม่นยำแบบ F1-Score 93.84% และแบบจำลองการถดถอยโลจิสติก (LoR) ได้ผลความแม่นยำแบบ F1-Score 94.53% เมื่อนำแบบจำลองฮีโตแกรมแมทซิ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน และแบบจำลองการถดถอยโลจิสติก (VE 16x16) ได้ผลความแม่นยำแบบ F1-Score 95.31% โดยมีค่าความแม่นยำแบบ F1-Score มากกว่าแบบจำลอง HM (16x16) 7.72% แบบจำลอง CNN 1.47% และแบบจำลอง LoR 0.78% จึงสรุปได้ว่าเทคนิคการรวมแบบจำลองโหวต ทำให้ความแม่นยำในการรู้จำลายมือเขียนอักษรล้านนาเพิ่มขึ้น

5.1 สรุปผลการทดลอง

สรุปผลการทดลองโดยเลือกแบบจำลองก่อนใช้แบบจำลองฮีโตแกรมแมทซิ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียม และแบบจำลองการถดถอยโลจิสติก แบบจำลองฮีโตแกรมแมทซิ่งเลือกใช้ HM (16x16) ในการสรุปผลการทดลองเนื่องจากมีค่าความแม่นยำแบบ F1-Score มากที่สุด ส่วนแบบจำลองหลังใช้แบบจำลองฮีโตแกรมแมทซิ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียม และแบบจำลองการถดถอยโลจิสติก แบบจำลองการรวมแบบโหวตเลือกใช้ VE (16x16)

1. ผลความแม่นยำก่อนใช้แบบจำลองฮีโตแกรมแมทซิ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียม และแบบจำลองการถดถอยโลจิสติก มีค่าความแม่นยำแบบ F1-Score ของแบบจำลอง HM (16x16) 87.59% แบบจำลอง CNN 93.84% และแบบจำลอง LoR 94.53%
2. ผลความแม่นยำหลังใช้แบบจำลองฮีโตแกรมแมทซิ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียม และแบบจำลองการถดถอยโลจิสติก VE (16x16) ในการรู้จำลายมือเขียนอักษรล้านนา มีค่าความแม่นยำแบบ F1-Score 95.31%

3. ผลความแม่นยำก่อนใช้แบบจำลองฮิสโตแกรมแมทซึ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียม และแบบจำลองการถดถอยโลจิสติก VE (16x16) เปรียบเทียบกับหลังใช้ พบว่ามีค่าความแม่นยำแบบ F1-Score มากกว่าแบบจำลอง HM (16x16) 7.72% แบบจำลอง CNN 1.47% และแบบจำลอง LoR 0.78%

5.2 อภิปรายผลการวิจัย

อภิปรายผลการวิจัยโดยเลือกแบบจำลองก่อนใช้แบบจำลองฮิสโตแกรมแมทซึ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียม และแบบจำลองการถดถอยโลจิสติก แบบจำลองฮิสโตแกรมแมทซึ่งเลือกใช้ HM (16x16) ในการสรุปผลการทดลองเนื่องจากมีค่าความแม่นยำแบบ F1-Score มากที่สุด ส่วนแบบจำลองหลังใช้แบบจำลองฮิสโตแกรมแมทซึ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียม และแบบจำลองการถดถอยโลจิสติก หรือแบบจำลองการรวมแบบโหวตเลือกใช้ VE (16x16)

1. จากผลการวิจัยพบว่าแบบจำลองฮิสโตแกรมแมทซึ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียม และแบบจำลองการถดถอยโลจิสติกหรือการรวมแบบจำลองโหวต (VE 16x16) มีความแม่นยำแบบ F1-Score มากที่สุด ทั้งนี้อาจเป็นเพราะว่าแบบจำลองฮิสโตแกรมแมทซึ่ง (HM 16x16) แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) และแบบจำลองการถดถอยโลจิสติก (LoR) มีค่าความแม่นยำของแต่ละตัวอักษรไม่เท่ากัน ซึ่งสอดคล้องกับค่าความแม่นยำแบบ F1-Score ของการรวมแบบจำลองโหวต โดยวิธีการรวมแบบจำลองโหวต เป็นเทคนิคที่ทำให้แต่ละแบบจำลองช่วยกันตรวจสอบผลการทำนายเพื่อเพิ่มความแม่นยำในการทำนายที่ส่งผลต่อการรู้จำลายมือเขียนอักษรล้านนา
2. จากผลการวิจัยพบว่าแบบจำลองฮิสโตแกรมแมทซึ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียม และแบบจำลองการถดถอยโลจิสติกหรือแบบจำลองโหวตมีความแม่นยำแบบ F1-Score มากที่สุด สอดคล้องกับผลความแม่นยำแบบ F1-Score ของการรวมแบบจำลองโหวต
3. จากผลการวิจัยพบว่าแบบจำลองฮิสโตแกรมแมทซึ่งร่วมกับแบบจำลองโครงข่ายประสาทเทียม และแบบจำลองการถดถอยโลจิสติก มีค่าความแม่นยำแบบ F1-Score มากที่สุด ทั้งนี้อาจเป็นเพราะว่าวิธีการรวมแบบจำลองโหวต เป็นเทคนิคที่ทำให้แต่ละแบบจำลองช่วยกันตรวจสอบผลการทำนายเพื่อเพิ่มความแม่นยำในการทำนายที่ส่งผลต่อการรู้จำลายมือเขียนอักษรล้านนา

5.3 ข้อเสนอแนะการวิจัย

5.3.1 ข้อเสนอแนะในการนำผลการวิจัยไปใช้

การจะใช้เครื่องมือในการรู้จำลายมือเขียนอักษรล่านาของการวิจัยเล่มนี้ จำเป็นต้องการมีโปรแกรมสำหรับทำงานโค้ดภาษา Python

5.3.2 ข้อเสนอแนะในการทำวิจัยครั้งต่อไป

1. พัฒนาเครื่องมือในการตรวจจับลายมือเขียนอักษรล่านา และแอปพลิเคชัน เพื่อให้สามารถใช้งานได้สะดวกมากขึ้น
2. พัฒนาปรับปรุงเทคนิค วิธีการ และแนวคิดสำหรับเพิ่มประสิทธิภาพของแบบจำลองการรู้จำลายมือเขียนอักษรล่านา
3. พัฒนาให้สามารถรู้จักรอักษรล่านาที่มากขึ้น เช่น สามารถรู้จำอักษรที่เป็นเสียงสระ 8 ตัวอักษร เป็นต้น

บรรณานุกรม

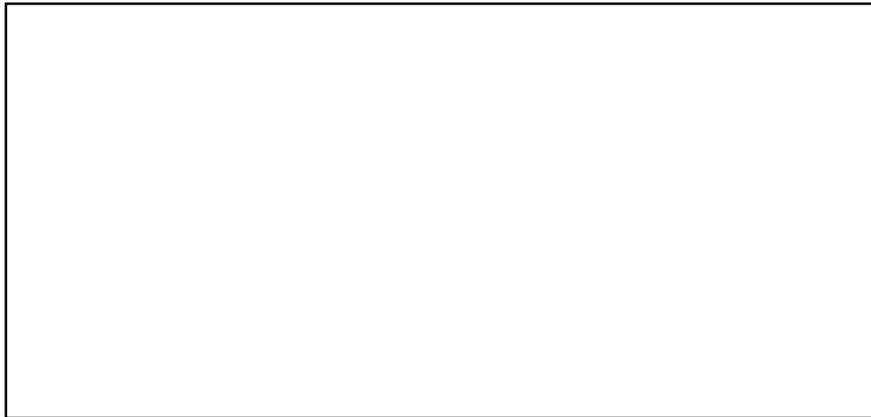
- [1] หนังสือเรียนอักษรธัมมล้านนา. [ออนไลน์]. ได้จาก:
<https://www.culture.cmru.ac.th/web60/wp-content/uploads/book/Textbook-thamma-lanna-alphabet.pdf>
 [สืบค้นเมื่อ วันที่ 23 กันยายน 2566].
- [2] Cheriet M, Kharma N, Liu CL, Suen CY. (2007). Character recognition systems a guide for students and practioners.
- [3] (1979). Otsu N. A Threshold Selection Method from Gray-Level Histograms. :62-66.
- [4] Niblack W. (1985). An Introduction to Digital Image Processing.
- [5] Khurshid K, Siddiqi I, Faure C and Vincent N. (2009). Comparison of Niblack inspired binarization methods for ancient documents.
- [6] Histogram. [ออนไลน์]. ได้จาก:
<https://en.wikipedia.org/wiki/Histogram>
 [สืบค้นเมื่อ วันที่ 23 กันยายน 2566].
- [7] Christopher M. Bishop. (2006). Pattern Recognition and Machine Learning. :196-197.
- [8] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, :85-117.
- [9] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. (2012). ImageNet Classification with Deep Convolutional Neural Networks.
- [10] Convolutional Neural Network (CNN) คืออะไร. [ออนไลน์]. ได้จาก:
<https://medium.com/@natthawatphongchit/>
 [สืบค้นเมื่อ วันที่ 24 กันยายน 2566].
- [11] Convolutional Neural Networks, Explained. [ออนไลน์]. ได้จาก:
<https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>
 [สืบค้นเมื่อ วันที่ 24 กันยายน 2566].

บรรณานุกรม (ต่อ)

- [12] เข้าใจการทำงานพื้นฐานของ Neurons ใน Neural Networks. [ออนไลน์]. ได้จาก:
<https://datarockie.com/blog/how-neurons-work/>
[สืบค้นเมื่อ วันที่ 12 ตุลาคม 2566].
- [13] Modified National Institute of Standards and Technology. [ออนไลน์]. ได้จาก:
<https://paperswithcode.com/dataset/mnist>
[สืบค้นเมื่อ วันที่ 12 ตุลาคม 2566].
- [14] โอฟาริก สุรินตะ และกวีพจน์ บันลือวงศ์. (2559). การรู้จำตัวอักษรลายมือเขียนโดยใช้
ตัวจำแนกข้อมูลด้วยวิธีการคำนวณเพื่อนบ้านใกล้ที่สุด k ตำแหน่ง (วิทยานิพนธ์
ปริญญามหาบัณฑิต). มหาสารคาม. มหาวิทยาลัยมหาสารคาม. สืบค้นจาก
<https://www.thaiscience.info/journals/Article/JSMU/10985200.pdf>
- [15] กันต์กมล วงศ์กา. (2558). เทคนิคฮิสโตแกรมแมทซึ่งสำหรับการรู้จำและวิเคราะห์ตัวอักษร
ล้านนา (วิทยานิพนธ์ปริญญามหาบัณฑิต). เชียงราย. มหาวิทยาลัยราชภัฏเชียงราย.

ภาคผนวก

ภาคผนวก ก
โค้ดการสร้างแบบจำลอง



ภาพที่ ก.1 โค้ดการสร้างแบบจำลอง

จากภาพที่ ก.1 เป็นโค้ดสำหรับการสร้างแบบจำลอง โดยประกอบด้วย โค้ดการสร้างแบบจำลองฮิสโตแกรมแมทซิง แบบจำลองโครงข่ายประสาทเทียมแบบคอนโวลูชัน แบบจำลองการถดถอยโลจิสติก และการรวมแบบจำลองโหวต บันทึกโค้ดภาษา Python อยู่ในรูปแบบ SD การ์ดเป็นจำนวน 1 โฟลเดอร์ ชื่อโฟลเดอร์ Lanna Handwriting Recognition

ประวัติผู้วิจัย

ชื่อ-นามสกุล	นายประกิต หมั่นนามหน่อ	
วัน เดือน ปี เกิด	19 เมษายน 2543	
ที่อยู่ปัจจุบัน	บ้านเลขที่ 1050 หมู่ 10 ตำบล เวียงพางคำ อำเภอ แม่สาย จังหวัด เชียงราย รหัสไปรษณีย์ 57130	
เบอร์โทร	083-572-0926	
ประวัติการศึกษา	พ.ศ. 2561	มัธยมศึกษาปีที่ 6
		โรงเรียน แม่สายประสิทธิ์ศาสตร์
	พ.ศ. 2555	ประถมศึกษาปีที่ 6
		โรงเรียนเทศบาล 1 (วัดพรหมวิหาร)