# **บทที่ 4 ผลการดำเนินงาน**

สำหรับผลการดำเนินงานการพัฒนาระบบรู้จำท่าทางภาษามือไทยด้วยโครงข่ายประสาทเทียมแบบวนกลับ แบ่งออกได้เป็น 3 ส่วน ได้แก่

4.1 ผลการเตรียมข้อมูล

4.2 ผลการฝึกฝนโมเดล

4.3 ผลการวัดประสิทธิภาพโมเดล

4.4 ผลการเปรียบเทียบประสิทธิภาพโมเดล

4.5 ผลการทดสอบโมเดล

## **4.1 ผลการเตรียมข้อมูล**

**4.1.1 การหาข้อมูลภาษามือไทย**

ในการรวบรวมข้อมูล สำหรับการสร้าง TSL10 (dataset ภาษามือไทย 10 ท่า) ผู้วิจัยต้องการวิดีโอท่าภาษามือที่ใช้ในชีวิตประจำวันของผู้พิการทางการได้ยินและการสื่อความหมาย เป็นจำนวน 10 คำ ผู้วิจัยได้มีการออกหนังสือขอความอนุเคราะห์จากศูนย์บริการสนับสนุนการนักศึกษาพิการระดับอุดมศึกษา (DSS) ประจำมหาวิทยาลัยราชภัฏสกลนครเพื่อเก็บข้อมูลสำหรับการเทรนโมเดลสำหรับการรู้จำภาษามือไทยจากทั้งผู้เชี่ยวภาษามือและผู้พิการที่ใช้ภาษามือเป็นหลักในการสื่อสาร



**ภาพที่ 4.1** ตัวอย่างภาษามือไทย ‘สวัสดี’ จากผู้เชี่ยวชาญภาษามือไทย

**4.1.2 การสกัดลักษณะเด่นของข้อมูล**

ภาษามือนั้นขึ้นอยู่กับการใช้มือและท่าทาง การนำวิดีโอที่เป็นภาษามือมาใช้ในการเทรนโมเดลนั้นจึงเป็นเรื่องยาก ผู้วิจัยจึงได้ใช้เครื่องมือ MediaPipe ที่เป็น Framework มาใช้ในการแก้ปัญหา ซึ่งวิธีการคือการใช้ MediaPipe ในการ Keypoints ขึ้นตามจุดต่าง ๆ ของร่างกายเป็นค่า มิติ X, Y, Z ของหน้า, มือและท่าทางรูปภาพที่ 4.2

รูปภาพประกอบด้วย ผนัง, ในร่ม, คน, เสื้อผ้า

คำอธิบายที่สร้างโดยอัตโนมัติ

**ภาพที่ 4.2** การใช้ MediaPipe ในการจำลอง Keypoints

**4.1.3 การเตรียมไฟล์สำหรับเทรนโมเดล**

เมื่อสามารถสร้าง Keypoints เสร็จขั้นตอนต่อไปคือการนำผลของค่า Keypoints ของแต่ละจุดของร่างกายเขียนเป็น .npy ไฟล์ ซึ่งมีขั้นตอนดังนี้

1. สร้างโฟลเดอร์สำหรับเก็บ Datasets

2. ในโฟลเดอร์ Datasets มี โฟลเดอร์ที่เป็นชื่อท่าภาษามือ ดังภาพที่ 4.3

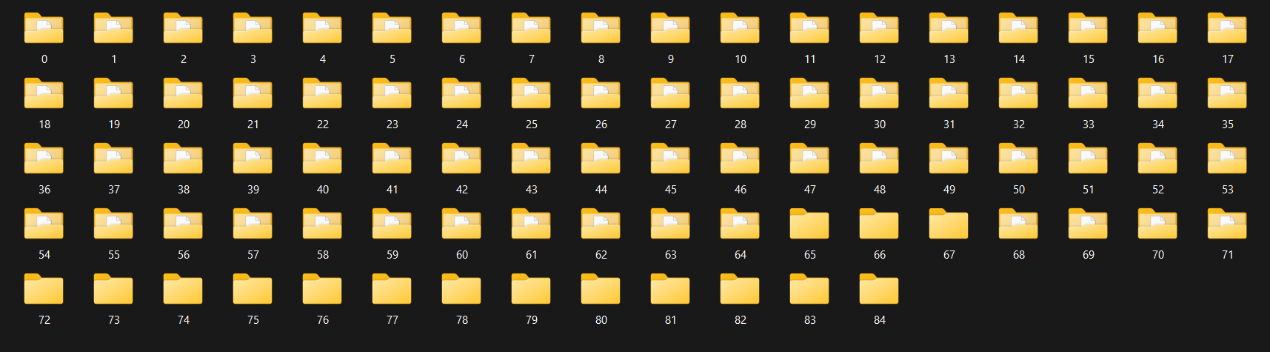
3. ในโฟลเดอร์ที่เป็นชื่อท่าภาษามือจะมีโฟลเดอร์สำหรับเก็บวิดีโอท่าภาษามือ 85 วิดีโอ โดยแยกเป็นโฟลเดอร์ละ 1 วิดีโอ ดังภาพที่ 4.4

4. ในโฟลเดอร์เก็บวิดีโอท่าภาษามือจะมีไฟล์ .npy 30 ไฟล์ ซึ่ง 1 ไฟล์ จะเก็บค่าที่ได้จากการสกัด Keypoints จาก Mediapipe X, Y, Z ใน 1 เฟรม ดังภาพที่ 4.5

รูปภาพประกอบด้วย ภาพหน้าจอ, ความมืด, สีดำ, เทียน

คำอธิบายที่สร้างโดยอัตโนมัติ

**ภาพที่ 4.3** โฟลเดอร์ชื่อท่าภาษามือ



**ภาพที่ 4.4** โฟลเดอร์ 60 โฟลเดอร์สำหรับเก็บ .npy ไฟล์

รูปภาพประกอบด้วย ภาพหน้าจอ, สมมาตร, หน้าต่าง, ศิลปะ

คำอธิบายที่สร้างโดยอัตโนมัติ

**ภาพที่ 4.5** ไฟล์ .npy 30 ไฟล์ ใน 1 โฟลเดอร์วิดีโอ

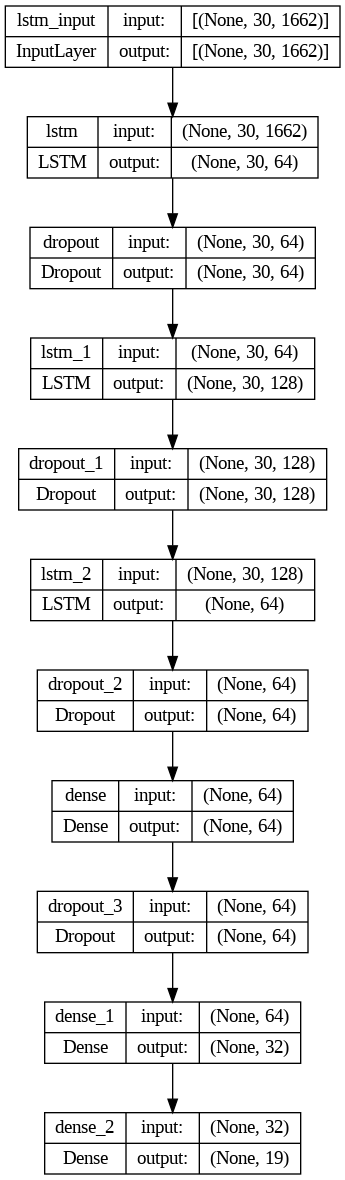
รูปภาพประกอบด้วย เสื้อผ้า, แบบแผน, ผ้า, ศิลปะ

คำอธิบายที่สร้างโดยอัตโนมัติ

**ภาพที่ 4.6** ไฟล์ .npy ที่เก็บค่า X, Y, Z ของ Keypoints

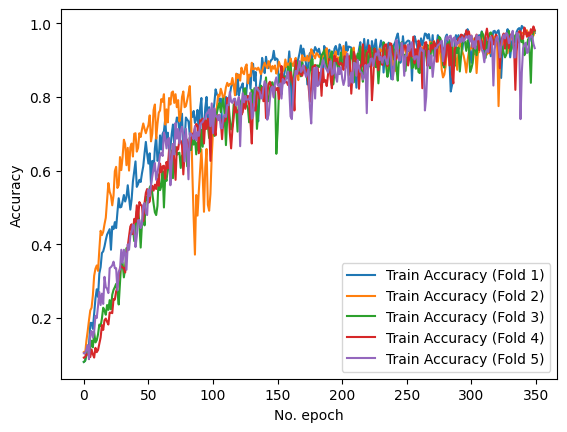
## **4.2 ผลการฝึกฝนโมเดล**

4.2.1 โมเดล LSTM

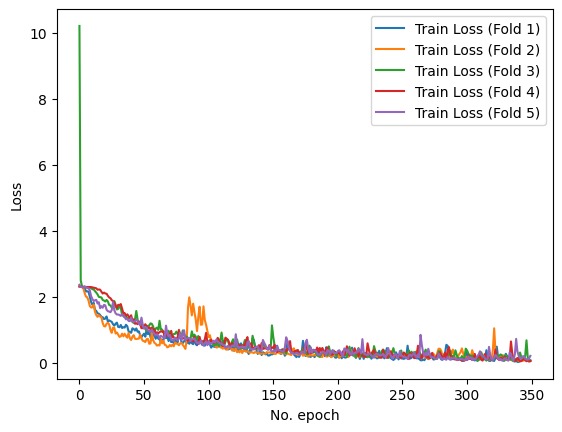


**ภาพที่ 4.7** โครงสร้างโมเดล LSTM

ผลจากการพัฒนาระบบรู้จำท่าทางภาษามือไทยด้วยโครงข่ายประสาทเทียมแบบวนกลับ พบว่า การกำหนดค่าอัตราการเรียนรู้ Epochs เท่ากับ 350 และ Batch Size เท่ากับ 32 ของโมเดล LSTM ให้ค่า Accuracy และค่า Loss ดังภาพด้านล่าง

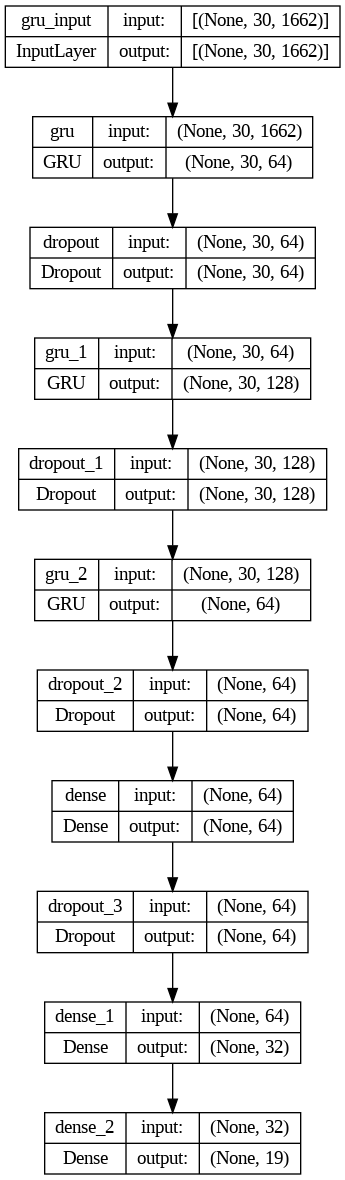
****

**ภาพที่ 4.8** กราฟแสดงจำนวนรอบการเทรนและค่าความถูกต้องของโมเดล LSTM



**ภาพที่ 4.9** กราฟแสดงรอบการเทรนและค่าความผิดพลาดของโมเดล LSTM

4.2.2 โมเดล GRU



**ภาพที่ 4.10** โครงสร้างโมเดล GRU

ผลจากการพัฒนาระบบรู้จำท่าทางภาษามือไทยด้วย RNN พบว่า การกำหนดค่าอัตราการเรียนรู้ Epochs เท่ากับ 370 และ Batch Size เท่ากับ 32 ของโมเดล GRU ให้ค่า Accuracy และค่า Loss ดังภาพด้านล่าง

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, พล็อต, แผนภาพ

คำอธิบายที่สร้างโดยอัตโนมัติ

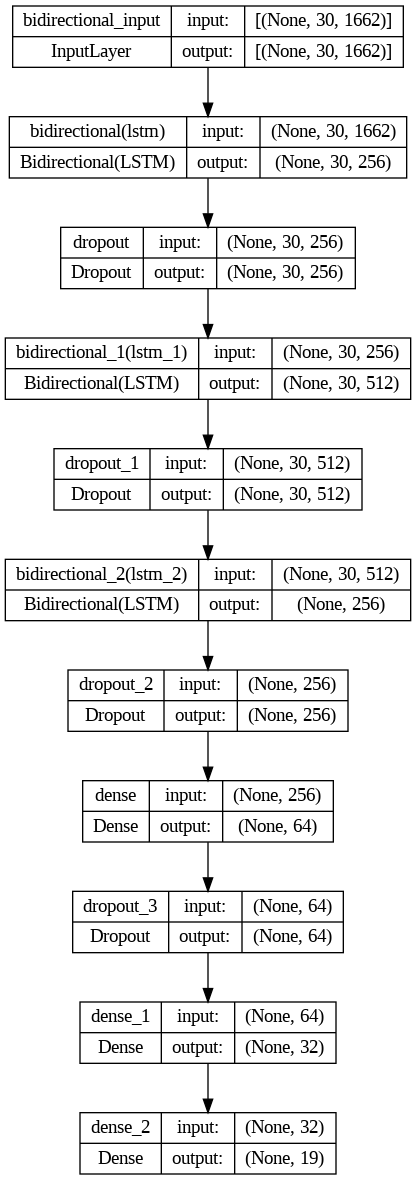
**ภาพที่ 4.11** กราฟแสดงจำนวนรอบการเทรนและค่าความถูกต้องของโมเดล GRU

รูปภาพประกอบด้วย ภาพหน้าจอ, ข้อความ, พล็อต, แผนภาพ

คำอธิบายที่สร้างโดยอัตโนมัติ

**ภาพที่ 4.12** กราฟแสดงรอบการเทรนและค่าความผิดพลาดของโมเดล GRU

4.2.3 โมเดล BiLSTM



**ภาพที่ 4.13** โครงสร้างโมเดล BiLSTM

ผลจากการพัฒนาระบบรู้จำท่าทางภาษามือไทยด้วย RNN พบว่า การกำหนดค่าอัตราการเรียนรู้ Epochs เท่ากับ 520 และ Batch Size เท่ากับ 32 ของโมเดล BiLSTM ให้ค่า Accuracy และค่า Loss ดังภาพด้านล่าง

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, พล็อต, แผนภาพ

คำอธิบายที่สร้างโดยอัตโนมัติ

**ภาพที่ 4.14** กราฟแสดงจำนวนรอบการเทรนและค่าความถูกต้องของโมเดล BiLSTM

รูปภาพประกอบด้วย ภาพหน้าจอ, ข้อความ, พล็อต, แผนภาพ

คำอธิบายที่สร้างโดยอัตโนมัติ

**ภาพที่ 4.15** กราฟแสดงรอบการเทรนและค่าความผิดพลาดของโมเดล BiLSTM

จากภาพของโครงสร้างโมเดลทั้ง 3 โมเดล ผู้วิจัยได้มีการใช้ Dense และ Dropout ซึ่ง Dense layer เป็นชั้นของโครงข่ายประสาทเทียมที่มีการเชื่อมต่อ (connection) ระหว่างโนด (node) ในชั้นก่อนหน้ากับชั้นต่อไป โดยทุกโนดในชั้นจะมีการเชื่อมต่อกับโนดทุกตัวในชั้นต่อไป ชั้น Dense จะมีการปรับค่าน้ำหนัก (weight) และค่าไบแอส (bias) ของโนดในแต่ละชั้น ซึ่งช่วยให้โมเดลสามารถเรียนรู้และสร้างรูปแบบ (pattern) ที่ซับซ้อนได้มากขึ้น

Dropout layer เป็นชั้นที่มีไว้เพื่อลดการเกิด overfitting ในโมเดล โดย Dropout layer จะสุ่มตัดการเชื่อมต่อระหว่างโนดในชั้นก่อนหน้ากับชั้นต่อไป โดยตัดการเชื่อมต่อเหล่านี้ด้วยการกำหนดค่าเป็นศูนย์ (zero) โดยสุ่มตัดบางโนดออกจากการคำนวณในแต่ละรอบการฝึกฝน การทำ Dropout จะช่วยให้โมเดลสามารถเรียนรู้และสร้างรูปแบบที่เหมาะสมกับข้อมูลได้ดีขึ้นโดยไม่เกิดการเรียนรู้ที่ผิดพลาดจาก overfitting

การใช้ Dense layer และ Dropout layer ในโมเดลประสาทเทียมช่วยให้โมเดลมีความแม่นยำในการทำนายข้อมูลมากขึ้น โดย Dense layer เป็นชั้นหลักในการสร้างโมเดลและ Dropout layer เป็นชั้นช่วยลดการเกิด overfitting ในโมเดลให้มีประสิทธิภาพสูงขึ้น การใช้ Dense layer และ Dropout layer นั้นเป็นเพิ่มความแม่นยำมากขึ้นนั่นเอง

## **4.3 ผลการวัดประสิทธิภาพโมเดล**

4.3.1 ผลการวัดประสิทธิภาพโมเดล LSTM

จากการเทรนโมเดลด้วยชุดข้อมูลสำหรับเทรนและข้อมูลสำหรับทดสอบได้ค่า Accuracy และค่า Loss ดังนี้

**ตารางที่ 4.1** ผลการวัดประสิทธิภาพโมเดล LSTM

|  |  |  |
| --- | --- | --- |
| **K-Fold** | **Accuracy** | **Loss** |
| **Fold 1** | 0.94 | 0.08 |
| **Fold 2** | 0.91 | 0.05 |
| **Fold 3** | 0.93 | 0.06 |
| **Fold 4** | 0.95 | 0.05 |
| **Fold 5** | 0.94 | 0.21 |
| **Mean** | **0.93** | **0.09** |

4.2.2 ผลการวัดประสิทธิภาพโมเดล GRU

จากการเทรนโมเดลด้วยชุดข้อมูลสำหรับเทรนและข้อมูลสำหรับทดสอบได้ค่า Accuracy และค่า Loss ดังนี้

**ตารางที่ 4.2** ผลการวัดประสิทธิภาพโมเดล GRU

|  |  |  |
| --- | --- | --- |
| **K-Fold** | **Accuracy** | **Loss** |
| **Fold 1** | 0.92 | 0.08 |
| **Fold 2** | 0.85 | 0.13 |
| **Fold 3** | 0.96 | 0.09 |
| **Fold 4** | 0.94 | 0.05 |
| **Fold 5** | 0.93 | 0.13 |
| **Mean** | **0.92** | **0.10** |

และผลจากการประเมินด้วย Confusion Matrix ของแต่ละท่าภาษามือไทยได้ดังนี้

4.3.3 ผลการวัดประสิทธิภาพโมเดล BiLSTM

จากการเทรนโมเดลด้วยชุดข้อมูลสำหรับเทรนและข้อมูลสำหรับทดสอบได้ค่า Accuracy และค่า Loss ดังนี้

**ตารางที่ 4.3** ผลการวัดประสิทธิภาพโมเดล BiLSTM

|  |  |  |
| --- | --- | --- |
| **K-Fold** | **Accuracy** | **Loss** |
| **Fold 1** | 0.89 | 0.13 |
| **Fold 2** | 0.97 | 0.04 |
| **Fold 3** | 0.98 | 0.05 |
| **Fold 4** | 0.91 | 0.23 |
| **Fold 5** | 0.97 | 0.06 |
| **Mean** | **0.94** | **0.11** |

## **4.4 ผลการเปรียบเทียบประสิทธิภาพโมเดล**

ผลจากการพัฒนาและวัดประสิทธิภาพของโมเดลทั้ง 3 แบบได้แก่ LSTM, GRU และ BiLSTM ที่ใช้ในการพัฒนาระบบรู้จำท่าทางภาษามือไทยด้วย RNN ซึ่งแสดงได้ดังต่อไปนี้

**ตารางที่ 4.4** เปรียบประสิทธิภาพของโมเดลทั้ง 3 โมเดล

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Epoch | Mean Accuracy | Mean Loss |
| LSTM | 350 | 0.93 | 0.09 |
| GRU | 370 | 0.92 | 0.10 |
| BiLSTM | 520 | 0.94 | 0.11 |

จากตารางเปรียบเทียบประสิทธิภาพโมเดลทั้ง 3 โมเดล BiLSTM มีค่าความถูกต้องสูงอยู่ที่ 0.94 ซึ่งมากที่สุดในโมเดลทั้ง 3 และมีค่า Loss อยู่ที่ 0.09 มีจำนวนรอบการในการเทรนอยู่ที่ 520 รอบและโมเดล GRU ที่มีจำนวนการเทรน 370 รอบ แต่ให้ค่า Accuracy ใกล้เคียงกับโมเดล LSTM นอกจากความแตกต่างของจำนวนในการเทรนของโมเดลแล้วยังมีเรื่องของเวลาที่ใช้ในการเทรนที่แตกต่างระหว่างโมเดล LSTM และโมเดล GRU แม้ว่าจำนวนในการเทรนของโมเดล LSTM จะน้อยกว่าโมเดล GRU แต่ใช้เวลาในการเทรนมากกว่าเนื่องจากจำนวนพารามิเตอร์ของโมเดล GRU นั้นน้อยกว่าจึงทำให้ใช้เวลาในการเทรนน้อยกว่าแม้จำนวนรอบที่ใช้ในการเทรนจะมากกว่าส่วนโมเดล BiLSTM มีจำนวนรอบในการเทรนมากที่สุดและใช้เวลาในหนึ่งรอบการเทรนมากที่สุดเนื่องจากตัวโมเดลมีการใช้จำนวนโหนดมากกว่า 2 โมเดล จึงทำให้ใช้เวลานานในการเทรน

## **4.5 ผลการทดสอบโมเดล**

จากผลการทดสอบโมเดลทั้ง 3 โมเดลที่ผ่านการเทรนแล้วนำมาทดสอบด้วยวิดีโอที่จัดเตรียมไว้เพื่อทดสอบโมเดล โดยใช้วิธี Confusion Matrix ในการประเมิน