# **บทที่ 3 วิธีดำเนินการวิจัย**

สำหรับวิธีการดำเนินการวิจัยการพัฒนาระบบการรู้จำภาษามือไทยและท่าทางด้วยเทคนิค LSTM นั้นสามารถแบ่งออกเป็น 5 ส่วนดังนี้

3.1 การเตรียมข้อมูล

3.2 การฝึกฝนโมเดล

3.3 การวัดประสิทธิภาพโมเดล

3.4 การเปรียบเทียบประสิทธิภาพโมเดล

3.5 การนำไปใช้งาน

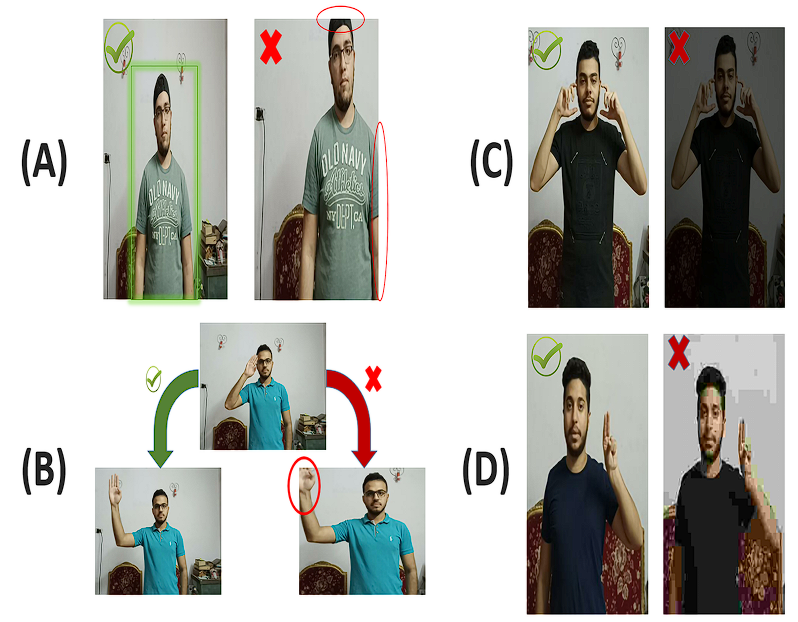
## **3.1 การเตรียมข้อมูล**

**3.1.1 การรวบรวมข้อมูล**

ในการรวบรวมข้อมูล สำหรับการสร้าง Dataset ผู้วิจัยต้องการวิดีโอท่าภาษามือที่ใช้ในชีวิตประจำวันของผู้พิการทางการได้ยินและการสื่อความหมาย เป็นจำนวน 10 คำ โดยทำเป็นวิดีโอ 30 วิดีโอต่อ 1 คำ และใน 1 วิดีโออัตราเฟรมต่อวินาทีที่ 30 FPS ขนาดของวิดีโอคือ 640 x 480 โดยแบ่งคำตามตำแหน่งของมือ ตำแหน่งละ 2 คำเพื่อใช้ต้นแบบในการสร้าง Data สำหรับเทรนโมเดล ดังตารางที่ 3.1.1.1

**ตารางที่ 3.1.1.1** ตำแหน่งของมือและคำศัพท์

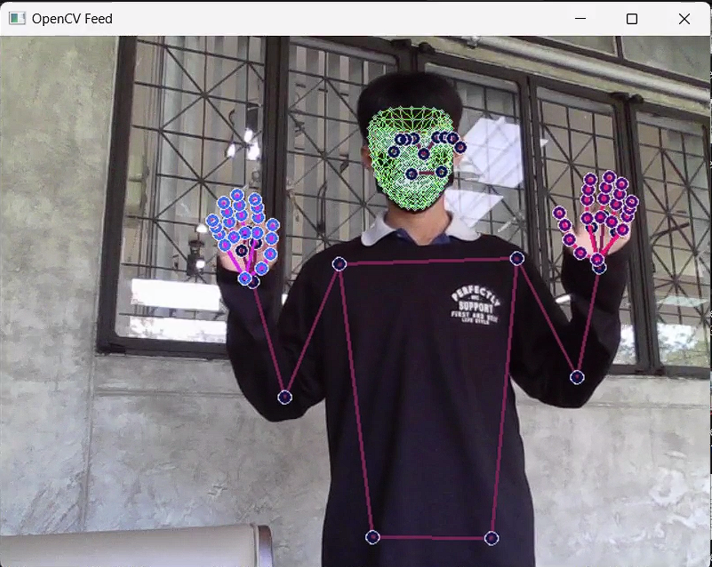
|  |  |
| --- | --- |
| **ตำแหน่งของมือ** | **คำศัพท์** |
| ศรีษะ | คิดบวก, ฉลาด |
| ใบหน้า | อาจารย์, น้ำเปล่า |
| คาง/คอ | เก่ง, รัก |
| อก | ล็อค, ปลดล็อค |
| ท้อง/เอว | นั่ง, ปวดท้อง |



**ภาพที่ 3.1.1.1** วิธีบันทึกข้อมูลวิดีโอ

**3.1.2 การสกัดลักษณะเด่นของข้อมูล**

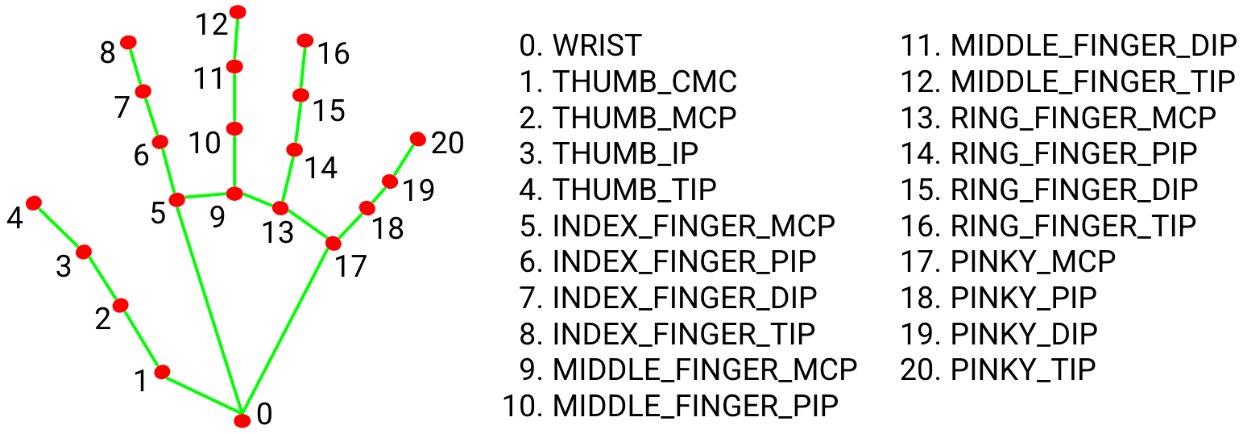
ภาษามือนั้นขึ้นอยู่กับการใช้มือและท่าทาง การนำวิดีโอที่เป็นภาษามือมาใช้ในการเทรนโมเดลนั้นจึงเป็นเรื่องยาก ผู้วิจัยจึงได้ใช้เครื่องมือ MediaPipe ที่เป็น Framework มาใช้ในการแก้ปัญหา ซึ่งวิธีการคือการใช้ MediaPipe ในการ Keypoints ขึ้นตามจุดต่าง ๆ ของร่างกายเป็นค่า มิติ X, Y, Z ของหน้า, มือและท่าทางรูปภาพที่ 3.1.2.1



**ภาพที่ 3.1.2.1** การใช้ MediaPipe ในการ Keypoints

ในมือแต่ละข้างนั้น MediaPipe จะสกัดออกมาได้ 21 Keypoints ซึ่ง Keypoint จะถูกคำนวณแบบ 3 มิติ X, Y, Z ของมือทั้งสองข้าง โดยจะได้ Keypoints จากการสกัดจากมือดังนี้

Keypoins in hand x Three dimensions x No. of hands = (21 x 3 x2) = 126 Keypoints ดังภาพที่ 3.1.2.2

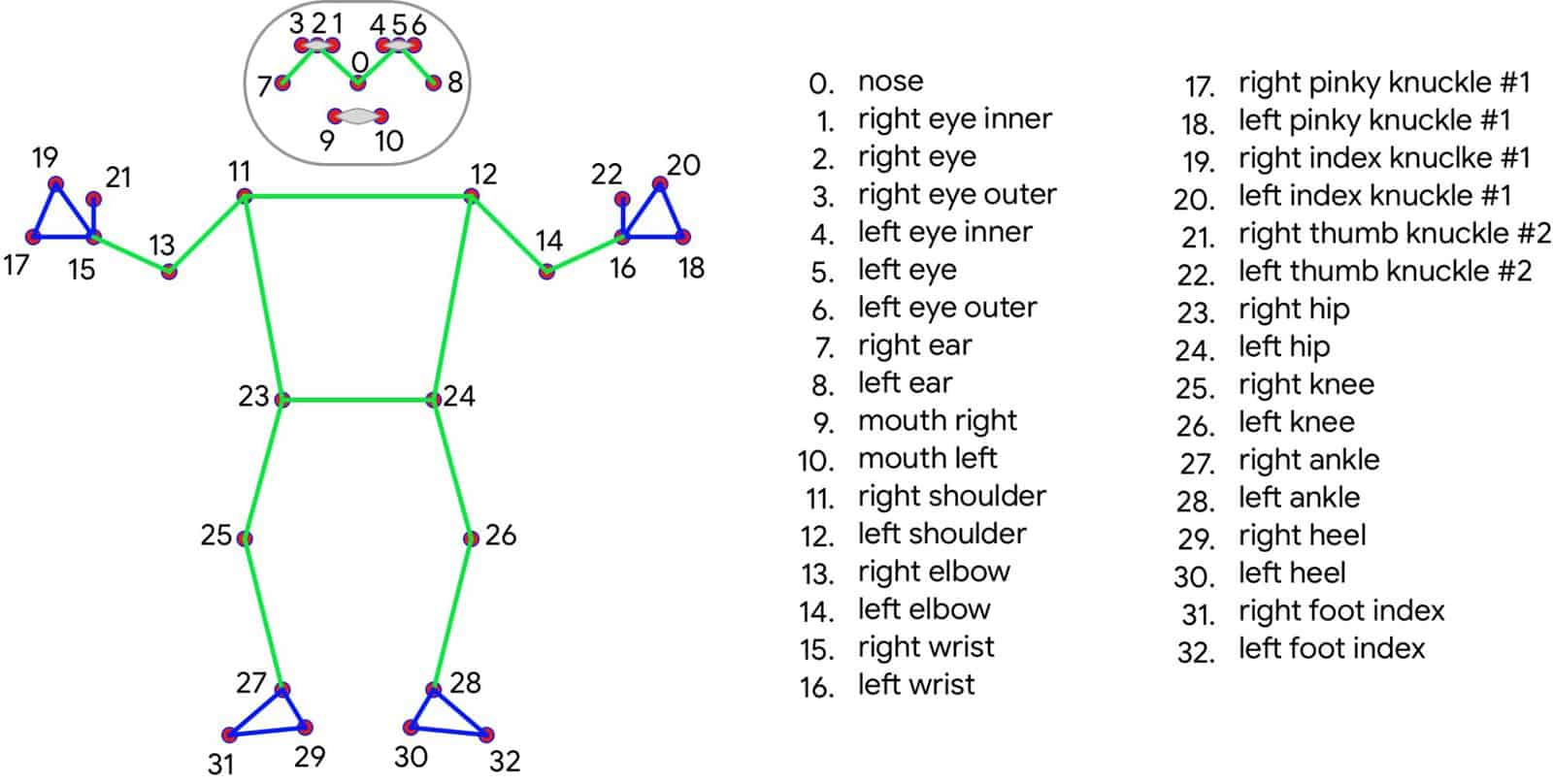


**ภาพที่ 3.1.2.2** ลำดับและป้ายกำกับ Keypoints ของมือใน MediaPipe

**ที่มา :** MediaPipe (2023: Online)

ในส่วนของท่าทางนั้น MediaPipe จะสกัดออกมาได้ 33 Keypoints คำนวณแบบ 3 มิติ X, Y, Z และเพิ่มค่า Visibility เข้าไปซึ่งเป็นค่าที่จะระบุว่าจุดนั้นมองเห็นหรือซ่อนอยู่ (ที่ถูกปิดโดยจุดอื่นของร่างกาย) บนเฟรมดังนั้นจะได้ค่า Keypoints ดังนี้

Keypoints in pose x (Three dimenstions + Visibility) = (33 + (33 + 1)) = 132 Keypoints ดังภาพที่ 3.1.2.3

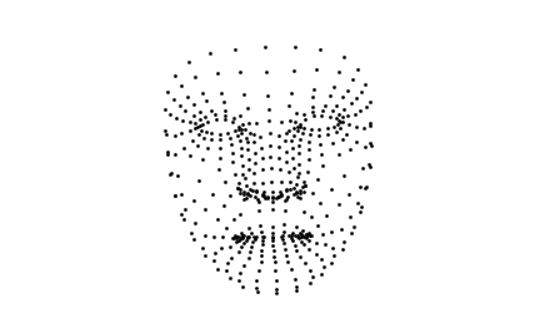


**ภาพที่ 3.1.2.3** ลำดับและป้ายกำกับ Keypoints ของท่าทางใน MediaPipe

**ที่มา :** MediaPipe (2023: Online)

สำหรับหน้านั้น Mediapipe สกัดออกมาได้ 468 Keypoints ได้แก่ รูปทรงรอบหน้าและหน้า, ตา, ปากและคิ้ว ซึ่งคำนวณค่า 3 มิติ X, Y, Z ได้ดังนี้

Keypoints in face x Three dimensions = (468 x 3) = 1404 Keypoints ดังภาพที่ 3.1.2.3



**ภาพที่ 3.1.2.3** Keypoints บนหน้า

ดังนั้นเมื่อรวม Keypoint ทั้งหมดเข้าด้วยกันไม่ว่าจะเป็นจาก หน้า ท่าทางและมือจะสามารถคำนวณได้ดังนี้

Keypoints in hands + in pose + inface = (126 + 132 + 1404) = 1662 Keypoints

**3.1.3 การเตรียมไฟล์**

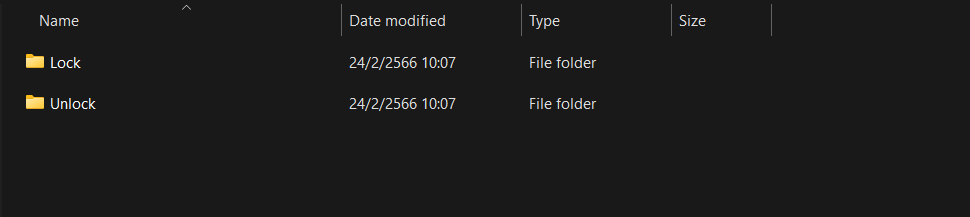
เมื่อสามารถสร้าง Keypoints เสร็จขั้นตอนต่อไปคือการนำผลของค่า Keypoints ของแต่ละจุดของร่างกายเขียนเป็น .npy ไฟล์ ซึ่งมีขั้นตอนดังนี้

1. สร้างโฟลเดอร์สำหรับเก็บ Datasets

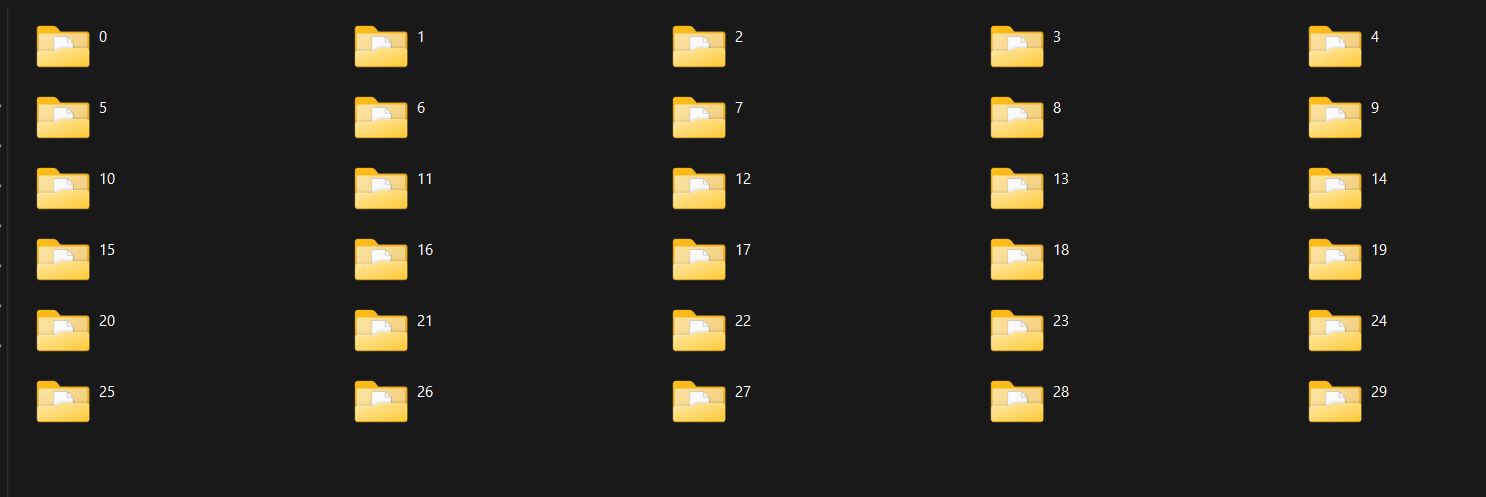
2. ในโฟลเดอร์ Datasets มี โฟลเดอร์ที่เป็นชื่อท่าภาษามือ ดังภาพที่ 3.1.3.1

3. ในโฟลเดอร์ที่เป็นชื่อท่าภาษามือจะมีโฟลเดอร์สำหรับเก็บวิดีโอท่าภาษามือ 30 วิดีโอ โดยแยกเป็นโฟลเดอร์ละ 1 วิดีโอ ดังภาพที่ 3.1.3.2

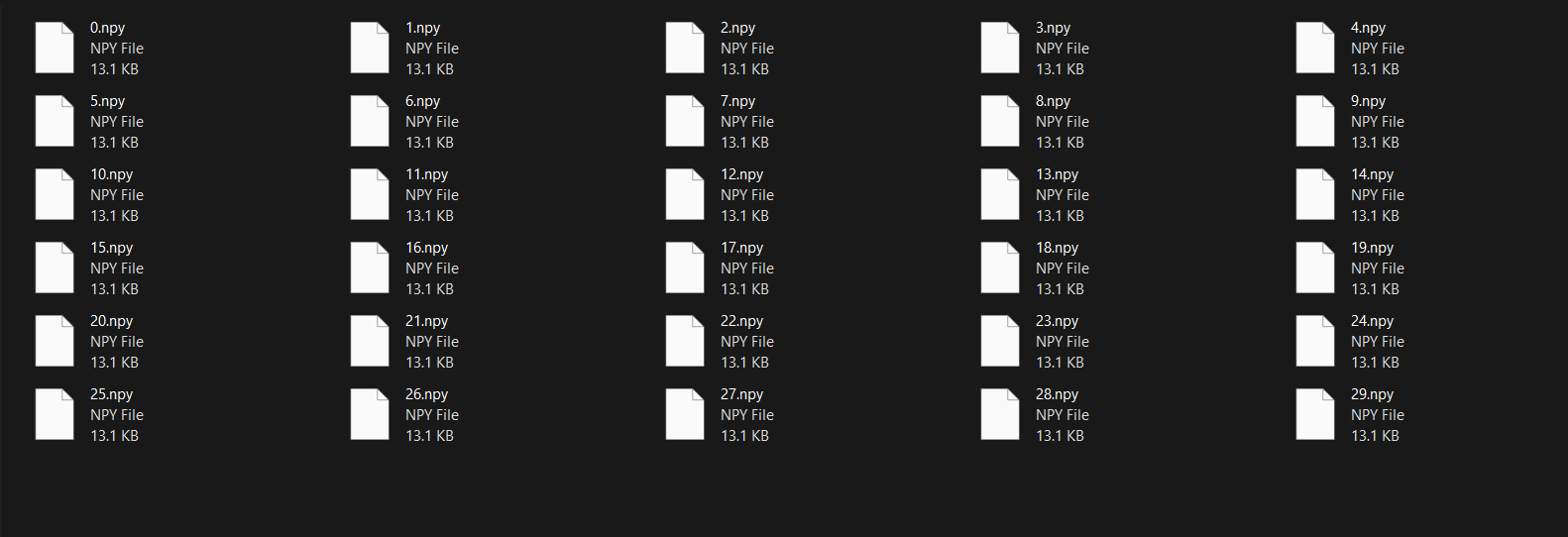
4. ในโฟลเดอร์เก็บวิดีโอท่าภาษามือจะมีไฟล .npy 30 ไฟล์ ซึ่ง 1 ไฟล์ จะเก็บค่าที่ได้จากการสกัด Keypoints จาก Mediapipe X, Y, Z ใน 1 เฟรม ดังภาพที่ 3.1.3.3



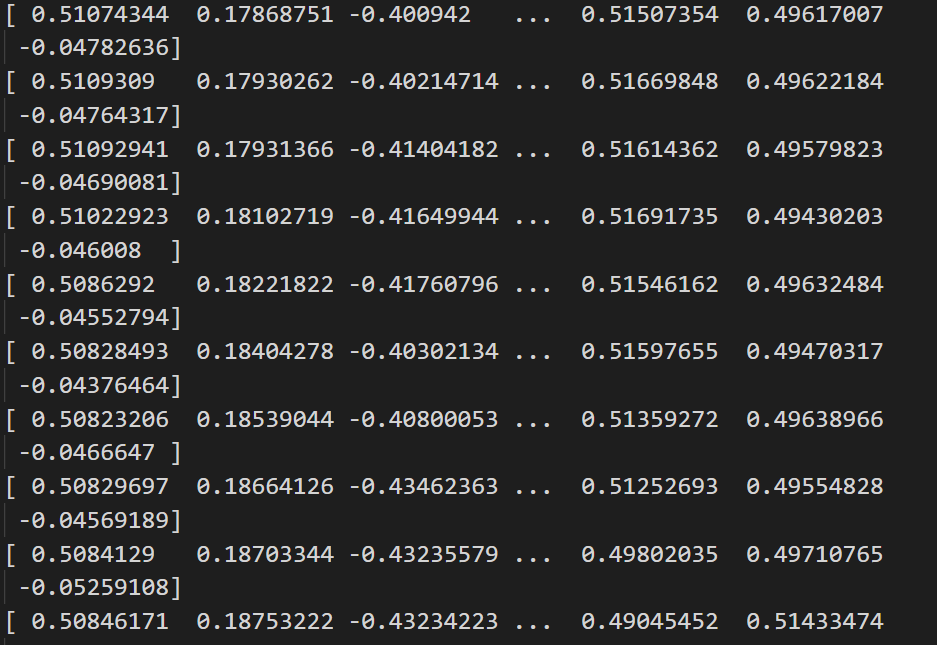
**ภาพที่ 3.1.3.1** โฟลเดอร์ชื่อท่าภาษามือ



**ภาพที่ 3.1.3.2** โฟลเดอร์ 30 โฟลเดอร์สำหรับเก็บ .npy ไฟล์



**ภาพที่ 3.1.3.3** ไฟล์ .npy 30 ไฟล์ ใน 1 โฟลเดอร์วิดีโอ



**ภาพที่ 3.1.3.4** ไฟล์ .npy ที่เก็บค่า X, Y, Z ของ Keypoints

## **3.2 การฝึกฝนโมเดล**

ผู้วิจัยได้ใช้โมเดลในการเทรนทั้งหมด 3 โมเดลได้แก่ LSTM, GRU, BiSLTM ในงานวิจัยครั้งนี้ซึ่งเป็นโมเดลของ Recurrent Neurons Networks (RNN)  
Number of Nodes คือ จำนวนของ Input Node ซึ่งผู้วิจัยกำหนดขั้นต่ำไว้ 64 จนถึง 256

Activation คือตัวฟังก์ชันที่ใช้ในการรับผลรวมจากการประมวลผลทั้งหมดจากทุก Input Node เข้ามาพิจารณาตามกลไกลการคำนวณของ Activation Function นั้น ๆ แล้วส่งต่อไปเป็น Output ซึ่งในงานวิจัยนี้ได้เลือกใช้ 2 ตัว คือ Rectified Linear Unit (ReLU) และ Softmax

Optimizer คือ อัลกอริทึมการเพิ่มประสิทธิภาพ (Optimizer) ทำหน้าที่เป็นกลไกการปรับปรุงค่าน้ำหนักของตัวแปรต้นต่าง ๆ รวมถึงค่าความคลาดเคลื่อน (Bias) ในงานวิจัยนี้ได้เลือกใช้ Optimizer ได้แก่ Adagrad. Adamax, Adam or RMSprop ดังตารางที่ 3.2.1

**ตารางที่ 3.2.1** พารามิเตอร์ของเลเยอร์โมเดล

|  |  |
| --- | --- |
| **Parameters** | **Value** |
| RNN Model | GRU, LSTM, BiLSTM |
| Number of Nodes | Between (64, 256) |
| Activation | ‘Relu’ or ‘Softmax’ |
| Optimizer | ‘Adagrad’, ‘Adamax’, ‘Adam’ or ‘RMSprop’ |

## **3.3 การวัดประสิทธิภาพโมเดล**

การวัดประสิทธิภาพของโมเดล ผู้วิจัยได้ใช้ตัวชี้วัดคือค่า Accuracy หรือก็คือค่าอัตราความถูกต้องของการทำนายของโมเดลโดยในการวิจัยครั้งนี้ ผู้วิจัยตั้งเป้าหมายของค่าความถูกต้องไว้ที่ > 90% และค่า Loss หรือก็คือค่าที่ใช้วัดว่าโมเดลทำนายได้ดีแค่ไหน ยิ่งค่า Loss น้อยเท่าไหร่ โมเดลจะยิ่งมีความแม่นยำในการทำนาย ซึ่งผู้วิจัยได้ตั้งเป้าหมายค่า Loss ครั้งนี้ไว้ที่ <= 0.2 จะทำการทดสอบค่าความถูกต้องในการทำนายด้วยวิธี Cross Validation โดยทำการแบ่งข้อมูลออกเป็น 2 ส่วน ได้แก่ ส่วนที่เอาไว้ใช้สำหรับการเทรนและอีกส่วนคือส่วนสำหรับการทดสอบ จะทำการสุ่มข้อมูลตามอัตราส่วนร้อยละ 60:40 และ 70:30

## **3.4 การเปรียบเทียบประสิทธิภาพโมเดล**

ในขั้นตอนการเปรียบประเทียบประสิทธิภาพ ผู้วิจัยจะนำโมเดลที่ผ่านการเทรนทั้งหมด 3 โมเดล ได้แก่ LSTM, GRU, BiLSTM ซึ่งจะเปรียบประสิทธิภาพเรื่องของ ค่า Accuracy, ค่า Loss และ จำนวนรอบที่ใช้ในการเทรนโมเดล (epochs) เพื่อหาว่าโมเดลใด มีความแม่นยำมากที่สุด

## **3.5 การนำไปใช้งาน**

เป็นการนำโมเดลที่ผ่านการผ่านการเทรนทั้ง 3 โมเดลมาทดสอบใช้ผ่านกล้อง Web Cam จริง