รายงานเรื่อง

การรับรู้คำศัพท์ภาษามือมาตรฐานสหรัฐอเมริกา

Word Level American Sign Language Recognition

จัดทำโดย

นาย ถิรวัฒน์	พงศ์ปฏิสนธิ	6410450958
นาย ศรัณย์	วงษ์คำ	6410451415
นาย ธนภัทร	เชื้อโตหลวง	6410401060

เสนอ

ผศ.ดร.ชาคริต วัชโรภาส

รายงานนี้เป็นส่วนหนึ่งของวิชา

01418364 Practical Deep Learning

ภาคปลาย ปีการศึกษา 2566

สารบัญ

ที่มาและความสำคัญ

American Sign Language (ASL) หรือภาษามือ เป็นภาษาที่ใช้มือ ร่างกาย และริมฝีปาก ในการสื่อสารเป็นท่าทางแทนคำพูด ภาษามือถูกใช้อย่างแพร่หลายในกลุ่มผู้ที่ไม่สามารถใช้เสียงในการสื่อสารแต่กลับมีกลุ่มผู้ที่ใช้เสียงและคำพูดในการสื่อสารจำนวนมากที่ไม่สามารถเข้าใจความหมายของภาษามือเนื่องจากภาษามือมีท่าทางในการสื่อสารที่หลากหลายละเอียดและซับซ้อน ทำให้การสื่อสารระหว่างผู้ที่ใช้ภาษามือและไม่ใช้ภาษามือเป็นเรื่องยาก

โครงงานนี้จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อให้ผู้ที่ไม่ใช้ภาษามือสามารถเข้าใจในสิ่งที่ผู้ใช้ภาษามือต้องการสื่อสารได้ โดยพัฒนาโมเดลที่สามารถช่วยในการระบุและจำแนกคำศัพท์ภาษามือได้อย่างถูกต้อง นอกจากนี้ยังทำให้เข้าถึงเทคโนโลยีที่มีอยู่ในปัจจุบันได้อย่างมีประสิทธิภาพและมีคุณภาพชีวิตที่ดีขึ้น และถือเป็นการส่งเสริมการใช้เทคโนโลยีในทิศทางที่สร้างสรรค์และเหมาะสม

รายละเอียดของปัญหา/ประโยชน์ที่จะได้รับ

ในการตรวจจับคำศัพท์ในภาษามือ สามารถกำหนดปัญหาให้อยู่ในรูปแบบการจำแนกหมวดหมู่ (Classification) โดยสร้างโมเดลที่สามารถระบุและจำแนกคำศัพท์ภาษามือได้อย่างถูกต้องและมีประสิทธิภาพเพื่อให้ผู้ใช้งานภาษามือสามารถสื่อสารได้ง่ายขึ้น

ทั้งบุคคลที่ใช้ภาษามือและไม่ได้ใช้ภาษามือในการสื่อสารจะสามารถสื่อสารกันได้อย่างมีประสิทธิภาพและสมบูรณ์มากยิ่งขึ้นโดยผู้ที่ไม่ใช้ภาษามือสามารถใช้โมเดลที่ผู้จัดทำกำลังจัดทำขึ้นเพื่อจับท่าทางของผู้ใช้ภาษามือและแปลเป็นภาษาที่ผู้ที่ไม่ใช้ภาษามือสามารถเข้าใจได้การแก้ไขปัญหานี้จะส่งผลดีให้กับทั้งสองฝ่ายและช่วยเพิ่มคุณภาพชีวิตในการสื่อสารในชีวิตประจำวันได้มากขึ้นอย่างมีนัยยะสำคัญ

วิธีการเก็บรวบรวมข้อมูลและการแบ่งชุดข้อมูล

1. คัดเลือกวิดีโอการสื่อสารด้วยภาษามือจากชุดข้อมูลสาธารณะดังนี้

   1.1 WLASL Video จาก www.kaggle.com

   1.2 WLASL Dataset จาก www.paperswithcode.com

2. การถ่ายวิดิโอการสื่อสารด้วยภาษามือด้วยตนเอง

อัตราส่วนของข้อมูลที่ได้จากวิธีที่ 1 ต่อวิธีที่ 2 คิดเป็นร้อยละ 60 ต่อร้อยละ 40

อัตราส่วนของชุดข้อมูลฝึกต่อชุดข้อมูลทดสอบ คิดเป็นร้อยละ 80 ต่อร้อยละ 20

ลักษณะของชุดข้อมูล

ประกอบด้วยวิดีโอของบุคคลที่กำลังสื่อสารด้วยภาษามือโดยถ่ายเฉพาะส่วนใบหน้าและลำตัวด้วยคำศัพท์ 16 คำ จำนวน 200 วิดีโอ แต่ละวิดีโอเป็นการสื่อสารด้วยคำศัพท์ 1 คำ มีความยาวประมาณ 3-5 วินาที และถูกตั้งชื่อด้วยเลขห้าหลักที่แตกต่างกัน

มีไฟล์ WLASL_data.json เพื่อบ่งบอกความหมายและช่วงเวลาที่ใช้ของแต่ละวิดีโอ โดยแต่ละ string มี 4 property ดังนี้

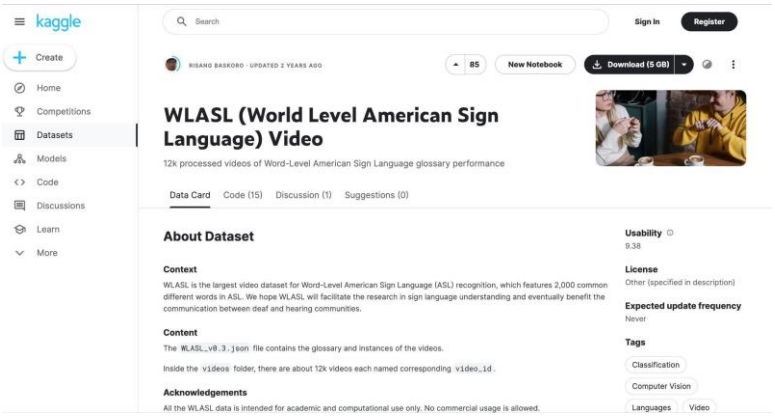
- gloss คำศัพท์ภาษามือ
- video_path ที่อยู่ของวิดีโอภาษามือซึ่งเป็นเส้นทางไฟล์ที่ชี้ไปยังวิดีโอภาษามือนั้น
- frame_start เฟรมแรกที่มีการเก็บข้อมูลคำศัพท์ในวิดีโอ
- frame_end เฟรมสุดท้ายที่มีการเก็บข้อมูลในวิดีโอ โดยค่า -1 หมายถึงไม่มีการระบุเฟรมสุดท้าย

วิธีการสกัดตัวแปรอิสระและตัวแปรตาม

สำหรับแต่ละวิดีโอในชุดข้อมูล จะมีการตรวจตำแหน่งของมือ ใบหน้า และลำตัวด้วยจุดจำนวน 180 โดยแบ่งตรวจจับแต่ละส่วนดังนี้

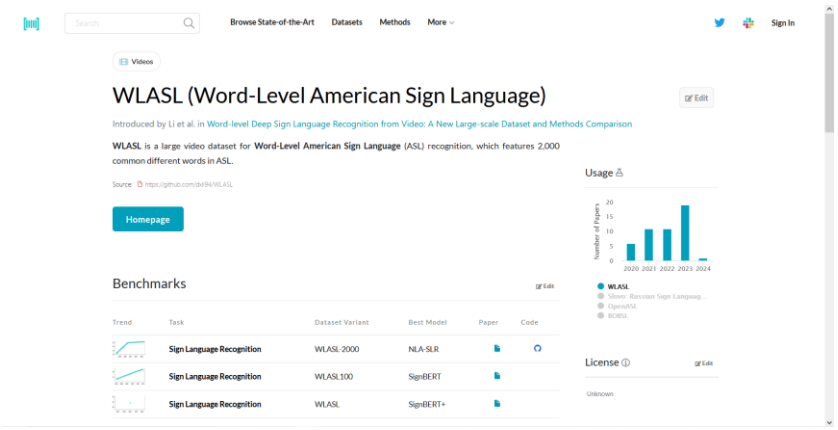
- บริเวณมือ 42 จุด (ข้างละ 21 จุด)
- ใบหน้า 132 จุด
- ลำตัว 6 จุด

สำหรับแต่ละเฟรมภายในวิดีโอ พิกัดของแต่ละจุดจะถูกเก็บเป็นตัวแปรอิสระของ 1 ตัวอย่าง และกำหนดตัวแปรตามเป็นอาร์เรย์ที่มีความยาวเท่าจำนวนคำศัพท์ ทุกจำนวนภายในอาร์เรย์มีค่าเท่ากับ 0 ยกเว้นตำแหน่งที่มีหมายเลขสอดคล้องกับผลเฉลยของวิดีโอที่มีเฟรมนั้นประกอบอยู่



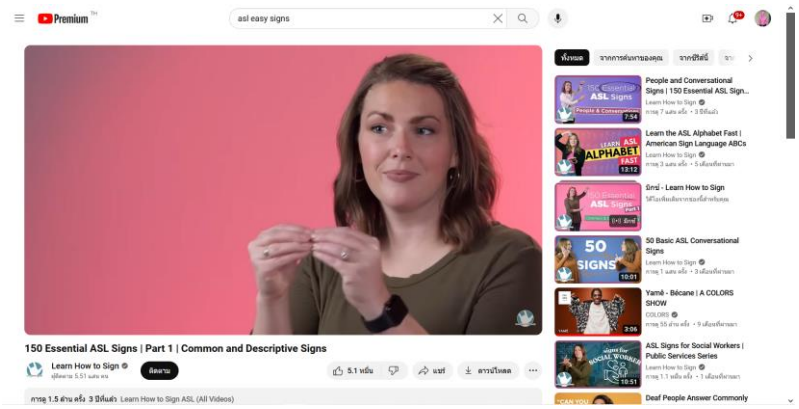
รูปที่ 1 แสดงแหล่งข้อมูลที่ทำการค้นหาวิดีโอที่จะใช้ในการเทรนโมเดล

แหล่งที่มา :https://www.kaggle.com/datasets/risangbaskoro/wlasl-processed



รูปที่ 2 แสดงแหล่งข้อมูลที่ทำการค้นหาวิดีโอที่จะใช้ในการเทรนโมเดล

แหล่งที่มา :https://paperswithcode.com/dataset/wlasl



รูปที่ 3 แหล่งข้อมูลที่ทำการศึกษาเพื่อจัดทำวิดีโอที่จะใช้ในการเทรนโมเดล

แหล่งที่มา :https://www.youtube.com/watch?v=4Ll3OtqAzyw

การเลือกประเภทโมเดล

เลือกใช้โมเดลประเภท Multilayer Perceptron (MLP)เนื่องมาจากลักษณะของข้อมูลที่ใช้ในการฝึกโมเดลข้อมูลที่ใช้ในการฝึกโมเดลนั้น อยู่ในรูปแบบของ array ซึ่งแสดงถึงตำแหน่งต่างๆ และมีจำนวน feature ค่อยข้างมาก การสร้างโมเดลโดยใช้ MLP จึงมีเหมาะสม นอกจากนี้โมเดล MLP มีข้อได้เปรียบในเรื่องเวลาที่ใช้ในการฝึกฝนโมเดล

การเลือกตัวแปร

ใช้ feature ทั้งหมดที่ได้จากการเตรียมข้อมูลในการฝึกฝนโมเดล เนื่องจากทุก feature มีผลต่อท่าทางสำหรับการใช้สื่อสารด้วยภาษามือ

กระบวนพัฒนาโมเดล

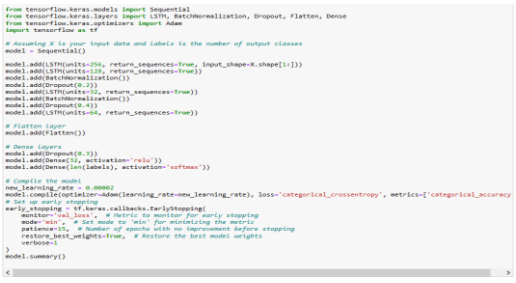
ในขั้นตอนแรกได้ทดลองใช้โมเดลแบบ Long Short-Term Memory (LSTM) โดยผลการจัดประสิทธิภาพจากโมเดล LSTM มี accuracy อยู่ที่ร้อยละ 2 โดยประมาณ

```
]: # Evaluate the model on train data
   print(' Test Loss: {:.6f}, Accuracy: {:.6f}'.format(*model.evaluate(X_test, y_test, verbose=0)))
   # model_evaluation_history = model.evaluate(X_test, y_test)

    Test Loss: 20.280115, Accuracy: 0.023173
```

รูปที่ 4 แสดงการวัดประสิทธิภาพจากการใช้ LSTM

ต่อมาทดลองใช้โมเดล MLP พบว่า accuracy ที่ได้มีค่าสูงขึ้นเมื่อเปรียบเทียบกับโมเดลที่ฝึกฝนด้วย LSTM



รูปที่ 5 model แรกที่สร้างขึ้น

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 180, 256)          266240

 lstm_1 (LSTM)               (None, 180, 128)          197120

 batch_normalization (BatchN  (None, 180, 128)         512
 ormalization)

 dropout (Dropout)           (None, 180, 128)          0

 lstm_2 (LSTM)               (None, 180, 32)           20608

 batch_normalization_1 (Batc  (None, 180, 32)          128
 hNormalization)

 dropout_1 (Dropout)         (None, 180, 32)           0

 lstm_3 (LSTM)               (None, 180, 64)           24832

 flatten (Flatten)           (None, 11520)             0

 dropout_2 (Dropout)         (None, 11520)             0

 dense (Dense)               (None, 32)                368672

 dense_1 (Dense)             (None, 10)                330

=================================================================
Total params: 878,442
Trainable params: 878,122
Non-trainable params: 320
_________________________________________________________________
```

รูปที่ 6 แสดงโครงสร้างของ model แรก

ในการพัฒนาโมเดล MLP มีการกำหนดจำนวน unit ใน layer ให้มีจำนวนมากในชั้นแรก และค่อยๆ ลดจำนวนลงในชั้นถัดไปซึ่งมีส่วนช่วยในการแก้ปัญหา overfitting แต่จากการทดสอบโมเดลในเบื้องต้น พบว่าโมเดลเกิดปัญหา overfitting จึงมีการใช้เทคนิค dropout โดยการสุ่มปิดการใช้งาน neuron บางส่วนในโมเดล นอกจากนี้ยังมีการใช้เทคนิค early stopping ช่วยตรวจสอบผลลัพธ์ของ loss function หาก loss function ไม่ได้ลดลง โมเดลจะหยุดการฝึกและมีการใช้ optimizer Adam ช่วยปรับปรุงประสิทธิภาพของโมเดล

In [14]:
```python
from tensorflow.keras.models  import Sequential
from tensorflow.keras.layers import Dense, Flatten, Dropout
from tensorflow.keras.optimizers.schedules import ExponentialDecay
from tensorflow.keras.optimizers import Adam
from keras.callbacks import EarlyStopping

model = Sequential()
model.add(Dense(512, input_shape=X.shape[1:], activation='relu'))
model.add(Dropout(rate=0.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(rate=0.5))
model.add(Dense(128, activation='relu'))
model.add(Dropout(rate=0.5))
model.add(Flatten())
model.add(Dense(len(word_index), activation='softmax'))

initial_learning_rate = 0.001
lr_schedule = ExponentialDecay(initial_learning_rate, decay_steps=10000, decay_rate=0.9)
early_stopping = EarlyStopping(
    monitor='loss',  # Metric to monitor for early stopping
    mode='min',  # Set mode to 'min' for minimizing the metric
    patience=15,  # Number of epochs with no improvement before stopping
    restore_best_weights=True,  # Restore the best model weights
    verbose=1
)

model.compile(optimizer=Adam(learning_rate=lr_schedule), loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

รูปที่ 7 model ที่สร้างขึ้นใหม่

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_6 (Dense)             (None, 180, 512)          2048

 dropout_4 (Dropout)         (None, 180, 512)          0

 dense_7 (Dense)             (None, 180, 256)          131328

 dropout_5 (Dropout)         (None, 180, 256)          0

 dense_8 (Dense)             (None, 180, 128)          32896

 dropout_6 (Dropout)         (None, 180, 128)          0

 flatten_1 (Flatten)         (None, 23040)             0

 dense_9 (Dense)             (None, 15)                345615

=================================================================
Total params: 511,887
Trainable params: 511,887
Non-trainable params: 0
_________________________________________________________________
```

รูปที่ 8 แสดงโครงสร้างของ model

```
In [17]:
# Evaluate the model on train data
print('Train Loss: {:.6f}, Accuracy: {:.6f}'.format(*model.evaluate(X_train, y_train, verbose=0)))
```

```
Train Loss: 0.377358, Accuracy: 0.876121
```

```
In [18]:
# Evaluate the model on test data
print('Test Loss: {:.6f}, Accuracy: {:.6f}'.format(*model.evaluate(X_test, y_test, verbose=0)))
model_evaluation_history = model.evaluate(X_test, y_test, verbose=0)
```

```
Test Loss: 0.434277, Accuracy: 0.859282
```

รูปที่ 9 แสดงประสิทธิภาพของโมเดล

การวิเคราะห์ผลลัพธ์

การวัดประสิทธิภาพของโมเดล

วัดประสิทธิภาพโมเดลโดยใช้อัตราการทำนายที่ถูกต้องจากการทำนายทั้งหมด หรือค่า accuracy

กระบวนการทดสอบโมเดล

ประกอบด้วย 2 ส่วน

1. การวัดประสิทธิภาพโดยการทดลองนำ instance บางส่วนไปที่ถูกแบ่งเป็น test set มาทำนายเพื่อหา accuracy
2. วัดประสิทธิภาพโดยแสดงท่าทางแบบ real-time

ตัวอย่างผลลัพธ์การทดสอบ

รูปที่ 10 แสดงท่าทาง “สวัสดี” เป็นภาษามือ

รูปที่ 11 แสดงท่าทาง “ใช่” เป็นภาษามือ

รูปที่ 12 แสดงท่าทาง “กิน” เป็นภาษามือ

สรุปผล

ผลลัพธ์ของโมเดลนั้นมีความเหมาะสมต่อการแก้ไขปัญหาที่กำหนด เนื่องจากเป็นการแปลภาษาตามเวลาจริงทำให้สามารถสามารถรับสารได้อย่างทันท่วงที ในอนาคตสามารถปรับปรุงให้เพิ่มจำนวนคำศัพท์ที่สามารถจำแนกได้ และตรวจจับได้อย่างแม่นยำมากยิ่งขึ้น

ข้อจำกัดและอุปสรรคในการทำโครงงาน

1. ข้อจำกัดของอุปกรณ์
    - คุณภาพของภาพที่ป้อนเข้าส่งผลต่อความแม่นยำในการคาดการณ์ผลลัพธ์ของโมเดล
2. ความไม่เสถียรของซอฟต์แวร์
    - การคาดการณ์ผลลัพธ์ของโมเดลอาจมีความผิดพลาดในบางสถานการณ์ เนื่องจากซอฟต์แวร์ไม่สามารถค้นหาจุดที่สำคัญในภาพได้
3. ชุดข้อมูล
    - ขนาดของชุดข้อมูลส่งผลต่อความแปรปรวนในการทำนาย

การทดลองทำซ้ำ

สามารถ clone git hub repository โดยใช้ URL https://github.com/zoneul/Word-Level-American-Sign-Language-Recognition โดยมีรายละเอียดดังต่อไปนี้

- สามารถ run ไฟล์ชื่อ “wlasl_recognition.ipynb” เพื่อใช้จำแนกคำผ่านกล้องแบบ real-time ได้ โดยต้องติดตั้งโมดูล mediapipe และ tensorflow ก่อน
- Dataset ที่เป็นวิดีโอทั้งหมดจะอยู่ใน folder “videos” และใช้ไฟล์ WLASL_data.json เพื่อบ่งบอกผลเฉลยของแต่ละวิดีโอ
- โค้ดส่วนที่ใช้ในการฝึกฝนโมเดลจะอยู่ในไฟล์ “wlasl_model.ipynb” ซึ่งจะมารถนำไปปรับปรุงและต่อยอดได้

บรรณานุกรม

150 Essential ASL Signs | Part 1 | Common and Descriptive Signs. สืบค้นเมื่อ[6/03/2024].จาก https://www.youtube.com/watch?v=4Ll3OtqAzyw.

MuteMotion: WLASL Translation Model.สืบค้นเมื่อ [6/03/2024]. จาก https://www.kaggle.com/code/abd0kamel/mutemotion-wlasl-translation-model

WLASL (World Level American Sign Language) Video.สืบค้นเมื่อ [6/03/2024]. จาก https://www.kaggle.com/datasets/risangbaskoro/wlasl-processed

WLASL (World Level American Sign Language) Videos.สืบค้นเมื่อ [6/03/2024]. จาก https://www.kaggle.com/datasets/gazquez/wlasl-processed.

WLASL: A large-scale dataset for Word-Level American Sign Language (WACV 20' Best Paper Honourable Mention).สืบค้นเมื่อ [6/03/2024].จาก https://github.com/dxli94/WLASL.

Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison.สืบค้นเมื่อ [6/03/2024].จากhttps://openaccess.thecvf.com/content_WACV_2020/papers/Li_Word-level_Deep_Sign_Language_Recognition_from_Video_A_New_Large-scale_WACV_2020_paper.pdf.