รายงาน

Optical Character Recognition to text

เสนอ

อาจารย์ ชาคริต วัชโรภาส

จัดทำโดย

นายชวิศ สิทธิธรรมจักษ์ 6410450842

นายวิทวัส พิณรัตน์ 6410451393

รายงานนี้เป็นส่วนหนึ่งของวิชา 01418364

Practical Deep Learning

ภาคต้น ปีการศึกษา 2566

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตบางเขน

1. ความชัดเจนของปัญหา

1. 1 ลักษณะของปัญหา

ลักษณะโครงงานเป็นปัญหา muticlassification โดยมี class ทั้งหมด 36 คลาส

1.2 ปัญหาที่ต้องการแก้ไข

ปัญหา เนื่องจากบุคคลบางคนถนัดในด้านการเขียนมากกว่าการพิมพ์จึงทำให้เวลาทำรายงานจำเป็นต้องใช้เวลามากกว่าปกติ หรือสำหรับบุคคลที่ต้องใช้งานการพิมพ์รายงานเป็นระยะเวลานานส่งผลต่อข้อมือ โครงานนี้จึงมุ่งหวังเพื่อที่จะช่วยประหยัดเวลาในการทำรายงานสำหรับผู้ที่ไม่ถนัดในการพิมพ์ และช่วยลดปัญหาเหล่านี้ให้น้อยลง

1.3 ประโยชน์ที่จะได้รับจากโครงงาน

กลุ่มเป้าหมาย คือ ต้องการช่วยในการทำรายงานของนักเรียนและนักศึกษาและผู้ใช้งานที่มีอายุเนื่องจากปัญหาด้านสุขภาพที่ได้กล่าวไปข้างต้น

2. การเก็บรวบรวมข้อมูล

2.1 แหล่งข้อมูล

ข้อมูลนำมาจากเว็บไซต์ kaggle.com

Dataset : English Handwritten Characters

ผู้ให้ข้อมูล : DHRUVIL DAVE

สามารถเข้าถึงข้อมูลได้จาก

https://www.kaggle.com/datasets/dhruvildave/english-handwritten-characters-dataset/data

2.2 ลักษณะของข้อมูล

ข้อมูลที่ได้มาจาก Dataset English Handwritten Characters เป็นข้อมูล

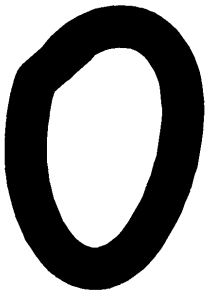
รูปภาพขนาด 1200*900 ทั้งหมด 3410 รูปภาพ สามารถแบ่ง 3 ประเภทดังนี้

1. ตัวเลข 0-9 อย่างละ 55 รูปภาพ
2. ตัวอักษรภาษาอังกฤษพิมพ์ใหญ่ A-Z อย่างละ 55 รูปภาพ
3. ตัวอักษรภาษาอังกฤษพิมพ์เล็ก a-z อย่างละ 55 รูปภาพ

โดยมี Feature ทั้งหมด 2 Feature คือ

1. ไฟล์รูปภาพ (ตัวอย่าง img001-001.png)
2. ผลเฉลยของภาพนั้น

ลักษณะของรูปภาพเป็นภาพพื้นหลังสีขาวและมีตัวอักษรหรือตัวเลขที่มีเส้นขอบสีดำเพียงตัวเดียวต่อ 1 รูป ตัวอย่างดังรูปด้านล่าง



รูปภาพแสดงถึงตัวอย่างของข้อมูลใน dataset

2.3 ข้อจำกัด

ในการสร้างโมเดลจะนำข้อมูลตัวอักษรภาษาอังกฤษพิมพ์ใหญ่มาแปลงเป็นตัวพิมพ์เล็กเพื่อลดจำนวน class ในการ classifier ลงจากทั้งหมด 62 เหลือเพียง

32 class แต่จำนวนรูปทั้งหมดยังคงเท่าเดิม

2.4 การเก็บข้อมูลเพิ่มเติม

เพื่อเป็นการทดสอบโมเดลที่ได้จากแบบจำลองจึงมีการเก็บข้อมูลเพิ่มโดยข้อมูลที่เก็บเพิ่มจะเป็นเครื่องชาร์บ (#) จำนวน 20 รูปเพื่อนำไปใช้เป็น class ในการทดสอบประสิทธิภาพของโมเดลว่าสามารถเทรนข้อมูลเพิ่มขึ้นและนำไปใช้งานเพิ่มเติมได้หรือไม่โดยลักษณะของข้อมูลประกอบด้วย

Feature ทั้งหมด 2 Feature คือ

1. ไฟล์รูปภาพ (ตัวอย่าง img001-001.png)
2. ผลเฉลยของภาพนั้น

รวมทั้งหมด 3410+20 = 3430 รูปภาพ และมี class ทั้งหมด 37 class

3. การแบ่งชุดข้อมูล

3.1 สัดส่วนในการแบ่งข้อมูล

แบ่งข้อมูลออกเป็นชุดข้อมูลสำหรับเทรนและชุดข้อมูลสำหรับเทสโดยมีสัดส่วนในการแบ่ง 80% สำหรับชุดข้อมูลเทรนและ 20% สำหรับชุดข้อมูลเทสโดยข้อมูล

จะถูก shuffle ก่อนที่จะถูกส่งไปให้โมเดลเรียนรู้

3.2 จำนวนข้อมูลในแต่ละเซต

ชุดข้อมูลเทรน : 2744 รูป

ชุดข้อมูลเทส : 686 รูป

4. การเลือกประเภทและพัฒนาโมเดล

โมเดลที่เลือกใช้ในการทำแบบจำลองนี้คือ CNN และ MLP เหตุผลที่เลือกใช้ CNN เนื่องจาก input ที่ถูกนำมาเทรนโมเดลเป็นประเภทของรูปภาพเราจึงเห็นว่าควรใช้เทคนิค CNN เนื่องจากเป็นเทคนิคที่ได้รับความนิยมและสามารถทำงานได้ดีกับรูปภาพ

และเหตุผลที่เลือก MLP เนื่องจากกลุ่มของผู้นำเสนอมีความเข้าใจและมีประสบการณ์ในได้ใช้งาน MLP มากกว่า เทคนิคอื่นๆ เช่น LSTM, RNN, RCNN จึงคิดว่าสามารถปรับแต่งโมเดลได้ง่ายกว่าเทคนิคอื่นๆ

4.1 การสร้างโมเดลและการปรับพารามิเตอร์

เริ่มจากการทดลองปรับ Architectures ของ CNN โดยใช้ AlexNet, VGG แล้วดูผลลัพธ์ที่ได้หลังจากที่ได้ Architectures ที่ต้องการแล้วนั้นคือ VGG เราได้ทำการลดขนาดลอง parameter ใน VGG ลงเพื่อลดระยะเวลาในการประมวลผลหลังจากนั้นเราจะทำการไปปรับแต่ง layers ของ MLP เช่นการเพิ่มหรือลดจำนวน node ในแต่ละ layers ปรับจำนวน epoch และทำการ regularize โดยการทำการ dropout

```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_202 (Conv2D)         (None, 64, 64, 64)        1792

 max_pooling2d_141 (MaxPool  (None, 32, 32, 64)        0
 ing2D)

 conv2d_203 (Conv2D)         (None, 32, 32, 128)       73856

 max_pooling2d_142 (MaxPool  (None, 16, 16, 128)       0
 ing2D)

 conv2d_204 (Conv2D)         (None, 16, 16, 128)       147584

 max_pooling2d_143 (MaxPool  (None, 8, 8, 128)         0
 ing2D)

 flatten_39 (Flatten)        (None, 8192)              0

 dense_116 (Dense)           (None, 1024)              8389632

 dropout_13 (Dropout)        (None, 1024)              0

...
Total params: 9700389 (37.00 MB)
Trainable params: 9700389 (37.00 MB)
Non-trainable params: 0 (0.00 Byte)
```

รูปภาพแสดงถึงจำนวน parameter ที่ใช้

5. การวิเคราะห์ผลลัพธ์

ในการวัดประสิทธิภาพของโมเดลวัดจากค่า Accuracy ที่ได้

5.1 การทดสอบโมเดล

นำโมเดลที่ได้จากการเทรนมาทดสอบโดยการนำข้อความที่ถูกเขียนขึ้นมาเป็นรูปภาพเข้ามาเป็น input ของโมเดลและนำผลลัพธ์จากการทำนายของโมเดลทีละตัวอักษรของโมเดลไปใช้เปรียบเทียบกับผลเฉลยว่ามีค่าตรงกันหรือไม่ โดยมีเกณฑ์การตัดสินจากจำนวนตัวอักษรที่ตรงกับผลเฉลยมากที่สุด

5.2 วิเคราะห์และสรุปผล

เนื่องจากโมเดลที่ได้ยังไม่สามารถใช้งานได้ตามวัตถุประสงค์ของผู้จัดทำจึงสามารสรุปได้ว่าโมเดลที่ได้ยังไม่เหมาะสมกับการแก้ปัญหาตามวัตถุประสงค์เนื่องจากมีความผิดพลาดในการทำนายมากเกินกว่าที่ได้ตั้งเป้าไว้ในการทำนายตัวอักษรอาจจำเป็นต้องใช้เทคนิคในการจดจำตัวอักษรก่อนหน้าโดยการใช้ LSTM หรือ GRU เพื่อเพิ่มประสิทธิภาพของโมเดลให้ดียิ่งขึ้นและมีการเก็บข้อมูลเพิ่มเติมเพื่อให้มีตัวอย่างในการเรียนรู้ของโมเดลมากยิ่งขึ้นหรือทำการใช้ tranfer model ของ yolo มาปรับใช้เพื่อให้ได้ประสิทธิภาพที่ดียิ่งขึ้น

5.3 ขั้นตอนในใช้งานโมเดล

1. https://github.com/BoostChavit/DeepLearning-Project.git

   clone project จากลิงค์ด้านบน
2. ติดตั้ง library กรณีที่เครื่องผู้ใช้ยังไม่มี
   - cv2
   - pandas
   - numpy
   - keras, tensorflow
   - matplotlib

- sklearn

3. ทดสอบลอง run program ที่ main.ipynb
4. หลังจากที่ทดสอบ run model ไปแล้ว 1 ครั้งในกรณีที่ไม่มีอะไรผิดพลาดและไม่มี error จะได้ไฟล์ SSFF_model.h5 หากผู้ใช้ต้องการทดสอบโมเดลหรือนำโมเดลไปใช้สามารถใช้ผ่านโมเดล SSFF_model.h5 นี้ได้เลยโดยไม่ต้องเทรนใหม่อีกครั้งโดยมีนำโค๊ดด้านล่างไปใส่ไว้บรรทัดล่างสุดหลังจากการเทรนแล้วสามารถใช้งานได้เลย

6. ข้อจำกัดและอุปสรรค

ปัญหาการทำ bounding box

ข้อจำกัดของโมเดลคือในการทดสอบโมเดลเราจะนำข้อความเข้ามาเป็น input ไฟล์ภาพและทำการทำ bounding box เพื่อแยกเป็นตัวอักษรทีละตัวอักษรแล้วจึงค่อยๆส่งเข้าไปให้โมเดลทำนายทีละตัวอักษรปัญหาเกิดจากการแบ่ง boundingbox ของตัว i เนื่องจากเป็นเส้นที่ไม่ได้เชื่อมต่อกันทำให้เมื่อทำการทำ bounding box แล้วถูกแบ่งออกมาเป็น 2 ตัวอักษรที่ส่งไปให้โมเดลทำนายและโมเดลไม่เคยเห็นตัวอักษร “.” จึงไม่สามารถทำนายได้ถูกต้องทำให้เกิดเป็น noise และมีปัญหาหากข้อความที่เข้ามายาวเกินกว่า 1 บรรทัด

ปัญหาความคล้ายคลึงกันของ class

เนื่องจากในการเทรนโมเดลเราเลือกเทรนจากตัวอักษรภาษาอังกฤษและมีการทำ

data augmentation โดยการ rotate ไม่เกิน 30 องศา เการ shif รูปแนวตั่งแนว
นอนและมีการ zoom เข้าออก 40% เพื่อโมเดลมีความแม่นยำมากขึ้นแต่จะมีบาง class ที่เมื่อนำไปหมุนแล้วเกิดความคล้ายคลึงกันของ class เช่นเลข 5 และตัว s
ทำให้โมเดลที่ได้มีการทำนายตรงส่วนนี้ผิดค่อนข้างเยอะหรือตัว i ทั้งพิมพ์เล็กและพิมพ์ใหญ่ กับตัว l กรณีของตัว i พิมพ์เล็ก (lower case) ปัญหาเกิดจากการแบ่งของ bounding box ทำให้รูปที่ถูกส่งไปทำนายคล้ายกับตัว l และกรณีของตัว I พิมใหญ่(upper case) ที่มีความคล้ายกับตัว l อยู่แล้วจึงส่งผลให้ผลลัพธ์ที่ได้จาการทำนายมีความคลาดเคลื่อนระหว่าง class ค่อนข้างมาก

ปัญหาในการใช้ yoloV8

เนื่องจากประสิทธิภาพของโมเดลที่ได้ไม่ได้ตรงตามเป้าที่วางไว้ทางผู้จัดทำโครงงานจึงลองมองหาทางออกเพิ่มเติมในระยะเวลาอันใกล้และได้พบกับ
tranfer model ที่ได้รับความนิยมคือ โมเดล yolo ทางผู้จัดทำจึงได้นำโมเดล yoloV8 มาทดสอบในการใช้งานตั่งแต่การแปลง dataset เป็น custom data ตาม format ของ yoloV8 เพื่อให้สามารถนำมาใช้งานได้และมีการปรับปรุงพารามิเตอร์ภายในโมเดลเพื่อผลลัพธ์ที่ดียิ่งขึ้นแต่งเนื่องจากปัญหาทางด้านเวลาและความเข้าใจจึงส่งผลให้ผลลัพธ์ที่ได้จากการเทรนนั้นแย่กว่าโมเดลต้นแบบที่ได้จัดทำในตอนแรก