

Associate in Practical Deep Learning 01418364

ຮະບບຮູ້ຈໍາວັກເຊຣຄັນຈິຈາກລາຍເມືອ ເຂີຍນ ແບບວັຈະຮີຍະ *(NIHONGO-NET)*

Team : ຕຽບຕຽບຕຽບຕຽບຕຽບຕຽບຕຽບຕຽບຕຽບຕຽບ

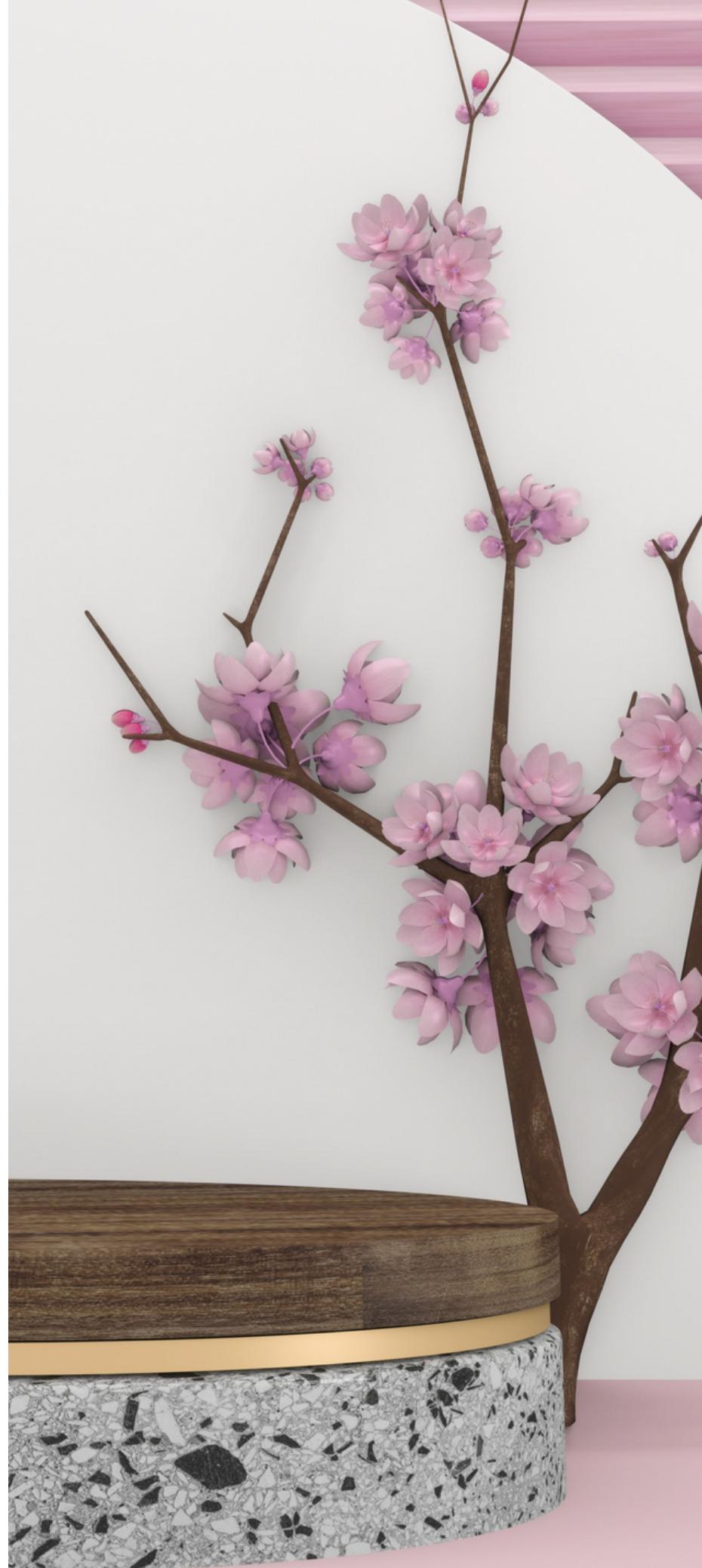
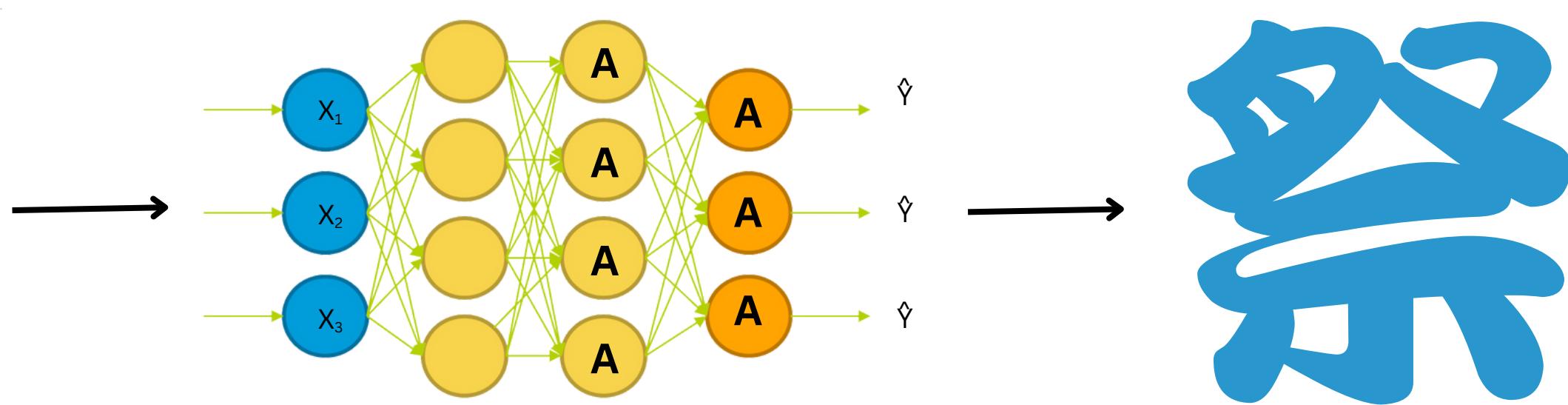
ที่มาและปัญหาของโครงงาน

- ภาษาญี่ปุ่น เป็นภาษาที่ได้รับความนิยมจากชาวต่างชาติ แต่ตัวอักษรคันจิมีรูปร่างซับซ้อน
- OCR เทคโนโลยีสกัดตัวอักษรจากภาพ
- ความท้าทาย: แยกแยะตัวอักษรคันจิที่คล้ายคลึงกัน
- โปรเจค NihonGO! ศึกษาการใช้ Convolutional Neural Network (CNN)
- เป้าหมายเพื่อรู้จำประเภทของตัวอักษรคันจิแบบเขียนมือ
- ประโยชน์เพื่อพัฒนาและปรับใช้กับระบบ OCR ในอนาคต



วัตถุประสังค์ของโครงงาน

- เพื่อนำไปประยุกต์ใช้เป็นระบบกำนายตัวอักษรคันจิ auto-correction
- เพื่อศึกษาประสิทธิภาพของการใช้เทคนิคของ CNN ไปใช้ในการจำแนกประเภทของตัวอักษรคันจิ ในรูปแบบของการจำแนกรูปภาพ



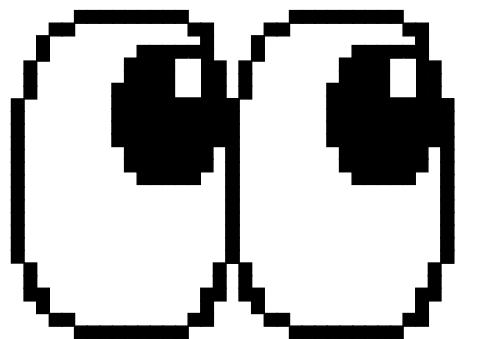
ประโยชน์ที่ได้รับ

- เป็นแนวทางในการพัฒนาต่อยอดต่อในเทคโนโลยีน ၅ เช่น ในลักษณะของการทำ OCR ในการวิเคราะห์ในลักษณะของประโยชน์
- ต่อยอดไปในด้านการนำไปใช้งาน หรือเป็นกรณีในการศึกษา เช่น การประยุกต์ใช้ในงานด้านการแปลภาษาต่อไป



ผู้ที่ได้รับประโยชน์

1. บุคคลทั่วไปที่กำลังศึกษาเทคนิคการทำ OCR
2. นักเรียนและนักศึกษา



Main resources from

- ETL Dataset จากสถาบันวิทยาศาสตร์และเทคโนโลยีอุตสาหกรรมขั้นสูงแห่งชาติของญี่ปุ่น (AIST)

The screenshot shows the ETL Dataset interface. On the left, there is a grid of handwritten Chinese character '前' samples. Each sample has a file name below it: ETL8G_076729.pn9, ETL8G_077685.pn9, ETL8G_078641.pn9, ETL8G_079597.pn9, ETL8G_081509.pn9, ETL8G_082465.pn9, ETL8G_083421.pn9, and ETL8G_084377.pn9. To the right of the samples is an email inbox entry from ETLCDB. The email subject is 'to me, etlcdb-ml'. The body of the email is:

Dear Tanaanan Chalermpan / Tanaanan Chalermpan様
下記の情報を登録させていただきました。
Name / 氏名: Tanaanan Chalermpan
E-mail / メールアドレス:
Organization / 所属: Kasetsart University
Country / 国: Thailand
Purpose / 使用目的: I would like to use dataset doing and learnig on final project learning on Deep learning subject at faculty of Science Kasetsart University Thailand .
Date-Time / 日時: February 21, 2024 20:30

電総研文字データベースは下記のURLからダウンロードできます。

5 • ข้อมูลตัวอักษรเขียนคันจิจำนวน 160 ภาพต่อ 1 labels



© Loki Vu

Datasets details

- คันจิจำนวน 81 ตัวจากการสอบวัดระดับภาษาญี่ปุ่น JLPT-N5 เนื่องจากตัวอักษรที่มีลักษณะที่หลากหลาย และมีความคล้ายคลึงกัน

JLPT Kanji N5 all (80 kanji)

[DETAILS VIEW](#) [QUICK STUDY](#) [FLASHCARDS](#) [MY MASTERY](#) [DOWNLOAD](#)

人	一	日	大	年	出	本	中	子	見	国	上	分	生	行	二	間	時	氣	十	女	三	前	入	小
後	長	下	学	月	何	来	話	山	高	今	書	五	名	金	男	外	四	先	川	東	聞	語	九	食
八	水	天	木	六	万	白	七	円	電	父	北	車	母	半	百	土	西	諺	千	校	右	南	左	友
火	每	雨	休	午																				

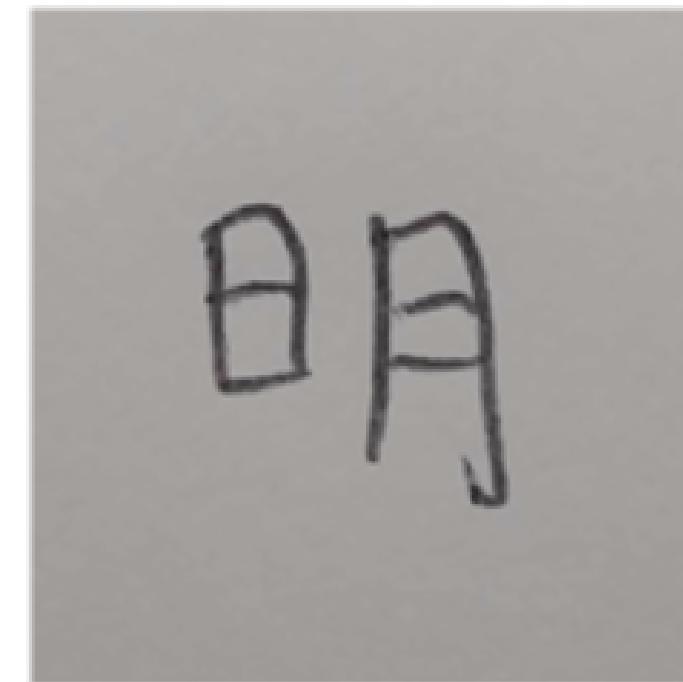
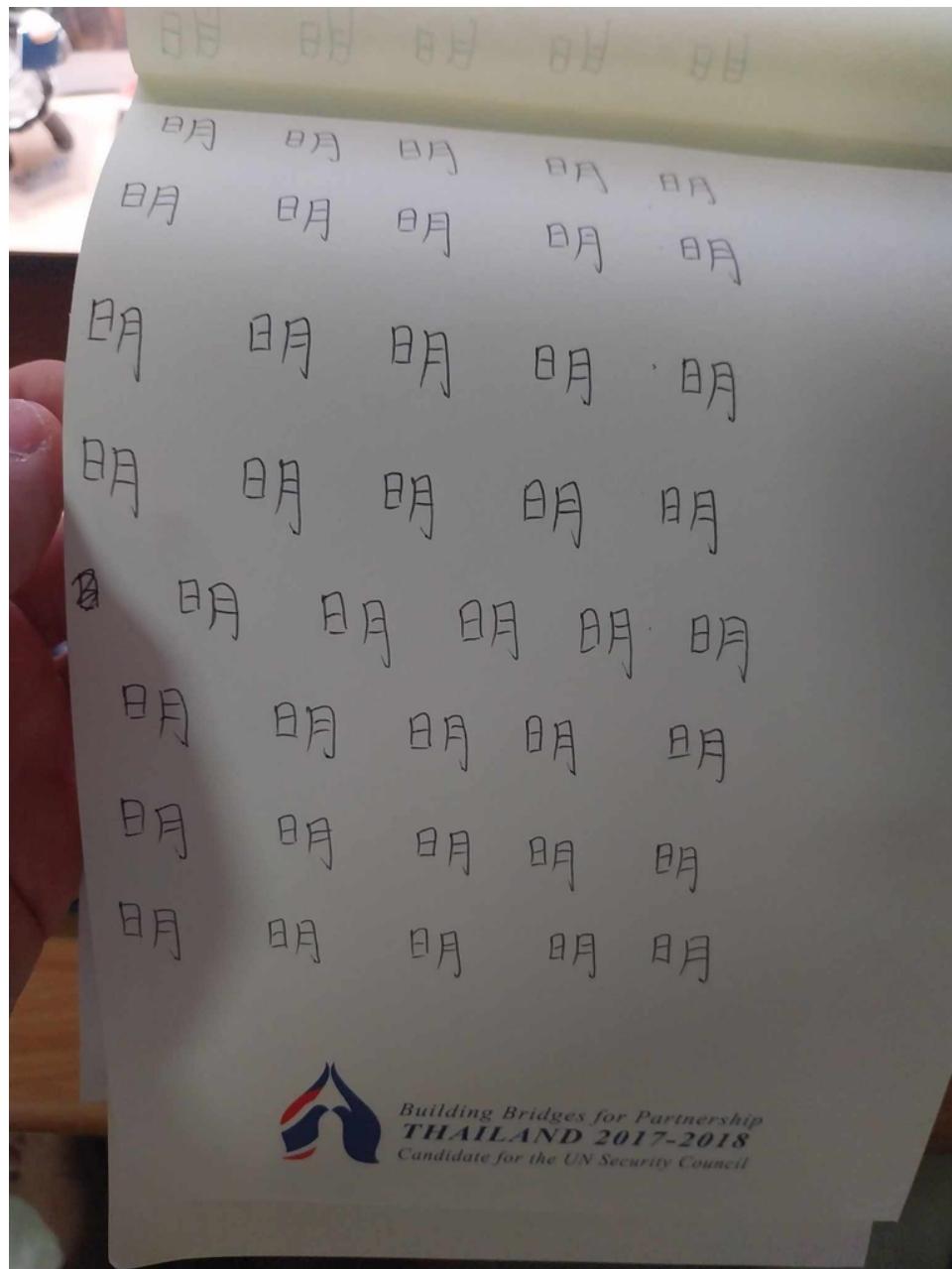
+

明



Additional resources from

- กำการเขียนขึ้นมาเองจำนวน 160 ตัวอักษร ต่อหนึ่งตัวอักษรคันจิ



Data preprocessing

1. Data cleaning คัดข้อมูล Sample สำหรับเพื่อที่จะนำไป Training กับ โมเดล
2. Image processing โดยใช้ binary-threshold เพื่อเร่งค่าสีให้มีความชัดเจนมากยิ่งขึ้นในกรณีที่ภาพ datasets หรือ input data ภาพมีความจำจริง
3. การแบ่งข้อมูล โดยใช้ Data Augmentation
 - 2.1) Training set
 - 2.2) Validation set
 - 2.3) Test set



Image Preprocessing

- ทำ binary-thresholding ในการ mapping ค่าให้อยู่ในช่วงของค่า 0 กับ 1 จาก OpenCV2

```
import cv2
import numpy as np
image1 = cv2.imread(img_path)

img = cv2.cvtColor(image1, cv2.COLOR_BGR2GRAY)

ret, thresh1 = cv2.threshold(img, 255-0.1, 255, cv2.THRESH_BINARY)
```

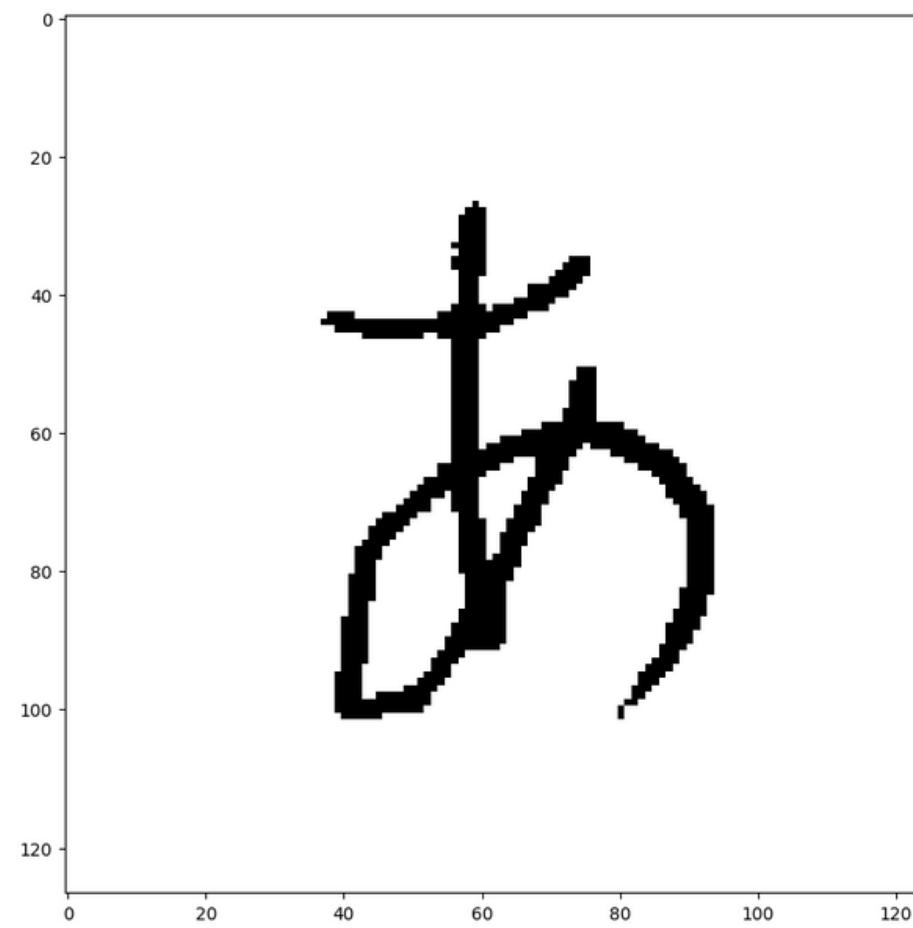
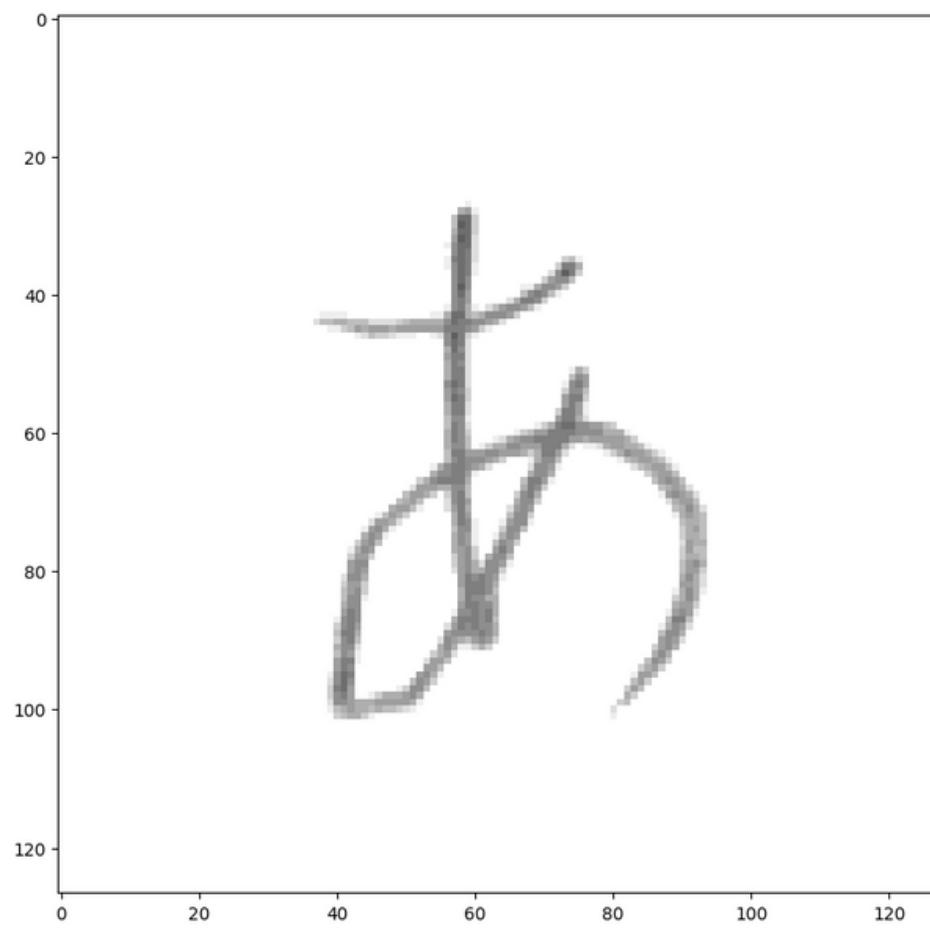
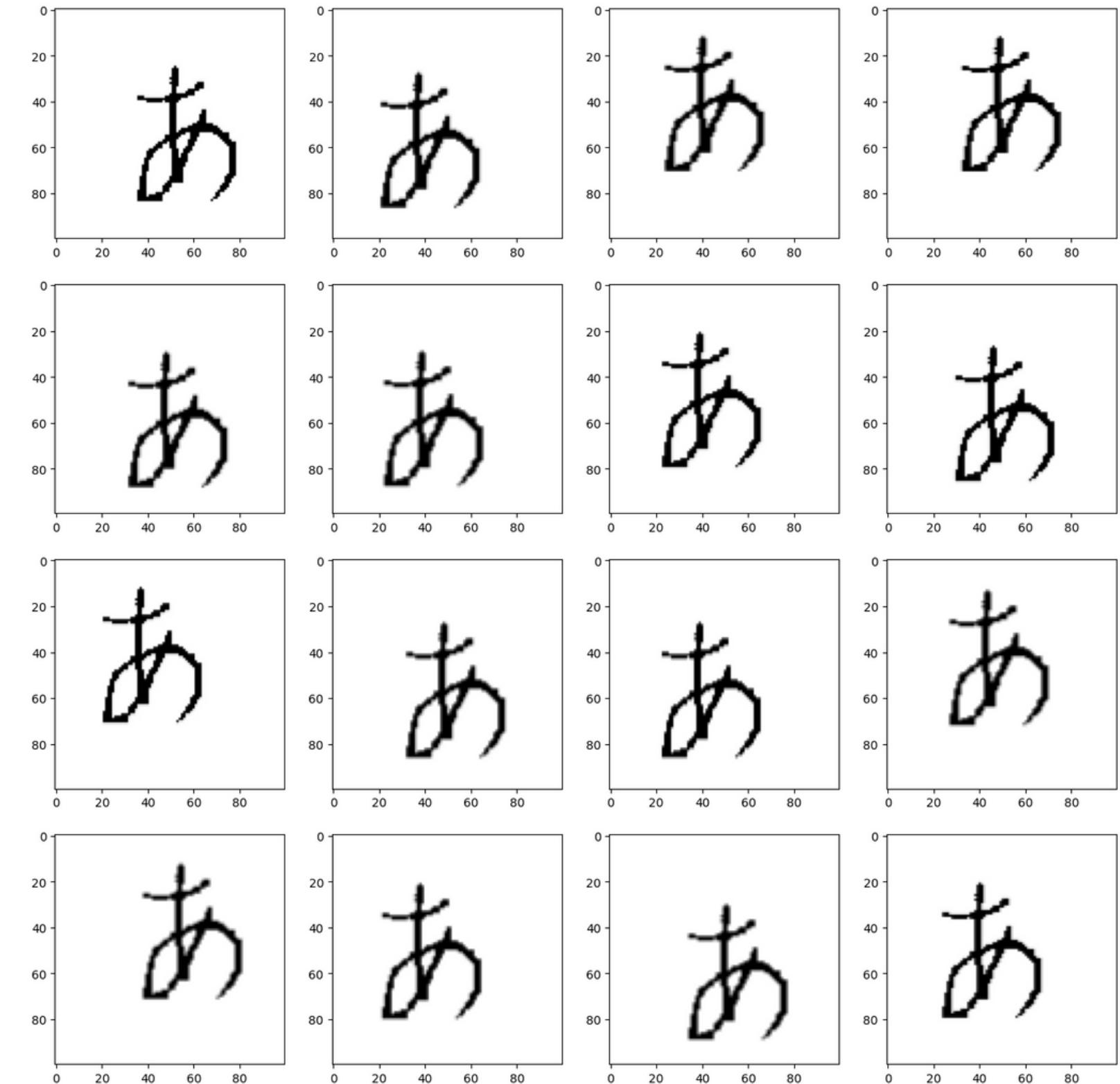


Image Augmentation

```
tf.keras.preprocessing.image.ImageDataGenerator(  
    height_shift_range=0.1,  
    width_shift_range=0.1,  
    zoom_range=0.1,  
    rescaling = 1./255
```

- zoom_range = 0.1
- height_shift_range = 0.1
- width_shift_range= 0.1
- rescaling = True



Creating a model

โมเดล Pre-Train (VGG16)

- โหลดโมเดล VGG16 ที่ผ่านการฝึกกับชุดข้อมูล ImageNet
- ตั้งค่าโมเดลให้ไม่สามารถเรียนรู้ (trainable) ได้
- รักษาค่าของน้ำหนัก (weights) ที่ถูกโหลดมาจากโมเดล VGG16

Layers Fine-Tune

- Global Average Pooling 2D: ลดขนาดของ feature maps
- Fully connected layer: เรียนรู้และปรับค่า
- Dropout: ลดโอกาสการเกิด overfitting
- Softmax: ผลลัพธ์เป็นความน่าจะเป็นของแต่ละคลาส



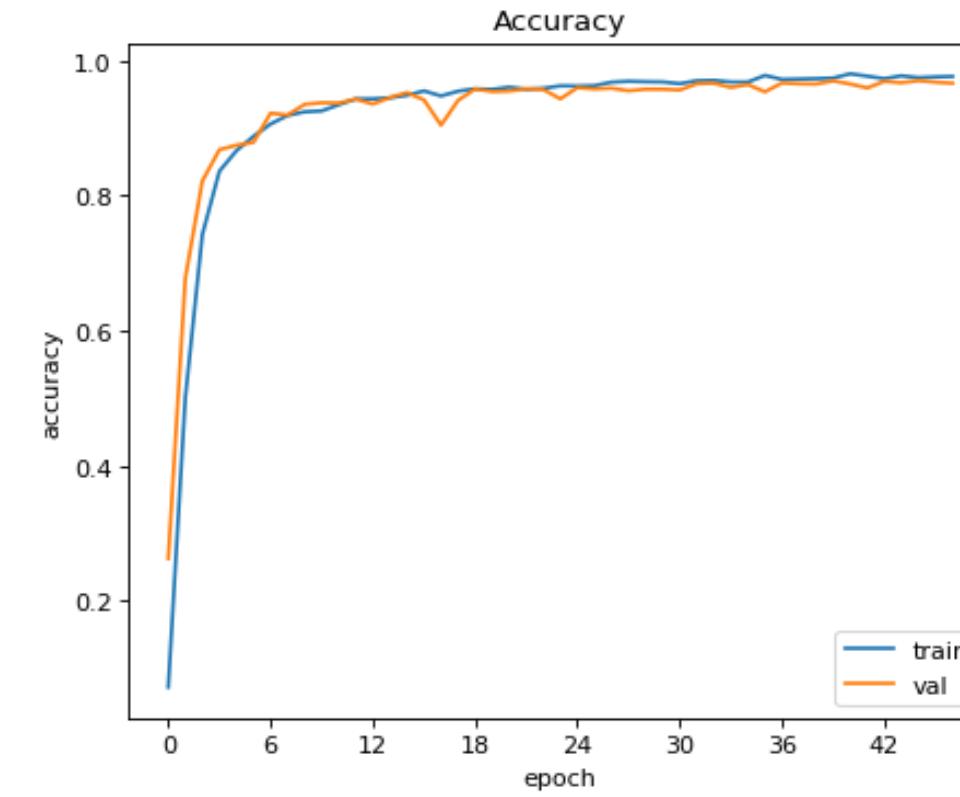
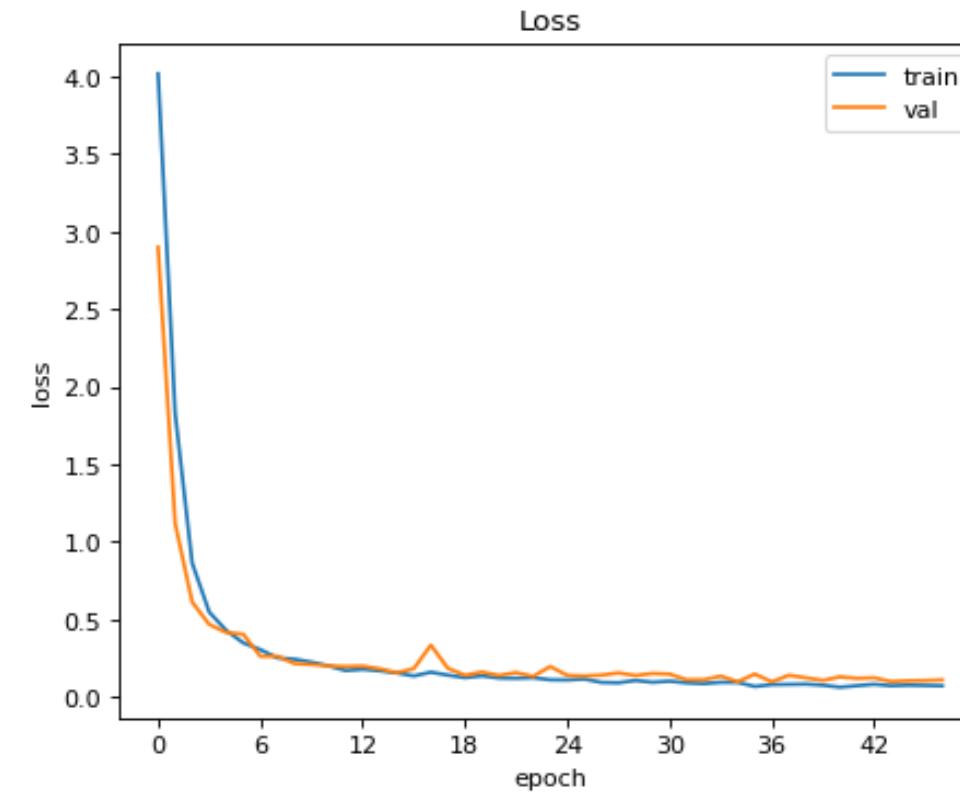
Creating a model (ต่อ)

โมเดล LeNet-5

- โมเดล CNN อ้างอิงจาก LeNet-5
- Convolutional layers: เรียนรู้คุณลักษณะของภาพ
- Max pooling layers: ลดขนาดภาพและลดความซับซ้อน
- Dropout: ป้องกันการ overfitting
- Flatten: แปลงข้อมูลจากพีเจ้อร์
- Dense layers: เรียนรู้และจำแนกประเภท
- Softmax: ผลลัพธ์เป็นความน่าจะเป็นของแต่ละคลาส



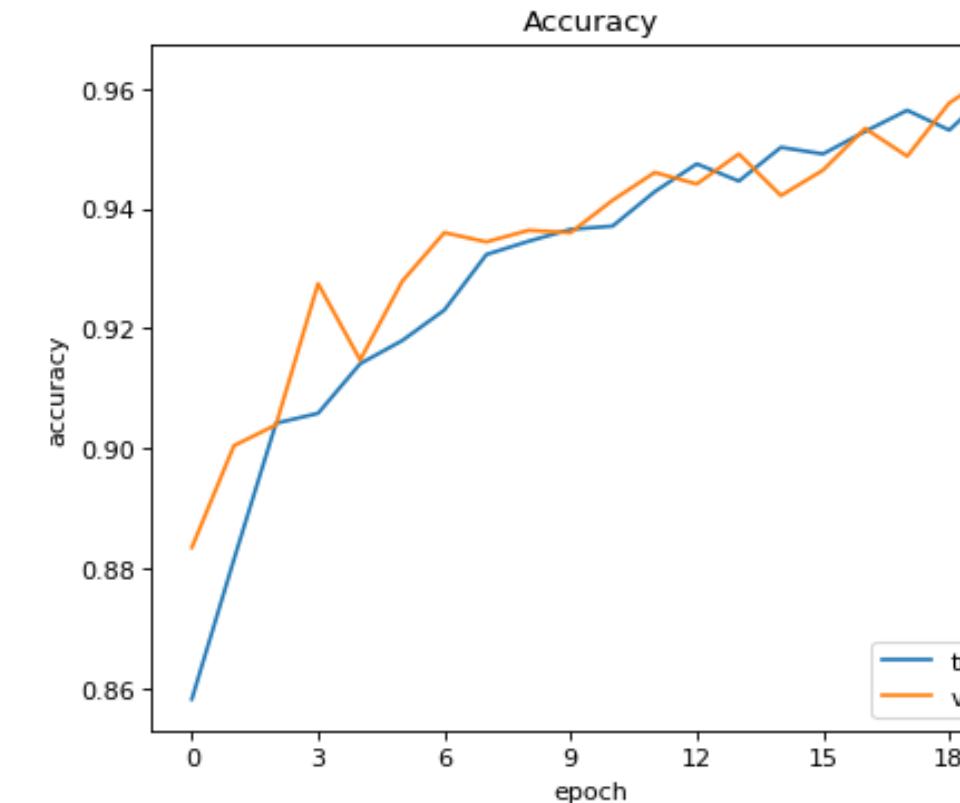
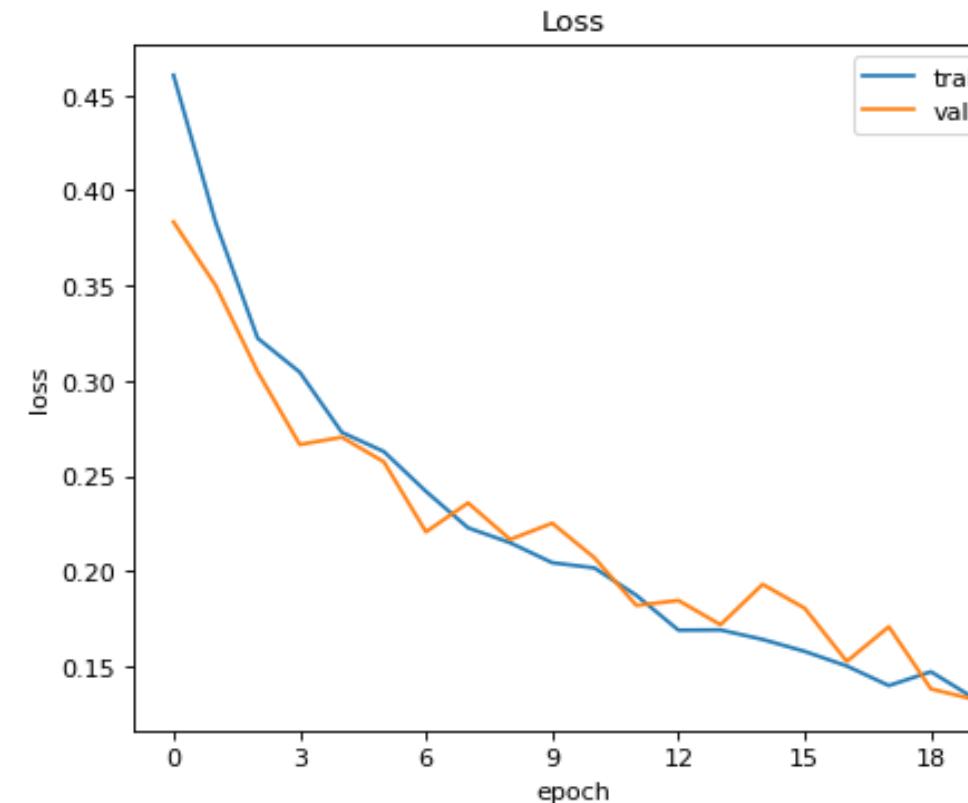
Performance Measurement



VGG16 (ImageNet)

- Dropout(0.5)

```
Train Loss: 0.038875, Accuracy: 0.987212
Validation Loss: 0.099707, Accuracy: 0.966821
Test Loss: 0.022804, Accuracy: 0.992012
```



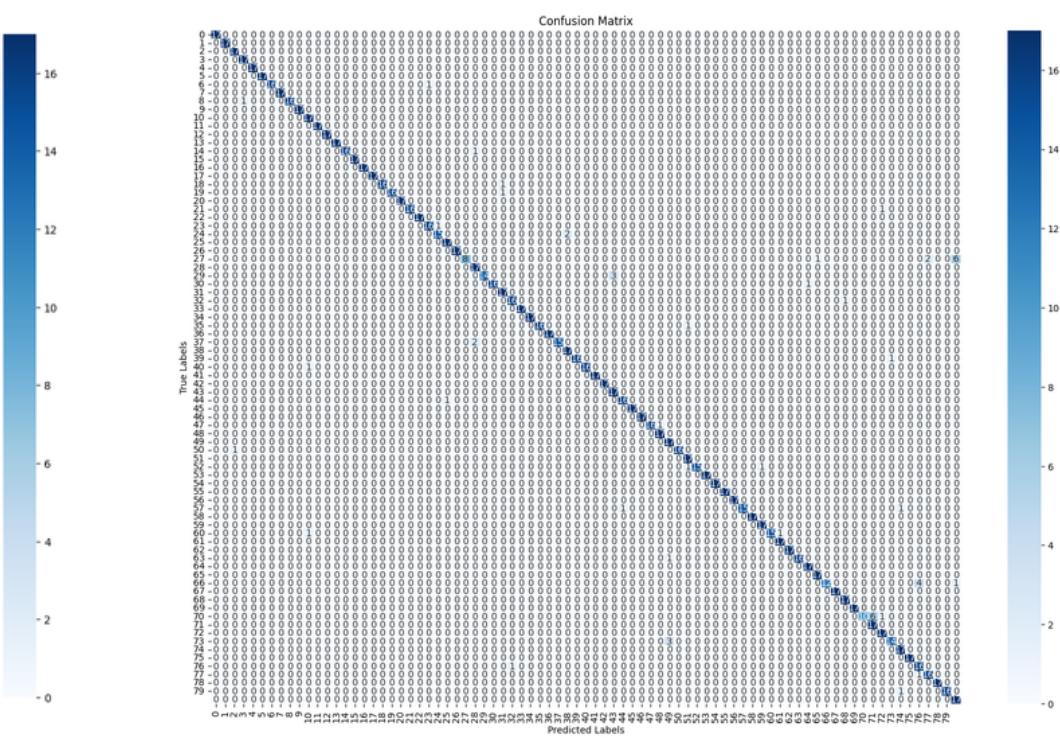
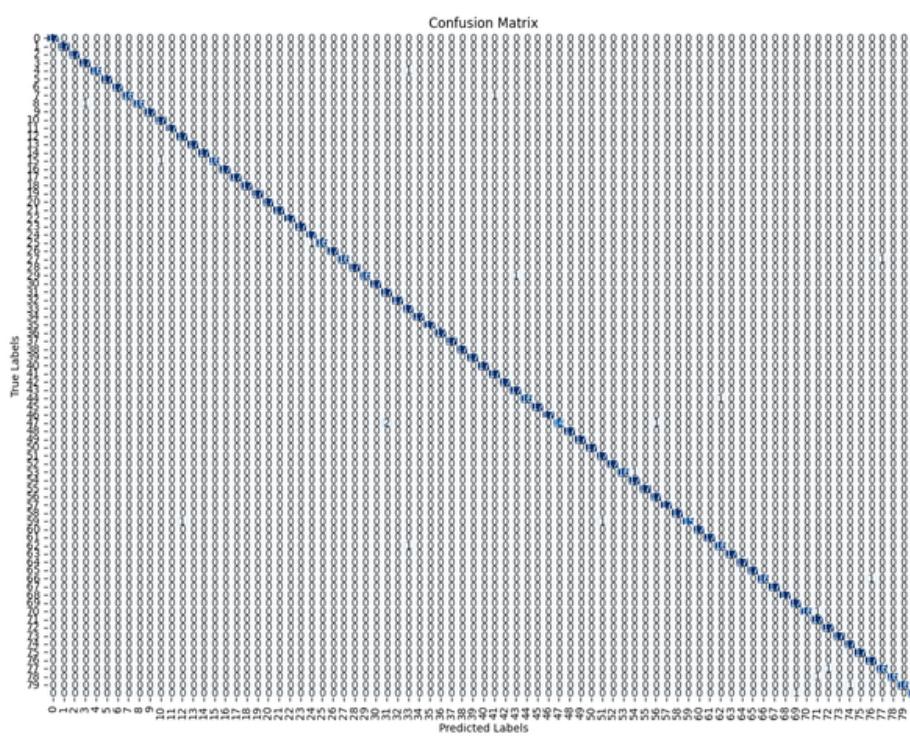
LeNet-5

- Dropout(0.25)

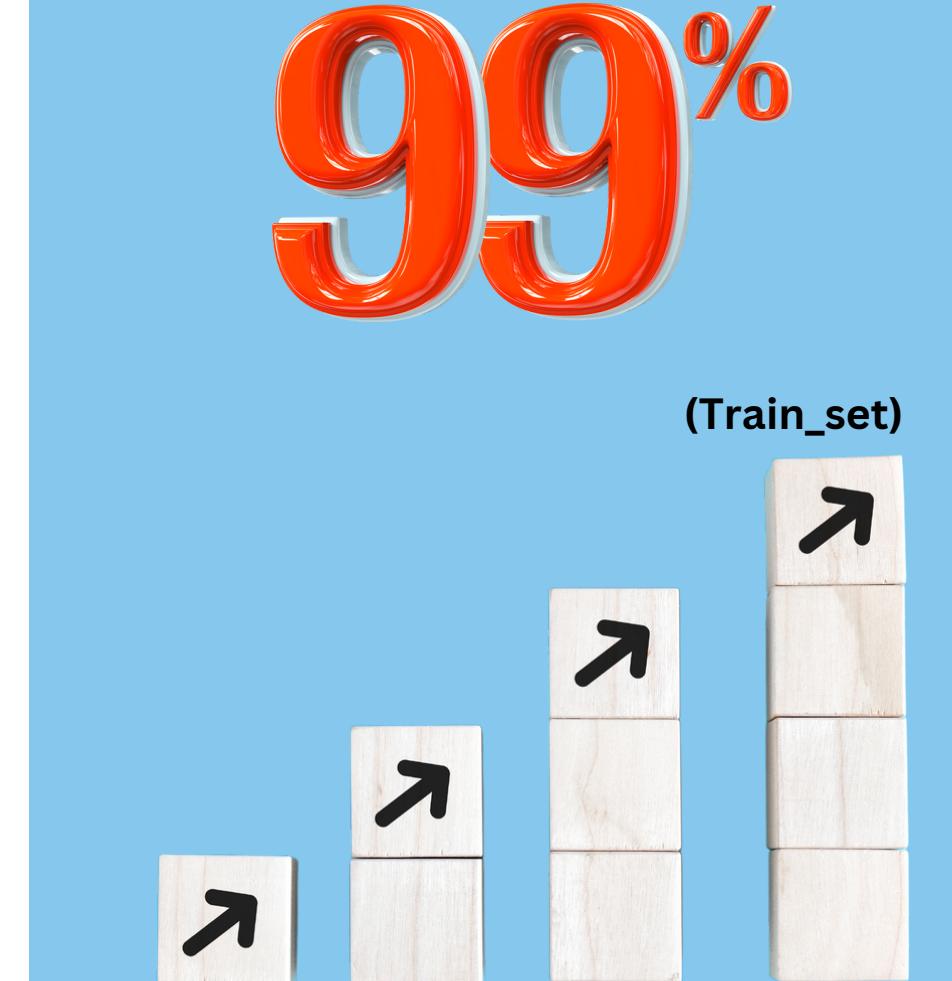
```
Train Loss: 0.056786, Accuracy: 0.982472
Validation Loss: 0.163081, Accuracy: 0.952546
Test Loss: 0.050809, Accuracy: 0.983297
```

Performance Measurement

Model	Validation Accuracy	Validation Loss	Testing Accuracy	Testing Loss
Lenet-5	96%	9%	99.2%	2.2%
VGG16	95%	16%	98%	5%



Loss Function



Performance Measurement (ต่อ)

木 ตับไม้ , 本 หนังสือ, 日 วัน, 明 สว่าง, 時 ชั่วโมง

Model	木	本	日	明	時	Total
LeNet-5	5	4	5	4	5	23/25
VGG16	5	5	5	3	3	21/25

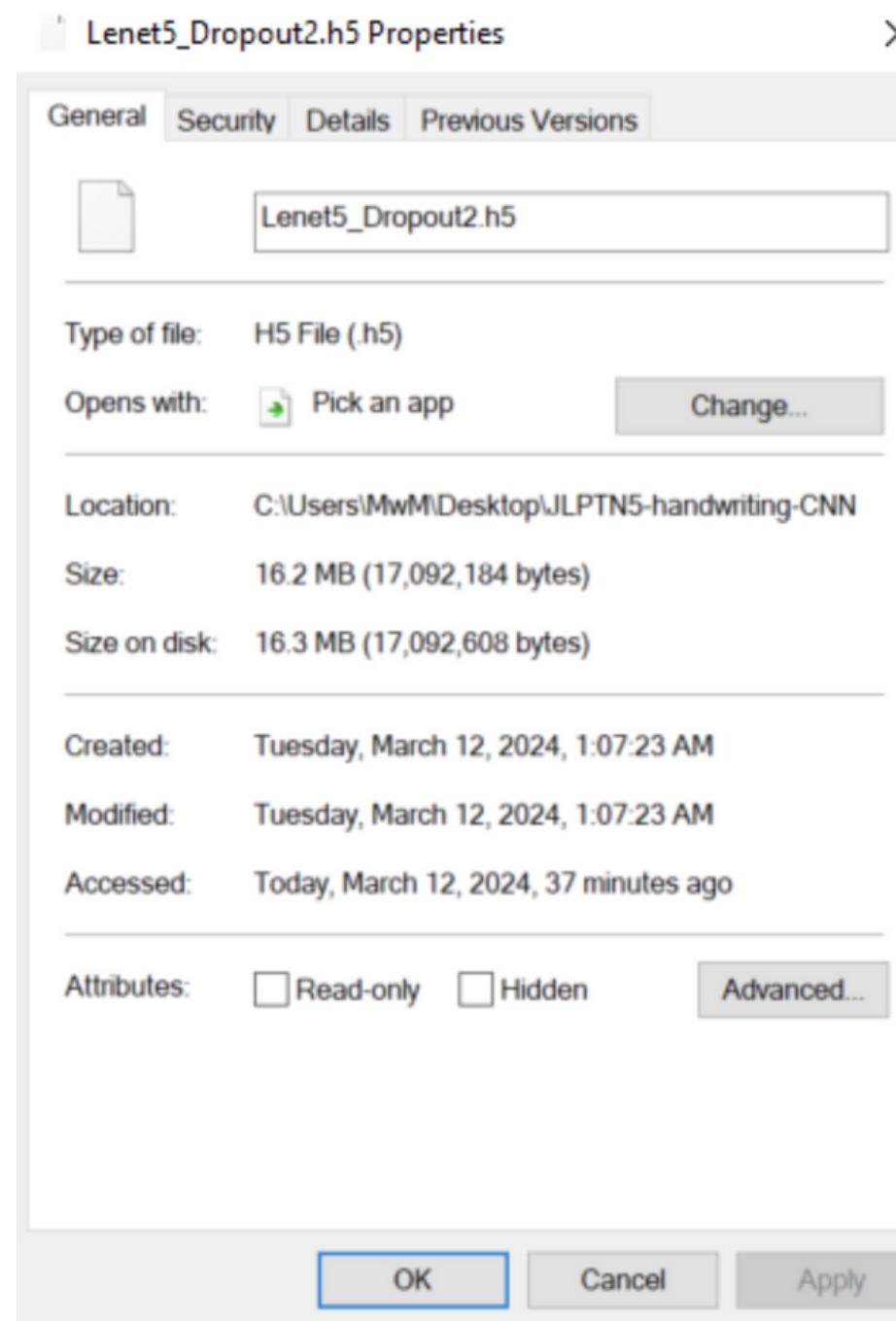
- จากการทดลองสรุปได้ว่าจากตัวอย่างตัวอักษรคันจิที่มีลักษณะที่ใกล้เคียงกัน 5 ตัวอักษร โดย VGG16 คิดเป็น 84 % LeNet-5 คิดเป็น 92 %



Performance Measurement

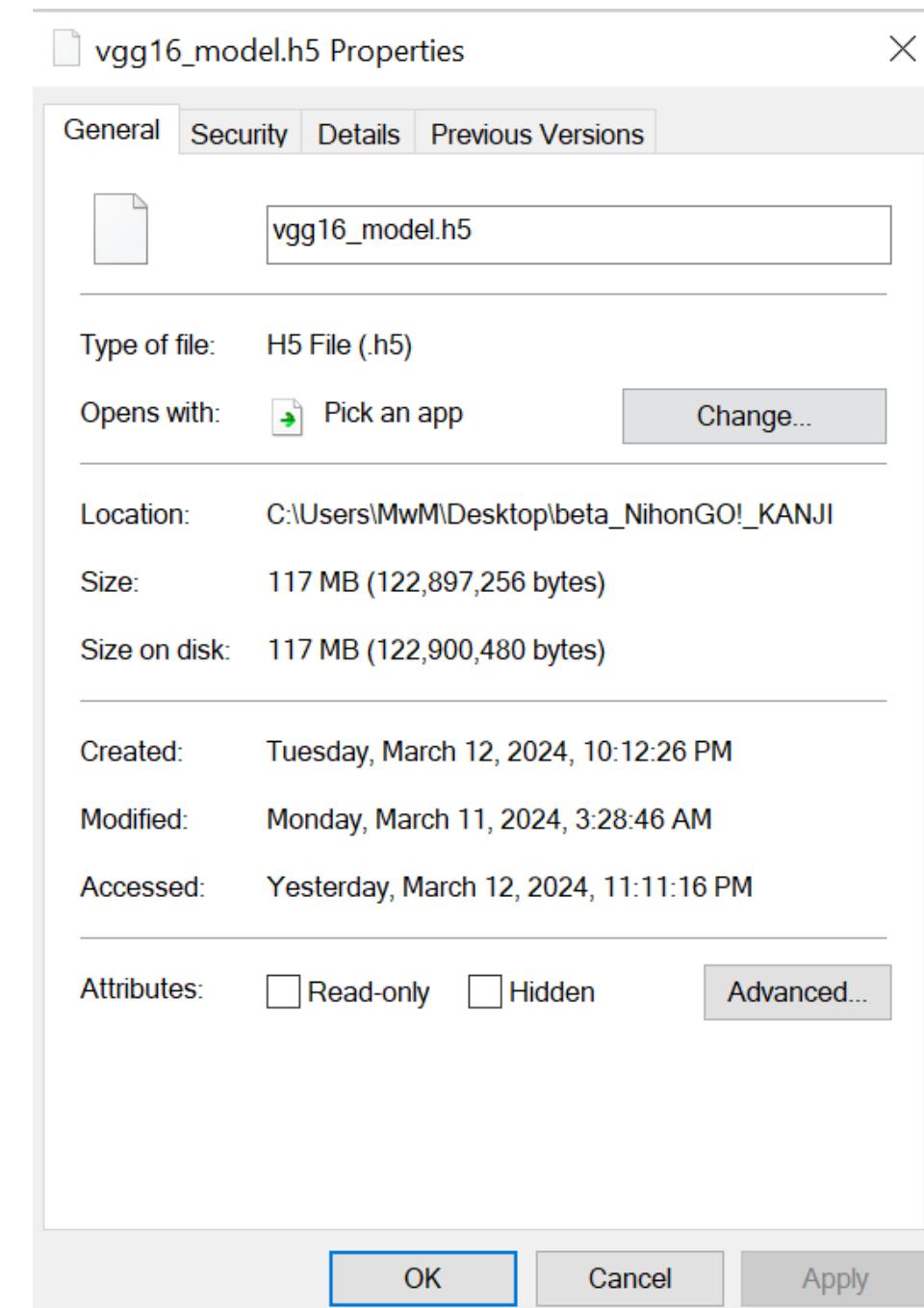
(ciθ)

LeNet-5



16.2 MB

VGG16



117 MB



Conclusion.

Layer (type)	Output Shape	Param #
<hr/>		
conv2d (Conv2D)	(None, 116, 116, 6)	456
max_pooling2d (MaxPooling2D)	(None, 58, 58, 6)	0
conv2d_1 (Conv2D)	(None, 54, 54, 16)	2416
max_pooling2d_1 (MaxPooling2D)	(None, 27, 27, 16)	0
dropout (Dropout)	(None, 27, 27, 16)	0
flatten (Flatten)	(None, 11664)	0
dense (Dense)	(None, 120)	1399800
dense_1 (Dense)	(None, 84)	10164
dense_2 (Dense)	(None, 81)	6885
<hr/>		
Total params:	1,419,721	
Trainable params:	1,419,721	
Non-trainable params:	0	

Layer (type)	Output Shape	Param #
<hr/>		
input_2 (InputLayer)	[(None, 120, 120, 3)]	0
block1_conv1 (Conv2D)	(None, 120, 120, 64)	1792
block1_conv2 (Conv2D)	(None, 120, 120, 64)	36928
block1_pool (MaxPooling2D)	(None, 60, 60, 64)	0
block2_conv1 (Conv2D)	(None, 60, 60, 128)	73856
block2_conv2 (Conv2D)	(None, 60, 60, 128)	147584
block2_pool (MaxPooling2D)	(None, 30, 30, 128)	0
block3_conv1 (Conv2D)	(None, 30, 30, 256)	295168
block3_conv2 (Conv2D)	(None, 30, 30, 256)	590080
block3_conv3 (Conv2D)	(None, 30, 30, 256)	590080
block3_pool (MaxPooling2D)	(None, 15, 15, 256)	0
...		
Total params:	15,323,025	
Trainable params:	608,337	
Non-trainable params:	14,714,688	

LeNet-5

with Dropout = 0.25

VGG16 (ImageNet)

with Dropout = 0.50

การนำใช้

NihonGO !
Handwriting kanji recognition system.

Select input options
Sketchpad input

Dev by : Tanaanan Chalermpan
Computer Science Kasetsart U. Thailand

JLPTN5 handwriting Kanji recognition.

Stroke width: 8

Predicted result.
Class : 時
with probability : 100.00 %

Strokes : 10
Meaning : Time
Onyomi: 江 (ji)
Kunyomi: とき (toki)
for more detail : [Shirabe jisho](#)



18

MEMBER AND ROLES



6610402230 ศิริสุข ทานธรรม

- model fine-tuning VGG16
- code checker and optimization for data loader, LeNet5 Greyscale Augment CNN
- instruction and theorem documentation



6610402078 รุนันท์ เวสิมพันธ์

- model - training, testing
- data collection, augmentation, cleaning
- data visualization, pre-processing



6610402141 ปภิกาทร อุคำ

- making model from LeNet-5 concept
- Documents checker
- Prompt Engineering Expert
- Optimizer Data (large file can't push to git)



LAST BOSS

**Asst. Prof. Dr.
Chakrit Watcharopas**

อาจารย์ประจำวิชา
make us cry