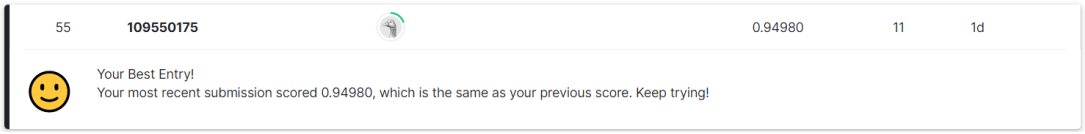
109550175 HW5 report

Environment Details :

python version = 3.9.15
tensorflow version = 2.11.0
keras version = 2.11.0
keras-preprocessing version = 1.1.2

Result Screenshots :



Hyper-parameter :

For task1 , I use batch_size = 100 , epochs = 16 , size of the input images is (100 , 100 ,3) ;
For task2 , I use batch_size = 100 , epochs = 40 , size of the input images is (100 , 100 , 3) ;
For task3 , I use batch_size = 400 , epochs = 20 , size of the input images is (72 , 96 , 3) ; (avoid to stretch the original photo to squared-like , which may be hard for NN to learn to split and identify 4 digits)

Image Augmentation :

Use the package provided in keras to generate augmented images , and the possible augmented conditions is in the ImageDataGenerator() block .
For task1 to 3 , I used 5 , 8 and 20 augmented images in exchange for an original image.

```
img = np.expand_dims(img , 0)
gen = ImageDataGenerator(
rotation_range = 10 ,
width_shift_range = 0.15 ,
height_shift_range = 0.15 ,
shear_range = 0.15 ,
zoom_range = 0.1 ,
channel_shift_range = 10 ,
)
aug_iter = gen.flow(img)
aug_images = [next(aug_iter)[0].astype(np.uint8) for i in range(20)]
```

Implementation Details :

這是一個單/多輸出的 NN 架構 ， 每個模型最後的每個 dense 輸出層都是一個有著 36 種 output 的預測者 ， 第 i 個 dense layer(which is named “digit i” in the Layer(type) column)期望用來分辨出畫面上由左至右第 i 個 digit 是 0-9 , A-Z 中哪一種。

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 72, 96, 3)]	0	[]
conv2d_7 (Conv2D)	(None, 72, 96, 32)	896	['input_2[0][0]']
conv2d_8 (Conv2D)	(None, 70, 94, 32)	9248	['conv2d_7[0][0]']
batch_normalization_4 (BatchNo rmalization)	(None, 70, 94, 32)	128	['conv2d_8[0][0]']
max_pooling2d_4 (MaxPooling2D)	(None, 35, 47, 32)	0	['batch_normalization_4[0][0]']
dropout_4 (Dropout)	(None, 35, 47, 32)	0	['max_pooling2d_4[0][0]']
conv2d_9 (Conv2D)	(None, 35, 47, 64)	18496	['dropout_4[0][0]']
conv2d_10 (Conv2D)	(None, 33, 45, 64)	36928	['conv2d_9[0][0]']
batch_normalization_5 (BatchNo rmalization)	(None, 33, 45, 64)	256	['conv2d_10[0][0]']
max_pooling2d_5 (MaxPooling2D)	(None, 16, 22, 64)	0	['batch_normalization_5[0][0]']
dropout_5 (Dropout)	(None, 16, 22, 64)	0	['max_pooling2d_5[0][0]']
conv2d_11 (Conv2D)	(None, 16, 22, 128)	73856	['dropout_5[0][0]']
conv2d_12 (Conv2D)	(None, 14, 20, 128)	147584	['conv2d_11[0][0]']
batch_normalization_6 (BatchNo rmalization)	(None, 14, 20, 128)	512	['conv2d_12[0][0]']
max_pooling2d_6 (MaxPooling2D)	(None, 7, 10, 128)	0	['batch_normalization_6[0][0]']
dropout_6 (Dropout)	(None, 7, 10, 128)	0	['max_pooling2d_6[0][0]']
conv2d_13 (Conv2D)	(None, 5, 8, 256)	295168	['dropout_6[0][0]']
batch_normalization_7 (BatchNo rmalization)	(None, 5, 8, 256)	1024	['conv2d_13[0][0]']
max_pooling2d_7 (MaxPooling2D)	(None, 2, 4, 256)	0	['batch_normalization_7[0][0]']
flatten_1 (Flatten)	(None, 2048)	0	['max_pooling2d_7[0][0]']
dropout_7 (Dropout)	(None, 2048)	0	['flatten_1[0][0]']
digit1 (Dense)	(None, 36)	73764	['dropout_7[0][0]']
digit2 (Dense)	(None, 36)	73764	['dropout_7[0][0]']
digit3 (Dense)	(None, 36)	73764	['dropout_7[0][0]']
digit4 (Dense)	(None, 36)	73764	['dropout_7[0][0]']

(若最後 dense layers 層有 digit1 , digit2 , digit3 跟 digit4 , 代表此為 task3 的 model 架構, 同理如果最後顯示的是 digit1 , digit2 , 代表的就是 task2 的 model 架構)

這是我選用的 loss 計算方式 , optimizer 以及 metrics

```
model.compile(loss='categorical_crossentropy', optimizer='Adamax', metrics=['accuracy'])
```

Model weight links :

Task 1 :
https://drive.google.com/file/d/1tUfxdpjKoreNmxlOXTdhZtRJWeoQjKLD/view?usp=sharing

Task 2 :
https://drive.google.com/file/d/1Waj7yD5DOiGd8xm91Ij2U5VTlUSZThYC/view?usp=sharing

Task 3 :
https://drive.google.com/file/d/1sxl4pQKsfj7B6RBdKKefAV3VhQr4qWq5/view?usp=sharing

In case that the links have some trouble , I also submitted a folder named model_weight , and the weight is the same as the one provided in links. TA can directly run the inference.py file and get the same prediction , and there's no need to download the weight from above links(you can still do so if wanted , but remember to read the requirement.txt to change the path to your downloaded weight in inference.py , thanks)