

>>>>



>>>>



IIE FINAL PROJECT

基於深度學習網路於手語辨識

指導教授：邱銘傳 | 110034543 戴佩怡





CONTENT



01

研究目的

- 研究背景
- 5W1H
- 資料介紹
- 研究流程



02

研究方法

- 方法選擇
- WGAN-GP
- VGG19



03

實驗過程

- 資料處理
- Data Augmentation
- 訓練結果



04

參數優化

- 實驗設計
- OA L_9
- Minitab結果



05

結論

- 研究結果&未來展望

>>>>

>>>>

—

○

〰️

•

〰️

•

+

—

01

研究目的



>>>>

研究背景

>>>>

雖然手語在主流社會不普及，但是對聾人族群來說是一個主要語言，也是一個凝聚這個族群的重要因素。

手語就像口語一樣，會因地域關係而有所不同，大則是國家差異，小則有城市之分。

>>>>

手語「方言」很麻煩，作為一種替代殘缺的語言，不同學校和不同地方的打法都不一樣，學生理解起來困難，很多志工去外地都無法和當地聾啞人溝通。



>>>>

5W1H



>>>>

Why

欲了解深度學習運用於影像辨識之原理並理解如何運用於手語辨識領域

When

欲了解手語表達族群在表達的內容時

Who

需要使用手語表達與溝通的族群

Where

聾啞人士需要表達溝通的任何地方

What

手語準確辨識

How

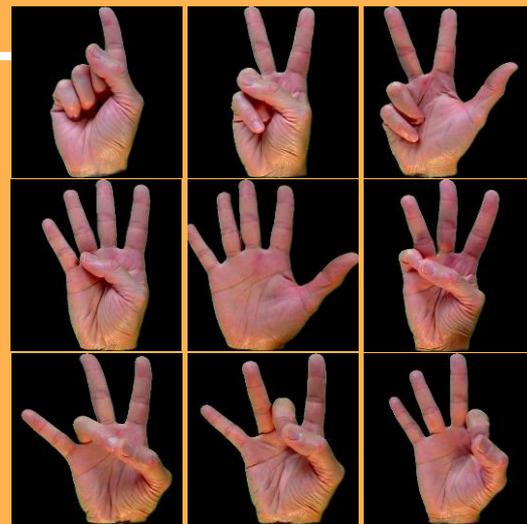
運用WGAN-GP來增加分類模型的訓練以達成準確率的提升



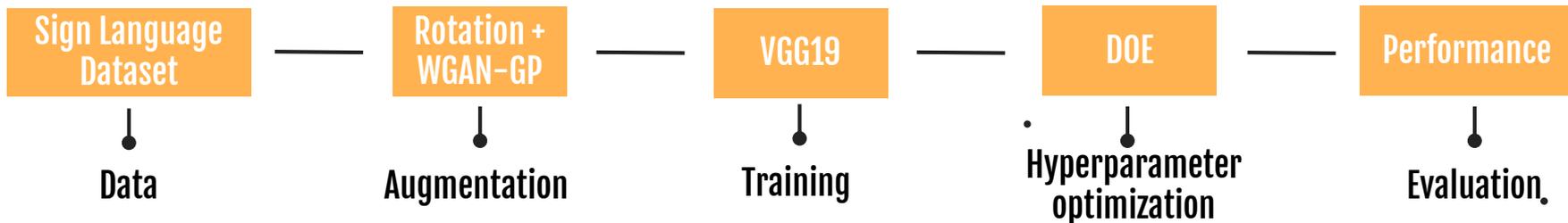
資料介紹



- 來自於Kaggle的American Sign Language Dataset
- 本資料集共有2520張數字0~9及字母A~Z之圖片
- 此報告採取其中的數字1~9類別進行實驗。



研究流程



>>>>

>>>>

—

○

〰️

•

+

—

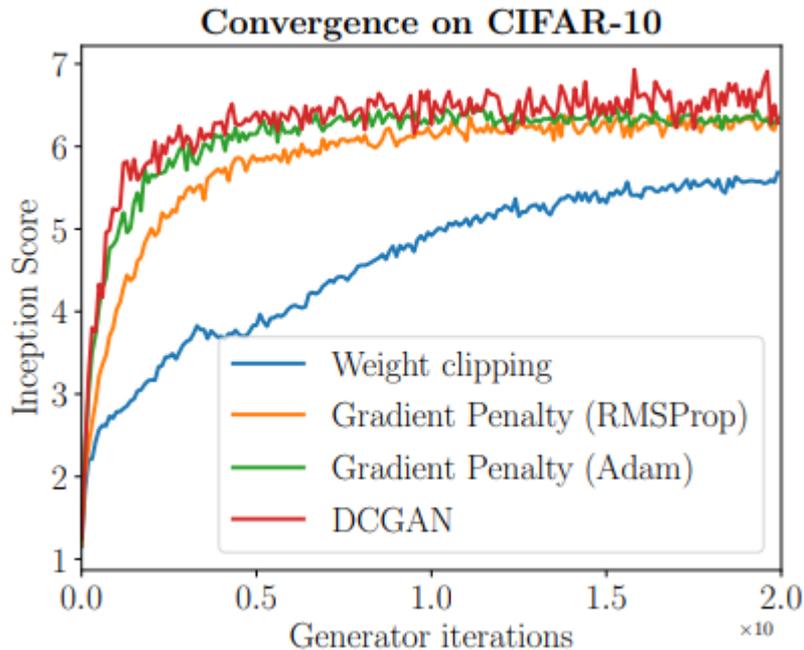
○

•

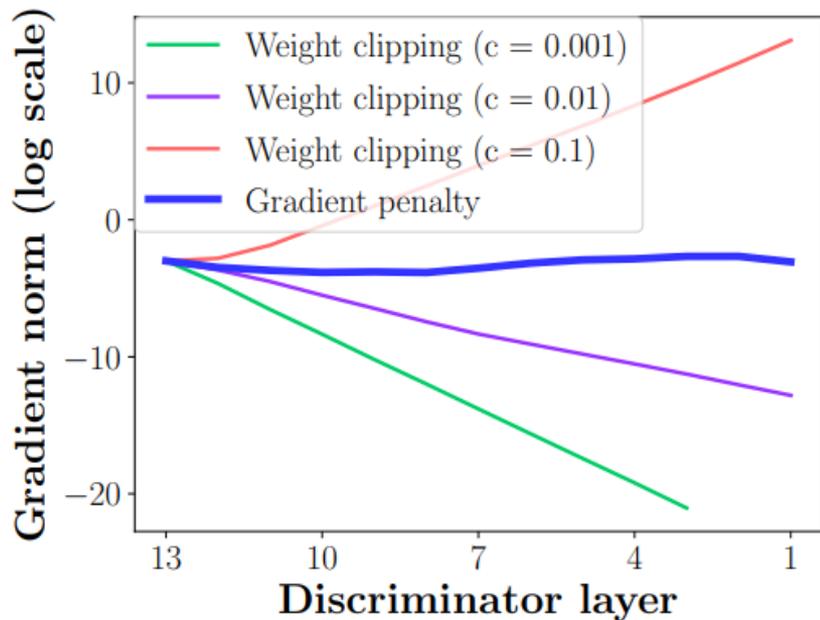
02

方法介紹



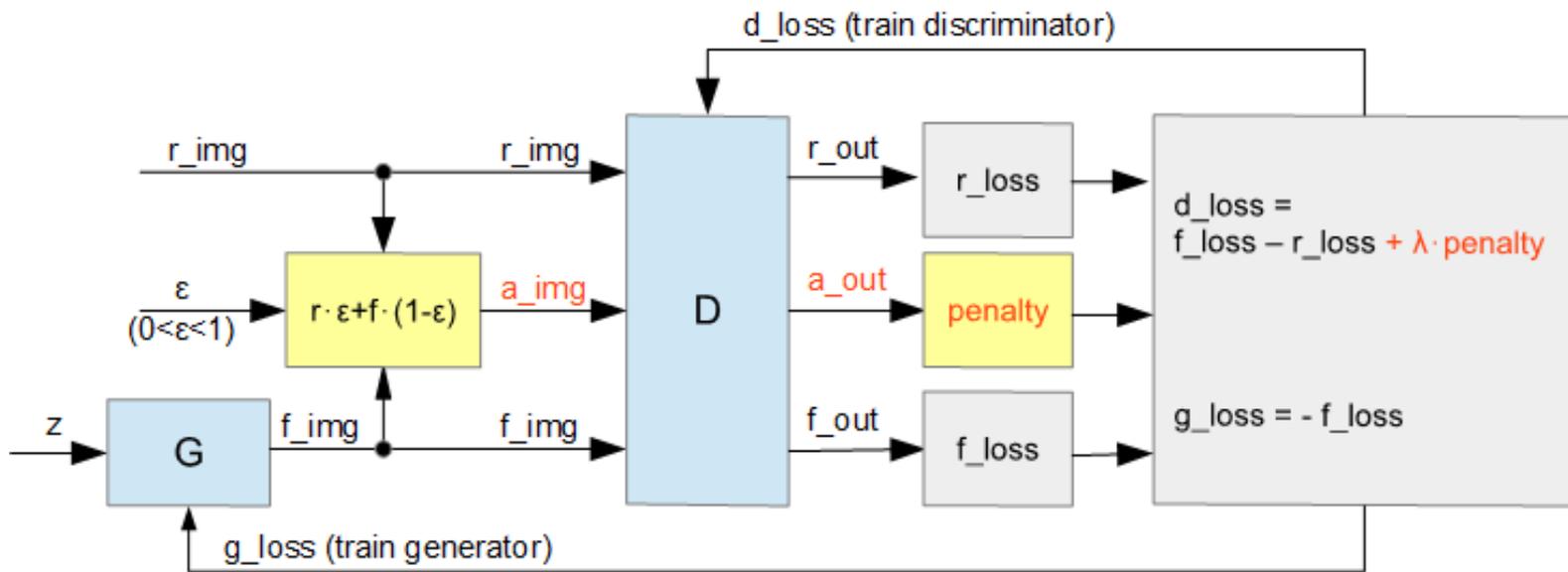


- 在同樣實驗設置下WGAN-GP的結果比WGAN效果好，跟DCGAN差不多，但是訓練要比DCGAN穩定。
- 雖然訓練時間比DCGAN要來的長，但因為訓練穩定所以生成的圖片質量要比DCGAN好。
- 所以本次研究採WGAN-GP來進行實驗。



- weight clipping會容易導致梯度消失或者梯度爆炸。
- 原因是判别器是一个多層網路，如果把clipping threshold設得稍微小了一點，每經過一層網路，梯度就變小一點點，多層之後就会指數衰減；反之，如果設得稍微大了一點，每經過一層網路，梯度變大一點點，多層之後就會指數爆炸。只有設得不大不小，才能讓生成器獲得恰到好處的回傳梯度。

WGAN-GP(2/2)



r_{img} : real image
 f_{img} : fake image
 a_{img} : averaged image

VGG19(1/2)

- VGG19模型是使用了大量資料訓練出來的模型，且使用非常多層的處理，透過遷移學習的方式可以直接將以訓練好的模型拿來使用。
- 較深的層數能提高效能

Model: "vgg19"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 48, 48, 3)]	0
block1_conv1 (Conv2D)	(None, 48, 48, 64)	1792
block1_conv2 (Conv2D)	(None, 48, 48, 64)	36928
block1_pool (MaxPooling2D)	(None, 24, 24, 64)	0
block2_conv1 (Conv2D)	(None, 24, 24, 128)	73856
block2_conv2 (Conv2D)	(None, 24, 24, 128)	147584
block2_pool (MaxPooling2D)	(None, 12, 12, 128)	0
block3_conv1 (Conv2D)	(None, 12, 12, 256)	295168
block3_conv2 (Conv2D)	(None, 12, 12, 256)	590080
block3_conv3 (Conv2D)	(None, 12, 12, 256)	590080
block3_conv4 (Conv2D)	(None, 12, 12, 256)	590080
block3_pool (MaxPooling2D)	(None, 6, 6, 256)	0
block4_conv1 (Conv2D)	(None, 6, 6, 512)	1180160
block4_conv2 (Conv2D)	(None, 6, 6, 512)	2359808
block4_conv3 (Conv2D)	(None, 6, 6, 512)	2359808
block4_conv4 (Conv2D)	(None, 6, 6, 512)	2359808
block4_pool (MaxPooling2D)	(None, 3, 3, 512)	0
block5_conv1 (Conv2D)	(None, 3, 3, 512)	2359808
block5_conv2 (Conv2D)	(None, 3, 3, 512)	2359808
block5_conv3 (Conv2D)	(None, 3, 3, 512)	2359808
block5_conv4 (Conv2D)	(None, 3, 3, 512)	2359808
block5_pool (MaxPooling2D)	(None, 1, 1, 512)	0

Model: "sequential"

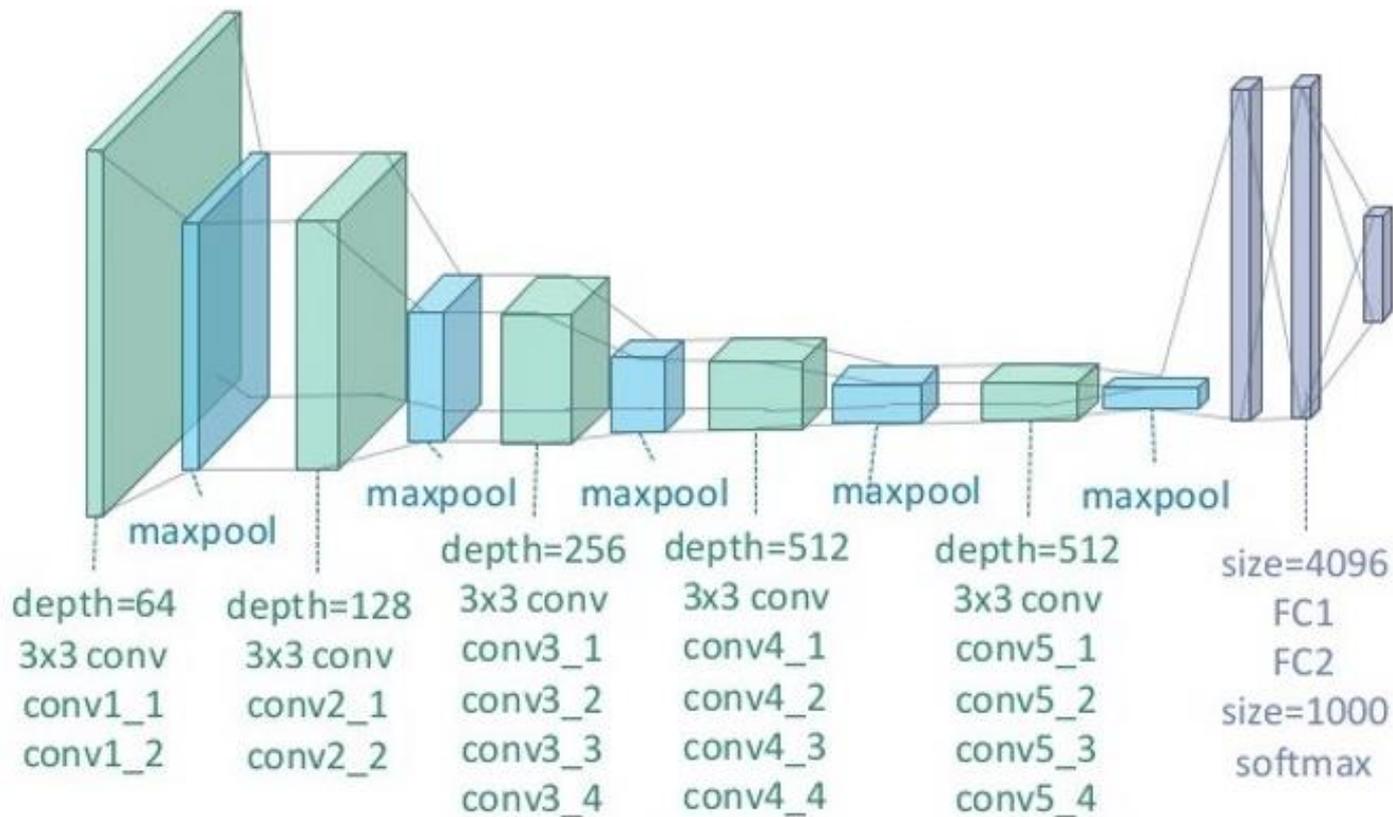
Layer (type)	Output Shape	Param #
vgg19 (Functional)	(None, 1, 1, 512)	20024384
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 1024)	525312
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 9)	9225

Total params: 20,558,921
Trainable params: 20,558,921
Non-trainable params: 0

```
lrd = ReduceLRonPlateau(monitor = 'val_loss',
                        patience = 20,
                        verbose = 1,
                        factor = 0.50,
                        min_lr = 1e-10)
```

可以在訓練過程中優化學習率
閾值最下限為1e-10

VGG19(2/2)

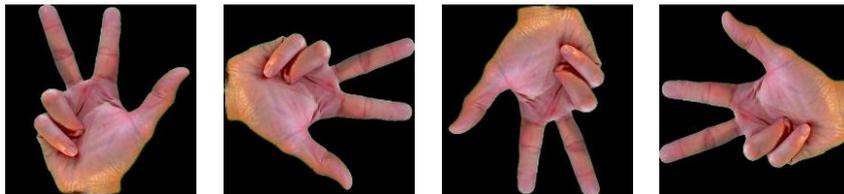


| 旋轉增強

```
for file in files:
    img = cv2.imread(os.path.join(file_path, file))
    name = file.split(".")
    name_90 = name[0]+"_90.jpg"
    name_180 = name[0]+"_180.jpg"
    name_270 = name[0]+"_270.jpg"

    img90 = np.rot90(img, 1)
    img180 = np.rot90(img, 2)
    img270 = np.rot90(img, 3)

    cv2.imwrite(path+"/"+classs+"/"+name_90, img90)
    cv2.imwrite(path+"/"+classs+"/"+name_180, img180)
    cv2.imwrite(path+"/"+classs+"/"+name_270, img270)
```



1 class

70



280

total

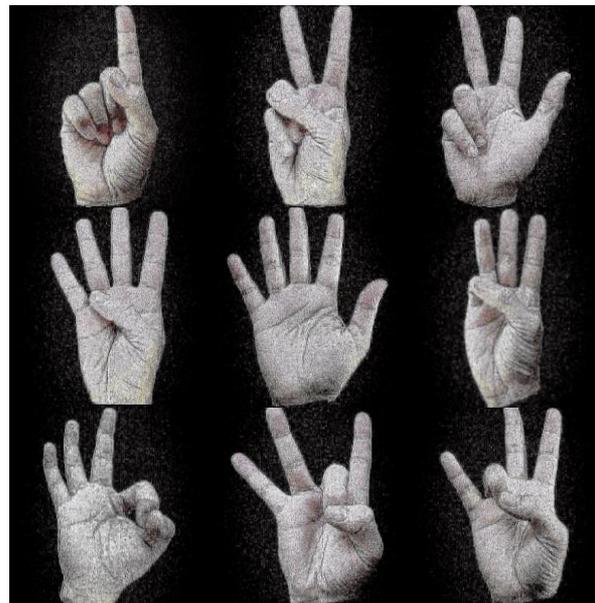
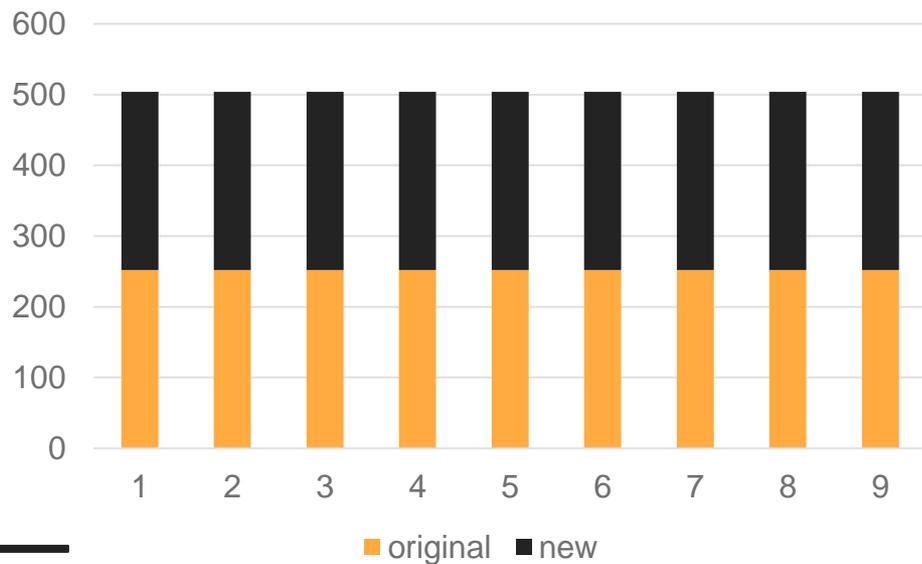
630



2520

| WGAN-GP增強

WGAN-GP



資料處裡

WGAN-GP	Train set	Val set	Test set
No	252	28	28
Yes	504	28	28



比例為8 : 1 : 1

Test set 資料來源為自己收集之資料
取14人所比之1~9數字手語，各類別每
個人拍兩張，總共收集到28筆資料。

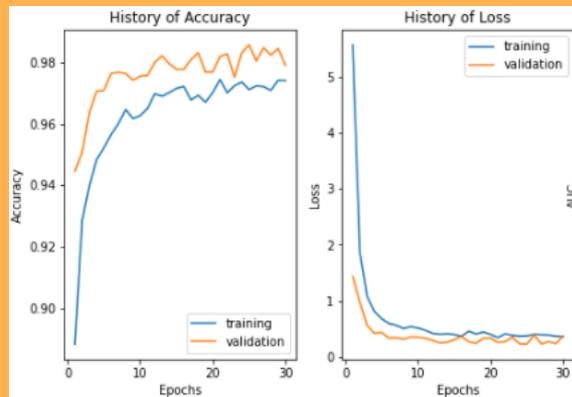
訓練結果

>>>>

>>>>

無資料增強

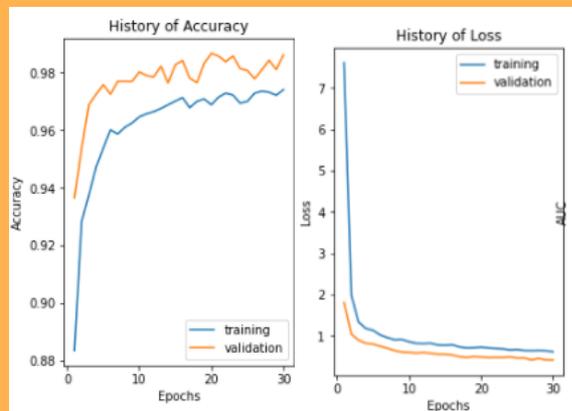
參數	設定
Optimizer	adam
Batch size	32
Dropout	0.5
Epoch	30



	Accuracy
Train	0.9741
Val	0.9791
Test	0.8466

資料增強後

參數	設定
Optimizer	adam
Batch size	32
Dropout	0.5
Epoch	30



	Accuracy
Train	0.9844
Val	0.9862
Test	0.8658

04

參數優化





實驗設計



Factor	Level1	Level2	Level3
Optimizer	Adam	AdaDelta	Adagrad
Dropout	0.4	0.5	0.6
Batch Size	16	32	64

- » 利用實驗設計的田口方法，使用直交表以較少的實驗來獲得更可靠的因子效果估計。

>>>>

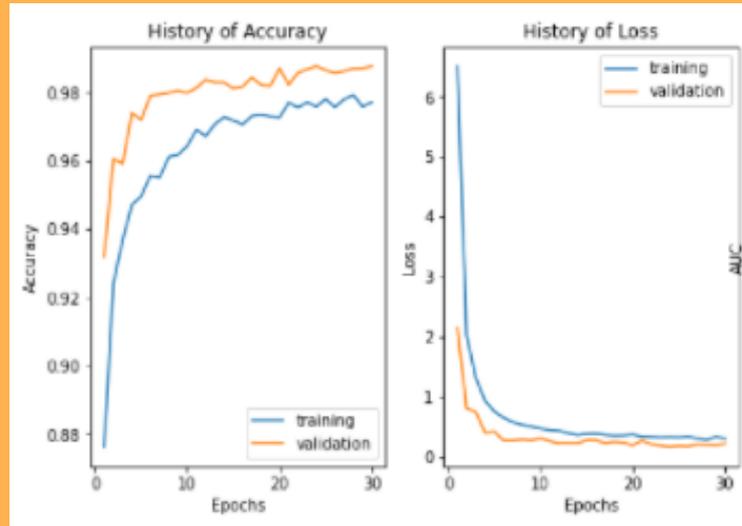
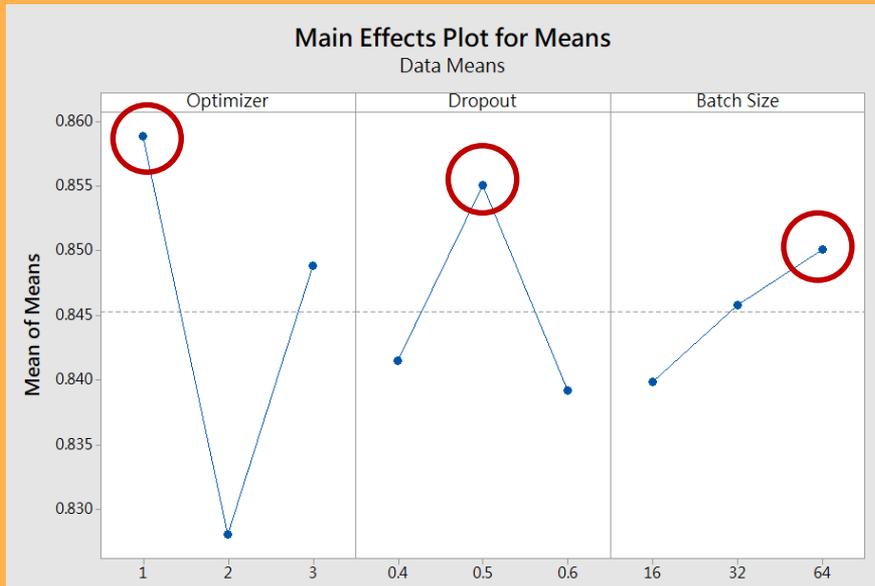
Orthogonal Array L_9

實驗編號	Optimizer	Dropout	Batch Size	Accuracy
1	Adam	0.4	16	0.8532
2	Adam	0.5	32	0.8658
3	Adam	0.6	64	0.8576
4	AdaDelta	0.4	32	0.8247
5	AdaDelta	0.5	64	0.8463
6	AdaDelta	0.6	16	0.8132
7	Adagrad	0.4	64	0.8465
8	Adagrad	0.5	16	0.8532
9	Adagrad	0.6	32	0.8469

>>



Minitab結果



Response Table for Means

Level	Optimizer	Dropout	Batch Size
1	0.8589	0.8415	0.8399
2	0.8281	0.8551	0.8458
3	0.8489	0.8392	0.8501
Delta	0.0308	0.0159	0.0103
Rank	1	2	3

	Accuracy
Train	0.9846
Val	0.9873
Test	0.8689

05 結論



>>>>

研究結果& 未來展望

>>>>

01

透過WGAN-GP數據增強後的資料，進行VGG19訓練，結果提高手語分類的準確率。

02

利用自行收集的Test資料集，驗證所訓練模型的泛化能力。

03

除了手語的辨識外，未來可將手語辨識結合翻譯機功能，讓不會手語的族群也能方便快捷的與手與族群順利溝通。

參考文獻

Reference

- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*.
- Gao, X., Deng, F., & Yue, X. (2020). Data augmentation in fault diagnosis based on the Wasserstein generative adversarial network with gradient penalty. *Neurocomputing*, 396, 487-494.
- <https://cloud.tencent.com/developer/article/1645877>
- <http://www.twistedwg.com/2018/02/02/WGAN-GP.html>
- <https://zhuanlan.zhihu.com/p/52799555>

>>>>

>>>>

—

○

THANKS

Does anyone have any questions?

