Report

Team 7　105060003 王文依　105060007 鍾佳秀　106061218 李丞恩

---

1. Methodology

a. Classifier

我們用 keras application 裡提供的 model 來實作[1]。再試過 CNN（自己加 layer）和其他可以套的架構（如 mobilenet、densenet）之後，我們選擇使用做出來效果最好的 model 是 Resnet50。其引用如以下兩行 code 所示：

```
from tensorflow.keras.applications.resnet_v2 import ResNet50V2
model = ResNet50V2(input_shape=(64,64,1), weights=None, classes=30）
```

此後，再幫 layer 加上 L2 Regularization 減輕一些 overfitting。

b. Generator

我們在 github 上找到別人寫的 catGAN，但是效果不佳，所以把 generator 的部分改成他 tutorial 有提到的 DCGAN，修改結合成我們最後的 generator。

2. How to train our model

a. parameter setting

在實驗的過程中，我們發現 batch size 對正確率影響不大，此外迭代次數越多，validation 的正確率反而不斷往下降。因此我們設定 learing rate = 0.001、batch_size = 50、epochs = 200。此外，較大的 learing rate 也會導致模型很快 overfitting。

b. optimizer

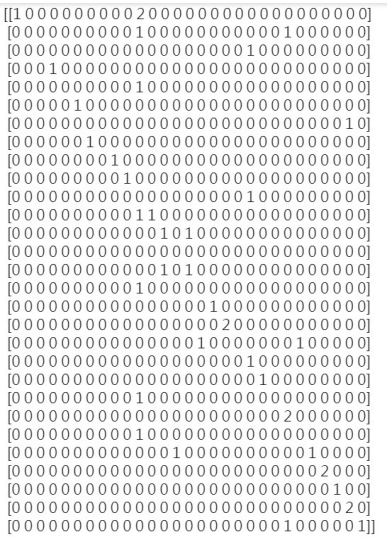
最佳之參數為 Adam(0.0002, 0.5)。

c. evaluation

我們將 img_per_class 設為 13000，training data 是從每種 class 挑 13000 筆，所以最後會有 13000*30=300000 筆 sample 作為 training dataset。因為受限於 google colab 提供的 RAM 限制，所以無法拿取更多來訓練。而用在 model。fit 中的 validation data 是從 valid.csv 中挑 13000 筆 data。30*30 之 confusion matrix 則如下圖所示：

---

[1] https://keras.io/applications/#available-models



d. others

因為 classifier trian 出來的結果都沒有很好，也一直提升不了 accuracy，所以最後想到了曾經在機器學習學過 ensemble learing，把很多個表現沒有很好的 model，最後利用 Voting model 來實作。

我們總共結合 4 個 model，model 原來的 testing accuracy 分別是：model 1：48.78%、model 2：51.22%、model 3：41.46%、model 4：46.34%。最後做出來的 enemble model 的 accuracy 是 53.66%，可以看得出來的確有提升一些。

我們原本直接使用 catGAN 實作 Generator，但效果不佳，因此修改了網路上 32*32 DCGAN 的 generator 成 64*64 的版本，最後效果不錯，迭代十次以內就能看到一些圖形的輪廓。

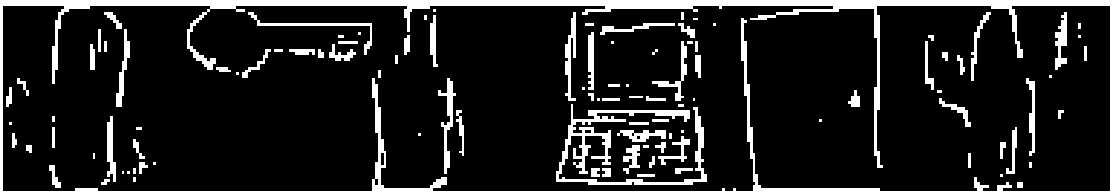
3. Test Result

a. Classifier：

	drawing	word
0	[[[51, 43,	key
1	[[[115, 11	giraffe
2	[[[67, 59,	light_bulb
3	[[[11, 8, 1	bed
4	[[[0, 21, 4	roller_coaster
5	[[[8, 6, 11	door
6	[[[95, 74,	light_bulb
7	[[[7, 8, 5,	laptop
8	[[[62, 63,	paintbrush
9	[[[97, 60,	key
10	[[[52, 98,	giraffe
11	[[[62, 64,	cake
12	[[[55, 43,	popsicle
13	[[[174, 13	giraffe
14	[[[21, 20,	whale
15	[[[231, 32	spoon
16	[[[32, 23,	hand
17	[[[3, 0, 1,	toaster
18	[[[4, 7, 7,	toaster
19	[[[4, 1, 0,	banana
20	[[[93, 78,	giraffe
21	[[[6, 5, 8,	spoon
22	[[[102, 11	giraffe
23	[[[2, 10, 3	popsicle
24	[[[2, 1, 9,	spoon
25	[[[6, 27, 8	spoon
26	[[[60, 66,	rain
27	[[[0, 24, 6	giraffe
28	[[[225, 24	whale
29	[[[188, 19	giraffe
30	[[[22, 14,	rain
31	[[[49, 26,	whale
32	[[[37, 4, 0	rain
33	[[[19, 74,	train
34	[[[3, 8, 52	The_Great_Wall_of_China
35	[[[69, 59,	fork
36	[[[94, 105	cactus
37	[[[138, 89	giraffe
38	[[[208, 18	marker
39	[[[28, 26,	wine_bottle
40	[[[63, 50,	swan

# of instances classified correctly：22 / 41
Accuracy：53.66%

b. Generator：



4. Demo Result

Classifier： 23；Generator：92、Subjective： 7。

5. Discussion

首先，因為 googl colab 提供的 RAM 大小不夠，所以 img_per_class 我們試過最高只能到 13000。另外，如果長時間 training 的話，google colab 還會暫時停止提供 GPU，大概要過個 3、4 小時才能再用。

另外我們一開始自己寫的 CNN layer 也有嚴重 Overfitting 的問題。在少數幾個 epoch 後，testing accuracy 跟 validation accuracy 會相差越來越大，常常發生訓練集正確率達 80%，但 validation accuracy 卻是 35%，且隨著每次迭代，validation accuracy 還會往下掉。所以最後才換成 Resnet50，雖然還是一樣有 overfitting 的問題，所以我們也有嘗試再加上 Batch Normalization 跟 L2 Regularization，但也只有稍微變好一點(約 10%)。因此我們決定在 overfitting 發生前就終止訓練，並儲存得到的模型，以供後續 ensemble learning 時使用。如果硬體設備或 RAM 能改善，一次讀取大量數據集的話，或許能解決這個問題。

至於 Generator，我們發現 discriminator loss 十分大，不過好消息是，對簡單的圖形而言，大約訓練 20～50 個 epoch，即可見到清楚的輪廓。對於較複雜的圖形而言，可能要 train 到 200 多個 epoch，才能依常理判斷所畫為何物。

6. Conclusion

在這次專題中我們發現，實作深度學習網路時應擁有足以支撐訓練的硬體設備，以利 training set 的讀取。此外，應多嘗試不同的模型，並思考是否能利用 ensemble learning 整合不同模型，使效能提升。同時若能解決 overfitting 的問題，則當提升訓練時的迭代次數。