**Report**

Team 7 105060003 王文依 105060007 鍾佳秀 106061218 李丞恩

1. **Methodology**
   1. Classifier

　　我們用keras application裡提供的model來實作[[1]](#footnote-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。

* 1. 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。

1. **How to train our model**
   1. parameter setting

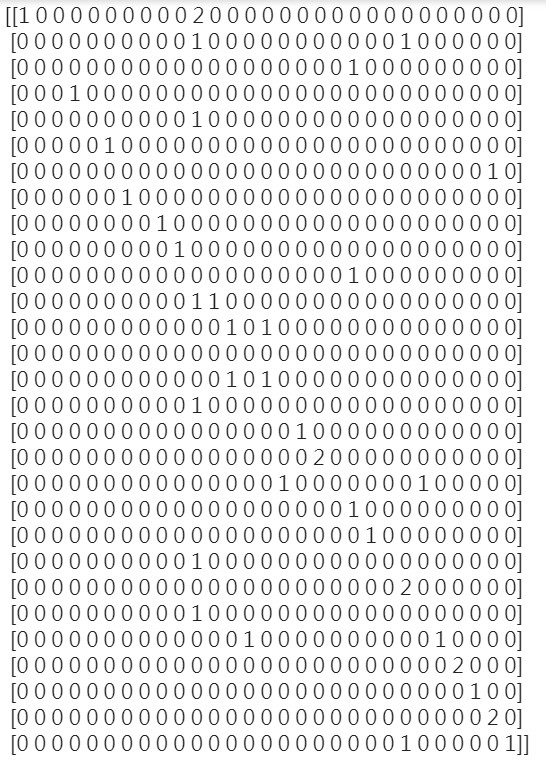
　　在實驗的過程中，我們發現batch size對正確率影響不大，此外迭代次數越多，validation的正確率反而不斷往下降。因此我們設定learing rate = 0.001、batch\_size = 50、epochs = 200。此外，較大的learing rate也會導致模型很快overfitting。

* 1. optimizer

　　最佳之參數為Adam(0.0002, 0.5) 。

* 1. 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：



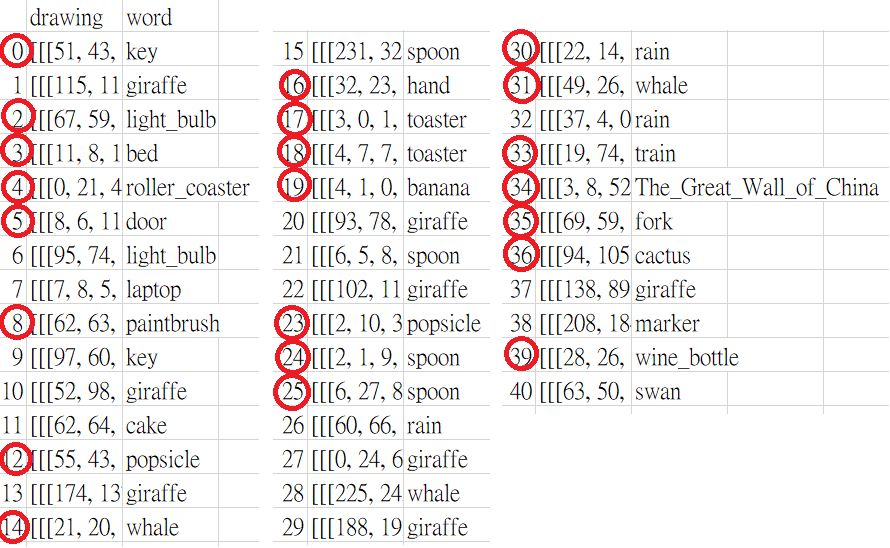
* 1. others

　　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。

1. **Test Result**
2. Classifier：

****

# of instances classified correctly：22 / 41

Accuracy：53.66%

1. Generator：

****

1. **Demo Result**

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

1. **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。

1. **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。

1. <https://keras.io/applications/#available-models> [↑](#footnote-ref-1)