

**國立臺灣大學資訊管理學系研究所**  
**資料探勘**  
**小組作業二**

**基於MLP、CNN、AlexNet神經網路  
之CIFAR-10圖像辨識準確度比較**

**第二組**

**蘇達立、蕭溥辰、王辰豪**

**中華民國 109 年 05 月 05 日**

# 目錄

目錄	1
壹、作業概述	2
貳、研究動機與目的	2
參、資料集	2
肆、實驗方法	3
伍、實驗結果與分析	5
陸、結論	8
柒、附錄	9

## 壹、作業概述

在本次作業中我們採用三種分類方法，分別是MLP(一般Neural Network)、CNN、AlexNet，對CIFAR-10 datasets做圖像辨識，比較在相同datasets下，套用三種神經網路後的Accuracy、Precision、Recall和F1-score。

## 貳、研究動機與目的

基於課堂上Neural Network分類圖像的理論所學，我們想針對該理論進程式碼的實作，了解如何將圖像datasets放入Neural Network model進行訓練和測試，最後根據實作結果知道各種Neural Network model的分類準確率之優劣。因為CIFAR-10在分類問題上，是相當知名的問題，且具有充足和乾淨的資料集，因此，在本次作業中，我們以CIFAR-10圖像辨識來當作我們本次作業的題目。

## 參、資料集

在本次研究中，我們是採用CIFAR-10 dataset作為實驗的資料集，如圖一所示，其包含60000筆32x32x3的RGB照片，總共有10個class，每個class都有6000筆照片，我們將資料集其中40000筆照片來做training，10000筆來做validation，10000筆來做testing。

airplane



automobile



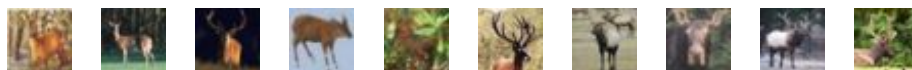
bird



cat



deer

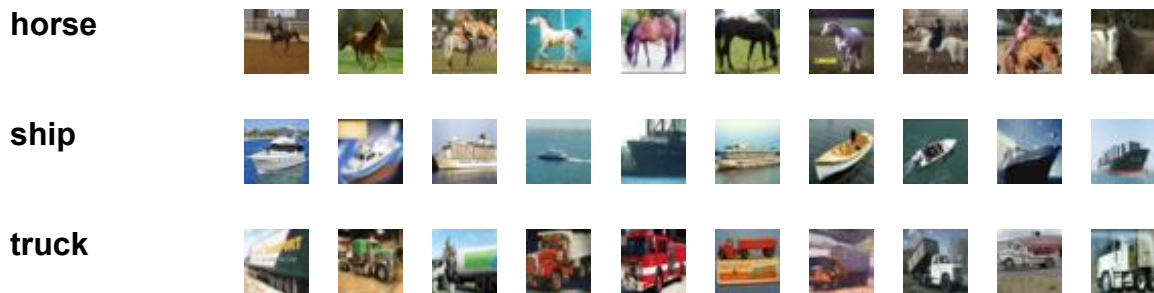


dog



frog





(圖一) CIFAR-10 Datasets

## 肆、實驗方法

我們在將datasets放入model訓練前，採用兩個preprocessing方式，第一是進行one-hot encoding，我們將每筆資料的label轉換成長度為10的List來表示，以1、0表示原始label值，讓程式可以更好的去理解及運算。第二是調整圖片的位元值，將每張圖片的位元數除以255，讓原本介於0到255的位元數，調整至介於0到1之間，方便model進行training和testing。

當資料preprocessing完後，我們先將40000筆data放入三種不同的model(MLP、CNN、AlexNet)進行training，三種model的神經網路架構如圖二到圖四所示，並用10000筆data進行驗證，最後再用10000筆data測試三種model的辨識準確率。

Model: "sequential\_3"

Layer (type)	Output Shape	Param #
flatten_3 (Flatten)	(None, 3072)	0
dense_7 (Dense)	(None, 1024)	3146752
dense_8 (Dense)	(None, 10)	10250
Total params: 3,157,002		
Trainable params: 3,157,002		
Non-trainable params: 0		

(圖二) MLP 神經網路架構

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 32, 32)	896
dropout_1 (Dropout)	(None, 32, 32, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	18496
dropout_2 (Dropout)	(None, 16, 16, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 64)	0
flatten_1 (Flatten)	(None, 4096)	0
dropout_3 (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 1024)	4195328
dropout_4 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 10)	10250
Total params: 4,224,970		
Trainable params: 4,224,970		
Non-trainable params: 0		

(圖三) CNN 神經網路架構

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 32, 32, 3)	0
conv2d_1 (Conv2D)	(None, 8, 8, 96)	34944
max_pooling2d_1 (MaxPooling2D)	(None, 4, 4, 96)	0
conv2d_2 (Conv2D)	(None, 4, 4, 256)	614656
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 256)	0
conv2d_3 (Conv2D)	(None, 2, 2, 384)	885120
conv2d_4 (Conv2D)	(None, 2, 2, 384)	1327488
conv2d_5 (Conv2D)	(None, 2, 2, 256)	884992
max_pooling2d_3 (MaxPooling2D)	(None, 1, 1, 256)	0
flatten_1 (Flatten)	(None, 256)	0
dense_1 (Dense)	(None, 4096)	1052672
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 4096)	16781312
dropout_2 (Dropout)	(None, 4096)	0
dense_3 (Dense)	(None, 10)	40970
Total params: 21,622,154		
Trainable params: 21,622,154		
Non-trainable params: 0		

(圖四) AlexNet 神經網路架構

## 伍、實驗結果與分析

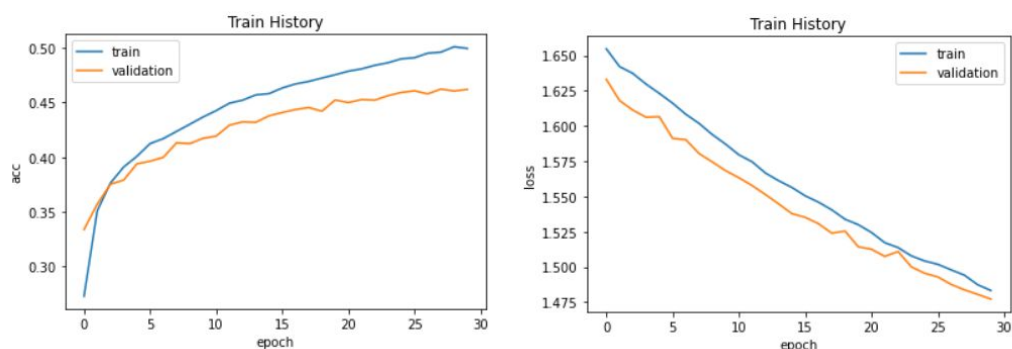
### (1) MLP:

MLP model在training時，如圖五所示，其accuracy隨著訓練的次數增加而穩定上升，最後達到約0.5，loss則是隨著訓練的次數增加而穩定下降，最後達到約1.485；而在validation時，其accuracy則是隨著訓練的次數增加而穩定上升，最後達到約0.45，loss則是隨著訓練的次數增加而下降，最後達到約1.48。根據Confusion matrix的結果，如圖六所示，MLP預測的結果在class 8的表現最佳，正確率為620/1000。

MLP model的testing結果，如表一所示，Accuracy平均為0.51，Precision平均為0.52，Recall平均為0.52，F1平均為0.51。

	Accuracy	Precision	Recall	F1
Average	0.51	0.52	0.52	0.51

(表一) MLP testing result



(圖五) MLP training result

predict	0	1	2	3	4	5	6	7	8	9
label										
0	577	35	36	26	24	23	26	35	149	69
1	45	577	11	30	14	32	26	26	72	167
2	115	32	315	104	100	89	130	51	32	32
3	43	33	73	376	34	183	108	41	33	76
4	81	26	135	75	337	67	146	78	29	26
5	33	26	76	238	49	349	88	68	37	36
6	16	22	67	124	80	65	546	30	22	28
7	52	36	50	90	72	82	45	464	22	87
8	120	76	9	35	15	21	11	10	620	83
9	52	166	7	42	11	23	30	34	68	567

(圖六) MLP Confusion matrix

## (2) CNN:

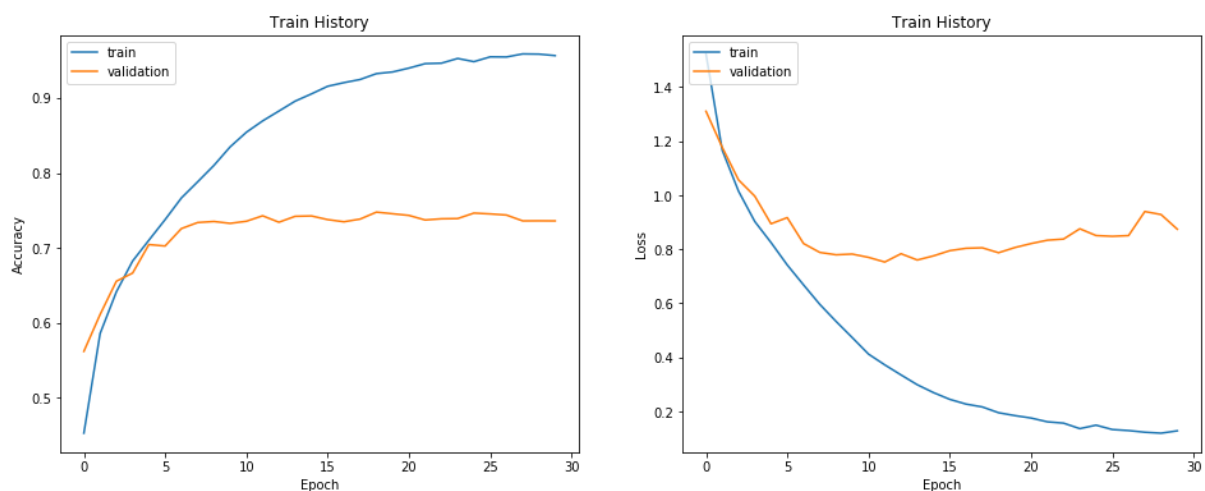
CNN model在training時，如圖七所示，其accuracy隨著訓練的次數增加而穩定上升，最後達到約0.95，loss則是隨著訓練的次數增加而穩定下降，最後達到約0.12；而在validation時，其accuracy在epoch 8以後達到0.73並趨於穩定，loss也是在epoch 8以後，便沒有顯著下降，穩定在0.78到0.85之間。根據Confusion matrix的結果，如圖八所示，CNN預測的結果在class 1的表現最佳，正確率為845/1000。

CNN model的testing結果，如表二所示，Accuracy平均為0.7350，Precision平均為0.74，Recall平均為0.74，F1平均為0.74。

	Accuracy	Precision	Recall	F1
Average	0.7350	0.74	0.74	0.74

(表二) CNN testing result





(圖七) CNN training result

predict	0	1	2	3	4	5	6	7	8	9
label										
0	786	14	52	18	22	8	4	17	48	31
1	19	845	6	12	3	5	5	4	21	80
2	57	3	664	51	97	46	39	31	4	8
3	28	6	78	565	55	158	58	38	4	10
4	21	3	69	65	697	38	31	68	7	1
5	11	2	55	197	48	602	21	56	4	4
6	5	6	58	67	42	25	782	11	1	3
7	11	3	24	42	57	42	3	812	1	5
8	59	35	19	19	12	8	11	6	806	25
9	32	79	14	14	3	7	8	30	22	791

(圖八) CNN Confusion matrix

### (3) AlexNet:

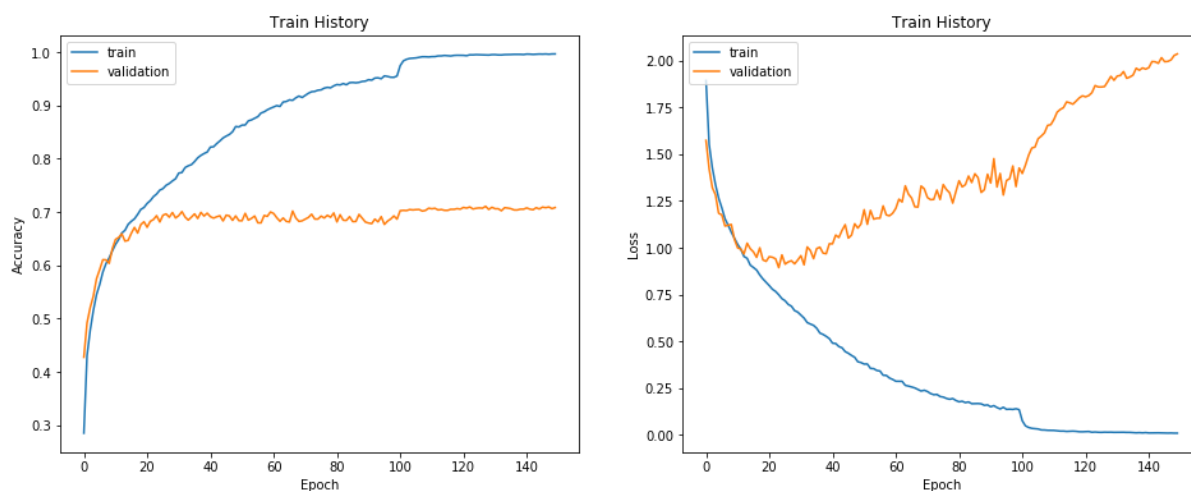
AlexNet model在training時，如圖九所示，其accuracy隨著訓練的次數增加而穩定上升，最後達到約0.99，loss則是隨著訓練的次數增加而穩定下降，最後達到約0.01；而在validation時，其accuracy在epoch 32達到0.7並趨於穩定，loss則是隨著訓練的次數增加先逐漸下降，在epoch 40以後逐漸上升，最後達到約2.03。根據Confusion matrix的結果，如圖十所示，AlexNet預測的結果在class 8的表現最佳，正確率為821/1000。

AlexNet model的testing結果，如表三所示，Accuracy平均為0.7006，Precision平均為0.7，Recall平均為0.7，F1平均為0.7。



	Accuracy	Precision	Recall	F1-score
Average	0.7006	0.7	0.7	0.7

(表三) AlexNet testing result



(圖九) AlexNet training result

predict	0	1	2	3	4	5	6	7	8	9
label										
0	814	18	36	7	14	6	10	15	54	26
1	32	775	10	8	2	8	8	5	26	126
2	81	9	592	71	66	54	69	35	12	11
3	29	19	52	500	47	187	78	48	17	23
4	40	5	92	66	567	47	74	87	9	13
5	19	5	50	185	49	561	39	70	12	10
6	9	8	41	56	30	34	799	8	7	8
7	23	1	16	41	40	58	8	793	5	15
8	75	25	9	12	9	13	2	3	821	31
9	51	69	10	16	7	8	11	17	27	784

(圖十) AlexNet Confusion matrix

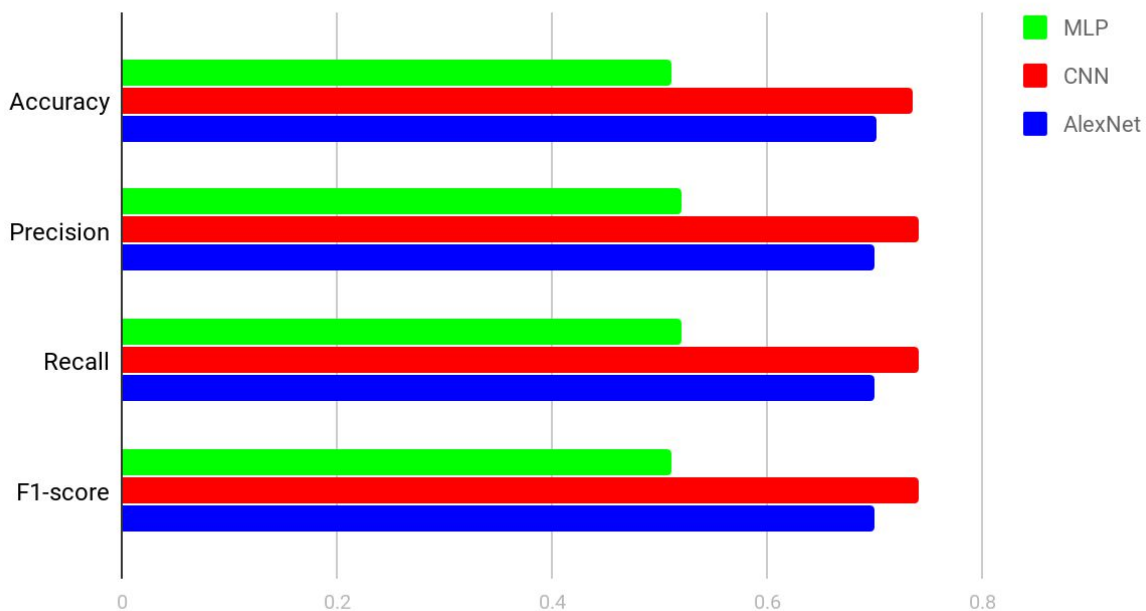
## 陸、結論

根據CIFAR-10 datasets套用MLP、CNN、AlexNet這三種神經網路模型進行訓練和測試後的數據結果我們發現，如圖十一所示，根據比較三者的Accuracy、

Precision、Recall和F1-score後，其中Accuracy是以CNN的表現最佳，Precision是以CNN的表現最佳，Recall是以CNN的表現最佳，F1-score是以CNN的表現最佳。

因此綜合這四項指標後我們的結論是，CIFAR-10 datasets採用CNN的神經網路模型會有最佳的辨識表現。

### Testing Result



(圖十一) Testing Result BarChart

## 柒、 附錄

1. The CIFAR-10 dataset  
[https://www.cs.toronto.edu/~kriz/cifar.html?fbclid=IwAR0K2U1dW8NB\\_TxvngThEy6qTsxQMFHcmHv1YmPMj1qwu2cTALgB5r0KPLg](https://www.cs.toronto.edu/~kriz/cifar.html?fbclid=IwAR0K2U1dW8NB_TxvngThEy6qTsxQMFHcmHv1YmPMj1qwu2cTALgB5r0KPLg)
2. MLP\_MODEL  
“tensorflow+keras深度學習人工智慧實務應用”
3. CNN model參考資料  
<http://yhhuang1966.blogspot.com/2018/04/keras-cnn-cifar-10.html>
4. AlexNet model參考資料  
[https://blog.csdn.net/bryant\\_meng/article/details/86527282](https://blog.csdn.net/bryant_meng/article/details/86527282)
5. CNN、AlexNet程式碼實作  
[https://github.com/jonathanaa/data\\_mining\\_ntu/tree/master/hw2](https://github.com/jonathanaa/data_mining_ntu/tree/master/hw2)