

課堂活動與討論：  
CNN小專案  
(2020/05/05)

曾宏鈞 06160485

徐友笙 05360365

蔡毓丞 06370136

盧君彥 05360153

# 1. 截圖

buildings images:

1867.jpg



1636.jpg



1593.jpg



1107.jpg

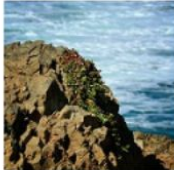


1325.jpg



sea images:

1832.jpg



1706.jpg



124.jpg



1251.jpg



1251.jpg



street images:

1121.jpg



1411.jpg



1435.jpg



106.jpg



1472.jpg



mountain images:

1582.jpg



1257.jpg



1464.jpg



153.jpg



1065.jpg



glacier images:

1358.jpg



1519.jpg



1013.jpg



1516.jpg



105.jpg



forest images:

175.jpg



1720.jpg



1530.jpg



1765.jpg



1402.jpg



## 第一層

(1, 1, 148, 148, 32)

Shape of conv: (32, 148, 148)

len of conv: 32

(1, 1, 73, 73, 32)

Shape of conv: (32, 73, 73)

len of conv: 32

(1, 1, 71, 71, 64)

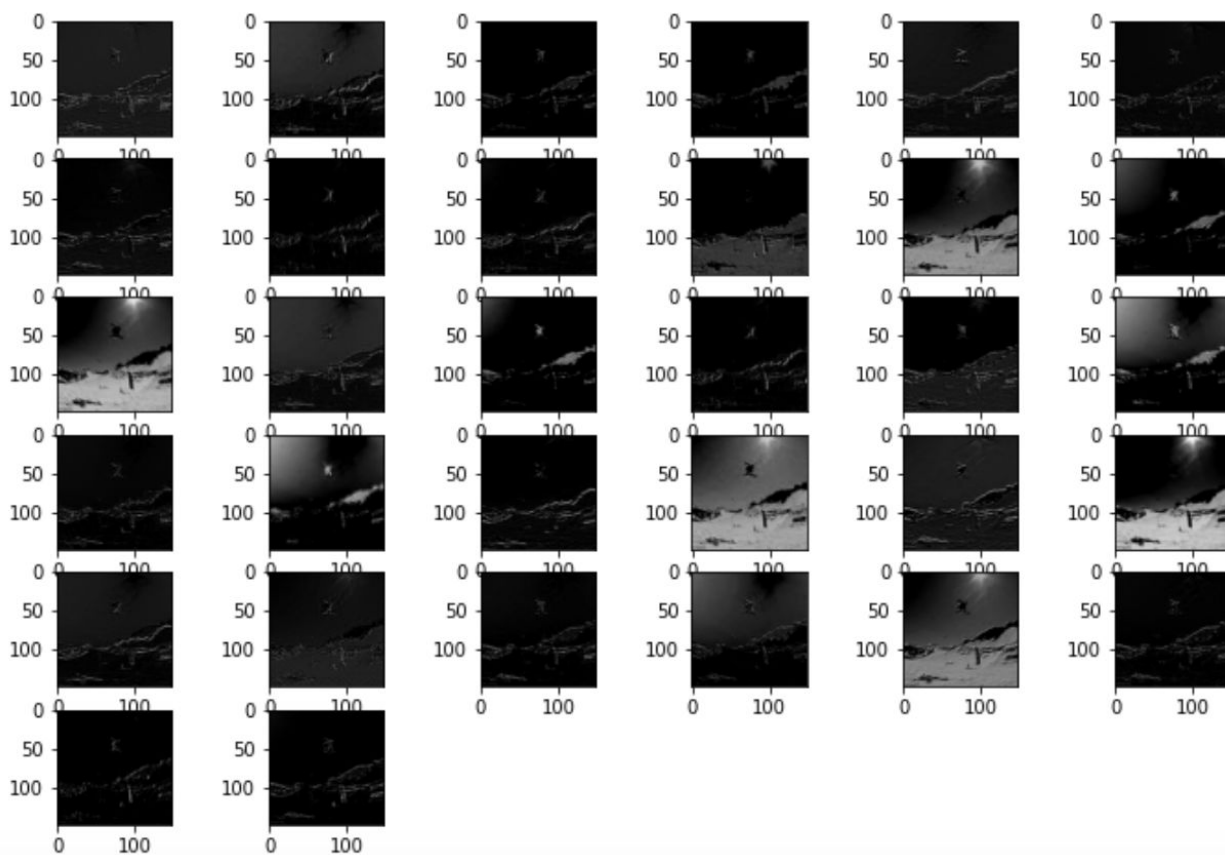
Shape of conv: (64, 71, 71)

len of conv: 64

(1, 1, 34, 34, 64)

Shape of conv: (64, 34, 34)

len of conv: 64

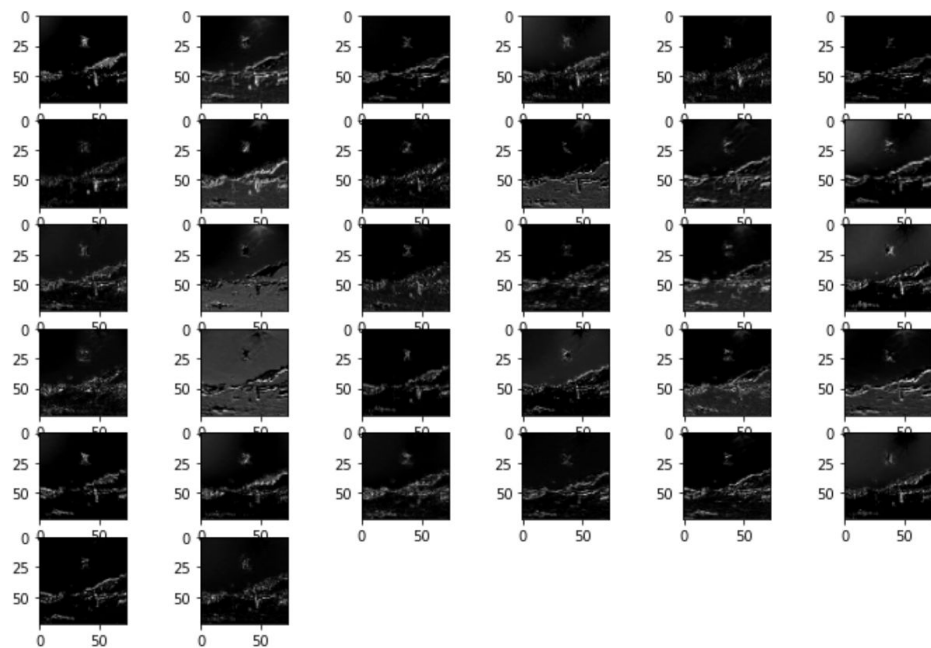


Conv2D

BatchNormalization

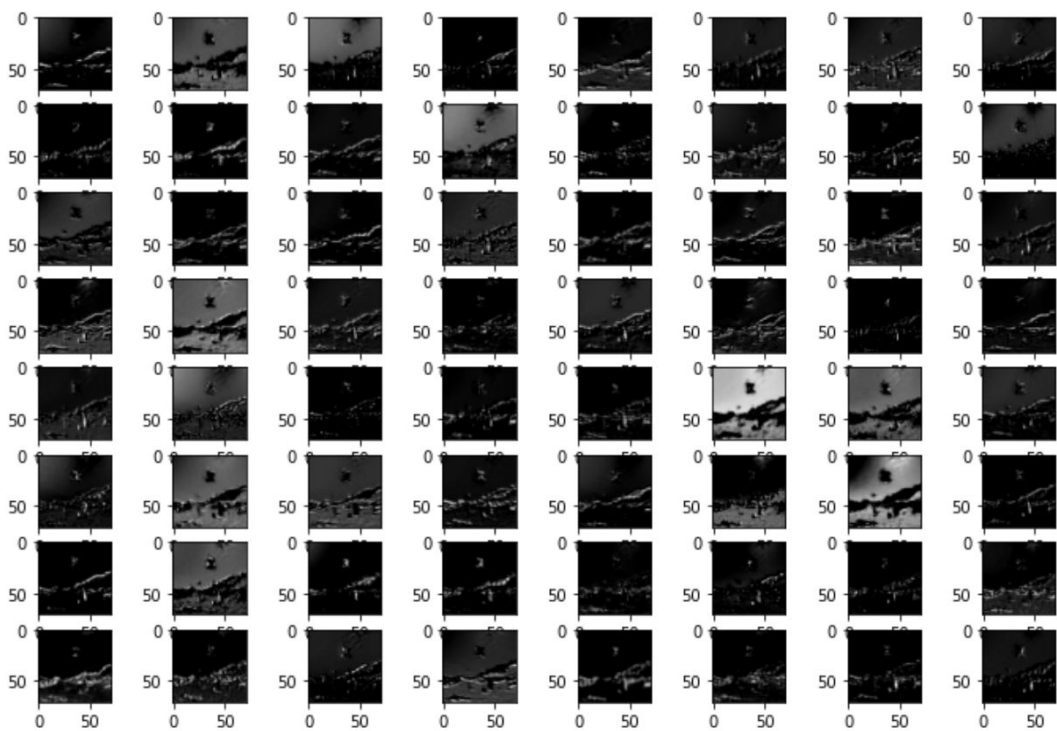
Activation

## 第二層

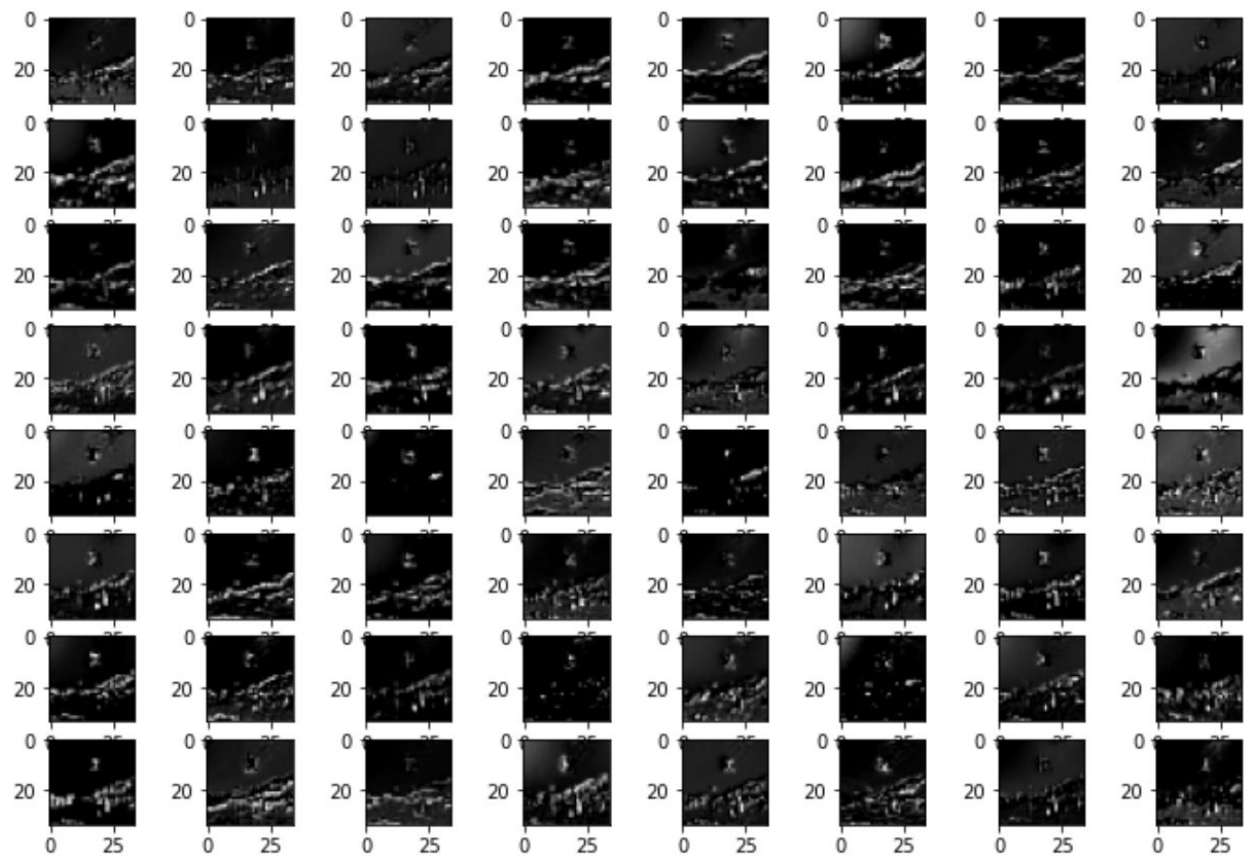


Conv2D  
BatchNormalization  
Activation  
MaxPooling2D

## 第三層



Conv2D  
BatchNormalization  
Activation



Conv2D  
BatchNormalization  
Activation  
MaxPooling2D

## 2. 參數計算

Model: "sequential_2"		
Layer (type)	Output Shape	Param #
=====		
conv2d_5 (Conv2D)	(None, 148, 148, 32)	896
batch_normalization_6 (Batch Normalization)	(None, 148, 148, 32)	128
activation_7 (Activation)	(None, 148, 148, 32)	0
conv2d_6 (Conv2D)	(None, 146, 146, 32)	9248
batch_normalization_7 (Batch Normalization)	(None, 146, 146, 32)	128
activation_8 (Activation)	(None, 146, 146, 32)	0
max_pooling2d_3 (MaxPooling2D)	(None, 73, 73, 32)	0
conv2d_7 (Conv2D)	(None, 71, 71, 64)	18496
batch_normalization_8 (Batch Normalization)	(None, 71, 71, 64)	256
activation_9 (Activation)	(None, 71, 71, 64)	0
conv2d_8 (Conv2D)	(None, 69, 69, 64)	36928
batch_normalization_9 (Batch Normalization)	(None, 69, 69, 64)	256
activation_10 (Activation)	(None, 69, 69, 64)	0
max_pooling2d_4 (MaxPooling2D)	(None, 34, 34, 64)	0
flatten_2 (Flatten)	(None, 73984)	0
dense_3 (Dense)	(None, 512)	37880320
batch_normalization_10 (Batch Normalization)	(None, 512)	2048
activation_11 (Activation)	(None, 512)	0
dropout_2 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 6)	3078
activation_12 (Activation)	(None, 6)	0
=====		
Total params: 37,951,782		
Trainable params: 37,950,374		
Non-trainable params: 1,408		
=====		

Conv1 :  $(3 \times 3 \times 3 + 1) \times 32 = 896$

BN :  $4 \times 32 = 128$

Conv2 :  $(3 \times 3 \times 32 + 1) \times 32 = 9248$

BN :  $4 \times 32 = 128$

Conv3 :  $(3 \times 3 \times 64 + 1) \times 32 = 18496$

BN :  $4 \times 64 = 256$

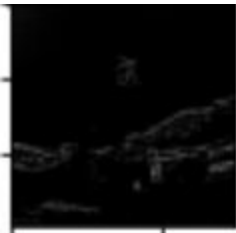


Conv4 :  $(3 \times 3 \times 64 + 1) \times 64 = 36928$

BN :  $4 \times 64 = 256$

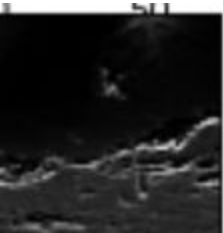
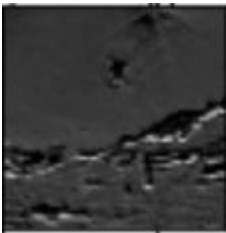

## 2.心得

這次的實驗中可以看出，上次的作業模型裡面每一層的CNN學到了什麼，原理是透過抽取訓練資料集裡面的資料丟進模型裡面去做訓練，視覺化每一張圖片經過CNN層的結果。然而可以看出第一層有點像對圖片去做不同的filter的結果，由於CNN是對原始圖片去做kernel的運算。雖然在mnist資料及中可以明顯地看出，模型越深代表在抓取細部特徵。不過或許因為我們的資料集是風景照，因此視覺化的效果不是這麼明顯，或許可以再嘗試根據視覺化的結果更改模型架構去更改我們的模型架構。

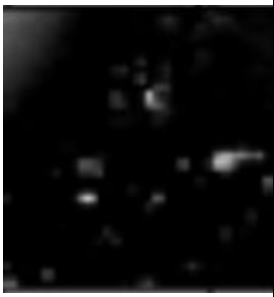


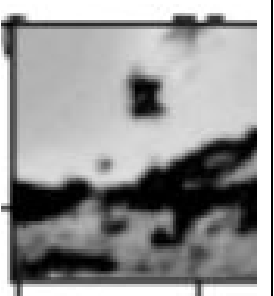
我們透過將測試讀入圖片，輸出一張圖片進入我們每個學習好的filter去看每一層是學習到了什麼，由第一層的CNN可以看出，我們有一些的filter正在對圖片做線段化，有些在強調深色區域，有一些較接近原圖。

線段化	深色區	接近原圖
		





第二層有加MaxPooling後，圖片變模糊，但使線段化效果更明顯，且看起來有更不一樣的特徵針對影像中的小物件。

線段化更明顯	與其亮區不同1	與其亮區不同2
		

第三層是單純conv，但filter數目增加了兩倍，可以看出有更多部分特徵被特別擷取出來，但亮區的差異有限，亮區比較接近一區一區。

特徵小物件1	特徵小物件2	特徵小物件3	特徵小物件4
			

第四層除了conv層，也有MaxPooling層，因此線段化更明顯，也使原本看起來不太清楚的東西亮點增強許多，雖然圖片縮小了，但明顯可以看得出有些filter學習到的特徵，不過有一些特徵與上層有點相似。

特徵小物件1	特徵小物件2	特徵小物件3	特徵小物件4
			

雖然這次的實驗中，相較於其他人的辨識物體，由於我們要辨識的主題是風景，觀察資料集後發現，即便是同一屬性，但資料相去甚多，可能是上次作業實驗準確率只能在0.7多的原因，不過這次的作業可以觀察到，模型愈深特徵之間的差異愈大，在意義而顏代表愈深我們更可以去擷取到更細部的細節，雖然此次在辨識場景的應用看的差異有限，但還是可以從中瞭解到深層卷積網路在做的事情，就像是在影像處理中所做的卷積運算所得到的圖，只是影像處理我們需要使用已知的kernel值，而深度卷積網路是根據輸入的資料去訓練kernel的權重，跟MLP相比效率又大增許多，可見深度卷積網路的興起是有原因的，可以改善傳統影像處理方法的不足。





 4441.jpg



 4437.jpg



 4433.jpg



 4431.jpg



 4428.jpg



 4423.jpg



 4412.jpg



 4410.jpg