

# Machine Learning

## Final Project

組別:第32組

組員:

0459604蕭義橙

0556150王強毅

A023140楊博翔

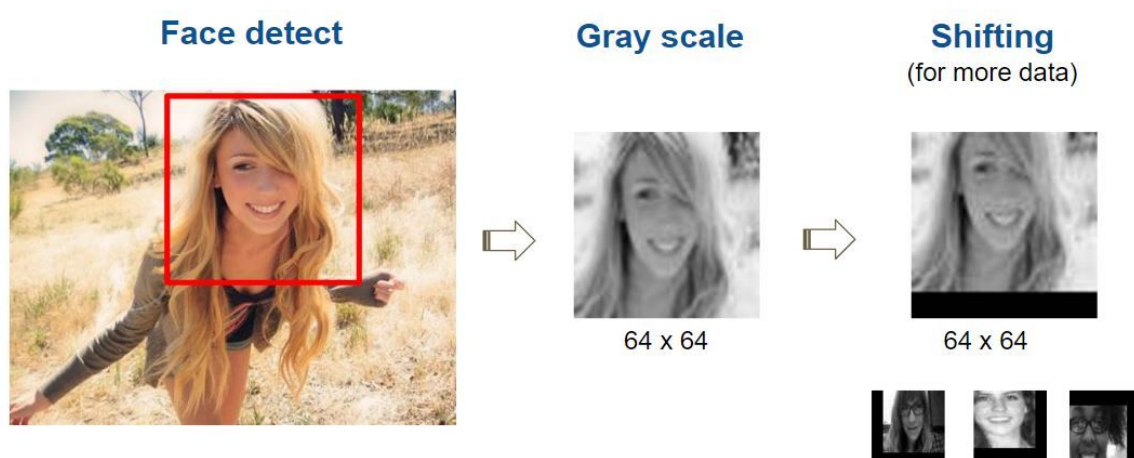
# 目錄

<b>Introduction</b>	<b>3</b>
<b>Idea &amp; Data Preprocessing</b>	<b>3</b>
Face detect	3
Gray scale	3
Shifting (照片平移，加黑邊)	3
<b>Model Design-CNN</b>	<b>4</b>
Model Structure	4
Based Model	5
Unbalanced Data	5
Faster Converging	5
<b>Model Design-SVM</b>	<b>6</b>
<b>Experiment</b>	<b>6</b>
CNN	6
SVM	8
<b>Conclution</b>	<b>9</b>
CNN is better than SVM	9
<b>Other Discussion</b>	<b>9</b>
Rotation	9
Blur	9
Gaussian derivatives	10
<b>Reference</b>	<b>11</b>

# 1. Introduction

先利用 [ **face detect** ] 將人臉萃取出來，並且轉成 [ **64 x 64灰階** ]，利用 [ **平移 Shifting** ] 的方式增加data數量，再餵入我們的 model [ **CNN為主，SVM為輔** ]。

## 2. Idea & Data Preprocessing



### a. Face detect

利用 OpenCV 將人臉擷取出來，若無法辨識人臉的話，則使用原圖片。

### b. Gray scale

將人臉 (或原圖) downsize 成 64 X 64，並轉成灰階。

### c. Shifting (照片平移，加黑邊)

為了解決 data 數量不足的問題，我們採用 Shifting 的方式，將照片隨機平移，並加入黑邊。

### 3. Model Design-CNN

#### a. Model Structure

Layer Index	Input Size	Output Size	Layer	Activation Function
1	64*64*3	64*64*192	CNN(5*5,192)	ReLU
2	64*64*192	64*64*160	CNN(1*1,160)	ReLU
3	64*64*160	64*64*96	CNN(1*1,96)	ReLU
4	64*64*96	32*32*96	Maxpool(3*3)	-
5	32*32*96	32*32*96	Dropout(0.5)	-
6	32*32*96	32*32*192	CNN(5*5,192)	ReLU
7	32*32*192	32*32*192	CNN(1*1,192)	ReLU
8	32*32*192	32*32*192	CNN(1*1,192)	ReLU
9	32*32*192	16*16*192	Maxpool(3*3)	-
10	16*16*192	16*16*192	Dropout(0.5)	-
11	16*16*192	16*16*192	CNN(5*5,192)	ReLU
12	16*16*192	16*16*192	CNN(1*1,192)	ReLU
13	16*16*192	16*16*192	CNN(1*1,8)	ReLU
14	16*16*8	1*1*8	Avgpool(16*16)	-

首先我們先餵入64\*64\*3 的灰白相片，可以注意到我們雖然用的是灰白相片，但是為了方便未來有可能會想用彩色照片來訓練model，於是我們將灰白相片的一層layer複製成三層，這樣 model 的自由度更高。

最後的layer會output出八個值，經過 softmax 過後即為一張照片屬於任一個類別的機率值了，之後在 demo predict 時，就是基於這個機率值去做預測。

## b. Based Model

在 CNN 的 model 設計上，我們主要是參考 **NIN(Network in Network)** 這篇論文提出的架構，我們會選擇以這個 model 做為參考的原因主要是因為這個架構有兩個對我們來說很重要的特點：

第一點是需要的參數比較少，在這個架構中有三分之二的 CNN layer 的 **kernal size 使用 1\*1 的大小**，這樣大幅地減少了需要訓練的參數量，由於我們的資料量比較少，所以若參數量過大，則有可能訓練不起來；

第二點是由於資料量少容易 overfitting 的問題，而這篇論文利用了 **dropout** 以及在最後一層使用 **average pool** 而非一般其他架構會用的 fullyconnected 這兩種方法降低 overfitting 的發生率，因此我們選擇以這個 model 為基礎去做修改。

## c. Unbalanced Data

另外一個資料的問題是，有的類別資料量較多，有的類別資料量較少，也就是資料不平衡的問題，可能會導致 model 最後訓練出來完全放棄某一個類別，對於這個問題我們的解決方法是，在 gradient 前做 loss 計算時，若是在 **資料量較少的類別** 時判斷錯誤，則會有較高的 **penalty weight**，這樣可以提高資料量較少的類別對 model 訓練的影響。

## d. Faster Converging

最後我們為了讓 model 可以更快的收斂，在每一層 CNN layer 加上了 **Batch Normalization**。

## 4. Model Design-SVM

老師課堂上有提到，目前人臉辨識的冠軍是使用 CNN，但幾年前是用 SVM，所以我們想試試看 SVM 與 CNN 的差別。

根據 HW4 的設置，我們嘗試使用 **Linear**、**Polynomial (degree 2 to 4)**、**radial** 這幾種 kernel function。

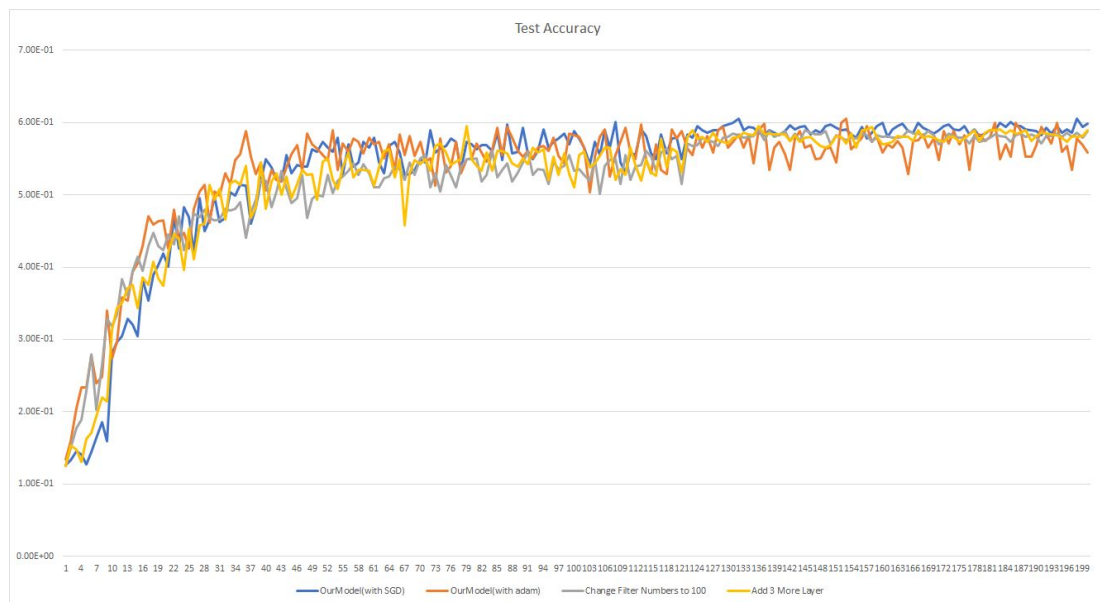
## 5. Experiment

### a. CNN

- Test Data：每一組類別的 **前100張** 做為test data，總共**800張**。
- Training Data：剩餘的照片，要注意的是我們擔心若使用助教幫我們翻轉好的照片，每一類會有100張 training images只是該類 testing images的翻轉照片，**對於某些照片來說，可能翻轉前後根本差不多**（例如短頭髮的男性正面照片，或者是中分髮型女性的正面照片），這樣會導致training data與testing data產生重複的問題，所以**我們只取用翻轉前的部分**，在針對training data自己去做data augmentation，包括水平翻轉與先在周圍加黑邊再重新random crop回64\*64大小的照片。
- Batch Size：64
- Epoch：200
- Learning rate:
  - epoch 1~100 : 0.1
  - epoch 101~180 : 0.01
  - epoch 181~200 : 0.001
- Optimization

- SGD with momentum 0.9
- For Comparison we also try to use Adam optimizer
- Results

curve's color	model調整描述
藍色	我們 <b>最終選擇的model</b> ，以此架構為基礎
橘色	將藍色的optimizer由SGD改為Adam
灰色	將藍色的後面6層CNN的filter numbers為192的部分減少至100
黃色	把藍色的網絡架構再加深



可以看到不管哪一種設法的結果都差不多，而我們最後取用最佳的組合(藍色的曲線)作為我們最終的model設定，最後的 testing accuracy 為 **59.3%**。

## b. SVM

kernel	Accuracy (加黑邊)	Accuracy (不加黑邊)
linear	21.8 %	25.8 % 🍷
poly_2	27.8 %	28.5 % 🍷
poly_3	35.2 % 🍷	30.8 %
poly_4	26.6 %	28.1 % 🍷
radial	34.4 %	<b>37.9 %</b> 🍷

整體而言，加了黑邊反而會讓準確率下降（只有 poly\_3 上升），猜測是因為加了黑邊後，讓很多照片反而產生類似的區塊（黑邊），導致 SVM 的準確率下降。

而在本次實驗中，表現最好的是採用 radial（不加黑邊），accuracy 達到 **37.9%**，雖然已經是 SVM 中表現最好的，但還是輸給 CNN 一大截，與上課提及的比賽結果一致。



## 6. Conclusion

### a. CNN is better than SVM

在本次實驗中，CNN 的準確率遠高於 SVM ( 59.3% > 37.9% )  
，與老師上課提及的比賽結果一致。

### b. Shifting 不適合 SVM

平移並加黑邊後，會造成SVM準確率下降，猜測原因可能為：這些黑邊導致許多圖片產生類似的區塊，因而誤導 SVM 的判斷。

然而，CNN 是一小塊一小塊來學的，因此黑邊的影響就不大。平移後可以增加 data 數量，就有更多資源可以來學 (學習眼睛、嘴....等)。

## 7. Other Discussion

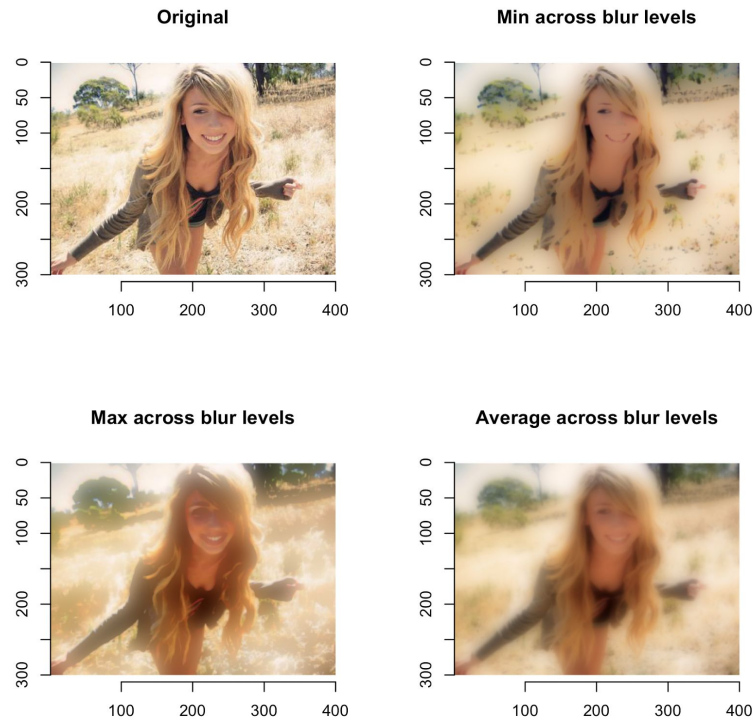
### a. Rotation

可以利用 **眼睛、嘴吧** 來定位，就可以把每個人臉都固定在同一個位置。



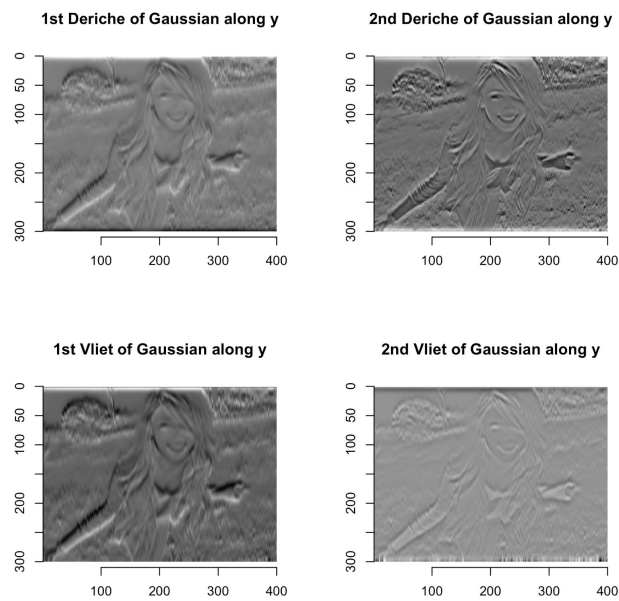
### b. Blur

為了增加data量，而把原圖加入noise，讓照片變模糊，但發現效果並不好。



### c. Gaussian derivatives

為了增加data量，使用Gaussian derivatives(一階、二階)，但發現效果並不好。



## 8. Reference

- a. NIN paper (<https://arxiv.org/abs/1312.4400>)
- b. Batch Normalization (<https://arxiv.org/abs/1502.03167>)
- c. TensorFlow Website (<https://www.tensorflow.org/>)
- d. face detection  
([http://cn.mathworks.com/matlabcentral/fileexchange/36855-face-parts-detection?s\\_tid=srchtitle#userconsent#](http://cn.mathworks.com/matlabcentral/fileexchange/36855-face-parts-detection?s_tid=srchtitle#userconsent#))