數學解題方法
期末報告第二組

組員：
410831103 謝德錦
410831105 顏佑任
410831117 馬嘉笛
410831127 董皓旻
410831140 溫柏勛

影像處理介紹

對圖像進行分析、加工和處理，以達到我們心中所希望圖像的樣子。

目前大多數的圖像均是以數位形式儲存，因而影像處理很多情況下指**數位影像處理**。此外，基於光學理論的處理方法依然占有重要的地位。

與**電腦科學、人工智慧**等領域也有密切的關係。

影像處理介紹(相關科目)

微積分、高等微積分

線性代數

機率、統計

數值分析

Matlab、Octave相關知識(程式)

圖片形式

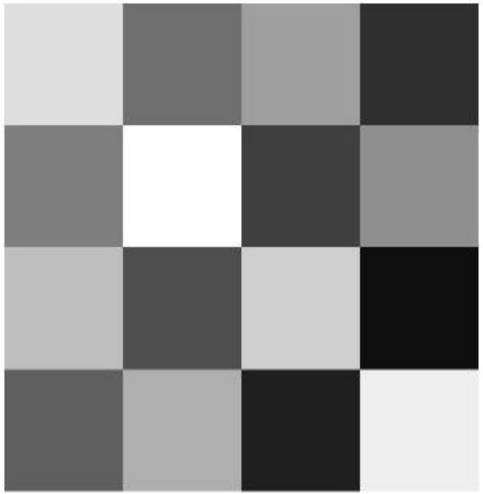
圖片在電腦裡皆會被儲存為M*N的矩陣

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N-1} & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N-1} & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{M-1,1} & a_{M-1,2} & \cdots & a_{M-1,N-1} & a_{M-1,N} \\ a_{M,1} & a_{M,2} & \cdots & a_{M,N-1} & a_{M,N} \end{bmatrix}$$

黑白編碼

矩陣內會填上數字0~255，0為最黑，255為最白。

$$\begin{bmatrix} 223 & 111 & 159 & 47 \\ 127 & 255 & 63 & 143 \\ 191 & 79 & 207 & 15 \\ 95 & 175 & 31 & 239 \end{bmatrix}$$



程式碼介紹

- A=imread(‘圖片名稱.圖片檔’);(將A設定為圖片)

  例 : A=imread(‘angrycat.jpg’);
- [M,N]=size(A);(M,N分別為A的列數與行數)

  M=123, N=456, 則A為123*456的矩陣
- A=uint8(A);(將A儲存為uint8形式)
- A=double(A);(將A儲存為double形式)

程式碼介紹

- imshow(A,[0,255]);(秀出圖片A,且值介於0~255)
- subplot(m,n,p);(在m*n的方格中，第p個位置顯示圖片)
- A=imhist(A);(秀出圖片A的色彩分布圖，以直方圖顯示)
- A=rgb2gray(A);(將圖片A轉變為灰階圖片)

程式碼介紹

- A=rgb2ycbcr(A);**(將A轉為YCbCr格式)**

  Y：亮度

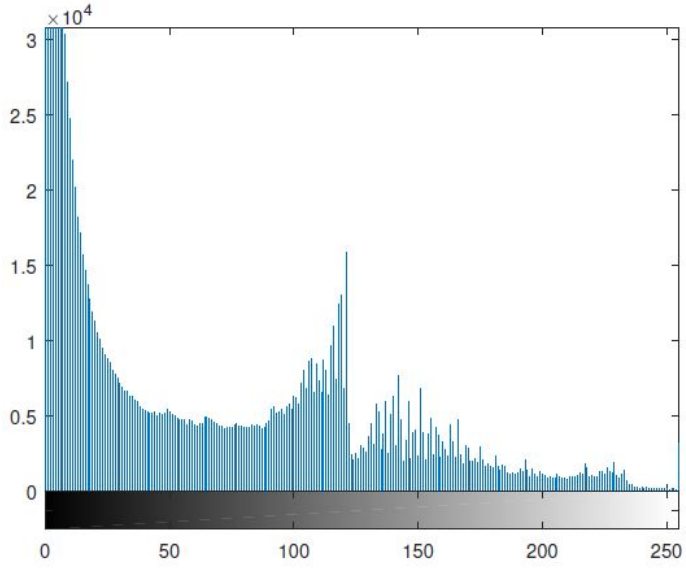
  Cb：藍色色差

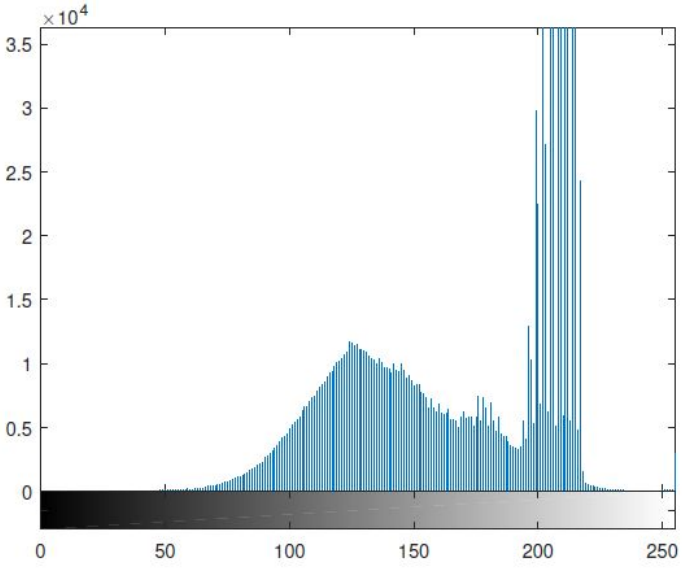
  Cr：紅色色差

  人們對於亮度的敏感度比對彩度高，且RGB格式無法對於亮度做處理(柔和度、銳利度)，故需轉成YCbCr格式做處理

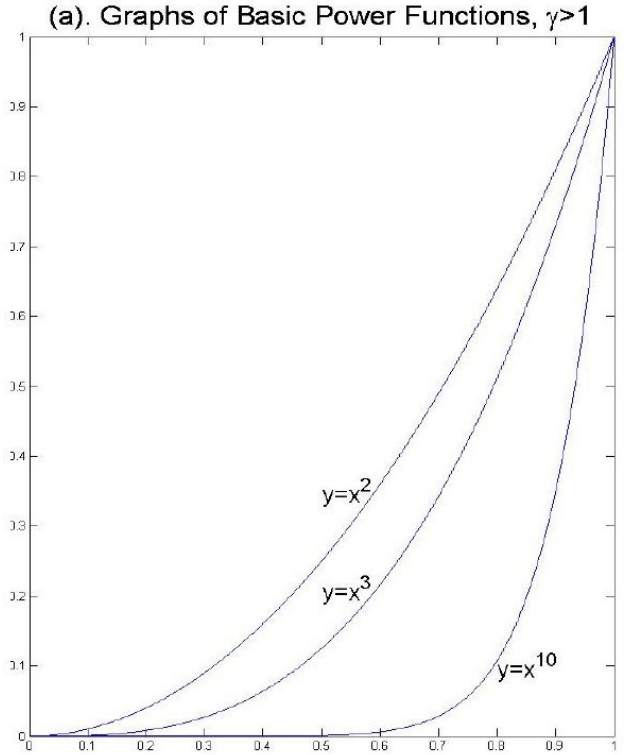
先從簡單的例子:曝光不足vs曝光過度

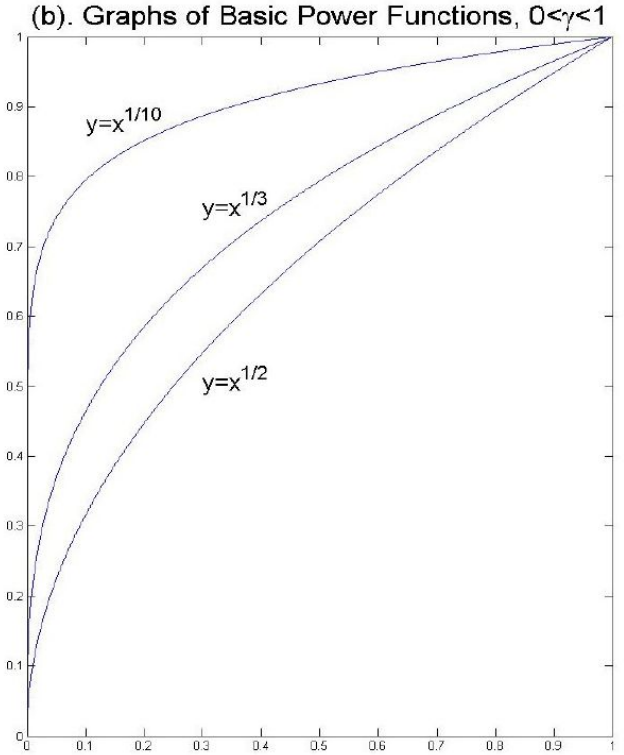
如何調整曝光度？


×10^4
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250


×10^4
3.5
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250

考慮冪函數(Power functions)



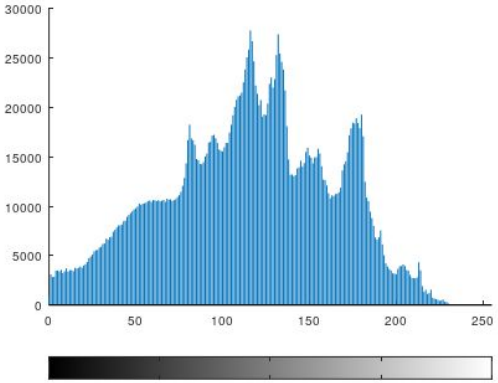


Then every pixel in B can be calculated using the formula

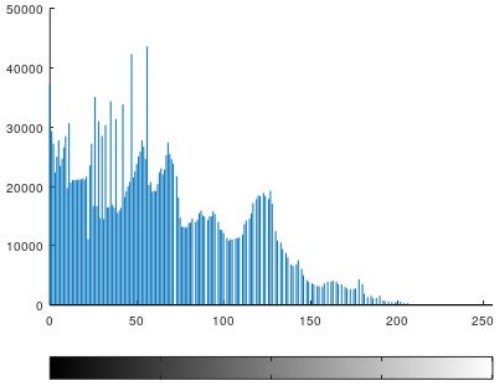
$$B(m,n) = A(m,n)^{\gamma}.$$

```
#初始化
clear all
clc
pkg load image

#讀檔+彩色轉灰階
A = rgb2gray(imread('angrycat.jpg'));

#標準化數據
A = double(A)/255;

#參數
gamma1 = 2;
gamma2 = 0.5;

#函式
B1 = A.^gamma1;
B2 = A.^gamma2;

#反標準化
A = uint8(A*255);
B1 = uint8(B1*255);
B2 = uint8(B2*255);

#原始圖檔+轉換後圖檔+直方圖
subplot(2,3,1)
imshow(A)
title('Original Image', 'fontsize', 18);
subplot(2,3,2)
imshow(B1)
title('The f(x)=x^2 Effect', 'fontsize', 18);
subplot(2,3,3)
imshow(B2)
title('The g(x)=sqrt(x) Effect', 'fontsize', 18);
subplot(2,3,4)
imhist(A)
subplot(2,3,5)
imhist(B1)
subplot(2,3,6)
imhist(B2)
```

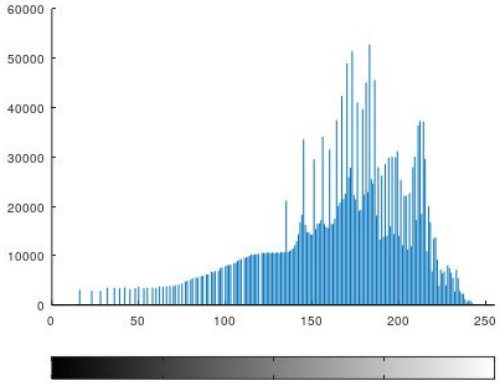
Original Image



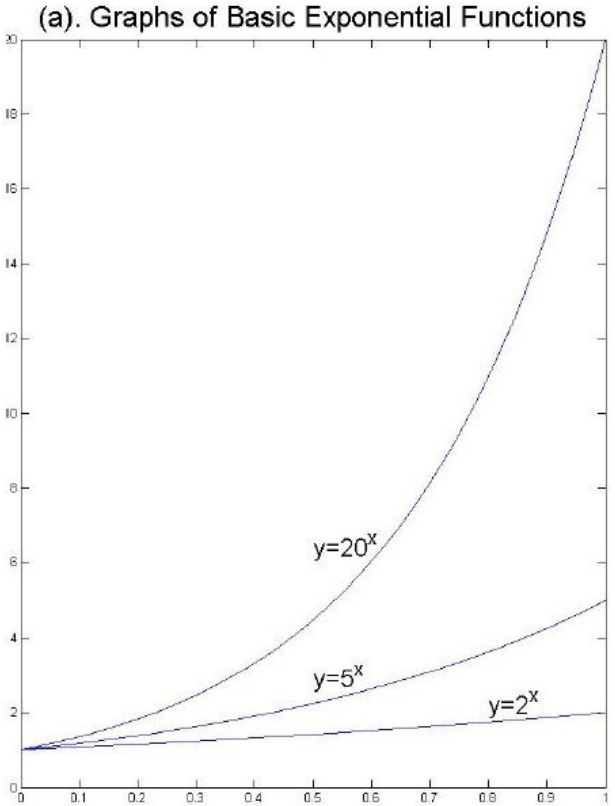
The $f(x)=x^2$ Effect

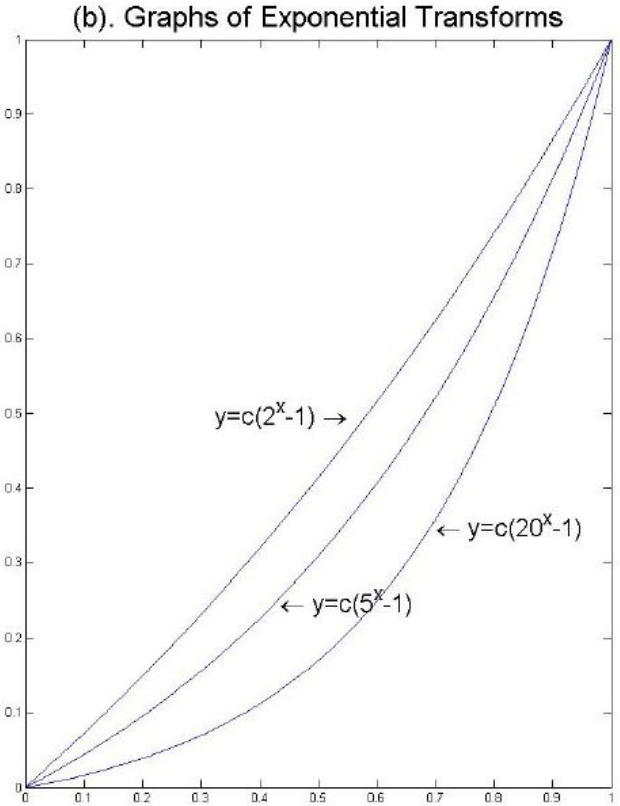


The g(x)=sqrt(x) Effect



考慮指數函數(Exponential functions)





會有什麼問題？

We would like our function to map the interval $[0, 1]$ into the interval $[0, 1]$.

Putting all our wishes together, we would like to have a function described by the general formula

$$f(x) = c(b^x - 1)$$

that satisfies the conditions

$$f(0) = 0 \quad \text{and} \quad f(1) = 1.$$

The condition $f(1) = 1$ implies that

$$c = \frac{1}{b - 1}.$$

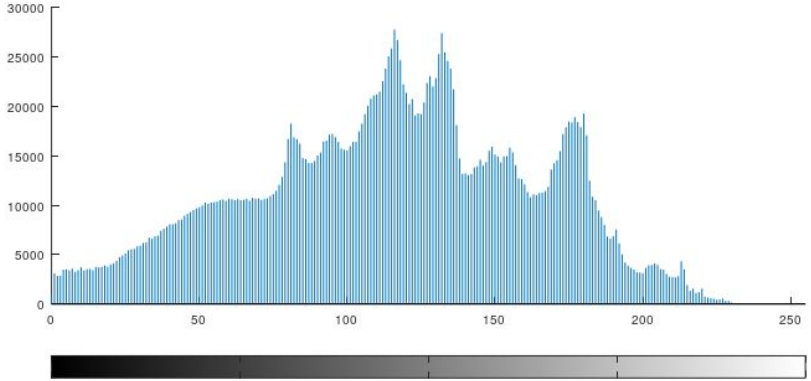
We can then calculate every pixel value

$$B(m,n) = c\left(b^{A(m,n)} - 1\right).$$

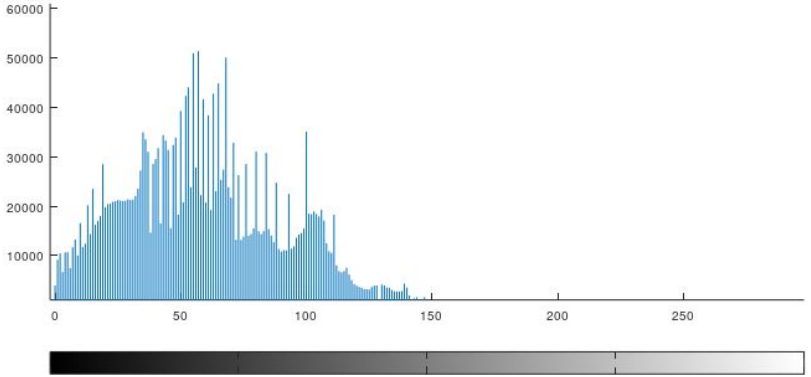
```
#初始化
clear all
clc
pkg load image

#讀檔+彩色轉灰階+標準化數據
A = double(rgb2gray(imread('angrycat.jpg')))/255;

#參數+函式
b = 4;
B = (1/b)*(b.^A-1);
```

```
#反標準化
A = uint8(A*255);
B = uint8(B*255);

#原始圖檔+轉換後圖檔+直方圖
subplot(2,2,1)
imshow(A)
title('Original Image', 'fontsize', 18);
subplot(2,2,2)
imshow(B)
title('The f(x)=b^x Effect', 'fontsize', 18);
subplot(2,2,3)
imhist(A)
subplot(2,2,4)
imhist(B)
```

Original Image



The $f(x)=b^x$ Effect



圖像對比度

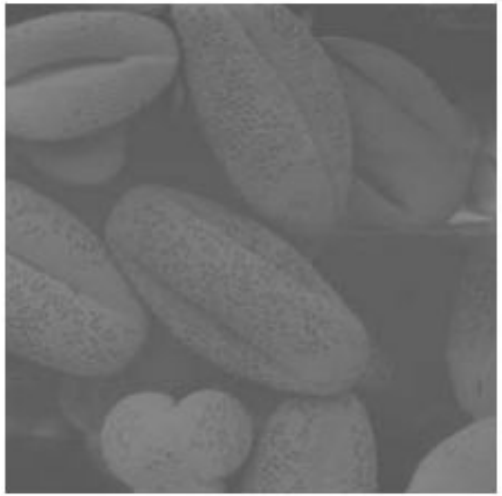
前面我們已經介紹了多種方法來使曝光過度的圖像變暗，使曝光不足的圖像變亮，並提高那些顯得灰暗和乏味的圖像對比度。

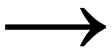
這些方法的共同點是都使用直方圖。

作法傾向於使偏斜的直方圖更加對稱，而高度集中在中心值附近的直方圖變得更加均勻。

圖像對比度

低對比

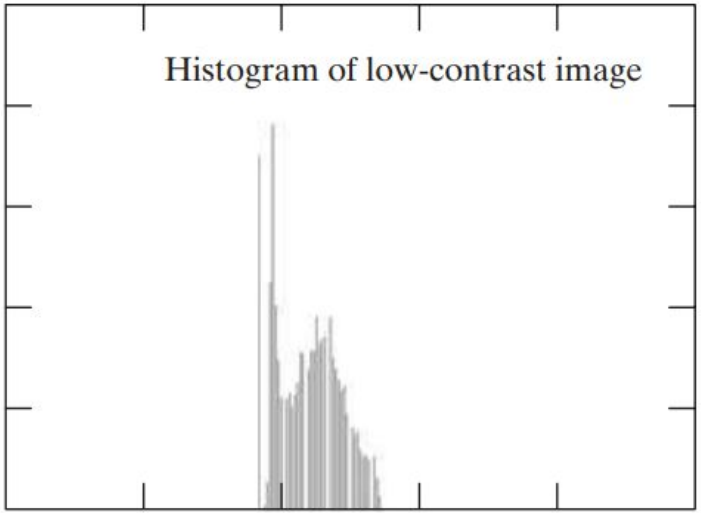




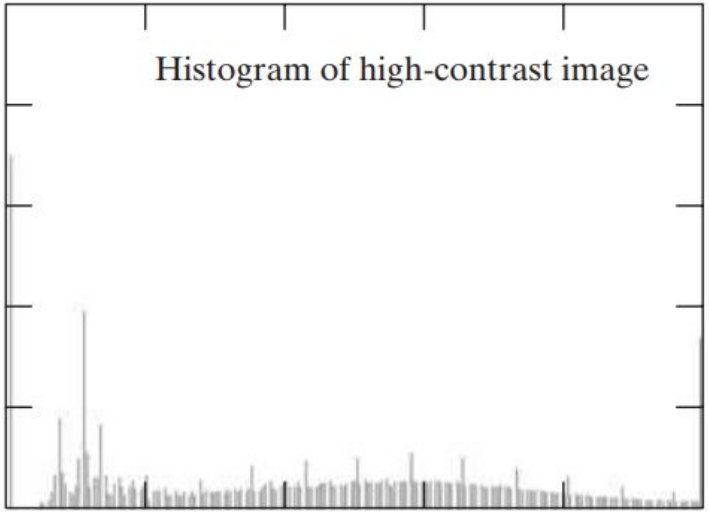
高對比

圖像對比度

低對比



高對比



想法概念

如果我們用 X 表示隨機選取的像素值，則圖像的直方圖提供了機率函數 p(x) 的圖形，而圖像直方圖的平滑輪廓正好是 X的密度曲線。

我們必須設計一個變換函數 g，使由 Y = g(X) 定義的隨機變量 Y 均勻分佈。

想法概念

這意味著我們希望 Y 的機率密度函數在 [0, 255] 或 [0, 1] 區間內的值保持不變。

其實我們可以讓 Y 具有任何指定的分佈，但我們將從使像素值分佈均勻的最簡單情況開始。

例子

假設 X 為連續的隨機變數，其 pdf 為 $f_X(x) = \begin{cases} 2x, & 0 \le x \le 1, \\ 0, & \text{otherwise}, \end{cases}$
且隨機變數 Y 定義為 Y = g(X) = 3X

所以，Y 的 cdf 為 $F_Y(y) = P(Y \le y) = P(3X \le y) = P(X \le \frac{y}{3})$

$$= \int_0^{y/3} 2x dx = \frac{y^2}{9} \quad \text{for all values } 0 \le y \le 3$$

得到 $f_Y(y) = F'_Y(y) = \begin{cases} 2y/9, & 0 \le y \le 3, \\ 0, & \text{otherwise}. \end{cases}$

一般化

假設 X 為連續的隨機變數，落在 [a, b] 區間
假設隨機變數 Y 定義為 Y = g(X)，g 為嚴格遞增且可微分的函數

所以，Y 的 cdf 為

$$
\begin{aligned}
F_Y(y) &= P(Y \le y) = P(g(X) \le y) \\
&= P(X \le g^{-1}(y)) \\
&= \int_a^{g^{-1}(y)} f_X(x)dx \qquad \text{for all values } g(a) \le y \le g(b)
\end{aligned}
$$

一般化

再使用微積分基本定理和反函數定理

可得 $$f_Y(y) = F'_Y(y) = \frac{d}{dy}\left[\int_a^{g^{-1}(y)} f_X(x)dx\right]$$

$$= f_X(g^{-1}(y)) \cdot \frac{d}{dy}[g^{-1}(y)]$$

$$= f_X(g^{-1}(y)) \cdot \frac{1}{g'(g^{-1}(y))}$$

for all values of y for which $g(a) \leq y \leq g(b)$ and $f_Y(y) = 0$ otherwise.

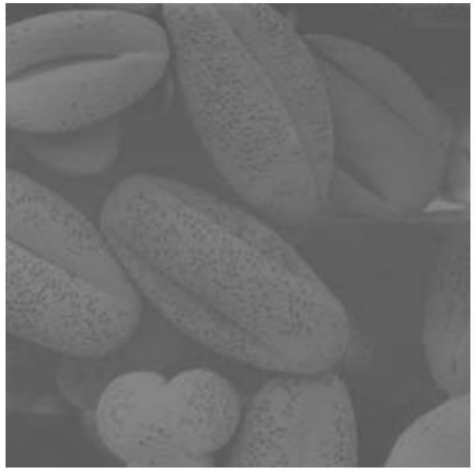
直方圖處理

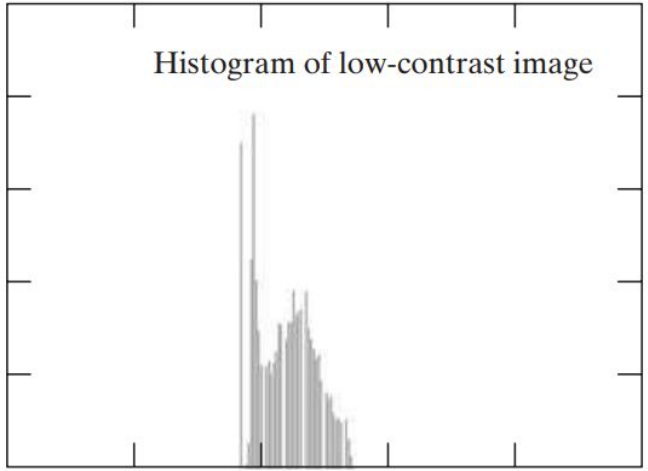
回想一下，強度水平在 [0, L − 1] 範圍內的圖像直方圖是一個離散函數 h(xk) = nk，其中 xk 是第 k 個強度值，nk 是圖像中像素強度 xk 的數量。

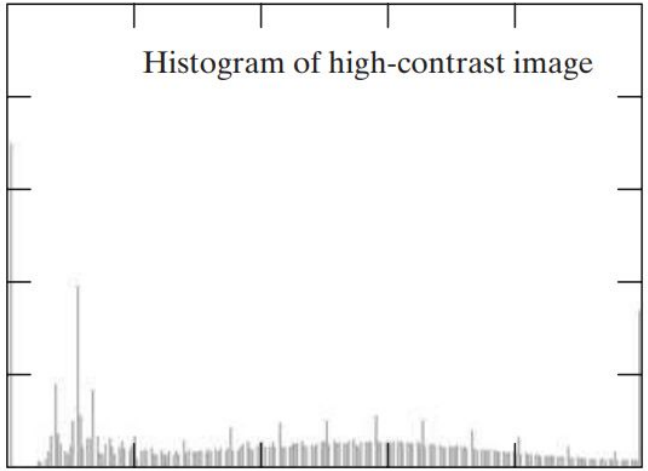
通常的做法是透過將直方圖的每個分量除以圖像中的像素總數來對直方圖進行標準化，用乘積 MN 表示，其中，M 和 N 是圖像的行和列。
因此，標準化直方圖為 p(xk) = nk/MN，for k = 0, 1, 2, · · · , L − 1。

p(xk) 是對圖像中強度水平 xk 出現機率的估計。標準化直方圖的所有分量之和等於 1。

結果




Histogram of low-contrast image


Histogram of high-contrast image

來個小短片放鬆一下

如何旋轉圖片？

The Original Image

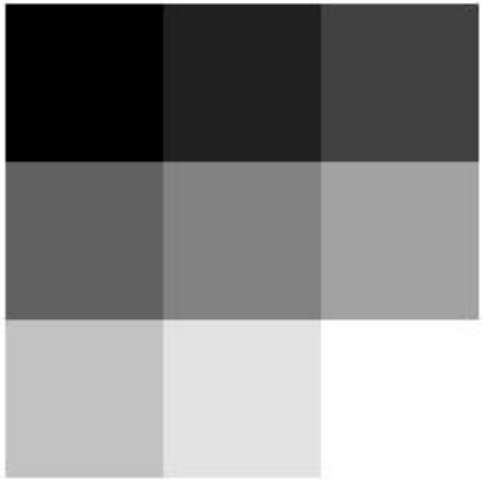
Image Rotated 90°

旋轉矩陣

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

操作過程

```
#範例圖形
Im=[   0   32   64;
      96 128 160;
     192 224 255]
[rows, cols] = size(Im);
```



操作過程

```
#設置旋轉矩陣
theta=pi/2;
A = [cos(theta) -sin(theta);
     sin(theta) cos(theta)];
```

```
A =

    6.1230e-17   -1.0000e+00
    1.0000e+00    6.1230e-17
```

操作過程

```
#找出能包下旋轉後圖形的矩陣
[xG, yG] = meshgrid(0:rows-1, 0:cols-1);
rxy=round(A*[xG(:)'; yG(:)']);
M=max(rxy(1,:))-min(rxy(1,:))+1;
N=max(rxy(2,:))-min(rxy(2,:))+1;
rIm = zeros(M,N);
```

```
rxy =

     0    -1    -2     0    -1    -2     0    -1    -2
     0     0     0     1     1     1     2     2     2

rIm =

     0     0     0
     0     0     0
     0     0     0
```

操作過程

```
#調整座標
invxy=max(round(A\rxy),0)+1;
shifrxy=rxy-min(rxy,[],2)+1;
```

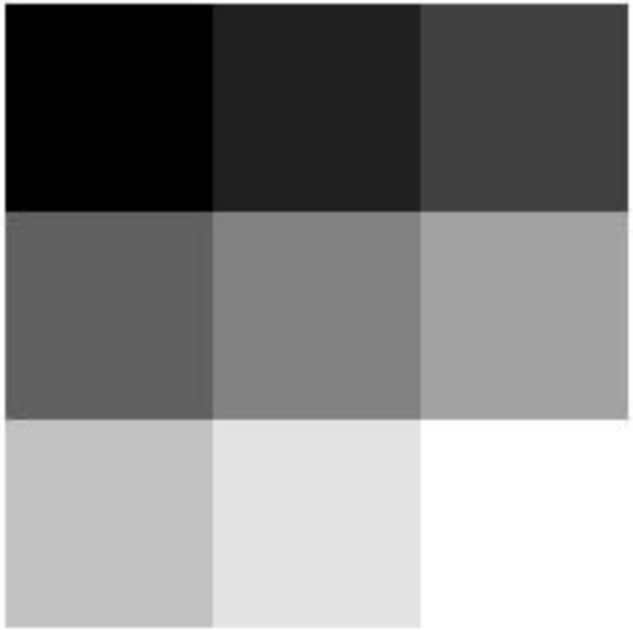
```
invxy =

   1   1   1   2   2   2   3   3   3
   1   2   3   1   2   3   1   2   3

shifrxy =

   3   2   1   3   2   1   3   2   1
   1   1   1   2   2   2   3   3   3
```

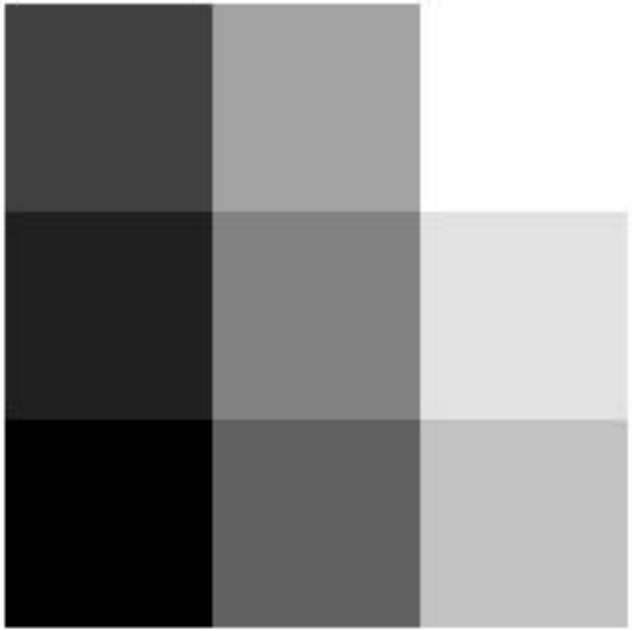
操作過程

```
#交換
for j = 1:cols
  for i = 1:rows
    xy=invxy(:,(j-1)*(rows)+i);
    x = xy(1);
    y = xy(2);
    rx=shifrxy(1,(j-1)*(rows)+i);
    ry=shifrxy(2,(j-1)*(rows)+i);
    if (x<=rows && y<=cols );
       rIm(rx,ry) = Im(x,y);
    end
  end
end
```

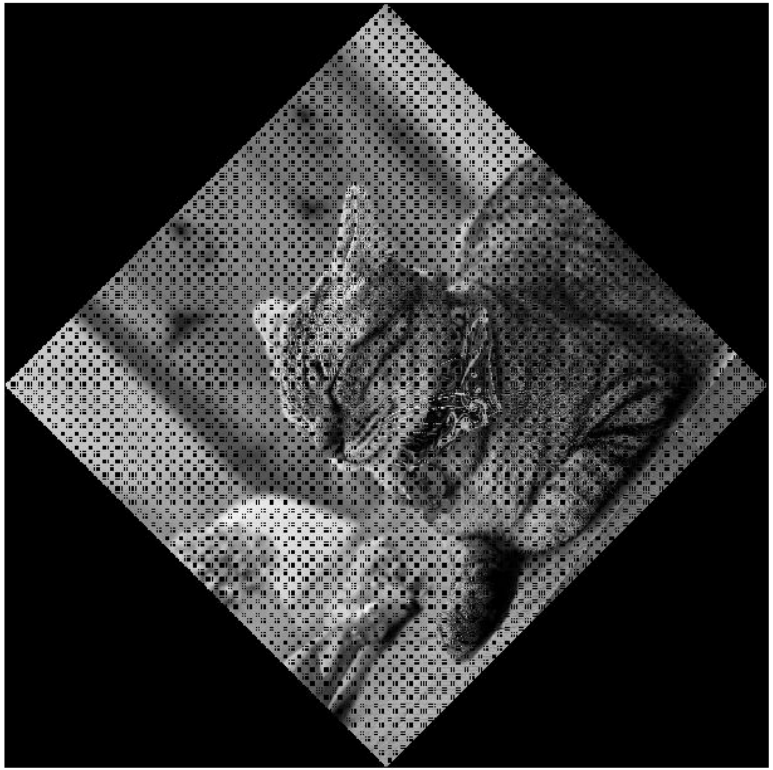
```
#展示圖
figure
subplot(1,2,1)
imshow(Im,[])
subplot(1,2,2)
imshow(rIm,[])
```

操作過程





破洞



懶人福音—imrotate

```
Im=imread('angrycat.jpg');
Im=rgb2gray(Im);
rIm=imrotate(Im,45);
figure
subplot(1,2,1)
imshow(Im,[])
subplot(1,2,2)
imshow(rIm,[])
```

成果展示

圖像模糊

圖像模糊

會需要圖像模糊的可能原因：

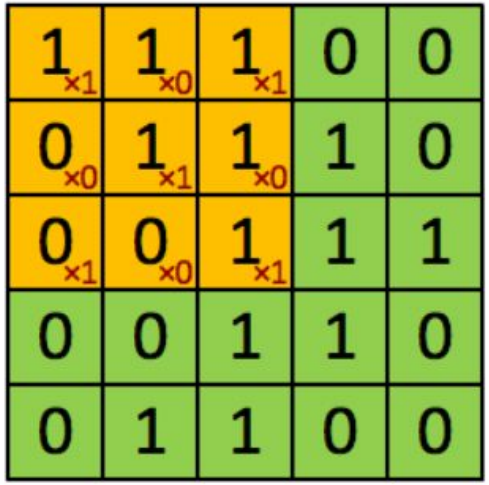
1. **保護隱私**
2. **淡化氛圍，突顯焦點**
3. **產生柔和的輪廓，降低噪點**

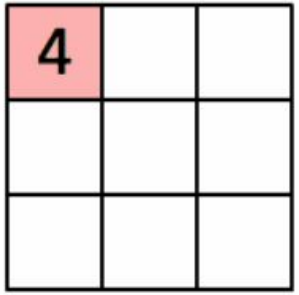
圖像模糊

$$B(m,n) = \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} A(m+k, n+l)$$

實現方法:

右圖是示意圖，我們會取九宮格的數字全部加起來除以9，創造新的圖片。





圖像模糊

原圖

躁點

模糊

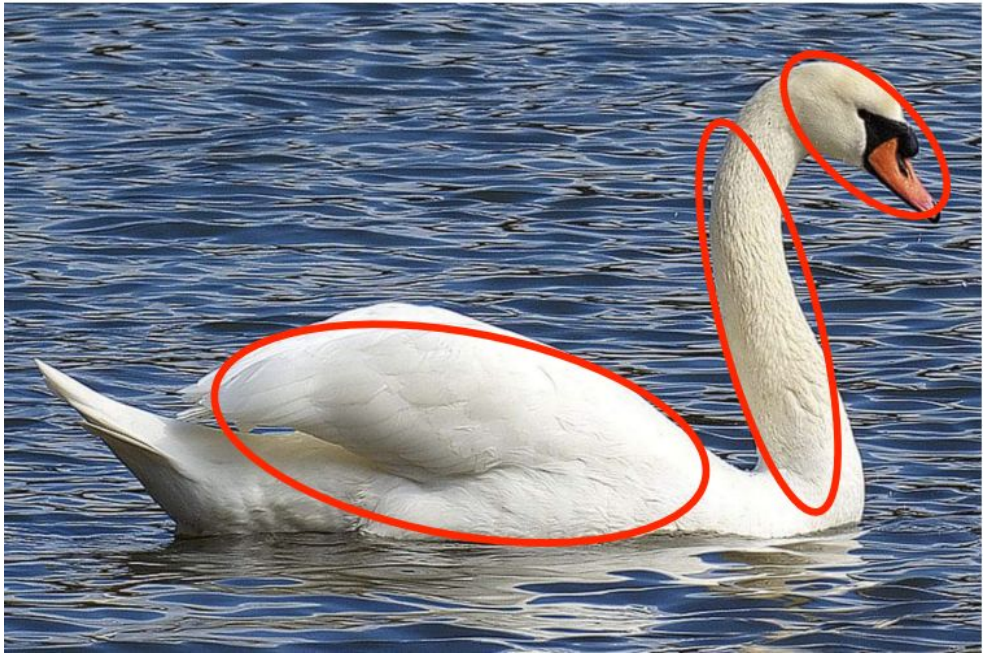
我們來看一張圖片

以下出處:葉倚任老師的深度學習簡報

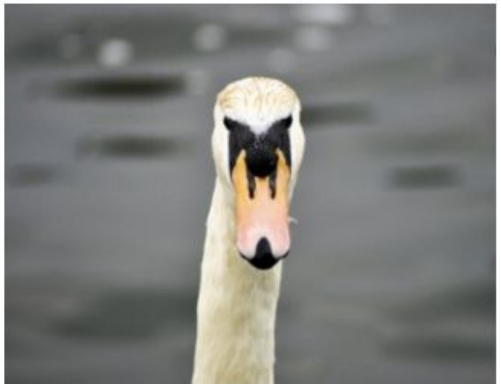
我們如何判別這是一隻鵝？

通常是:

1. **紅色的嘴巴**
2. **細長的脖子**
3. **白色的羽毛**

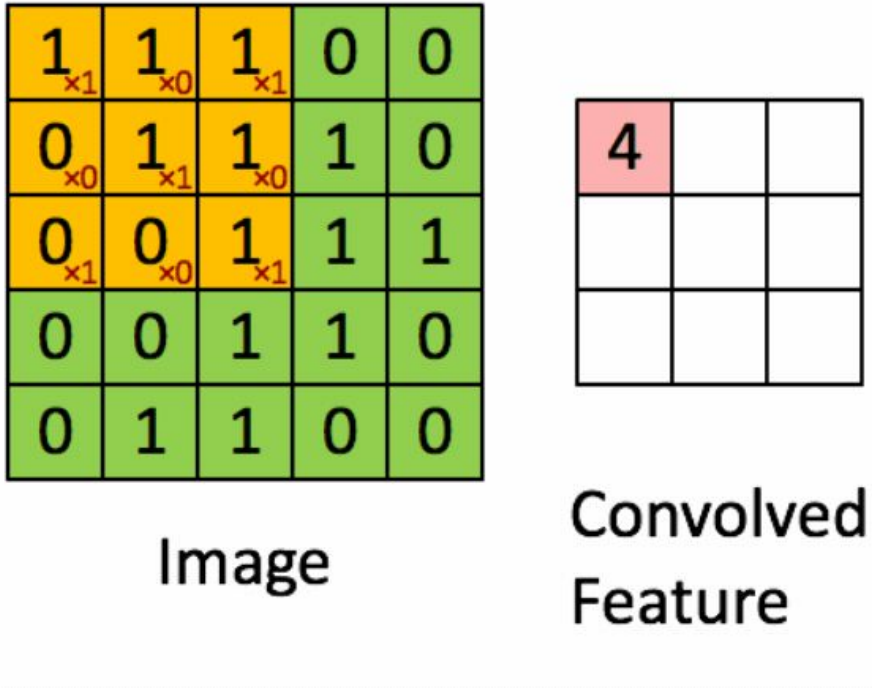


但是，如果是這樣，我們要怎麼辨別他是鵝？



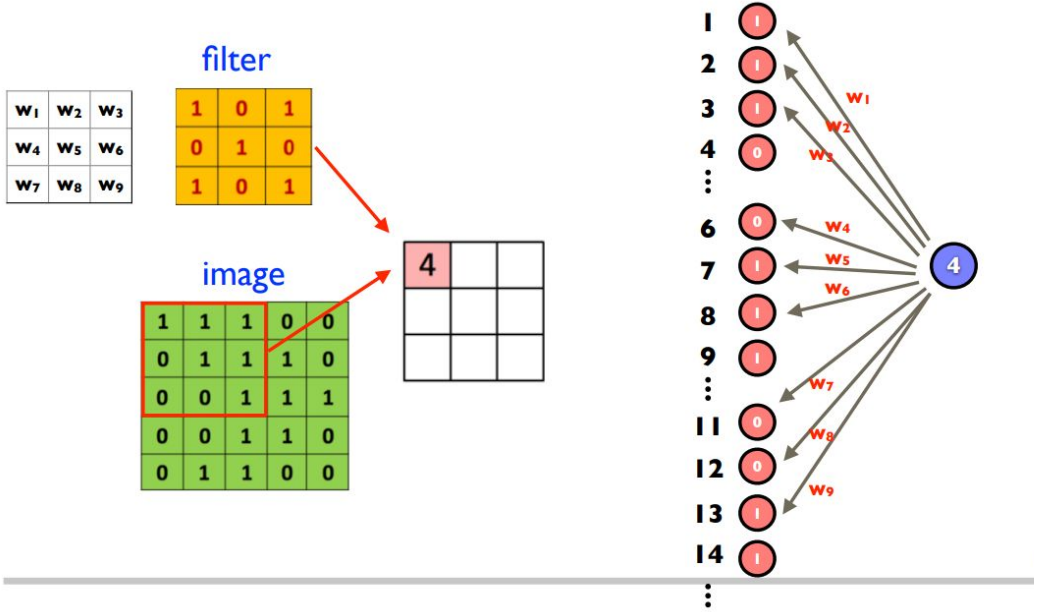
為了抓取圖像特徵, 我們先 recall 前面的東西

現在橘色的九宮格我們稱為過濾器 (filter)，它會對於圖片的像素做卷積, 算出來的數字會記錄在右邊的卷積特徵圖片。



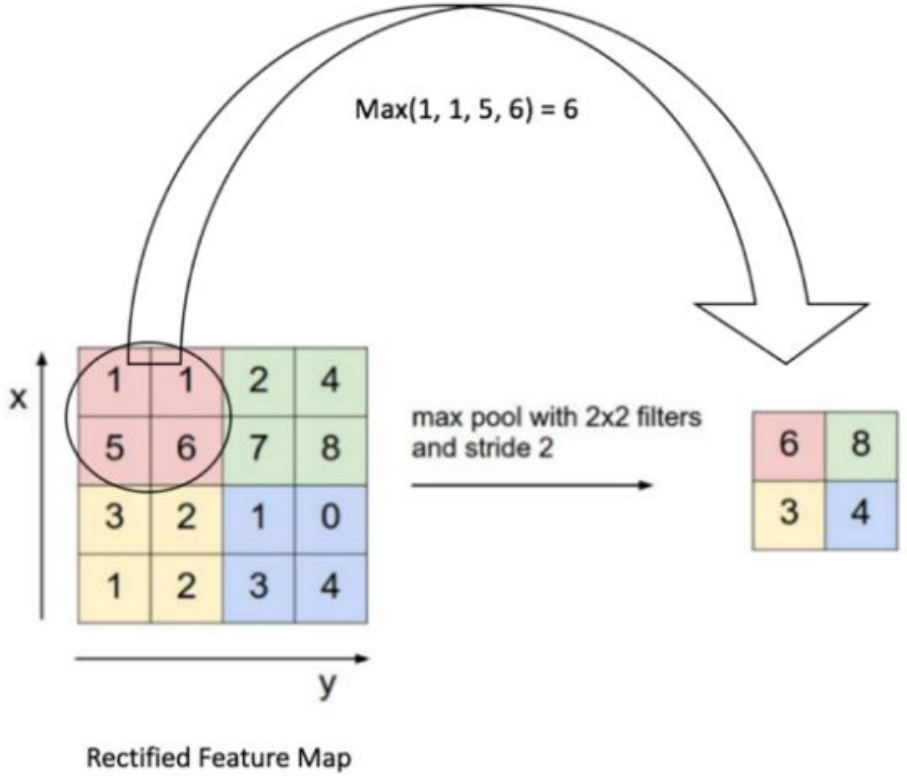
為了抓取圖像特徵，我們使用卷積神經網路(CNN)

過濾器中的數字我們稱為權重，不同權重對於找圖片特徵有不同效果，右圖的 4 就是將 5x5 的 image 拉成 25x1 的向量，再與權重內積算出數字。



池化層 (Pooling)

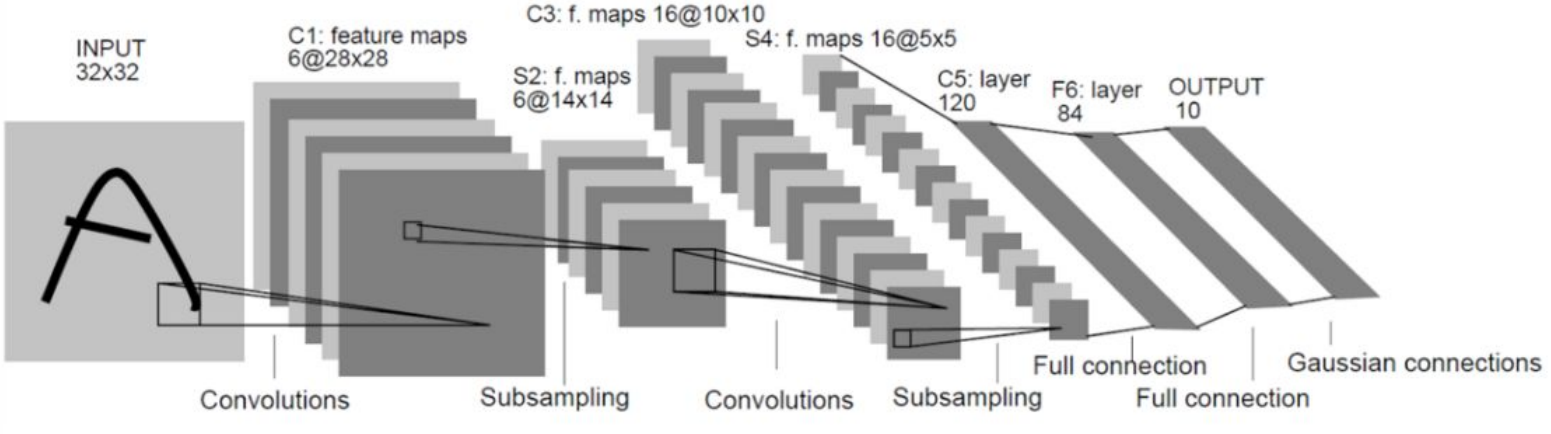
透過過濾器所得到的特徵圖維度很大，所以我們需要池化層來降維度。其中以最大池化層（Max pooling）最為常見。

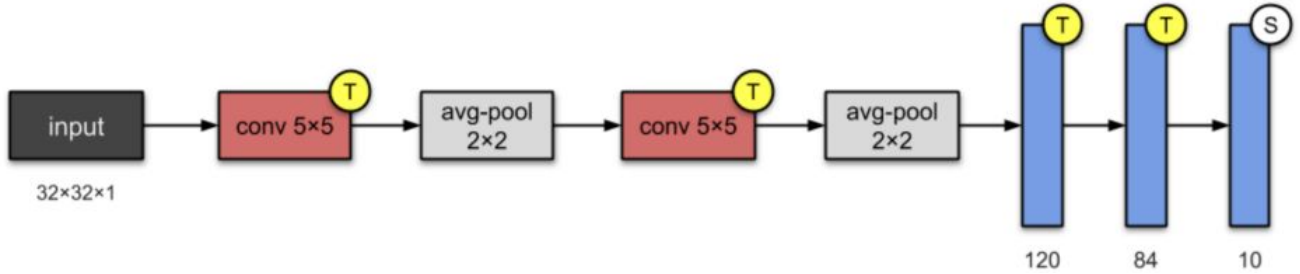


CNN的基本架構

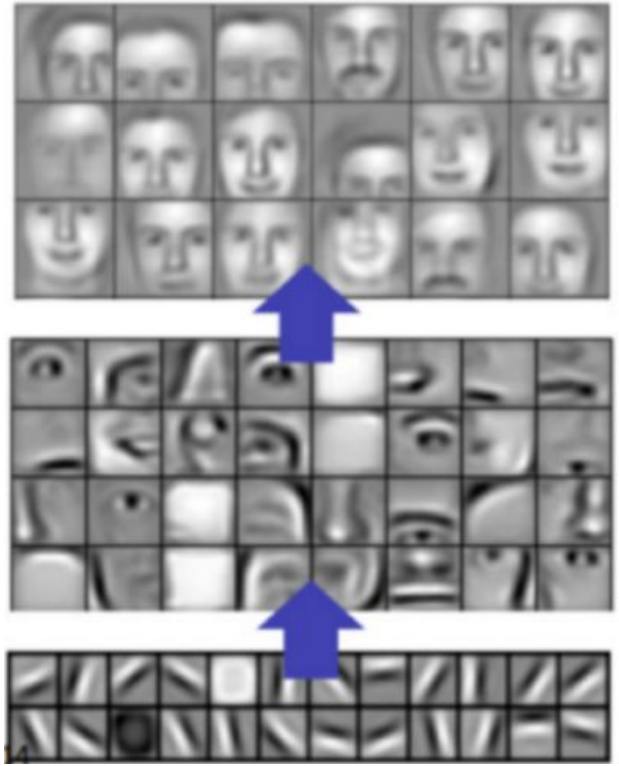
5x5過濾器會使其少4x4

2x2池化層會使圖片大小變成 1/4





從簡單到複雜的特徵形狀



Layer 3 Learn the filters to detect high level features

Layer 2 Learn the filters to detect simple shapes

Layer 1 Learn the filters to detect edges

實際應用：Alpha Go

19x19 matrix
(similar to an image)

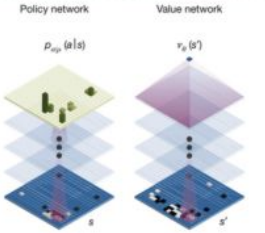
black: 1
white: -1
none: 0

CNN

Next move
(19x19 locations)

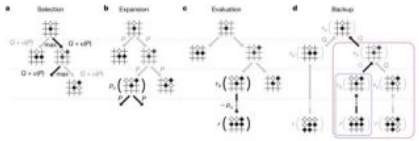
AlphaGo

卷積神經網路
Convolution Neural Network



+

蒙地卡羅搜尋樹
Monte Carlo Search Tree



總結

數學影像處理是將圖像上所遇到的需求和問題，嘗試用數學的方法去解決（其中包含了微積分、線性代數、機率與統計等），方法若可行還必須證明其正確性，接著再將數學的邏輯轉化為程式語言，方能解決。

為此，大家要好好學數學系的學科，才能找到方法來處理影像上的問題，還能昇華數學思維與解題方法運用在不同領域（例如機器學習）上，為他們找到更好、更快、更穩定的方法處理他們的問題。

總之，好好讀數學，不要被當掉。