

計算機視覺基礎

電腦感知世界的方式

人



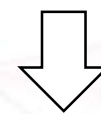
眼睛



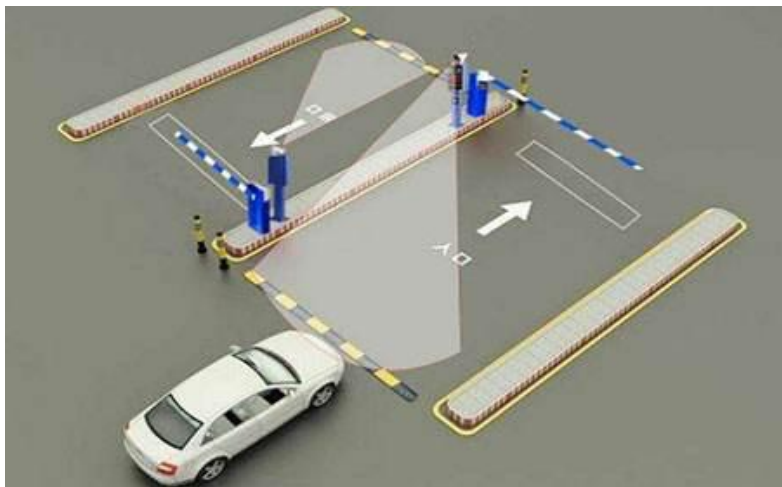
電腦

?

電腦視覺技術



以圖像為原料



圖：車牌號識別



圖：自動身份驗證



圖：自動駕駛的無人車

影像處理技術的發展大大改變了人們的**工作、娛樂、生活**

SenseStudy課程平臺 “人工智能入門（上）”

實驗4 – 1 圖像資料基礎

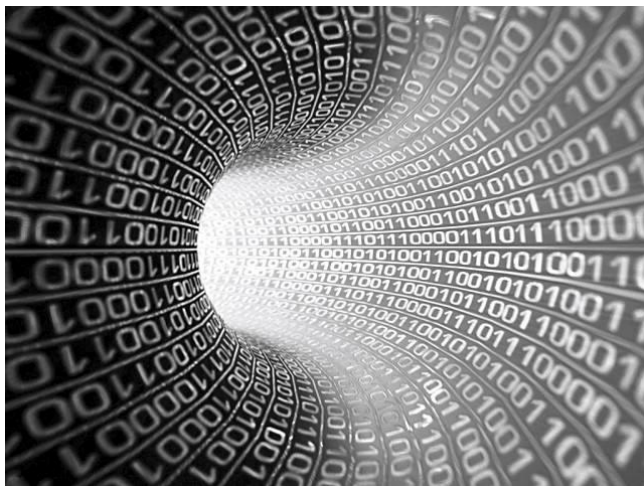
使用命令 `fig() + image(img_gray, cmap= 'gray')` 顯示一幅圖像





Q：利用電子成像設備獲取的圖像可以被電腦直接利用嗎？

答：不能。

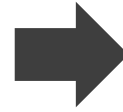
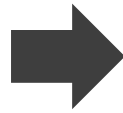
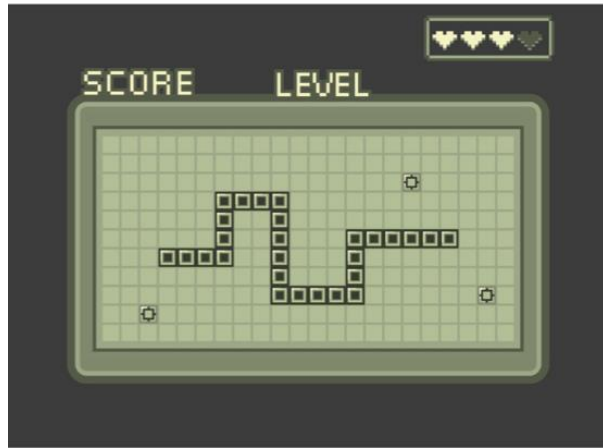


電腦只認識一組一組的資料。

電腦理解和處理圖像資料

要對圖像定義一個統一的資料結構，以及其使用方式

像素與二維陣列



Q：這三個遊戲介面的變化主要體現在哪些方面？

更精美、更清楚、色彩更豐富.....

Q：這三個遊戲介面的表示方式變化了嗎？

數字圖像的表示方式都由**像素**組成的**二維陣列**。

像素與二維陣列——以黑白圖像為例

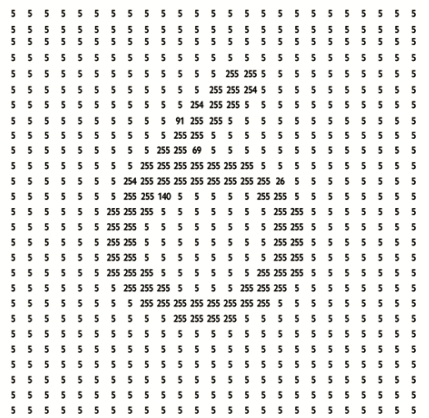
像素

- 數字圖片最**小**的組成單位
- 像素值代表像素的亮度
- 像素值範圍： 0 ~ 255 的整數



圖：一張圖片

- 把像素點理解為一個個小燈泡，把像素值理解為燈泡的旋鈕。
- 0 表示關掉這盞燈，一片漆黑，所以是黑色；
- 255表示把這盞燈旋鈕調到最亮，也就是白色。



圖：該圖片的像素圖

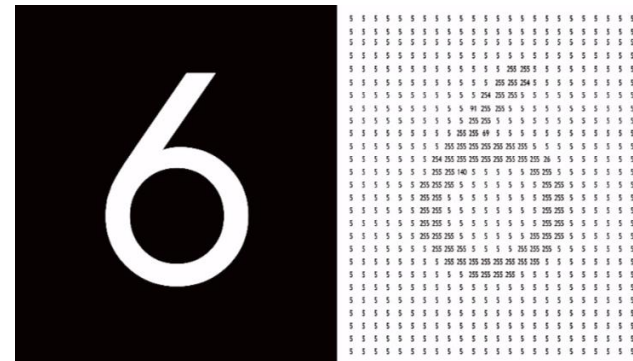
像素與二維陣列——以黑白圖像為例

圖像的表示 —— 像素組成的二維陣列

(陣列，是有序的元素序列。)



- 像素組成的二維陣列 類似 儲物櫃
- 圖像上存儲和修改像素點 類似 儲物櫃中存放和拿取物品



圖：圖片與其像素圖



圖：儲物櫃

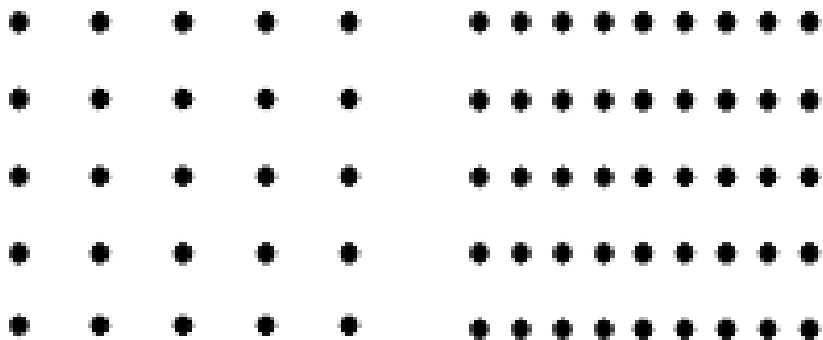
像素與二維陣列——以黑白圖像為例

觀察兩個字母 “a” ， 說說有什麼區別？



圖：鋸齒與像素

觀察下方形狀和大小相同的兩個圖形，每一個點表示一個像素。



Q：為什麼左邊的圖像中物體會有鋸齒狀的邊緣？

鋸齒的形成原因與**像素個數**有關，像素是圖片的最小單位，不能進一步分割，一張圖片由多個像素點排列而成。

因此，把這些像素點連接成物體邊框的時候，就會出現鋸齒。**螢幕像素密度越低，其顯示的物體輪廓越鋸齒化。**

Q：像素點的個數與圖像的清晰度是什麼關係？

SenseStudy課程平臺 “人工智能入門（上）”

實驗4 – 1 圖像資料基礎

```
print(img_gray.shape)
```

```
printMat2D(img_gray)
```



確定像素的多少：

一幅通過數碼相機獲取的圖像，它的像素數量是通過數碼相機的**解析度**確定的。



Q：哪個解析度高？

解析度：

- 定義：顯示器所能表示的像素個數。像素越密，解析度越高，圖像越清晰。
- 表示：“列數*行數”

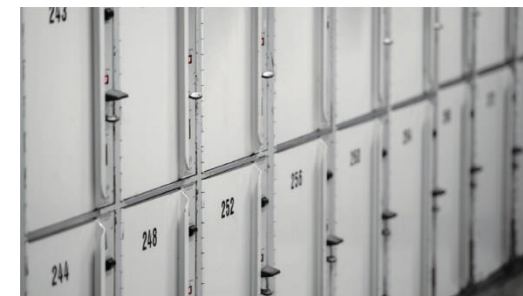
↙
沿水準方向分成的
像素列的個數

↘
沿垂直方向分成的
像素行的個數



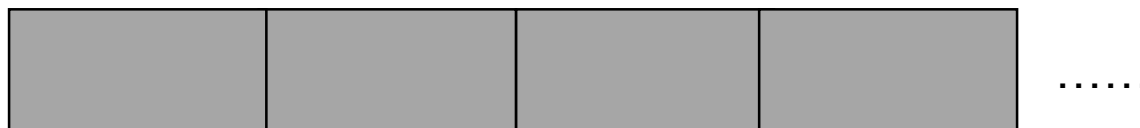
解析度:1920*1080

- 像素組成的二維陣列 類似 儲物櫃
- 圖像上存儲和修改像素點 類似 儲物櫃中存放和拿取物品



圖：儲物櫃

陣列：一種特殊的變量



n個存儲單元



存儲n個數據

變量



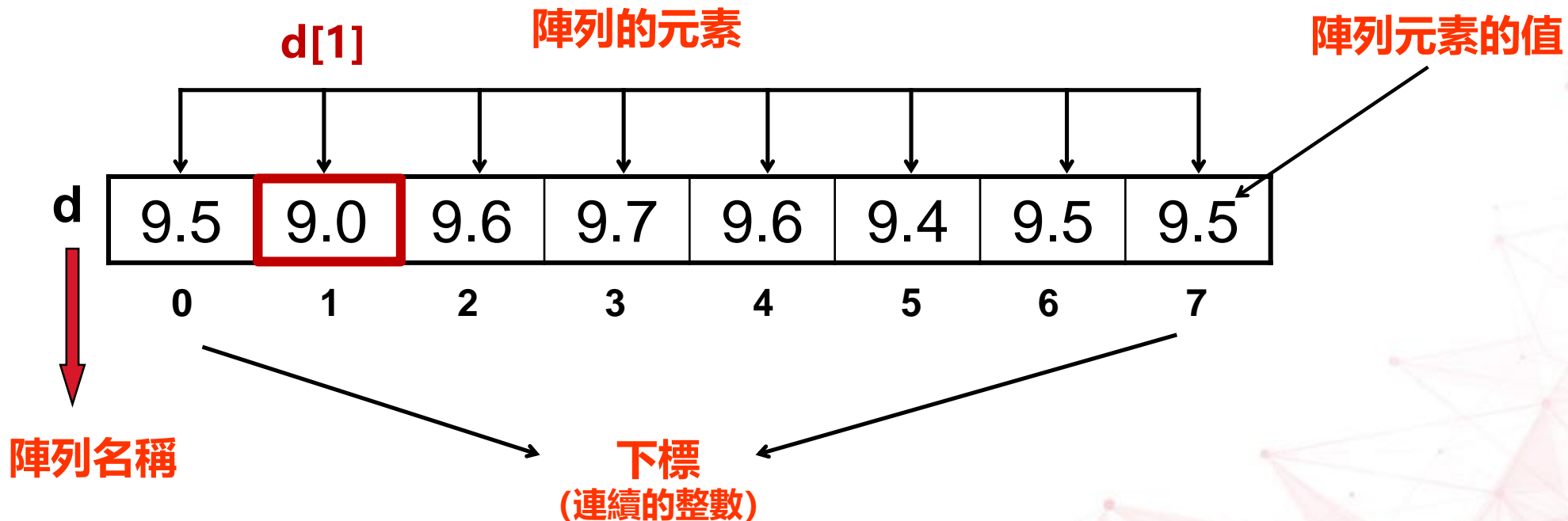
一個存儲單元



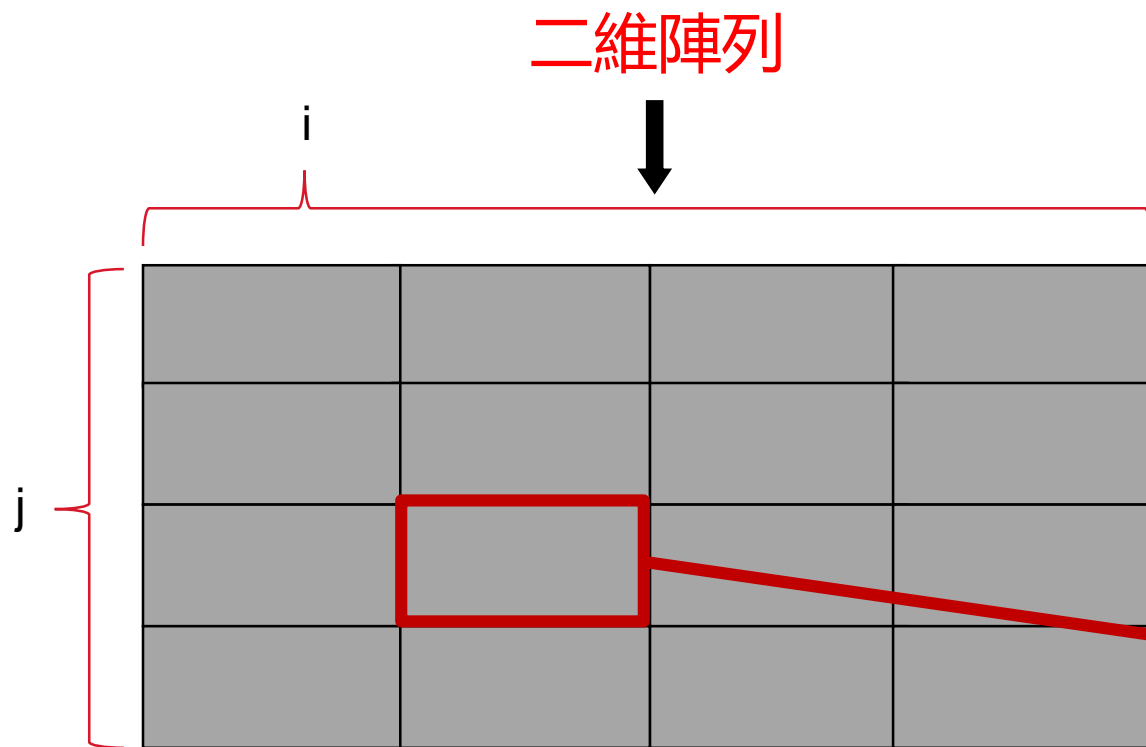
存儲一個資料

陣列就是把 **有限個 類型相同** 的變數用同一個名字命名，然後用編號區分每一個變數。
如下所示：

- **d** 稱為陣列名稱 **編號**稱為下標。
- 組成陣列的各個變數稱為陣列的元素。
- 某個陣列元素用 **陣列名稱[下標]** 表示（從0開始）



二維陣列

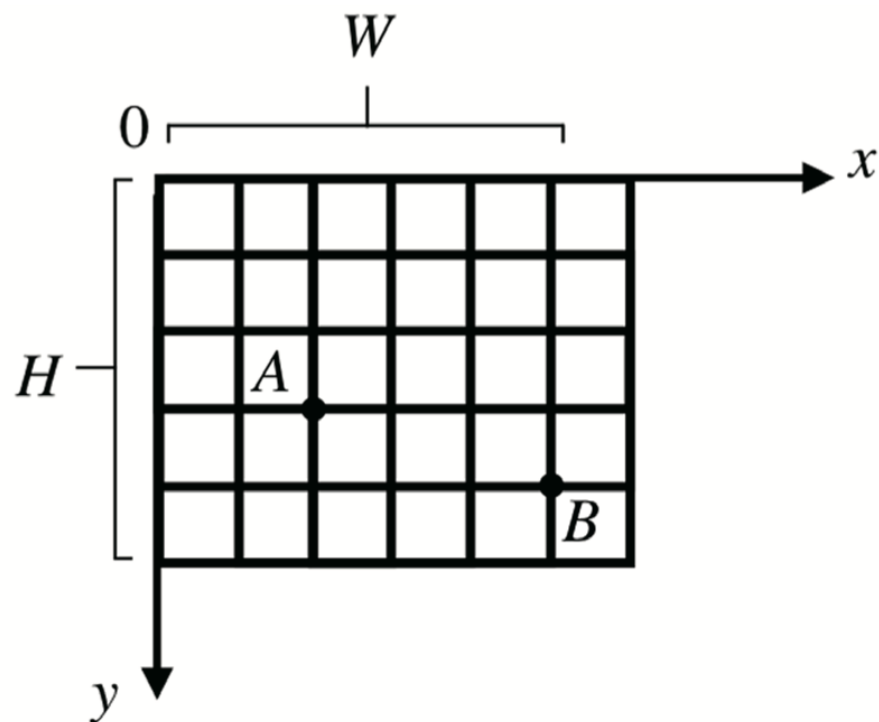


若用 `img` 表示一個數字圖像，
那麼 `img[i,j]` 就是這個圖像中第從上往下數第 $i+1$ 行，
從左往右數第 $j+1$ 列的像素。

$i*j$ 個存儲單元

存儲 $i*j$ 個數據

在程式設計的時候，
我們可以通過設定陳述式 `pix=img[i,j]`
把獲取的像素存在變數 `pix` 裡。



為了電腦讀取資料準確，方便，我們建立如左圖的坐標系。起始位置為左上角 $(0,0)$ 點，表示圖片的左上角。

訪問圖片的某個像素就是查看二維陣列的某一個元素。

Q: A 點和 B 點分別表示的像素點對應圖片裡的什麼？

SenseStudy課程平臺 “人工智能入門（上）”

實驗4 – 1 圖像資料基礎


```
img_gray_copy = image_copy(img_gray)
img_gray_copy[18,16] = 255
fig() + image(img_gray_copy, cmap='gray' )
img_gray_copy = image_copy(img_gray)
img_gray_copy[18:25,16] = 255
fig() + image(img_gray_copy, cmap='gray' )
img_gray_copy = image_copy(img_gray)
img_gray_copy[18:25,16:25] = 255
fig() + image(img_gray_copy, cmap='gray' )
img_gray_copy = image_copy(img_gray)
img_gray_copy[10:,:] = 255
fig() + image(img_gray_copy, cmap='gray')
```



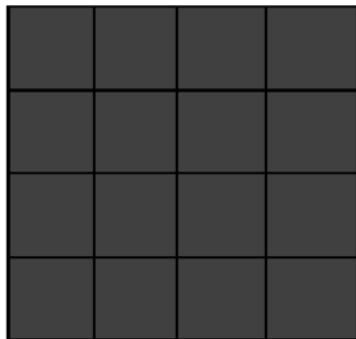
訪問 圖片的 某個像素 —— 查看 二維陣列的 對應元素

修改 圖片的 某個像素 —— 修改 二維陣列的 對應元素

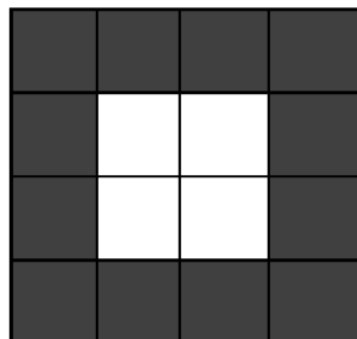
修改 圖片像素點的亮度

在 Python 裡面，`img[i,j]=50` 可以把圖像中行列號為 (i,j) 的像素值設定為50。

Q：在一張寬 4 個像素，高 4 個像素的全黑圖片中心畫一個 2x2 的白色正方形，
我們需要怎樣修改二維陣列？



圖：修改前



圖：修改後

二維陣列只能表示黑白圖片，彩色圖像如何表示？

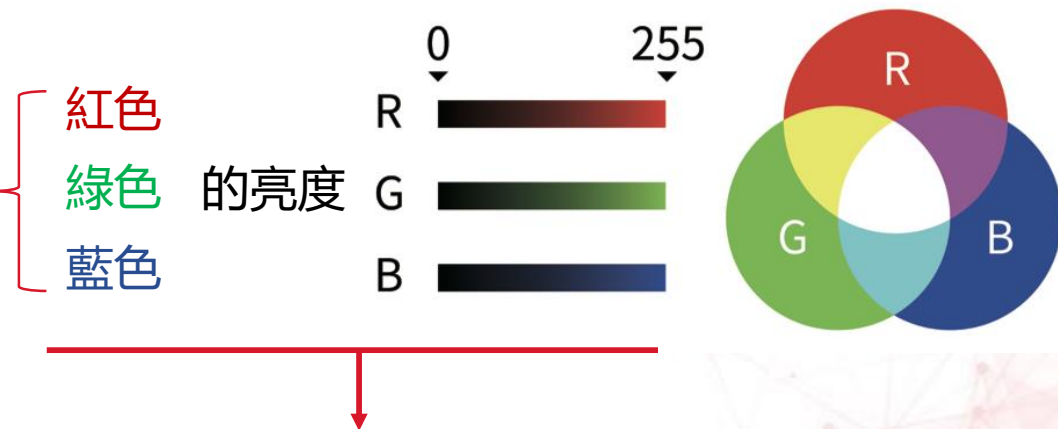
顏料 的三原色：紅黃藍

光 的三原色：紅綠藍（RGB） 色彩的三通道

黑白圖片，每個像素點有 1 個像素值，表示該像素點的亮度。



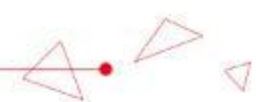
彩色圖片，每個像素點有 3 個像素值，分別代表該像素點的



Q:RGB 模式顏色總數？

RGB 模式顏色總數： $256 \times 256 \times 256$ 種 = 16777216。

紅色有 256 種不同亮度的紅
綠色有 256 種不同亮度的綠
藍色有 256 種不同亮度的藍



若分別單獨增加R、G、B通道的亮度，圖像會怎麼變化？

當 $R=0$ ， $G=0$ ， $B=0$ 或 $R=255$ ， $G=255$ ， $B=255$ 時，會顯示什麼顏色？

若想顯示紅色，RGB值應該分別為多少？

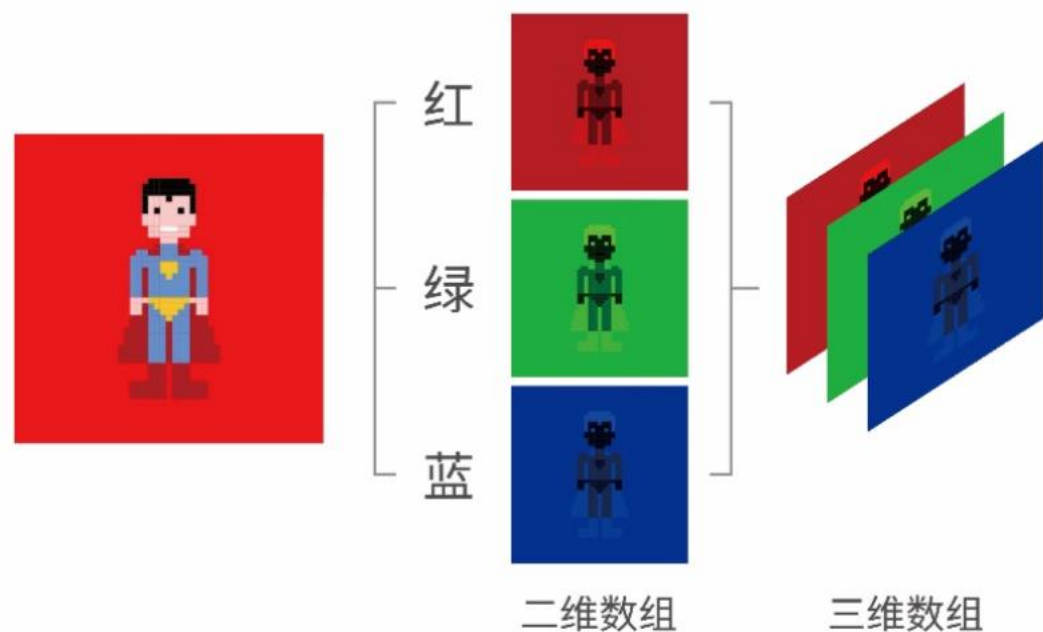
怎樣調整RGB值才能得到下圖的結果呢？



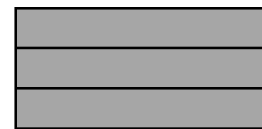
三維陣列表示彩色圖片

Q：電腦如何存儲 RGB 圖的呢？

定義彩色圖片的資料結構需要 3 個二維陣列，分別對應 R、G、B 三個通道的像素亮度。
因此，我們將 3 個二維陣列疊起來，組成一個立體的、長方體的“三維陣列”，如下圖所示。



彩色圖片相當於給每個“格子”分了三個檔，
每個檔代表 RGB 三通道中的一個。



因此，讀取的時候，除了明確哪一行哪一列之外，
還需要說明是第幾檔。

SenseStudy課程平臺 “人工智能入門（上）”

實驗4 – 1 圖像資料基礎

```
fig() + image(img_shibe)
```

```
print(img_shibe.shape)
```

```
r=img_shibe[:, :, 0]
```

```
g=img_shibe[:, :, 1]
```

```
b=img_shibe[:, :, 2]
```

```
fig(1,3)+[image(r, cmap='Reds'), image(g, cmap='Greens'),  
image(b, cmap='Blues')]
```



SenseStudy課程平臺 “人工智能入門（上）”

實驗4 – 2 影像處理技術-色彩變換

```
fig() + image(img_raw)
img_bgr = image_copy(img_raw)
tmp = image_copy(img_bgr[:, :, 0])
img_bgr[:, :, 0] = img_bgr[:, :, 2]
img_bgr[:, :, 2] = tmp
fig() + image(img_bgr)
```



SenseStudy實驗 – 圖像風格變換

SenseStudy課程平臺 “人工智能入門（上）”：實驗4 – 2 影像處理技術-色彩變換

```
url = 'https://raw.githubusercontent.com/yzhang0301/codes/master/chow.jpg'
```

```
img_raw = imread(url)
```

```
fig() + image(img_raw)
```

```
mat = [
```

```
    [0.393, 0.769, 0.189],
```

```
    [0.349, 0.686, 0.168],
```

```
    [0.272, 0.534, 0.131]
```

```
]
```

```
img_retro = image_copy(img_raw)
```

```
img_retro = image_int2float(img_retro)
```

```
img_retro = map_color_space(img_retro, mat)
```

```
img_retro = bound(img_retro, 0, 255)
```

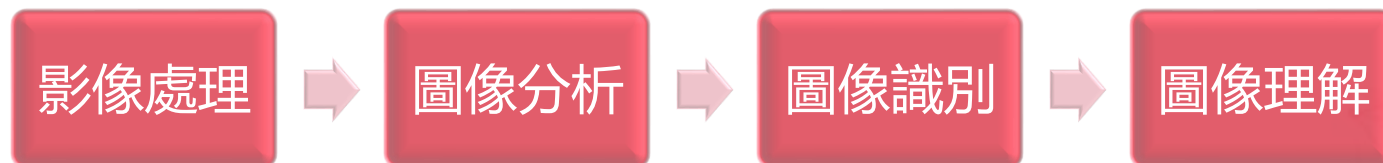
```
img_retro = image_float2int(img_retro)
```

```
fig() + image(img_retro)
```



Q:下圖中是什麼?

Q:人能很輕鬆地認出這是貓，但任何圖像資訊進入電腦之後都是一堆雜亂的像素點，電腦如何確定這是一隻貓呢?



影像處理是指對圖像進行分析、加工和處理，從而使其達到某種要求或者實現某種效果的技術。

常用的**影像處理技術**：包括幾何變換、顏色調節、去噪音、圖像融合、圖像分割等。

影像處理操作：針對儲存彩色圖像的三維陣列進行。



下圖是一個表情包製作的例子。而絕大多數動態表情包製作，運用到的正是**圖像拉伸技術**。
圖像拉伸的效果類似捏橡皮泥或者捏臉一樣。



- 圖像拉伸技術需定義3個控制變數：

}	start	起點
	offset	位移
	radius	半徑

圖像變換之拉伸

Q: 圖1是一個人臉的正面照，如果能把這個人的鼻尖往下方捏，該怎麼實現這個效果呢？

- 原圖圖片的大小為 400×400 ，鼻子的位置為 $(240, 200)$ ，
- 假設我們拉伸距離為 40，影響範圍與拉伸距離一致。

start: $(240, 200)$ ，指地點位置 $(240, 200)$

offset: $(40, 0)$ ，指往下拉伸 40，左右不變

radius: 40，指拉伸半徑為 40



圖1



圖2

Q: 圖2可以通過下拉鼻子得到。請思考，我們能否反過來把鼻子往上拉同樣的距離，來將這張圖片還原成上圖左？或者說，拉伸變換是否可逆？

SenseStudy課程平臺 “人工智能入門（上）”

實驗4 – 4 表情包1 – 圖像扭曲

```
print(img_girl.shape)
```

```
warp_params = [(250,300), (0,20), 40]
```

```
img_warp = local_warp_image(img_girl, warp_params)
```

```
fig(1,2) + [image(img_girl),image(img_warp)]
```



 生活中最常用的影像處理就是P人像了，下一節我們要完成一個製作表情包的小項目。

- 製作人臉表情包的重要技術：“人臉關鍵點檢測技術”



作用：提取出一張人臉圖片中**關鍵部位**的點的位置，
去除與面部的資訊無關的**冗餘信息**，如身體、毛髮.....

技術分類：人臉關鍵點技術可按照使用的關鍵點的數量分類
如本章中使用的人臉關鍵點技術就是 **68.關鍵點技術**

製作動態表情包

人的表情是由許多部位表現出來的，
最重要的莫過於**眉毛**、**眼睛** 和**嘴巴**。

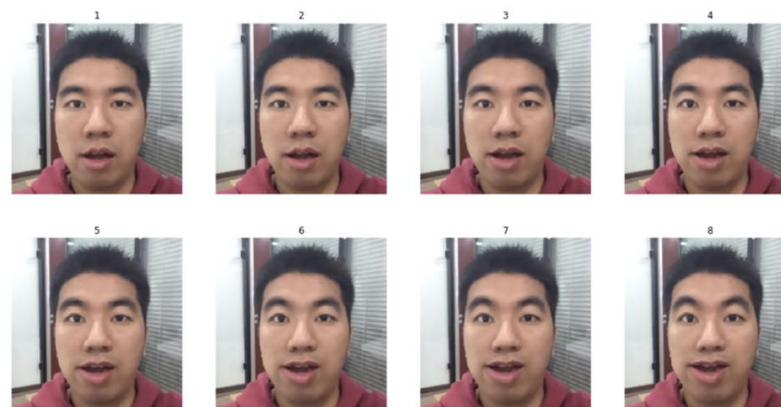
通過部位的拉伸，
就可以改變圖中人的表情。



表情包的動態效果：

多張表情照片間如果不是一次拉伸到位，
而是**慢慢地移動**，就可以得到一張張**漸變**的表情圖，如右圖所示。

再把這些表情圖連接成一個 gif 動圖，就
可以得到最終的表情包了。





Q：通過之前的學習，你能總結出製作動態表情包的步驟嗎？

1. 定點：通過人臉關鍵點技術找到想要拉伸的點。
2. 拉伸：對每個點設置控制變數，運用拉伸函數進行逐步拉伸，形成一張張表情漸變圖。
3. 連接：將這些圖連成 gif 動圖。

SenseStudy課程平臺 “人工智能入門（上）”：實驗4 – 5 表情包2-人臉關鍵點檢測

```
url = 'https://raw.githubusercontent.com/yzhang0301/codes/master/chow.jpg'
img_wzj = imread(url)

points=detect_keypoints(img_wzj)
img_girl_with_points=draw_points(img_wzj,points)
fig()+image(img_wzj)
fig()+image(img_girl_with_points)

mouth_left = points[48]
mouth_right = points[54]
left_eyebrow = points[19]
right_eyebrow = points[24]
warp_params = [
    [left_eyebrow, (0, -5), 35],
    [right_eyebrow, (0, -5), 35],
    [mouth_left, (-5, -5), 40],
    [mouth_right, (5, -5), 40]
]
sticker = make_sticker(img_wzj, warp_params)
fig() + gif(sticker)
```



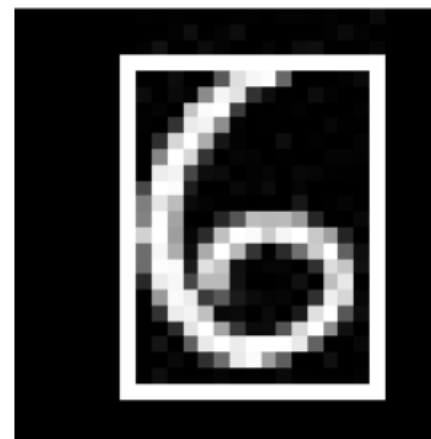
SenseStudy課程平臺 “人工智能入門（上）”

實驗4 – 1 圖像資料基礎



使用命令 `fig() + image(img_gray, cmap= 'gray')` 顯示一幅圖像

請為顯示的 “6” 字添加一個白色方框



SenseStudy課程平臺 “人工智能入門（上）”

實驗4 – 2 影像處理技術-色彩變換



使用命令 `fig() + image(img_wzj)` 顯示一幅頭像

請將這幅頭像變換為黑白木雕效果



SenseStudy課程平臺 “人工智能入門（上）”

實驗4 – 3 影像處理技術-空間變換

使用命令 `url = 'Your website link'`
`img_mine = imread(url)`
`fig() + image(img_mine)` 上傳并顯示一張自己喜歡的風景畫

使用實驗4 -3技術將此畫變換為油畫風格

