



多媒體程式設計

影像資料處理

Instructor: 馬豪尚

認識影像

- › 相片 (picture)
- › 圖像 (image)
- › 影片 (video)
- › 影片擷取出的畫面 (frame)

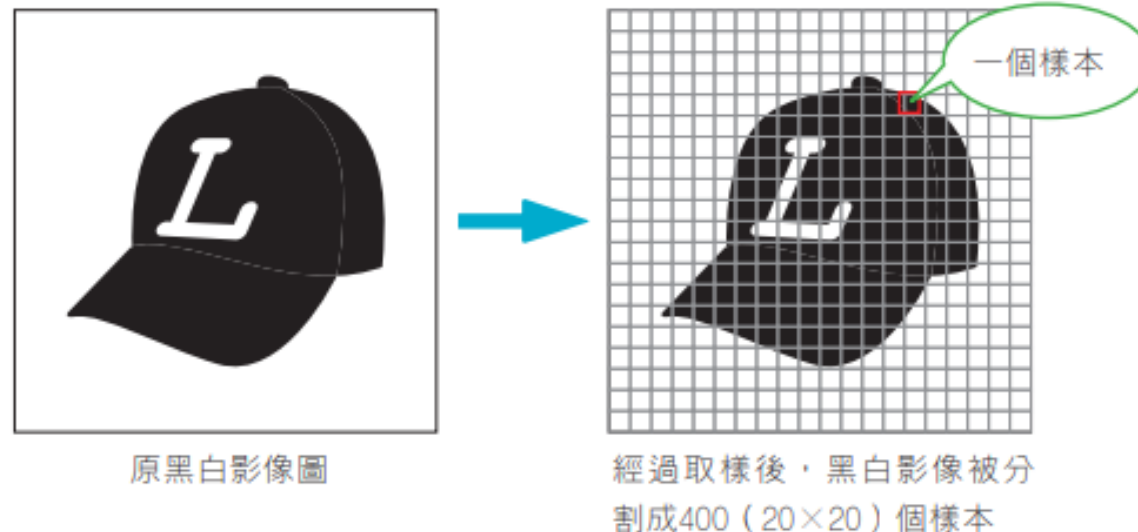
相片

- › 相片(picture)：連續色彩變化的圖畫(也許有顆粒，但裸眼看不出)，因此從數學的觀點來描述，相片是一個連續二維空間的亮度函數



圖像

- › 圖像(image)：將相片分割成一個個整齊排列的顆粒，再給予每一顆粒一個數值表示該顆粒的亮度；這樣的空間分割及亮度數值指定合稱為數位化(digitalization) 或離散化(discretization)。數位化後的相片就稱為圖像



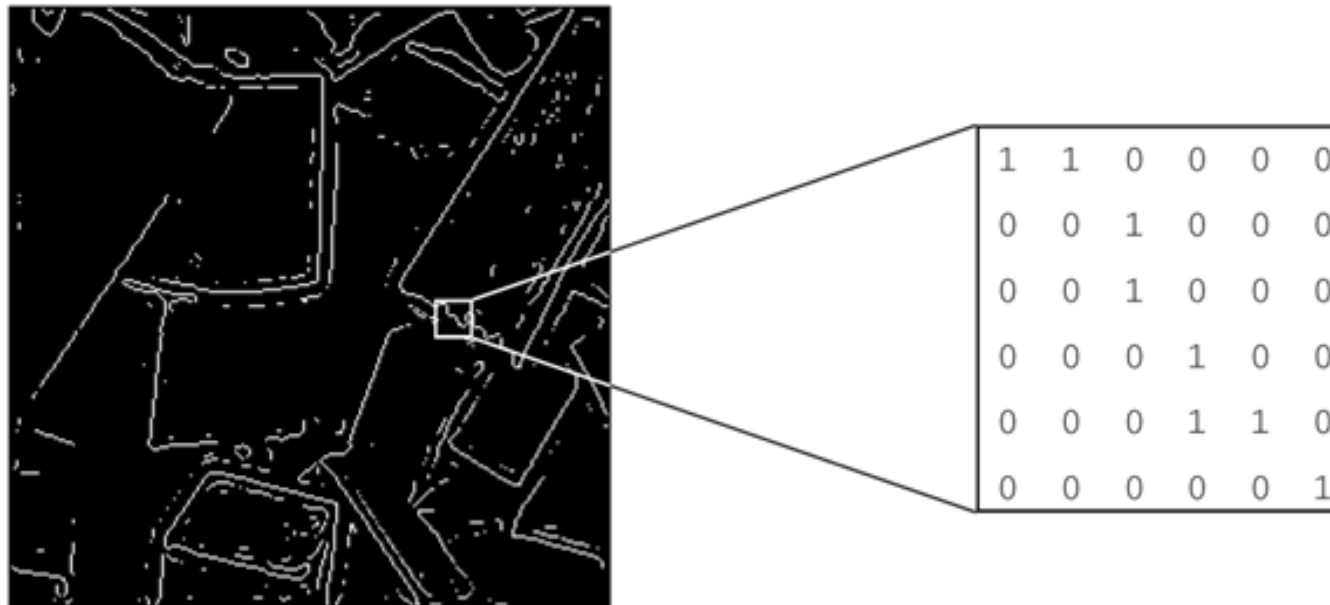
圖像

- › 二元圖像：圖像中每個像素的亮度值 (Intensity) 僅可以取自0或1的圖像，因此也稱為1-bit圖像。
- › 灰度圖像：也稱為灰階圖像：圖像中每個像素可以由0(黑)到255(白)的亮度值 (Intensity) 表示。0-255之間表示不同的灰度級。
- › 彩色圖像：RGB的彩色圖像是由三種不同顏色成分組合而成，一個為紅色，一個為綠色，另一個為藍色。



二元圖像

- › 圖像中每個像素的亮度值(Intensity)僅可以取自0或1的圖像，因此也稱為1-bit圖像。



灰度圖像

- › 圖像中每個像素可以由0(黑)到255(白)的亮度值(Intensity)表示。



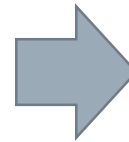
48	219	168	145	244	188	120	58
49	218	87	94	133	35	17	148
174	151	74	179	224	3	252	194
77	127	87	139	44	228	149	135
138	229	136	113	250	51	108	163
38	210	185	177	69	76	131	53
178	164	79	158	64	169	85	97
96	209	214	203	223	73	110	200

像素

3X5鄰域

彩色圖像

- › 由Red、Green、Blue三種顏色組成，每一種顏色的圖都是用0-255的數值來表示，最後合併成為一張彩色的圖像



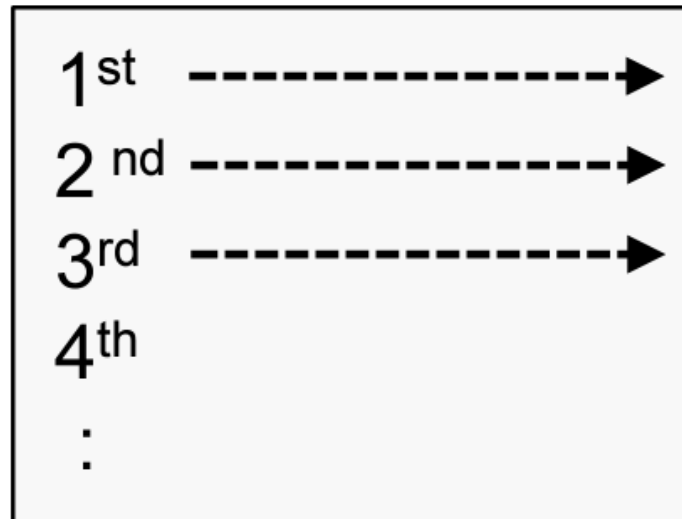
像素(Pixel) 和 解析度(Resolution)

- › 像素為組成數位影像的最小單位
- › 解析度為數位影像中的像素數量(例如:1920*1080 或 2,073,6000像素)



影像檔案格式(image file formats)

- › 一列一列的紀錄影像灰階或色彩值，又稱為以列為主(row major) 的紀錄方式



Example (a 5×6 image)

1	3	4	6	8	6
5	7	5
:					
:					

至少要記錄 [5; 6; 1, 3, 4, 6, 8, 6, 5, 7, 5, ...]

Python 基本影像處理

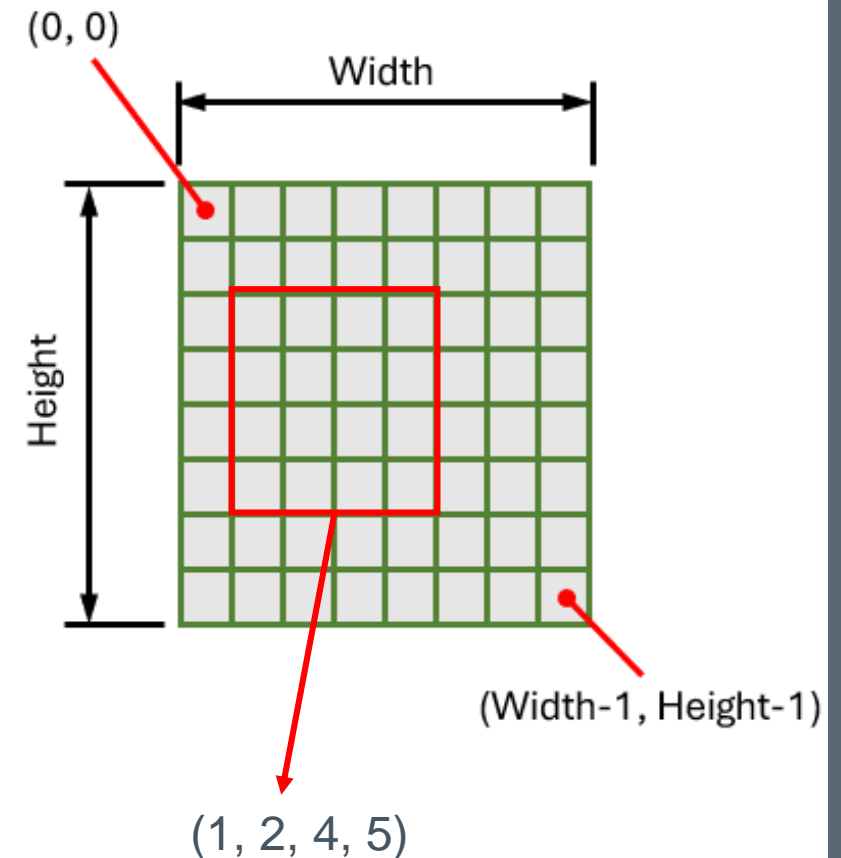
- › PIL(Python Imaging Library)是一套影像處理的模組，可以做到一些常見的影像處理和操作
 - 裁切、平移、旋轉、縮放
 - 調整亮度、色調，套用濾鏡
- › PIL早在Python 2.7後就斷更，Pillow為基於PIL開發支援Python3以上版本的套件

Python 基本影像處理

- › Pillow模組安裝與載入
 - pip install pillow
 - From PIL import Image
- › Pillow模組中的RGBA 分別代表紅色、綠色、藍色、透明度
 - From PIL import ImageColor
 - ImageColor.getrgb(color)
 - › 返回(r, g, b)的色彩元組
 - ImageColor.getcolor(color, "RGB")
 - › 返回(r, g, b)的色彩元組
 - ImageColor.getcolor(color, "RGBA")
 - › 返回(r, g, b, a)的色彩元組

Pillow模組的影像座標

- › 最左上角的像素座標(x,y)是(0,0)，x軸像素往右遞增，y軸像素向下遞增
- › 圖片內的選取box的座標可以用(left, top, right, bottom)四個位置的座標來標示
 - left: box左上角的x軸座標
 - top: box左上角的y軸座標
 - right: box右下角的x軸座標
 - bottom: box右下角的y軸座標



Pillow模組影像物件

- › 開啟影像物件
 - `Imageobj = Image.open("圖像檔")`
- › 獲得影像大小屬性
 - `Imageobj.size`
 - › 返回影像寬度和高度
- › 取得原始圖檔名稱
 - `Imageobj.filename`
 - › 返回影像物件的檔案名稱

Pillow模組影像物件

- › 取得影像物件的格式
 - Imageobj.format
 - › 返回檔案副檔名
 - Imageobj.format_description
 - › 返回副檔名以及物件描述
- › 在螢幕顯示影像
 - Imageobj.show()

Pillow建立新的影像物件

- › 使用new(模式, 影像大小, 顏色), 建立新的影像物件
 - `Imageobj.new("RGB", (100, 100), "ff0000")`
- › 存檔
 - `Imageobj.save("檔名")`

Pillow基本影像編輯

› 更改影像大小

– Imageobj.resize((width, height), resample)

- › 第一個參數為size: (寬度, 高度)
- › 第二個參數為resample取樣方法參數

resample 參數	說明
Image.NEAREST	最鄰近插值法
Image.BOX	類似NEAREST
Image.BILINEAR	雙線取樣法
Image.HAMMING	產生比 BILINEAR 更清晰的圖像
Image.BICUBIC	三次方取樣法
Image.LANCZOS	高品質LANCZOS濾波器

Pillow基本影像編輯

› 影像旋轉

- `Imageobj.rotate(旋轉角度, expand=True)`
 - › 旋轉都是以逆時針旋轉
 - › `Expand`是設定是否顯示超出影像大小的部分，並以黑色填滿

› 影像翻轉

- `Imageobj.transpose(Image.FLIP_LEFT_RIGHT)`
 - › 影像左右翻轉
- `Imageobj.transpose(Image.FLIP_TOP_BOTTOM)`
 - › 影像上下翻轉

Pillow影像像素編輯

- › 取得影像中某一個像素的顏色
 - `Imageobj.getpixel((x, y))`
 - › (x, y) 為指定像素在影像內的座標
- › 修改或填入色彩到某一個指定像素
 - `Imageobj.putpixel((x, y), (r, g, b, a))`
 - › (x, y) 為指定像素在影像內的座標
 - › (r, g, b, a) 為顏色值

Pillow影像裁切、複製和合成

- › `Imageobj.crop`(裁切區間)
 - 裁切區間為(left, top, right, bottom)
- › `Imageobj.copy`()
 - 複製影像
- › `Imageobj.paste`(影像, (座標位置))
 - 影像是要貼上的影像物件
 - 座標位置為指定要放的開始座標位置(x, y)

Pillow影像濾鏡(效果)

› Imageobj.filter(濾鏡參數)

濾鏡參數 ImageFilter.	效果說明
BLUR	模糊效果
CONTOUR	描繪輪廓
DETAIL	強調細節
EDGE_ENHANCE	邊界增強
EDGE_ENHANCE_MORE	加強邊界增強
EMBOSS	浮雕效果
FIND_EDGES	尋找邊界
SMOOTH	平滑效果
SMOOTH_MORE	深度平滑效果

在影像內繪製圖案

- › Pillow模組內有另一個ImageDraw模組提供繪圖
 - 可以繪製點、線、矩形、橢圓、多邊形
- › 載入模組
 - `From PIL import ImageDraw`
- › 建立圖案繪製物件
 - `Drawobj = ImageDraw.Draw(Imageobj)`
 - › Imageobj為圖像物件，可以用自己建立的圖像或讀取的圖像

在影像內繪製圖案

› 繪製點

- Drawobj.point([(x1,y1), (x2, y2)...], fill)
 - › (x1, y1)為繪製點在影像內的像素座標，用列表包起來
 - › fill=color，為顏色參數

› 繪製線條

- Drawobj.line([(x1,y1), (x2, y2)...], width, fill)
 - › (x1, y1)為繪製點在影像內的像素座標，如果有兩個以上的點就會連起來
 - › width=線條寬度，預設為1
 - › fill=color，為顏色參數

在影像內繪製圖案

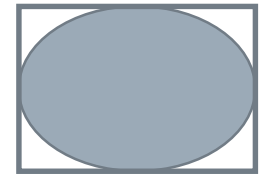
› 繪製圓或橢圓

– Draw.ellipse((left, top, right, bottom), fill, outline)

› (left, top, right, bottom)元組是包住圓形或橢圓形的矩形左上角與右下角座標

› fill=color , 為填滿顏色

› outline=color , 為外框顏色



right, bottom

› 繪製矩形

– Draw.rectangle((left, top, right, bottom), fill, outline)

› (left, top, right, bottom)元組是矩形左上角與右下角座標

› fill=color , 為填滿顏色

› outline=color , 為外框顏色

在影像內繪製圖案

› 繪製多邊形

– Drawobj.polyon $[(x1, y1), (x2, y2) \dots], \text{width}, \text{fill}$

- › $(x1, y1)$ 為繪製點在影像內的像素座標，如果有兩個以上的點就會連起來
- › width = 線條寬度，預設為1
- › fill = color，為顏色參數

› 填寫文字

– Drawobj.text $((x, y), \text{text}, \text{fill}, \text{font})$

- › (x, y) 為文字在影像內的開始座標
- › text 為想填入的文字
- › fill = color 為文字顏色
- › font = 字型

練習

- › 讀取image資料夾內的30張image
- › 將所有image的size調整為512*512
- › 將nutc logo加上矩形框線並加到每張圖片的最右上角
- › 儲存處理過後的影像