國立臺灣科技大學
NATIONAL TAIWAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

實習五
影像特徵的研究

課程大綱

- 實習00: Colab 環境
- 實習05: 影像特徵的研究

實習 00
Colab 環境

Colab Env.

Colab Env.

Before we start...

```python
#mount drive
from google.colab import drive
drive.mount('/content/drive')
```

```python
# import libraries
import sys
import os
import cv2
import numpy as np
from matplotlib import pyplot as plt
from google.colab.patches import cv2_imshow
```

實習 05
影像特徵的研究

Function

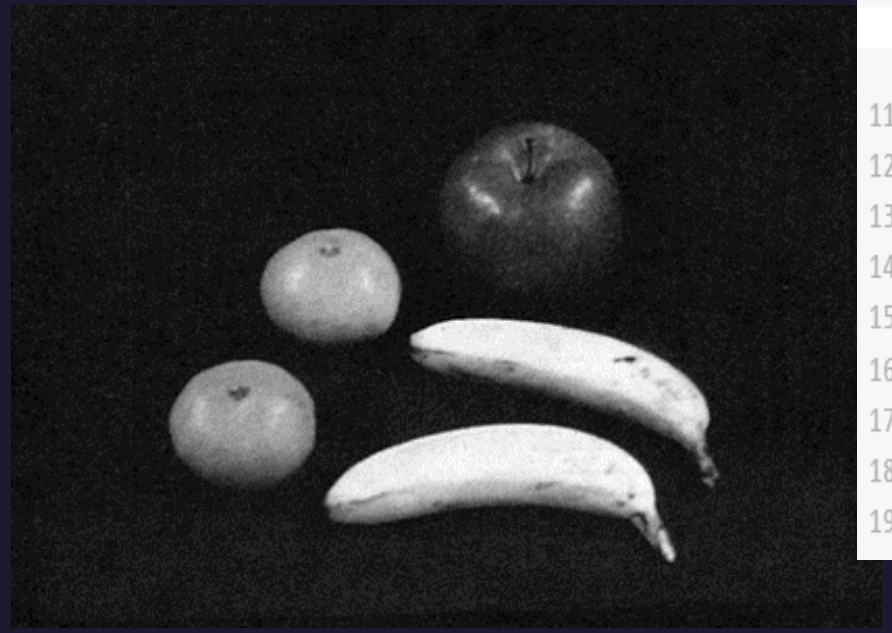
- 影像二值化
  - OpenCV的cv2.threshold (Python)
  - OpenCV的void cv::threshold (C++)
  - https://reurl.cc/OqZAK9
- 影像標籤化
  - OpenCV的cv2.connectedComponentsWithStats (Python)
  - OpenCV的int cv::connectedComponentsWithStats (C++)
  - https://reurl.cc/Mdp0En
- 輪廓偵測
  - OpenCV的cv2.findContours (Python)
  - OpenCV的void cv::findContours (C++)
  - https://reurl.cc/odk1jl
- 繪圖輪廓：
  - OpenCV的cv2.drawContours (Python)
  - OpenCV的void cv::drawContours (C++)
  - https://reurl.cc/q875x0
- 計算周長：
  - OpenCV的cv2.arcLength (Python)
  - OpenCV的double cv::arcLength (C++)
  - https://reurl.cc/zz3Zvk
- 真圓度計算公式：
  - 4*math.pi*面積/(周長)^2

TASK

- 利用影像特徵進行影像處理操作。對影像做
  1. 二值化(Image Binarization)，
     a. 先對目標影像進行二值化，Thresholding 設為 55
  2. 標籤化(Labeling)並計算特徵參數，再由特徵參數分割影像:
     a. 影像標籤化並計算特徵參數
     b. 將面積小於100的標籤連通元件去除
     c. 將剩餘的標籤連通元件透過輪廓偵測計算周長和真圓度
     d. 將真圓度小於0.5的標籤連通元件去除，為了留下圓形的物體

實驗影像: fruit.bmp

https://reurl.cc/MX8krk



```python
#mount drive
from google.colab import drive
drive.mount('/content/drive')
```

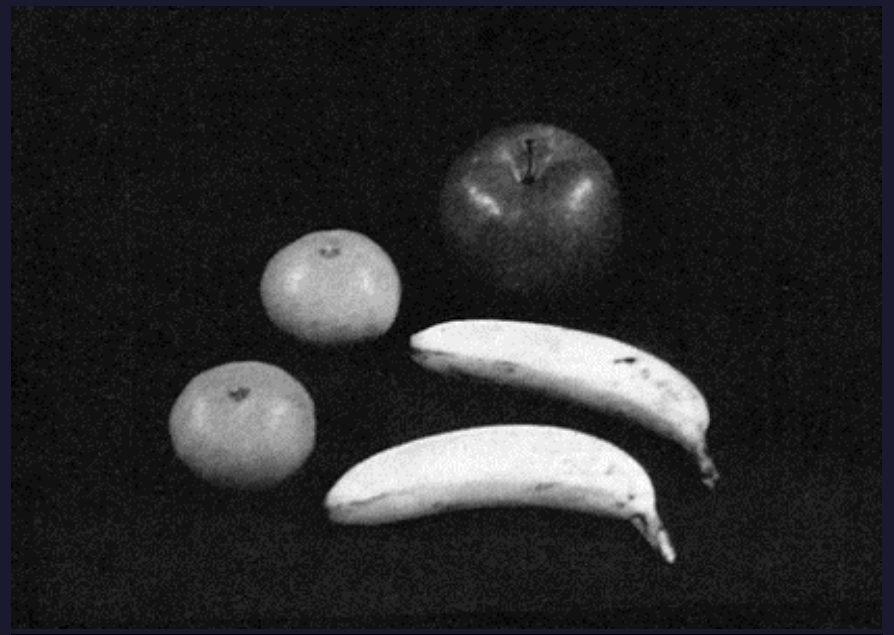
```python
# import libraries
import sys
import os
import cv2
import numpy as np
from matplotlib import pyplot as plt
from google.colab.patches import cv2_imshow
import math
```

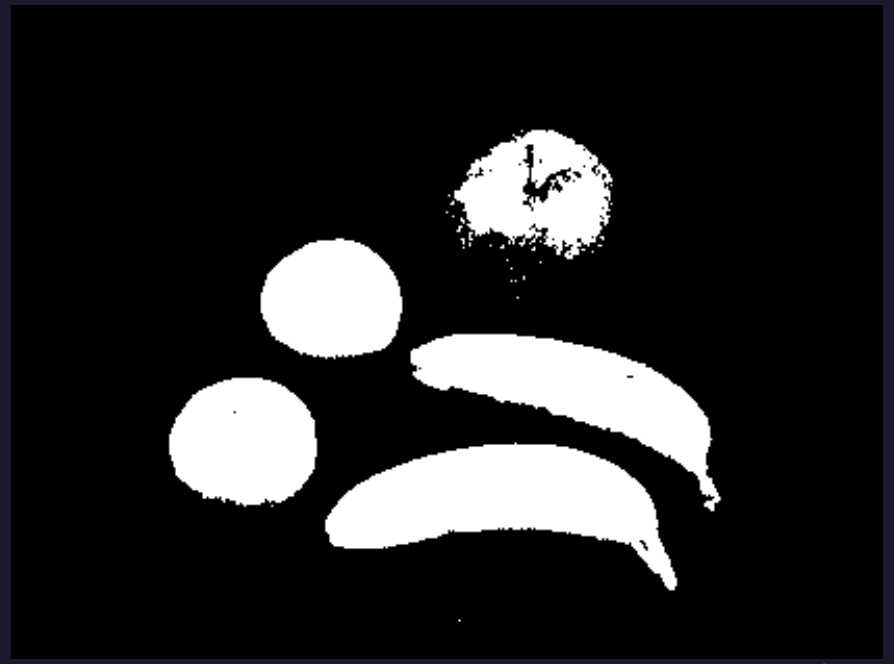
```python
# read an image
folder = r'/content/drive/MyDrive/images'
path_img = os.path.join(folder,'fruit.bmp')
img = cv2.imread(path_img)
# Afterwards, a check is executed, if the image was loaded correctly.
if img is None:
    sys.exit("Could not read the image.")
cv2_imshow(img)
img_gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
```

TASK

- 利用影像特徵進行影像處理操作。對影像做
  1. 二值化(Image Binarization)，
     a. 先對目標影像進行二值化，Thresholding 設為 55
  2. 標籤化(Labeling)並計算特徵參數，再由特徵參數分割影像:
     a. 影像標籤化並計算特徵參數
     b. 將面積小於100的標籤連通元件去除
     c. 將剩餘的標籤連通元件透過輪廓偵測計算周長和圓度
     d. 將圓度小於0.5的標籤連通元件去除，為了留下圓形的物體

```python
def Image_Binary(src,Thresholding=55):
  ret, thresh = cv2.threshold(src,Thresholding,255,cv2.THRESH_BINARY)
  return thresh
```





TASK

- 利用影像特徵進行影像處理操作。對影像做
  1. 二值化(Image Binarization)，
     a. 先對目標影像進行二值化，Thresholding 設為 55
  2. 標籤化(Labeling)並計算特徵參數，再由特徵參數分割影像:
     a. 影像標籤化並計算特徵參數
     b. 將面積小於指定大小(e.g.100)的標籤連通元件去除
     c. 將剩餘的標籤連通元件透過輪廓偵測計算周長和圓度
     d. 將圓度小於0.5的標籤連通元件去除，為了留下圓形的物體

`analysis = cv2.connectedComponentsWithStats(src,connectivity,ltype)`
computes the connected components labeled image of boolean image and also produces a statistics output for each label

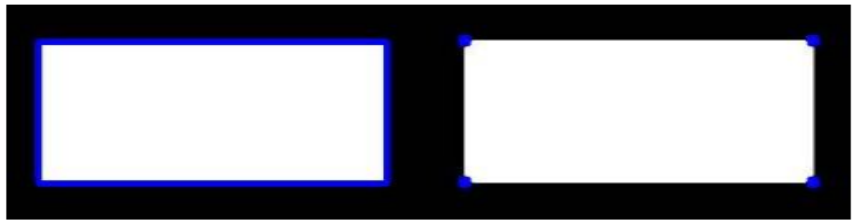
- Args[input]:
  - src: input image, must be a binary image and 8-bit single-channel image to be labeled
  - connectivity: 4 or 8 for 4-way or 8-way connectivity respectively
  - ltype: output image label type. Currently CV_32S or CV_16U are supported.
- Arge[output]:
  - num_labels: total labels
  - labels: label_ids
  - stats: values
  - center: centroid

```python
def Image_Labeling(src,src_gray,src_image_binary):
  # 連通元件
  kernel2 = cv2.getStructuringElement(cv2.MORPH_RECT,(3,3))
  bin_clo = cv2.dilate(src_image_binary,kernel2,iterations=2)
  # Applying the Component Analysis Method 應用成分分析方法
  # cv2.connectedComponentsWithStats(src,connectivity,ltype=None)
  # Args[input]:
  #   src: input image, must be a binary image
  #   connectivity: 4 or 8
  #   ltype: CV_32S or CV_16U
  # Arge[output]:
  #   num_labels: total labels
  #   labels: label_ids
  #   stats: values
  #   center: centroid
  analysis = cv2.connectedComponentsWithStats(
      bin_clo,
      connectivity=8,
      ltype=cv2.CV_32S)
  num_labels,labels,stats,centers = analysis

  # creating empty image 建立空影像
  src_image_labeling = np.copy(src)
  src_h, src_w = src_image_binary.shape
  src_image_contour = np.zeros((src_h,src_w,1), dtype = np.uint8)
  src_image_segmenation = np.zeros((src_h,src_w,1), dtype = np.uint8)

  # Computing feature parameters 計算特徵參數
  for i in range(1,num_labels):
    x,y,w,h, area = stats[i]
    cx, cy = centers[i]

    # 去除面積小於100的連通元件
    if (area >100):
      for row in range(src_h):
        for col in range(src_w):
          if (labels[row,col]==i):
            src_image_contour[row,col]=255
```

TASK

- 利用影像特徵進行影像處理操作。對影像做
  1. 二值化(Image Binarization)，
     a. 先對目標影像進行二值化，Thresholding 設為 55
  2. 標籤化(Labeling)並計算特徵參數，再由特徵參數分割影像:
     a. 影像標籤化並計算特徵參數
     b. 將面積小於100的標籤連通元件去除
     c. **將剩餘的標籤連通元件透過輪廓偵測計算周長和圓度**
     d. 將圓度小於0.5的標籤連通元件去除，為了留下圓形的物體

```
contours, hierachy = cv2.findContours(src,mode,Contour Approximation Method)
```

- Args[input]:
  - src:
    - input image
  - mode:
    - `cv2.RETR_EXTERNAL`：只檢測最外圍輪廓，包含在外圍輪廓內的內圍輪廓被忽略
    - `cv2.RETR_LIST`：檢測所有的輪廓，包括內圍、外圍輪廓。但是檢測到的輪廓不建立等級關係，彼此之間獨立
    - `cv2.RETR_CCOMP`：檢測所有的輪廓，但所有輪廓只建立兩個等級關係，外圍爲頂層
    - `cv2.RETR_TREE`：檢測所有輪廓，所有輪廓建立一個等級樹結構
  - Contour Approximation Method:
    - cv2.CHAIN_APPROX_SIMPLE: 保存物體邊界上所有連續的輪廓點到contours向量內
    - cv2.CHAIN_APPROX_NONE: 僅保存輪廓的拐點信息，把所有輪廓拐點處的點保存入contours向量內，角點與角點之間直線段上的點不予保留
    - 
- Args[output]:
  - contours: 輪廓，是一個 Python list，其中存儲這圖像中的所有輪廓。每一個輪廓都是一個 Numpy 數組，包 含對象邊界點（x，y）的座標。
  - hierachy: 層級結構

```python
    # Computing contours 計算輪廓
    # cv2.findContours(src,mode,Contour Approximation Method)
    # Args[input]:
    #   src: input image
    #   mode: cv2.RETR_EXTERNAL or cv2.RETR_LIST or cv2.RETR_CCOMP or cv2.RE
    #   Contour Approximation Method: cv2.CHAIN_APPROX_SIMPLE or cv2.CHAIN_A
    contours, hierarchy = cv2.findContours(
        src_image_contour,
        cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
    src_image_labeling = cv2.drawContours(
        src_image_labeling,
        contours,
        -1,
        (255,0,0),
        1)

    # drawing the contours and computing the perimeter 繪製輪廓並計算周長
    cnt = contours[0]
    perimeter = cv2.arcLength(cnt,True)

    # 計算圓度
    e = 4*math.pi*area*(1/perimeter*perimeter)

    # 劃出重心位置
    cv2.circle(src_image_labeling, (int(cx),int(cy)),2,(0,255,0),2,8,0)
    # 劃出外圍矩形
    cv2.rectangle(src_image_labeling,(x,y),(x+w,y+h),(0,0,255),1,8,0)
    cv2.putText(
        src_image_labeling,
        "No. "+str(i),
        (x,y-10),
        cv2.FONT_HERSHEY_SIMPLEX,
        .5,
        (0,0,255),
        1)
    print("No. "+str(i)+" 周長: %d, 面積: %d, 圓度: %lf"%(perimeter,area,e))
```

TASK

- 利用影像特徵進行影像處理操作。對影像做
  1. 二值化(Image Binarization)，
     a. 先對目標影像進行二值化，Thresholding 設為 55
  2. 標籤化(Labeling)並計算特徵參數，再由特徵參數分割影像:
     a. 影像標籤化並計算特徵參數
     b. 將面積小於100的標籤連通元件去除
     c. **將剩餘的標籤連通元件透過輪廓偵測計算周長和圓度**
     d. 將圓度小於0.5的標籤連通元件去除，為了留下圓形的物體

```
cv2.drawContours(image, contours, contourIdx, color, thickness, lineType, hierarchy, maxLevel, offset)
```

- Args[input]:
  - image：輸入圖像
  - contours：輪廓列表
  - contourIdx：指定要繪製輪廓的編號，如果是負數，則繪製所有的輪廓
  - color：輪廓顏色
  - thickness：輪廓線寬
  - lineType：輪廓線型
  - hierarchy：層級
  - maxLevel：繪製輪廓的最高級別，這個參數只有hierarchy有效的時候有效
    - maxLevel=0，繪製與輸入輪廓屬於同一等級的所有輪廓即輸入輪廓和與其相鄰的輪廓
    - maxLevel=1, 繪製與輸入輪廓同一等級的所有輪廓與其子節點。
    - maxLevel=2，繪製與輸入輪廓同一等級的所有輪廓與其子節點以及子節點的子節點

```
39        # Computing contours 計算輪廓
40        # cv2.findContours(src,mode,Contour Approximation Method)
41        # Args[input]:
42        #   src: input image
43        #   mode: cv2.RETR_EXTERNAL or cv2.RETR_LIST or cv2.RETR_CCOMP or cv2.RE
44        #   Contour Approximation Method: cv2.CHAIN_APPROX_SIMPLE or cv2.CHAIN_A
45        contours, hierarchy = cv2.findContours(
46            src_image_contour,
47            cv2.RETR_EXTERNAL,
48            cv2.CHAIN_APPROX_SIMPLE)
49        src_image_labeling = cv2.drawContours(
50            src_image_labeling,
51            contours,
52            -1,
53            (255,0,0),
54            1)
55
56        # drawing the contours and computing the perimeter 繪製輪廓並計算周長
57        cnt = contours[0]
58        perimeter = cv2.arcLength(cnt,True)
59
60        # 計算圓度
61        e = 4*math.pi*area*(1/perimeter*perimeter)
62
63        # 劃出重心位置
64        cv2.circle(src_image_labeling, (int(cx),int(cy)),2,(0,255,0),2,8,0)
65        # 劃出外圍矩形
66        cv2.rectangle(src_image_labeling,(x,y),(x+w,y+h),(0,0,255),1,8,0)
67        cv2.putText(
68            src_image_labeling,
69            "No. "+str(i),
70            (x,y-10),
71            cv2.FONT_HERSHEY_SIMPLEX,
72            .5,
73            (0,0,255),
74            1)
75        print("No. "+str(i)+" 周長: %d, 面積: %d, 圓度: %lf"%(perimeter,area,e))
```

TASK

- 利用影像特徵進行影像處理操作。對影像做
  1. 二值化(Image Binarization)，
     a. 先對目標影像進行二值化，Thresholding 設為 55
  2. 標籤化(Labeling)並計算特徵參數，再由特徵參數分割影像:
     a. 影像標籤化並計算特徵參數
     b. 將面積小於100的標籤連通元件去除
     c. 將剩餘的標籤連通元件透過輪廓偵測計算周長和圓度
     d. 將圓度小於0.5的標籤連通元件去除，為了留下圓形的物體，最後返回

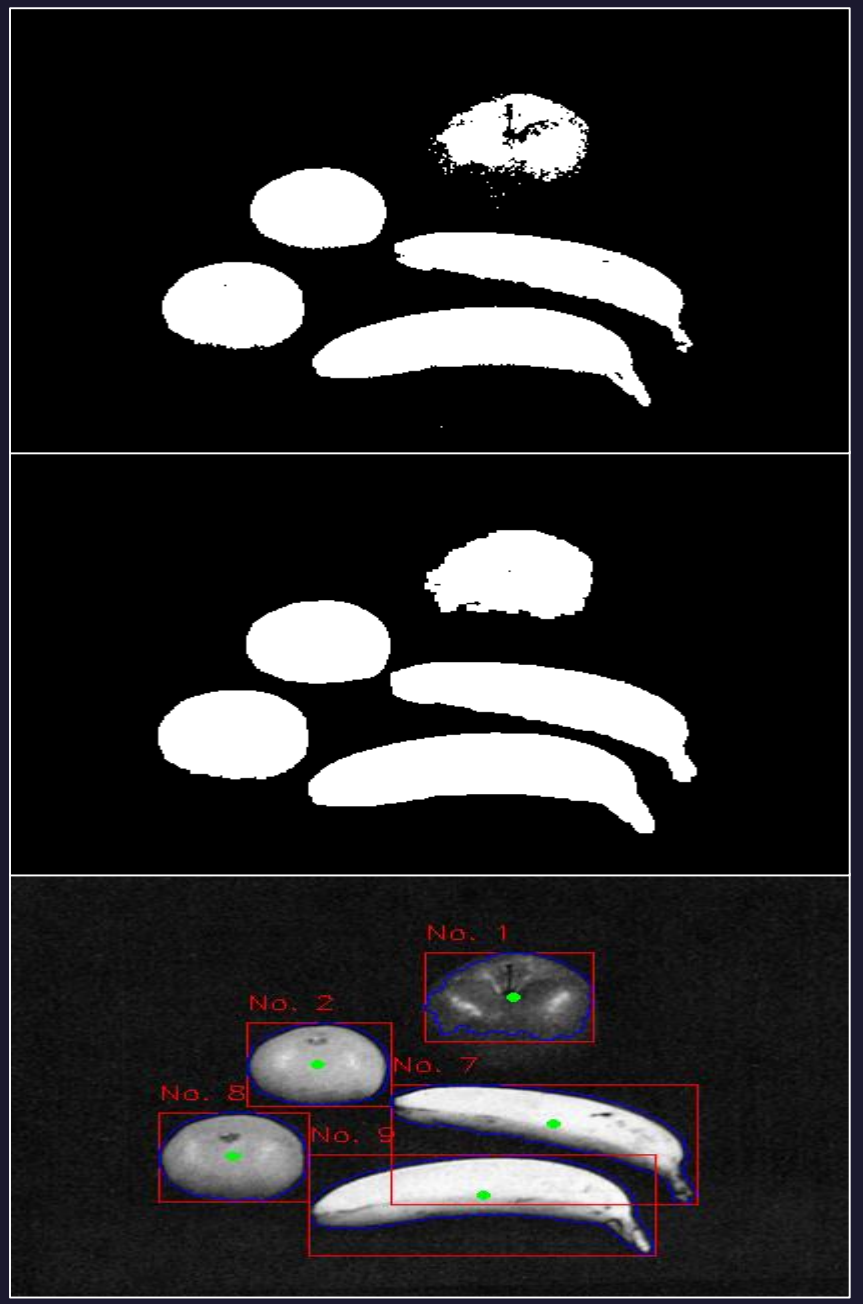
```
77        # 去除圓度小於0.5的連通元件
78        if (e<0.5):
79          continue
80        else:
81          for row in range(src_h):
82            for col in range(src_w):
83              if (labels[row,col]==i):
84                src_image_segmenation[row,col]==255
85    return src_image_contour, src_image_segmenation,src_image_labeling
```

TASK

- 利用影像特徵進行影像處理操作。對影像做
  1. 二值化(Image Binarization)，
     a. 先對目標影像進行二值化，Thresholding 設為 55
  2. 標籤化(Labeling)並計算特徵參數，再由特徵參數分割影像:
     a. 影像標籤化並計算特徵參數
     b. 將面積小於100的標籤連通元件去除
     c. 將剩餘的標籤連通元件透過輪廓偵測計算周長和圓度
     d. 將圓度小於0.5的標籤連通元件去除，為了留下圓形的物體

主函式

```
87  # main function
88  src_image_binary = Image_Binary(img_gray)
89  src_image_contour, src_image_segmenation,src_image_labeling = Image_Labeling(
90      img,
91      img_gray,
92      src_image_binary)
93
94  cv2_imshow(img)
95  cv2_imshow(src_image_binary)
96  cv2_imshow(src_image_contour)
97  cv2_imshow(src_image_labeling)
```



NATIONAL TAIWAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Thanks for listening

Thank You

詹宏澤

Email: chanhts323@gmail.com

https://sites.google.com/view/peter-chan