

# 牡蠣重量辨識系統

---

- 0. 環境設置：建立資料夾
- 1. 圖像處理
- 2. 牡蠣肉重預測

# 執行步驟

## 0. 環境設置：建立資料夾

## 1. 圖像處理

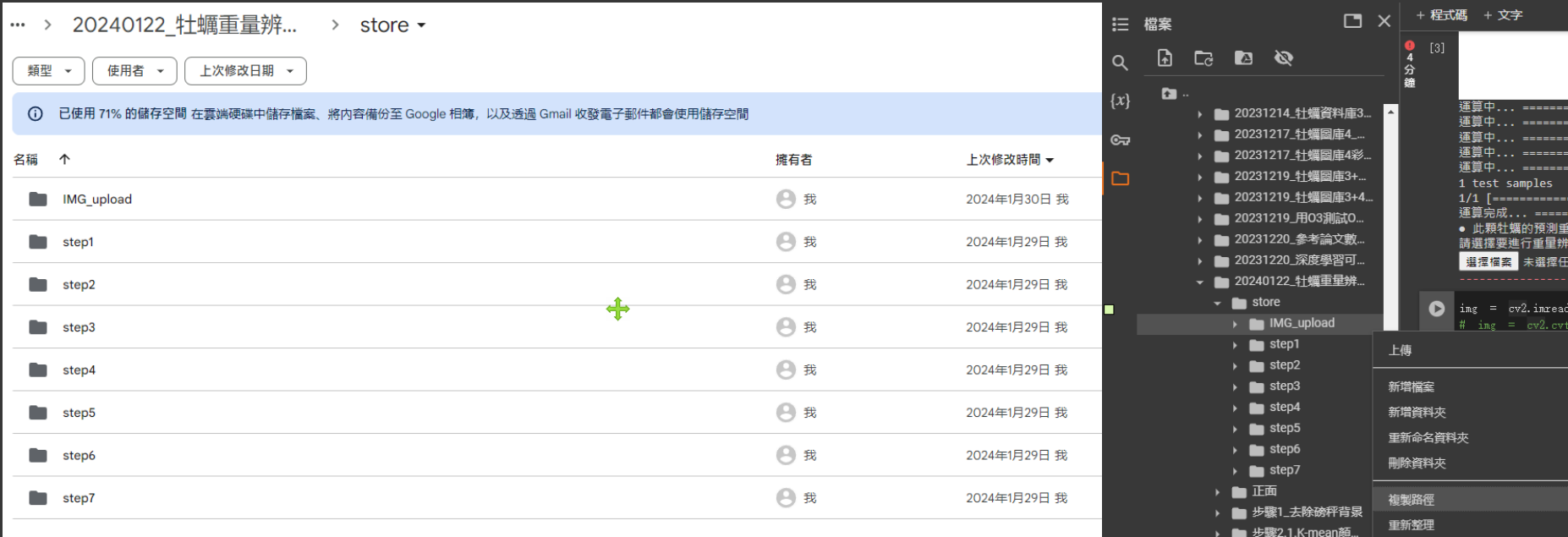
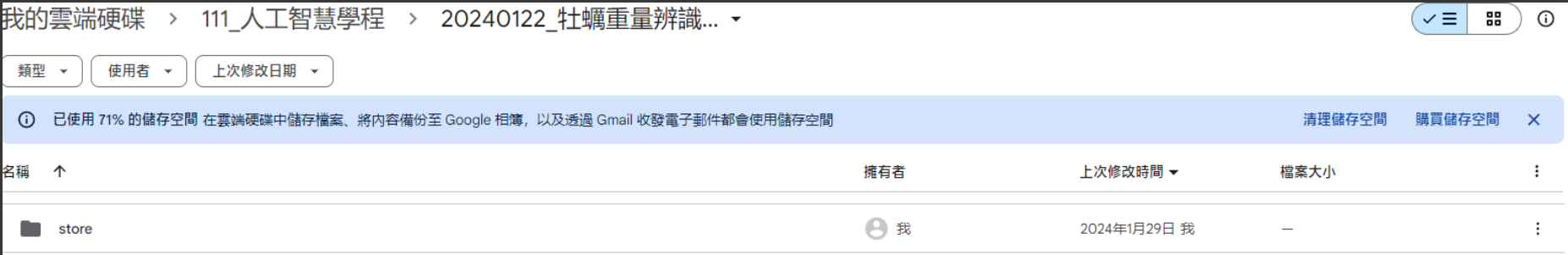
- 步驟1：去除磅秤背景
- 步驟2：K-mean顏色分群
- 步驟3：圖片二值化
- 步驟4：畫矩形框出蚵仔邊緣
- 步驟5：將圖片依蚵仔邊緣剪下
- 步驟6：尺寸不動，統一大小1200\*1200
- 步驟7：調整大小為400\*400

## 2. 牡蠣肉重預測

- 步驟1：載入模型
- 步驟2：重量預測

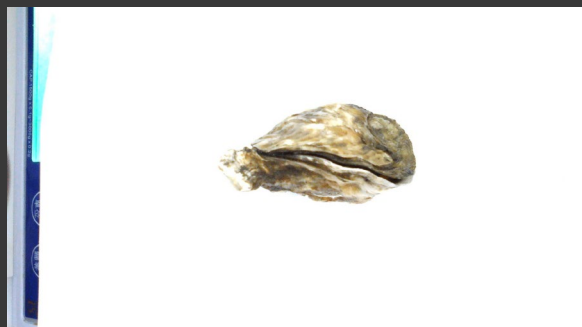
# 環境設置

- 建立資料夾



# 圖像處理

- 步驟1：去除磅秤背景



```
def step1(img_path): # 步驟1. 去除磅秤背景

    img = cv2.imread(img_path)
    x = 300
    y = 100
    w = 1420
    h = 880
    crop_img = img[y:y+h, x:x+w] # 取出陣列的範圍(裁切圖片範圍)

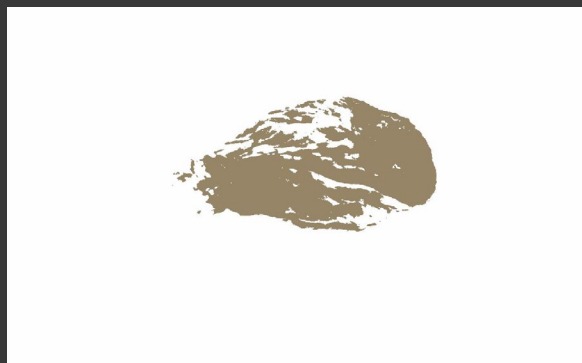
    split_result = str(img_path).split('/') # 取出圖片檔名
    file_name = split_result[-1].split('.')
    # print(file_name[0])
    cv2.imwrite("/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step1/" + file_name[0] + ".jpg", crop_img) # 依照原圖片檔名儲存圖片
```

```
os.chdir('/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/IMG_upload') #切換該目錄
os.listdir() #確認目錄內容

# IMG and DATA input
print('請選擇要進行重量辨識的牡蠣圖片', end = '')
upload = files.upload() #開啟上傳介面並載入圖片
upload_list = list(upload)
img_filename = upload_list[0]
img = cv2.imread("/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/IMG_upload/" + img_filename) #讀取上傳的圖片
```

# 圖像處理

## • 步驟2：K-mean顏色分群



store > step2

類型 使用者 上次修改日期

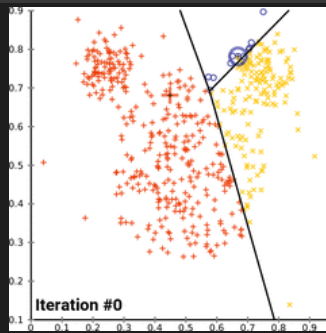
已使用 71% 的儲存空間 在雲端硬碟中儲存檔案、將內容備份至 Google 相簿，以及透過 Gmail 收發電子郵件都會使用儲存空間 清理儲存空間 購買儲存空間

名稱	擁有者	上次修改時間	檔案大小
01.jpg	我	2024年1月30日 我	42 KB

```
def step2(img_path): # 步驟2.K-mean顏色分群
# read the image 加載圖像
img = cv2.imread(img_path)

# convert to RGB 圖像轉換為RGB格式
image = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
# reshape the image to a 2D array of pixels and 3 color values (RGB)
pixel_values = image.reshape((-1, 3))
# convert to float
pixel_values = np.float32(pixel_values)
# 印出像素值的shape
print(pixel_values.shape)
# 定義停止標準
# define stopping criteria
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.2)
# number of clusters (K)
k = 2
_, labels, (centers) = cv2.kmeans(pixel_values, k, None, criteria, 10, cv2.KMEANS_RANDOM_CENTERS)
# convert back to 8 bit values
centers = np.uint8(centers)
# flatten the labels array
labels = labels.flatten()
# convert all pixels to the color of the centroids
segmented_image = centers[labels.flatten()]
# 轉換回原始圖像並顯示
# reshape back to the original image dimension
segmented_image = segmented_image.reshape(image.shape)
# show the image
pp = plt.imshow(segmented_image)
plt.show()
pp = Image.fromarray(segmented_image) # numpy轉image
split_result = str(img_path).split('/') # 取出圖片檔名
file_name = split_result[-1].split('.')
# print(file_name[0])
pp.save("/content/drive/MyDrive/111 人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step2/" + file_name[0] + ".jpg")# 依照原圖片檔名儲存圖片

step2("/content/drive/MyDrive/111 人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step1/" + img_filename) #讀取step1的圖片，執行 步驟2.K-mean顏色分群
```

A scatter plot titled "Iteration #0" showing the results of K-means clustering. The x-axis ranges from 0.1 to 0.9, and the y-axis ranges from 0.1 to 0.9. The plot displays two clusters of data points: one cluster is colored red and the other is colored yellow. Two black lines represent the decision boundaries between the clusters. The red cluster is located in the lower-left region, and the yellow cluster is in the upper-right region.

# 圖像處理

## • 步驟3：圖片二值化



```
... > store > step3 ▾  
類型 ▾ 使用者 ▾ 上次修改日期 ▾  
① 已使用 71% 的儲存空間 在雲端硬碟中儲存檔案、將內容備份至 Google 相簿，以及透過 Gmail 收發電子郵件都會使用儲存空間 清理儲存空間 購買儲存空間 ×  
名稱 ↑ 擁有者 上次修改時間 ▾ 檔案大小  
01.jpg 我 2024年1月30日 我 72 KB  
def step3(path): # 步驟3. 圖片二值化  
  
    img = cv2.imread(path, cv2.IMREAD_UNCHANGED) # 開啟圖片  
    # img = cv2.cvtColor(img, cv2.COLOR_BGR2BGRA) # 因為是 jpg, 要轉換顏色為 BGRA  
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # 新增 gray 變數為轉換成灰階的圖片  
  
    bin_threshold = min(gray.flatten()) #二值化判斷 閾值、門檻值(threshold)  
    # bin_threshold = 240 #二值化判斷 閾值、門檻值(threshold)  
  
    h = img.shape[0] # 取得圖片高度  
    w = img.shape[1] # 取得圖片寬度  
  
    # 依序取出圖片中每個像素  
    for x in range(w):  
        for y in range(h):  
            if gray[y, x] > bin_threshold : # 如果該像素的灰階度大於 二值化判斷閾值 設為0(黑點)  
                gray[y, x] = 0  
            else:  
                gray[y, x] = 255 # 如果該像素的灰階度小於 二值化判斷閾值 設為255(白點)  
  
step3("/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step2/" + img_filename) #讀取step2的圖片，執行 步驟3. 圖片二值化
```

# 圖像處理

• 步驟4：畫矩形框出蚵仔邊緣



```
... > store > step4
類型 使用者 上次修改日期
已使用 71% 的儲存空間 在雲端硬碟中儲存檔案、將內容備份至 Google 相簿，以及透過 Gmail 收發電子郵件都會使用儲存空間 清理儲存空間 購買儲存空間
名稱 擁有者 上次修改時間 檔案大小
01.jpg 我 2024年1月30日 我 93 KB

# 步驟4. 畫矩形框出蚵仔邊緣
def draw_coordinates(img): #在蚵仔邊緣畫矩形的四座標
    img = cv2.imread(img, cv2.IMREAD_GRAYSCALE) #以灰階模式讀取之前轉換好的二值化圖像
    #依列讀取每行像素值，算出平均值並正規化與0~1之間
    r = [] #儲存每一列正規化值
    for i in range(len(img)):
        add = 0
        for j in range(len(img[i])):
            p=img[i][j]
            p=p/255 #像素值/255正規化
            add = add+p #同行正規化像素值相加
        add = add/len(img[i]) #同行正規化像素值全相加後/len(img[i]) (整行像素點個數)再正規化一次
        r.append(add) #存入每一行的像素值總和正規化值
    # print(add)
    # 設雜訊門檻值：依據每一行的像素值總和正規化值(黑白點比例)去除雜訊
    r_index = [] #儲存大於雜訊門檻值的行數index
    for ii in range(len(r)):
        if r[ii] >= 0.01: #雜訊門檻值設0.01
            r_index.append(ii) #如當列像素值總和正規化值大於雜訊門檻值存入r_index

# r_index
c = []
# for k in range(len(img)):
for L in range(len(img[0])):
    add = 0
    for m in range(len(img)):
        p=img[m][L]
        p=p/255
        add = add+p
    add = add/len(img)
    c.append(add)
# print(add)
c_index = []
for jj in range(len(c)):
    if c[jj] >= 0.01:
        c_index.append(jj)
# c_index
return min(c_index),min(r_index),max(c_index),max(r_index)

def draw_rectangle(Path2, x1, y1, x2, y2): #在蚵仔邊緣畫矩形
    # image = cv2.imread(Path) #讀二值化圖辨識範圍
    image2 = cv2.imread(Path2) #讀原圖畫矩形
    # 畫矩形
    left_up = (x1, y1)
    right_down = (x2, y2)
    color = (0, 0, 255) # red
    thickness = 2 # 寬度 (-1 表示填滿)
    cv2.rectangle(image2, left_up, right_down, color, thickness)
    # cv2.imwrite('draw_rectangle.jpg', image2)
    cv2.imshow(image2) #顯示圖片
    return image2

# 步驟4. 畫矩形框出蚵仔邊緣
imagePaths4_1 = '/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step3/' + img_filename
imagePaths4_2 = '/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step1/' + img_filename
aaa = draw_coordinates(imagePaths4_1) #讀二值化圖辨識範圍
bbb = draw_rectangle(imagePaths4_2,aaa[0],aaa[1],aaa[2],aaa[3]) #讀原圖畫矩形
cv2.imwrite('/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step4/' + img_filename , bbb)# 依照原圖片檔名儲存圖片，存檔儲存為 png 到 step4
```

# 圖像處理

- 步驟5：將圖片依蚵仔邊緣剪下



```
# 步驟5. 將圖片依蚵仔邊緣剪下
def cut_rectangle(Path2, x1, y1, x2, y2): #將圖片依蚵仔邊緣剪下

    # image = cv2.imread(Path) #讀二質化圖辨識範圍
    image2 = cv2.imread(Path2) #讀原圖畫矩形

    cut_img = image2[y1:y2, x1:x2]

    # cv2.imshow(cut_img) #顯示圖像

    return cut_img
```

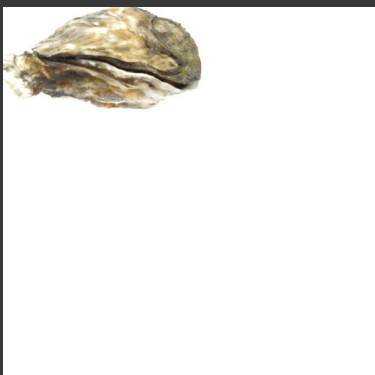
```
# 步驟5. 將圖片依蚵仔邊緣剪下
ccc = cut_rectangle(imagePaths4_2, aaa[0], aaa[1], aaa[2], aaa[3]) #將圖片依蚵仔邊緣剪下

cv2.imwrite("/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step5/" + img_filename , ccc) # 依照原圖片檔名儲存圖片，存檔儲存為 png 到 step5
```



# 圖像處理

- 步驟6：尺寸不動，統一大小1200\*1200



...

>

store

>

step6

類型

使用者



上次修改日期

已使用 71% 的儲存空間

在雲端硬碟中儲存檔案、將內容備份至 Google 相簿，以及透過 Gmail 收發電子郵件都會使用儲存空間

清理儲存空間

購買儲存空間

名稱	擁有者	上次修改時間	檔案大小	
 01.jpg	 我	2024年1月30日 我	86 KB	

```
# 步驟6. 尺寸不動，統一大小1200*1200
def scale(img_prth, long_size, short_size):

    img = cv2.imread(img_prth) #讀原圖畫矩形
    h, w = img.shape[0:2]

    a = short_size - h
    b = long_size - w

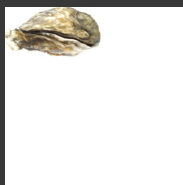
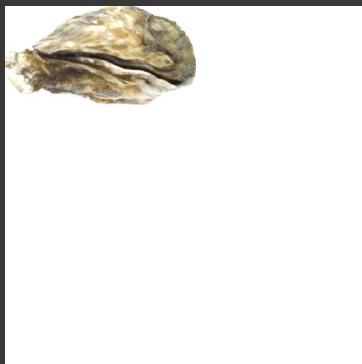
    if a > 0:
        img = np.pad(img, ((0, a), (0, 0), (0, 0)), "constant", constant_values=255)
    if b > 0:
        img = np.pad(img, ((0, 0), (0, b), (0, 0)), "constant", constant_values=255)
    h2, w2 = img.shape[0:2]
    scale = short_size * 1.0 / h2
    scale2 = long_size*1.0/w2
    img = cv2.resize(img, dsize=None, fx=scale2, fy=scale)
    # print(img.shape)
    return img
```

```
# 步驟6. 尺寸不動，統一大小1200*1200
img_step6 = scale("/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step5/" + img_filename ,1200,1200)

cv2.imwrite('/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step6/' + img_filename , img_step6)
```

# 圖像處理

- 步驟7：調整大小為400\*400



...

>

store

>

step7

類型

使用者

上次修改日期

已使用 71% 的儲存空間

在雲端硬碟中儲存檔案、將內容備份至 Google 相簿，以及透過 Gmail 收發電子郵件都會使用儲存空間

清理儲存空間

購買儲存空間

名稱	擁有者	上次修改時間	檔案大小
 01.jpg	 我	2024年1月30日 我	14 KB

```
# 步驟7. 調整大小為400*400
img_step7_1 = cv2.imread("/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step6/" + img_filename) # 開啟圖片
img_step7_2 = cv2.resize(img_step7_1, (400, 400), interpolation=cv2.INTER_AREA)
# cv2.imshow(img_step7_2)
cv2.imwrite('/content/drive/MyDrive/111_人工智慧學程/20240122_牡蠣重量辨識系統整合/store/step7/' + img_filename , img_step7_2)
```

# 牡蠣肉重預測

- 步驟1：載入模型

我的雲端硬碟 > 111_人工智慧學程 > 20231217_牡蠣圖庫4彩... ▾					✓ 三	田	i
類型 ▾ 使用者 ▾ 上次修改日期 ▾							
名稱 ↑	擁有者		上次修改時間 ▾	檔案大小	⋮		
 20231219_O4_Color_Feature4_CNN_e1000.h5	 我		2023年12月19日 我	3.59 GB	⋮		

```
from keras.models import load_model #載入模型
model = load_model("/content/drive/MyDrive/111_人工智慧學程/20231217_牡蠣圖庫4彩色+label關係強特徵模型建立/20231219_O4_Color_Feature4_CNN_e1000.h5")
```

# 牡蠣肉重預測

## • 步驟2：重量預測

```
# IMG and DATA input
print('請選擇要進行重量辨識的牡蠣圖片',end = "")
upload = files.upload() #開啟上傳介面並載入圖片

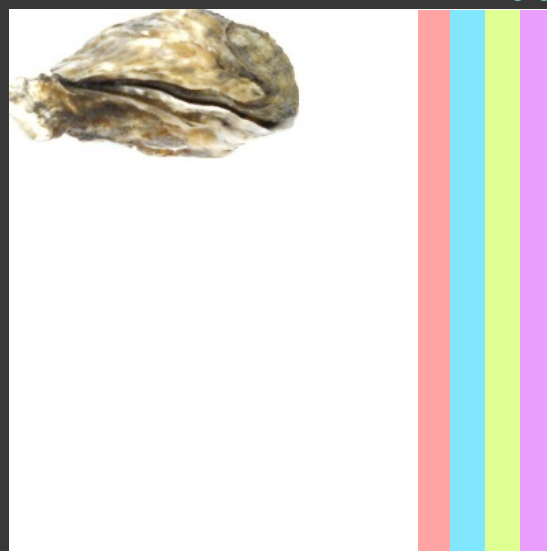
print('=====請輸入牡蠣基本數據=====')
Total_weight = input('總重量 :') #輸入牡蠣基本數據
Length = input('殼長度 :')
Width = input('殼寬度 :')
Height = input('殼高度 :')

count1 = 0
for count1 in range(len(img)):
    img[count1][-1][:] = Total_weight
    count1 = count1 + 1

count2 = 0
for count2 in range(len(img)):
    img[count2][-4][:] = Length
    count2 = count2 + 1

count3 = 0
for count3 in range(len(img)):
    img[count3][-3][:] = Width
    count3 = count3 + 1

count4 = 0
for count4 in range(len(img)):
    img[count4][-2][:] = Height
    count4 = count4 + 1
```



Total weight  
Width

Length  
Height

```
#模型 x data 設定
x = np.array(img)
xx = x.reshape(1, 400, 400, 3)
x_test = xx
# x data 正規化
# 1. 整數轉浮點數
x_test = x_test.astype('float32')
# 2. 直接除以最大值，標準化成0~1浮點數
x_test /= 255
print(x_test.shape[0], 'test samples')

y_pred = model.predict(x_test)

print("運算完成... =====> 100%")
print("● 此顆牡蠣的預測重量為 : ", y_pred[0][0], " 公克(g)")
```

Saving 01.jpg to 01.jpg  
=====請輸入牡蠣基本數據=====

總重量 : 6.8  
殼長度 : 1  
殼寬度 : 2  
殼高度 : 3

運算完成... =====> 100%  
● 此顆牡蠣的預測肉重為 : 5.451725 公克(g)