

# 實習一

## 影像中物體的分割



# 課程大綱

- 實習00: Colab 環境
- 實習01: 影像中物體的分割





實習 00

Colab 環境

Colab Env.

*Before we start...*

```
1 | #mount drive
2 | from google.colab import drive
3 | drive.mount('/content/drive')

4 | # import libraries
5 | import sys
6 | import os
7 | import cv2
8 | import numpy as np
9 | from matplotlib import pyplot as plt
10 | from google.colab.patches import cv2_imshow
```



實習 01

# 影像中物體的分割

Image Segmentation

# TASK: 對影像作二值化(Threshold), 分割文字(前景)與白底(背景)。

實驗影像(test\_P05.bmp)下載處:

• <https://reurl.cc/ZbkzEI>

## 使用函式:

- Python : OpenCV 的 **cv2.threshold**
- C++ : OpenCV 的 **double cv::threshold**

## 使用以下 5 種方法對影像做二值化:

- **cv2.THRESH\_BINARY**: Threshold Binary, 即二值化, 將大於門檻值的灰階值設為最大灰階值, 小於門檻值的值設為0。
- **cv2.THRESH\_BINARY\_INV**: Threshold Binary, Inverted, 將大於門檻值的灰階值設為0, 其他值設為最大灰階值。
- **cv2.THRESH\_TRUNC**: Truncate, 將大於門檻值的灰階值設為門檻值, 小於門檻值的值保持不變。
- **cv2.THRESH\_TOZERO**: Threshold to Zero, 將小於門檻值的灰階值設為0, 大於門檻值的值保持不變。
- **cv2.THRESH\_TOZERO\_INV**: Threshold to Zero, Inverted, 將大於門檻值的灰階值設為0, 小於門檻值的值保持不變。

## 門檻值: 130

### Threshold

```
cv.threshold( src, thresh, maxval, type[, dst] ) -> retval, dst
```

The function applies fixed-level thresholding to a multiple-channel array. The function is typically used to get a bi-level (binary) image out of a grayscale image ( compare could be also used for this purpose) or for removing a noise, that is, filtering out pixels with too small or too large values. There are several types of thresholding supported by the function. They are determined by type parameter.

該函數將固定級別的閾值應用於多通道陣列。該函數通常用於從灰度圖像中獲取雙層（二進制）圖像（比較也可用於此目的）或用於去除噪聲，即過濾掉值過小或過大的像素。該函數支持幾種類型的閾值。它們由類型參數確定。

#### Parameters:

- **src** input array (multiple-channel, 8-bit or 32-bit floating point).
- **dst** output array of the same size and type and the same number of channels as src.
- **thresh** threshold value.
- **maxval** maximum value to use with the THRESH\_BINARY and THRESH\_BINARY\_INV thresholding types.
- **type** thresholding type (see [ThresholdTypes](#)).

### Enumerator

#### THRESH\_BINARY

Python: cv.THRESH\_BINARY

$$\text{dst}(x, y) = \begin{cases} \text{maxval} & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

#### THRESH\_BINARY\_INV

Python: cv.THRESH\_BINARY\_INV

$$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{maxval} & \text{otherwise} \end{cases}$$

#### THRESH\_TRUNC

Python: cv.THRESH\_TRUNC

$$\text{dst}(x, y) = \begin{cases} \text{threshold} & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$$

#### THRESH\_TOZERO

Python: cv.THRESH\_TOZERO

$$\text{dst}(x, y) = \begin{cases} \text{src}(x, y) & \text{if } \text{src}(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

#### THRESH\_TOZERO\_INV

Python: cv.THRESH\_TOZERO\_INV

$$\text{dst}(x, y) = \begin{cases} 0 & \text{if } \text{src}(x, y) > \text{thresh} \\ \text{src}(x, y) & \text{otherwise} \end{cases}$$

# 程式碼與結果

Colab : [https://colab.research.google.com/drive/1jGRz\\_gH-LjRfRRiONZV31E3muMqBR3a?usp=sharing](https://colab.research.google.com/drive/1jGRz_gH-LjRfRRiONZV31E3muMqBR3a?usp=sharing)

HackMD : <https://hackmd.io/@chanhts/rky7Hfwfj>

```
1 # read an image
2 folder = r'/content/drive/MyDrive/images'
3 path_img = os.path.join(folder, 'test_P05.bmp')
4 img = cv2.imread(path_img)
5 # Afterwards, a check is executed, if the image was loaded correctly.
6 if img is None:
7     sys.exit("Could not read the image.")
8 cv2.imshow(img)
9 img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
10 # Threshold function
11 def Threshold(img):
12     Thresholding=130
13     ##retval, dst=cv.threshold(src, thresh, maxval, type[, dst])
14     ret,thresh1 = cv2.threshold(img,Thresholding,255,cv2.THRESH_BINARY)
15     ret,thresh2 = cv2.threshold(img,Thresholding,255,cv2.THRESH_BINARY_INV)
16     ret,thresh3 = cv2.threshold(img,Thresholding,0,cv2.THRESH_TRUNC)
17     ret,thresh4 = cv2.threshold(img,Thresholding,0,cv2.THRESH_TOZERO)
18     ret,thresh5 = cv2.threshold(img,Thresholding,255,cv2.THRESH_TOZERO_INV)
19     titles = ['Original Image', 'BINARY', 'BINARY_INV', 'TRUNC', 'TOZERO', 'TOZERO_INV']
20     images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]
21     for i in range(len(images)):
22         if(i==0):
23             cv2.imwrite('original image.jpg', images[i])
24         if(i>0):
25             cv2.imwrite('thresh'+str(i)+'.jpg', images[i])
26     for i in range(len(images)):
27         plt.subplot(2,3,i+1),plt.imshow(images[i], 'gray', vmin=0, vmax=255)
28         plt.title(titles[i])
29         plt.xticks([], plt.yticks([]))
30     plt.savefig('Image_Threshold_Result.jpg')
31     plt.show()
32
33 Threshold(img_gray)
```

Original

BINARY

BINARY\_INV

TRUNC

TOZERO

TOZERO\_INV

der Natur, und diese sind die Uraanfänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraanfänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

der Natur, und diese sind die Uraanfänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraanfänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

der Natur, und diese sind die Uraanfänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraanfänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

der Natur, und diese sind die Uraanfänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraanfänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

der Natur, und diese sind die Uraanfänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraanfänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.

der Natur, und diese sind die Uraanfänge der Dinge.  
Ihre Vereinigung ist der Grund aller Urstoffe, oder  
die Fähigkeit, die die Uraanfänge erlangen zur Bil-  
dung der Urstoffe in der Körperwelt.





**Thanks for listening**