

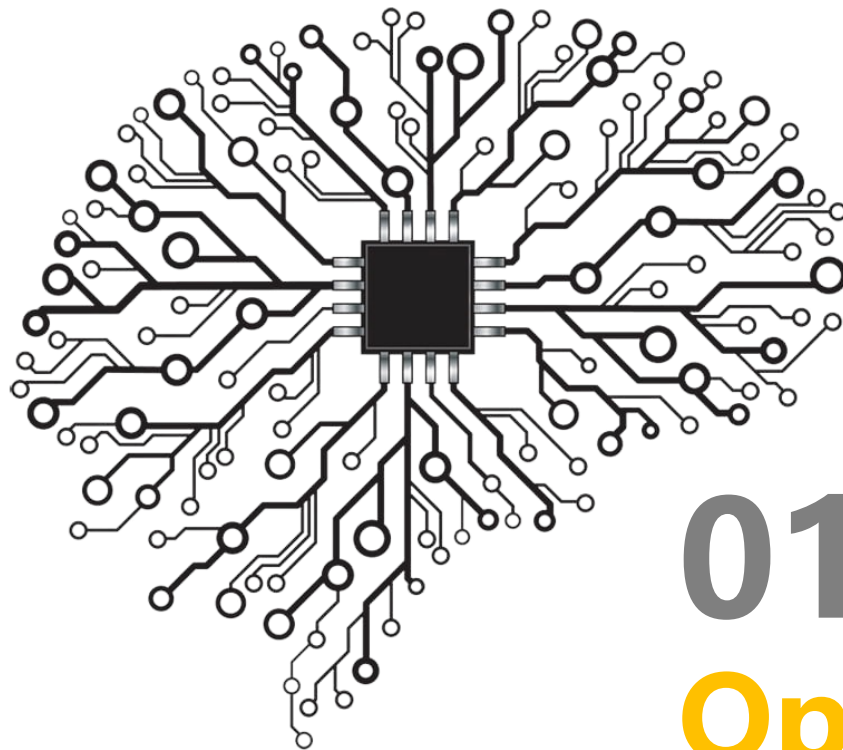


# 影像強化處理

# 目標任務



- 對此次課程提供的影像使用**直方圖等化**(Histogram Equalization, HE)和**限制對比度自適應直方圖等化**(Contrast Limited Adaptive Histogram Equalization, CLAHE)進行**影像增強**(Image Enhancement)
- 使用函數：
  - cv2.equalizeHist
  - cv2.createCLAHE
- 使用以下參數對影像做CLAHE操作：
  - clipLimit: 2.0
  - tileGridSize: 8



01

# OpenCV: Histogram



```
cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]]) -> hist
```

Parameters	Description
<b>images</b>	Source arrays. They all should have the same depth, CV_8U, CV_16U or CV_32F , and the same size. Each of them can have an arbitrary number of channels.
<b>channels</b>	List of the dims channels used to compute the histogram. The first array channels are numerated from 0 to images[0].channels()-1 , the second array channels are counted from images[0].channels() to images[0].channels() + images[1].channels()-1, and so on.
<b>mask</b>	Optional mask. If the matrix is not empty, it must be an 8-bit array of the same size as images[i] . The non-zero mask elements mark the array elements counted in the histogram.
<b>histSize</b>	Array of histogram sizes in each dimension.
<b>ranges</b>	Array of the dims arrays of the histogram bin boundaries in each dimension. ( <a href="#">see more...</a> )
<b>hist</b>	Output histogram, which is a dense or sparse dims -dimensional array.
<b>accumulate</b>	Accumulation flag. If it is set, the histogram is not cleared in the beginning when it is allocated. This feature enables you to compute a single histogram from several sets of arrays, or to update the histogram in time.

# 影像直方圖



```
def calc_histogram(image, use_bgr=False):  
    c = 3 if use_bgr else 1  
    hist = [cv2.calcHist([image], [i], None, [256], [0, 255]) for i in range(c)]  
    return hist
```

```
def plot_pic(hist, use_bgr=False):  
    color = ['b', 'g', 'r'] if use_bgr else ['gray']  
    for i, col in enumerate(color):  
        plt.plot(hist[i], color=col)  
        plt.xlim([0, 255])  
    plt.show()
```

# 影像直方圖



MIT Lab.

Multimedia  
Intelligent Technical  
Laboratory

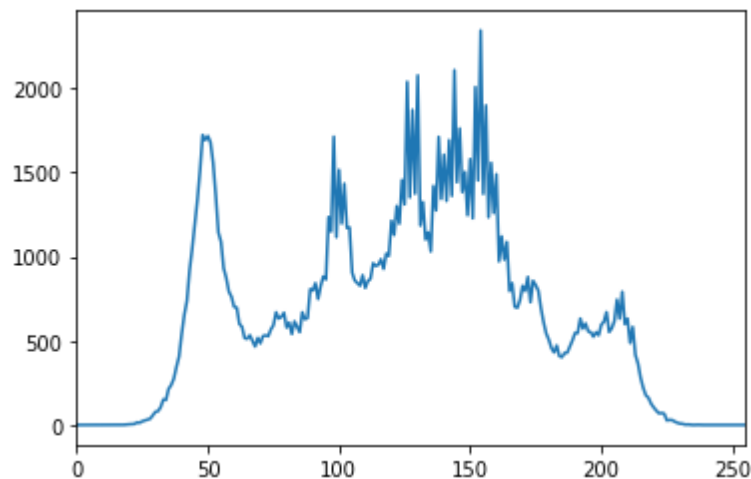
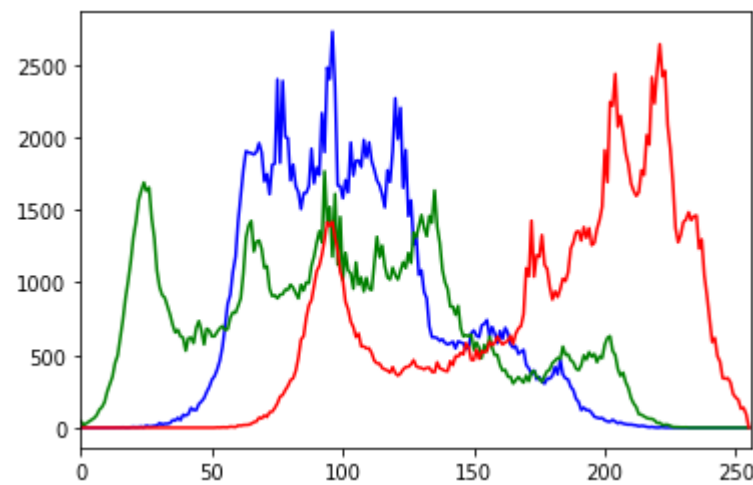
```
image = cv2.imread('./pout.bmp')
grayimage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

bgr_hist = calc_histogram(image, use_bgr=True)
plot_pic(bgr_hist, use_bgr=True)
gray_hist = calc_histogram(grayimage, use_bgr=False)
plot_pic(gray_hist, use_bgr=False)
```

# 影像直方圖

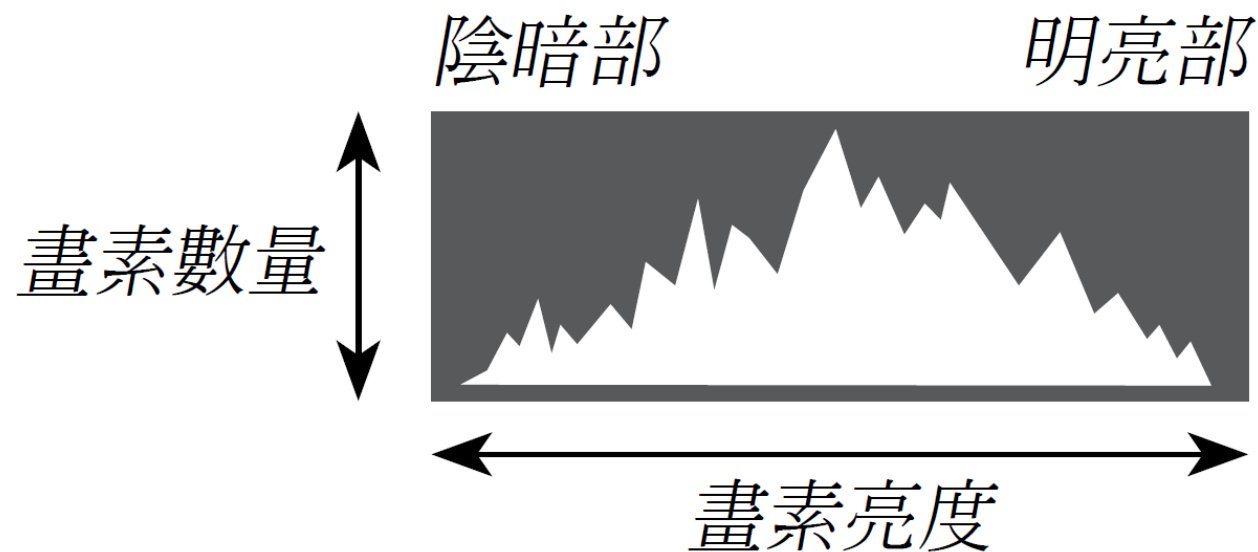


MIT Lab.  
Multimedia  
Intelligent Technical  
Laboratory





## 影像直方圖



**最佳曝光：**畫素在整個色調曲線範圍內平均分佈。



**曝光過度：**畫素聚集在色階分佈圖的右側。



**曝光不足：**畫素聚集在色階分佈圖的左側。





# 直方圖等化



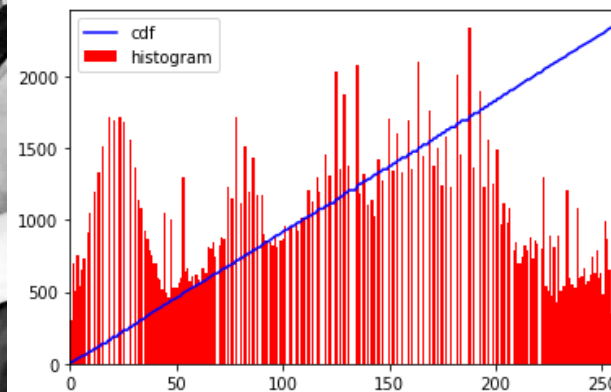
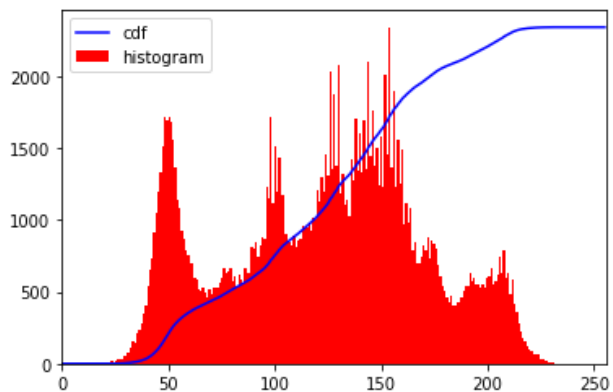
```
cv2.equalizeHist(src[, dst]) -> dst
```

Parameters		Description
<b>src</b>	Source 8-bit single channel image.	
<b>dst</b>	Destination image of the same size and type as src .	

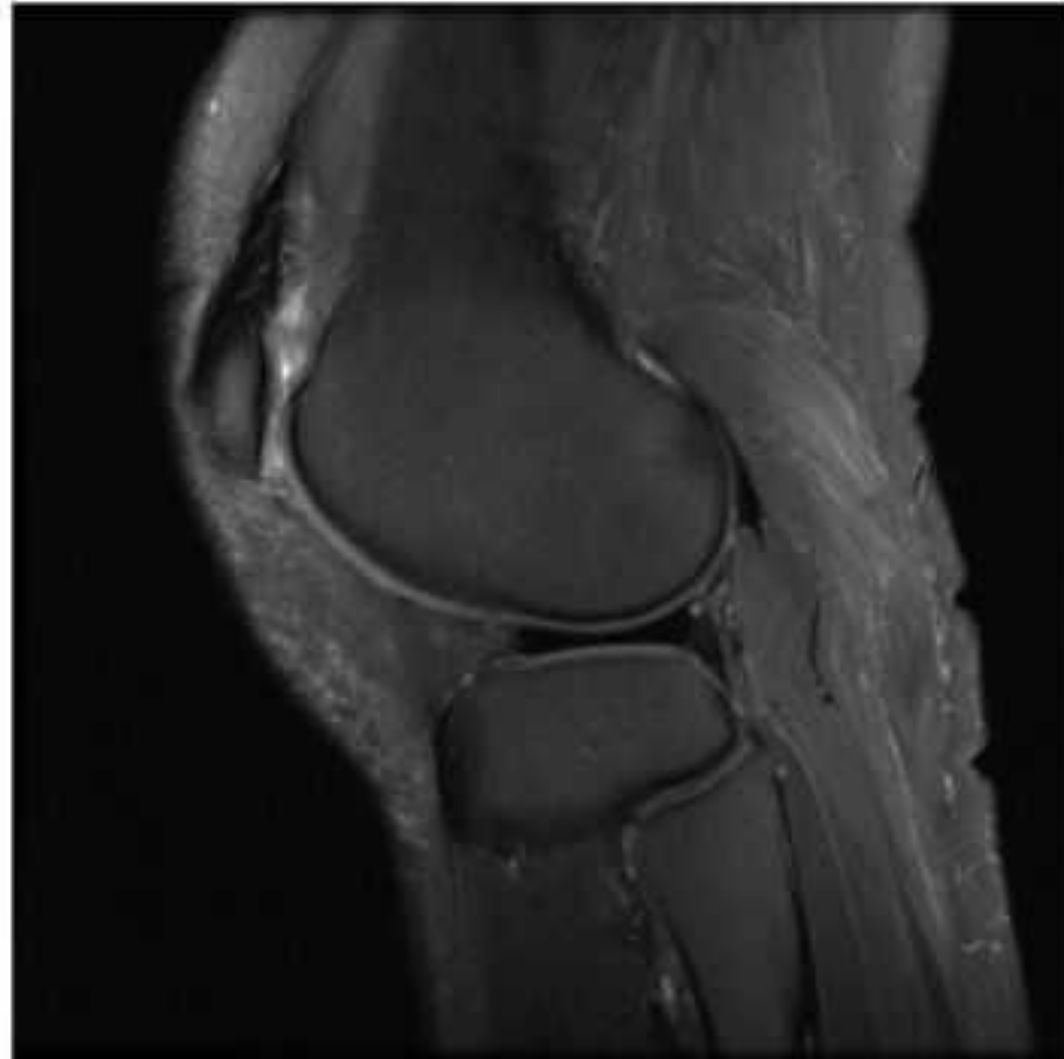
# 直方圖等化



```
def HE( ):  
    return cv2.equalizeHist( )
```



# 直方圖等化



# 限制對比度自適應直方圖等化



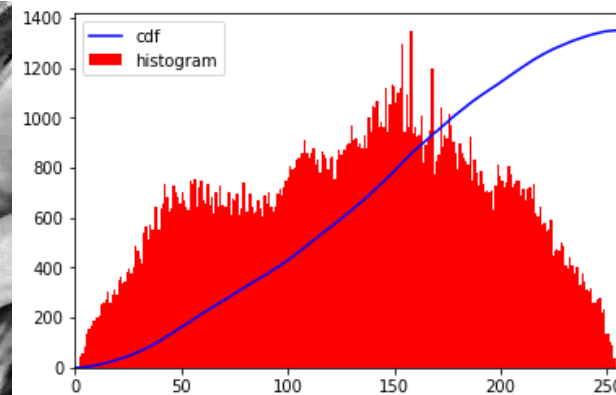
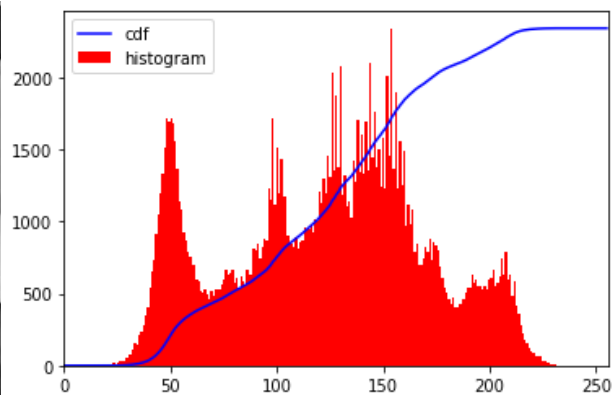
```
cv2.createCLAHE([, clipLimit[, tileGridSize]]) -> retval
```

Parameters		Description
<b>clipLimit</b>	Threshold for contrast limiting.	
<b>tileGridSize</b>	Size of grid for histogram equalization. Input image will be divided into equally sized rectangular tiles. tileGridSize defines the number of tiles in row and column.	

# 限制對比度自適應直方圖等化



```
def CLAHE(grayimage, clipLimit=, tileGridSize=):  
    clahe = cv2.createCLAHE(  
        )  
    return clahe.apply(grayimage)
```



# 超級比一比



MIT Lab.  
Multimedia  
Intelligent Technical  
Laboratory



原圖

HE

CLAHE





```
image = cv2.imread('./pout.bmp')
grayimage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
he_image = HE(grayimage)
clahe_image = CLAHE(grayimage)

images = [grayimage, he_image, clahe_image]
titles = ['GRAY', 'HE', 'CLAHE']
plt.figure()
for i in range(3):
    plt.subplot(1, 3, i+1), plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([], plt.yticks([]))
plt.show()
```

# 實作結果



原圖

HE

CLAHE







# Thanks for listening