



Principles and Applications of Digital Image Processing

【Fall, 2020】

工科所 R08525116 吳承哲

Homework 2

Part 1: (30%)

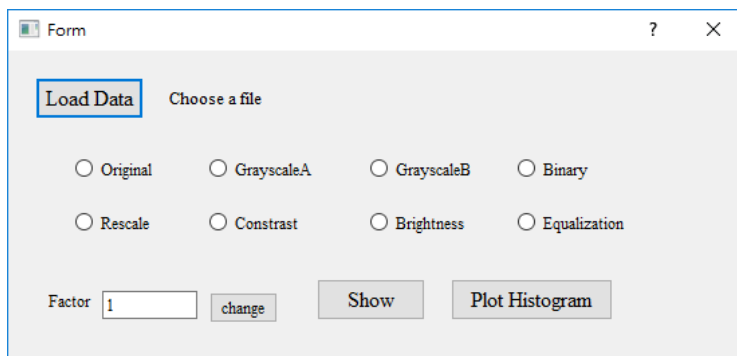
Solve the problems 2.12, 2.16, 2.18, 2.37, 3.12, 3.18 in the textbook.

Part 2: (70%) Image File Reading, Display and Basic Processing

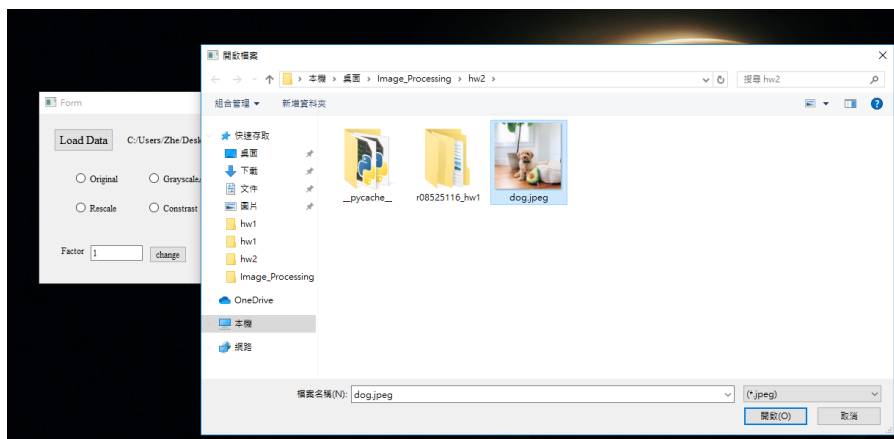
Design a software program that can achieve the following image processing operations:

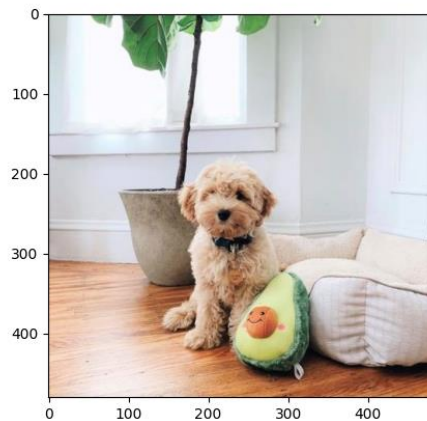
1. Read a color BMP or JPEG image file and display it on the screen. You may use the functions provided by Qt, OpenCV, or MATLAB to read and display an image file. (10%)

```
def load_data(self):  
    filename, filetype = QFileDialog.getOpenFileName(  
        self, "開啟檔案", "./", "(*.jpeg);;(*.bmp)")  
    if filename != "":  
        self.ui.label.setText(filename) # 選好檔案後，把label改成檔案名稱  
        print(filename)  
        self.original_img = cv2.imread(filename) # 路徑不能為中文  
        self.original_img = cv2.cvtColor(  
            self.original_img, cv2.COLOR_BGR2RGB) # convert BGR to RGB
```



使用 GUI 時，如果要調整 factor，請先點選 original，然後新的值，按 change，在選擇有* 的選項。





2. Convert a color image into a grayscale image using the following equations:

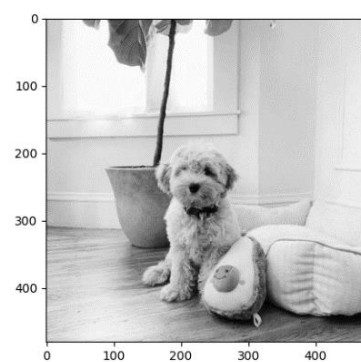
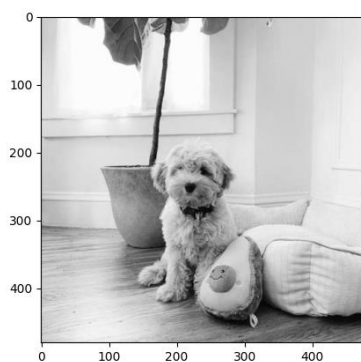
A. $GRAY = (R+G+B)/3.0$

B. $GRAY = 0.299*R + 0.587*G + 0.114*B$

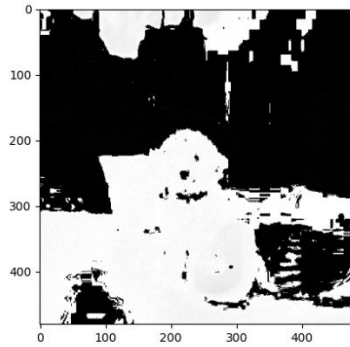
Compare the grayscale images obtained from the above equations. One way to compare the difference between two images is by image subtraction (10%)

```
def grayA(img):  
    img = img.astype(np.float32)  
    img = (img[:, :, 0] + img[:, :, 1] + img[:, :, 2])/3  
    img = img.astype(np.uint8)[:, :, np.newaxis] # 增加channel維度  
    return img  
  
def grayB(img):  
    img = img.astype(np.float32)  
    img = img[:, :, 0]*0.299 + img[:, :, 1]*0.587 + img[:, :, 2]*0.114  
    img = img.astype(np.uint8)[:, :, np.newaxis] # 增加channel維度  
    return img  
  
img_grayA = grayA(img)  
img_grayB = grayB(img)  
img_gray_diff = img_grayA - img_grayB
```

GrayA / GrayB



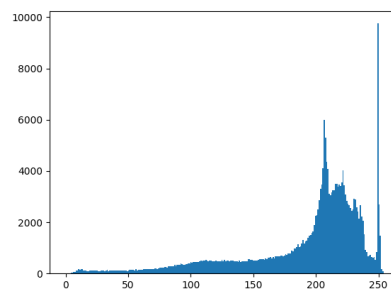
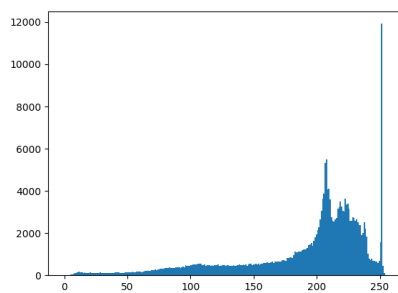
GrayA – GrayB



此兩種灰階值的算法不一樣，得出的 pixel value 會不同，但呈現的圖片沒辦法用肉眼判斷兩張的差異，而經過相減就能明顯比較出來差異。

3. Determine and display the histogram of a grayscale image. (10%)

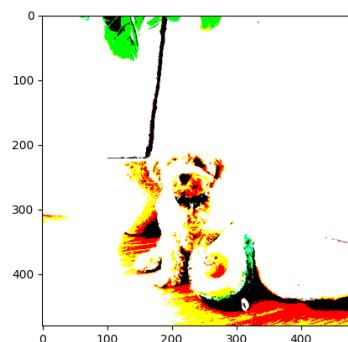
GrayA / GrayB



4. Implement a manual threshold function to convert a grayscale image into a binary image. (10%)

```
def binary(img, threshold):  
    img = img.astype(np.float32)  
    img[img > threshold] = 255  
    img[img <= threshold] = 0  
    img = img.astype(np.uint8)  
    return img
```

Threshold = 100

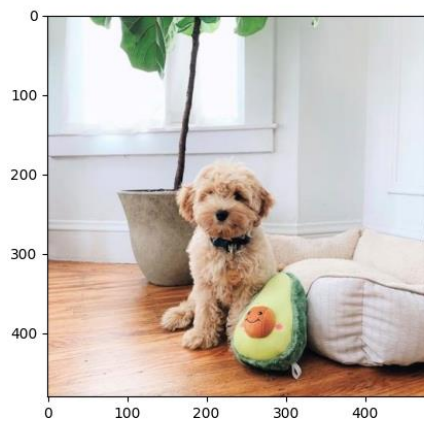


5. Implement a function to adjust the spatial resolution (enlarge or shrink) and grayscale levels of an image. Use interpolation on enlarging the image. (10%)

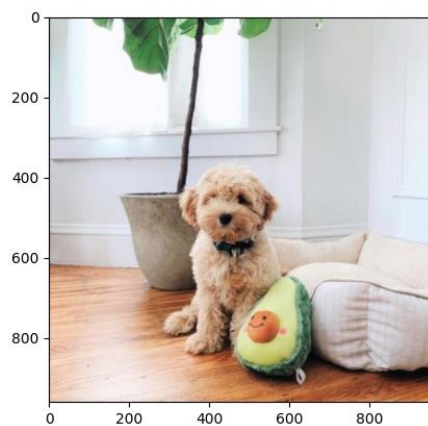


```
def rescale(img, factor):  
    size = img.shape[0]  
    rescale_factor = factor # 放大倍率  
    rescale_size = int(size*rescale_factor)  
    img_rescale = cv2.resize(img, (rescale_size, rescale_size),  
                             interpolation=cv2.INTER_LINEAR) # 雙線性插植  
    return img_rescale
```

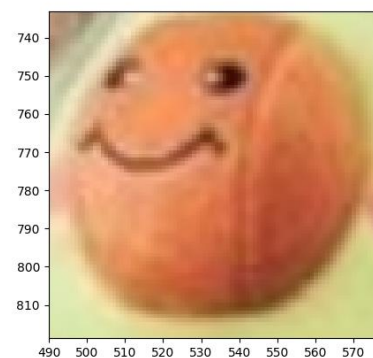
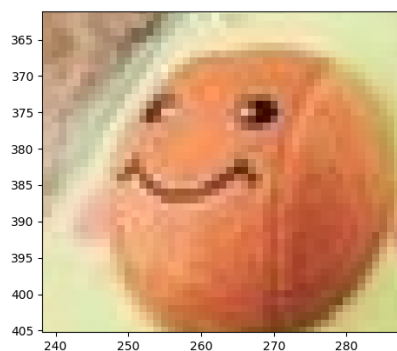
Original



Enlarge 2x



Compare Original / Enlarge 2x



經過 rescale 的圖片在放大檢視後，相較於原圖比較圓滑。

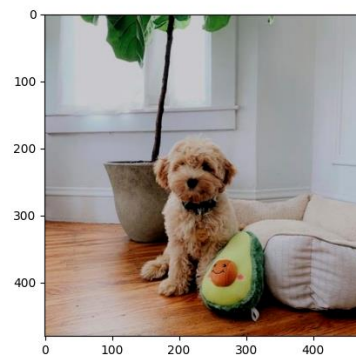
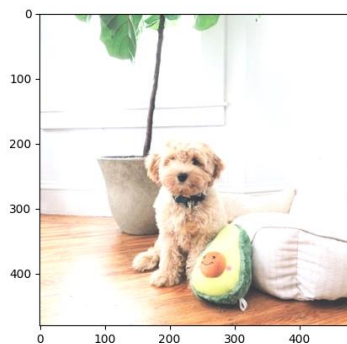


6. Implement a function to adjust the brightness and contrast of an image. (10%)

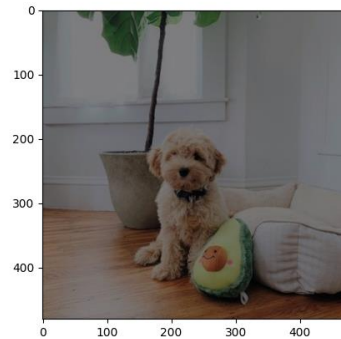
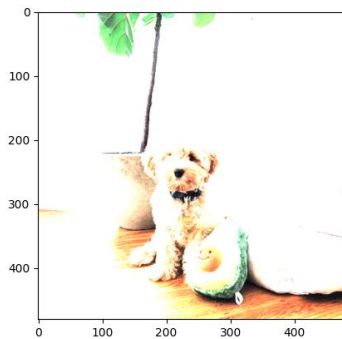
```
def contrast(img, factor):  
    img = img.astype(np.float32)  
    img *= factor  
    img[img >= 255] = 255  
    img[img <= 0] = 0  
    img = img.astype(np.uint8)  
    return img
```

```
def brightness(img, factor):  
    img = img.astype(np.float32)  
    img += factor  
    img[img >= 255] = 255  
    img[img <= 0] = 0  
    img = img.astype(np.uint8)  
    return img
```

Brightness (+50 / -50)

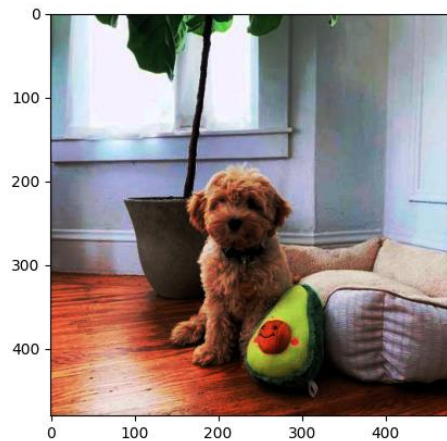


Contrast (x2 / x0.5)



7. Implement a histogram equalization function for automatic contrast adjustment. (10%)

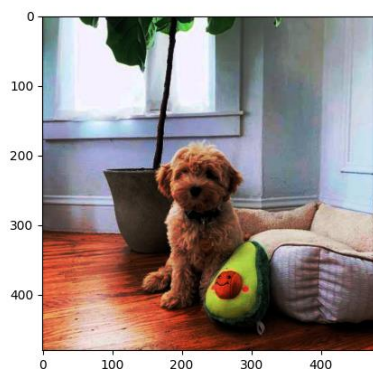
```
def equalized(img):  
    hist, bins = np.histogram(img.flatten(), 256, [0, 256]) # 得到直方圖的分布  
    cdf = hist.cumsum() # 逐元素進行累加  
    cdf_m = np.ma.masked_equal(cdf, 0) # 把 0 遮掉  
    cdf_m = ((cdf_m - cdf_m.min())*255) / (cdf_m.max() - cdf_m.min())  
    cdf = np.ma.filled(cdf_m, 0).astype(np.uint8)  
    img_equal = cdf[img]  
    return img_equal
```



Test your image processing functions with various images and compare the processed image with those processed with Photoshop, PhotoImpact, or other similar commercial image processing software.

My work / PhotoImpact

Equalization



Notes:

1. Please submit your programs and report to the CEIBA course website before **Oct. 14 (2:20PM)**.
2. Late submission will have a penalty of 10% discount per day of your homework total score toward a maximum of 50% discount. No late submission over five days will be accepted.