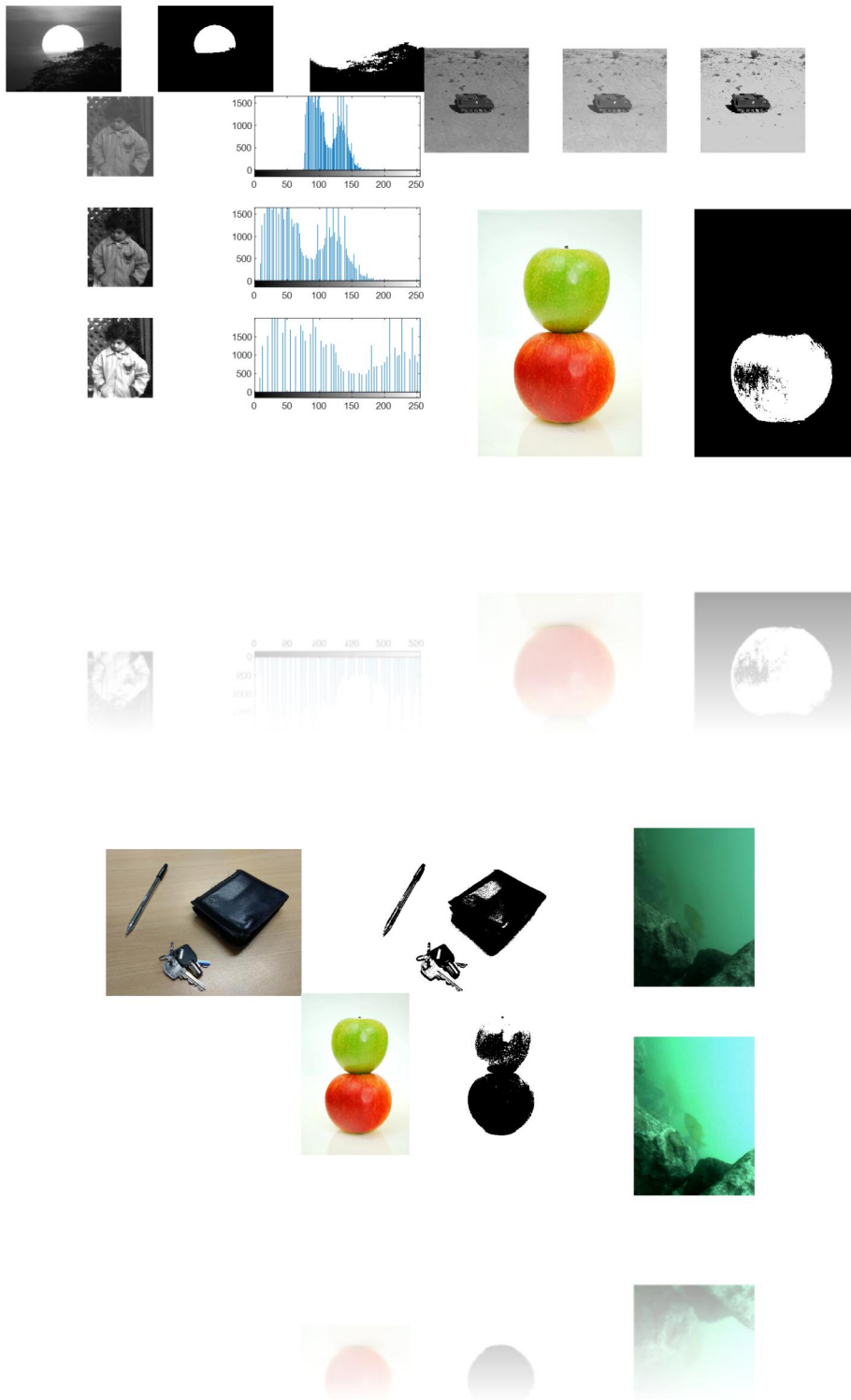


一、图像的基本操作



笔记:

1. Reading images.

```
f = imread('...jpg');
```

```
k = imfinfo('...jpg');
```

2. Image presentation.

$[M \times N \times D]$
↓ ↓ ↓
rows columns \rightarrow $\begin{cases} 1: \text{greyscale and binary images.} \\ 3: \text{RGB.} \end{cases}$

△ data classes: double, uint8, uint16, uint32, int8, int16, int32, single, char, logical.

△ $\begin{cases} \text{red channel: } \text{imr} = \text{im}(:, :, 1); \\ \text{green channel: } \text{img} = \text{im}(:, :, 2); \\ \text{blue channel: } \text{imb} = \text{im}(:, :, 3); \end{cases}$

△ $\text{im_crop} = \text{im}(100:150, 120:170, :);$

△ $\text{im}_2 = \text{cat}(3, \text{im}(:, :, 3), \text{im}(:, :, 2), \text{im}(:, :, 1));$ 交换红蓝channel.

△ $\text{im_subsamp} = \text{im}(1:2:\text{end}, 1:3:\text{end}, :);$ 采样

△ $\text{im_flipped} = \text{im}(\text{end}:-1:1, :, :);$ 垂直翻转

3. Displaying Images.

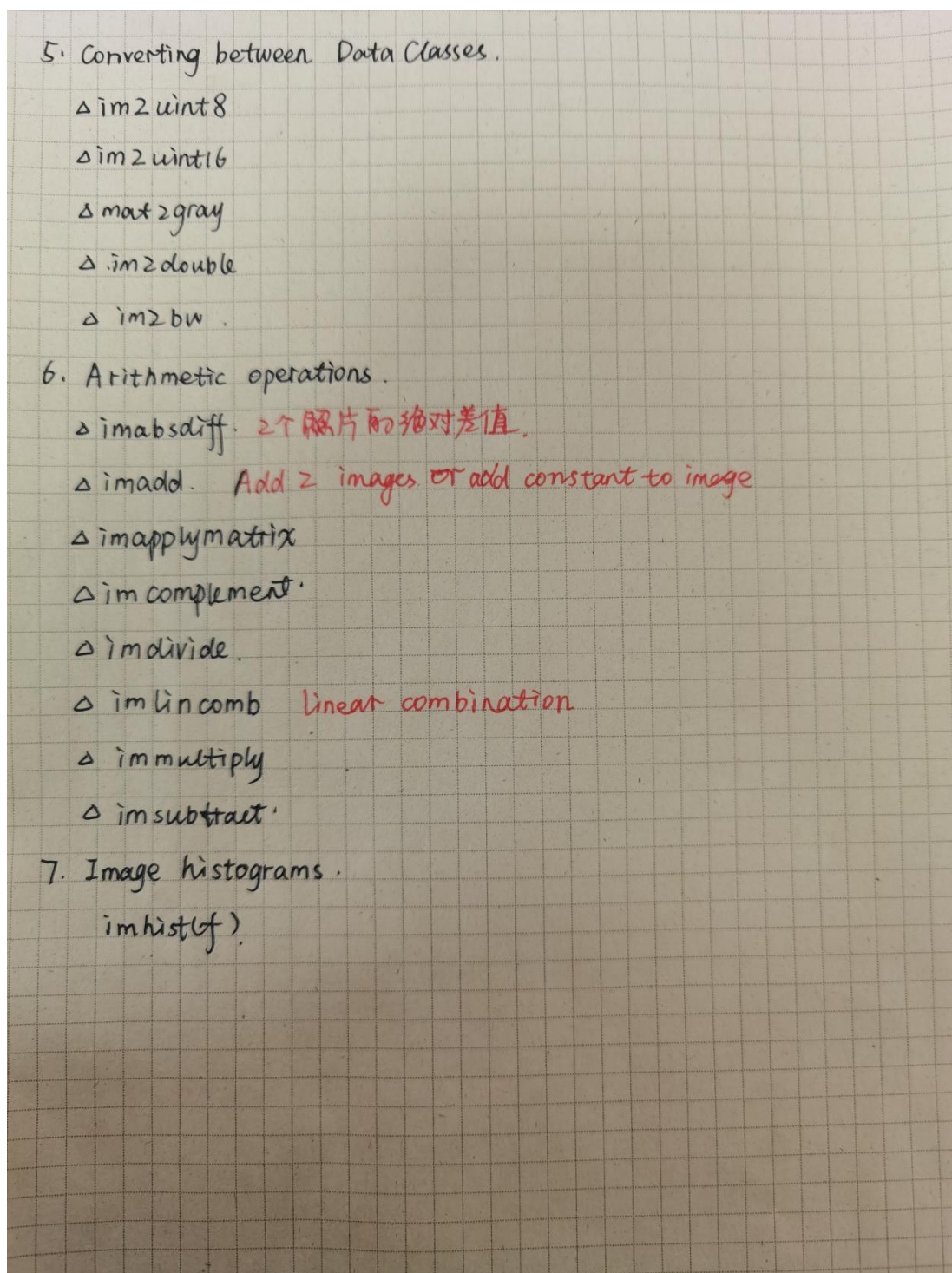
△ $\text{imshow}(f, [low\ height])$ 小于low的显示为黑, 大于height的显示为白.

△ impixelinfo

△ imdistline .

4. Writing Images.

△ $\text{imwrite}(f, \text{filename})$



Tutorial Activity 1: Basic Image Operations

题目: 写一个函数实现输入一个坐标将对应位置的图标切割并显示出来; 根据灰度值判断这个位置是否是空 (即白色)

代码 1 (matlab):

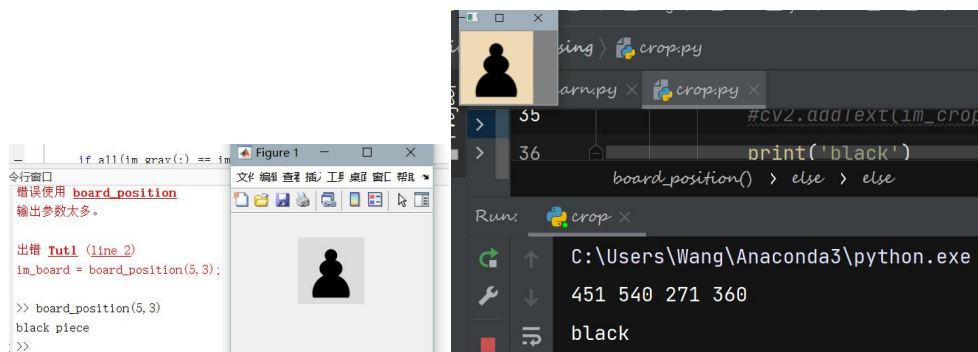
```
1. function board_position(row,col)
2.
3.     % Load Image
4.     im = imread('chess.png');
5.     imshow(im)
6.
7.     % Obtain the size of the image
8.     [rows, columns] = size(im);
9.
10.    % Calculate the board size
11.    board_size = rows/8; %total 8
12.
13.    % Calculate the require position
14.    row_start = row * board_size + 1; %begin with 0
15.    col_start = col * board_size + 1;
16.
17.    row_end = row_start + board_size - 1;
18.    col_end = col_start + board_size - 1;
19.
20.    % Crop region starting at (100,120)
21.    im_crop = im(row_start : row_end,col_start:col_end,:);
22.
23.    im_gray = rgb2gray(im_crop);
24.
25.    % Detect how many zero(white) values in the image
26.    n = nnz(im_gray); %返回矩阵 X 中非零元素的个数
27.    m = numel(im_gray); %返回元素个数
28.
29.    if all(im_gray(:) == im_gray(1))
30.        disp('Empty board');
31.    else
32.        white_pixel = (n/m) * 100;
33.
34.        if white_pixel > 80
35.            disp('white piece');
36.        else
37.            disp('black piece');
38.        end
39.    end
40.
41.    % Display the cropped image
42.    imshow(im_gray)
43.
44. end
```


代码 2 (python) :

```
1.  #crop an area and determine black or white
2.
3.  import cv2
4.  import numpy as np
5.  import pandas as pd
6.
7.  im = cv2.imread('chess.png')
8.
9.
10. cv2.imshow('chess',im)
11. cv2.waitKey(0)
12. def board_position(im,row,col):
13.     [rows,columns,n] = np.shape(im)
14.     board_size = int(rows/8)
15.     row_st=row*board_size+1
16.     col_st=col*board_size+1
17.     row_end=row_st+board_size-1
18.     col_end=col_st+board_size-1
19.     print(row_st,row_end,col_st,col_end)
20.
21.     im_crop=im[row_st:row_end,col_st:col_end]
22.     print(im_crop)
23.
24.
25.
26.     cv2.imshow('cropped',im_crop)
27.
28.     im_gray =cv2.cvtColor(im_crop,cv2.COLOR_BGR2GRAY)
29.
30.     n = np.count_nonzero(im_gray)
31.     m = np.size(im_gray)
32.     s = im_gray[0]
33.     im_gray = [i-im_gray[0] for i in im_gray ]
34.     if np.count_nonzero(im_gray) == np.size(im_gray):
35.         #cv2.putText(im_crop,'empty',(300,300),cv2.FONT_HERSHEY_COMPLEX,1,(0,150,0),1)
36.         print('empty')
37.     else:
38.         white_pixel = int(n/m*100)
39.         print(white_pixel)
40.         if white_pixel>80:
41.             #cv2.putText(im_crop,'white',(300,300),cv2.FONT_HERSHEY_COMPLEX,1,(0,150,0),1)
42.             print('white')
```

```
43.         else:
44.             #cv2.putText(im_crop,'black',(300,300),cv2.FONT_HERSHEY_COMPLEX,1,(0,150,0),1)
45.             print('black')
46.         im_gray = [i + s for i in im_gray]
47.
48.         cv2.waitKey(0)
49.         print(np.shape(im_gray))
50.         return im_gray
51.
52.
53.     '''a = board_position(im,7,1)
54.     df = pd.DataFrame(a)
55.     writer = pd.ExcelWriter('hahaha.xlsx')
56.     df.to_excel(writer, 'page_1')
57.     writer.save()'''
58.     a = board_position(im, 7, 1)
59.     df = pd.DataFrame(a)
60.     writer = pd.ExcelWriter('hahaha.xlsx')
61.     df.to_excel(writer, 'page_1')
62.     writer.save()
```

结果:



Tutorial Activity 2: Spot the Difference !

题目：找不同并数出来

代码 1: (matlab)

```
1. %% Activity 2: Spot the Difference
2. im = imread('spot_the_difference.png');
3. im_info = imfinfo('spot_the_difference.png') %该函数用于获取一张图片的具体信息。这些具体信息包括图片的格式、尺寸、颜色数量、修改时间等等
4. im1 = im(:,1:350,:);
5. im2 = im(:,351:700,:); %两张图分别占据一半的宽度
6. im_diff=im1-im2; %找差别
```

```
7.   im_diff=rgb2gray(im_diff);
8.   im_diff=im_diff>40; %True 则为 1, False 为 0
9.
10.
11.   im_diff=cat(3,im_diff*255,zeros(size(im_diff)),zeros(size(im_diff))); %构造 n 维数组, n=3;差别过大的像
    素标注出来且为红色
12.
13.
14.   se=strel('square',8);
15.   imdilate(im_diff,se)
16.   disp(bwconncomp(im_diff))
17.   im_diff=uint8(im_diff);
18.
19.   im_diff = imlincomb(0.4,im1,10,im_diff,'uint8');
20.   %%计算线性组合 0.4*im1+10*im_diff, 应该是调节亮度的
21.   figure;
22.   subplot(1,3,1);imshow(im1);
23.   subplot(1,3,2);imshow(im2);
24.   subplot(1,3,3);imshow(im_diff);
```

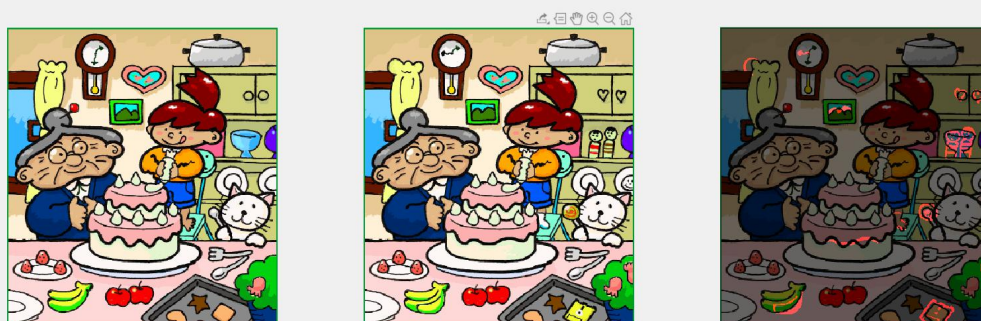
代码二（python）：

```
1.   import cv2
2.   import numpy as np
3.   import pandas as pd
4.
5.   im = cv2.imread('dif.png')
6.
7.   im1 = im[:, 0:350, :]
8.   im2 = im[:, 350:701, :]
9.
10.  '''im_diff = (np.asarray(im1)-np.asarray(im2)).tolist()
11.  print(np.shape(im_diff))'''
12.  im_diff = cv2.subtract(im1, im2)
13.  cv2.imshow('diff', im_diff)
14.  im_diff = cv2.cvtColor(im_diff, cv2.COLOR_RGB2GRAY)
15.
16.
17.  sub = im_diff.reshape(1, np.size(im_diff))
18.  sub = np.where(sub > 40, 1, 0)
19.  im_diff = sub.reshape(np.shape(im_diff))
20.  im_diff_cw=255*np.array(im_diff, dtype='uint8')
21.  kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (20,15), anchor=None)
22.  im_diff_cw = cv2.dilate(im_diff_cw,kernel)
```

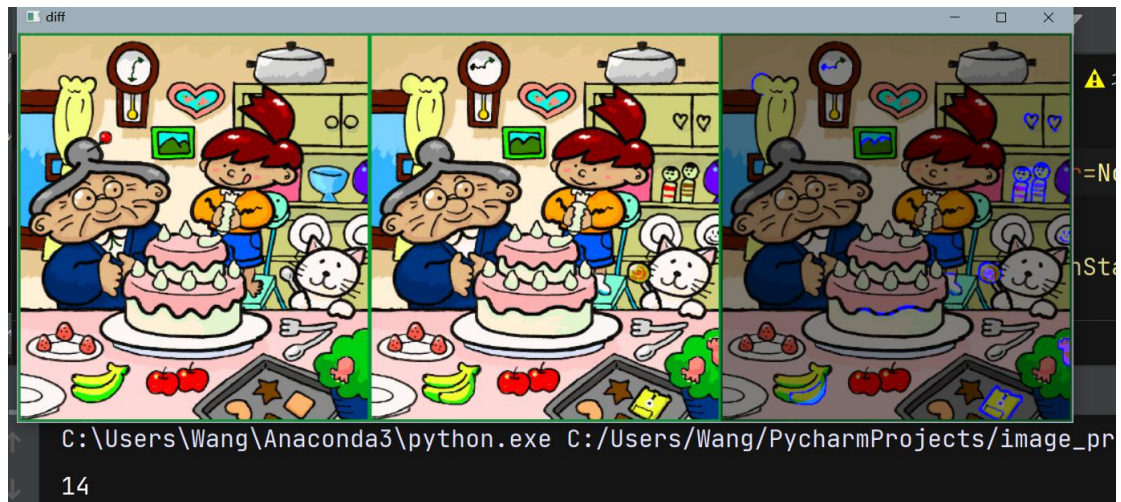
```
23. num_labels, labels, stats, centroids = cv2.connectedComponentsWithStats(im_diff_cw, connectivity=8)

24. print(num_labels)
25. im_diff = im_diff.tolist()
26. a,b = np.shape(im_diff)
27.
28.
29. ##
30. for i in range(a):
31.     for j in range(b):
32.         im_diff[i][j] = [im_diff[i][j]*255,0,0] #opencv 的颜色体系是 BGR
33.
34. print(np.shape(im_diff))
35. im_diff=np.array(im_diff, dtype='uint8')
36.
37. im_diff = cv2.addWeighted(im_diff,4,im2,0.5,0)
38. im_diff = np.hstack((im1,im2,im_diff))
39.
40.
41. cv2.imshow('diff', im_diff)
42.
43. cv2.waitKey(0)
```

结果:



```
Connectivity: 26
ImageSize: [386 350 3]
NumObjects: 67
PixelIdxList: {1×67 cell}
```

备注：使用了膨胀函数，但是由于一个物体的差异点距离太远，导致被判断成多个物体；python 的结果更加理想，也许是二者的 dilate 函数的算法有差异吧