Notice (2/2)

- Python (recommended)
  - ➢ Python 3.7 (https://www.python.org/downloads/)
  - ➢ opencv-contrib-python (3.4.2.17)
  - ➢ Matplotlib 3.1.1
  - ➢ UI framework: pyqt5 (5.15.1)

- C++ (check MFC guide in ftp)
  - ➢ OpenCV 3.3.1 (https://opencv.org/release.html)
  - ➢ Visual Studio 2015 (download from http://www.cc.ncku.edu.tw/download/)
  - ➢ UI framework: MFC

Assignment scoring (Total: 100%)

1. (20%) Image Processing   (出題：Sam)
   - 1.1 (5%) Color Separation
   - 1.2 (5%) Color Transformation
   - 1.3 (5%) Color Detection
   - 1.4 (5%) Blending

2. (20%) Image Smoothing   (出題：Jack)
   - 2.1 (6%) Gaussian blur
   - 2.2 (7%) Bilateral filter
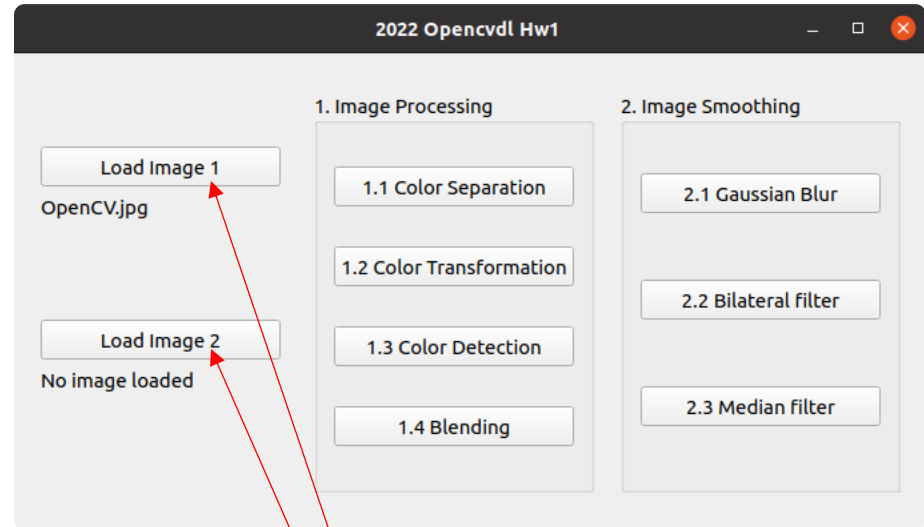   - 2.3 (7%) Median filter

3. (20%)   (出題：)

4. (20%)   (出題：)

5. (20%) Training Cifar10 Classifier Using Resnet101   (出題：Wen)

2022 Opencvdl Hw1

1. Image Processing
- Load Image 1
- OpenCV.jpg
- 1.1 Color Separation
- 1.2 Color Transformation
- Load Image 2
- No image loaded
- 1.3 Color Detection
- 1.4 Blending

2. Image Smoothing
- 2.1 Gaussian Blur
- 2.2 Bilateral filter
- 2.3 Median filter
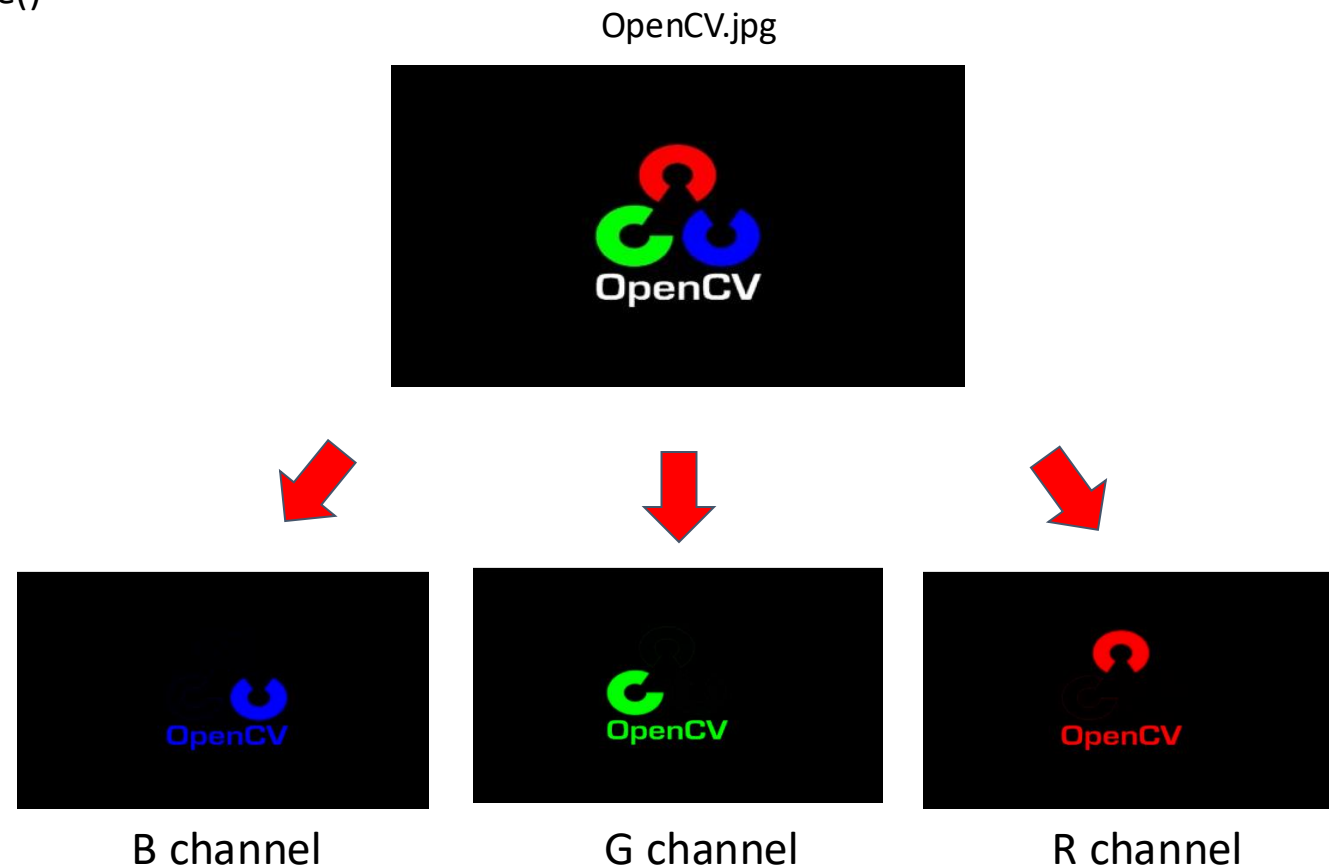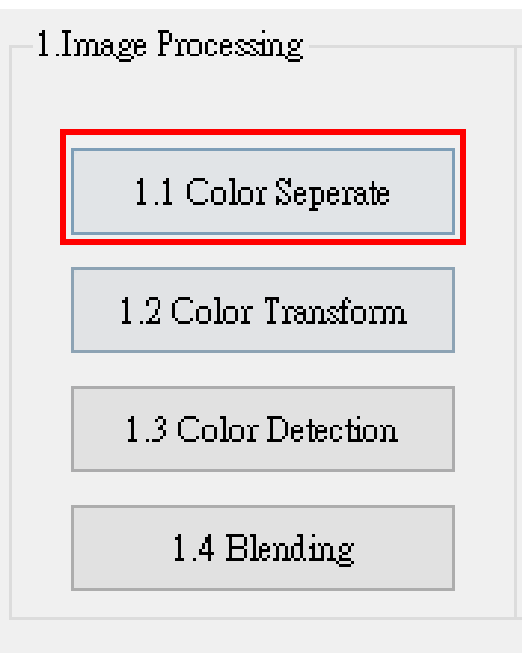
Don't fix your image path
(There is another dataset for demonstration)

1. Image Processing (20%)

1.1 (5%) Color Separation

1.2 (5%) Color Transformation

1.3 (5%) Color Detection

1.4 (5%) Blending

1.Image Processing

1.1 Color Seperate

1.2 Color Transform

1.3 Color Detection

1.4 Blending
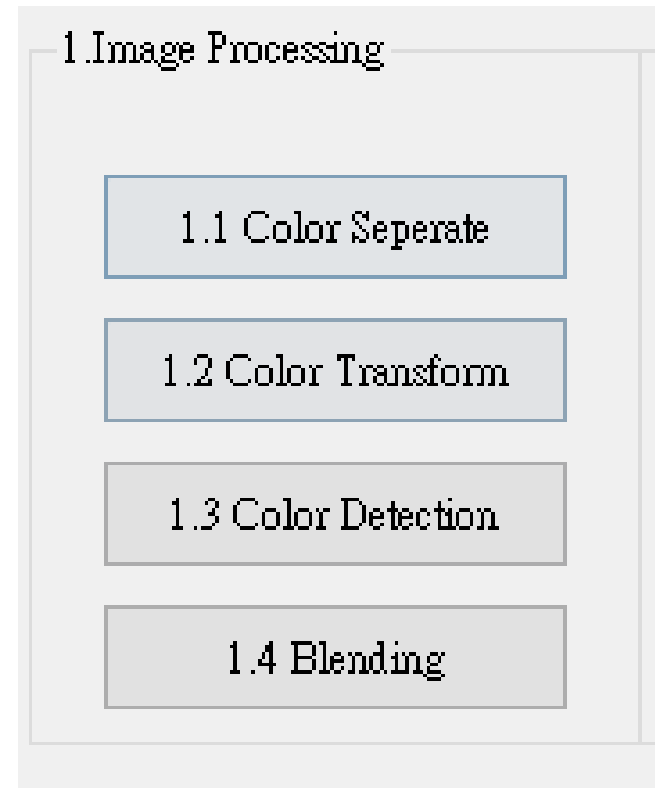
1.1 Color Separation (5%)

（出題：Sam)

- ☐ Given: a color image, "OpenCV.jpg"
- ☐ Q: 1) Extract 3 channels of the image BGR to 3 separated channels and show the result images.

- ☐ Hint:
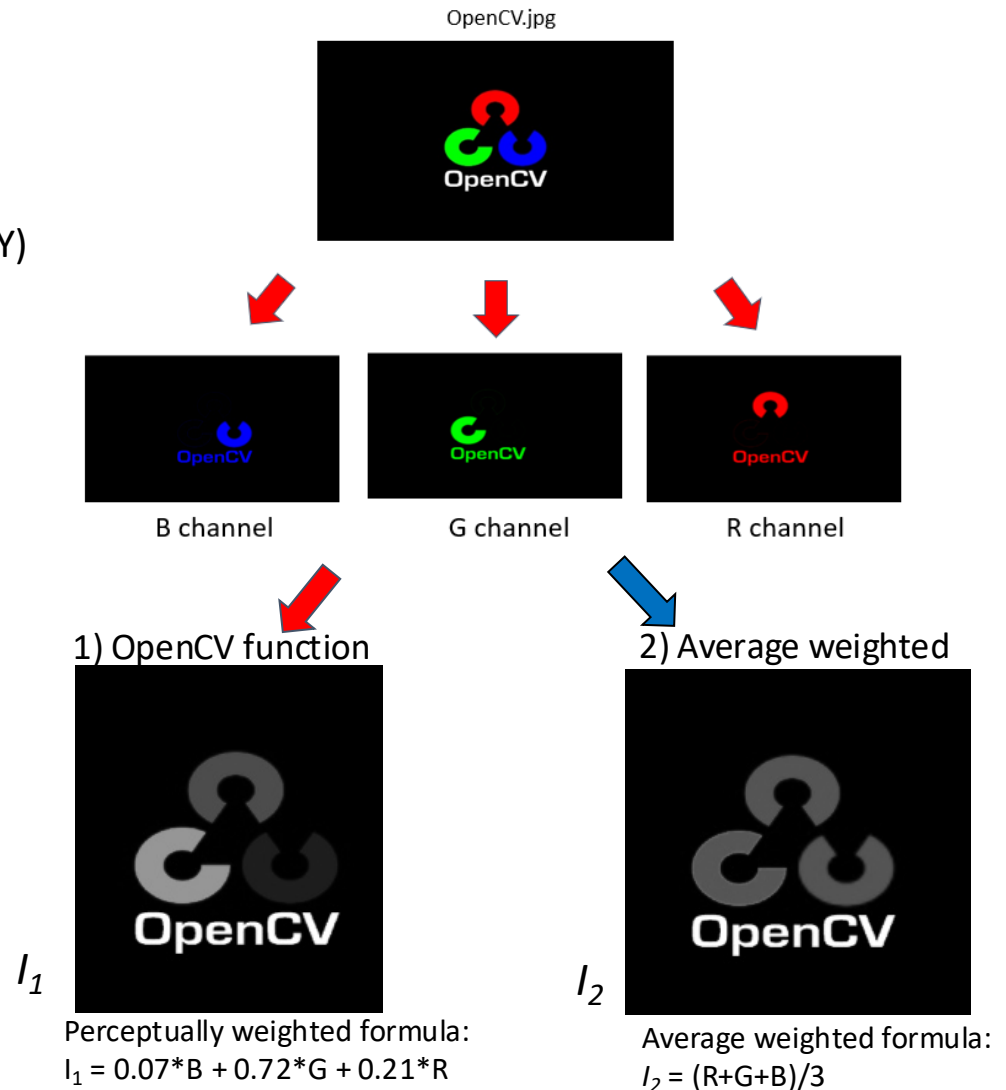 - Textbook Chapter 3, p.31 ~ p.49
  - cv2.split(), cv2.merge()

OpenCV.jpg

1.Image Processing

1.1 Color Seperate

1.2 Color Transform

1.3 Color Detection

1.4 Blending

B channel   G channel   R channel

1.2 Color Transformation (5%)

- ☐ Given: 1 color image: "OpenCV. jpg"
- ☐ Q: 1) Transform "OpenCV.jpg" into grayscale image I_1 by calling OpenCV function directly.
  2) Merge BGR separated channel images from problem 1.2 into grayscale image I_2 by $I_2 = (R+G+B)/3$.
 3) Show the above 2 results.

- ☐ Hint:
  - Textbook Chapter 3, p.56 ~ p.59
  - cv2.cvtColor(..., cv2.COLOR_BGR2GRAY)

OpenCV.jpg

1.Image Processing

1.1 Color Seperate

1.2 Color Transform

1.3 Color Detection

1.4 Blending

B channel   G channel   R channel

1) OpenCV function

2) Average weighted

I_1

I_2

Perceptually weighted formula:
$I_1 = 0.07*B + 0.72*G + 0.21*R$

Average weighted formula:
$I_2 = (R+G+B)/3$

1.3 Color Detection (5%)

(出題：Sam)

☐ Given: 1 color image: "OpenCV. jpg"

☐ Q: 1) Transform "OpenCV.jpg" from BGR format to HSV format.
   2) Generate mask by calling :
   cv2.inRange(hsv_img , lower_bound , upper_bound)
   3) Detect **Green** and **White** color in the image by calling :
   cv2.bitwise_and(bgr_img , bgr_img , mask)
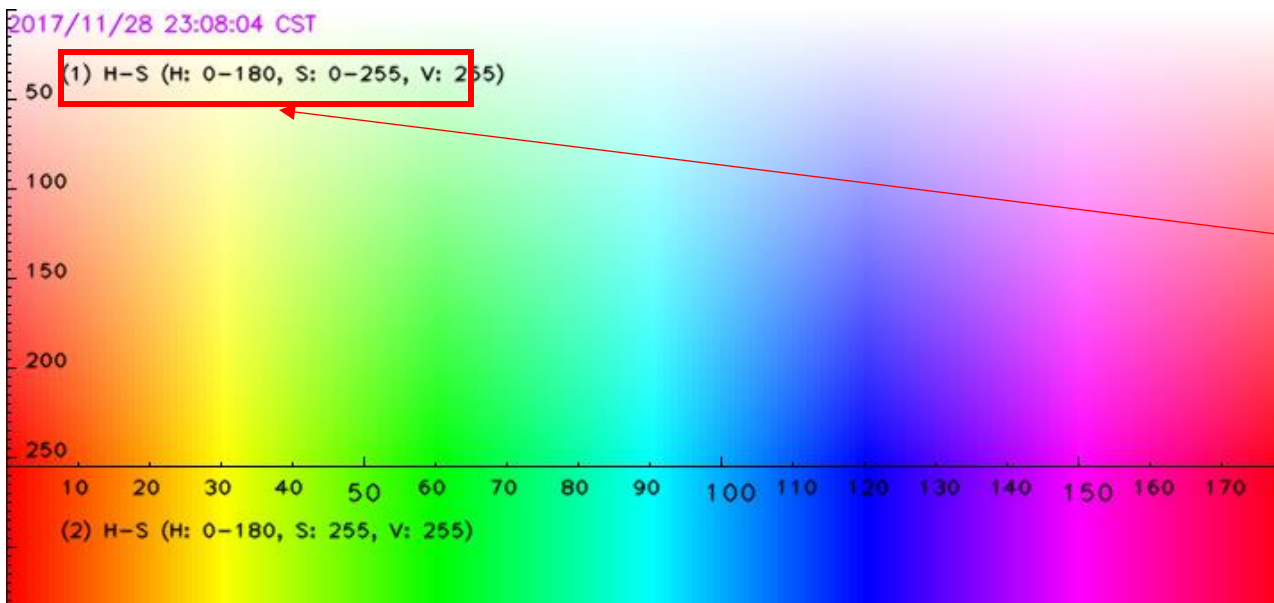   4) Show the result

1.3 Color Detection (5%)

(出題：Sam)
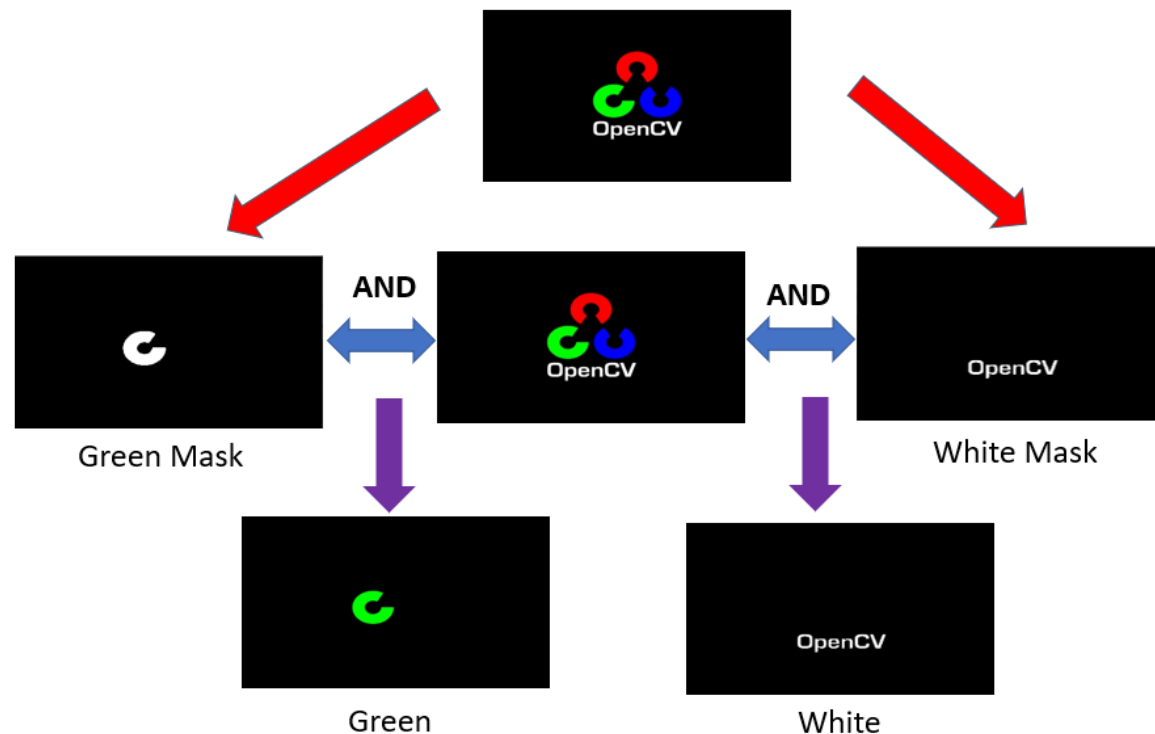
☐ Hint:

- cv2.cvtColor(…, cv2.COLOR_BGR2HSV)
- cv2.inRange(hsv_img , lower_bound , upper_bound)
- cv2.bitwise_and(bgr_img , bgr_img , mask)
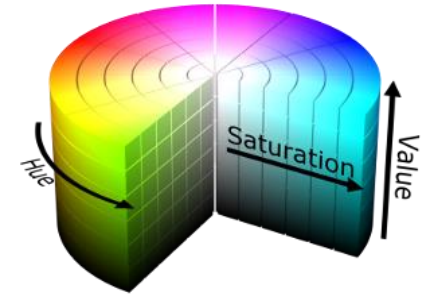
HSV values ranges between (0–180, 0–255, 0–255)

H(Hue) : x axis
**S(Saturation) : y axis**
V(Value) : 255

Using this range should be fine
Green Range : (40-80,50-255,20-255)
White Range : (0-180,0-20,200-255)

REF: https://medium.com/@gowtham180502/how-to-detect-colors-using-opencv-python-98aa0241e713

1.4 Blending (5%)

(出題：Sam)

☐ Given: 2 images, "Dog_Strong.jpg" and "Dog_Weak.jpg"

☐ Q: 1) Combine two images (Dog_Strong.jpg and Dog_Weak.jpg).
2) Use Trackbar to change the weights and show the result in the new window.

☐ Hint:

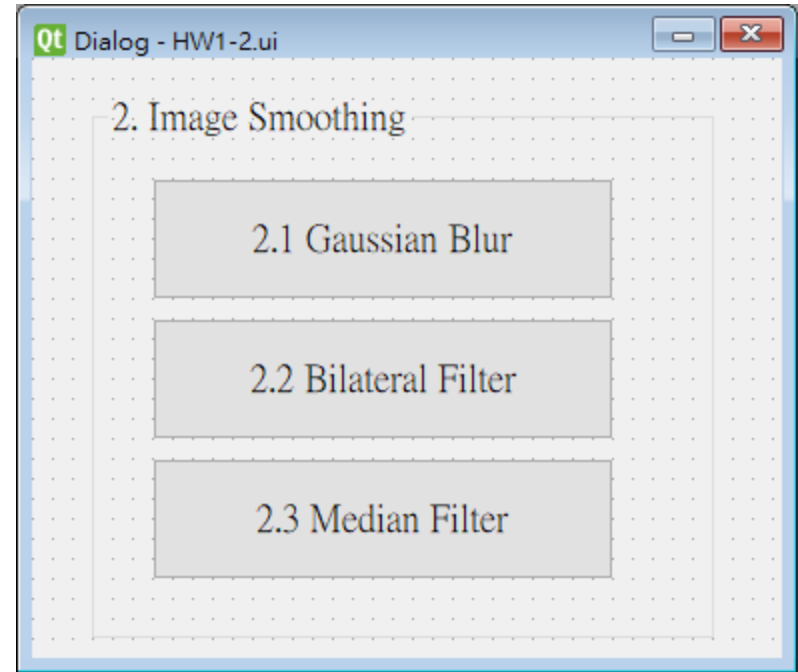- Textbook Chapter 3, p. 50 ~ 52

- cv2.addWeighted(), cv2.createTrackbar()

Demo:
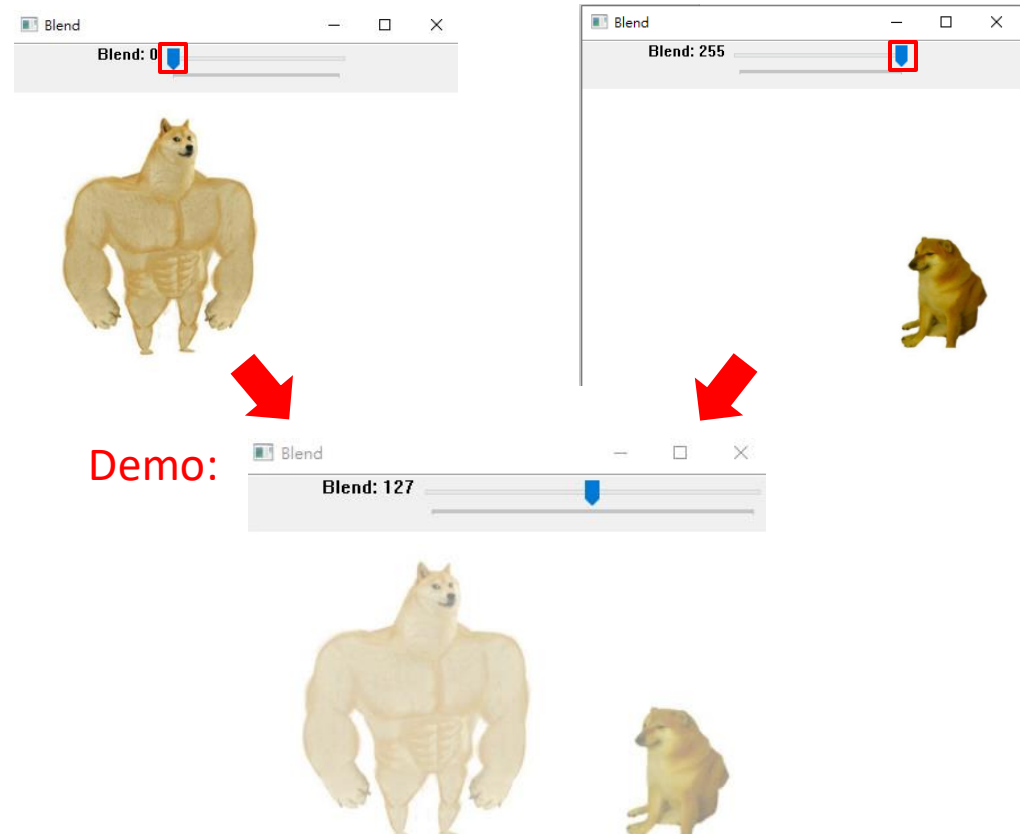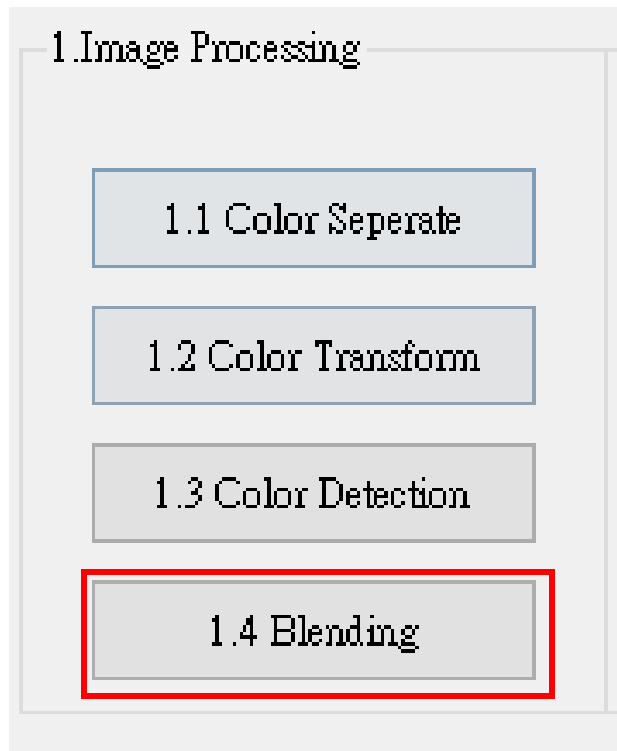
2. Image Smoothing (20%)

(出題：Jack)

2.1 (6%) Gaussian blur

2.2 (7%) Bilateral filter

2.3 (7%) Median filter

2.1 Gaussian Blur

(出題：Jack)

Given: "image1.jpg

Define: gaussian magnitude 0 ~ 10,

Condition1. magnitude = m > 0 then Apply gaussian filter k x k to "image1.jpg" (k=2m+1)

Condition2. magnitude = m = 0 then output "image1.jpg"

Q: Use Trackbar to change the magnitude and show the result in the popup window.

Hint:

- Textbook Chapter 3, p. 50 ~ 52, p.109~115
- cv2.GaussianBlur(), cv2.createTrackbar()

	1	2	1	
1/16	2	4	2	
	1	2	1	

	1	4	7	4	1
	4	16	26	16	4
1/273	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

	0	0	1	2	1	0	0
	0	3	13	22	13	3	0
	1	13	59	97	59	13	1
1/1003	2	22	97	159	97	22	2
	1	13	59	97	59	13	1
	0	3	13	22	13	3	0
	0	0	1	2	1	0	0

Qt Dialog - HW1-2.ui

2. Image Smoothing

2.1 Gaussian Blur

2.2 Bilateral Filter

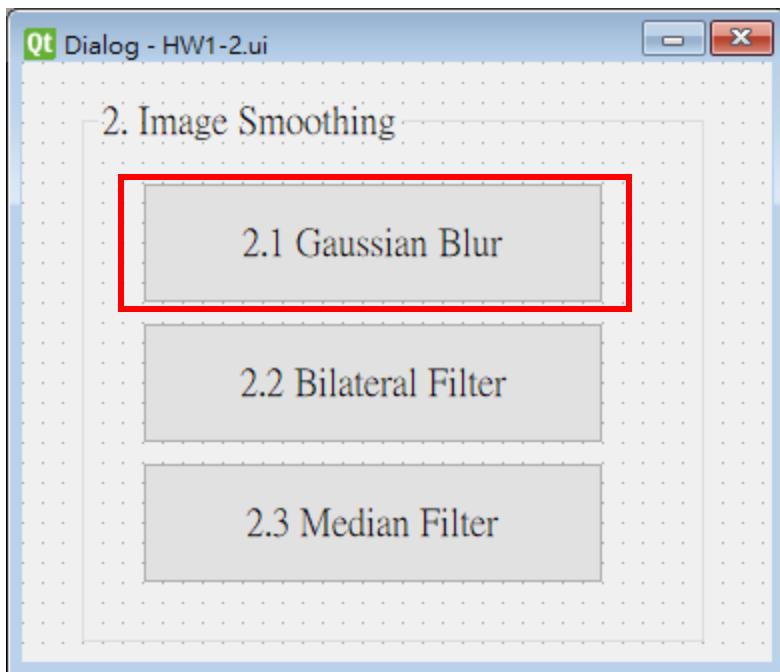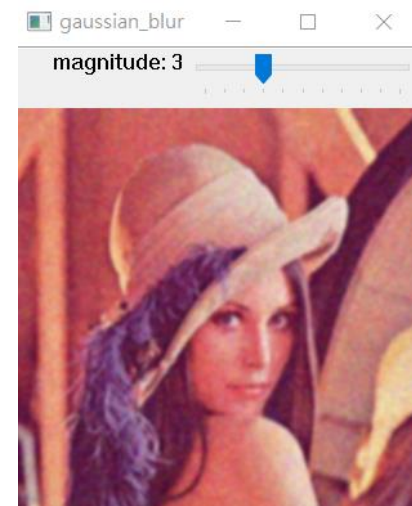2.3 Median Filter

image1.jpg

gaussian_blur

magnitude: 3

2.2 Bilateral Filter

(出題：Jack)

Given: "image1.jpg

Define: Bilateral magnitude 0 ~ 10, sigmaColor = 90 and sigmaSpace = 90.
Condition1. magnitude = m > 0 then Apply bilateral filter k x k to "image1.jpg" (k=2m+1)
Condition2. magnitude = m = 0 then output "image1.jpg"

Q: Use Trackbar to change the magnitude and show the result in the popup window.

Hint:

- Textbook Chapter 3, p. 50 ~ 52, p.109~115

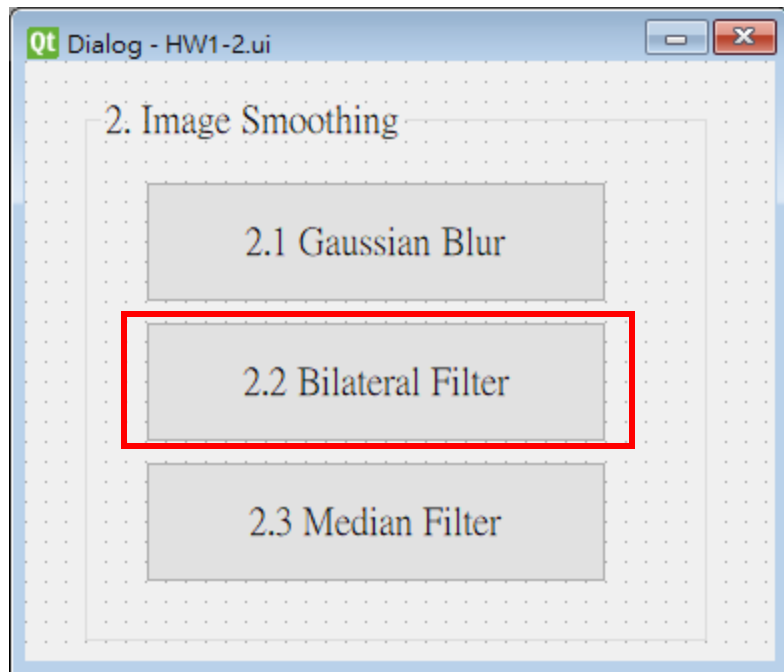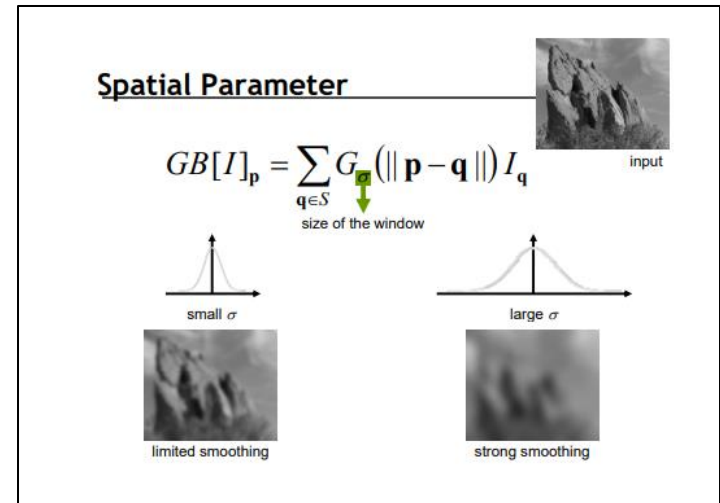- cv2.bilateralFilter(), cv2.createTrackbar()

Spatial Parameter

$$GB[I]_{\mathbf{p}} = \sum_{\mathbf{q} \in S} G_{\sigma}\left(\|\mathbf{p} - \mathbf{q}\|\right) I_{\mathbf{q}}$$

size of the window

input

small σ   large σ

limited smoothing   strong smoothing

2. Image Smoothing

2.1 Gaussian Blur

2.2 Bilateral Filter

2.3 Median Filter

image1.jpg

bilateral_filter

magnitude: 3

2.3 Median Filter

Given: "image2.jpg

    Define: Median magnitude 0 ~ 10,

      Condition1. magnitude = m > 0 then Apply median filter k x k to "image2.jpg" (k = 2m+1)

      Condition2. magnitude = m = 0 then output "image2.jpg"

  Q: Use Trackbar to change the magnitude and show the result in the popup window.

Hint:

- Textbook Chapter 3, p. 50 ~ 52, p.109~115

- cv2.medianBlur(), cv2.createTrackbar()

Median filter example

223	186	114
204	161	106
219	194	138

106 114 138 161 186 194 204 219 223