# 影像處理、電腦視覺及深度學習概論
# (Introduction to Image Processing, Computer Vision and Deep Learning)

## Homework 1

TA:

Kevin: i1007673219@gmail.com

Office Hour: 19:00~21:00, Mon.

09:00~11:00, Wed.

At CSIE 9F Robotics Lab.

# Notice (1/2)

- Copying homework is strictly prohibited!! Penalty: Grade will be zero for both persons!!
- If the code can't run, you can come to our Lab within one week and show that your programming can work. Otherwise you will get zero!!
- Due date => 2019/11/07 (Thu.) 23:59:59
  - No delay. If you submit homework after deadline, you will get 0.
- Upload to => 140.116.154.1 -> Upload/Homework/OpenCv_Hw1
  - User ID: opencvdl2019 Password: opencvdl2019
- Format
  - Filename: Hw1_StudentID_Name_Version.rar
    - Ex: Hw1_F71234567_林小明_v1.rar
    - If you want to update your file, you should update your version to be v2, ex: Hw1_F71234567_林小明_v2.rar
  - Content: project folder*( including the pictures )
    *note: remove your "Debug" folder to reduce file size

# Notice (2/2)

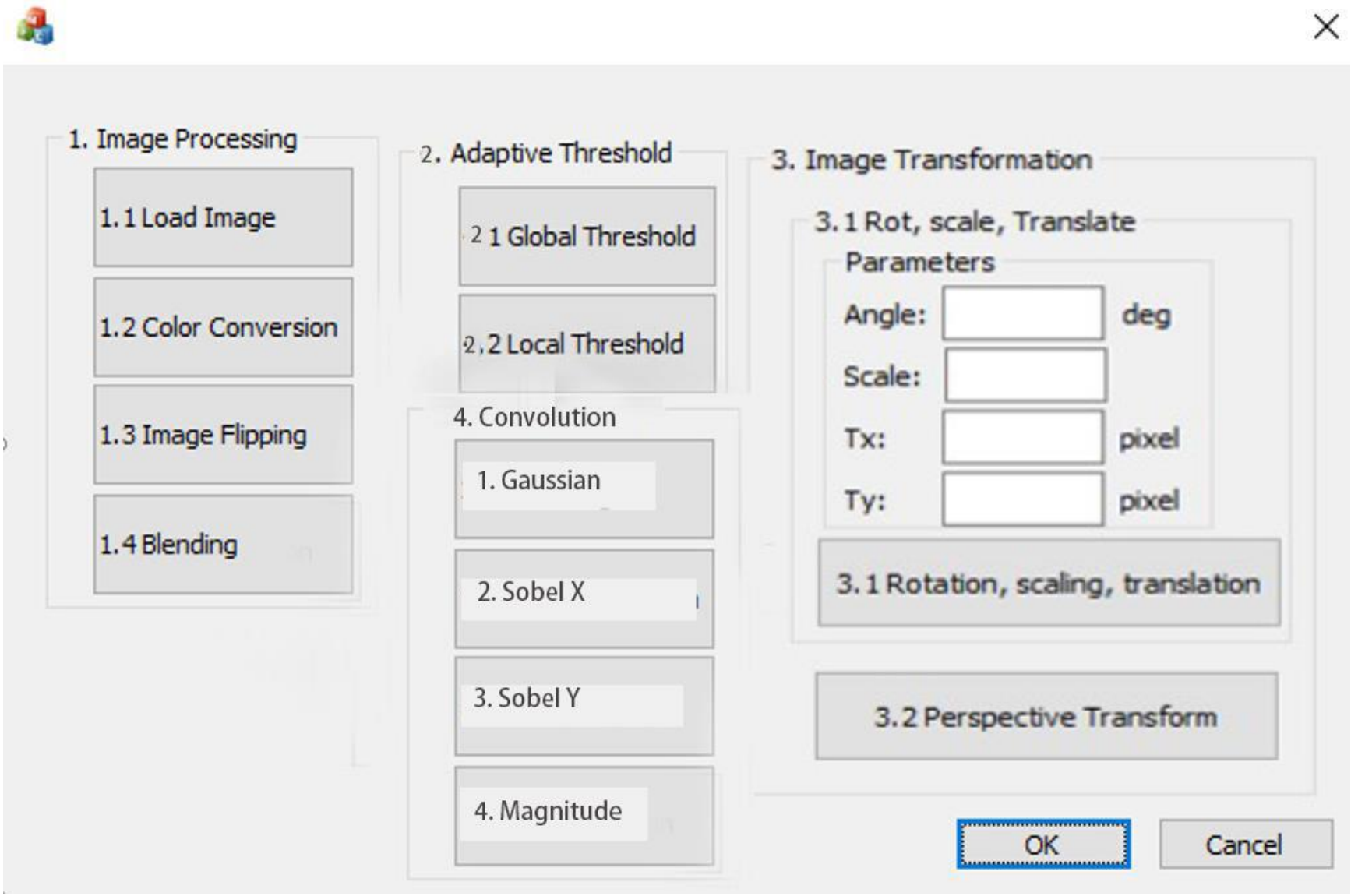
- C++ (check MFC guide in ftp)
  - OpenCV 3.3.1 (https://opencv.org/release.html)
  - Visual Studio 2015 (download from http://www.cc.ncku.edu.tw/download/)
  - UI framework: MFC
- Python
  - Python 3.7 (https://www.python.org/downloads/)
  - Tensorflow 2.0 / PyTorch 1.3.0
  - opencv-contrib-python (3.4.2.17)
  - Matplotlib 3.1.1
  - UI framework: pyqt5 (5.11.3)

# Assignment scoring (Total: 100%)

0. GUI
1. (10%) Image Processing (出題：Shaku)
   - 1.1 Load Image File
   - 1.2 Color Conversion
   - 1.3 Image Flipping
   - 1.4 Blending
2. (10%) Adaptive Threshold (出題：Shaku)
   - 2.1 Global Threshold
   - 2.2 Local Threshold
3. (10%) Image Transformation (出題：YiYuan)
   - 3.1 Rotation, scaling, translation
   - 3.2 Perspective transform
4. (20%) Convolution (出題: Kris)
5. (50%) Training MNIST classifier using LeNet (出題：Charlie)

# 0. GUI

- Use MFC to create GUI like following figure.



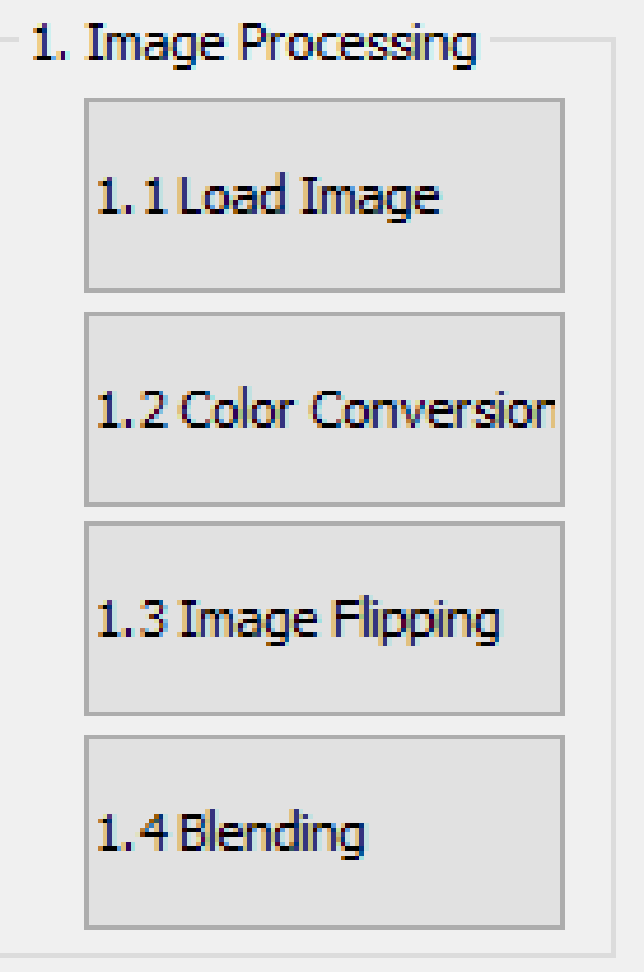
# 1. Image Processing (出題： Shaku)

1.1 Load Image File

1.2 Color Conversion

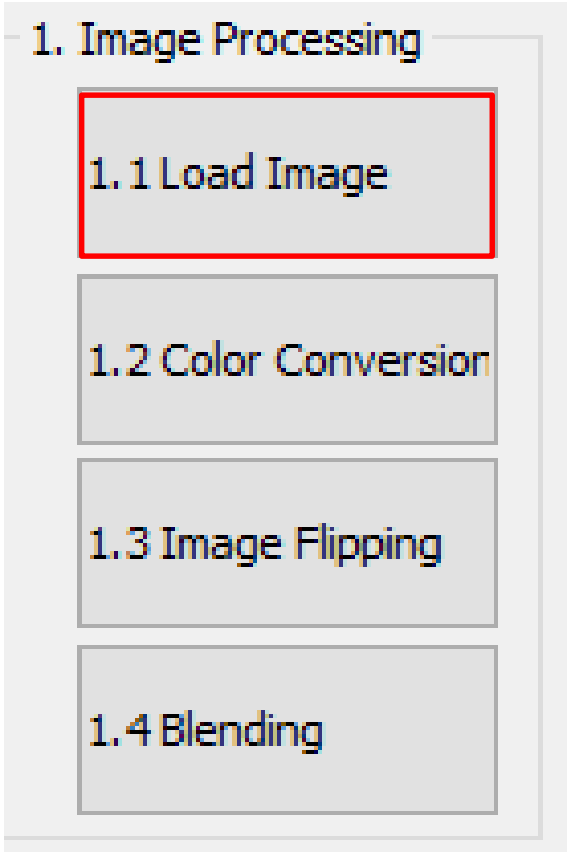
1.3 Image Flipping

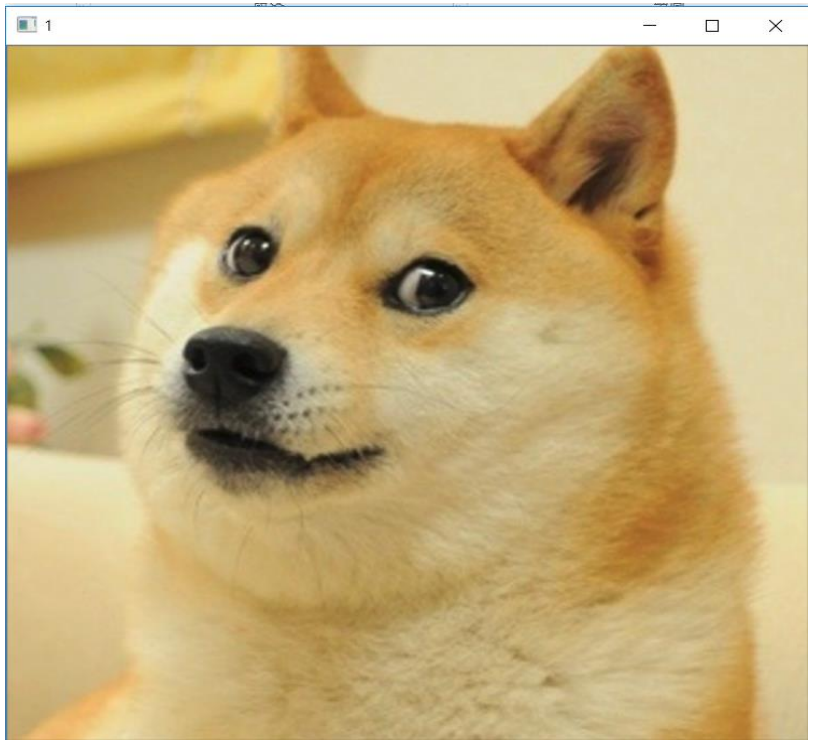
1.4 Blending



# 1.1 Load Image File

- Given: dog.bmp image
- Q: 1) Open a new window to show the image (dog.bmp)
  2) Show the height and width of the image in console mode
- Hint : Textbook Chapter 2, p. 22~23

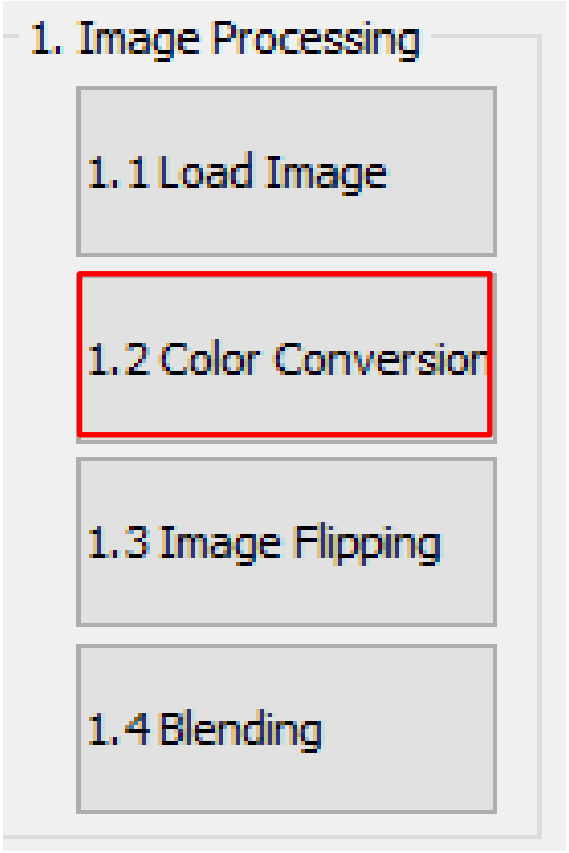


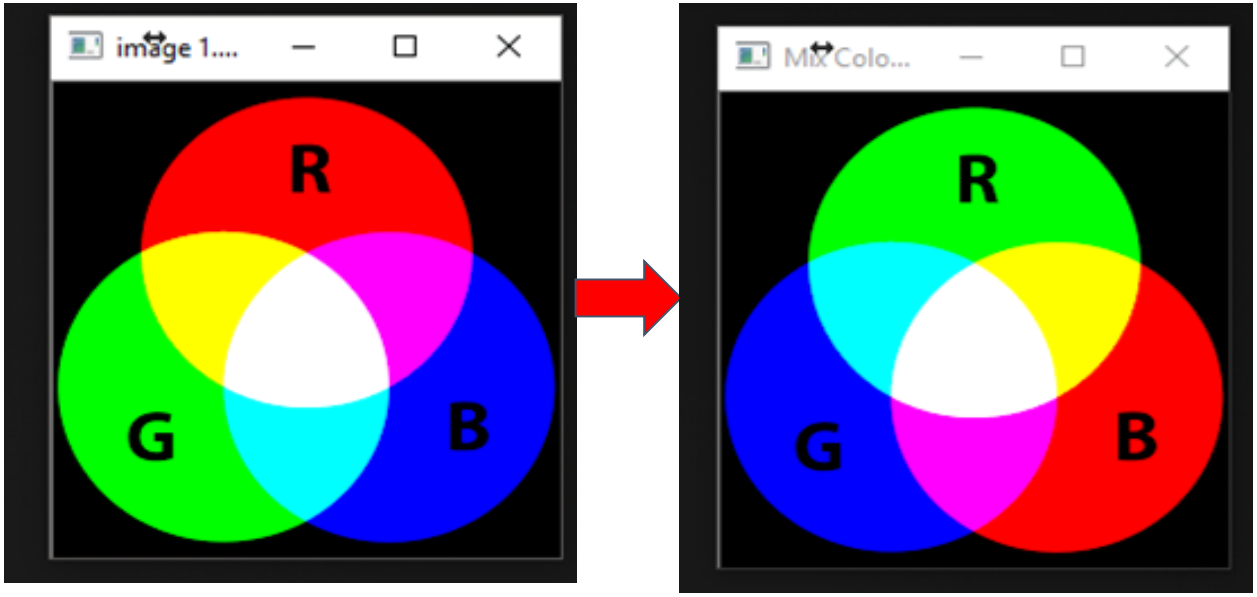




```
Height = 559
Width = 595
```

# 1.2 Color Conversion

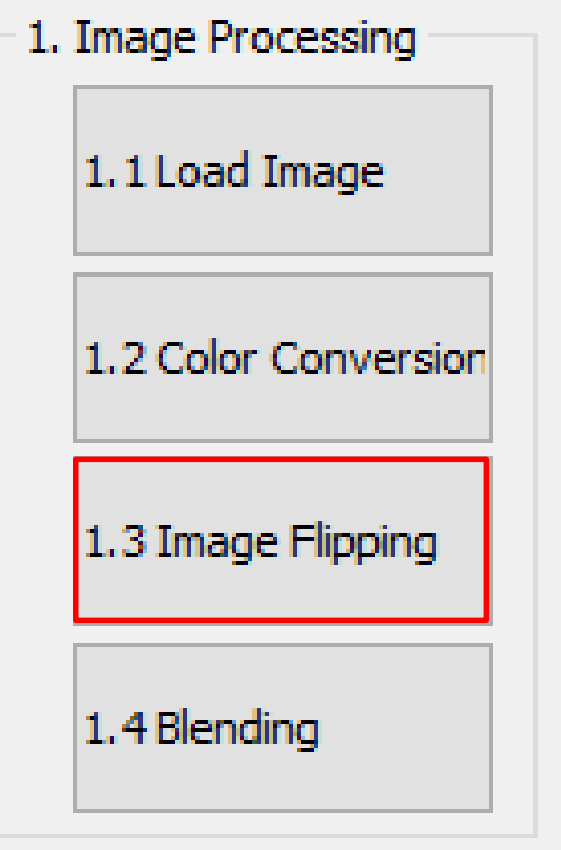
- Given: a color image, “color.png”
- Q: 1) Exchange 3 channels of the image BGR to RBG
  2) Open a new window to show the result.
- Example：Original BGR value of a pixel P is P(a,b,c), the value of the pixel P will change to P(b,c,a).
- Hint: Textbook Chapter 3, p.31 ~ p.44

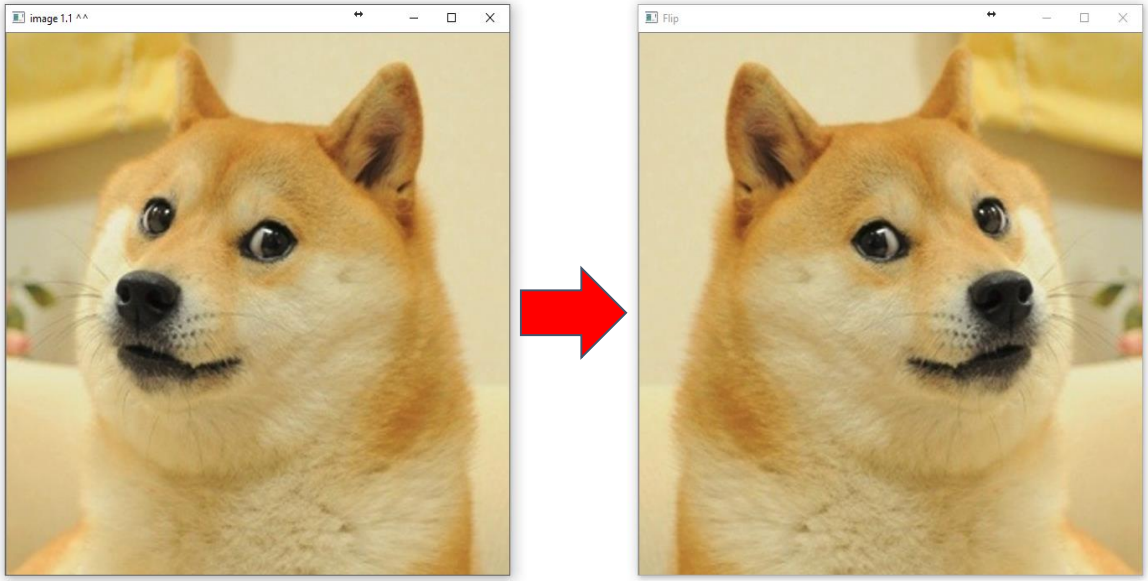


# 1.3 Image Flipping

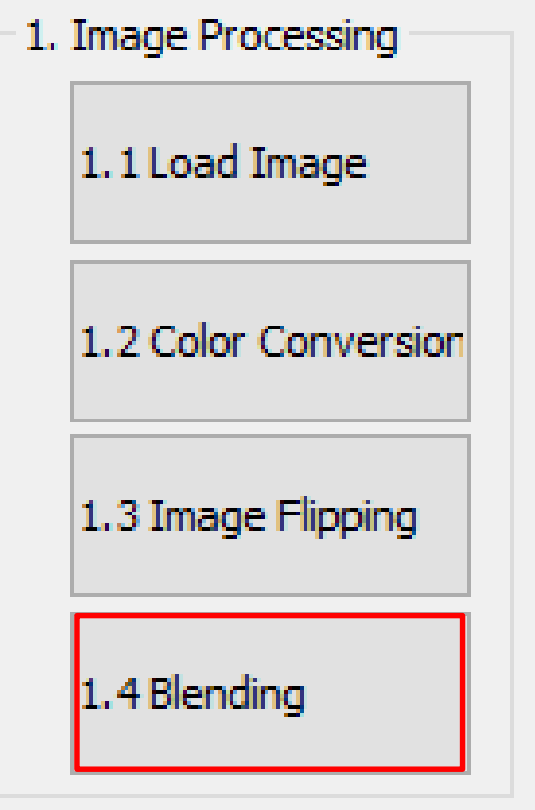
- Given: an image, dog.bmp
- Q: 1) Flip the image (dog.bmp) and open a new window to show the result.
- Hint: Textbook Chapter 3, p.31 ~ p.44

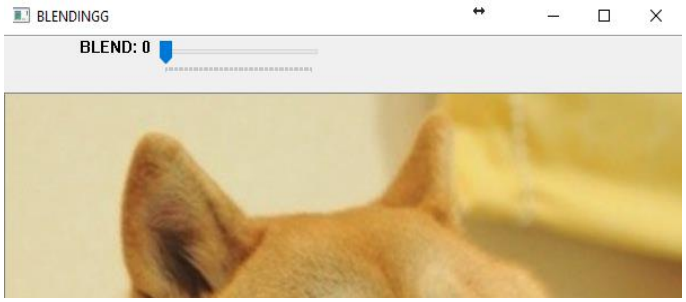


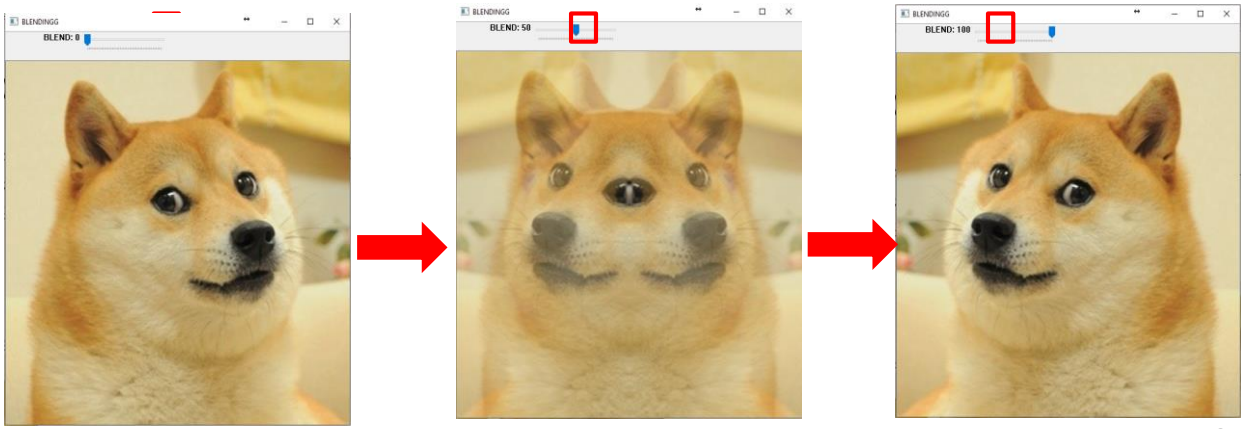



# 1.4 Blending

- Given: 2 images, dog.bmp and the result of 1.3
- Q: 1) Combine two images (dog.bmp and the result of 1.3).
  2) Use Trackbar to change the weights and show the result in the new window.
- Hint:
  - Textbook Chapter 3, p. 51 ~ 51
  - createTrackbar(...);



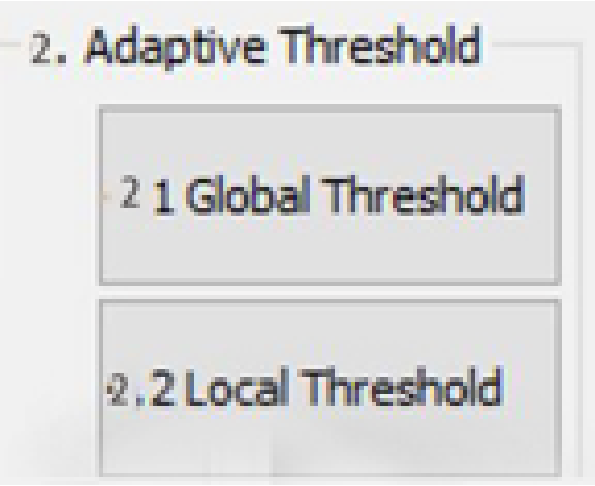



# 2. Adaptive Threshold

(出題: Shaku)

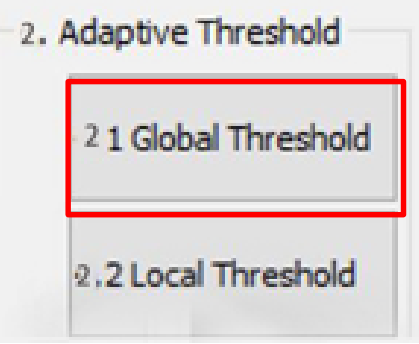
2.1 Global Threshold

2.2 Local Threshold



# 2.1 Global Threshold

- Given: a non-uniformly illuminated QR.png
- Q: 1) Show original image
  2) Show the result after applying global threshold
- Hint:
  - OpenCV Textbook Chapter 5 (p.136~138)
  - threshold(args) func with threshold value = 80



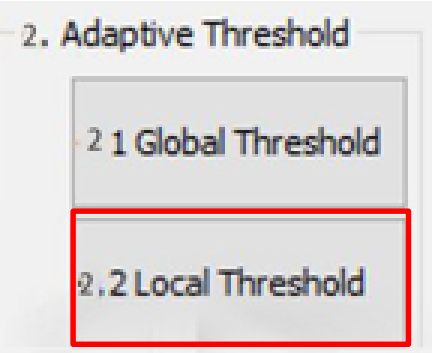


QR.png


global threshold

# 2.2 Local Threshold

- Given: a non-uniformly illuminated QR.png
- Q: 1) Show original image
  2) Show the result after applying local threshold
- Hint:
  - OpenCV Textbook Chapter 5 (p.136~138)
  - adaptiveThreshold(args) func with blockSize = 19, offset = -1





QR.png


local threshold

# 3.1 Transforms: Rotation, Scaling, Translation

(出題：YiYuan)

- Given: ***OriginalTransform.png*** *image*
- Q: 1) Click button "**3.1**", ***OriginalTransform.png*** should be showed.
  2) Please rotate, scale and translate the small squared image (as Figure 3.1) with following parameters (should be entered in the GUI):
  **(1) Angle = $45^0$** (*counter-clockwise*)
  **(2) Scale = 0.8,**
  **(3) Translation** with:
  - $x_{new} = x_{old}$ + 150 pixels = 130 + 150 = 280
  - $y_{new} = y_{old}$ + 50 pixels = 125 + 50 = 175

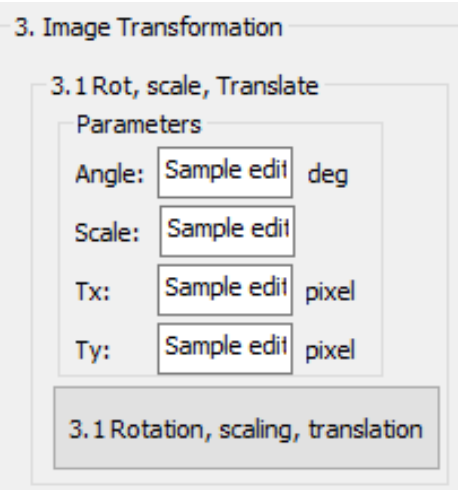
  Point C (130,125) is center of small square image
- Hint :

  OpenCV Textbook Chapter 12 (p. 407 ~ p. 412)

  warpAffine(…);

Small square image

- EX:



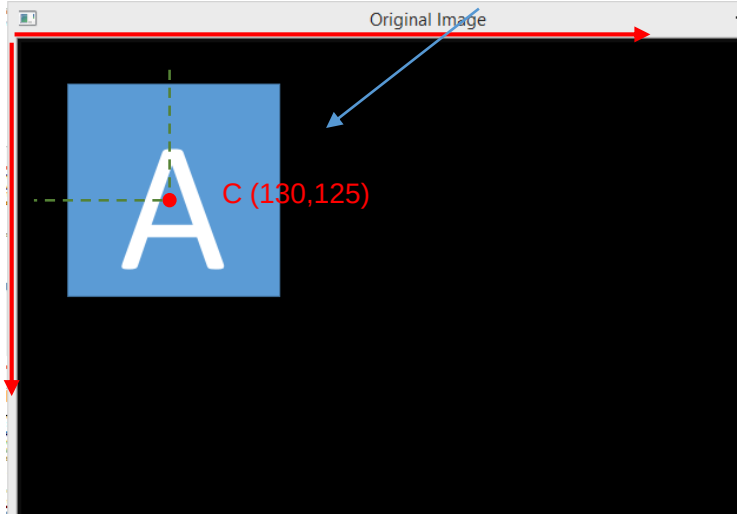




Figure 3.1 Original Image

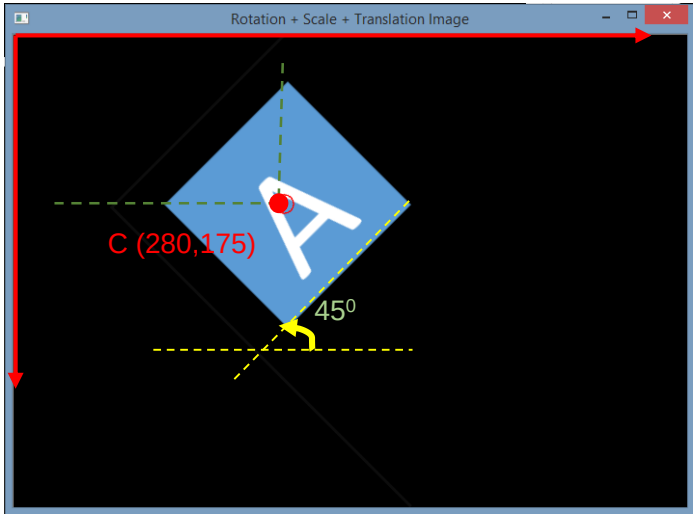
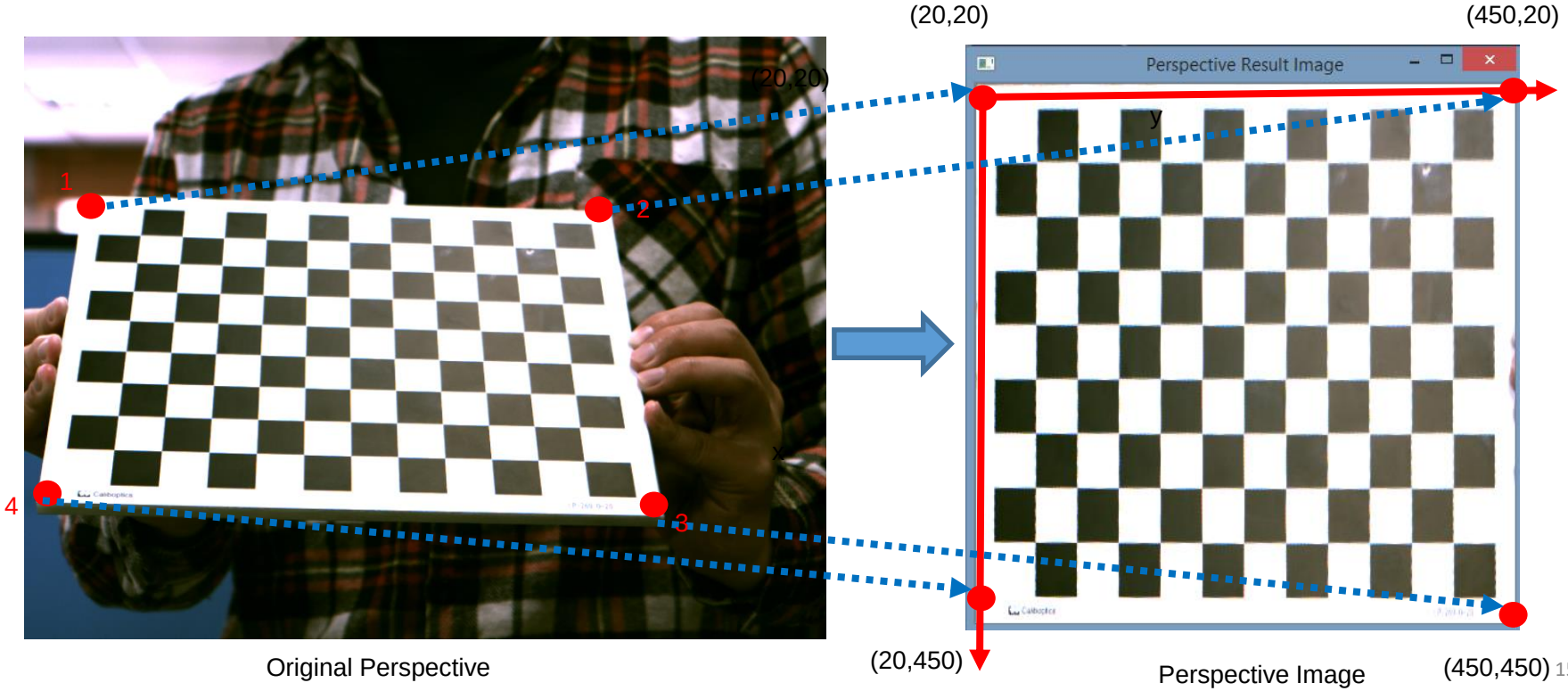



Figure 3.2 Rotation, Scale and Translation Image

# 3.2 Perspective Transformation (出題：YiYuan)

- Given: ***OriginalPerspective.png*** *image*
- Q: Use OpenCV functions to straighten the image
  1) Click button “3.2” to show image in the new window. Then do:
     a) Click 4 points showed in console window. (start from top-left corner of the original image, and then click clock-wise)
     b) Warp the original image to the location (20,20), (20,450), (450,450), (450,20). Open second window to show the result.
- Hint:
  - Textbook Chapter 6, p. 170~171
  - mouse callback function(p.96)



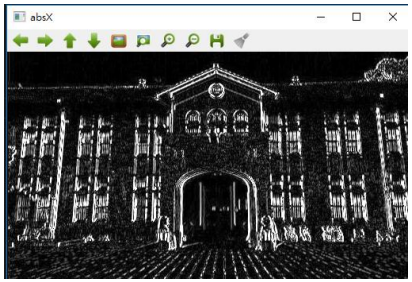
# 4. Edge Detection

**(出題：Kris)**

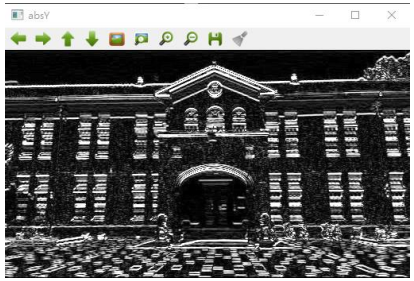
- Given: an RGB image, School.jpg
- Q: 1) Convert the RGB image to grayscale image and then smooth the grayscale image by using your own 3x3 Gaussian smoothing filter and show the result. (5%) (**can not** use OpenCV function)
  2) Program and show the result using your own code - **Sobel edge detection** to detect horizontal edge (5%) and vertical (5%) edge. (**can not** use OpenCV function)
  2.1) Normalize 2) result to 0~255 and show.
  2.2) Use 2.1) result to calculate the magnitude and show.(5%)
- Hint: Textbook Chapter 6, p.148 ~ 149
- Hint: magnitude = $|| Sobel\_x^{\wedge 2}+Sobel\_y^{\wedge 2} ||^{1/2}$


School.jpg

3x3 Gaussian smooth filter

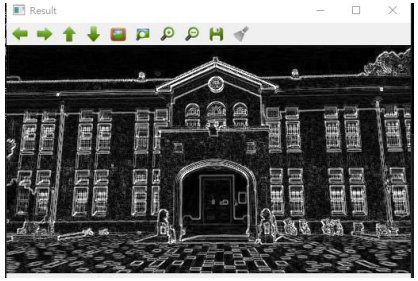


vertical edges

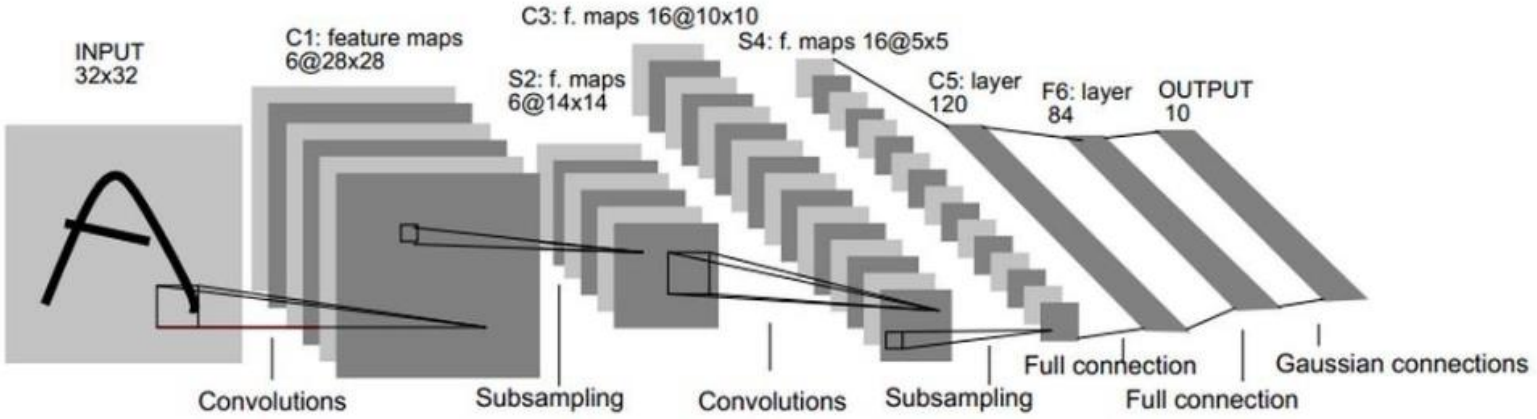


Horizontal edges

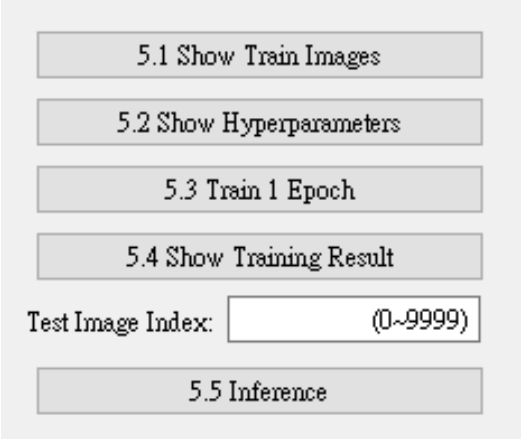
magnitude



# 5.0 Training MNIST Classifier Using LeNet5（出題：Charlie)




1. Learning to construct LeNet and training it on MNIST.
2. Environment Requirement
   1) Python 3.7
   2) Tensorflow 2.0 / PyTorch 1.3.0
   3) opencv-contrib-python 3.4.2.17
   4) Matplotlib 3.1.1
3. Reference
   1) Gradient-Based Learning Applied to Document Recognition (http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf) (LeNet)
   2) MNIST (http://yann.lecun.com/exdb/mnist/)



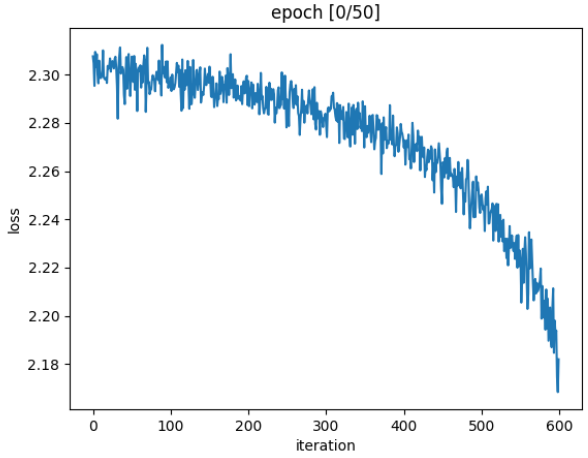
5.1 Load MNIST training dataset and randomly show 10 images and labels respectively. (10%)

Label: 7 2 1 0 …

5.2 Print out training hyperparameters (batch size, learning rate, optimizer). (10%)

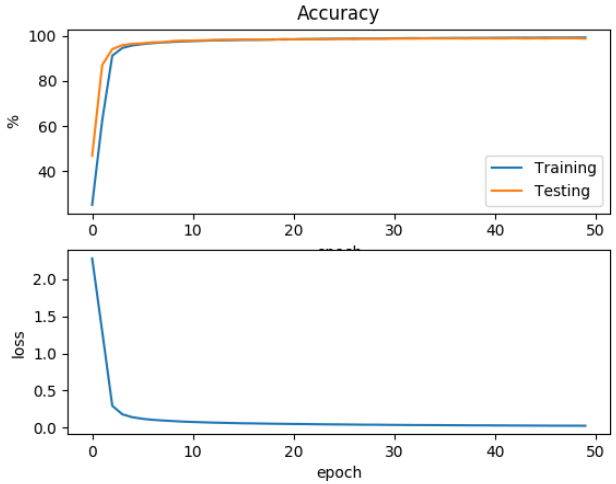
```
hyperparameters:
batch size: 32
learning rate: 0.001
optimizer: SGD
```

5.3 Train 1 epoch from initial status and show training loss at the end of the epoch. (10%)




(record loss per iteration)

5.4 Training your model at least 50 epochs by your own computer, save your model and take a screenshot of your training loss and accuracy. (10%)




(record accuracy/loss per epoch)

5.5 Load your model trained at 5.4, let us choose one image from MNIST test images, inference the image, show image and estimate the image as following. (10%)