# Notice (2/2)

- Python
  - 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)
  - Pytorch
  - Tensorflow

# 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%) Edge Detection (出題：Chong)
   - 3.1 (5%) Gaussian Blur
   - 3.2 (5%) Sobel X
   - 3.3 (5%) Sobel Y
   - 3.4 (5%) Magnitude
4. (20%) Transforms (出題：Jeffin)
   - 4.1 (5%) Resize
   - 4.2 (5%) Translation
   - 4.3 (5%) Rotation, Scaling
   - 4.4 (5%) Shearing
5. (20%) Training Cifar10 Classifier Using VGG19 (出題：Wen)
   - 5.1 (4%) Load Cifar10 and Random Show 9 Images with Label
   - 5.2 (4%) Load Model and Show Model Structure
   - 5.3 (4%) Show Data Augmentation Result
   - 5.4 (4%) Show Accuracy and Loss
   - 5.5 (4%) Inference

# 3. Edge Detection (20%)

(出題: Chong)

3.1 Gaussian Blur (5%)

3.2 Sobel X (5%)

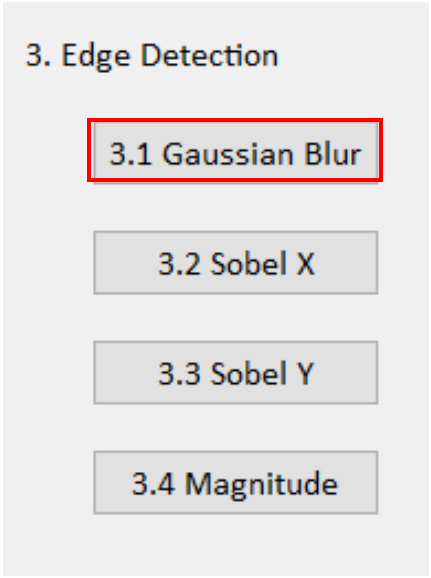
3.3 Sobel Y (5%)

3.4 Magnitude (5%)

# 3.1 Gaussian Blur (5%)

(出題：Chong)

- Given: a RGB image, “building.jpg”
- Q: 1) **Gaussian Blur**: Convert the RGB image into a grayscale image, then smooth it by your own 3x3 Gaussian smoothing filter (Can not use OpenCV Function, Sobel, GaussianBlur and conv2d). Please show the result.




- Hint: Textbook Chapter 5, p.109 ~ p.114

  How to generate Gaussian Filter:

  ① Let $G_{init}(x,y) = \begin{bmatrix} (-1,-1) & (0,-1) & (1,-1) \\ (-1,0) & (0,0) & (1,0) \\ (-1,1) & (0,1) & (1,1) \end{bmatrix}$

  ② Calculate $G(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$, $\sigma = \sqrt{0.5}$

  ③ Normalize $G(x,y), G_{norm}(x,y) = \begin{bmatrix} 0.045 & 0.122 & 0.045 \\ 0.122 & 0.332 & 0.122 \\ 0.045 & 0.122 & 0.045 \end{bmatrix}$

building.jpg

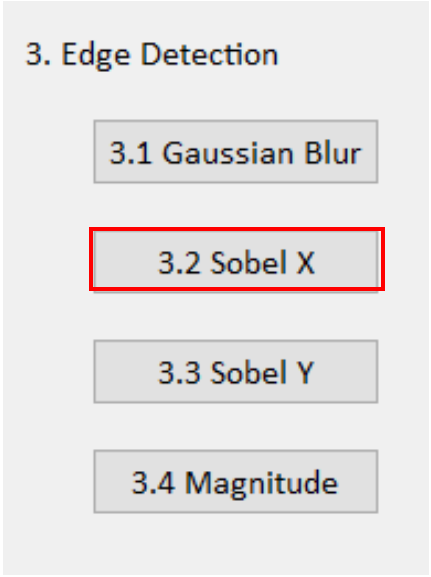
Grayscale

Gaussian Blur

# 3.2 Sobel X (5%)

(出題: Chong)

- Given: the result of 3.1) Gaussian Blur
- Q: 2) **Sobel X**: Use Sobel edge detection to detect vertical edge by your own 3x3 Sobel X operator (Can not use OpenCV Function, Sobel, GaussianBlur and conv2d). Please show the result.
- Hint: Textbook Chapter 6, p.148 ~ 149




Gaussian Blur

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

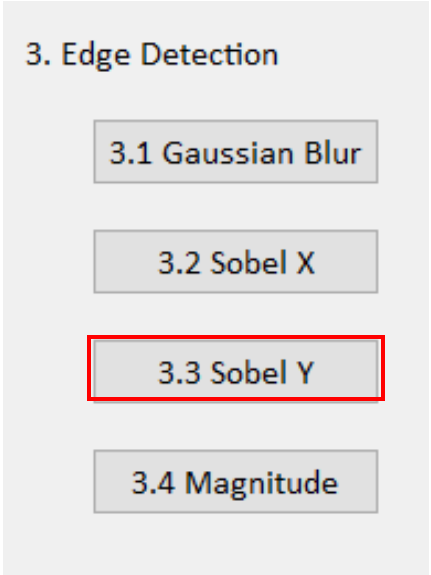
Sobel X Filter

Sobel X

# 3.3 Sobel Y (5%)

(出題: Chong)

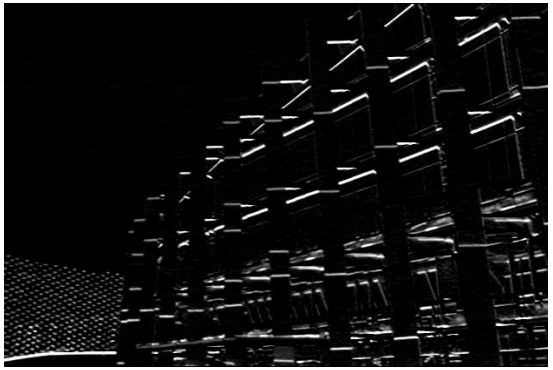
- Given: the result of 3.1) Gaussian Blur
- Q: 3) **Sobel Y**: Use Sobel edge detection to detect horizontal edge by your own 3x3 Sobel Y operator (Can not use OpenCV Function, Sobel, GaussianBlur and conv2d). Please show the result.
- Hint: Textbook Chapter 6, p.148 ~ 149




Gaussian Blur

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

Sobel Y Filter



Sobel Y

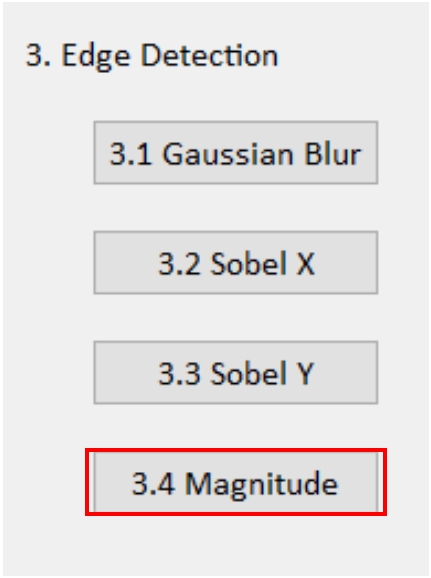
# 3.4 Magnitude (5%)

(出題：Chong)

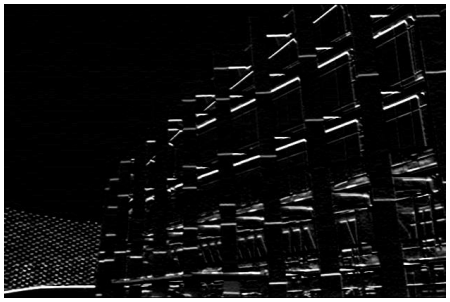
- Given: the result of 3.2) Sobel X and 3.3) Sobel Y
- Q: 4) **Magnitude**: Use the results of 3.2) Sobel X and 3.3) Sobel Y to calculate the magnitude. Please show the result.
- Hint: Textbook Chapter 6, p.148 ~ 149

$$\text{Magnitude} = \sqrt{\left\| Sobel_X{}^2 + Sobel_Y{}^2 \right\|}$$

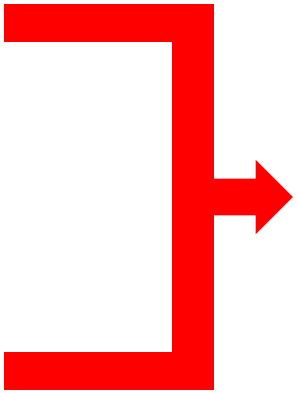
Normalize the result to 0~255.




Sobel X



Sobel Y

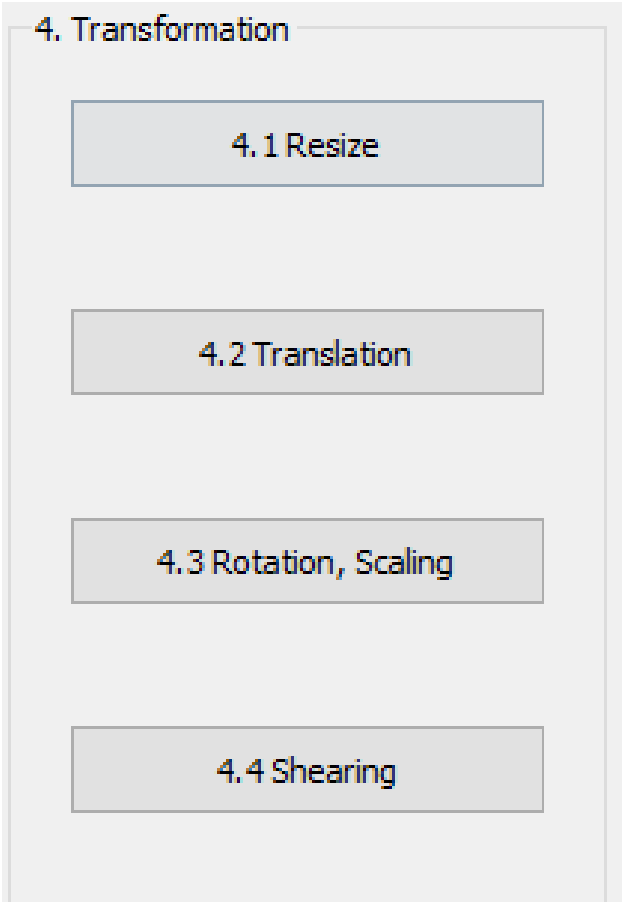


Magnitude

# 4. Transforms (20%)

(出題: Jeffin)

- 4.1 Resize (5%)
- 4.2 Translation(5%)
- 4.3 Rotation, Scaling (5%)
- 4.4 Shearing (5%)



# 4.1 Transforms: Resize, Translation, Rotation, Scaling, Shearing(20%)

- Given: "*Microsoft.png*", Image Size (430, 430)

(出題: Jeffin)

- Q:Please resize, translate , rotate, scale and shearing the picture (*Microsoft.png*)

  4.1) Resize:
  From (430,430) to (215,215)
  and cv2.imshow with (430, 430) window (image center: (108, 108)
  top left of window)

  4.2) Image Translation:
  Xnew = Xold + 215 pixels = 108 + 215 = 323
  Ynew = Yold + 215 pixels = 108 + 215 = 323
  Point C (108, 108) is center of resized image
  Point C'(323, 323) is new center of image (bottom right of window)
  (Then overlay with *result image of 4.1)*

  4.3) Rotation, Scaling:
  Center: Center of Image
  Angle = 45$^{o}$ (counter-clockwise)
  Scale = 0.5 , window size (430,430)

  4.4) Shearing:
  Old location: ([[50,50],[200,50],[50,200]])
  New location: ([[10,100],[100,50],[100,250]])
  (Note: Please save your image after each section)

W:430

H:430


*Microsoft.png*

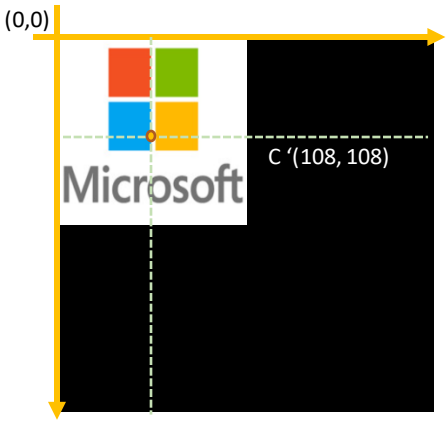
- Hint: Textbook Chapter 12, (p.407 ~ 412)
  python: cv2.warpAffine(),
  Textbook Chapter 3, (p.50 ~ 52)
  cv2.addWeighted()

# 4.2 Transforms: Resize, Translation, Rotation, Scaling, Shearing(20%)

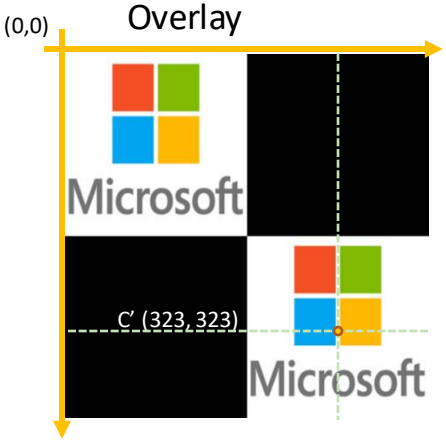
- EX: Given: “Microsoft.*png*”, image size (430, 430) (出題: Jeffin)

*Microsoft.png*

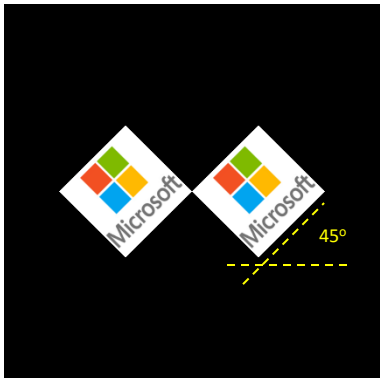

4.1) Resized Image



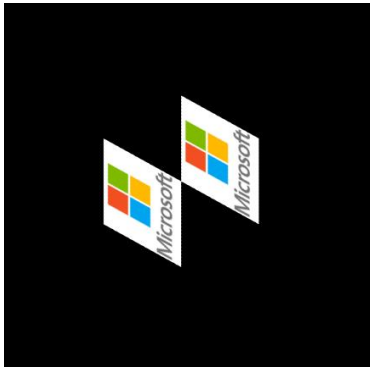

4.2) Translate + Overlay




4.3) Rotate and Scale




4.4) Shearing




- Hint: Textbook Chapter 12, (p.407 ~ 412)
  python: cv2.warpAffine(),
  Textbook Chapter 3, (p.50 ~ 52)
  cv2.addWeighted()