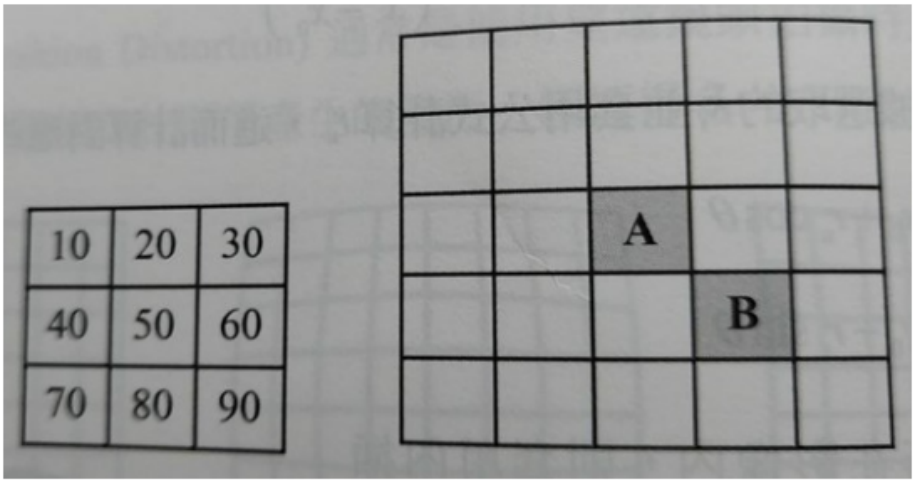
實戰 (1)

- 題目：將 3x3 的灰階影像，放大為 5x5 的灰階影像要求：分別採用 (a) 最近鄰內插法 (b) 雙線性內插法算出 A 及 B
- 方法：可以寫程式或手算




```python
from numpy.matrixlib.defmatrix import matrix
import numpy as np
import cv2
import math
from matplotlib import pyplot as plt
matrix=np.array ([[10,20,30],[40,50,60],[70,80,90]])
print (matrix.shape)
def NN_interpolation(img):
    srcH, srcW = img.shape
    dstH, srcW = 5,5
    retIMG = np.zeros([5,5],dtype = 'uint8')
    for i in range (5) :
        for j in range (5) :
            srcX = int (round (i) * (3/5))
            srcY = int (round (j) * (3/5))
            if srcX >= srcW:
                srcX = srcW -1
            if srcY >= srcH:
                srcY = srcH -1
            retIMG[i,j] = img[srcX,srcY]
    return retIMG
newIMG = NN_interpolation(matrix)
print ( newIMG.shape)
print ( newIMG )
```

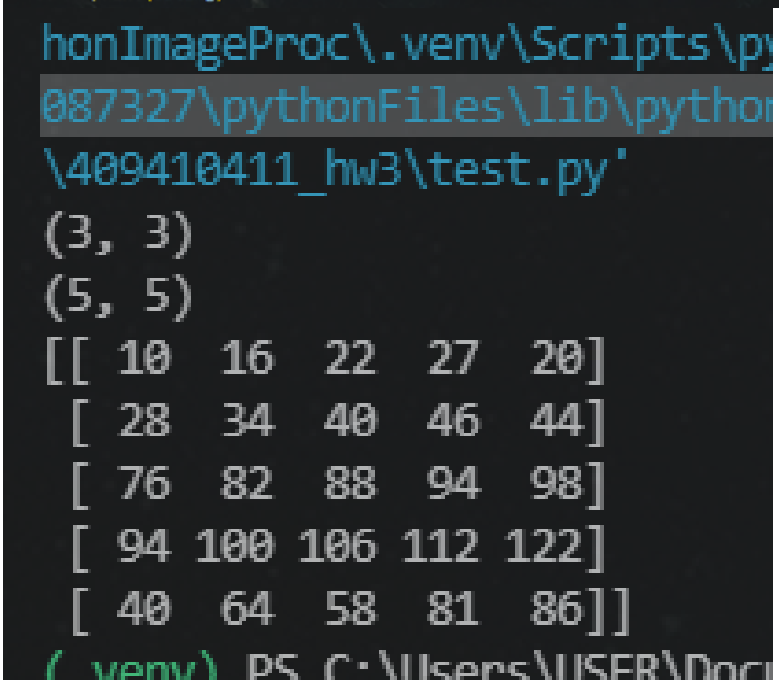
```
(3, 3)
(5, 5)
[[10 10 20 20 30]
 [10 10 20 20 30]
 [40 40 50 50 60]
 [40 40 50 50 60]
 [70 70 80 80 90]]
```

(a)

A = 50

B = 50


pythonImageProc > 409410411_hw3 > test.py > ...
C:\Users\USER\Documents\pythonImageProc • Contains emphasized items
import cv2
import math
matrix=np.array([[10,20,30],[40,50,60],[70,80,90]])
print(matrix.shape)
def double_linear(input_signal,zoom_multiples):
input_row,input_col = input_signal.shape
output_row = 5
output_col = 5
output_signal = np.zeros((output_row,output_col))
for i in range(output_row):
for j in range(output_col):
temp_x = i / output_row * input_row
temp_y = j / output_col * input_col
x1 = int(temp_x)
y1 = int(temp_y)
x2 = x1;y2 = y1 + 1
x3 = x1 + 1;y3 = y1
x4 = x1 + 1;y4 = y1 + 1
t = temp_x;u = temp_y - y1
if x4 >= input_row:
x4 = input_row -1
x2 = x4
x1 = x4 -1
x3 = x4 -1
if y4 >= input_col:
y4 = input_col -1
y3 = y4
y1 = y4 -1
y2 = y4 -1
output_signal[i,j]=(1-t)*(1-u)*input_signal[x1,y1] + (1-
t)*u*input_signal[x2,y2] + t*(1-u)*input_signal[x3,y3]+t*u*input_signal[x4,y4]
return output_signal
dstImg = double_linear(matrix,(5/3)).astype(np.uint8) "astype": Unknown word.
print(dstImg.shape)
print(dstImg)


honImageProc\.venv\Scripts\py
087327\pythonFiles\lib\python
\409410411_hw3\test.py'
(3, 3)
(5, 5)
[[10 16 22 27 20]
[28 34 40 46 44]
[76 82 88 94 98]
[94 100 106 112 122]
[40 64 58 81 86]]
(.venv) PS C:\Users\USER\Doc

(b)

A = 88

B = 112

實戰 (2)

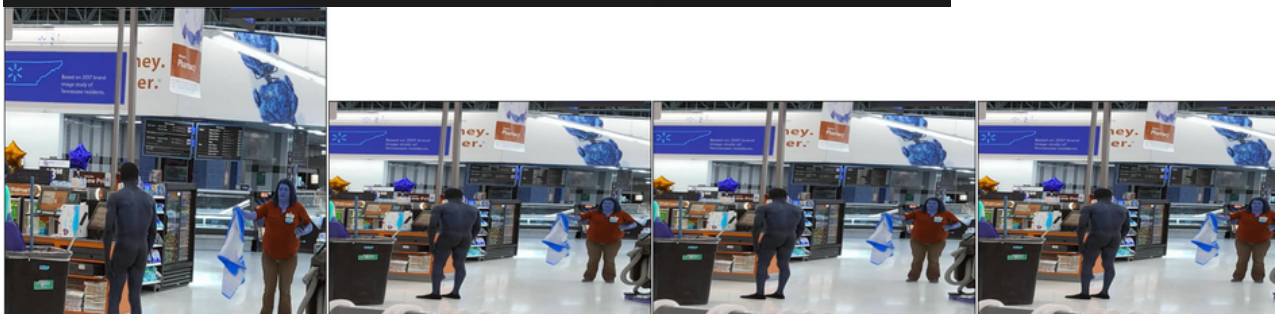
- 題目：設計 Python 程式，使用放大函式，輸出放大影像的結果
- 要求：分別使用最近鄰內插法、雙線性內插法、雙立方內插法，並針對結果做一些評論。

```
import numpy as np
import cv2
import math
from matplotlib import pyplot as plt    "pyplot": Unknown word.

image = cv2.imread(r'C:\Users\USER\Documents\pythonImageProc\409410411_hw3\blck.jpg')

scale = eval(input('Please enter scale:'))
nr,nc=image.shape[:2]
nr2 = int(nr*scale)
nc2 = int(nc*scale)
newImg = cv2.resize(image,(nr2,nc2),interpolation=cv2.INTER_NEAREST)
newImg_2 = cv2.resize(image,(nr2,nc2),interpolation=cv2.INTER_LINEAR)
newImg_3 = cv2.resize(image,(nr2,nc2),interpolation=cv2.INTER_CUBIC)

images = [image,newImg,newImg_2,newImg_3]
plt.figure(figsize=(100,100))
for i in range(4):
    plt.subplot(1,4,i+1),plt.imshow(images[i],'gray')    "imshow": Unknown word.
    plt.xticks([]),plt.yticks([])    "xticks": Unknown word.
plt.tight_layout()
plt.show()
```



to be honest i cant tell the difference at all, other than the insane bluing

簡答題

1.試定義空間轉換?空間轉換的方法有哪兩種

2.試定義幾何轉換?幾何轉換可以分成哪幾種?

3. 試舉出常用的數位影像的內插法有哪些?

1. expanding/compressing area by duplicating/cutting and taking away space
(x',y')=T{(x,y)}

正向映射

反向映射

2. changing the relative points of an image through vectors, this does not alter color corrections

仿射轉換 - 縮放、旋轉、平移、翻轉、偏移

透視轉換

3. 最鄰近內插法

雙線性內插法

雙立方內插法