NTNU 影像處理 HW6

廖家緯

2020.4.21

- **Outline**:

  Image Enlargement

  1. Choose a grayscale image I

  2. Zero interleave $I'(i,j) = \begin{cases} I(\frac{i+1}{2}, \frac{j+1}{2}) & \text{if } i,j : \text{odd} \\ 0 & \text{otherwise} \end{cases}$

  3. Fill values by convolving I' with

     (1) NN interpolation: $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

     (2) Bilinear interpolation: $\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

  4. Output enlarged images

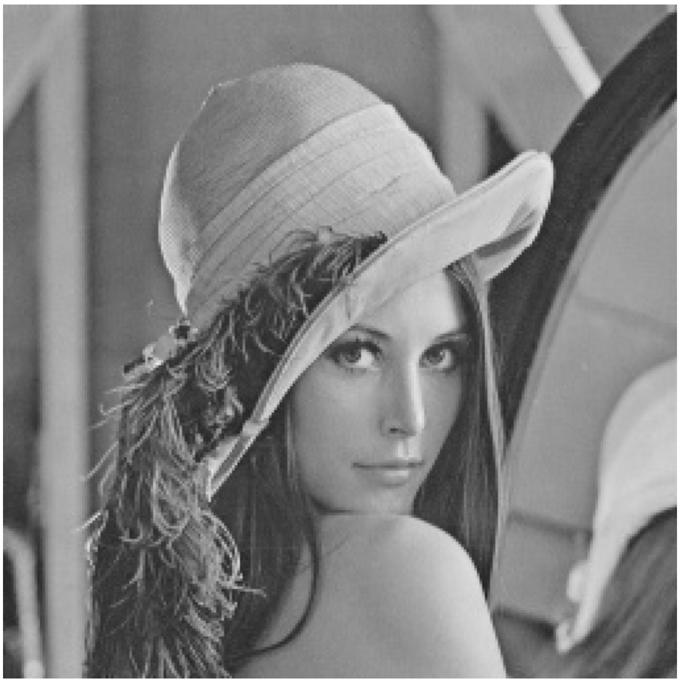
- **Code(Python):**

```python
# coding: utf-8
import numpy as np
import cv2

# Convolution
def conv(Matrix, Filter):
    Matrix = np.pad(Matrix,((1,1),(1,1)),'constant',constant_values = (0,0))
    n, m = np.shape(Matrix)
    Matrix2 = np.zeros((n, m)).astype('double')
    for i in range(1, n-1):
        for j in range(1 ,m-1):
            Matrix2[i,j] = (Matrix[i-1:i+2, j-1:j+2]*Filter).sum()

    Matrix2 = Matrix2.astype('uint8')
    return Matrix2[1:n-1, 1:m-1]

# Step 1: Choose a grayscale image I
I = cv2.imread('Gray.jpg', cv2.IMREAD_GRAYSCALE)
n, m = np.shape(I)

# Step 2: Zero interleave
I1 = np.zeros((n*2,m*2), np.double)

for i in range(n*2):
    for j in range(m*2):
        if i%2==0 and j%2==0:
            I1[i, j] = I[i//2, j//2]

# Step 3: Fill values by convolving I1 with
# (i) NN interpolation
NN = np.array([[1, 1, 0], [1, 1, 0], [0, 0, 0]])

# (ii) Bilinear interpolation
Bilinear = (1/4)*np.array([[1, 2, 1], [2, 4, 2], [1, 2, 1]])

img_NN = conv(I1, NN)
img_Bilinear = conv(I1, Bilinear)

cv2.imwrite('img_NN.jpg', img_NN)
cv2.imwrite('img_Bilinear.jpg', img_Bilinear)
```

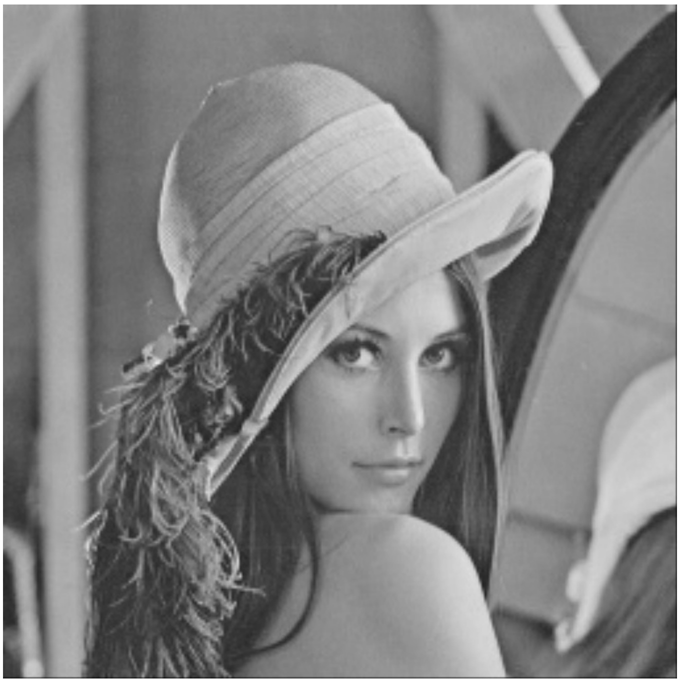
- **Input image:**

Original

- **Result:**



NN



Bilinear

- **Experience:**

  在影像處理中，通常從1開始編號，而我使用的Python程式語言是從0開始編號，因此我的code和老師的步驟有點不同。另外，為了方便對每個pixel做convolution，我在函式中加入padding (補0)，做完後再刪去，使得邊界不會產生條紋。