Digital Image Processing

B05902118 資工三 陳盈如

1. First, I write my code in Python. I use ***cv2.imread( )*** to read my image, DIP.png, use ***cv2.resize( )*** to implement two interpolation methods and use ***cv2.imwrite( )*** to store the results in the same file. You can use command "python DIP.py" to run the code. I've already attached two pictures in the file, named "bilinear" and "bicubic".

```
bilinear = cv2.resize(原始檔大小, 縮放後大小, interpolation =
cv2.INTER_LINEAR)
bicubic = cv2.resize(原始檔大小, 縮放後大小, interpolation =
cv2.INTER_CUBIC)
```

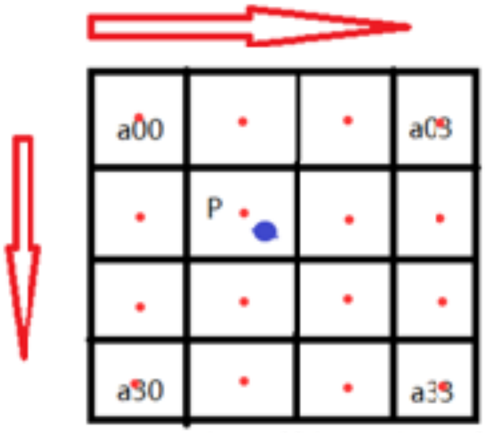
2. I also use ***cv2.imread( ), cv2.resize( )*** and ***cv2.imwrite( )***, only changing the scaling factors, namely the second variable in ***cv2.resize( )***, to original size multiplying 0.2, 3.0, and 10.0. Use command "python DIP.py" to run the code. Likewise, I've already attached six pictures in the file, named "bilinear 0.2", "bilinear 3.0", "bilinear 10.0", "bicubic 0.2", "bicubic 3.0", and "bicubic 10.0" respectively.
3. It's difficult for us to identify the differences between the two pictures with the scaling factor of 0.2. Then, when I change the scaling factor to 3.0, we can find that the profile of the bicubic interpolation is clearer than the bilinear interpolation. Besides, the effect of the contrast of colors is also a little stronger in the bicubic interpolation. And then, we change the scaling factor to 10.0. Likewise, the outline and the effect of the contrast of colors are better in the bicubic interpolation. Therefore, the bigger we amplify the pictures, the more obviously they are different.

4. Assume the original picture A is m*n, the final picture B is M*N, and K = M/m = N/n. Every pixel of A is known, and B is unknown.

   If we want to get the value of the pixel (X, Y) on picture B, we need to find the corresponding spot (x, y) on picture A and use the nearest sixteen pixels around (x, y) as parameters to calculate the pixel (X, Y).

   A(x, y) = A(X*(m/M), Y*(n/N)) = A(X/K, Y/K)

   P is the position of A(X/K, Y/K), and it will have a decimal. We assume P(x+u, y+v), x, y are integers, and u, v are decimal parts.



   We use the distance between a_{ij} and P(x+u, y+v) to get the weight by S(x). For example, distance between a_{00} and P(x+u, y+v) is (1+u, 1+v), so i_0 = S(1+u) and j_0 = S(1+v). Then, the contribution of a_{00} on B(X, Y) is (a_{00}'s pixel value)*(i_0)*(j_0).

   S(x) is based on the function on WIKI-Bicubic interpolation:

$$S(x) = \begin{cases} 1-(a+3)x^2+(a+2)|x|^3, & 0 \le |x| \le 1 \\ -4a+8a|x|-5ax^2+a|x|^3, & 1 < |x| \le 2 \end{cases}$$

   , where a is usually set to -0.5 or -0.75.

   The final function is as follows.

$$\mathrm{B(X,Y)} = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} \times W(i) \times W(j)$$

   The computational complexity of bicubic interpolation and bilinear interpolation are both O(N^2), but bicubic interpolation takes longer time because it takes more time to compute the gradients at each re-sampled pixel location. The processing time increases with the increase in the number of pixels considered.