

Introduction to Image Processing HW1

Name: 王昱力 Student id: 110550090

March 20 2024

Method

這次的作業使用了 cv2 的讀寫和 numpy 來處理圖片

Rotate 30 degree:

(1) Nearest Neighbor Interpolation

- 首先我先計算出圖片的中心點，center_x 和 center_y，之後將每個 pixel 扣去中心點座標移回座標原點 (0,0)，apply 旋轉矩陣 ($\theta = 30^\circ$):

$$\text{Rotate} = \begin{bmatrix} \cos(30^\circ) & -\sin(30^\circ) \\ \sin(30^\circ) & \cos(30^\circ) \end{bmatrix}$$

讓順時鐘旋轉 30° 後的圖的 pixel 做逆時鐘旋轉回到還沒旋轉前原始圖片的位置，之後再把每個 pixel 加回中心點座標平移回去原本位置，並取上課講授過的最接近 pixel 位置的點當作旋轉後圖片的值

(2) Bilinear Interpolation

- 也是先取和上述先移回原點，旋轉，再移回去的方式，但取 pixel 的方式則是取最接近的 4 個 pixel，依照上課講授的方式先對 x 做線性插值，得到兩個值，再對其 y 值做線性插值，並得到結果。

(3) Bicubic Interpolation

- 一樣先移回原點，旋轉，再移回去，透過最接近的 16 個 pixel，依照 spec 上的方式求出多條 3 次方程式，再透過 3 次函式插值出目標的值。具體方式為先對 x 軸做線性插值，可以得到 4 條 3 次函數，並得到 4 個相對應的值，再對這 4 個值求出一條 3 次函數，最後在對 y 軸的方向做插值。因為在插值的過程可能出現 overflow 的問題，所以在最後使用 clip 的 function 將值限制在 (0,255)

放大 2 倍:

(1) Nearest Neighbor Interpolation

- 對於放大 2 倍的圖，一樣使用 backward 的方式，將新的 pixel 除上放大倍率，對應到原本的圖，找到最接近的點就是它的 pixel 值

(2) Bilinear Interpolation

- 不同 nearest neighbor 找到最接近的值，依照課堂講授的方式，找出最接近的 4 個點，並對 x 做線性插值，得到的兩個值，再對其 y 值做線性插值，得到結果。

(3) Bicubic Interpolation

- 一樣，除上放大倍率，找到最接近的 16 的 pixel，採取和 Rotate 30 degree 方式相同做法，求出目標值，因此不多加贅述

Result



Figure 1: nearest-neighbor_rotate

Figure 2: bilinear_rotate



Figure 3: bicubic_rotate



Figure 4: nearest-neighbor_scale2



Figure 5: bilinear_scale2



Figure 6: bicubic_scale2

Feedback

這次的作業讓我更加了解各個插值法的原理，對影像處理有更多的認識，另外也了解了 rotate 背後使用的原理。