HW1 Report

110550108 施柏江

1. Method

1.1 Rotation (Nearest neighbor interpolation)

在跑遍旋轉後圖片的每個像素時，根據計算出的新坐標去找到原始圖片中對應的像素值，將原始圖片中最接近新坐標的像素值複製到旋轉後圖片中。

1.2 Rotation (Bilinear interpolation)

首先在水平方向上使用兩個最近的像素進行線性插值，然後在垂直方向上使用這兩個線性插值的結果進行插值，最後將得到的插值填入旋轉後圖片的對應位置中。

1.3 Rotation (Bicubic interpolation)

在每個（x, y）坐標處，首先得到在原始圖片中最接近（x, y）的十六個像素的坐標，然後根據這十六個像素的值，分別在水平和垂直方向上進行三次插值。最後將得到的插值結果填入旋轉後圖片的對應位置中。為了避免出現超出 0 到 255 的像素值範圍，我使用了 numpy.clip() 將像素值限制在此範圍內。

1.4 Magnification (Nearest neighbor interpolation)

採用了四捨五入的方式來將目標像素位置（i, j）映射到原始圖片的位置（x, y），並取最近的整數坐標。這樣做是為了在放大時獲得最近鄰插值的效果。

1.5 Magnification (Bilinear interpolation)

計算（x, y）附近的四個像素的值（x1, y1）、(x1, y2)、(x2, y1)、(x2, y2)，並根據雙線性插值的公式計算了（x, y）的插值結果。最後將得到的插值結果填入放大後圖片的對應位置（i, j）中。

1.6 Magnification (Bicubic interpolation)

先取得（x, y）附近的16個像素的值，計算水平方向和垂直方向上的雙三次插值，最後將得到的插值結果填入放大後圖片的對應位置（i, j）中。為了避免出現超出 0 到 255 的像素值範圍，我使用了 numpy.clip() 將像素值限制在此範圍內。

2. Result

2.1 Nearest Neighbor

2.2 Bilinear

2.3 Bicubic

2.4 Nearest Neighbor

2.5 Bilinear

2.6 Bicubic

3. Feedback

這次的作業讓我學會了如何使用 OpenCV 和 NumPy 來操作圖片，不僅熟悉了圖片的基本操作，還深入了解了如何進行圖片的放大和旋轉，以及如何利用不同的插值方法來實現這些操作。