**影像處理  
學號 : 41147047S  
姓名 : 黃國展  
系級 : 資工 115**

1. **實作功能說明**

本次實驗使用 Python 搭配 Gradio 實作影像旋轉功能，並比較不同插值法的影響。實作內容如下：

- 建立一張黑底白色正方形影像

- 實作影像旋轉功能，支援以下插值方式：

- Nearest Neighbor Interpolation（最近鄰插值）

- Bilinear Interpolation（雙線性插值）

- 自行手動實作兩種插值演算法，並與 OpenCV 提供的對應方法作比較

- 使用 Gradio 介面，提供滑桿調整旋轉角度（0°–360°）

1. **實驗結果**

|  |  |  |
| --- | --- | --- |
|  | 30度 ( 自製 ) | 30度 ( opencv ) |
| nn |  |  |
| 雙直 |  |  |

|  |  |  |
| --- | --- | --- |
|  | 80度 ( 自製 ) | 80度 ( opencv ) |
| nn |  |  |
| 雙直 |  |  |

1. **實驗結果討論**

**最近鄰（Nearest Neighbor）與雙線性（Bilinear）插值比較**

| **插值方式** | **優點** | **缺點** |
| --- | --- | --- |
| **Nearest Neighbor** | 實作簡單、運算速度快 可保留純黑純白等明確像素值 | 易產生鋸齒狀邊緣（aliasing） |
| **Bilinear** | 邊緣較平滑、畫質較自然 | 計算稍慢 |

整體而言，**Bilinear interpolation 在大多數旋轉角度下效果較為平滑自然**，適合用於視覺品質要求較高的情境。而 Nearest Neighbor 插值法因為只取最接近的整數像素，**容易在旋轉或放大時產生不連續的邊緣**，視覺上會出現明顯的鋸齒。

**為何 Nearest Neighbor 在 80° 時鋸齒感更明顯？**

以下為我個人整理的可能原因：

* **旋轉角度與原始方形邊緣方向產生「非對齊」的交角（非 45° 或 90°）**
* 在 80° 時：
  + 白方塊邊緣轉成「斜對角線」分布於像素網格上
  + 最近鄰插值**只能選擇最接近的單一像素點**，無法在斜線上「混合」像素
  + 因此會導致邊緣明顯呈現「階梯狀」的 **aliasing 鋸齒效應**

相比之下，Bilinear 插值則會依據周圍 4 個像素進行加權平均，即使在非整數位置也能插出柔和過渡的值，所以在 80 度這類非對齊角度下，**畫面更平順且自然**。