**學號：D12922014**

**姓名：莊謹譽**

**程式執行環境與使用套件**

* 執行環境:
  + Windows 10 專業版 22H2 (19045.3570)
  + Python - 3.12.0
  + Visual Studio Code - 1.83.1
  + Git for Windows - 2.42.0
* 使用套件:
  + opencv-python - 4.8.1.78

**For requirements 1 & 2, you need to show**

* **which function you use or implement**

程式使用 **opencv-python**所提供的相關函數與參數。

* + **函數**

|  |  |
| --- | --- |
| cv2.imread | 讀取圖檔 |
| cv2.imwrite | 寫入圖檔 |
| cv2.resize | 圖檔縮放 |

* + **參數**

|  |  |
| --- | --- |
| cv2.INTER\_LINEAR | 對應Bilinear |
| cv2.INTER\_CUBIC | 對應Bicubic |

* **how does your program work**

程式從**\_\_name\_\_ == “\_\_main\_\_”**開始，分別用**cv2.INTER\_LINEAR**及**cv2.INTER\_CUBIC**來呼叫自定義的**scale\_by\_method()**函式，之後**scale\_by\_method()**再以此參數呼叫**cv2.resize()**，達成**0.2x, 5x, 32x**的縮放，最後再用**cv2.imwrite()**將結果存起來。一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

* **how to use your program**

先開啟**Git Bash**，再切換到**d12922014**資料夾，然後依序執行以下指令

$ python -m venv .venv

$ source .venv/Scripts/activate

$ pip install opencv-python

$ python code/hw1.py

實際執行情況可參考下圖一張含有 文字, 電子產品, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

**For requirements 3 & 4, you need to provide**

* **Resulted images for comparison**

0.2x的情況下，Bilinear跟Bicubic看不出明顯差異。

|  |  |
| --- | --- |
| **Bilinear** | **Bicubic** |
| 一張含有 人的臉孔, 人員, 眼鏡, 前額 的圖片  自動產生的描述 | 一張含有 人的臉孔, 人員, 眼鏡, 前額 的圖片  自動產生的描述 |

5x的情況下，觀察左下角背景窗簾光線與右下角額頭反光，會看到Bicubic比Bilinear來的亮，而且面積比較多。此外，在黑色背景中，Bicubic的毛髮反射比較明顯，Bilinear則有點融入在背景中。總結來說，比起Bilinear，Bicubic明暗對比較強烈，線條較銳利。

|  |  |
| --- | --- |
| **Bilinear** | **Bicubic** |
| 一張含有 文字, 螢幕擷取畫面, 編輯, 皮膚 的圖片  自動產生的描述 | |

32x的情況下，相較Bilinear，Bicubic在邊界輪廓較明顯，而且非邊界處的色塊，也是Bicubic較大較清晰，此外，Bicubic在背景也是白的地方較白、黑的地方較黑，比較沒有像Bilinear糊成一片，而是有比較好的對比。

|  |  |
| --- | --- |
| **Bilinear** | **Bicubic** |
| 一張含有 文字, 螢幕擷取畫面 的圖片  自動產生的描述 | |

* **Explanation**

|  |  |
| --- | --- |
| **Bilinear** | **Bicubic** |
| 一張含有 文字, 螢幕擷取畫面, 行, 繪圖 的圖片  自動產生的描述 | undefined  R4  R3  R2  R1 |
| 先找到包含(x, y)的2\*2網格，再從網格的4個頂點對x方向做線性內插(上下各做一次)，得到R1、R2，之後再如法炮製，用R1、R2在y方向做線性內插，得到最後結果。  由於Bilinear只用2\*2網格上的4個頂點，因此，如果圖片放大，要填補的細節過多，會顯得有點模糊，反之縮小就比較沒什麼問題。  至於複雜度，由於每個點都需要鄰近4個點做3次內插，因此N個點時間複雜度為O(3N) = O(N)，比Bicubic快。 | 先找到包含(x, y)的4\*4網格，再從網格的16個頂點，由上而下，分別對x方向做三次多項式內插，得到R1、R2、R3、R4，之後再如法炮製，用R1、R2、R3、R4在y方向做線性內插，得到最後結果。  由於Bicubic用到4\*4網格上的16個頂點，因此，無論放大縮小，都能比Bilinear保留更多細節，讓線條較為銳利，對比較為明顯。  至於複雜度，由於每個點都需要鄰近16個點做5次內插，因此N個點時間複雜度為O(5N) = O(N)，比Bilinear慢。 |