|  |
| --- |
| 機器視覺作業報告 |
| 題目 |
|  |
|  |
| 姓名:岳紀伶  學號:01057113  日期:2024/04/16 |
|  |

|  |
| --- |
|  |

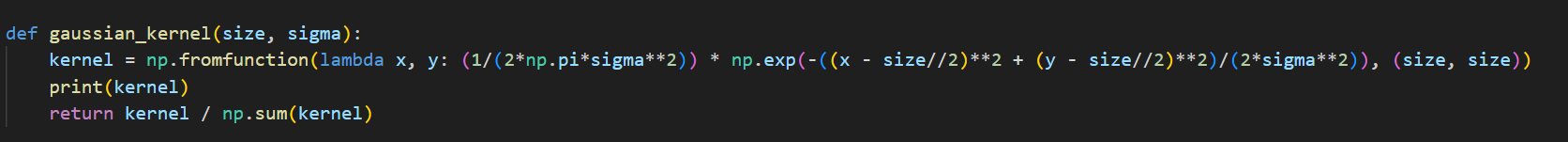
# 方法

做法:

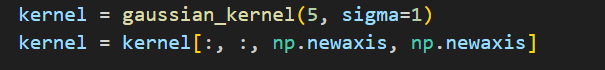
根據需要的filter製作kernel，Version1直接做高斯模糊，所以只需要做Gaussian filter就好。Version2需要先轉換到YUV座標再做高斯模糊，最後轉回RGB，所以需要增加一個轉換到YUV座標的矩陣，並透過將這個矩陣做inverse得到轉回RGB的矩陣。

重要程式片段:

Version1



藉由np.fromfunction從0到size把值帶入x、y，形成一個size\*size(5\*5)的矩陣，再把這個矩陣正規化得到Gaussian kernel

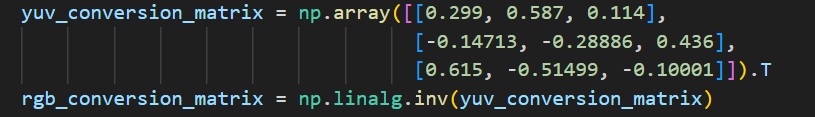


為了符合DepthwiseConv2D的格式，要把kernel擴充成四維，增加in\_channels和out\_channels



透過for迴圈依序讀入TestImage1、TestImage2、TestImage3，由於有三張測試照片，且每張測試照片要產生三種變化，加上原圖後共12張，形成4列三行的格式，因此用plt.subplot(4, 3, j)來表示圖片位置，j為原圖的位置，下一種變化的位置需要加三。

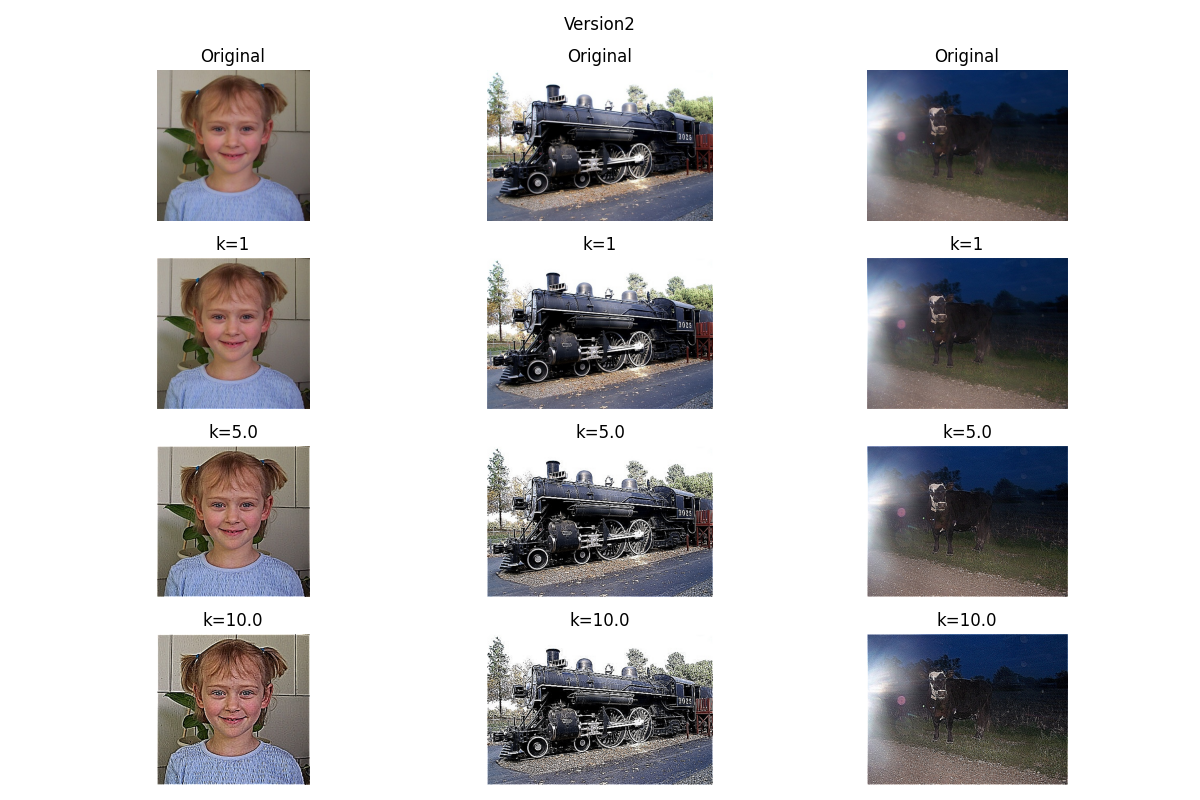
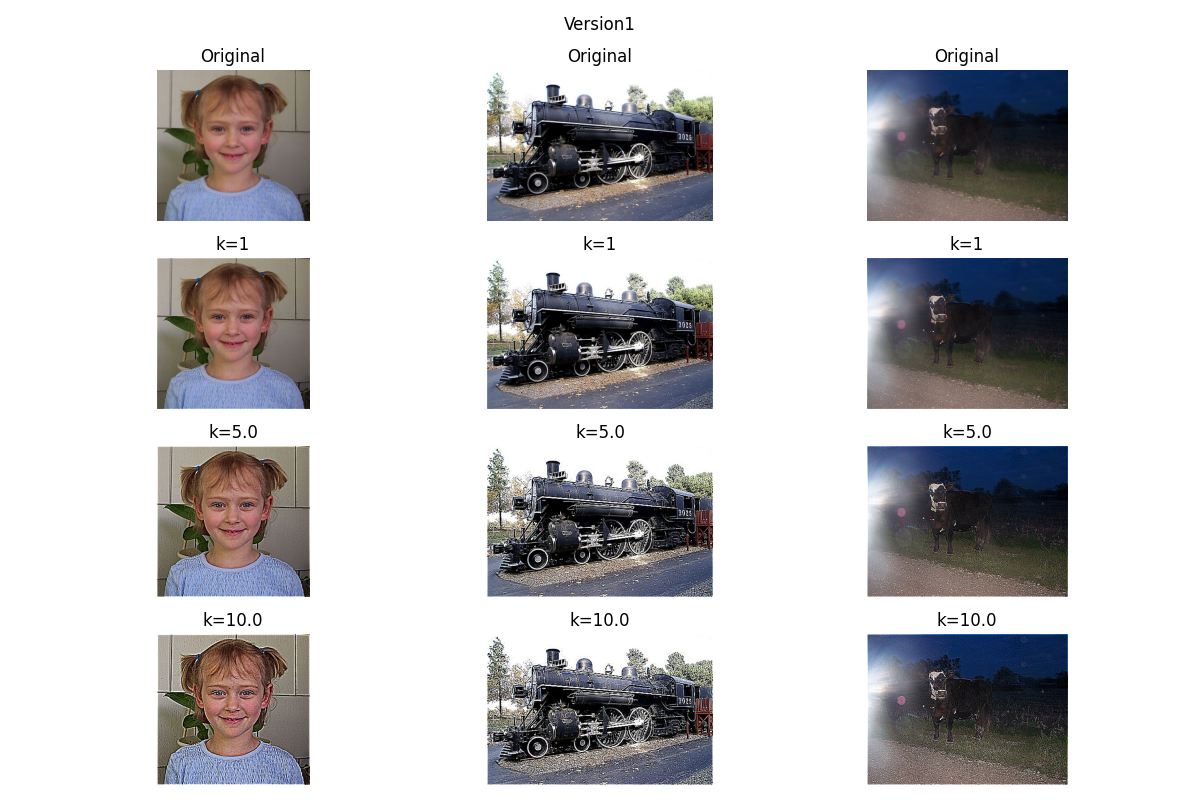
Version2



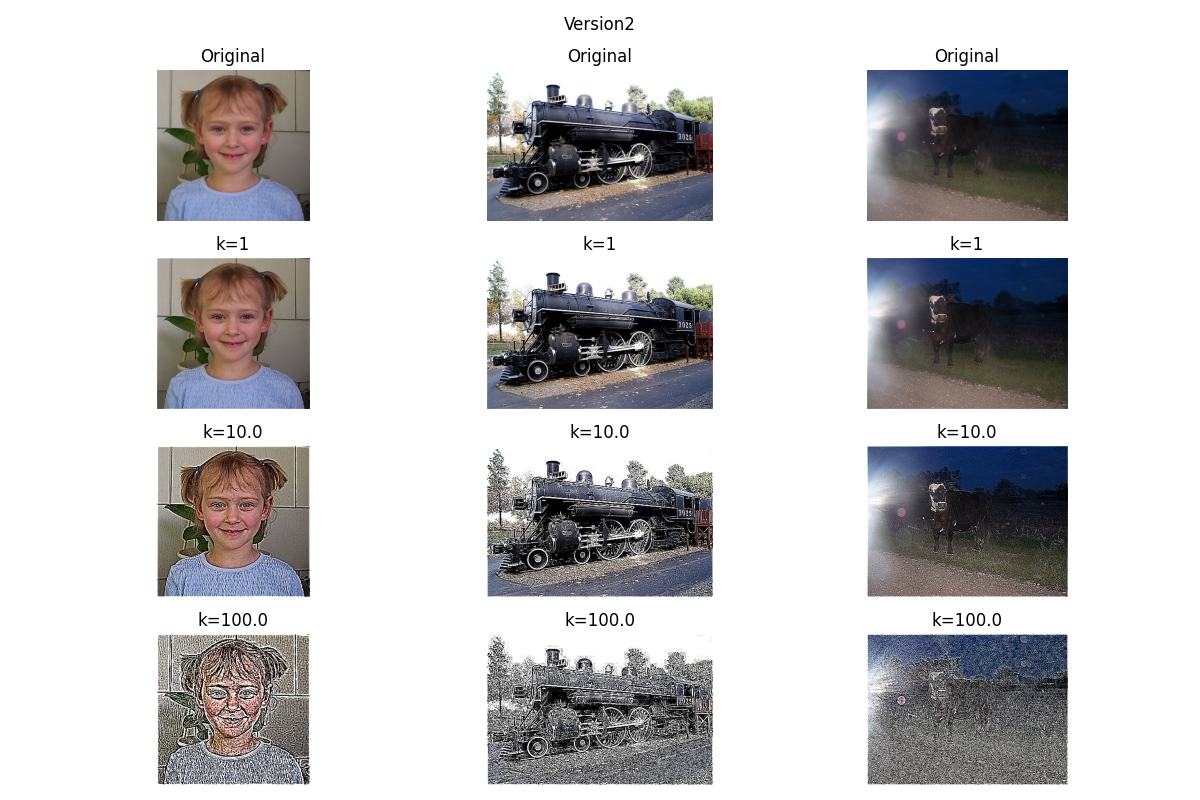
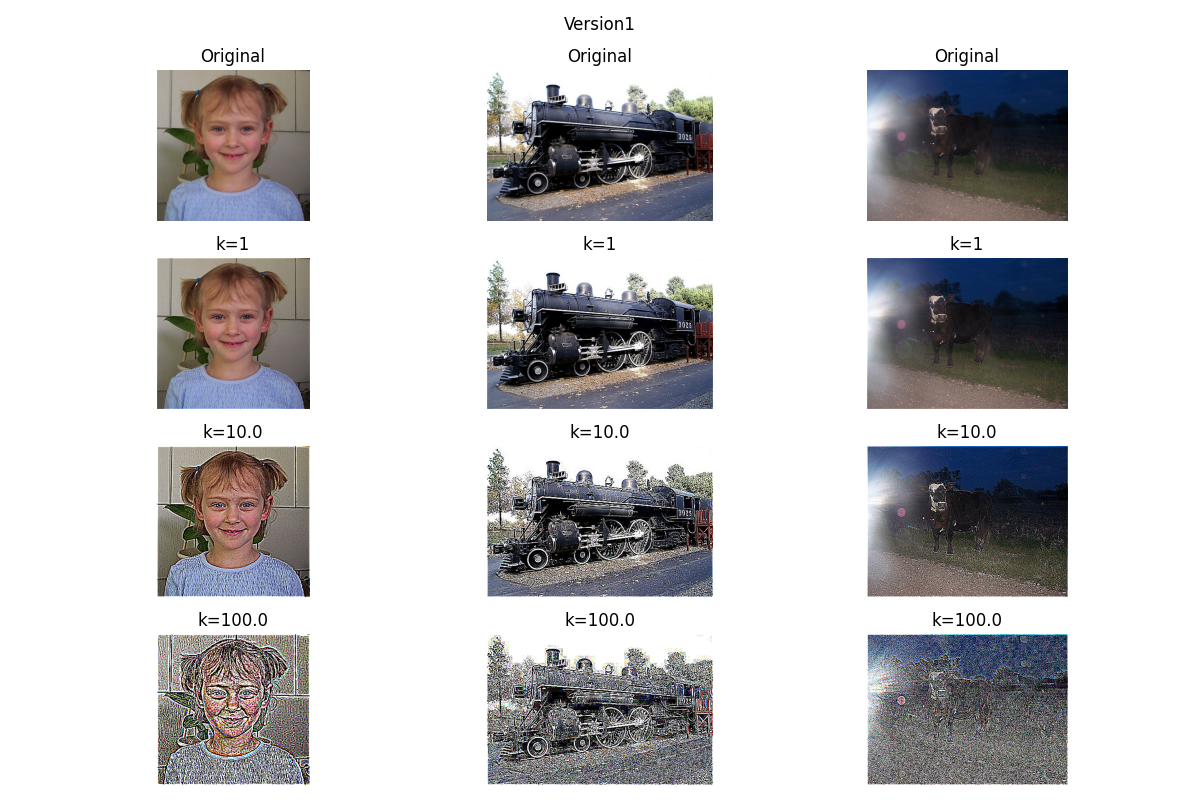
Version2在Version1的基礎上增加了RGB到YUV的轉換和YUV到RGB的轉換，YUV到RGB可以透過將RGB到YUV的轉換矩陣取inverse得到

# 結果

K=0.5、1.0、5.0、10.0



K=0.5、1.0、10.0、100.0



在k=0.5、1.0、5.0、10.0的情況下，兩個版本無法直接分辨差異。但若將k調大，可以發現在version1相較於version2顏色的飽和度更高

# 結論

在經過這次作業後，對於kernel的運作方式更加清楚，也發現如果要對RGB進行座標轉換需要將矩陣轉置才能得到正確的結果。另外在寫作業的過程中，我也發現我的寫法需要在keras 3中才能順利執行，否則會有TypeError。

# 參考文獻

Chatgpt

<https://keras.io/api/layers/convolution_layers/depthwise_convolution2d/>

<https://medium.com/@RiwajNeupane/convolutions-blurring-and-sharpening-images-44559460977d>

<https://antonmilev.medium.com/implement-image-color-filters-using-keras-conv2d-layers-3f2682105b7c>