|  |
| --- |
| 機器視覺作業報告 |
| Homework #2 |
| PyTorch 實作 Sharp Masking |
|  |
| 姓名:林宏儒  學號:01157139  日期:2025/3/30 |
|  |

|  |
| --- |
|  |

# 方法

1. Version1(分別對RGB三個通道做卷積)  
   參考老師提供的程式碼，補上設定卷積核的部分

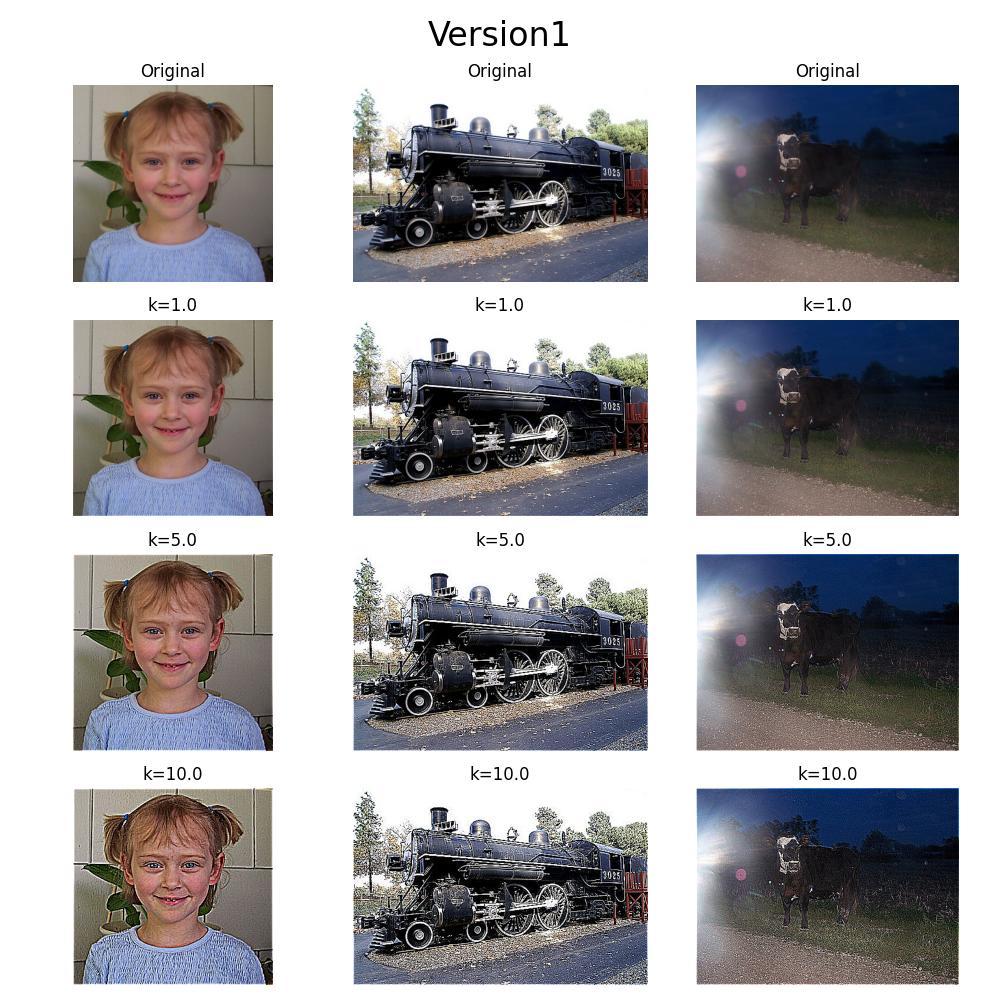
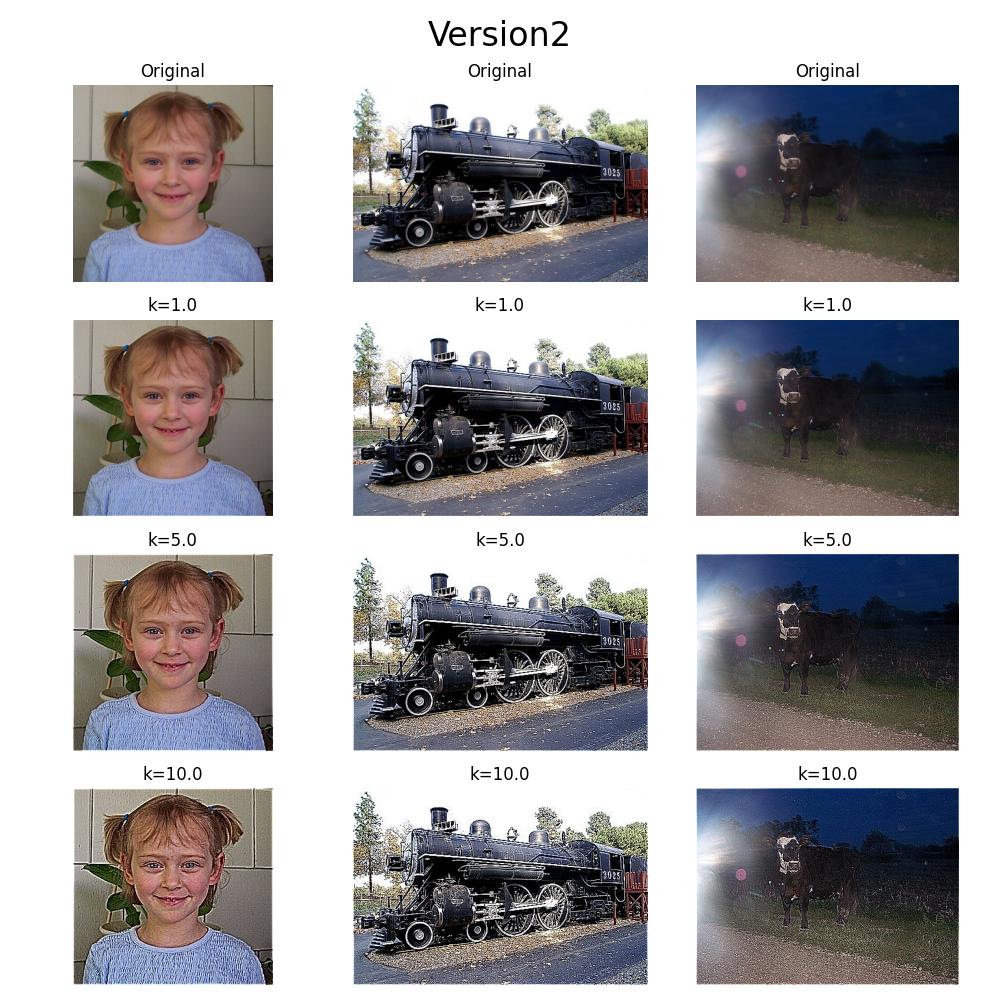
|  |
| --- |
| class DepthwiseConvModel(nn.Module):  def \_\_init\_\_(self):  super(DepthwiseConvModel, self).\_\_init\_\_()  self.depthwise\_conv = nn.Conv2d(in\_channels=3, out\_channels=3, kernel\_size=5, padding=2, groups=3, bias=False)    def forward(self, x, k):  x1 = self.depthwise\_conv(x)  return x + k \* (x - x1)  def get\_gaussian\_kernel(kernel\_size=5, sigma=1.0):  ax = torch.arange(-kernel\_size // 2 + 1., kernel\_size // 2 + 1.)  xx, yy = torch.meshgrid([ax, ax], indexing='ij')  kernel = torch.exp(-(xx\*\*2 + yy\*\*2) / (2. \* sigma\*\*2))  kernel = kernel / kernel.sum()  return kernel |

1. Version2(轉換成YUV色彩空間後對亮度做卷積)

因為只有一個通道要操作 所以初始化時設定的參數不一樣

|  |
| --- |
| class DepthwiseConvModel(nn.Module):  def \_\_init\_\_(self):  super(DepthwiseConvModel, self).\_\_init\_\_()  self.depthwise\_conv = nn.Conv2d(in\_channels=1, out\_channels=1, kernel\_size=5, padding=2, groups=1, bias=False)    def forward(self, x, k):  x1 = self.depthwise\_conv(x)  return x + k \* (x - x1)  def get\_gaussian\_kernel(kernel\_size=5, sigma=1.0):  ax = torch.arange(-kernel\_size // 2 + 1., kernel\_size // 2 + 1.)  xx, yy = torch.meshgrid([ax, ax], indexing='ij')  kernel = torch.exp(-(xx\*\*2 + yy\*\*2) / (2. \* sigma\*\*2))  kernel = kernel / kernel.sum()  return kernel |

# 結果

[Github](https://github.com/Ariexthesheep/HW2)  
輸出:  

# 結論

這次作業相較於上次簡單一些，主要的課題是練習卷積核的設定，除此之外，我對部分函式進行了封裝，只要更改k\_value與圖片路徑的陣列就可以處理其他輸入，最後再加上結果儲存與圖表顯示的程式碼，但這部分還沒有很好的模組化，完整的程式碼可以在上方的github連結中查看