

Mostafa Khaled Sa

```
import numpy as np
```

```
import cv2
```

```
from google.colab.patches import cv2_imshow
```

```
photo=cv2.imread('/content/WhatsApp Image 2023-04-12 at 14.16.20.jpg')
```

```
cv2_imshow(photo)
```

```
def avrage_3X3(old):
```

```
    image=cv2.copyMakeBorder(old,1,1,1,1,cv2.BORDER_REFLECT)
```

```
    templetm=np.ones([3,3])
```

```
    [row,col,chw]=np.shape(old)
```

```
    new_avg=np.zeros([row,col,chw])
```

```
    for ch in range(chw):
```

```
        for r in range(row):
```

```
            for c in range(col):
```

```
                new_avg[(r,c,chw)]=int((image[(r,c,chw)]*templetm[0][0]+image[(r+1,c,chw)]*templetm[1][0]+image[(r+2,c,chw)]*templetm[2][0]+image[(r,c+1,chw)]*templetm[0][1]+image[(r,c+2,chw)]*templetm[0][2]+image[(r+1,c+1,chw)]*templetm[1][1]+image[(r+1,c+2,chw)]*templetm[1][2]+image[(r+2,c+1,chw)]*templetm[2][1]+image[(r+2,c+2,chw)]*templetm[2][2])/9)
```

```
    new_avg=np.uint8(new_avg)
```

```
    return new_avg
```

```
X=avrage_3X3(photo)
```

```
cv2_imshow(X)
```

```
photo2=cv2.imread('/content/Screenshot 2023-04-13 165318.png')
```

```
# photo3=cv2.imread('/content/1_XAgKINgc2c2gNa2nV3zbNQ.png')
```

```
cv2_imshow(photo2)
```

```
# cv2_imshow(photo3)
```

```
def unsharppingdetection(old):
```

```
    image_avrage=avrage_3X3(old)
```

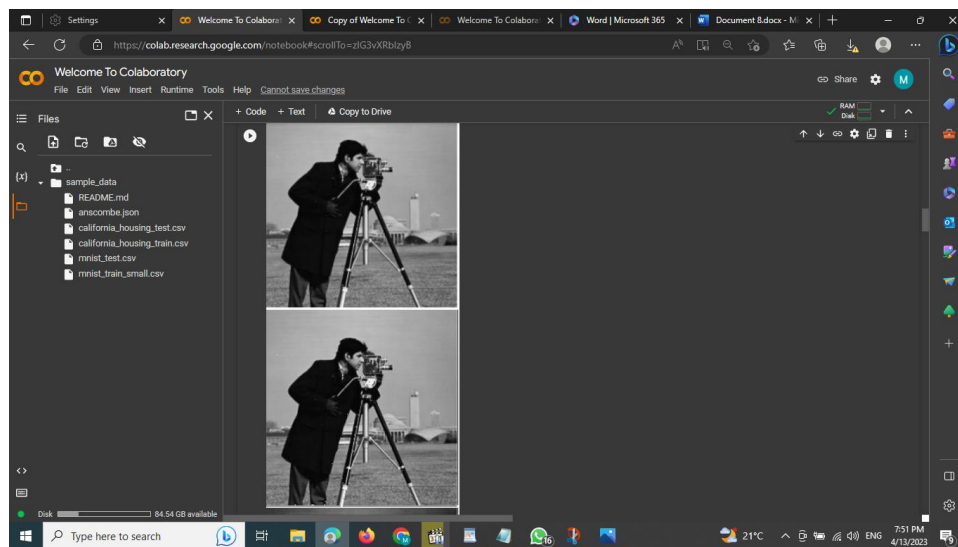
```
    image_avrage=photo2-image_avrage
```

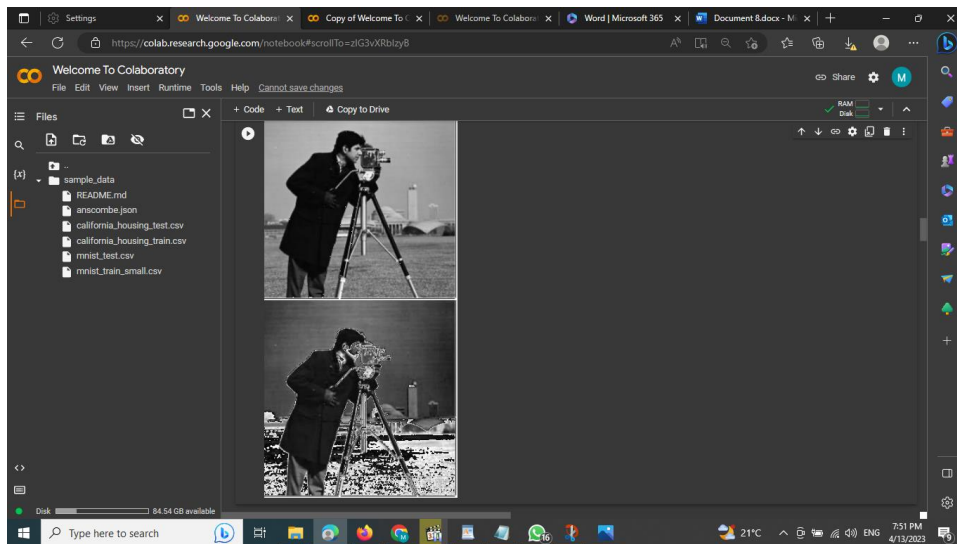
```
    cv2_imshow(image_avrage)
```

```
    image_avrage=image_avrage+photo2
```

```
    cv2_imshow(image_avrage)
```

```
unsharppingdetection(photo2)
```





```
import numpy as np

import cv2

from google.colab.patches import cv2_imshow

pho=cv2.imread('/content/railway-hdr-1361893.jpg')

cv2_imshow(pho)

def Guassian_function(image,sigma):

    N = int(3.7*sigma-0.5)

    mask_size = 2*N+1

    image = cv2.copyMakeBorder(image, N, N, N, N, cv2.BORDER_REFLECT)

    t = round(mask_size/2)

    x= range(-t, t)

    filter = np.zeros([mask_size, mask_size], dtype=float)

    coef=(1/(2*np.pi*(sigma**2)))

    for i in range(mask_size):

        for j in range(mask_size):

            power=-((x[i]**2)+(x[j]**2))/(2*(sigma**2))

            temp=float(coef*np.exp(power))
```

```
        filter[i,j]=temp
    return filter,N
```

```
def Smoothing_with_weighted_Filter( old,sigma ):
```

```
    [r1, c1, ch1] = np.shape(old)
    new = np.zeros([r1, c1, ch1])
```

```
    filter,N = Guassian_function(old,sigma)
```

```
    old=cv2.copyMakeBorder(old, N, N, N,N, cv2.BORDER_REFLECT)
```

```
    r,c,ch=old.shape
```

```
    for chh in range(0, ch):
```

```
        for i in range(N, r - N):
```

```
            for j in range(N, c - N):
```

```
                temp = old[i - N:i + N + 1, j - N:j + N + 1, chh]
```

```
                result = np.multiply(temp, filter)
```

```
                value = round(np.sum(result))
```

```
                new[i - N, j - N, chh] = value
```

```
    new=np.uint8(new)
```

```
    return (new)
```

```
# ggg=Guassian_function(pho,2)
```

```
f=Smoothing_with_weighted_Filter(pho,.5)
```

```

cv2_imshow(f)

photo2=cv2.imread('/content/Screenshot 2023-04-13 165318.png')

# photo3=cv2.imread('/content/1_XAgKINgc2c2gNa2nV3zbNQ.png')

cv2_imshow(photo2)

# cv2_imshow(photo3)

def unsharpingdetection(old):

    image_avrage=Smoothing_with_weighted_Filter(old,.5)

    image_avrage=photo2-image_avrage

    cv2_imshow(image_avrage)

    image_avrage=image_avrage+photo2

    cv2_imshow(image_avrage)

```

unsharpingdetection(photo2)

