

haoyuan9_mp2_part1_code

March 4, 2019

```
In [50]: import cv2
         from scipy.ndimage import gaussian_filter
         import matplotlib.pyplot as plt
         import skimage
         import numpy as np
         %matplotlib inline

In [51]: # Print input images
         c1 = cv2.imread("c1.jpg", 1).astype("float")
         c2 = cv2.imread("c2.jpg", 1).astype("float")
         plt.figure()
         plt.imshow(cv2.cvtColor(c1.astype("uint8"), cv2.COLOR_BGR2RGB))
         plt.figure()
         plt.imshow(cv2.cvtColor(c2.astype("uint8"), cv2.COLOR_BGR2RGB))
         plt.show()
```





```
In [52]: # Print output images
c1 = c1.transpose(2, 0, 1)
c2 = c2.transpose(2, 0, 1)

for i in range(3):
    c1[i] = c1[i] - gaussian_filter(c1[i], 4)
c1[c1 < 0] = 0

for i in range(3):
    c2[i] = gaussian_filter(c2[i], 5)

plt.figure()
plt.imshow(cv2.cvtColor(c1.transpose(1, 2, 0).astype("uint8"), cv2.COLOR_BGR2RGB))
cv2.imwrite("c1_filtered.jpg", c1.transpose(1, 2, 0))
plt.figure()
plt.imshow(cv2.cvtColor(c2.transpose(1, 2, 0).astype("uint8"), cv2.COLOR_BGR2RGB))
cv2.imwrite("c2_filtered.jpg", c2.transpose(1, 2, 0))

output = (c1 + c2)
output[output>255] = 255
output[output<0] = 0
output = output.transpose(1, 2, 0)

plt.figure()
plt.imshow(cv2.cvtColor(output.astype("uint8"), cv2.COLOR_BGR2RGB))
```

```
plt.show()
cv2.imwrite("part1_1.jpg", output)
```





Out [52]: True

```
In [53]: # Print input images
m1 = cv2.imread("m1.jpg", 1).astype("float")[::-12,:-10,:]
m2 = cv2.imread("m2.jpg", 1).astype("float")[12:,10:,:]
plt.figure()
plt.imshow(cv2.cvtColor(m1.astype("uint8"), cv2.COLOR_BGR2RGB))
plt.figure()
plt.imshow(cv2.cvtColor(m2.astype("uint8"), cv2.COLOR_BGR2RGB))
plt.show()
```



```
In [54]: # Print output images  
         m1 = m1.transpose(2, 0, 1)  
         m2 = m2.transpose(2, 0, 1)
```

```

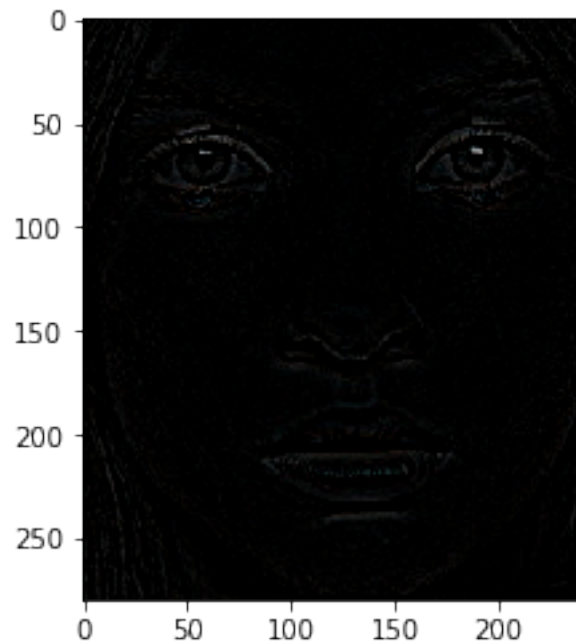
for i in range(3):
    m1[i] = m1[i] - gaussian_filter(m1[i], 2)
m1[m1 < 0] = 0
for i in range(3):
    m2[i] = gaussian_filter(m2[i], 4)

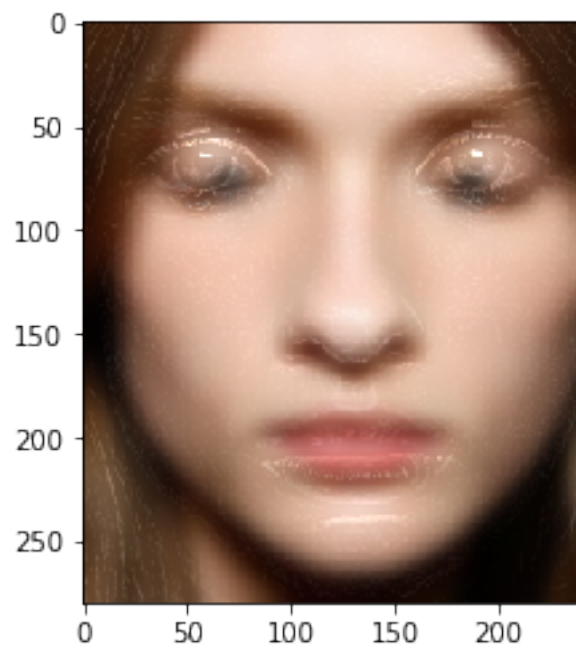
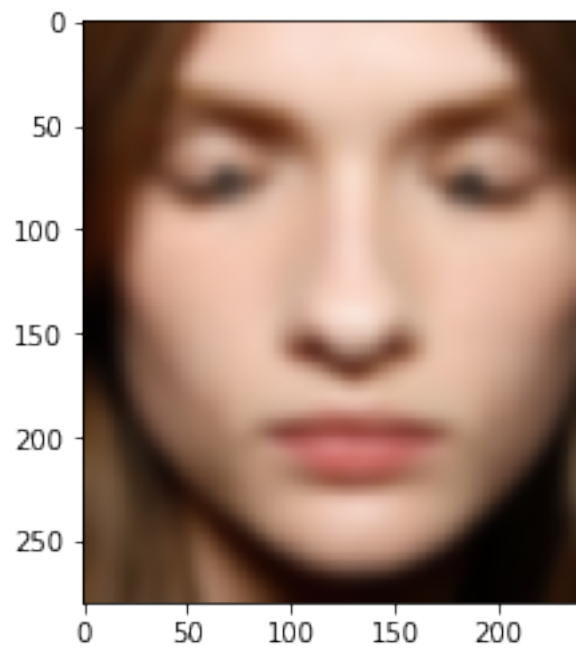
plt.figure()
plt.imshow(cv2.cvtColor(m1.transpose(1, 2, 0).astype("uint8"), cv2.COLOR_BGR2RGB))
cv2.imwrite("m1_filtered.jpg", m1.transpose(1, 2, 0))
plt.figure()
plt.imshow(cv2.cvtColor(m2.transpose(1, 2, 0).astype("uint8"), cv2.COLOR_BGR2RGB))
cv2.imwrite("m2_filtered.jpg", m2.transpose(1, 2, 0))

output = (m1 + m2)
output[output>255] = 255
output[output<0] = 0
output = output.transpose(1, 2, 0)

plt.figure()
plt.imshow(cv2.cvtColor(output.astype("uint8"), cv2.COLOR_BGR2RGB))
plt.show()
cv2.imwrite("part1_3.jpg", output)

```

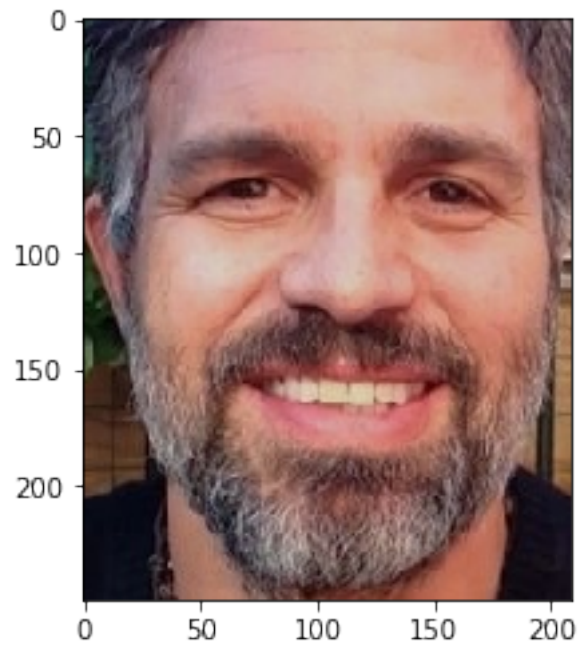




Out[54]: True

```
In [55]: # Print input images
         m3 = cv2.imread("m3.jpg", 1).astype("float")
```

```
m4 = cv2.imread("m4.jpg", 1).astype("float")[82:,15:-16,:]  
plt.figure()  
plt.imshow(cv2.cvtColor(m3.astype("uint8"), cv2.COLOR_BGR2RGB))  
plt.figure()  
plt.imshow(cv2.cvtColor(m4.astype("uint8"), cv2.COLOR_BGR2RGB))  
plt.show()
```




```

In [56]: # Print output images
m3 = m3.transpose(2, 0, 1)
m4 = m4.transpose(2, 0, 1)

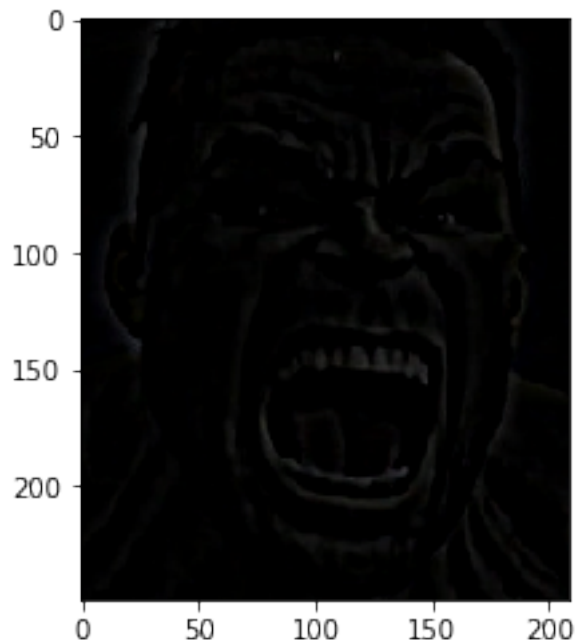
for i in range(3):
    m3[i] = m3[i] - gaussian_filter(m3[i], 4)
m3[m3 < 0] = 0
for i in range(3):
    m4[i] = gaussian_filter(m4[i], 5)

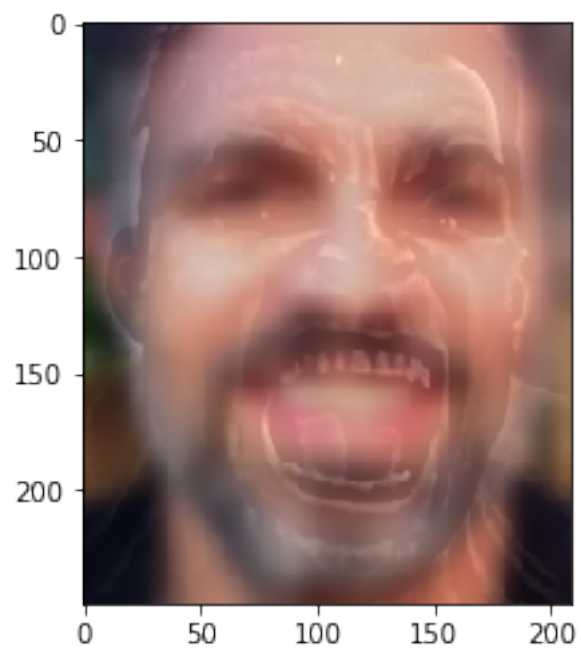
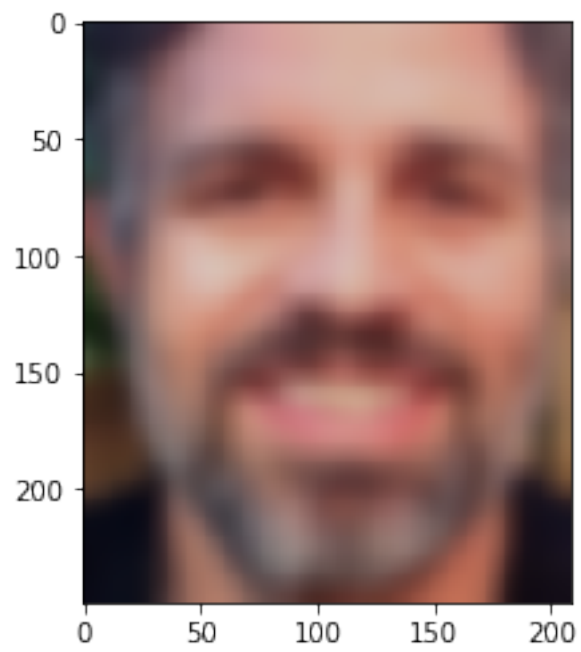
plt.figure()
plt.imshow(cv2.cvtColor(m3.transpose(1, 2, 0).astype("uint8"), cv2.COLOR_BGR2RGB))
cv2.imwrite("m3_filtered.jpg", m3.transpose(1, 2, 0))
plt.figure()
plt.imshow(cv2.cvtColor(m4.transpose(1, 2, 0).astype("uint8"), cv2.COLOR_BGR2RGB))
cv2.imwrite("m4_filtered.jpg", m4.transpose(1, 2, 0))

output = (m3 + m4)
output[output>255] = 255
output[output<0] = 0
output = output.transpose(1, 2, 0)

plt.figure()
plt.imshow(cv2.cvtColor(output.astype("uint8"), cv2.COLOR_BGR2RGB))
plt.show()
cv2.imwrite("part1_4.jpg", output)

```





Out [56]: True

```
In [ ]:
```