

```

import cv2
import numpy as np
from google.colab.patches import cv2_imshow

def sensor_level_fusion(img1, img2):
    img1_gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
    img2_gray = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)

    flow = cv2.calcOpticalFlowFarneback(img1_gray, img2_gray, None, 0.5, 3, 15, 3, 5, 1.2, 0)

    fused_image = np.zeros_like(img1)

    for i in range(fused_image.shape[0]):
        for j in range(fused_image.shape[1]):
            x_flow, y_flow = flow[i, j]
            source_pixel1 = img1_gray[i, j]

            # Calculate the coordinates for img2 based on the optical flow
            i2 = int(i + y_flow)
            j2 = int(j + x_flow)

            # Ensure the calculated coordinates are within bounds
            if i2 >= 0 and i2 < img2_gray.shape[0] and j2 >= 0 and j2 < img2_gray.shape[1]:
                source_pixel2 = img2_gray[i2, j2]
                fused_image[i, j] = (source_pixel1 + source_pixel2) / 2
            else:
                fused_image[i, j] = source_pixel1 # Use the original pixel if out of bounds

    return fused_image

```

```

img1 = cv2.imread("/content/drive/MyDrive/Datasets/face_img_1.jpg")
img2 = cv2.imread("/content/drive/MyDrive/Datasets/face_img_2.jpg")

```

[https://colab.research.google.com/drive/1kpTL2nMzesQ7y1uB7yMFjDx0JKwalt#scrollTo=\\_nsH08SR0QkB&printMode=true](https://colab.research.google.com/drive/1kpTL2nMzesQ7y1uB7yMFjDx0JKwalt#scrollTo=_nsH08SR0QkB&printMode=true)

1/4

10/17/23, 5:17 AM

senor\_level\_fusion.ipynb - Colaboratory

```
cv2_imshow(img1)
```



```
cv2_imshow(img2)
```



```

img2 = cv2.resize(img2, (img1.shape[1], img1.shape[0]))
fused_image = sensor_level_fusion(img1, img2)

```

[https://colab.research.google.com/drive/1kpTL2nMzesQ7y1uB7yMFjDx0JKwalt#scrollTo=\\_nsH08SR0QkB&printMode=true](https://colab.research.google.com/drive/1kpTL2nMzesQ7y1uB7yMFjDx0JKwalt#scrollTo=_nsH08SR0QkB&printMode=true)

2/4

```
cv2.imshow(fused_image)
cv2.waitKey(0)
cv2.destroyAllWindows()

<ipython-input-1-b14041217de2>:25: RuntimeWarning: overflow encountered in ubyte_s
fused_image[i, j] = (source_pixel1 + source_pixel2) / 2
```

