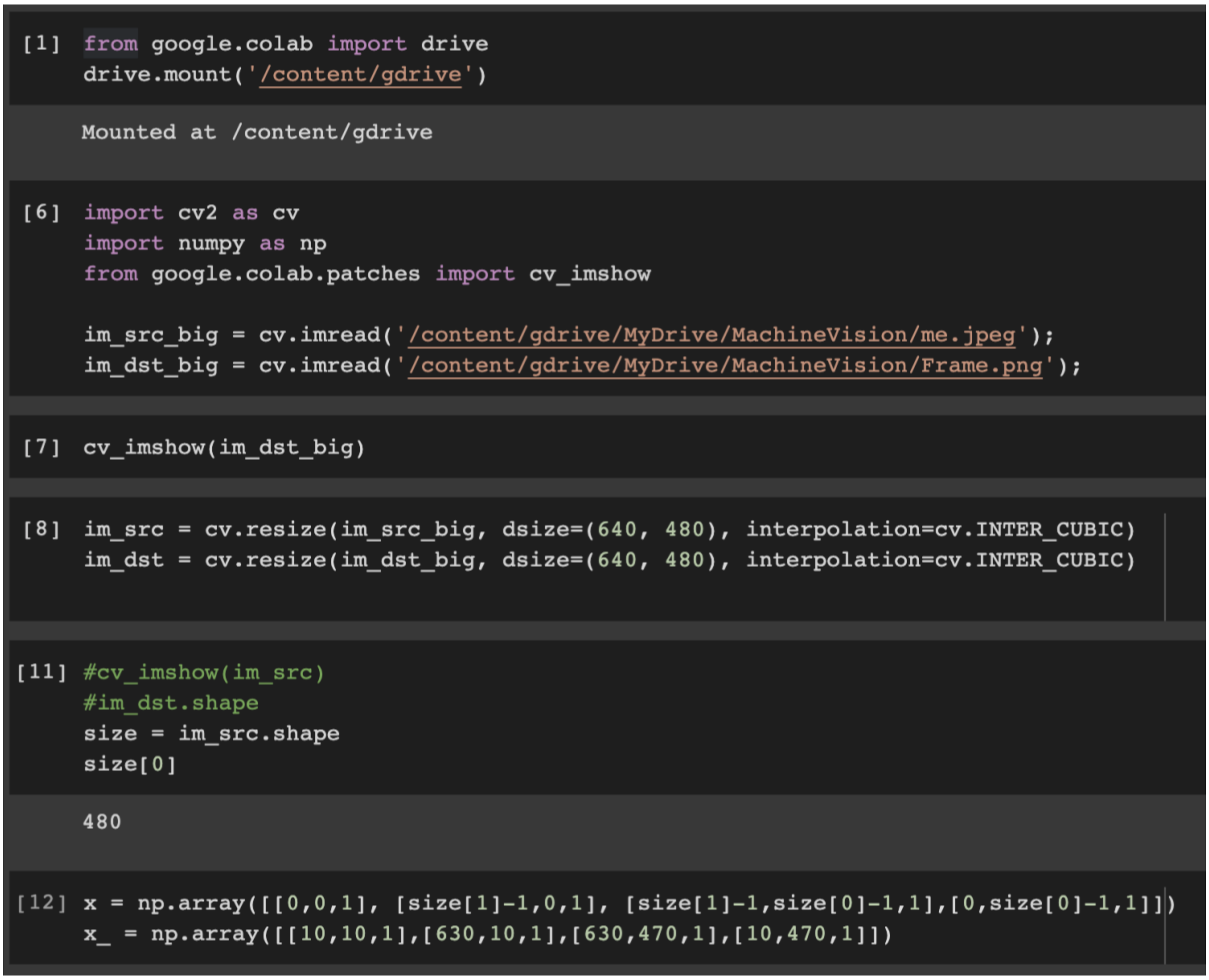
머신비전 시스템 과제5

18011789 조혜수

< 1번 >

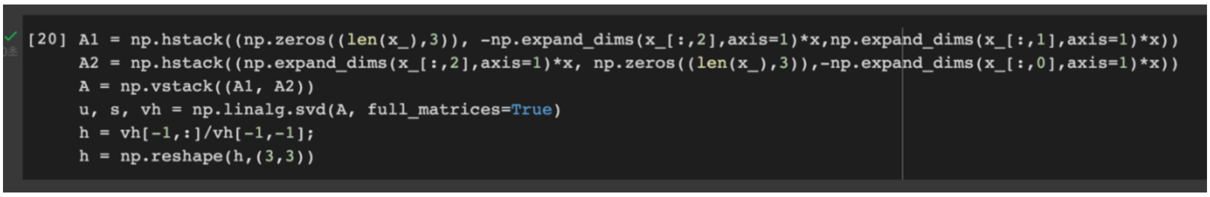
원본 영상

1)



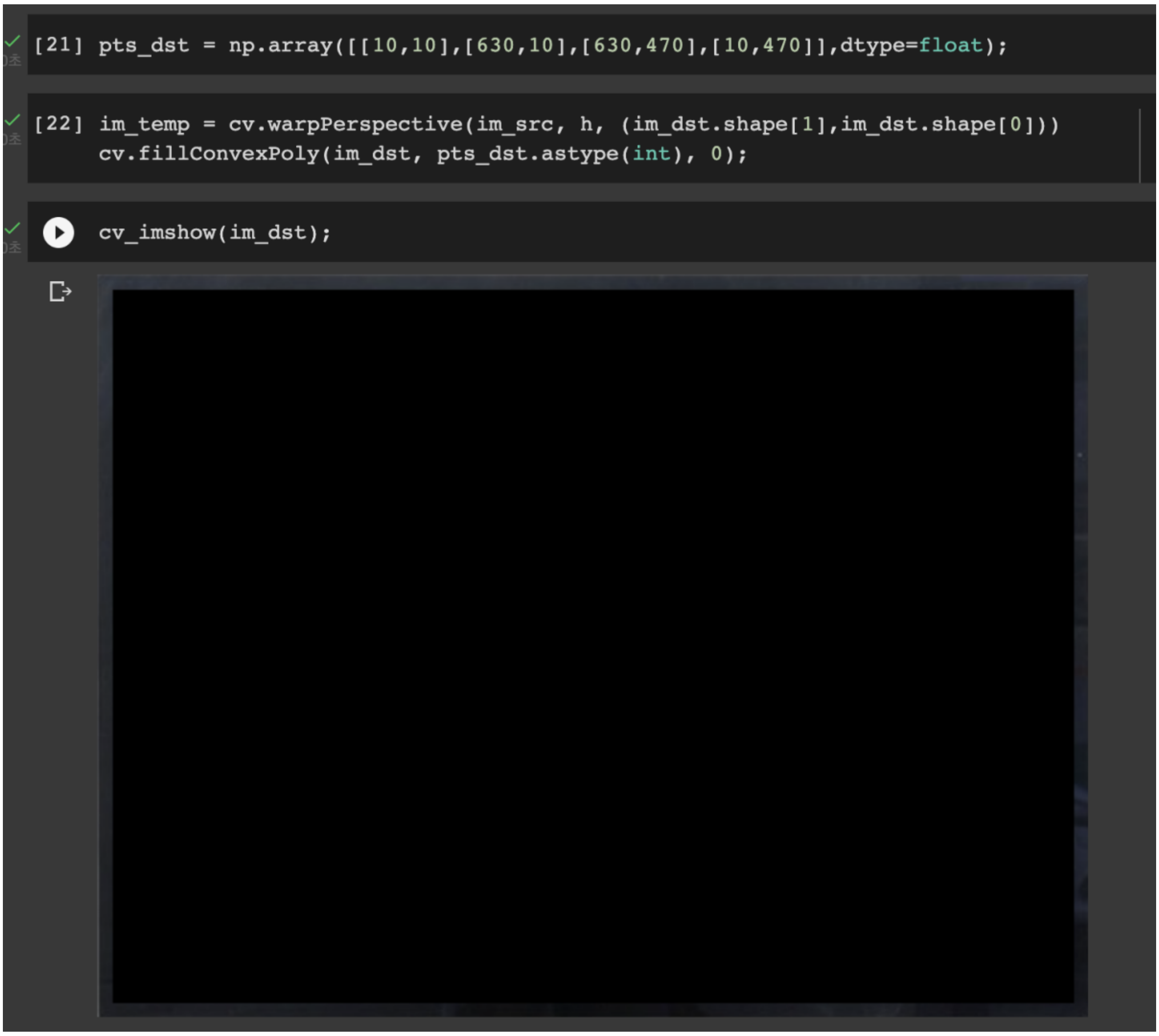
```
[1]  from google.colab import drive
     drive.mount('/content/gdrive')

     Mounted at /content/gdrive

[6]  import cv2 as cv
     import numpy as np
     from google.colab.patches import cv_imshow

     im_src_big = cv.imread('/content/gdrive/MyDrive/MachineVision/me.jpeg');
     im_dst_big = cv.imread('/content/gdrive/MyDrive/MachineVision/Frame.png');

[7]  cv_imshow(im_dst_big)

[8]  im_src = cv.resize(im_src_big, dsize=(640, 480), interpolation=cv.INTER_CUBIC)
     im_dst = cv.resize(im_dst_big, dsize=(640, 480), interpolation=cv.INTER_CUBIC)

[11] #cv_imshow(im_src)
     #im_dst.shape
     size = im_src.shape
     size[0]

     480

[12] x = np.array([[0,0,1], [size[1]-1,0,1], [size[1]-1,size[0]-1,1],[0,size[0]-1,1]])
     x_ = np.array([[10,10,1],[630,10,1],[630,470,1],[10,470,1]])
```

2)



```
[20] A1 = np.hstack((np.zeros((len(x_),3)), -np.expand_dims(x_[:,2],axis=1)*x,np.expand_dims(x_[:,1],axis=1)*x))
     A2 = np.hstack((np.expand_dims(x_[:,2],axis=1)*x, np.zeros((len(x_),3)),-np.expand_dims(x_[:,0],axis=1)*x))
     A = np.vstack((A1, A2))
     u, s, vh = np.linalg.svd(A, full_matrices=True)
     h = vh[-1,:]/vh[-1,-1];
     h = np.reshape(h,(3,3))
```

3)

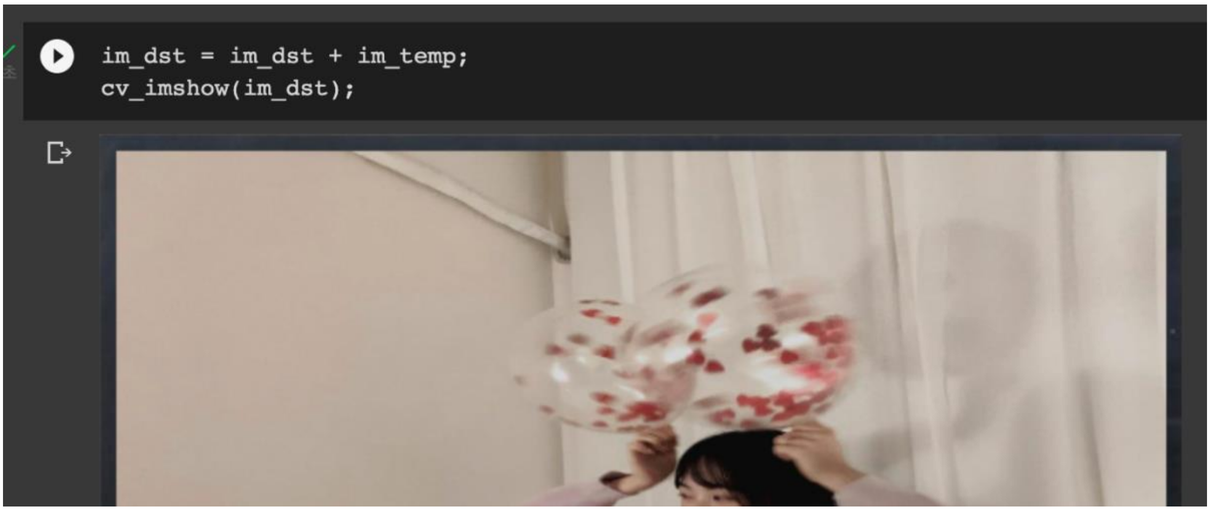


```
[21] pts_dst = np.array([[10,10],[630,10],[630,470],[10,470]],dtype=float);
```

```
[22] im_temp = cv.warpPerspective(im_src, h, (im_dst.shape[1],im_dst.shape[0]))
     cv.fillConvexPoly(im_dst, pts_dst.astype(int), 0);
```

```
cv_imshow(im_dst);
```

4)



```
im_dst = im_dst + im_temp;
cv_imshow(im_dst);
```

결과 영상

< 2번 >

원본영상

지금 우리는

코로나 19 바이러스

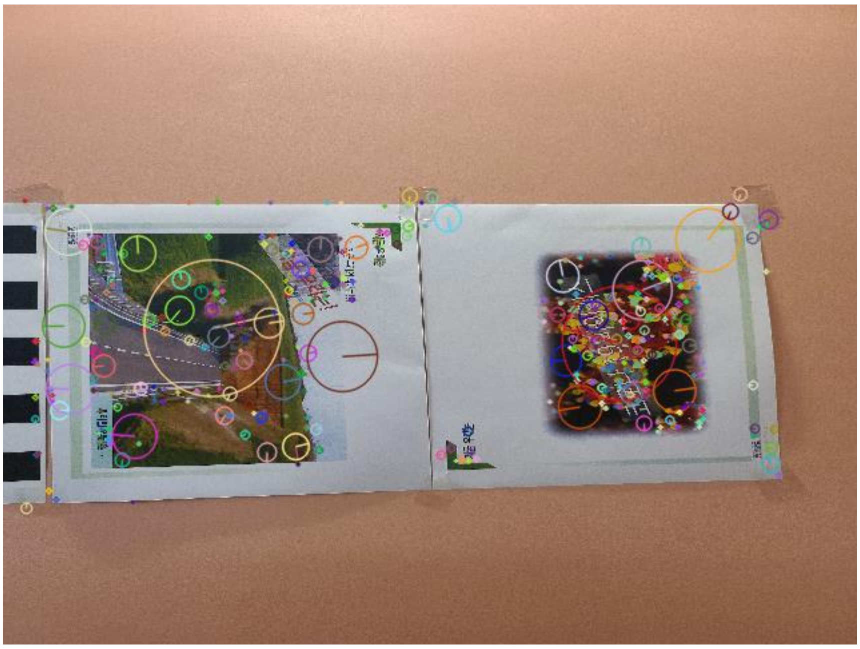
지금 우리는

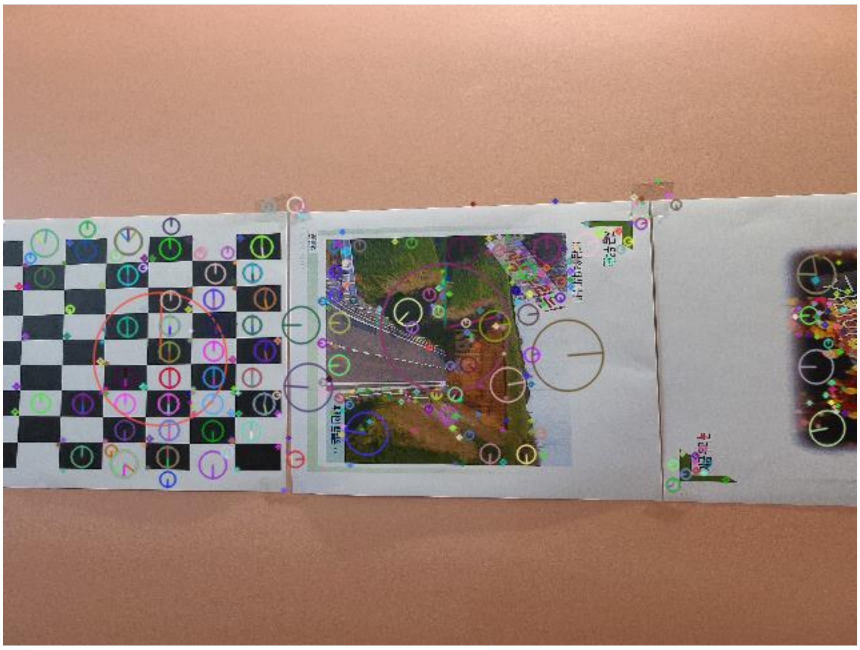
1. 지금 우리의 시대는!!!

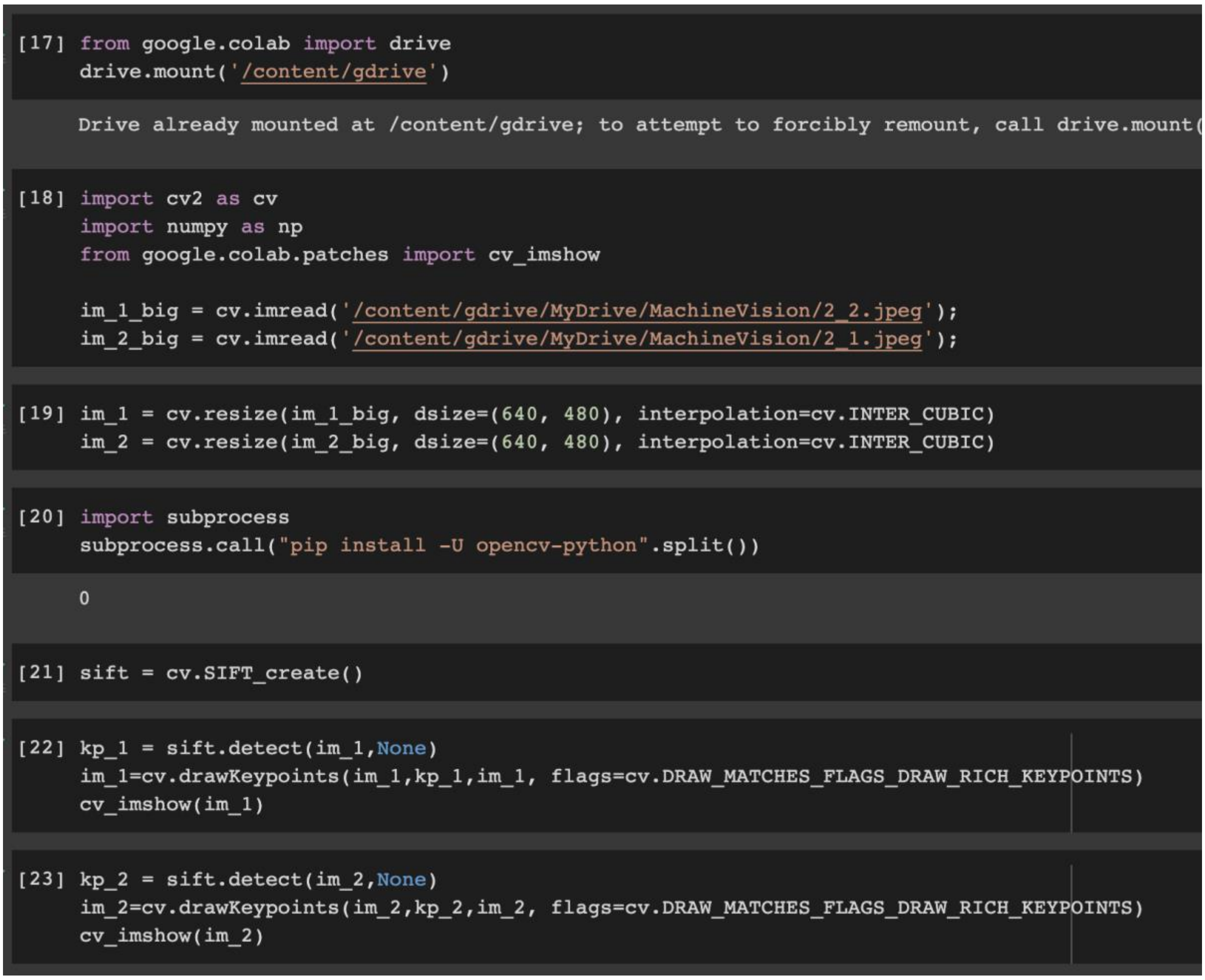
지금 우리는

1. 지금 우리의 시대는!!!

2-2) Blob 검출

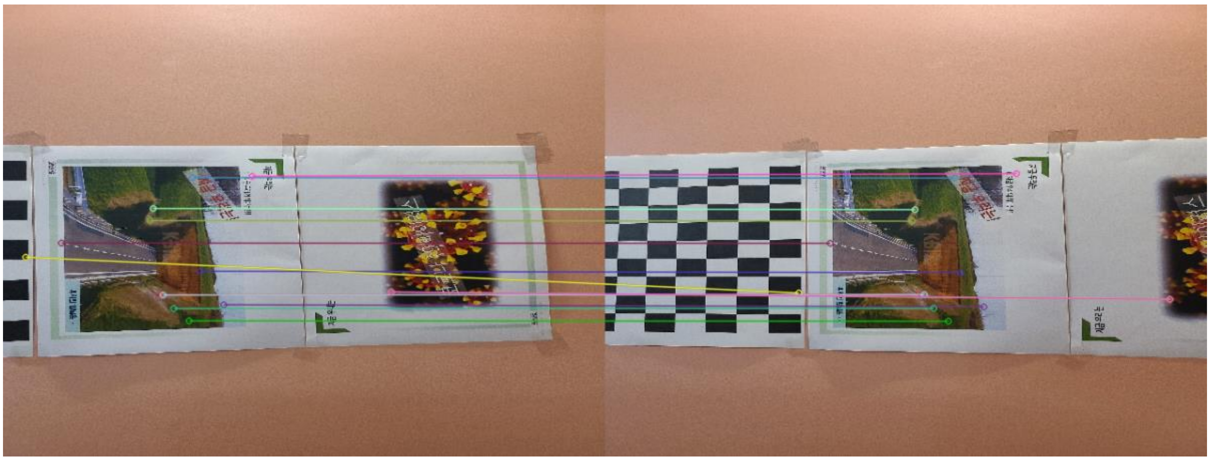


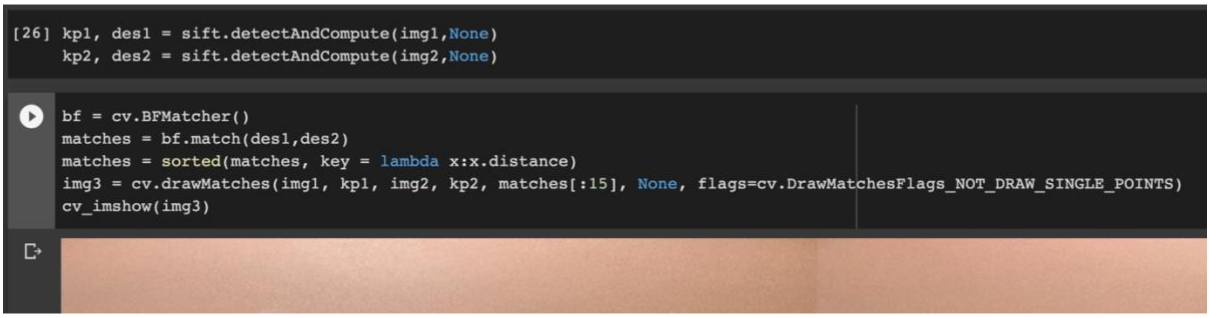




```
[17] from google.colab import drive
     drive.mount('/content/gdrive')

     Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount(

[18] import cv2 as cv
     import numpy as np
     from google.colab.patches import cv_imshow

     im_1_big = cv.imread('/content/gdrive/MyDrive/MachineVision/2_2.jpeg');
     im_2_big = cv.imread('/content/gdrive/MyDrive/MachineVision/2_1.jpeg');

[19] im_1 = cv.resize(im_1_big, dsize=(640, 480), interpolation=cv.INTER_CUBIC)
     im_2 = cv.resize(im_2_big, dsize=(640, 480), interpolation=cv.INTER_CUBIC)

[20] import subprocess
     subprocess.call("pip install -U opencv-python".split())

     0

[21] sift = cv.SIFT_create()

[22] kp_1 = sift.detect(im_1,None)
     im_1=cv.drawKeypoints(im_1,kp_1,im_1, flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
     cv_imshow(im_1)

[23] kp_2 = sift.detect(im_2,None)
     im_2=cv.drawKeypoints(im_2,kp_2,im_2, flags=cv.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
     cv_imshow(im_2)
```

2-3) Matching

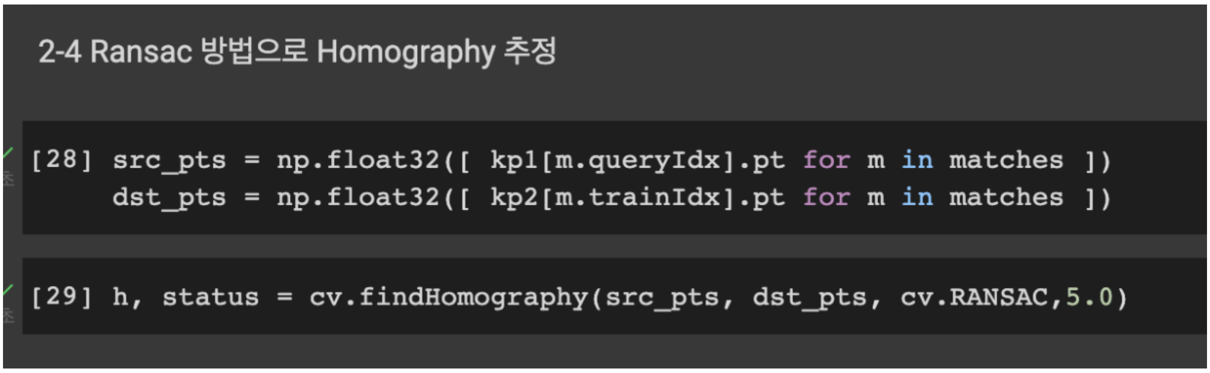




```
[26] kp1, des1 = sift.detectAndCompute(img1,None)
     kp2, des2 = sift.detectAndCompute(img2,None)
```

```
bf = cv.BFMatcher()
matches = bf.match(des1,des2)
matches = sorted(matches, key = lambda x:x.distance)
img3 = cv.drawMatches(img1, kp1, img2, kp2, matches[:15], None, flags=cv.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)
cv_imshow(img3)
```

2-4)

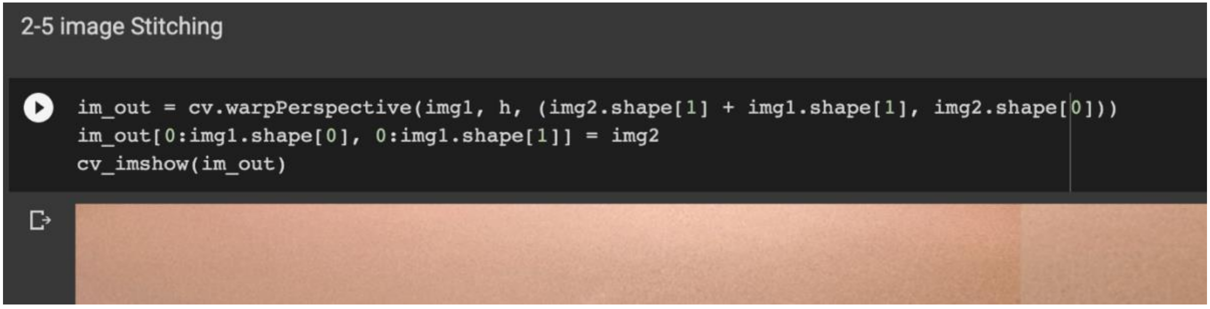


2-4 Ransac 방법으로 Homography 추정

```
[28] src_pts = np.float32([ kp1[m.queryIdx].pt for m in matches ])
     dst_pts = np.float32([ kp2[m.trainIdx].pt for m in matches ])
```

```
[29] h, status = cv.findHomography(src_pts, dst_pts, cv.RANSAC,5.0)
```

2-5)



2-5 image Stitching

```
im_out = cv.warpPerspective(img1, h, (img2.shape[1] + img1.shape[1], img2.shape[0]))
im_out[0:img1.shape[0], 0:img1.shape[1]] = img2
cv_imshow(im_out)
```

결과 영상