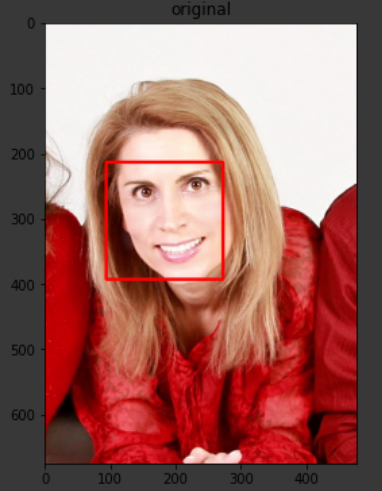
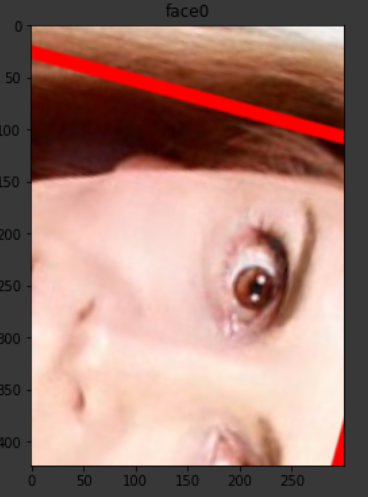
1.

(myFace\_Alignment.ipynb 파일을 활용하여 이미지 변화

교수님이 주신 예제의 랜드마크는 5개지만 현재 제가 사용하려는 것은 68개

그 때문인지 랜드마크 위치가 달라서 결과도 약간 이상하게 나온 것 같음



matplotlib.rcParams['figure.figsize'] = (6.0,6.0)

matplotlib.rcParams['image.cmap'] = 'gray'

np.set\_printoptions(precision=2)

#코딩 시작.

#얼굴 검출기 객체 생성

faceDetector = dlib.get\_frontal\_face\_detector()

#얼굴 랜드마크 검출기 객체 생성

#5 -> 68파일로 변경

landmarkDetector = dlib.shape\_predictor("data/models/shape\_predictor\_68\_face\_landmarks.dat")

#이미지 읽기

im = cv2.imread("images/face2.png")

#im = cv2.imread("./images/tv\_2person.jpg")

# 출력 얼굴 영상 크기 : 300x? (?는 원본과 가로세로비율 맞도록 자동설정)

w = 300

h = np.int(w \* im.shape[0]/im.shape[1])

#이미지에서 얼굴 검출하기

faceRects = faceDetector(im, 0)

print("Number of faces detected: ",len(faceRects))

#얼굴 주위에 사각형을 그리자.

for i in range(0, len(faceRects)):

# draw rect on detected face

x1 = faceRects[i].left()

y1 = faceRects[i].top()

x2 = faceRects[i].right()

y2 = faceRects[i].bottom()

cv2.rectangle(im, (x1, y1), (x2, y2), (0, 0, 255), 3)

print('face {} : ({},{})-({},{})'.format(i, x1, y1, x2, y2))

# For every face rectangle, run landmarkDetector

# 얼굴 랜드마크 검출해서 눈 좌표얻기

landmarks = landmarkDetector(im, faceRects[i])

point2 = (landmarks.part(2).x, landmarks.part(2).y) # right eye

point0 = (landmarks.part(0).x, landmarks.part(0).y) # left eye

eyecornerSrc = [point2, point0]

# Corners of the eye in normalized image

# 새 이미지에 눈을 위치시킬 좌표행렬인듯

eyecornerDst = [(np.int(0.3 \* w), np.int(h/3)),

(np.int(0.7 \* w), np.int(h/3))]

print('eyecornerSrc :', np.array([eyecornerSrc]))

print('eyecornerDst :', eyecornerDst)

# get transform matrix

# 변환 행렬을 생성하는 함수.

# 눈좌표를 원하는 곳으로 옮길 행열을 생성하자(이미지 눈좌표,원하는 곳의 좌표)

ret = cv2.estimateAffinePartial2D(np.array([eyecornerSrc]), np.array([eyecornerDst]))

xform = ret[0]

print('xform :', xform)

# apply transform

# 행열을 이용해서 이미지를 이동,,변환

imOut = cv2.warpAffine(im, xform, (w, h))

plt.figure()

plt.imshow(imOut[:,:,::-1]); plt.title('face'+np.str(i))

plt.figure()

plt.imshow(im[:,:,::-1])

plt.title("original")

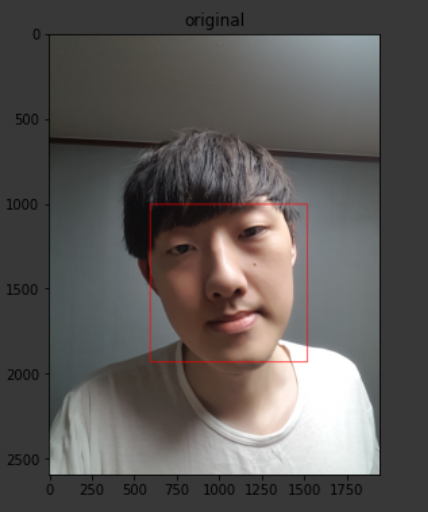
plt.show()

2.

목표 : 내 사진을 이용해서 이미지를 회전

빨간 박스로 얼굴을 인식함을 보여주고 출력에서는 지움

랜드마크를 이용해 눈의 위치를 따온 뒤, 이를 이용해 수평하도록 이미지를 회전



matplotlib.rcParams['figure.figsize'] = (6.0,6.0)

matplotlib.rcParams['image.cmap'] = 'gray'

np.set\_printoptions(precision=2)

#코딩 시작.

#얼굴 검출기 객체 생성

faceDetector = dlib.get\_frontal\_face\_detector()

#얼굴 랜드마크 검출기 객체 생성

#5 -> 68파일로 변경

landmarkDetector = dlib.shape\_predictor("data/models/shape\_predictor\_5\_face\_landmarks.dat")

#이미지 읽기

im = cv2.imread("images/self\_image2.jpg")

#im = cv2.imread("./images/tv\_2person.jpg")

# 출력 얼굴 영상 크기 : 300x? (?는 원본과 가로세로비율 맞도록 자동설정)

w = 300

h = np.int(w \* im.shape[0]/im.shape[1])

#이미지에서 얼굴 검출하기

faceRects = faceDetector(im, 0)

print("Number of faces detected: ",len(faceRects))

#얼굴 주위에 사각형을 그리자.

for i in range(0, len(faceRects)):

# draw rect on detected face

x1 = faceRects[i].left()

y1 = faceRects[i].top()

x2 = faceRects[i].right()

y2 = faceRects[i].bottom()

##cv2.rectangle(im, (x1, y1), (x2, y2), (0, 0, 255), 3)

print('face {} : ({},{})-({},{})'.format(i, x1, y1, x2, y2))

# For every face rectangle, run landmarkDetector

# 얼굴 랜드마크 검출해서 눈 좌표얻기

landmarks = landmarkDetector(im, faceRects[i])

point2 = (landmarks.part(2).x, landmarks.part(2).y) # right eye

point0 = (landmarks.part(0).x, landmarks.part(0).y) # left eye

eyecornerSrc = [point2, point0]

# Corners of the eye in normalized image

# 새 이미지에 눈을 위치시킬 좌표행렬인듯

eyecornerDst = [(np.int(0.3 \* w), np.int(h/3)),

(np.int(0.7 \* w), np.int(h/3))]

print('eyecornerSrc :', np.array([eyecornerSrc]))

print('eyecornerDst :', eyecornerDst)

# get transform matrix

# 변환 행렬을 생성하는 함수.

# 눈좌표를 원하는 곳으로 옮길 행열을 생성하자(이미지 눈좌표,원하는 곳의 좌표)

ret = cv2.estimateAffinePartial2D(np.array([eyecornerSrc]), np.array([eyecornerDst]))

xform = ret[0]

print('xform :', xform)

# apply transform

# 행열을 이용해서 이미지를 이동,,변환

imOut = cv2.warpAffine(im, xform, (w, h))

plt.figure()

plt.imshow(imOut[:,:,::-1]); plt.title('after image'+np.str(i))

plt.figure()

cv2.rectangle(im, (x1, y1), (x2, y2), (0, 0, 255), 3) #사각형 그리기

plt.imshow(im[:,:,::-1])

plt.title("original")

plt.show()