```python
import cv2
import numpy as np

# Creating face_cascade and eye_cascade objects
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade=cv2.CascadeClassifier('haarcascade_eye.xml')

# Lodaing the image - 얼굴 이미지 데이터 읽기
img = cv2.imread('./face.png')
cv2.imshow('image show',img)
cv2.waitKey(0)

# Converting the image into grayscale 그레이로 바꿔줌
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = face_cascade.detectMultiScale(gray, 1.1, 4)

# Defining and drawing the rectangles around the face
for (x,y,w,h) in faces:
    cv2.rectangle(img, (x,y), (x+w, y+h), (255, 0, 255), 2 )
                                            # 색상     # 선의굵기

roi_gray = gray[y:(y+h), x:(x+w)]
roi_color = img[y:(y+h), x:(x+w)]

# eyes
eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 4)
print(eyes)
'''
2개의 좌표가 잡혀있음
[[41 82 45 45]
 [112 48 52 52]]
'''
index = 0

# Creating for loop in order to divide one eye from another
for (ex, ey, ew, eh) in eyes :
    if index == 0:
        eye_1 = (ex, ey, ew, eh)
    elif index == 1:
        eye_2 = (ex, ey, ew, eh)

    cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0,0,255), 2)
    index = index + 1

# 왼쪽 오른쪽 눈 파악
if eye_1[0] < eye_2[0] :
    left_eye = eye_1
    right_eye = eye_2
else:
    right_eye = eye_1
    left_eye = eye_2

# central points of the rectangles
left_eye_center = (int(left_eye[0] + (left_eye[2]/2)),
                   int(left_eye[1] + (left_eye[3]/2)))

left_eye_center_x = left_eye_center[0]
left_eye_center_y = left_eye_center[1]

right_eye_center =(int(right_eye[0] + (right_eye[2]/2)),
                   int(right_eye[1] + (right_eye[3]/2)))

right_eye_center_x = right_eye_center[0]
right_eye_center_y = right_eye_center[1]

cv2.circle(roi_color, left_eye_center, 5, (255,0,0), -1)
cv2.circle(roi_color, right_eye_center, 5, (255,0,0), -1)
cv2.line(roi_color, right_eye_center, left_eye_center, (0,200,200),1)

if left_eye_center_y > right_eye_center_y :
    A = (right_eye_center_x, left_eye_center_y)
    direction = -1 # 정수 -1은 이미지가 시계방향으로 회전함을 나타냅니다.
else :
    A = (left_eye_center_x, right_eye_center_y)
    direction = 1 # 정수 1은 이미지가 시계 반대방향으로 회전함을 나타냅니다.

cv2.circle(roi_color, A, 5, (255,0,0), -1)
cv2.line(roi_color, right_eye_center, A, (0,200,200),1)
cv2.line(roi_color, A, left_eye_center, (0,200,200),1)

# 각도 구하기
# np.arctan 함수 단위는 라디안 단위
# 라디안 단위 -> 각도 : (theta * 180) / np.pi
delta_x = right_eye_center_x - left_eye_center_x
delta_y = right_eye_center_y - left_eye_center_y
angle = np.arctan(delta_y / delta_x)
angle = (angle*180) /np.pi
print(angle)
# -21.80140948635181 도

h, w = img.shape[:2]
h, w, _ = img.shape

center = (w//2, h//2)
M = cv2.getRotationMatrix2D(center, angle, 1.0)

rotated = cv2.warpAffine(img, M, (w,h))

cv2.imshow('face_drawed', img)
cv2.waitKey(0)

cv2.imshow('face_rotated', rotated)
cv2.waitKey(0)

dist_1 = np.sqrt((delta_x * delta_x) + (delta_y * delta_y))
dist_2 = np.sqrt((delta_x * delta_x) + (delta_y * delta_y))

ratio = dist_1/dist_2
```