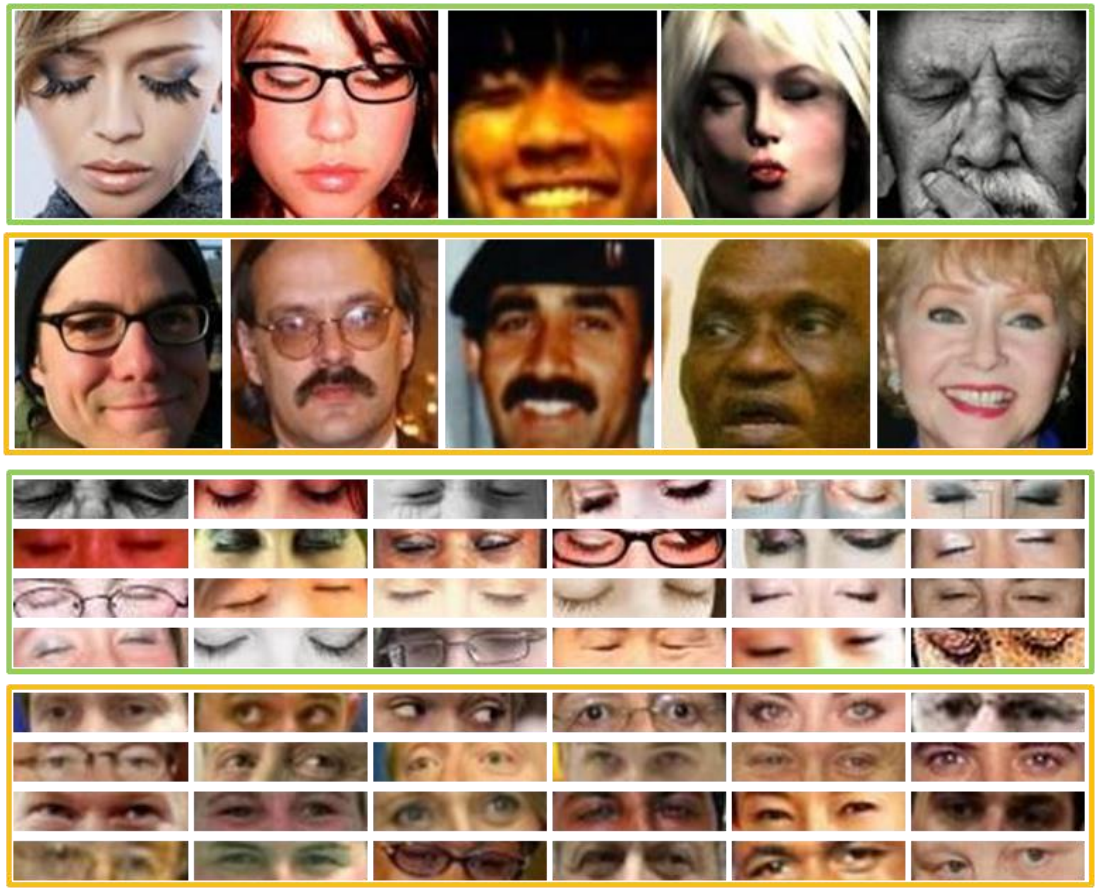
Machine Learning

Application 1

❖ Application 1

- Closeness detection



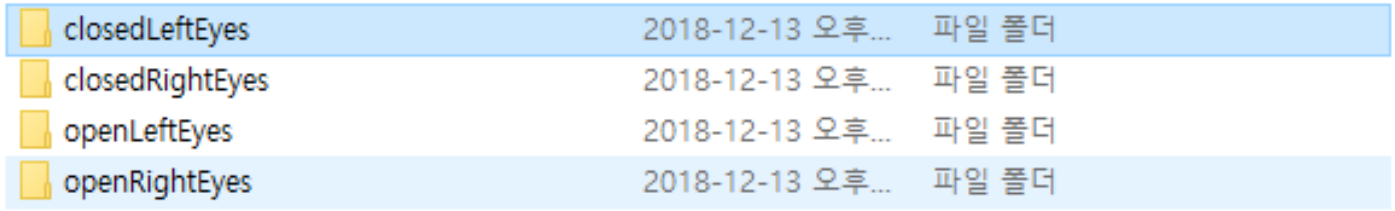
http://parnec.nuaa.edu.cn/xtan/data/ClosedEyeDatabases.html

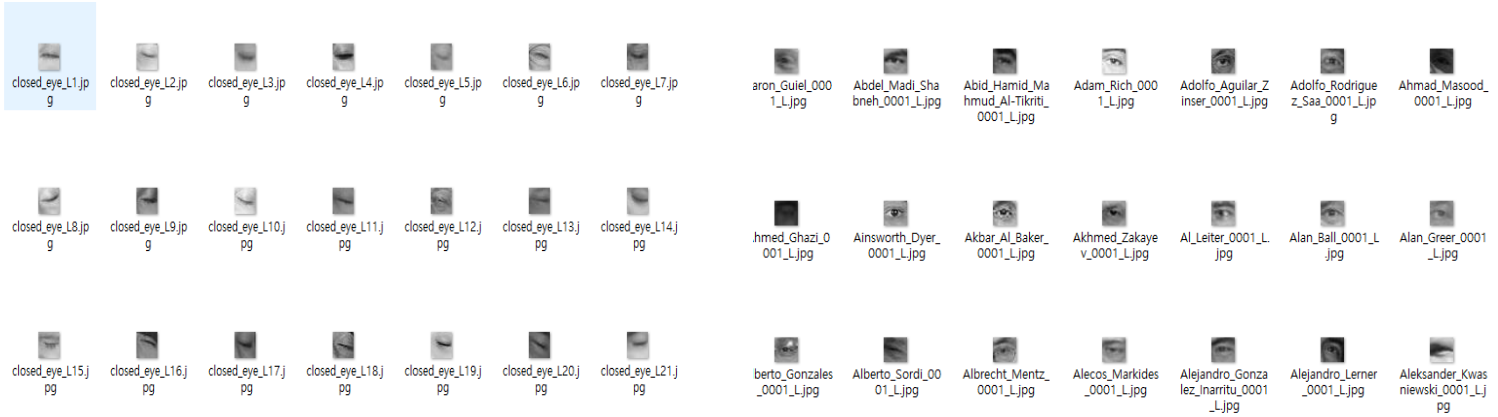
❖ Application 1

- Closed Eyes In The Wild (Nanjing University)

❖ Application 1

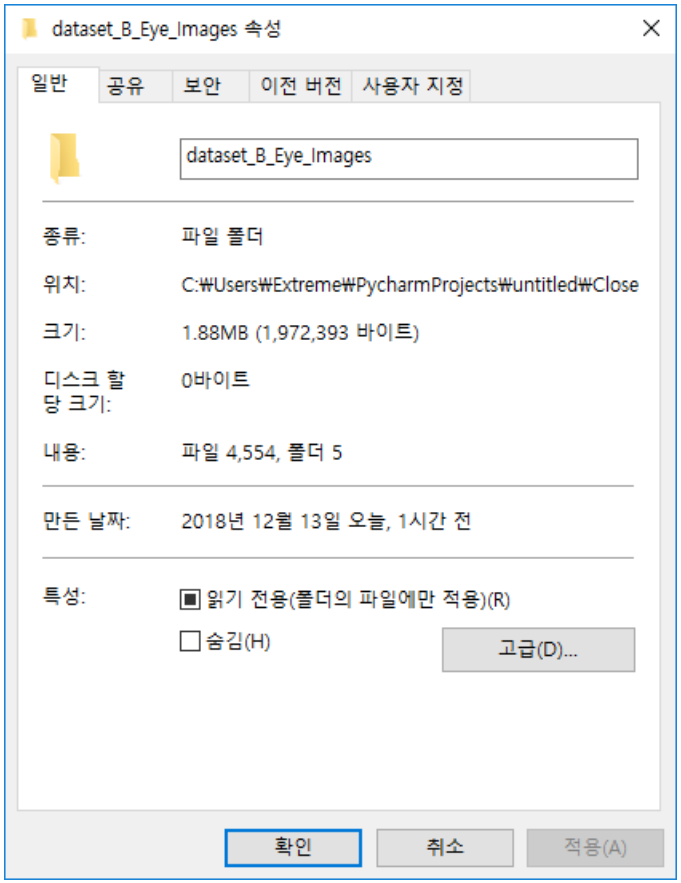
- Closed Eyes In The Wild





❖ Application 1

- Closed Eyes In The Wild



❖ Application 1

- Closed Eyes In The Wild

❖ Application 1

- 데이터 불러오기

```
import cv2
import sys, os
import numpy as np
from numpy import genfromtxt
sys.path.append(os.pardir)

filePath = "Closed Eyes In The Wild (CEW)\\dataset_B_Eye_Images\\dataset_B_Eye_Images"

folderNames = os.listdir(filePath) # 경로 내 파일 및 폴더 명

target = 0

targetData = []
trainData = [])
```

❖ Application 1

- 데이터 불러오기

```
for i in folderNames:
    fullPathName = filePath + "\\" + i
    fileNames = os.listdir(fullPathName)                    # 경로 내 파일 및 폴더
    for j in fileNames:
        srcImage = cv2.imread(fullPathName +  " \\ "  + j, 0)   # 이미지 로드
        srcImage = srcImage.astype(np.float32)              # 데이터 타입 변경
        srcImage /= 255.0                                   # 실수 데이터로 변환
        trainData.append(srcImage)
        targetData.append(target)
    target += 1                                             # 타겟 값 증가

trainData = np.array(trainData).reshape(-1, 1, 24, 24)
targetData = np.array(targetData)

s = np.arange(targetData.shape[0])
np.random.shuffle(s)

trainData = trainData[s] #학습 데이터 셋
targetData = targetData[s] #라벨 데이터 셋
```

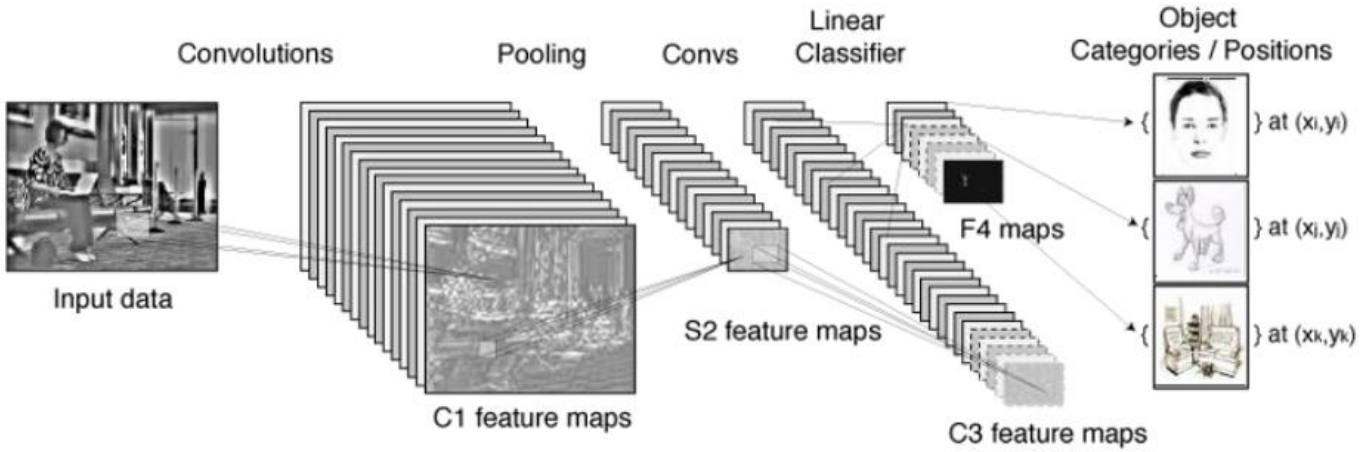
❖ Application 1

- 데이터 불러오기

```
for i in folderNames:
    fullPathName = filePath + "\\" + i
    fileNames = os.listdir(fullPathName)                    # 경로 내 파일 및 폴더
    for j in fileNames:
        srcImage = cv2.imread(fullPathName +  " \\ "  + j, 0)   # 이미지 로드
        data = srcImage.astype(np.float32)                  # 데이터 타입 변경
        data /= 255.0                                       # 실수 데이터로 변환
        trainData.append(srcImage)
        targetData.append(target)
    target += 1                                             # 타겟 값 증가

trainData = np.array(trainData).reshape(-1, 1, 24, 24)
targetData = np.array(targetData)

s = np.arange(targetData.shape[0])
np.random.shuffle(s)

trainData = trainData[s] #학습 데이터 셋
targetData = targetData[s] #라벨 데이터 셋
```

❖ Application 1

- 학습하기 & 학습 결과 파라미터 저장



❖ Application 1

- 테스트

```
import cv2
import sys, os
import numpy as np
sys.path.append(os.pardir)
from numpy import genfromtxt

def resultToStr(result):

    if np.argmax(result) == 0 or np.argmax(result) == 1:
        strTemp = "Closed"
    else:
        strTemp = "Opened"

    return strTemp
```

❖ Application 1

- **테스트(라벨 데이터 불러오기)**

```
filePath = "Closed Eyes In The Wild (CEW)\\dataset_B_Facial_Images\\dataset_B_FacialImages\\"
openedFolderName = "OpenFace\\"
closedFolderName = "ClosedFace\\"
openedEyeLabel = "EyeCoordinatesInfo_OpenFace.txt"
closedEyeLabel = "EyeCoordinatesInfo_ClosedFace.txt"

data = genfromtxt(filePath + openedEyeLabel, delimiter=' ', dtype="U50")

# 신경망 초기화
# 학습 신경망 파라미터 로드
```

❖ Application 1

- 테스트(라벨 데이터 불러오기)

EyeCoordinatesInfo_ClosedFace.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

closed_eye_0001.jpg_face_1.jpg 26 40 73 38closed_eye_0002.jpg_face_2.jpg
6 69 44closed_eye_0059.jpg_face_2.jpg 34 36 72 42closed_eye_0060.jpg_fa
jpg 36 48 66 43closed_eye_0124.jpg_face_2.jpg 31 45 71 38closed_eye_012
_face_1.jpg 37 43 78 43closed_eye_0189.jpg_face_1.jpg 30 43 70 36closed_
0243.jpg_face_1.jpg 32 46 72 39closed_eye_0243.jpg_face_2.jpg 28 32 69 4
sed_eye_0297.jpg_face_1.jpg 34 49 71 48closed_eye_0302.jpg_face_2.jpg 29
5 45closed_eye_0361.jpg_face_3.jpg 40 38 70 47closed_eye_0363.jpg_face_
32 44 70 46closed_eye_0426.jpg_face_1.jpg 31 43 64 38closed_eye_0427.jp
e_1.jpg 33 44 67 38closed_eye_0505.jpg_face_1.jpg 32 42 72 43closed_eye
.jpg_face_1.jpg 33 37 72 38closed_eye_0593.jpg_face_1.jpg 29 43 75 42clos
eye_0646.jpg_face_1.jpg 34 41 72 44closed_eye_0649.jpg_face_1.jpg 33 45
closed_eye_0698.jpg_face_1.jpg 39 43 72 43closed_eye_0708.jpg_face_1.jpg
4 69 45closed_eye_0767.jpg_face_3.jpg 33 40 65 44closed_eye_0769.jpg_fa
jpg 33 40 73 40closed_eye_0820.BMP_face_1.jpg 31 39 69 45closed_eye_08
_face_1.jpg 35 43 69 45closed_eye_0873.jpg_face_1.jpg 38 40 76 43closed_
0918.jpg_face_4.jpg 24 43 62 32closed_eye_0919.jpg_face_2.jpg 37 41 70 4
osed_eye_1015.png_face_1.jpg 34 43 74 44closed_eye_1023.jpg_face_1.jpg
82 44closed_eye_1113.jpg_face_3.jpg 26 28 61 43closed_eye_1114.jpg_face
g 33 49 82 45closed_eye_1181.jpg_face_1.jpg 41 41 67 47closed_eye_1190
ace_1.jpg 32 38 69 42closed_eye_1243.jpg_face_4.jpg 31 46 66 32closed_e
78.jpg_face_3.jpg 36 41 71 42closed_eye_1279.jpg_face_1.jpg 31 45 73 45c
d_eye_1318.jpg_face_2.jpg 32 43 65 37closed_eye_1319.jpg_face_1.jpg 31 4
39closed_eye_1350.jpg_face_2.jpg 27 48 73 32closed_eye_1351.jpg_face_2.j
42 43 72 48closed_eye_1386.jpg_face_2.jpg 33 47 70 41closed_eye_1387.B
e_2.jpg 31 37 71 42closed_eye_1460.jpg_face_1.jpg 28 42 71 38closed_eye
.jpg_face_1.jpg 32 44 73 35closed_eye_1496.jpg_face_1.jpg 34 47 63 38clos
eye_1531.jpg_face_1.jpg 34 45 69 41closed_eye_1533.jpg_face_2.jpg 35 36
closed_eye_1578.jpg_face_1.jpg 35 42 73 47closed_eye_1579.BMP_face_1.jp
3 68 38closed_eye_1640.jpg_face_1.jpg 33 36 81 30closed_eye_1641.jpg_fa
jpg 34 43 73 43closed_eye_1679.BMP_face_1.jpg 33 42 69 43closed_eye_16
_face_1.jpg 31 45 71 39closed_eye_1738.jpg_face_1.jpg 28 42 70 43closed_
1786.jpg_face_1.jpg 34 44 67 44closed_eye_1787.jpg_face_1.jpg 29 43 67 3

data

	0	1	2	3
0	closed_eye_0001.jpg_face_1.jpg	26	40	73
1	closed_eye_0002.jpg_face_2.jpg	36	42	70
2	closed_eye_0003.jpg_face_2.jpg	33	39	73
3	closed_eye_0007.jpg_face_1.jpg	32	39	68
4	closed_eye_0009.jpg_face_1.jpg	30	42	70
5	closed_eye_0012.jpg_face_1.jpg	37	46	69
6	closed_eye_0013.BMP_face_1.jpg	32	47	70
7	closed_eye_0014.jpg_face_2.jpg	41	44	76
8	closed_eye_0015.jpg_face_1.jpg	31	50	69
9	closed_eye_0019.jpg_face_1.jpg	35	46	68
10	closed_eye_0020.jpg_face_1.jpg	39	38	74
11	closed_eye_0021.jpg_face_1.jpg	33	44	72
12	closed_eye_0024.jpg_face_2.jpg	30	42	66
13	closed_eye_0030.jpg_face_2.jpg	31	49	71
14	closed_eye_0032.jpg_face_1.jpg	32	38	69
15	closed_eye_0033.jpg_face_2.jpg	32	39	71
16	closed_eye_0033.jpg_face_3.jpg	33	45	68
17	closed_eye_0034.jpg_face_4.jpg	26	43	67
18	closed_eye_0035.BMP_face_1.jpg	33	41	74
19	closed_eye_0038.jpg_face_1.jpg	33	39	69
20	closed_eye_0043.jpg_face_2.jpg	33	39	69
21	closed_eye_0045.jpg_face_1.jpg	34	44	69

data Format: %s

Colored cells

Resize Automatically

Close

❖ Application 1

- 테스트

```
for i in data:
    srcImage = cv2.imread(filePath + openedFolderName + i[0])
    gray = cv2.cvtColor(srcImage, cv2.COLOR_RGB2GRAY)

    leftCenterX = int(i[1]) - 12
    leftCenterY = int(i[2]) - 12
    rightCenterX = int(i[3]) - 12
    rightCenterY = int(i[4]) - 12
    #눈 영역 잘라내기
    leftEye = gray[leftCenterY: leftCenterY + 24, leftCenterX: leftCenterX + 24]/255
    rightEye = gray[rightCenterY : rightCenterY + 24, rightCenterX: rightCenterX + 24]/255

    leftEye = np.array(leftEye).reshape(-1, 1, 24, 24)
    rightEye = np.array(rightEye).reshape(-1, 1, 24, 24)
    # 신경망 인식
    resultLeft = network.predict(leftEye)
    resultRight = network.predict(rightEye)
```

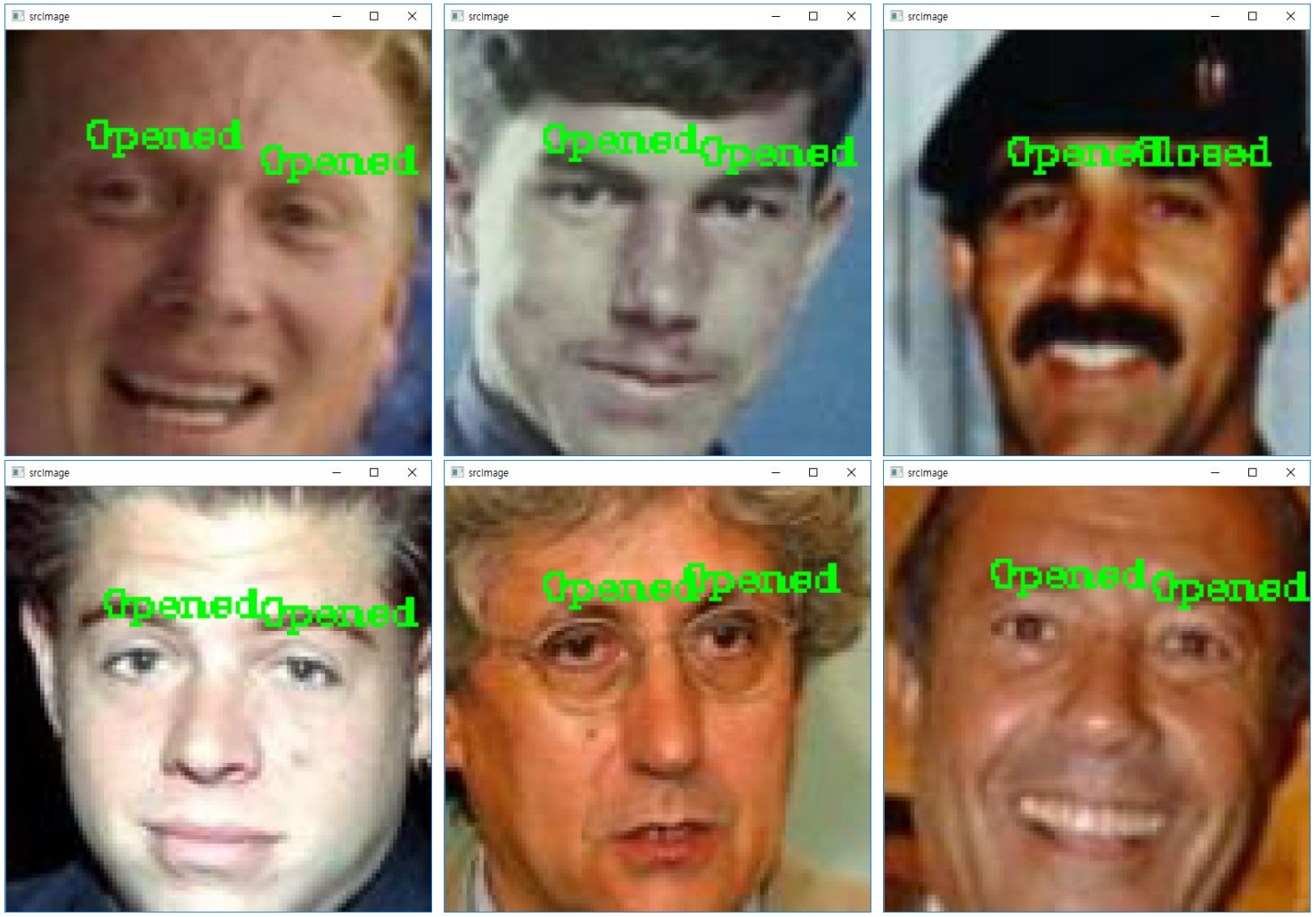
❖ Application 1

- 테스트(for 문 계속)

```
    strLeft = ""
    strRight = ""

    strLeft = resultToStr(resultLeft)
    strRight = resultToStr(resultRight)

    cv2.putText(srcImage, strLeft, (leftCenterX, leftCenterY),
cv2.FONT_HERSHEY_COMPLEX , 0.3, (0, 255, 0))
    cv2.putText(srcImage, strRight, (rightCenterX, rightCenterY),
cv2.FONT_HERSHEY_COMPLEX, 0.3, (0, 255, 0))

    srcImage = cv2.resize(srcImage, dsize=(512, 512), interpolation=cv2.INTER_AREA)
    cv2.imshow("srcImage", srcImage)
    cv2.waitKey(0)
```

❖ Application 1

- 결과



❖ Application 1

- 결과