[머신러닝을 통한 모델(.h5) 파일 생성 코드] - Jupyter notebook

[1]

```
import datetime
import numpy as np
import matplotlib.pyplot as plt
from keras.layers import Input, Activation, Conv2D, Flatten, Dense, MaxPooling2D
from keras.models import Model, load_model
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
plt.style.use('dark_background')
```

[2]

```
x_train = np.load('C:/Users/skt/Desktop/Twinkle/x_train.npy').astype(np.float32)
y_train = np.load('C:/Users/skt/Desktop/Twinkle/y_train.npy').astype(np.float32)
x_val = np.load('C:/Users/skt/Desktop/Twinkle/x_val.npy').astype(np.float32)
y_val = np.load('C:/Users/skt/Desktop/Twinkle/y_val.npy').astype(np.float32)

print(x_train.shape, y_train.shape)
print(x_val.shape, y_val.shape)
```

[3]

```
plt.subplot(2, 1, 1)
plt.title(str(y_train[0]))
plt.imshow(x_train[0].reshape((26, 34)), cmap='gray')
plt.subplot(2, 1, 2)
plt.title(str(y_val[4]))
plt.imshow(x_val[4].reshape((26, 34)), cmap='gray')
```

[4]

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2
)
```

```
val_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow(
    x=x_train, y=y_train,
    batch_size=32,
    shuffle=True
)

val_generator = val_datagen.flow(
    x=x_val, y=y_val,
    batch_size=32,
    shuffle=False
)
```

[5]

```
inputs = Input(shape=(26, 34, 1))

net = Conv2D(32, kernel_size=3, strides=1, padding='same', activation='relu')(inputs)
net = MaxPooling2D(pool_size=2)(net)

net = Conv2D(64, kernel_size=3, strides=1, padding='same', activation='relu')(net)
net = MaxPooling2D(pool_size=2)(net)

net = Conv2D(128, kernel_size=3, strides=1, padding='same', activation='relu')(net)
net = MaxPooling2D(pool_size=2)(net)

net = Flatten()(net)

net = Dense(512)(net)
net = Activation('relu')(net)
net = Dense(1)(net)
outputs = Activation('sigmoid')(net)
```

```
model = Model(inputs=inputs, outputs=outputs)

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])

model.summary()
```

[6]

```
start_time = datetime.datetime.now().strftime('%Y_%m_%d_%H_%M_%S')

model.fit_generator(
    train_generator, epochs=50, validation_data=val_generator,
    callbacks=[
        ModelCheckpoint('C:/Users/skt/Desktop/Twinkle/%s.h5' % (start_time),
monitor='val_acc', save_best_only=True, mode='max', verbose=1),
        ReduceLROnPlateau(monitor='val_acc', factor=0.2, patience=10,
verbose=1, mode='auto', min_lr=1e-05)
    ]
)
print("file name : "+start_time+".h5")
// h5 파일이 생성되었음을 보이기 위해 수정
```

[7]

```
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns

model = load_model('C:/Users/skt/Desktop/Twinkle/%s.h5' % (start_time))

y_pred = model.predict(x_val/255.)
y_pred_logical = (y_pred > 0.5).astype(np.int)

print ('test acc: %s' % accuracy_score(y_val, y_pred_logical))
cm = confusion_matrix(y_val, y_pred_logical)
sns.heatmap(cm, annot=True)
```

[8]

```
ax = sns.distplot(y_pred, kde=False)
```

[소스 코드]

```
import tensorflow.compat.v1 as tf
tf.disable_resource_variables()
tf.disable_v2_behavior()
import cv2, dlib
import numpy as np
from imutils import face_utils
from keras.models import load_model
import datetime
import time
import RPi.GPIO as GPIO

IMG_SIZE = (34, 26)

detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
model = load_model('models/2020_12_07_19_04_40.h5')
model.summary()
GPIO.setmode(GPIO.BCM)              #led 사용을 위한 코드를 추가
GPIO.setwarnings(False)
GPIO.setup(18, GPIO.OUT)

pwm = GPIO.PWM(18, 50)  # PWM 18번 핀에 50Hz의 주파수 설정
pwm.start(100)          # PWM의 값을 0~100까지 넣을 수 있다. (100의
dutycicle 설정)       - 100이 가장 밝고 0이 가장 어둡다.
dc = 100

fs = None                   # 시간의 흐름을 알기 위해 추가하였다.
fm = None
count = 0
end loop = 0          # 자동 종료하기 위해 추가하였다.

def crop_eye(img, eye_points):
    x1, y1 = np.amin(eye_points, axis=0)
    x2, y2 = np.amax(eye_points, axis=0)
    cx, cy = (x1 + x2) / 2, (y1 + y2) / 2
```

```
w = (x2 - x1) * 1.2
h = w * IMG_SIZE[1] / IMG_SIZE[0]

margin_x, margin_y = w / 2, h / 2

min_x, min_y = int(cx - margin_x), int(cy - margin_y)

max_x, max_y = int(cx + margin_x), int(cy + margin_y)

eye_rect = np.rint([min_x, min_y, max_x, max_y]).astype(np.int)

eye_img = gray[eye_rect[1]:eye_rect[3], eye_rect[0]:eye_rect[2]]

return eye_img, eye_rect

# main
#cap = cv2.VideoCapture('videos/2.mp4')
  cap = cv2.VideoCapture(0)          # video 대신 Pi Camera를 실행
#cap.set(3,640) # set Width
#cap.set(4,480) # set Height

while cap.isOpened():
    ret, img_ori = cap.read()

if not ret:
    break

img_ori = cv2.resize(img_ori, dsize=(0, 0), fx=0.5, fy=0.5)

img = img_ori.copy()
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = detector(gray)

for face in faces:
      shapes = predictor(gray, face)
```

```
        shapes = face_utils.shape_to_np(shapes)

        eye_img_l, eye_rect_l = crop_eye(gray, eye_points=shapes[36:42])
        eye_img_r, eye_rect_r = crop_eye(gray, eye_points=shapes[42:48])

        eye_img_l = cv2.resize(eye_img_l, dsize=IMG_SIZE)
        eye_img_r = cv2.resize(eye_img_r, dsize=IMG_SIZE)
        eye_img_r = cv2.flip(eye_img_r, flipCode=1)

# cv2.imshow('l', eye_img_l)
# cv2.imshow('r', eye_img_r)

eye_input_l = eye_img_l.copy().reshape((1, IMG_SIZE[1], IMG_SIZE[0],
1)).astype(np.float32) / 255.
eye_input_r = eye_img_r.copy().reshape((1, IMG_SIZE[1], IMG_SIZE[0],
1)).astype(np.float32) / 255.

pred_l = model.predict(eye_input_l)
pred_r = model.predict(eye_input_r)

# visualize
state_l= 'O %.1f' if pred_l > 0.1 else '- %.1f'
#출력되는 동영상에서 눈 부분의 네모 창 위에 떠있는 숫자이다. pred_l, pred_r이
0.1보다 작은 경우 -가 앞에 붙는다. 따라서 pred_l이 0.1보다 작으면 눈을 감았다
고 예측한다.
state_r = 'O %.1f' if pred_r > 0.1 else '- %.1f'

state_l = state_l % pred_l
state_r = state_r % pred_r

cv2.rectangle(img, pt1=tuple(eye_rect_l[0:2]), pt2=tuple(eye_rect_l[2:4]),
color=(255,255,255), thickness=2)
cv2.rectangle(img, pt1=tuple(eye_rect_r[0:2]), pt2=tuple(eye_rect_r[2:4]),
 color=(255,255,255), thickness=2)

cv2.putText(img, state_l, tuple(eye_rect_l[0:2]), cv2.FONT_HERSHEY_SIMPLEX,
0.7,
```

```
(255,255,255), 2)
cv2.putText(img, state_r, tuple(eye_rect_r[0:2]), cv2.FONT_HERSHEY_SIMPLEX,
0.7,
(255,255,255), 2)

#눈을 감았을때의 초(시간)를 세는 코드
if pred_l < 0.1 and pred_r < 0.1 :
    if fs is None :
#양쪽 눈을 모두 감았을 때 fs값이 None이면 이 코드를 실행
            fnow = datetime.datetime.now()
#datetime클래스를 사용해서 오늘의 날짜와 시간 값을 fnow에 저장
            fs = int(fnow.strftime('%S'))  // second
            fm = int(fnow.strftime('%M'))  // minute

   else :
            snow = datetime.datetime.now()
            ss = int(snow.strftime('%S'))
            sm = int(snow.strftime('%M'))

            if ss == 0 or ss == 60 : #입력 받은 초가 0 또는 60초면

count = count + 1        #count 변수에 1을 더한다.

sscount = ss + (count*60)
#0또는 60초가 반복된 횟수 count 변수에 60을 곱한 값을 더한 것

            else :          #그 이외의 경우
    sscount = ss      #숫자 그대로 사용한다.

#10초 후에 LED가 꺼짐
if (sscount - fs) < 10 :          #지나간 초가 10초보다 작을 경우
      pwm.ChangeDutyCycle(dc)  # dc = 100~0까지의 값을 설정할 수 있고
100에서 1초당 10씩 10번 감소하면 0이다.
      dc = 100 -(sscount - fs)*10
#10초 동안 led가 서서히 꺼진다.

else :
```

```
        pwm.stop()
        print("LED OFF")
        endloop = 1
        break

#10분 후에 LED가 꺼짐
#if sm == 0 or sm == 60 :
        # count = count + 1
        # smcount = sm + (count*60)

#else :
        # smcount = sm

#if (smcount - fm) < 10 :     #10분
      # pwm.ChangeDutyCycle(dc)

#if (sscount-fs)%10 == 0 :         #led 밝기가 약 10초에 1씩 감소
      # dc = dc - 1

#else :          #10분 이상인 경우
      # pwm.stop()        #led를 끈다.
      # print("LED OFF")
      # endloop = 1
      # break
print("closeing eye time :" + str(sscount-fs))

else :
      #pred_l과 pred_r이 0.1보다 클 때 실행. 눈을 감았다는 것을 인지하기 전의 초기 설정으로 리셋 시킬 필요가 있다.
        fs = None
        fm=None
        dc = 100
        pwm.ChangeDutyCycle(dc)
        print("Awke")       #눈을 떴다는 것을 터미널 창에 알려준다.

cv2.imshow('result', img)
```

```
if endloop == 1 :
    break                          # 프로그램을 자동으로 종료

if cv2.waitKey(1) == ord('q'):
        break

cap.release()
```