**AI-MIDTERM**

**REPORT**

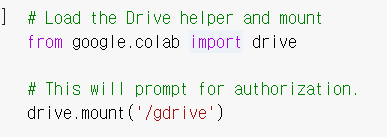
인하대학교

컴퓨터공학과

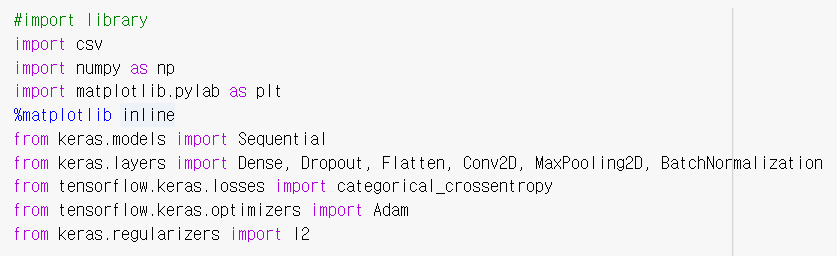
12171708

조승효

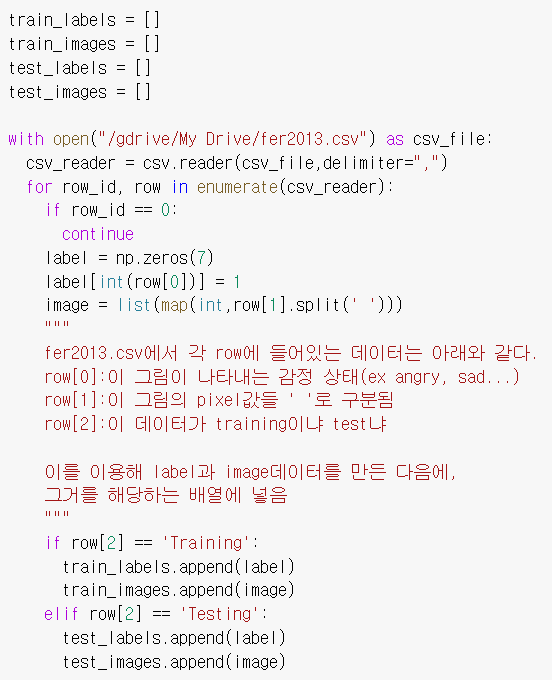
**1.Implementation, training and testing results**



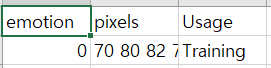
구글 colab에서 실행하였습니다. 그래서 구글드라이브에 있는 데이터를 사용하기 위해 drive를 mount하였습니다.



구현에 필요한 라이브러리를 import 하였습니다.



csv파일을 읽어서 train, test dataset을 만드는 부분입니다.



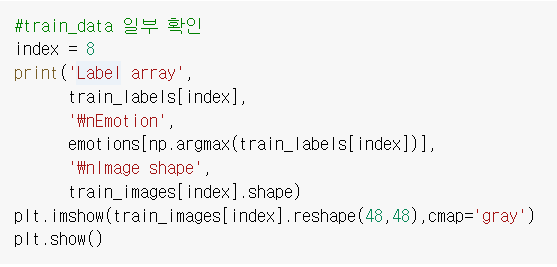
Fer2013중 일부입니다. ‘emotion’ 컬럼은 감정 상태를 나타냅니다. 0~6까지의 값을 가지며 이 숫자들은 특정 감정에 mapping됩니다. ‘pixels’컬럼은 그림에서 pixel값을 나타냅니다. 48\*48의 고정된 크기를 가지며 흑백으로만 표현됩니다. 0~255의 값을 가지며 ‘ ‘로 구분됩니다. ‘Usage’컬럼은 이 데이터가 training에 쓰이는지 test에 쓰이는지를 나타내는 부분입니다.



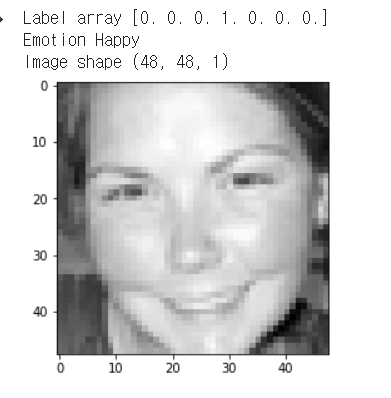
데이터를 가공하는 부분입니다. 데이터 크기를 줄이기 위해 float32로 형변환한 부분을 확인 할 수 있습니다. 그리고 train\_images,test\_images는 0~255의 값을 가지고 있을 것인데, training의 정확도를 올리기 위해서는 이 값들을 0~1로 변환시키는 것이 좋습니다. 그래서 train\_images,test\_images를 255로 나누었습니다.



감정을 mapping하는 부분입니다.

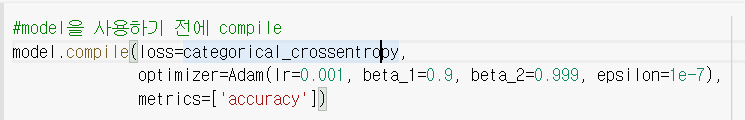


data의 일부를 출력하는 부분입니다. 아래와 같이 정상 출력 됨을 확인할 수 있습니다.

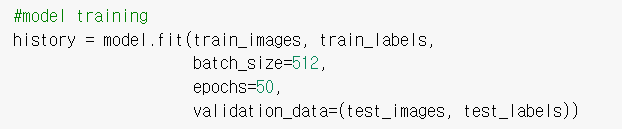




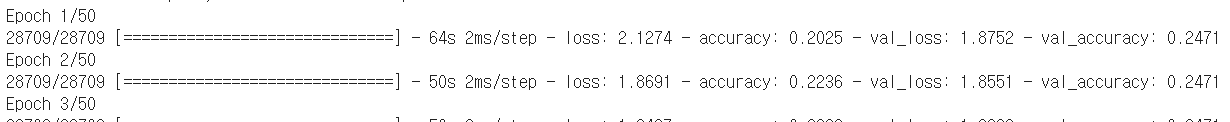
Model을 만드는 부분입니다. 이미지에서 특정 pattern을 사용해야 하므로 convolutional network를 이용해야 합니다. 사람의 감정을 구분하기 위해서는 사람의 눈꼬리가 올라갔는지, 입꼬리는 그대로 인지. 코는 찡그렸는지 등을 알아야 합니다. 그걸 알기 위해서는 이미지의 특정 부분을 취해서 그것의 상태가 어떤지 알아내는 과정이 필요합니다. 그래서 convolutional neural network가 여기에서 적합하게 됩니다. Maxpooling을 이용해 최대값을 뽑아내게 됩니다. 그리고 dropout과 batchnormalization layer도 있음을 확인할 수 있는데 이 것이 없다면 학습이 정체될 수도 있기 때문입니다. Dense layer를 이용해 이미지의 특성들을 감정들에 mapping시킵니다. Softmax를 써서 이미지가 무슨 감정을 표현하는지 알아냅니다.



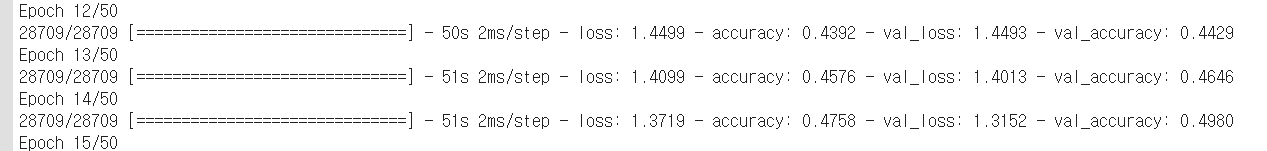
Model을 compile하는 부분입니다.



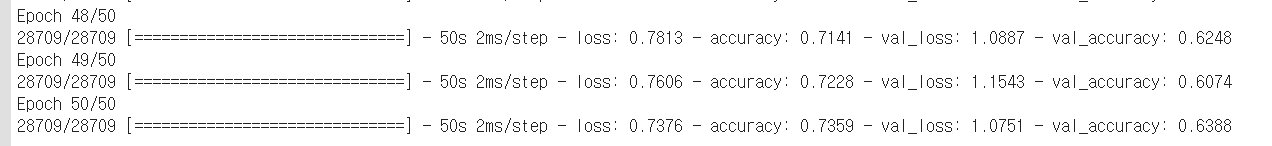
Model을 training하는 부분입니다.



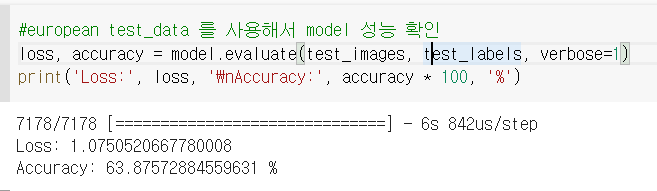
처음에는 이와 같이 정확도가 그리 높지 않음을 확인 할 수 있습니다.



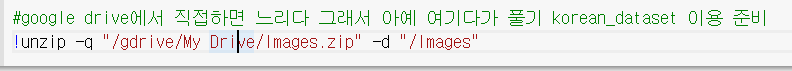
그러다가 정확도가 점점 높아지게 되고



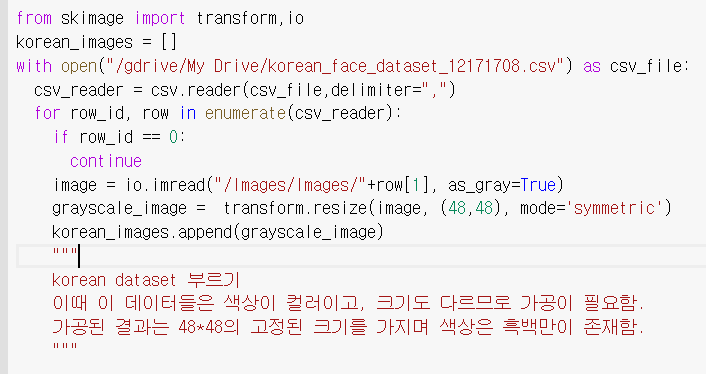
Training set의 정확도가 73%까지 올라감을 확인할 수 있습니다.



그리고 이 모델을 test해보면 test의 결과는 63%가 됨을 확인할 수 있습니다.



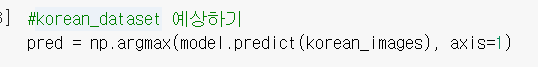
한국 사람들의 이미지 파일을 준비하는 부분입니다.



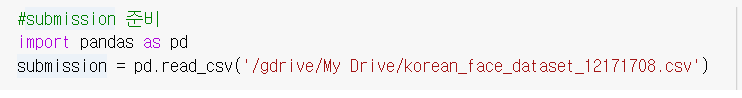
한국 사람들의 이미지를 불러오는 부분입니다. 이때 한국 사람들 이미지는 컬러이고 크기가 제각각 이므로 컬러와 크기를 바꾸는 작업이 필요합니다.



Data를 재가공하는 data를 확인하는 부분입니다. 이때 Korean\_images에는 0~1의 값이 있을 것이므로 별도의 나누기 작업은 필요하지 않습니다.



Model을 이용해 예상하는 부분입니다.



예상 결과를 파일에 쓰기위해 준비하는 부분입니다.

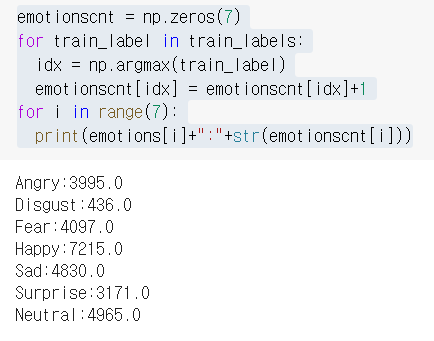


예상값을 넣는 부분입니다.

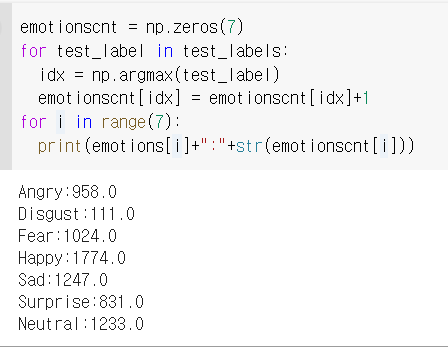


예상 값을 쓰는 부분입니다. 파일 이름은 label\_korean.csv입니다.

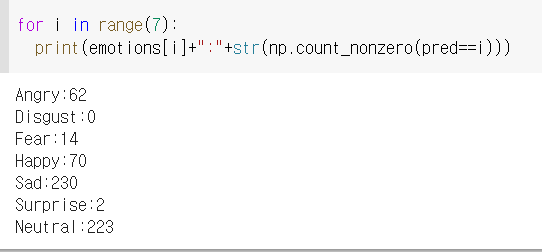
**2. my labeling**



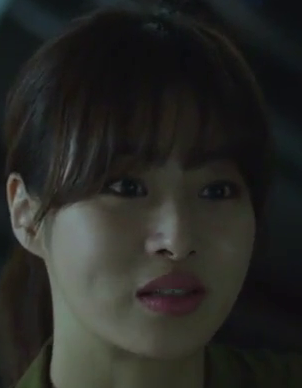
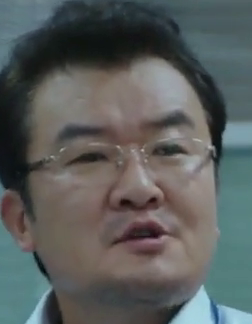
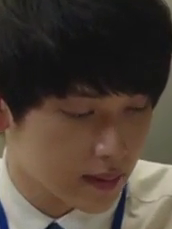
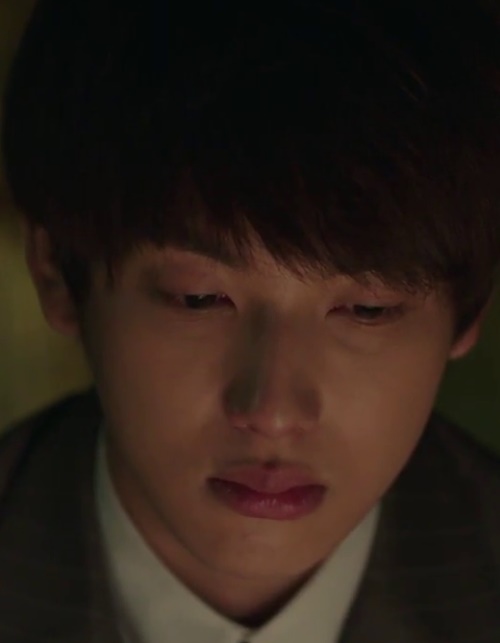
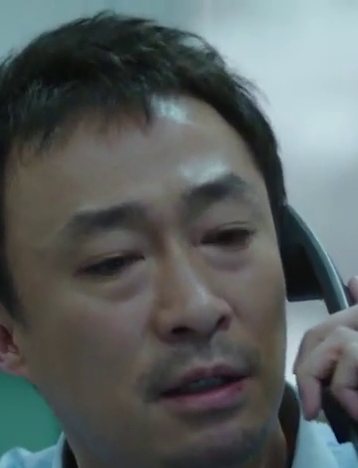
FER 2013에서 Training data를 분석한 부분입니다. Happy가 가장 많고, disgust가 상대적으로 적음을 확인할 수 있습니다.



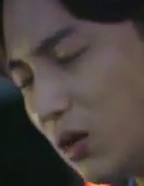
FER 2013에서 Test data를 분석한 부분입니다. 역시 Happy가 가장 많고 disgust가 가장 적습니다. 위와 같은 결과로 보았을 때 Happy, Sad와 같은 것들은 쉽게 구했지만 Disgust와 같은 표정은 다소 어렵게 구했음을 알 수 있습니다. 이는 Disgust한 표정이 있을 때 잘 분류하지 못할 수도 있음을 의미합니다.



Koeran dataset에서 감정에 따른 개수입니다. 제일 많은 것이 sad와 neutral 인 것을 확인할 수 있습니다. 그리고 Happy와 Angry가 그 뒤를 따름을 확인할 수 있습니다. Fear은 조금 있으며 Disgust, Surprise는 거의 분류가 되지 않았습니다.

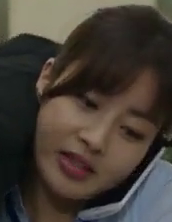
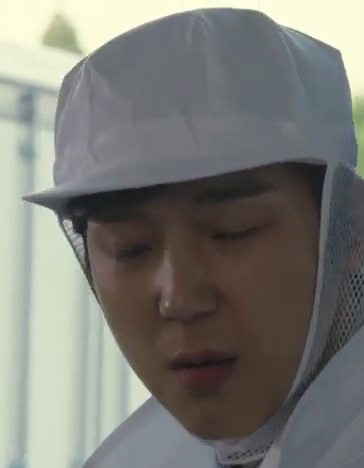


인공지능이 angry(0)으로 분류한 사진들. 사진들을 보면 화난 표정이라고 보기 어려운 게 많음

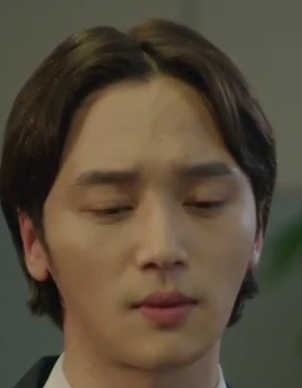
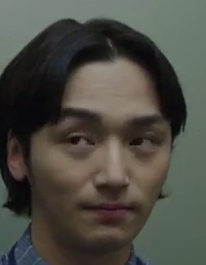
. 



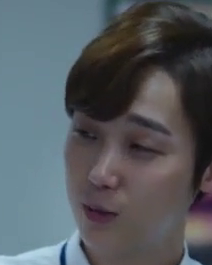
인공지능이 fear(2)으로 분류한 사진들. 이것들은 공포스러운 표정이라고 보다는 차라리 무표정 내지는 슬픔에 가까워 보임



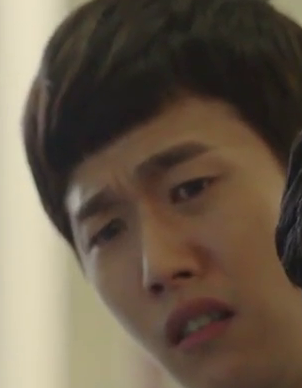
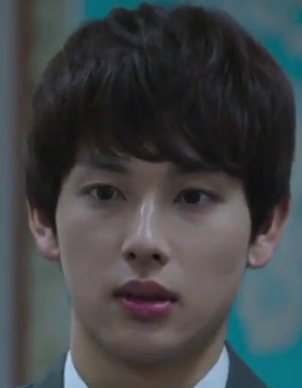
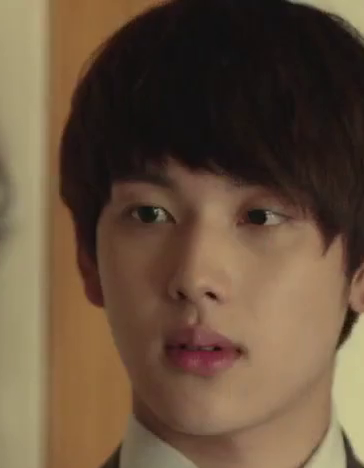
인공지능이 happy(3)으로 분류한 사진들. 행복한 표정이라 할 수 있는 입꼬리가 올라간 것을 잘 캡쳐해낸 것 같음.



인공지능이 sad(4)으로 분류한 사진들. 눈꼬리가 내려간 것을 어느정도는 캡쳐한거 같음.



인공지능이 surprise(5)으로 분류한 사진들. 놀란 표정으로 구분한 것이 거의 없음



인공지능이 neutral(6)로 분류한 사진들. 대개는 분리가 잘 되어 있는 것으로 보이나. Sad(위에 있는 것들 중에서 두 번째) Disgust(아래에 있는 것들 중에서 첫번째)로 분류될 여지가 있는 사진들도 있음.

**3.** **discussions and conclusion**

위와 같은 결과로 보았을 때 Angry, Disgust, Fear는 구분을 제대로 하지 못한 것 같고, Happy, Sad, Neutral은 나름 잘 구분한 거 같습니다. 많은 수의 데이터가 neutral로 분류되었는데 그 이유는 training data로 사용한 것 중에 neutral데이터가 많아서 이기 때문입니다. 그리고 Korean dataset 자체가 얼굴을 찡그린다거나, 입꼬리가 올라간게 거의 없기 때문이기도 합니다. 그러나 사실 감정이라는 것이 주관적인 부분이고 사람에 따라서 감정을 평가하는 것이 다르기 때문에, 이 모델의 성능을 분석하기는 어려울 꺼 같습니다. 더욱이 FER2013에서도 확인할 수 있듯이 감정의 개수가 고르지 못해서 어느 한 쪽으로만 쏠려 판단할 여지도 있습니다. 그래서 이와 같은 모델을 더욱 정확하게 하기 위해서는 데이터의 개수에 대해서 균형이 있어야 하며 감정을 구분할 때에 나름의 객관적인 기준이 필요할 꺼 같습니다. 그리고 대화 맥락, 주변 환경에 따라 같은 얼굴 모양을 가지고 있어도 사람은 충분히 다른 감정을 느낄 수도 있습니다. 그래서 감정을 판별하기 위해서는 얼굴 표정 뿐만 아니라, 대화한 내용, 주변 환경 등에 대한 데이터도 필요로 할 꺼 같습니다. 이와 같은 내용을 종합할 때, 인공지능도 틀릴 수 있음을 알 수 있으며, 결국 주어지는 데이터에 따라 성능이 상당히 좌우됨을 알 수 있습니다.