CNN기법을 이용한 점자문자 알파벳 분류

백대환[1], 데이터 수집, 프로그래밍, PPT, 보고서

요약

본 논문에서 점자문자 이미지의 분류를 위하여 CNN 기반 모델을 사용하였다. 사용된 데이터는 [1]Kaggle에 점자문자 이미지 데이터 세트를 이용하였다. 정확도를 높이기 위해 데이터 삭제 및 이미지 증강 전처리 과정을 진행하였고, 최적의 하이퍼파라미터를 찾기 위해 15개 정도의 모델들을 비교해 보았다. 최적의 모델을 여러 번 시행하였을 때 정확도 100% 정도가 나온 것을 확인할 수 있었다. 이러한 학습된 분류 모델을 상용화하여 시작장애인을 위해 여러 방편으로 활용되길 기대한다.

주요용어 : CNN, epoch, Dropout, BatchNormalization

1. 서론

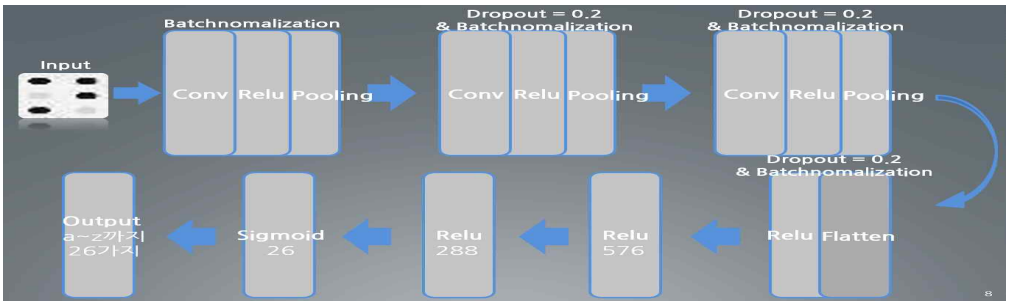
점자문자는 일반인들 눈에는 잘 보이지 않지만 다양한 시설물들에 존재하고 있다. 하지만 실제로 야외에서 시각장애인분들은 지팡이를 이용하여 바닥에 있는 점자블록들만 확인하므로 모든 점자문자를 확인하는 것은 쉽지 않다. 실제로 한 [2]뉴스 기사에서 시각장애인이 말하길 화장실을 이용할 때 법으로 정해진 위치에 점자문자가 있지 않으면 온 벽을 다 만져봐야 하는 어려움이 있어 화장실 이용에 불편함이 있다고 한다. 본 연구는 위와 같은 불편함을 줄이기 위해 점자문자 이미지 분류 모델을 연구하고 최고 성능을 보이는 모델을 탐색하였으며, 이를 활용해 점자문자 이미지를 a-z까지의 알파벳으로 분류하는 실험을 진행하였다. 이러한 분류 모델을 다양한 발명품에 활용하여 시각장애인분들이 공공시설물을 편리하게 이용할 수 있을 것이다.

2. 데이터

Kaggle 사이트로부터 점자문자 이미지 총 1560건을 사용하였다. 이 데이터세트는 알파벳 a-z까지 각 60개씩 이루어져 있다. 모델 학습에 앞서 정확도 향상을 위해 이미지 데이터 전처리 작업을 하였다. 먼저 학습에 방해되는 기울어진 데이터 480개를 제거하여 총 데이터 개수 1040개가 남아 CNN을 진행하기에는 데이터 수가 부족하였다. 그래서 각각의 이미지를 imgaug, PIL 패키지를 이용해 대비, 색상화, 흐림 정도를 변경하여 이미지 데이터 10400개를 추가하였고, 총 11440개의 데이터로 만들었다. cv2 패키지를 이용하여 28*28*3(RGB) 형태로 읽어 온 후 정규화를 하였다. 모델 학습에 앞서 과적합을 방지하기 위해 전처리 된 데이터를 Train/Validation/Test를 72.25%/12.75%/15%로 나누었다.

3. 본론

3.1 모델구성

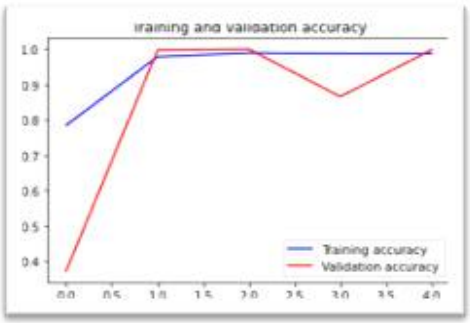


[그림1] 모델구성

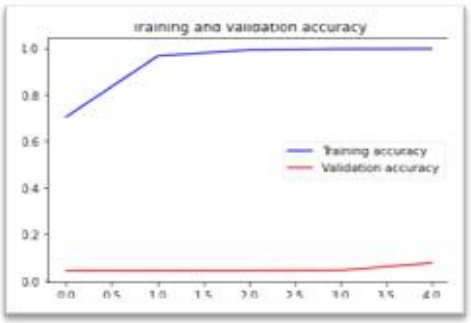
점자문자 이미지 분석을 위한 CNN모델 구성으로는 총 7개의 층으로 이루어져 있다. 1~3층에는 합성곱 레이어(Convolution Layer)와 풀링 레이어(Pooling Layer)의 쌍으로 이루어진 과정과 각 레이어마다 변형된 분포가 나오지 않도록 조절해주는 배치정규화(BatchNomalization) 과정이 반복된다. 합성곱, 풀링계층은 연산을 사용할 때 가중치들은 필터의 형태로 표현된다. 합성곱 레이어의 필터(Filters)수는 64, 커널(Kernel)사이즈는 5X5(1층), 3X3(2,3층), 활성화 함수는 Relu함수를 이용했고, 풀링(Pooling)사이즈는 2X2를 이용했다. 3층 마지막 풀링 레이어에서 나온 결과를 Dense함수를 이용해 출력 수를 축소하기 위하여 Flatten 함수를 사용하여 1차원 벡터로 나열한다. 4~6층까지 1차원 출력 576개에서 Relu함수와 Dense를 이용해 출력 수를 줄여나갔다. 최종적으로는 알파벳 a-z까지 26가지로 분류하기 위해 7층에서 Sigmoid함수를 사용하여 모델을 완성하였다.

3.2 하이퍼파라미터 최적화

(1) 에폭 = 5로 고정하고 배치사이즈 32, 64, 128로 설정하여 새 모델을 돌린 결과 배치사이즈 128일 때는 에폭 수가 충분하지 않아 진행이 되지 않았고, 배치사이즈 32일 때와 64일 때는 안정적으로 학습이 되지 않았다. 모델을 조금 더 안정적으로 학습하기 위해 에폭 수를 조금 더 늘려보았다.

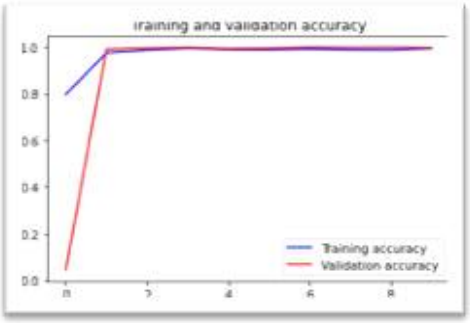


[그림2] 에폭=5, 배치사이즈=32

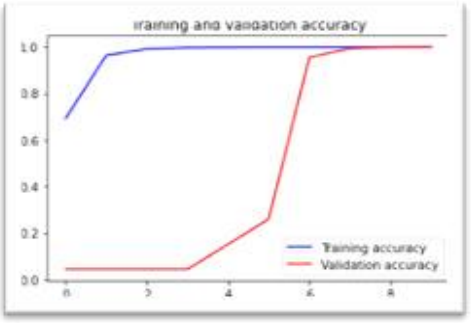


[그림3] 에폭=5, 배치사이즈=128

(2) 에폭 = 10으로 고정하고 배치사이즈 32, 64, 128로 설정하여 새 모델을 돌린 결과 결과수치와 그래프가 안정적으로 학습되었다. 배치사이즈가 증가할수록 학습 소요시간이 줄어듦을 알 수 있다. 배치사이즈 256으로 한다면 소요시간을 더 줄일 수 있을 것이라 생각이 들어 진행해보았다.

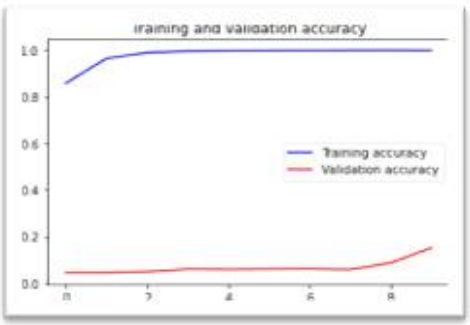


[그림4] 에폭=10, 배치사이즈=32,
소요시간=2분40초

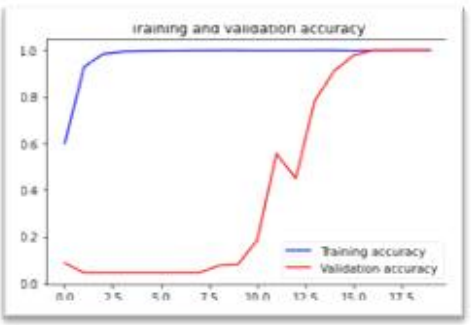


[그림5] 에폭=10, 배치사이즈=128,
소요시간=2분20초

(3) 배치사이즈 = 256으로 고정하고 에폭 10, 15, 20으로 설정하여 새 모델을 돌린 결과 에폭 20일 때 수치가 좋게 나왔지만 소요시간이 4분 40초로 나와 선택 할 이유가 없었다. 소요시간이 가장 짧게 나온 에폭10, 배치사이즈 128을 하이퍼파라미터로 지정하였다.

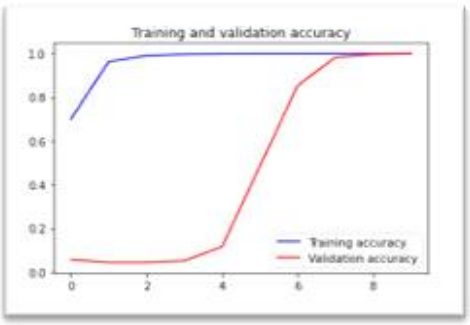


[그림6] 에폭=10, 배치사이즈=256,
소요시간=2분13초

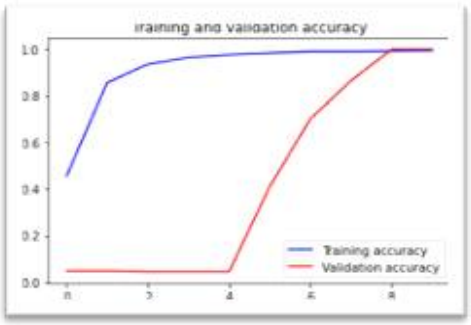


[그림7] 에폭=20, 배치사이즈=256,
소요시간=4분40초

(4) Dropout 파라미터의 최적값을 찾아보았다. 에폭 = 10, 배치사이즈 = 128로 고정하고, Dropout = 0.2, 0.4, 0.6으로 설정하고 모델을 돌린 결과 드롭아웃이 증가하면 학습이 더 늦게 진행이 된다는 것을 그래프를 보면 알 수 있다. 실제 소요시간도 더 오래걸림을 알 수 있다. 따라서 Dropout값은 0.2를 하이퍼파라미터로 지정하였다.

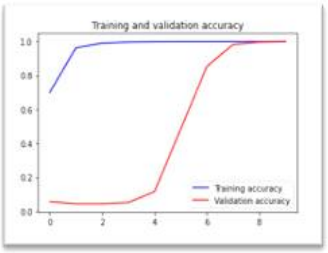


[그림8] Dropout=0.2, 소요시간=2분20초

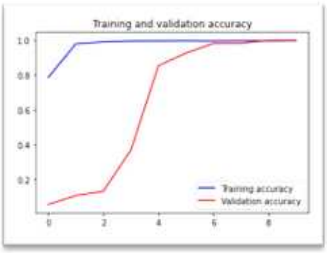


[그림9] Dropout=0.6, 소요시간=2분25초

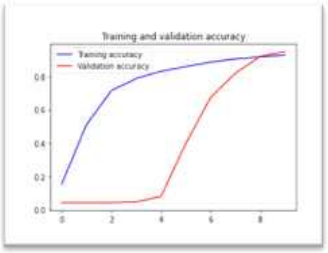
(5) 최적의 Optimizer 설정을 위해 모델들을 비교해보았다. 앞서 찾았던 파라미터인 에폭 = 10, 배치사이즈 = 128, Dropout = 0.2로 고정하고 Optimizer는 ADAM, RMS, SGB로 설정하고 세모델들을 돌린 결과 SGB는 두 모델에 비해 성능이 떨어졌고, RMS는 ADAM에 비해 학습 소요시간이 10초 더 걸렸다. 따라서 ADAM모델을 선택하였다.



[그림10] ADAM, 소요시간=2분20초



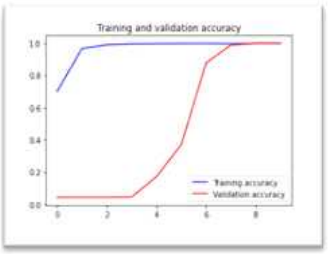
[그림11] RMS, 소요시간=2분30초



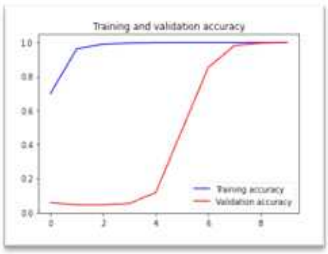
[그림12] SGB. 소요시간=2분30초

3.3 최적화 된 모델 반복 시행

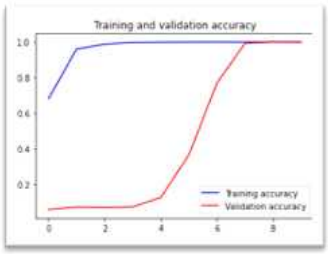
안정적인 모델들 중 학습 소요시간이 가장 적게 걸리는 최적의 하이퍼파라미터는 에폭은 10이고, 배치사이즈는 128, Dropout은 0.2이고, Optimizer는 ADAM일 때였다. 모델을 시행할 때마다 데이터 셋이 달라지므로 여러 번 시행하여 값이 일정하게 나오는지 확인해 보았다. 그래프를 보면 모두 안정적으로 학습이 잘 진행됨을 알 수 있다.



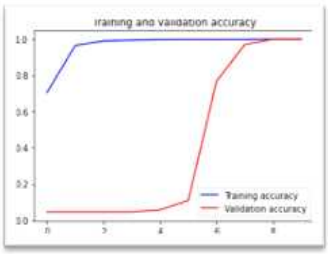
[그림13]



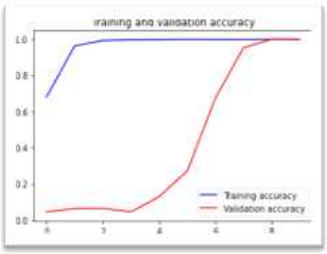
[그림14]



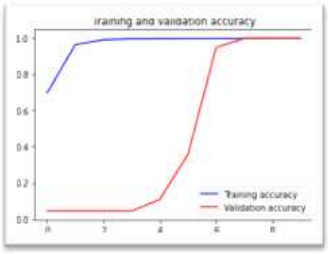
[그림15]



[그림18]



[그림16]



[그림17]

4. 결론

본 논문에서는 점자문자 이미지들을 분류하는 CNN모델에 대하여 연구하였다. 더 높은 정확도를 얻기 위하여 데이터 전처리와, 최적의 하이퍼파라미터를 찾는 과정을 거쳤다. 점자문자 이미지를 약 정확도 100%, 알파벳 a-z까지 분류를 성공적으로 마쳤다.

이러한 점자이미지 데이터 말고도 한글 점자이미지와 숫자 점자이미지 등의 데이터를 얻어 위와 같은 모델로 학습을 진행한다면 적용이 가능한 것들이 많을 것이다. 향후 연구에서 한 개의 점자이미지를 하나의 알파벳으로 분류하는 것을 넘어서 점자문장, 점자책 등을 입력값으로 받았을 때, 실시간으로 문장이 번역이 되는 딥러닝 알고리즘을 연구하여, 시각장애인분들의 생활이 조금 더 나아지길 희망한다.

하지만 비슷한 패턴의 이미지로만 학습을 진행했기 때문에 다른 패턴의 점자문자를 입력하면 예측 능력이 떨어진다. 학습데이터에 대한 문제점을 파악하고, 이를 해결하기 위한 방법을 탐구할 것이다.

참고문헌

[1]https://www.kaggle.com/shanks0465/braille-character-dataset
[2]https://www.fnnews.com/news/202206061801441085