

CNN을 이용한 거북목 증후군 진단기의 구현

손동협^o, 정유철^{*}

^o금오공과대학교 컴퓨터공학과,

^{*}금오공과대학교 컴퓨터공학과

e-mail: sondonghup@naver.com^o, jyc@kumoh.ac.kr^{*}

Implementation of Turtle Neck Syndrome Diagnosis using CNN

Dong-Hyeop Son^o, Yuchul Jung^{*}

^oDept. of Computer Engineering, Kumoh National Institute of Technology,

^{*}Dept. of Computer Engineering, Kumoh National Institute of Technology

● 요약 ●

최근 스마트폰과 컴퓨터 등의 비중이 커지면서 거북목 증후군의 관심사가 커졌다. 거북목 증후군은 잘못된 자세로 인해 어깨의 근육과 인대가 늘어나 통증이 생기는 증상을 의미한다. 이러한 잘못된 자세에는 대표적으로 일자목과 역자목이 있으며 일자목은 7개의 목뼈로 이루어진 경추라인이 c자 라인에서 일직선으로 뼈의 형태가 바뀌어 디스크가 일어나 통증을 유발하는 증상이고 역자목은 정상의 목뼈 구조를 잃어버린 형태로 곧 디스크를 보이며 고개를 드는 것이 힘들며 구부정한 자세를 취하게 되는 증상이다. 본 연구에서는 컨볼루션 신경망 (CNN) 학습 모델을 구현하여 주어진 자세가 올바른 자세인지 일자목인지 c자목인지를 진단할 수 있는 분류기를 구현하였다. 또한, 최근 코로나 사태로 인해 마스크 장착이 일상화되고 있는데, 추가 데이터를 보장하여, 마스크 착용상태에서도 적용가능한 모델로 확장하였다.

키워드: 거북목 증후군(turtle neck syndrome), CNN(Convolutional Neural Networks)

I. Introduction

현대 사회는 정보화 시대로 스마트폰, 태블릿, 노트북 데스크톱 등의 전자기기 사용율이 점차 높아지고 있고, 더군다나 현재 코로나 사태로 인해 더더욱 외부활동이 줄고 실내에서 전자기기를 활용하여 여가를 보내는 시간이 현저히 증가하였다. 이에 따라 거북목 증후군의 발병이 늘어날 것으로 예상된다. 거북목은 장시간 전자기기를 사용하거나 모니터의 높이가 눈높이 보다 낮을 경우 발생한다. 이러한 거북목 증후군의 예방 방법으로는 모니터의 높이를 조정하거나 스트레칭 정도가 있으며 프로그램 쪽으로는 핸드폰의 자이로 센서를 이용하여 핸드폰이 특정각도를 벗어나면 화면이 어두워지는 기능, psd센서를 이용하여 모니터와의 거리를 측정하여 거리가 가까워진다면 알람을 주는 기능 등이 현재 있다. 거북목 증후군에는 단계가 있는데 올바른 자세에서 일자목 (I자목)으로 변형되고 그다음이 C자목(역 커브 증후군)으로 이어진다. 이 논문에서는 컨볼루션 신경망 (CNN)을 이용하여 올바른 자세와 C자목 I자목 데이터를 학습하여 자신의 자세가 어떤 자세인지 거북목 증후군 진행 단계 중 어느 위치에 있는지 알려주는 모듈을 개발하였다.

II. Preliminaries

1. Related works

현대 21세기에서 위험한 병인 거북목 증후군이라는 것이 있다. 이 증후군은 머리가 구부정하게 앞으로 나오는 자세를 오래 취해 목이 일자 목으로 바뀌고 뒷목, 어깨, 허리 등에 통증이 생기는 증상을 말한다[1]. 이런 자세가 만들어지는 데에는 특히, 장시간 컴퓨터를 사용하게 될 경우 무의식적으로 머리를 앞으로 향한 채 구부정한 자세로 앉아있는데 이러한 자세를 장시간 하고 있으면 이 증후군에 걸리거나 증후군 증세가 진행될 수 있다. 이 거북목 증후군을 미리 예방하고자 컴퓨터를 많이 사용하는 현대인들을 위해 PSD(거리)센서와 피어센을 연동시킨 프로그램이 개발된바 있다 [2].

거북목 증후군은 목의 앞 근육이 길어지고, 위쪽 근육이 짧아져 몸통에 비해 머리가 앞으로 나와 있는 자세이며, 수술이나 약물치료보다 평소의 자세 습관을 고치는 방법이 효과적이다. [3]에서는 실시간으로 거북목 증후군을 유발할 수 있는 자세를 감지하고 경고하는 시스템이 제안되었다.

III. The Proposed Scheme

1. 시스템 구현

CNN은 필터링 기법을 인공신경망에 적용함으로써 이미지를 더욱 효과적으로 처리하기 위해 (LeCun et al., 1989)에서 처음 소개되었으며, 이후에 (LeCun et al., 1998)에서 현재 딥 러닝에서 이용되고 있는 형태의 CNN이 제안되었다. CNN의 기본 개념은 "행렬로 표현된 필터의 각 요소가 데이터 처리에 적합하도록 자동으로 학습되게 하자"는 것이다. 이는 거북목 증후군을 판별하는데 유용하다. 따라서 올바른 자세와 i자목과 c자목[4] 3개 카테고리 데이터를 모아 합성곱을 통해 학습을 한 뒤 자세 데이터를 받아오면 올바른 자세인지 i자목인지 c자목인지를 판별해 준다.

Fig 1은 구현 시스템의 전반적인 구성을 입력부터 출력에 이르기까지의 단계를 보여주고 있다. 단계별 설명은 III.2와 III.3에서 상세한다.

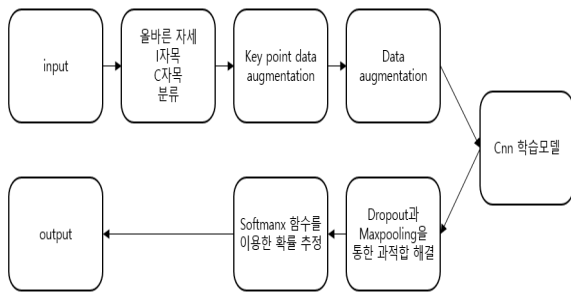


Fig. 1. 전체 시스템 구성

시스템 구현 환경으로는 jupyter notebook을 사용하였다. 언어는 python을 사용하였다. 라이브러리는 keras, matplotlib, pandas, numpy를 사용하였다.

2. 학습데이터 구성

학습에 사용한 트레인 데이터로는 처음에 인터넷에서 데이터를 다운받아 사용하였으나, i자목과 c자목의 구분이 애매한 경우가 많아 직접 웹 캠이나 셀프 카메라로 촬영하였다. 올바른 자세 i자목 c자목을 촬영 하였으며 더블모니터를 사용하는 사람의 경우 옆모습이 중요할 수도 있다고 생각하여 앞의 3카테고리의 정면 좌측 우측 사진 데이터도 추가 하였다.

또한 현재 코로나 사태로 마스크를 쓴 상태에서도 측정이 가능하도록 앞에 있던 3가지 카테고리를 마찬가지로 마스크를 쓴 상태에서도 같이 데이터를 구축 하였다. 여기서 제일 중요한 것은 데이터를 얼마나 잘 분류하느냐 인데 개나 고양이를 구분 하는 것은 명확한 답이 있기에 처음 데이터를 구축할 때 실수 없이 구축을 할 수 있으나 올바른 자세 i자목 c자목은 대부분 비슷한 양상을 띄어 애매한 데이터를 학습시키게 된다면 학습 모델의 성능이 현저하게 저하된다는 문제점이 있다.

해결 방법으로 매번 데이터를 구축할 때 마다 3가지 카테고리를 체크 하는 방법과 teachablemachine 사이트[5] 에서 3가지 class에 3가지 카테고리를 넣은 뒤 학습을 시켜 웹캠으로 테스트를 해본

뒤 데이터에 구축하는 두가지 방법으로 데이터를 모았다.

현재 구축된 c자목 248개, 올바른 자세 352개, i자목 304개 데이터이다. 각각은 여러 사람이 직접 찍은 사진으로 구성되어 있으며 앞서 인터넷에서 다운 받았던 데이터 들은 다양한 사람들이 찍은 데이터 이므로 테스트에 더 적합하다 생각하여 위와 마찬가지로 3가지 카테고리로 나누어서 테스트 데이터로 구축하였다.

데이터 부족 문제를 해결하기 위해 Data Augmentation을 진행하였는데, 거북목은 눈, 코, 입과 어깨의 관련성이 높다고 생각되어 눈, 코, 입, 귀와 어깨를 연결하는 Keypoint Augmentation을 진행하였다. 사용하였다. Keypoint Augmentation은 teachable machine 사이트에서 pose project에 기존 데이터를 카테고리 별로 넣어서 학습을 시킨 뒤 Keypoint가 적용된 데이터를 다시 다운받는 방식으로 실시하였다.

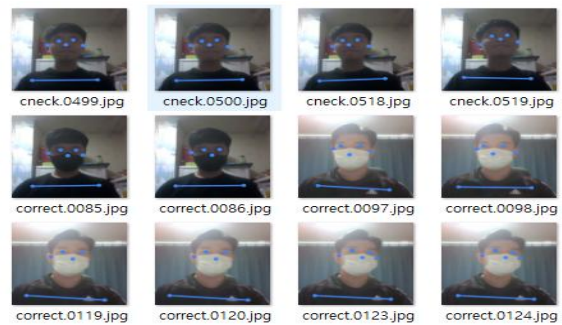


Fig. 2. Keypoint Augmentation

추가적으로 또한 ImageDataGenerator [6]를 통하여 rotation, resize, shear, zoom, flip, shift augmentation을 Fig. 3과 같이 진행하였다.

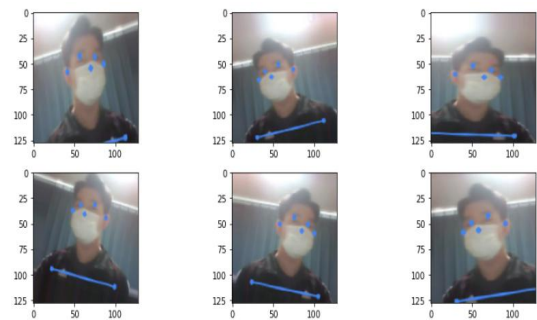


Fig. 3. Data Augmentation

3. 학습 진행

학습 모델로는 CNN의 구조 중 Fig 4와 같은 구조를 사용하였다. convolution은 데이터로부터 특징을 추출하고 max-pooling은 과적합을 막으면서 사이즈를 줄이는 역할을 한다 [7]. 또한, dropout을 통해 50%의 node만 사용하여 과적합을 회피하였다. 최종적으로 Softmax를 통해 올바른 자세 i자목 c자목 총 3가지 카테고리를 판별하도록 하여 확률을 추정하였다. Epoch는 50으로 들렸으며 총

11분정도의 시간이 걸렸으며, 학습정확도는 97%에 도달하였다 (Fig 5).

Fig 6의 상단 그래프는 푸른색이 training loss, 붉은색이 validation loss 인데 27epoch 즈음에서만 높은 loss를 보이고 epoch가 증가할수록 loss가 0.02~0.04 부근으로 낮아져 오차가 작아짐을 알 수 있다. 또한 아래 그래프에서도 마찬가지로 정확도가 27 epoch 즈음에서 감소하지만 계속 증가하여 모델로 쓰기에 적합함을 보여주었다.

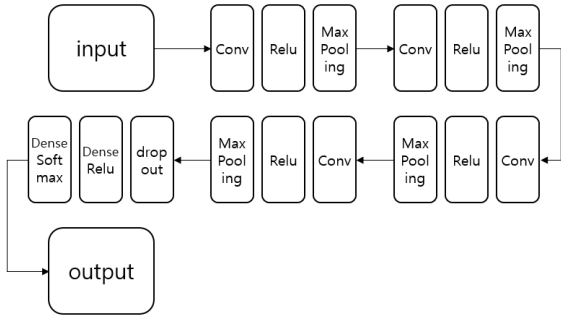


Fig. 4. 학습 모델 구조

```
13s 265ms/step - loss: 0.0694 - accuracy: 0.9760 - val_loss: 0.0409 - val_accuracy: 0.9778
ETA: 0s - loss: 0.1211 - accuracy: 0.9689WARNING:tensorflow:Reduce LR on plateau conditioned
e. Available metrics are: loss,accuracy,val_loss,val_accuracy,lr
13s 264ms/step - loss: 0.1211 - accuracy: 0.9689 - val_loss: 0.0069 - val_accuracy: 1.0000
ETA: 0s - loss: 0.0946 - accuracy: 0.9746WARNING:tensorflow:Reduce LR on plateau conditioned
e. Available metrics are: loss,accuracy,val_loss,val_accuracy,lr
13s 264ms/step - loss: 0.0946 - accuracy: 0.9746 - val_loss: 0.2603 - val_accuracy: 0.9722
ETA: 0s - loss: 0.0899 - accuracy: 0.9703WARNING:tensorflow:Reduce LR on plateau conditioned
e. Available metrics are: loss,accuracy,val_loss,val_accuracy,lr
13s 263ms/step - loss: 0.0899 - accuracy: 0.9703 - val_loss: 0.0265 - val_accuracy: 0.9778
```

Fig. 5. 학습 모델 결과

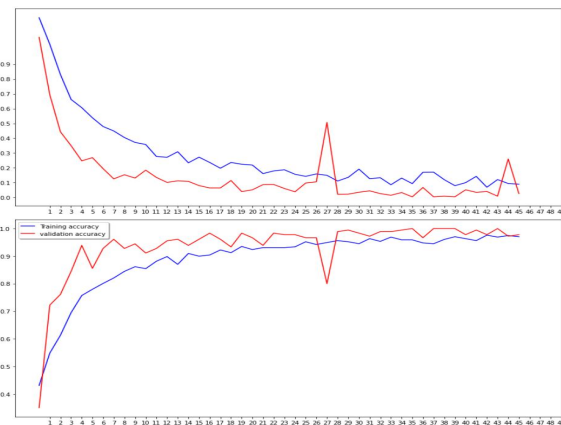


Fig. 6. 학습 모델 결과

4. 테스트 결과

테스트 데이터는 인터넷에서 다운받은 다양한 사람들의 데이터를 사용하였고 테스트 해본 결과 Fig 7과 같이 86.8%수준의 정확도를

보였다.

```
score_df['category'].value_counts(normalize=True)
True    0.868293
False   0.131707
Name: category, dtype: float64
```

Fig. 7. 테스트 결과

c자목 올바른자세 i자목의 각각 정확도는 85%, 88%, 88%로 크게 맞추지만 c자목이 대비 정확도가 낮았고 확인한 결과 c자목을 올바른자세로 오인을 하는 경우가 대부분이었다. 이는 c자목이 목이 굽고 고개가 위로 향하는 자세라 고개가 숙여지는 i자목에 비해 올바른 자세와 혼동을 하기 쉽다고 여겨진다. c자목의 정확도를 올리기 위해서는 트레인 데이터를 구축할 때 c자목과 올바른 자세의 차이점을 명확히 두고 분류를 해야 할 것으로 보여진다.

IV. Conclusions

구현한 거북목 증후군 진단모델의 정확도는 약 87%수준이며, 이는 다른 이미지 분류 테스크들과 비교할 때, 세가지의 자세들간의 경계가 모호하기 때문인것으로 사료된다. 향후 구분 기준을 더 명확하게 정립할 필요가 있다.

또한, 현재 카테고리 3가지 밖에 없어 3가지 모두 해당하지 않는 자세가 나올 때에 문제가 생겨 정확도에 안 좋은 영향을 끼친다. 이 점은 각각의 카테고리의 정확도를 구한 뒤 임계치 70%를 상회하지 않으면 또다른 카테고리 (즉, 애매모호한 자세)를 추가하는 것도 고려중이다.

향후 웹캠과 연동하여 실시간으로 매 자세 마다 i자목 c자목 올바른 자세 인지 알려주는 앱을 개발할 생각이다. 이로 인해 자신이 거북목 자세인지 알 수도 있으며 자주 나타나는 자세를 가지고 현재 자신의 목상태가 i자목인지 c자목이 진행 중인지를 알아 거북목 증후군을 미연에 방지 할 수 있을 것으로 기대한다.

REFERENCES

- [1] <https://spinedoctor.tistory.com/1166>
- [2] GitHub - InguChoi/Turtle-neck-Prevent-Program: Turtle Neck Prevent Program
- [3] Han, J.-Y., & Park, J.-H. (2020). CNN기반의 학습모델을 활용한 거북목 증후군 자세 교정 시스템. 한국콘텐츠학회논문지, 20(7), 47-55. <https://doi.org/10.5392/JKCA.2020.20.07.047>
- [4] <https://www.spinnetmj.net/c>
- [5] <https://teachablemachine.withgoogle.com/train>
- [6] https://keraskorea.github.io/posts/2018-10-24-little_data_powerful_model/

- [7] <https://m.blog.naver.com/PostView.nhn?blogId=tjdrud1323&logNo=221500725386&proxyReferer=https:%2F%2Fwww.google.com%2F>