

열영상과 IoT를 이용한 AI 바이러스 차단을 위한 딥러닝 알고리즘 비교에 대한 연구

노승현*, 서호준*, 김혜인*, 김정민

*안양대학교 컴퓨터공학과

charyuza98@naver.com, hoojoun@gmail.com, hik369@naver.com

A Study on Deep learning algorithm comparison for Block AI virus using thermal video and IoT

No Seunghyun*, seo hojun*, kim hyein*

*Dept. of Computer Engineering, Anyang University

요 약

열영상과 IoT를 이용한 AI 바이러스 차단 시스템 개발에 필요한 열화상 체온 측정기의 열 측정 정확도 향상과 얼굴 인식 시간 단축을 위해 열화상에 사용되는 딥러닝 알고리즘을 비교하며 효율적인 알고리즘 발굴 및 열영상을 이용한 바이러스 차단 시스템에 적합한 열영상 알고리즘 보완 방법을 찾는 연구이다.

I. 서론

코로나가 처음 발생한 2019년 12월부터 지금까지 1년 반이 넘는 기간 동안 유행하며 사람들에게 큰 변화를 안겨주었다. 일반인으로서 가장 쉽게 볼 수 있는 변화는 마스크와 열화상 체온 측정기이다. 그 중에서도 열화상 체온 측정기는 거의 모든 시설에 설치되어 확진자 격리에 사용되는데, 열화상 체온 측정기는 확진자를 가려내는 데 유의미한 도움이 되지 않는다. 단순히 체온만 측정하기 때문에 확진자 중에서도 발열 증상이 없는 환자는 찾아낼 수 없기 때문이다. 심지어 체온을 측정할 때도 실제 사람의 체온을 재는 것이 아닌 인체에서 발산되는 열을 측정할 뿐이기 때문에 고열 환자 이외에는 효과가 없는 것이 현실이다. (열화상 체온 측정기는 그 외에도 습도나 안개, 연기, 바람, 물체의 물기 등 여러 영향 요소로 인해 큰 차이가 발생한다.) 또한, 기존의 열화상 체온 측정기는 사람이 얼굴을 갖다 대야 제대로 인식되는 경우가 잦으며 얼굴을 인식하는 데 짧지만 나름의 시간이 걸리는 문제 또한 존재한다.

이러한 문제를 최대한 해결하기 위해 여러 성능 향상 방법들을 고민해 본 결과 현재 수준에선 그나마 예산이 적게 소모되는 소프트웨어, 그중에서도 딥러닝 알고리즘 부분을 수정해보자는 생각이 들었다. 또한 소프트웨어 부분을 수정할 경우 현재 사용되고 있는 제품을 구매해 가공하기엔 현실적이지 않

기 때문에 가공한 소프트웨어를 담을 수 있을 만한 mock-up을 따로 제작하기로 했다. 다음은 여러 알고리즘을 시도해 보며 비교한 연구이다.

II. 본론

1. 사용한 알고리즘과 모델

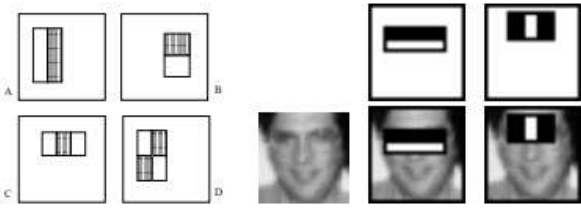
~6월 4주차	~8월 1주차	~8월 4주차
Haar-like feature 기반의 Viola-Jones 알고리즘		CNN 기반의 ResNet 알고리즘
HAAR cascades 모델	Facial landmark detector 모델	ResNet50

[표 1] 사용한 알고리즘과 모델 변경 이력

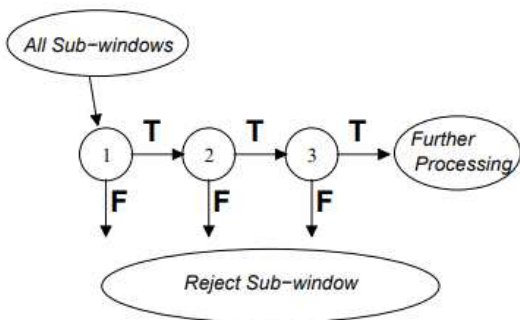
실제로 사용한 알고리즘은 Viola-Jones 알고리즘과 resnet 알고리즘으로 두 가지이지만 사용했던 모델은 OpenCV HAAR cascades, Facial landmark detector(5-point)[1], ResNet50 으로 3가지이다. 다음은 각 알고리즘과 모델의 설명이다.

1.1. Viola-Jones 알고리즘

Haar Cascades를 사용하는 라이브러리인 OpenCV는 Paul Viola와 Michael Jones에 의해 처음 개발되었다. 이렇게 개발된 얼굴 검출기를 Viola-Jones 검출기라고 부르며 OpenCV에서는 이 검출기를 Haar Cascades(하르 분류기)라고 지칭한다. 작동하는 방식은 다음과 같다.



[그림 1] OpenCV에서 사용되는 Haar-like wavelet
Viola-Jones 검출기는 적분 영상을 사용하여 [그림 1]에서 보이는 사각형 영역에서의 연산을 빠르게 수행한다. 또한, Viola-Jones 검출기는 각 노드에서 AdaBoost(약한 학습기)를 rejection cascade로서 사용한다. [2]



[그림 2] Viola-Jones 분류기의 rejection cascade
Viola-Jones 검출기는 각 집단을 분류할 때 약한 학습기를 사용하여 긍정 이미지를 고르는 것이 아닌 부정 이미지를 걸러내는 데 주력화되어 있다.
[그림 2]의 각 노드는 단순한 것부터 복잡한 것으로 정렬되어 있으며 얼굴이 아닌 영역을 걸러낼 때 필요한 연산 시간을 최소화할 수 있게 동작한다. 예를 들자면, Viola-Jones 분류기를 훈련할 때 얼굴인 영역이 99.99%의 확률로 검출되고, 얼굴이 아닌 영역이 40%의 확률로 검출된다고 가정한다. 이때의 작업은 노드를 3개만 거처도 얼굴 영역 99.97%(99.99%³), 얼굴이 아닌 영역은 6.4%(40%³)로 검출된다.

1.1.1. OpenCV HAAR cascades 모델

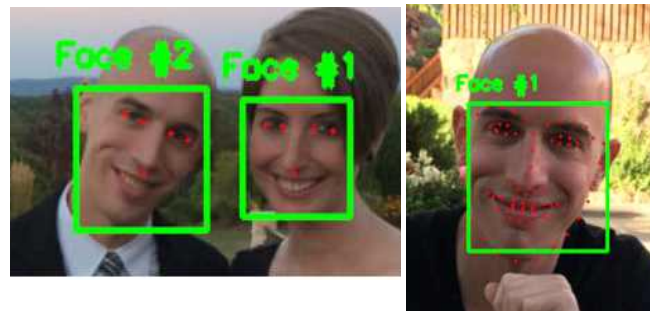
Haar cascades 모델은 OpenCV 라이브러리에 내장되어 있어 초기 개발 단계부터 쉽게 사용할 수 있다. 프로젝트를 진행하면서 하루빨리 얼굴 인식 결과물을 확인하고 싶었기 때문에 선택하게 되었다.

하지만 Haar cascades 모델은 정면 얼굴처럼 확실히 구분되는 객체만 검출할 수 있었고 카메라의 각도가 비틀어지거나 어두운 환경에서는 검출이 거의 불가능했기 때문에 폐기하였다.

1.1.2. Facial landmark detector(5-point) 모델

Haar cascades 모델은 완벽하게 준비된 환경 외에선 제대로 동작하지 않는 경우가 잦았다. 그렇기에 다른 모델을 찾다 이 모델을 발견하게 되었다.

Haar cascades 모델과 같이 Haar-like feature를 사용하는 모델이지만 알고리즘이 달라 성능이 더 뛰어났고 무엇보다 모델 용량이 적어서 라즈베리파이 기반인 본 프로젝트에 걸맞다고 생각해 선택하게 되었다.

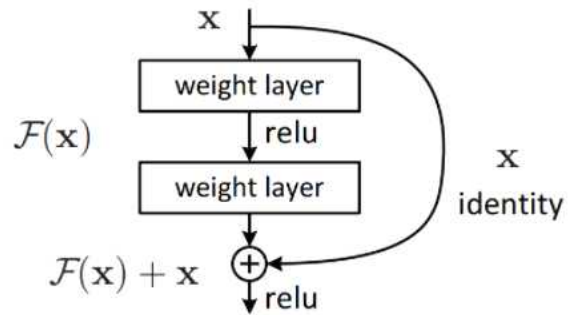


[그림 3] 5-point 얼굴 랜드마크(좌) [1]

Facial landmark detector 모델은 얼굴의 특징점을 찾아내는 방식으로 더 빠른 동작을 가능케 한 모델이다. 기존의 68-point 모델보다 8-10% 빠르게 동작하며 모델 크기는 10분의 1로 감소했다. 작업 환경이 라즈베리파이이기 때문에 최적화 명목으로 사용했지만 정작 마스크를 착용한 상태에선 인식하지 못했기 때문에 폐기하였다.

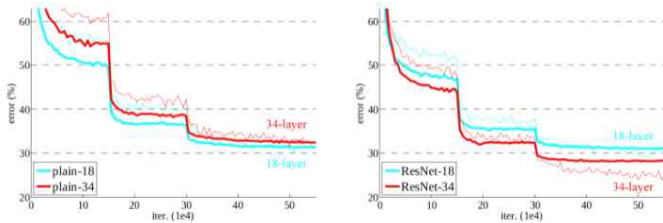
1.2. ResNet 알고리즘

2012년 ILSVRC 대회에서 AlexNet이 승리를 거둔 후 합성곱 신경망의 깊이는 점점 더 깊어졌다. 하지만 층이 깊어질수록 역으로 학습이 잘 안 되는 문제가 발생했는데, 그래서 제안된 개념이 Residual block이다.



[그림 4] Residual learning(잔여 학습) block
(해당 그림에서 F(x)는 Model, x는 input을 뜻한다.)
Residual block은 깊은 층을 가진 합성곱 신경망

의 문제인 훈련이 어려워지는 것과 정확도 저하 문제를 기존의 $y=F(x)+x$ 를 $F(x)=y-x$ 로 표현하여 출력과 입력 간의 차이를 학습하도록 설계되었다.[5] 따라서 입력값이 출력값에 들어감에 따라 입력값이 잊히는 정확도 저하 문제가 해결되었다.



[그림 5] Residual block 추가 후 성능 비교

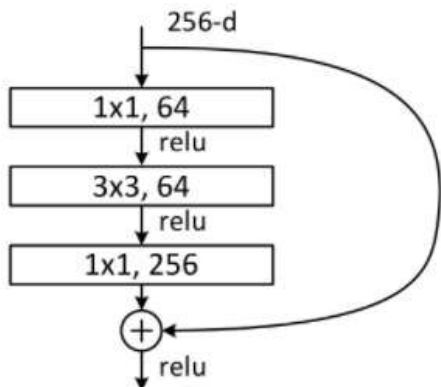
1.2.1. ResNet50 모델

ResNet50 모델은 전 모델을 폐기하고 다른 모델을 찾던 도중 최근 모델인 데다 성능도 뛰어나며 관련 레퍼런스도 많았기에 선택하게 되었다.

ResNet50 모델은 ImageNet 데이터베이스에서 수많은 이미지로 훈련된 합성곱 신경망이다. 다음은 ResNet50 모델의 구조와 ResNet50의 Residual block 구조이다.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

[그림 6] ResNet50 architecture



[그림 7] ResNet50의 Bottleneck building block 다음은 1×1 합성곱 층을 추가하여 3개의 층을 가지게 된 ResNet50의 Residual block 구조이다.

2. 모델 비교

HAAR cascades	Facial landmark detector	ResNet50
0.188s	0.164s	0.123s

[표 2] 얼굴 인식 후 다음 인식까지 걸리는 시간의 평균값 비교

인식 간격의 값은 큰 차이가 없어 보이지만 작동시에 ResNet50을 제외한 나머지 두 모델은 추적이 제대로 되지 않는 프레임의 수가 많았다.

HAAR cascades	Facial landmark detector	ResNet50
0.02s	0.04s	0.08s

[표 3] 한 프레임을 인식하는 데 걸리는 시간의 평균값 비교

모델이 복잡해질수록 한 프레임을 읽는 속도는 느려졌지만, 다음 프레임을 인식하는 데 걸리는 속도는 빨라졌다.

III. 결론

HAAR cascades 모델과 Facial landmark detector 모델, 서로 다른 모델이지만 두 모델 다 Haar-like feature를 기반으로 사용한다. 여기서 [표 2]와 [표 3]을 통해 사용하는 감지 모델이 같아도 얼굴을 추적하는 알고리즘에 따라 큰 성능 차이를 보이는 것을 알게 되었다. 또한, Facial landmark detector 모델과 ResNet50 모델의 비교를 통해서 기존에 학습된 모델을 사용하는 것과 사용자가 임의로 학습시켜 모델을 사용하는 것에 큰 차이가 있다는 것도 알게 되었다. 따라서, 관심 있는 문제점(각 모델의 성능 평가 지표)을 해결하는 데 큰 도움이 되었다.

하지만 ResNet50 모델의 경우 직접 학습한 모델을 사용해야 하는 점에 대해서 학습의 귀찮음 해결(데이터 증강기법 등)과 데이터셋의 용량, 데이터셋의 보안 문제를 해결하기 위해 보안 연구가 필요할 것으로 예상된다.

마지막으로, 이 연구가 향후 열화상 카메라 알고리즘 탐색에 도움이 되어 더 발전된 성능의 열화상 체온 측정기가 만들어지길 바라며 코로나 바이러스 차단에 사용되는 지금의 열화상 체온 측정기가 더 정확하게 감염병의심자를 파악하는 데 도움이 되었

으면 좋겠다고 생각한다.

참고문헌

- [1] Adrian Rosebrock "(Faster) Facial landmark detector with dlib" pyimagesearch, April 2, 2018
- [2] Paul Viola, Michael Jones "Rapid Object Detection using a Boosted Cascade of Simple Features" COMPUTER VISION AND PATTERN RECOGNITION 2001
- [3] Gary Rost Bradski, Adrian Kaehler "OpneCV 제대로 배우기" 한빛미디어 September 27, 2009
- [4] Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, "Deep Residual Learning for Image Recognition" COMPUTER VISION AND PATTERN RECOGNITION 2015
- [5] Aakash Kaushik, "Understanding ResNet50 architecture", OpenGenus IQ

-본 논문은 과학기술정보통신부 정보통신창의인재양성사업의 지원을 통해 수행한 ICT멘토링 프로젝트 결과물입니다-