실시간 표정 분석을 바탕으로 한 유사한 이모지 표시

Overlaying emoji similar to feeling by analyzing facial expressions in real-time

이석훈*, 홍건화*, 박성진*

SeokHoon Yi*, Kun-hwa Hong*, Sung-jin Park*

요 약

PC, 스마트폰 사용이 늘어나면서 인터넷을 활용한 의사소통이 늘어나고 있다. 하지만 이러한 방식의 의사소통은 대개 텍스트로만 이루어지기 때문에 텍스트에서 전해지지 않는 감정적 표현이 가능한 이모지의 활용이 늘어나고 있다. 본 논문에서는 실시간으로 사용자의 얼굴에서 나타나는 표정을 분석하여 사용자의 감정을 알아내고, 이를 통해 알아낸 감정과 유사한 이모지를 사용자의 얼굴에 오버레이(Overlay)시키는 기능을 제안한다. 실시간 표정 분석은 빠른 속도와 높은 정확도가 동시에 요구되기 때문에 CNN(Convolutional Neural Network)을 기반으로 한 mini-Xception 모델을 활용하였다. mini-Xception 모델은 기존의 CNN 모델보다 필요한 매개변수가 적어 실시간 표정 분석에 적합하다.

■ 중심어 : 표정 분석, 감정, 이모지, 오버레이, CNN(Convolutional Neural Network), mini-Xception 모델

Abstract

As the use of PCs and smartphones increases, communication using the Internet is increasing. However, since this type of communication usually consists only of text, the use of emoji capable of emotional expression that is not delivered from text is increasing. This paper proposes a function to find out the user's emotions by analyzing the expressions on the user's face in real-time, and to overlay an emoji similar to the found emotion on the user's face. Because real-time facial expression recognition requires fast speed and high accuracy, mini-Xception model based on CNN(Convolutional Neural Network) was used. The mini-Xception model is suitable for real-time facial expression recognition because it requires fewer parameters than the existing CNN model.

■ keywords : facial expression recognition, emoji, overlay, CNN(Convolutional Neural Network), mini-Xception model

Ⅰ. 서 론

PC 보급이 보편화되고 컴퓨터 활용 기술이 발달하면서 동영상 스트리밍 사이트, 커뮤니티 사이트 등의 웹사이트와 게임, 음악 플레이어 등의 다양한 프로그램들이 생겨났다. 그러면서 인간에게 컴퓨터는 단순한 계산 기능과 문서 작성 등을 담당하는 딱딱하고 사무적인 존재가 아니라 인터넷을 통해 일상생활에서 다른 사람들과 같이 소통하면서 감정을 공유할 수 있는 감성적이고 친근한 존재가 되었다. 또한, PC에 이어 스마트폰도 대중들에게 보급되기 시작하며 인터넷을 활용한 사회적 활동의 범위는 더욱 방대해졌고 우리들의 일상 속으로 스며들게 되었다. 페이스북, 트위터 등의 SNS를 통해 전 세계 사람들과 소통하기도 하고, 각종 포털 사이트나 커뮤니티 사이트에서 다양한 사람들과 정보와 의견을 주고받기도 하고, 동영상·음악 스트리밍 사이트에서 감상평 등의 댓글을 다는 등 사람들의 생각과 의견을 인터넷상에서 표현하는 일이 많아진 것이다.

인터넷을 통한 의사소통이 늘어남에 따라 자연스럽게 이모티콘, 이모지 등의 새로운 의사 표현 방식이 만들어졌다. 실시간으로 상대방의 표정·손짓·몸짓을 보고, 말의 억양·어조·크기 등을 파악할 수 있는 현실 세계에서의 의사소통과 다르게 인터넷에서의 의사소통은 대부분 텍스트로만 이루어지기 때문이다. 텍스트만으로 잘 전해지지 않는 추가적인 감정표현을 하기 위해 이모티콘(Emoticon)[1)]이 1982년 Scott E. Fahlman 교수에 의해 처음 사용된 이후 인터넷 시장이 커짐에 따라 점차 대중화되면서 전 세계의 네티즌들이 이모티콘을 사용하게 되었다[1]. 하지만 이모티콘은 단순한 기호와 숫자 등의 조합이라 표현의 정확도나 다양성 측면에서 뚜렷한 한계가 존재했다. 그리하여 1999

1) 이모티콘(Emoticon)은 문자, 기호, 숫자 등을 조합해 감정을 표현하는 것이고 이모지(Emoji, 繪文字)는 유니코드 체계를 이용한 그림문자로 감정을 표현하는 것이라 서로 다른 개념이지만 현재 우리나라에서는 이모티콘을 이모지의 의미까지 포함해서 사용하고 있다. 하지만 이 글에서 말하는 이모티콘은 한국에서 통용되는 이모티콘과 이모지를 아우르는 뜻이 아닌 원래의 이모티콘의 뜻만을 얘기하고 있다.

년 일본의 인터페이스 디자이너 Shigetaka Kurita에 의해 현대적인 이모지(Emoji)가 만들어졌고 스마트폰의 보급이 활성화되면서 점차 그 종류와 사용량이 많아졌다. 오늘날 페이스북에서 하루에만 약 50억 개의 이모지가 사용될 정도로 이모지의 활용도는 크게 늘어난 상태이다[2].

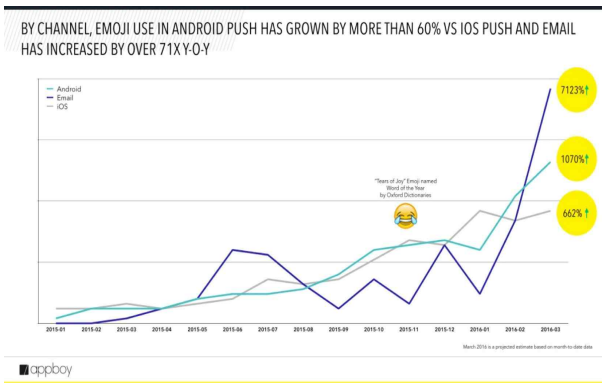

그림 1 이모지 사용량 추이 (2015~2016)

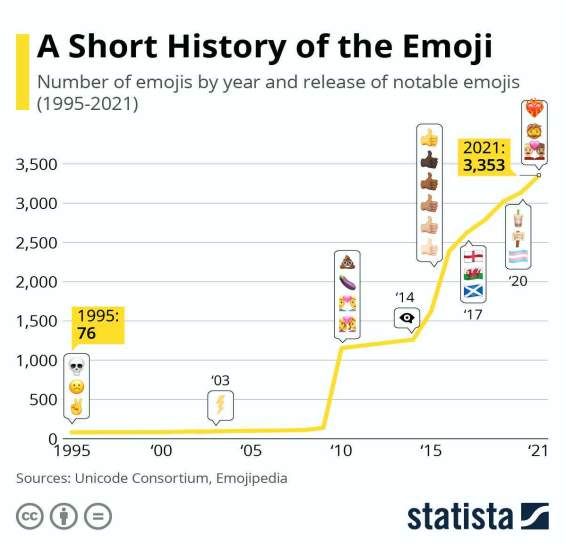

그림 2 해마다 늘어나는 이모지의 종류

한편, 2016년 세계를 놀라게 했던 바둑 인공지능 알파고(AlphaGo)와 이세돌 9단의 구글 딥마인드 챌린지 매치(Google DeepMind Challenge Match)로 대중들에게도 친숙해진 딥러닝(Deep Learning)은 압도적인 성능과 가능성을 보여주며 전 세계적으로 큰 관심을 불러일으키고 있다. 딥러닝은 기계학습에 속해 있는 개념으로, 업무를 수행함에 따라 자체적인 학습을 하는 알고리즘을 말한다[3]. 엄청난 양의 빅데이터(Big Data)를 다층 레이어를 이용하여 자가학습시킴으로써 예측 결과에 대한 정확도를 높일 수 있어 이미지 인식, 음성 인식, 언어 번역, 자연어 처리 등 기존의 방식과 알고리즘으로 해결하지 못하던 분야에 폭넓게 적용되며 다양한 문제를 해결하고 있다. 본 논문에서는 이러한 딥러닝을 활용한 방식 중 하나인 CNN(Convolutional Neural Network) 모델을 활용한 얼굴 표정 인식(Facial Expression Recognition, FER) 기술을 이용하여 사용자의 표정을 인식한다. 이를 통해 사용자의 표정을 분석하여 감정을 도출해낸 후 사용자의 얼굴에 그 감정에 해당하는 이모지를 오버레이(Overlay)시킴으로써 사용자의 감정을 나타내 주게 한다.

II. 본 론

1. CNN 모델을 활용한 얼굴 표정 인식

가. 초기의 얼굴 표정 인식 기술

딥러닝 기술이 발달하기 전에는 단일 이미지 또는 비디오 데이터를 통한 고전적인 기계학습 방식을 사용하여 얼굴의 특징을 분류하는 형태의 연구가 진행되었다. 주로 hand-crafted feature 방식을 사용하여 얼굴 표정의 고유한 특징을 추출하고 SVM(Support Vector Machine)이나 랜덤 포레스트(Random Forest) 등의 분류기로 그 특징들을 분석함으로써 얼굴의 표정을 파악하는 것이다[4]. 하지만 이와 같은 기술은 주변 배경 및 영상의 각도와 조도에 크게 영향을 받는다는 단점이 있다.

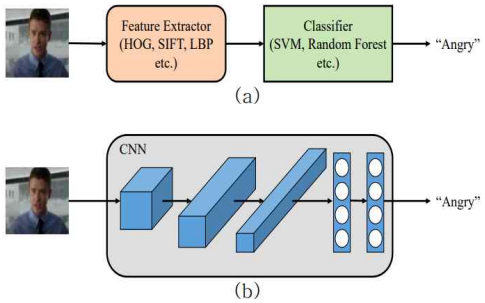

그림 3 (a) 고전적인 얼굴 표정 인식 기술
(b) CNN 기반의 얼굴 표정 인식 기술

나. CNN을 활용한 표정 인식 기술

최근 딥러닝 기술이 크게 주목받게 되면서 이미지 인식과 패턴 감지를 위한 최적의 아키텍처를 제공하는 CNN 모델을 기반으로 한 표정 인식 기술이 대세로 자리 잡게 되었다.

기존 사용되던 방식은 3차원 이미지를 1차원 배열로 바꾼 뒤 전 연결 신경망(Fully-connected Neural Network, FNN)으로 학습시키는 방법이었는데, 이 방식은 이미지를 벡터화하는 과정에서 매개변수가 너무 많이 요구되는 데다가, 인접 픽셀과

의 상관관계가 무시되고 공간 정보의 손실이 발생하는 문제점이 있었다. 그 단점을 보완하기 위해 이미지의 공간 정보를 유지한 상태로 학습할 수 있게 한 것이 바로 CNN 모델이다.

CNN은 크게 이미지를 추출하는 영역과 분류하는 영역으로 나눌 수 있다. 이미지를 추출하는 영역은 입력된 이미지에서 피쳐 맵(Feature Map)을 추출하는 Convolution Layer와 그렇게 추출된 피쳐 맵을 풀링(Pooling)하는 Pooling Layer가 여러 겹 쌓아져 있는 형태로 구성되어 있다. 이미지를 분류하는 영역은 앞선 과정에서 만들어낸 데이터를 기존의 전 연결 신경망으로 학습시킨 뒤 일련의 과정을 거쳐 최종적으로 이미지를 분류시킨다.

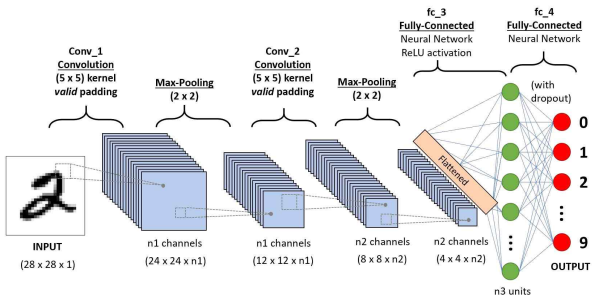

그림 4 CNN 모델의 전반적인 아키텍처

Convolution Layer에서는 이미지의 채널마다 필터를 이용해 합성곱 연산(Convolution)을 실행하여 피쳐 맵을 만드는 작업을 실행한다. 합성곱 연산이란 하나의 함수와 또 다른 함수를 반전 이동한 값을 곱한 다음, 구간에 대해 적분하여 새로운 함수를 만드는 개념으로, 2차원 이미지 데이터를 필터를 통해 순회하며 처리된다.

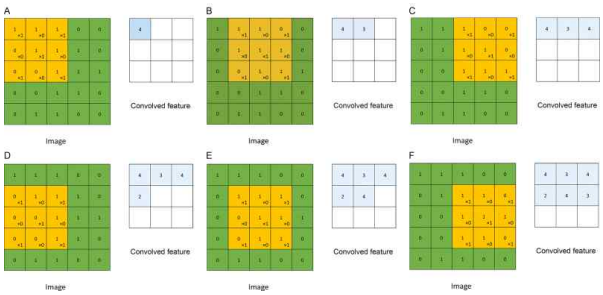

그림 5 Convolution Layer에서의 합성곱 연산 과정

여기서 말하는 필터는 이미지의 특징을 찾아내기 위한 공용 파라미터를 뜻하는 것으로 커널(Kernel)이라고도 하며, 입력된 이미지 데이터를 지정된 간격으로 순회하며 합성곱을 실행하여 피쳐 맵을 만드는 역할을 해준다. 이때 필터가 순회하는 지정된 간격을 스트라이드(Stride)라고 하며 필터의 크기가 클수록, 스트라이드 값이 클수록 피쳐 맵의 크기는 작아지게 된다. 또한, Convolution Layer에는 입력되는 이미지의 채널 개수만큼의 필터가 존재한다. 예를 들어 이미지가 흑백 명암으로 표현된 흑백사진이라면 1개의 채널만 생기게 되고, RGB 코드로 표현되었다면 R, G, B 값을 저장하는 3개의 채널이 생기게 된다. 그 후 각 채널마다 하나의 필터가 할당되기 때문에 전자의 경우에는 1개의 필터가, 후자의 경우에는 3개의 필터가 만들어지는 개념이다. 후자의 경우처럼 여러 개의 채널을 가진 경우, 채널별로 만들어진 피쳐 맵을 합산하여 최종 피쳐 맵으로 합성하는 과정을 거친다.

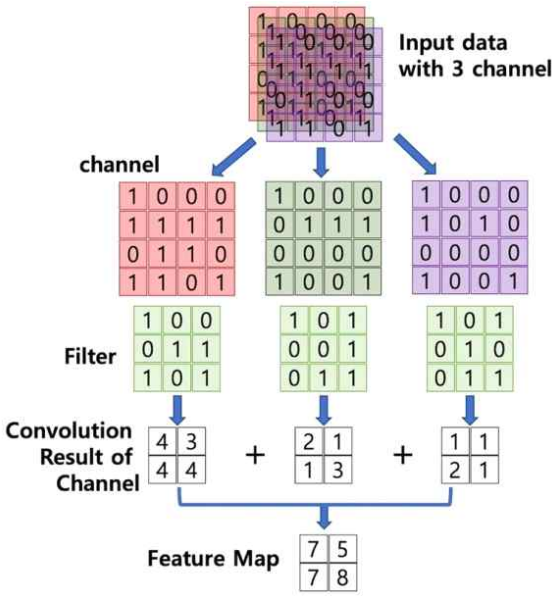

그림 6 멀티 채널에서 이루어지는 합성곱 연산 전체 과정

하지만 이러한 합성곱 연산을 실행하다보면 필터와 스트라이드에 의해 피쳐 맵이 초기의 입력 데이터보다 작아지게 되면서 이미지의 가장자리에 있는 픽셀들의 정보가 점점 사라지게 되고 만다. 이러한 문제점을 해결하기 위해서 패딩(Padding)이라는 방식이 도입되었다. 패딩은 이미지의 가장자리에 특정 값으로 설정된 픽셀들을 추가하여 합성곱 연산 후에도 이미지의 크기를 유지할 수 있게 만들어 주는 역할을 한다. 보통 특정 값은 0으로 설정하는 경우가 많으며, 이를 제로패딩(Zero-padding)이라고 한다.

이렇게 Convolution Layer에서는 각 채널별로 합성곱 연산을 통해 만들어진 최종 피쳐 맵에 활성화 함수(Activation Function)를 적용해 액티베이션 맵(Activation Map)을 만들어 최종 결과로 출력하는 기능을 담당한다. 이 과정에서 사용되는 활성화 함수는 입력된 데이터의 가중 합을 출력 신호로 변환하는 함수로, 선형시스템(Linear System)을 비선형시스템(Non-linear System)으로 바꿔주는 역할을 해준다. 활성화 함

수에는 Sigmoid, tanh, ReLU, Leaky ReLU 등 여러 함수가 있다. 그 중 ReLU 함수는 입력값이 0보다 작으면 0을 출력하고 0보다 크면 입력값 그대로 출력하는 함수로써, Sigmoid에서 발생하는 그래디언트 배니싱(Gradient Vanishing) 현상이 없고 단순한 데다가 성능도 좋기 때문에 CNN에선 보통 ReLU 함수가 많이 사용되는 편이다.

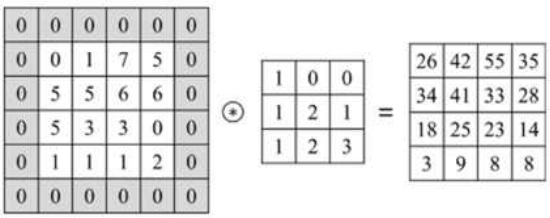

그림 7 제로패딩을 통한 합성곱 연산 (크기 유지)

Pooling Layer에서는 Convolution Layer에서 출력한 액티베이션 맵을 입력받아 크기를 줄이거나 특정한 값을 강조하는 풀링(서브 샘플링, Sub-sampling)을 한다. 풀링의 방법에는 맥스 풀링(Max Pooling), 평균 풀링(Average Pooling), 민 풀링(Min Pooling) 등이 있는데, 보통의 경우 맥스 풀링을 사용한다. 입력 데이터를 작은 구역으로 나누고 스트라이드도 동일하게 설정하면서 순회시켜 모든 원소가 한 번씩 처리될 수 있도록 한다. 맥스 풀링의 경우는 그 구역 중 최댓값을, 평균 풀링의 경우 그 구역의 평균값을 뽑아내는 등의 방식이다.

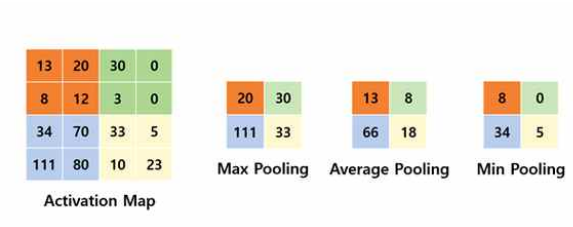

그림 8 맥스 풀링, 평균 풀링, 민 풀링의 예시

CNN은 이와 같은 Convolution Layer와 Pooling Layer를 반복적으로 겹쳐놓아 이미지의 강화된 특징을 추출하여 만들어진 데이터는 플래튼(Flatten) 과정을 거쳐 기존의 전 연결 신경망에 적용되는데, 플래튼 과정은 최종적인 Convolution Layer의 행렬을 신경망에 사용할 수 있게 1차원 데이터인 배열로 바꿔주는 역할을 해준다. 위의 신경망에서 분류하는 과정을 지난 후에 네트워크의 과적합(Overfitting) 현상을 해결하는 정규화(Regularization) 목적을 위해 학습 과정에서 무작위로 뉴런의 집합을 제거하는 드롭아웃(Dropout) 과정을 거쳐 최종적으로 소프트맥스 함수(Softmax Function)가 적용되어 이미지가 분류된다.

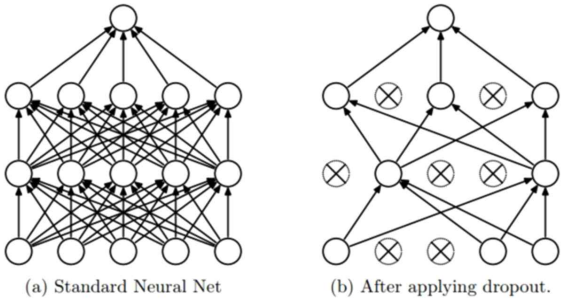

그림 9 (a) 드랍아웃을 하지 않은 경우
(b) 드랍아웃을 적용한 경우

다. CNN을 기반으로 한 향상된 모델들

위와 같은 CNN 모델이 이미지 인식 기술에서 두각을 나타내게 되면서 CNN 모델을 기반으로 하여 경량화·고성능화를 실현시킨 다양한 알고리즘이 등장하게 되었다.

기존의 CNN 모델에서 합성곱 연산에 사용되는 필터는 입력 데이터의 모든 채널에 영향을 받아 단일 채널의 공간 상호관계(Spatial Correlation)를 추출하는 것이 불가능했지만 Depthwise Convolution은 각각의 단일 채널을 분리하여 각 채널에 대해서만 이용되는 필터를 각기 생성하여 공간에서의 합성곱 연산을 한 뒤에 다시 합치는 방식으로써 프로세스에 사용하는 연산량을 줄일 수 있도록 하였다.

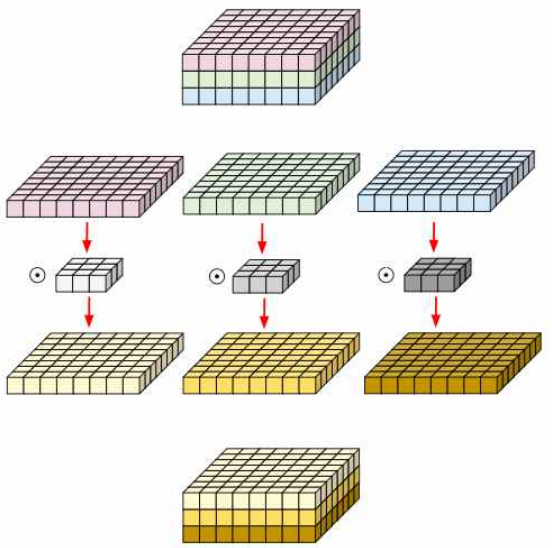

그림 10 Depthwise Convolution의 예시

Pointwise Convolution의 경우 공간에서의 합성곱 연산을 진행하는 것이 아니라 이와 반대로 채널 간의 상호관계

(Cross-channel Correlation)를 가능하게 하는 방식이다. 입력 데이터의 공간에서의 합성곱 연산 대신 1x1 크기의 필터를 사용하여 각 채널들에 대한 합성곱 연산을 진행함으로써 연산량을 낮춰 연산 속도를 향상시킨다.

앞서 소개한 Depthwise Convolution 방식과 Pointwise Convolution 방식을 혼합하여 사용하는 Depthwise Separable Convolution의 경우 곱연산 방식의 기존 합성곱 연산을 덧셈 방식으로 바꿔줌으로써 필요한 파라미터의 개수를 획기적으로 줄였다.

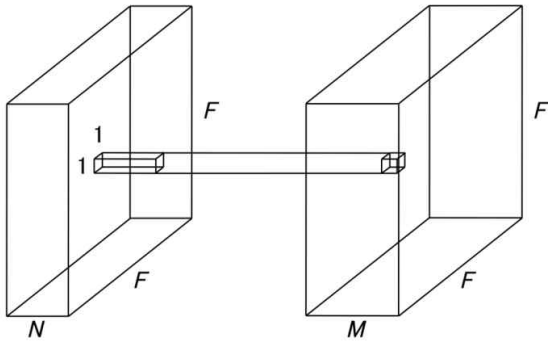

그림 11 Pointwise Convolution의 예시

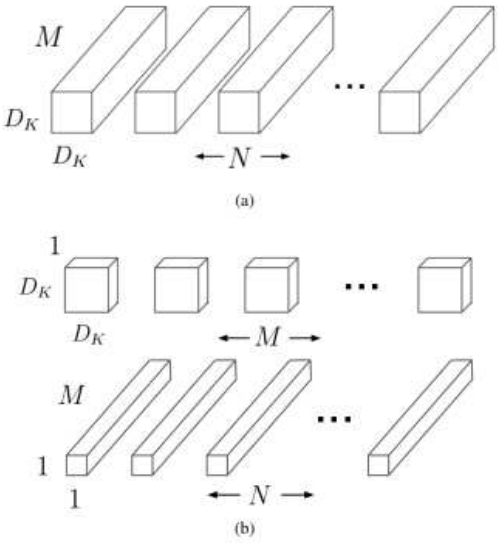

그림 12 (a) 기존의 Convolution 방식
(b) Depthwise Separable Convolution 방식

위의 Depthwise Separable Convolution을 기반으로 하여 피쳐 맵에서 단일 채널의 공간 상호관계와 채널 간의 상호관계를 완전하게 분리시켜서 매핑할 수 있다는 강력한 전제하에 만들어진 방식이 바로 Xception 모델이다. 본 논문에서 실시간 얼굴 표정 분석을 수행하기 위해 사용된 모델이 이 Xception 모델을 변형시킨 mini-Xception 모델인데, 잔차 연결(Residual Connection)을 활용한 4개의 중심 블록을 반복하고 합성곱 연산 후에 전역 평균 풀링을 거친 뒤 소프트맥스 함수를 적용해 최종적으로 이미지를 분류한다. 이 방식은 기존의 CNN 방식에서 사용된 파라미터의 개수를 80배나 줄여줌으로써 실시간 표정 분석에 적합한 알고리즘으로 선택되었다[5].

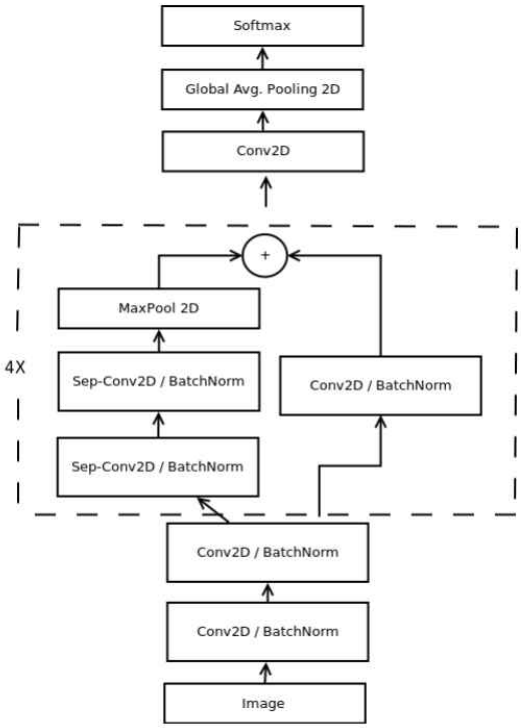

그림 13 얼굴 표정 인식에 사용된 mini-Xception 모델

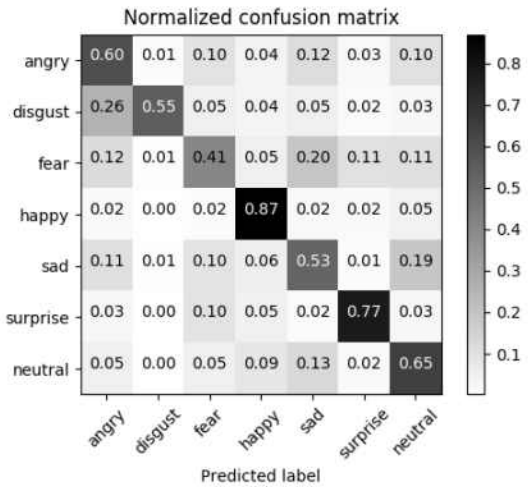

그림 14 mini-Xception 모델의 표정 분석 정확도

2. 제안한 방법

본 논문에서는 사용자의 얼굴 표정 분석을 위해 파라미터가 적어 연산속도가 빠른 위와 같은 mini-Xception 모델을 활용하였다. 이를 통해 사용자의 감정을 실시간으로 파악하여 '중립, 행복함, 화남, 슬픔, 놀람'의 5가지 감정으로 분류한다. 분석해낸 감정에 따라 미리 설정해놓은 해당 이모지를 사용자의 얼굴에 직접 오버레이시킴으로써, 사용자의 감정을 직접적으로 표현할 수 있게 하였다[6].

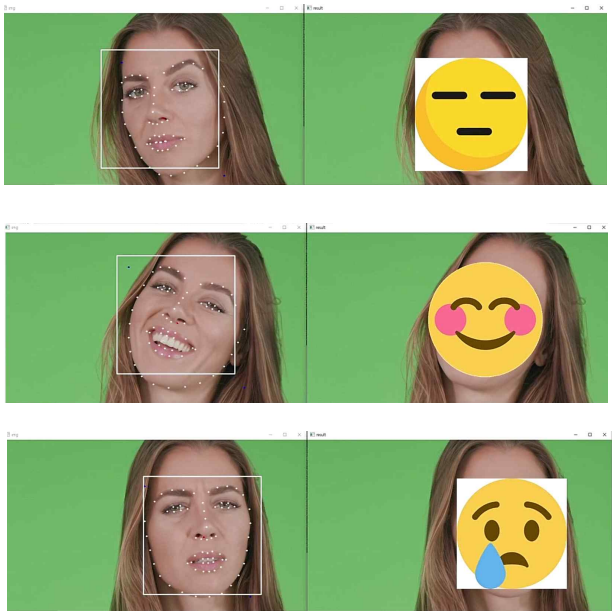

그림 15 실시간 얼굴 표정 분석 후 해당하는 이모지를 사용자의 얼굴에 오버레이시킨 실현 영상

III. 결 론

본 논문에서는 CNN을 활용한 많은 알고리즘 중에서 매개변수가 적어 실시간 얼굴 표정 분석에 적합하다고 여겨진 mini-Xception 모델을 차용하였다. 그리하여 다른 알고리즘보다 수월하게 사용자의 표정을 실시간으로 분석하고 감정을 도출해낼 수 있었다. 또한, 근래 들어 대중들 사이에서 사용이 늘어난 이모지를 활용하여, 위의 과정으로 도출해낸 감정과 결합함으로써 더욱 효과적이고 재미있게 사용자의 감정을 표현할 수 있게 만드는 데 성공했다. 최근 틱톡(TikTok), 페이스앱(FaceApp) 등 실시간 영상 보정을 활용한 애플리케이션이 유행하는 것을 감안한다면 이와 같이 이모지를 활용한 실시간 영상 보정도 충분히 수요가 있음을 짐작할 수 있다. 물론 이모지의 종류가 수없이 많은 데에 비해 인식할 수 있는 감정이 5종류에 불과한 것은 아쉬운 점이다. 향후 얼굴 표정 인식 기술이 발달하면 더욱 더 다양한 감정들을 이모지로 표현하는 것도 기대해 볼 만하다.

REFERENCES

[1] https://blog.emojipedia.org/emoji-trends-that-defined-2020/
[2] https://www.statista.com/chart/17275/number-of-emojis-from-1995-bis-2019/
[3] https://scienceon.kisti.re.kr/srch/selectPORSrchReport.do?cn=KOSEN000000000000293
[4] https://www.koreascience.or.kr/article/CFKO201904533827554.pdf
[5] https://arxiv.org/pdf/1710.07557.pdf
[6] https://github.com/shyicom12/cariphoto

저 자 소 개

이석훈(제1저자)
2015년~ 현재 아주대학교 소프트웨어학과 학사 재학 중

<주관심분야 : 컴퓨터네트워킹>

홍건화(제2저자)
2019년~ 현재 아주대학교 소프트웨어학과 학사 재학 중

<주관심분야 : 없음>

박성진(제3저자)
2015년~ 현재 아주대학교 소프트웨어학과 학사 재학 중

<주관심분야 : 데이터베이스>