



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공 학 석 사 학 위 논 문

감정노동자를 위한 딥러닝 기반의 스트레스 감지시스템의 설계

2022년 2월

부 경 대 학 교 산 업 대 학 원

컴 퓨 터 공 학 과

옥 유 선

공 학 석 사 학 위 논 문

감정노동자를 위한 딥러닝 기반의 스트레스 감지시스템의 설계

지도교수 조 우 현

이 논문을 공학석사 학위논문으로 제출함.

2022년 2월

부 경 대 학 교 산 업 대 학 원

컴 퓨 터 공 학 과

옥 유 선

옥유선의 공학석사 학위논문을
인준함.

2022년 2월 25일



주 심 공학박사 정 신 일



위 원 이학박사 윤 성 대



위 원 공학박사 조 우 현



목 차

목 차	i
표 목 차	ii
그 립 목 차	iii
Abstract	iv
I. 서론	1
II. 관련 연구	4
2.1 스트레스 감지(Stress Recognition)	4
2.2 CNN(Convolutional Neural Network)	7
2.3 이미지 분류 CNN 알고리즘	8
2.4 얼굴감정인식(FER:Facial Expression Recognition)	9
III. 본론	10
3.1 학습 데이터셋 구성	11
3.2 데이터 전처리 (Data Pre-Processing)	14
3.3 얼굴검출(Face Detection and Cropping) 모듈	15
3.4 Facial Expression Recognition(FER)모듈	17
3.5 스트레스 데이터 수치화(Digitalize) 모듈	24
IV. 실험결과 및 테스트	26
4.1 FER 모듈 실험결과	26
4.2 실시간 스트레스 감지 테스트	29
4.3 스트레스 감지 시스템 시각화	32
V. 결론	33
VI. 참고문헌	34

표 목 차

<표 1> 백인/아시아인의 인종별 평균 얼굴인식 응답시간	11
<표 2> 얼굴 감정 데이터셋 종류별 이미지 개수	13
<표 3> 얼굴 감정 분류별 이미지 데이터 개수	13
<표 4> 3가지 CNN 모델의 Top-1 Accuracy 및 파라미터 수	17
<표 5> FER 모듈 생성을 위한 CNN 모델 Transfer Learning 테스트	23
<표 6> 스트레스 데이터 수치화 및 연산기준	24
<표 7> FER 모듈 생성을 위한 CNN 모델 테스트 결과	26
<표 8> CNN 모델 테스트 결과 중 모델별 Top 3 Accuracy	27
<표 9> FER 모듈 인식률 테스트를 위한 표정 기준	29
<표 10> FER 모듈 인식률 테스트 정확도 판단 기준	30
<표 11> 피실험자별 실시간 인식률 테스트 결과 및 최종선정 FER 모듈	31

그 립 목 차

<그림 1> 전국 감정노동자 규모	1
<그림 2> 감정노동자 정신건강실태 및 산재현황	2
<그림 3> 콜센터 운영형태 현황 및 협력사 비중	2
<그림 4> CK+, Oulu-CASIA, KMU-FED 데이터셋	5
<그림 5> 스탠다드 CNN의 구조	7
<그림 6> ILSVRC 연도별 정확도	8
<그림 7> 이미지 오류를 포함한 Kaggle FER2013 dataset 일부	9
<그림 8> 스트레스 감지 시스템 설계 프로세스 전반	10
<그림 9> 백인/아시아인의 인종별 평균 얼굴인식 정확도	11
<그림 10> 데이터 셋 구성 프로세스 전반	12
<그림 11> 얼굴영역 검출 / 자르기 - 데이터 전처리	14
<그림 12> InceptionV3 모델의 아키텍처	18
<그림 13> Xception 모델의 아키텍처	18
<그림 14> MobileNetV2 모델의 아키텍처 및 퍼포먼스	19
<그림 15> InceptionV3 모델의 딥러닝 학습 로그 - Accuracy, Loss	28
<그림 16> Xception 모델의 딥러닝 학습 로그 - Accuracy, Loss	28
<그림 17> MobileNetV2 모델의 딥러닝 학습 로그 - Accuracy, Loss	28
<그림 18> FER 모듈 실시간 인식률 테스트	30
<그림 19> 실시간 스트레스 감지 시스템 수치변화 모니터링 그래프	32

Stress Detection System for Emotional Labor Based On Deep Learning
Facial Expression Recognition

Yu-Seon Og

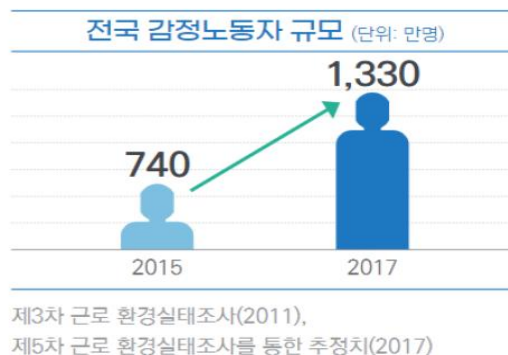
Department of Computer Engineering, The Graduate School of Industry,
Pukyong National University

Abstract

According to the growth of the service industry, mental illness and stresses from emotional labor workers have been emerging as a social problem, thereby so-called the Emotional Labor Protection Act was implemented in 2018. However, most emotional labor workers including customer service representatives are not being protected. this emphasizes insufficient substantial protection systems and the necessity of a digital stress management system. Thus, in this paper, we suggest a stress detection system for customer service representatives based on deep learning facial expression recognition algorithm. This system consists of a real-time face detection module, an emotion classification FER module that deep-learned big data sets including Korean emotion images, and a monitoring module that only visualizes stress levels. We designed the system to aim to monitor stress and prevent mental illness in emotional workers without any surveillance by management.

I. 서론

감정노동(emotional labor)의 개념은 Hochschild(1983)에 의하여 최초로 제시되었고[1] 서비스 산업에서 그 비중이 증가하고 있다. Morris & Feldman(1996)은 감정노동이란 서비스를 제공하는 중에 나의 감정이 아닌 조직에서 필요한 목표를 달성하기 위한 감정을 표현하는 행동을 의미한다[2]라고 정의하였으며, 우리나라에서는 서울특별시에서 ‘고객 응대 등 업무수행과정에서 자신의 감정을 절제하고 자신이 실제 느끼는 감정과는 다른 특정 감정을 표현하도록 업무상, 조직상 요구되는 노동 형태’라고 감정노동을 정의하고 있다.[3] 감정노동은 전 세계적으로 서비스 산업의 국가 경제에 영향력이 증가하는 서비스 경제화와 함께 중요성이 확대되고 있다. 우리나라의 산업별 GDP 구성 비율에서 서비스 산업의 비중은 1970년 44.3%, 2000년 57.5%에서 10년 후 2010년 59.3%로 증가하였으며 2017년 58%가 넘어 과반수 이상을 차지하고 있다.[4] 안전보건공단에서 실시하는 근로환경실태 조사에 의하면 전국 감정노동자의 규모는 2011년 740만 명에서 2017년 1,330만 명의 추정치로 대폭 증가한 것을 알 수 있다.



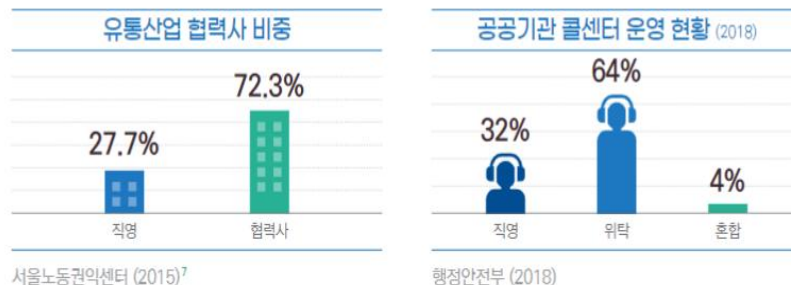
<그림 1> 전국 감정노동자 규모 [3]

이러한 증가 추세와 함께 그림2와 같이 감정노동자의 스트레스, 우울, 극단적 선택 등의 심리적 건강과 정신질환에 대한 사회문제가 대두되어 2018년 10월 '감정노동자 보호법(산업안전보건법 제41조)'[5]으로 전국의 감정노동자를 보호하려는 정책이 시행되었다.



<그림 2> 감정노동자 정신건강실태 및 산재현황 [3]

대표적인 서비스 감정노동자인 고객센터 고객상담사의 경우 그림 3과 같이 과반수 이상 직영이 아닌 위탁운영이 되고 있으며, 업무에 대한 감시와 모니터링이 관리자에 의하여 높은 수준으로 유지되고 조직이 감정노동자의 심리적 건강에 대한 보호보다는 무조건적인 고객지향적 친절과 응대를 강요하고 있다.



<그림 3> 콜센터 운영형태 현황 및 협력사 비중 [3]

민주노총 공공운수노조의 '콜센터 노동자의 노동건강 실태'[6]에 의하면 고객 상담사 1300여 명의 조사 대상자 중 80% 이상이 우울증 고위험군으로 판정되어 부정적인 사회문제가 지속되고 있다. 이는 사업주의 의무사항 미이행에 대한 처분이 신고에 의한 조사로만 이루어지고 있어 감정노동자의 보호조치가 문제가 발생한 이후에 이루어지고 있는 실정이다. 따라서, 감정노동자의 심리적 건강 상태에 대한 선제적인 조치를 취할 수 있는 방안에 관심을 가질 필요가 있다. 이에, 본 논문에서는 고객 상담사의 스트레스 변화를 실시간으로 관리할 수 있는 스트레스 감지 시스템을 제안한다. 시스템은 실시간 얼굴검출 모듈, 딥러닝으로 생성된 감정분류 FER 모듈, 스트레스 수치만을 시각화하는 모니터링 모듈로 구성된다. 시스템의 설계를 위하여 먼저 기존연구에 주로 사용되었던 FER2013 data-set[7]이 아닌 한국인 감정 이미지 중심의 다양한 고품질 데이터를 확보하였다. 이 데이터를 이미지 처리 분야에서 이용되고 있는 딥러닝 기술 중의 하나인 CNN(Convolutional Neural Network)모델에 학습하여 감정분류 FER 모듈을 생성한다. 실시간 얼굴검출 모듈에서 생성된 얼굴 이미지를 FER 모듈에 입력하여 감정 상태를 분류 후 스트레스 수치로 환산하였고 모니터링 모듈에서 시각화하였다. 이 시스템을 통하여 감정노동자의 스트레스 모니터링과 정신질환 예방을 목표로 설계하였다. 본 논문의 구성은 2장에서 제안하는 시스템과 관련된 연구를 서술하고, 3장에서 제안하는 시스템 설계 방법에 대하여 기술한다. 4장에서는 실험 결과 및 테스트를 기술하고 마지막 5장에서 논문의 결론을 요약하고 향후 연구에 대한 방향성을 제시한다.

II. 관련 연구

2.1 스트레스 감지(Stress Recognition)

기존의 스트레스 감지 시스템을 위한 연구로는 먼저 Liu et al. [8]가 피부의 전기적 신호를 탐색하는 방법(EDA)을 제시하였고, Kataoka, Hisanori, et al.[9]은 피부의 온도 모니터링으로 구별하는 방법을 개발하였다. Shan et al. [10]는 호흡기 신호를 원격측정하여 스트레스를 감지하는 모델을 제안하였고, Sakri et al. [11]의 연구에서는 심박 수와 모든 피부 반응을 토대로 스트레스를 모니터링하였다. 그러나 이러한 연구는 스트레스 측정을 위하여 별도의 수치를 감지하는 시스템이 필요하다. 이는 모든 고객 상담사를 위한 시스템 구성해야하며 비용적인 측면만 고려해서도 국내 고객 상담사 근무환경에 적용하기 적합하지 않다. 최근에는 인공지능 분야에서 이용되는 딥러닝(Deep Learning)으로 감정상태를 분류하고 스트레스를 감지하는 연구가 진행되고 있다. 딥러닝(Deep Learning)이란 사람의 뉴런 연결 구조인 신경망을 빗대어 구성한 인공신경망(Artificial Neural Network)을 여러 층으로 쌓아 머신러닝을 수행하는 것을 말한다.[12] 스트레스 감지 연구에 사용되는 딥러닝은 이미지 분류 분야에서 지속적인 연구와 발전을 이루고 있는 CNN(Convolutional Neural Network)으로써[13] 사람 얼굴 이미지를 이용해서 7가지의 감정 상태(angry, disgust, fear, neutral, happy, sad, surprise)를 분류하는 얼굴 감정인식(Facial Expression Recognition)에 기반한다. 이 감정 상태를 Stress와 Not-Stress로 분류하는 연구가 활발히 진행되어 왔다. ZHANG, Jin, et al.[14]의 연구에서는 13단계의 얇은 CNN 모델을 구성하여 그림 4와 같이 CK+, Oulu-CASIA 그리고 KMU-FED 데이터셋을 기반으로 부정적 감정 스트레스 검출 시스템을 설계하였다.

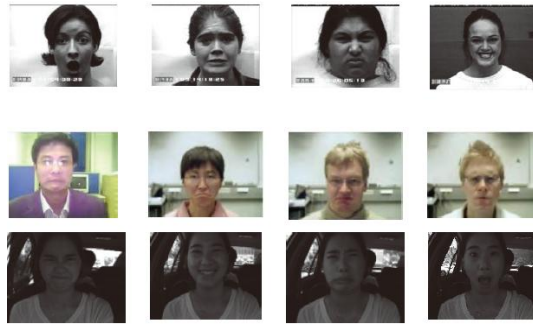


Figure 4. Examples of the datasets used in our experiments. Top CK+, Middle Oulu-CASIA, Bottom KMU-FED.

<그림 4> CK+, Oulu-CASIA, KMU-FED 데이터셋[14]

Almeida, J., & Rodrigues, F.[15]은 CK+, KDEF, Net Images 데이터셋을 ILSVRC의 이미지 분류 모델인 VGG16, VGG19, InceptionResNetV2에 학습하여 angry, disgust, fear를 Stress 감정으로 정의하고 neutral, happy, sad, surprise 감정을 Not-Stress로 이진 분류하는 모듈을 제시하였다. 이때, 이미지 분류 모델의 분류(Classification)파트를 2가지로 구성하여 총 6개의 모델을 실험하였고, 약 5,200개의 부족한 데이터셋에서 발생할 수 있는 과적합(Over-Fitting)을 방지하기 위하여 Data Augmentation 기법을 사용하였다. 마찬가지로 CHIWANDE, Sujata, et al.[16]의 연구에서도 7가지 감정 상태를 Stress와 Not-Stress로 분류하는 모델을 Facial Expression Recognition Challenge[7]의 데이터셋인 Kaggle FER2013 dataset을 사용하여 설계하는 등 다양한 데이터셋을 통한 스트레스 감지 모델이 제안되고 있다. 그러나, 대부분의 연구는 서양 인종의 데이터로만 구성되거나 아시아인 데이터가 소규모 포함된 데이터셋을 중심으로 진행되었다. Michel, Caldara, & Rossion(2006)의 연구에 따르면[17] 인간은 서로 다른 인종의 얼굴보다 같은 인종의 얼굴을 더 정확하고 빠르게 처리한다. 따라서, 국내 고객 상담사의 스트레스 감지를 위해서는 정제된 한국인 데이터셋의 확보를 고려해야 한다. [15]과 [17]에서 영감을 받아 본 논문에서는 CNN기반의 이미지 분류 모델에 한국인 이미지 데이

터 중심의 데이터셋에 Data Augmentation 기법을 적용하고 다양한 방법으로 딥러닝 학습한다. 이를 통하여 실시간으로 얼굴감정을 분류할 수 있고, 스트레스와 관련된 감정 상태를 감지하여 수치화 및 모니터링 할 수 있는 시스템을 설계한다. 딥러닝 코드를 구현하기 위한 라이브러리는 텐서플로우(TensorFlow), 케라스(Keras)를 사용한다.



2.2 CNN(Convolutional Neural Network)

CNN(Convolutional Neural Network)은 이미지 분석을 위한 패턴을 찾는 데 유용한 알고리즘이다. 이미지의 공간정보를 유지한 상태로 신경망 학습이 가능토록 한 형태로 LECUN, Yann, et al.(1988)[18]에 의하여 현재의 CNN 형태가 제안되었다. 컨볼루션(Convolution)은 입력 데이터의 특징을 추출하는(Features Mapping) 요소로 각 채널마다(RGB 3채널) 정사각 행렬의 특징 필터(Filter, Kernel)가 지정 간격(Stride)으로 이동하며 입력데이터와 컨볼루션 연산을 통해 특징을 추출한다. 이후 채널의 수를 유지하고 특징을 강조하며 출력데이터의 크기를 줄이는 방법으로 풀링(Pooling) 레이어를 적용한다. CNN은 그림6과 같이 컨볼루션 레이어와 풀링 레이어를 쌓은 특징추출(Feature Extraction) 파트와 Fully Connected Layer로 구성된 이미지 분류(Classification)파트로 구성된다. 즉, 컨볼루션 레이어와 풀링 레이어를 반복하여 특징을 추출하고, Fully Connected Layer에서 최종 분류하는 알고리즘이다.

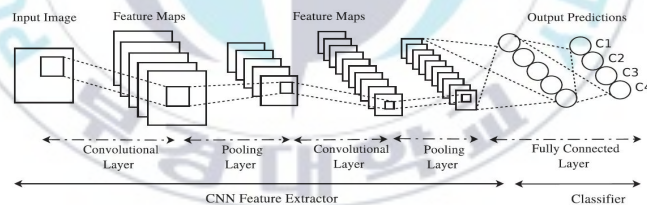
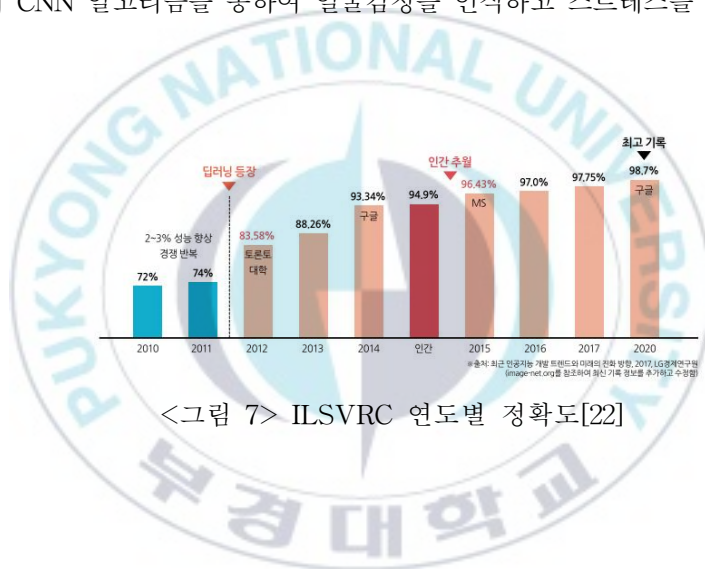


Fig. 4. Architecture of a standard CNN.

<그림 5> 스탠다드 CNN의 구조[19]

2.3 이미지 분류 CNN 알고리즘

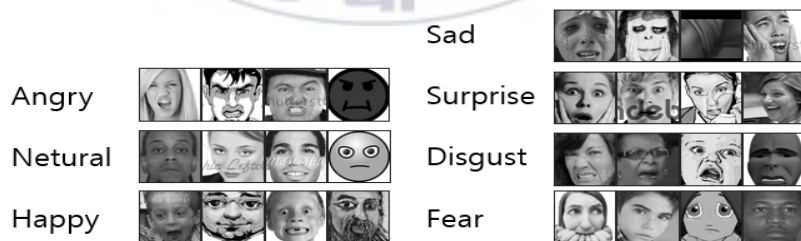
ImageNet ILSVRC(Large Scale Visual Recognition Challenge)[20]는 대규모 물체감지와 이미지 분류 알고리즘 평가 대회로써 2012년 SuperVision팀의 AlexNet을 시작으로 CNN기반 알고리즘이 좋은 성적을 차지하고 있다. 2015년 MSRA팀(Microsoft Research)의 ResNet은 분류오차율(Classification Error)을 3.57%를 기록하며[21] 인간 평균 분류정확도(94.9%)를 추월, 2020년에 98.7%까지 발전하였다.[22] 본 논문에서는 2014년 우승한 구글의 GoogLeNet의 단점을 보완하고 성능을 향상시킨 3가지의 CNN 알고리즘을 통하여 얼굴감정을 인식하고 스트레스를 분류한다.



<그림 7> ILSVRC 연도별 정확도[22]

2.4 얼굴감정인식(FER:Facial Expression Recognition)

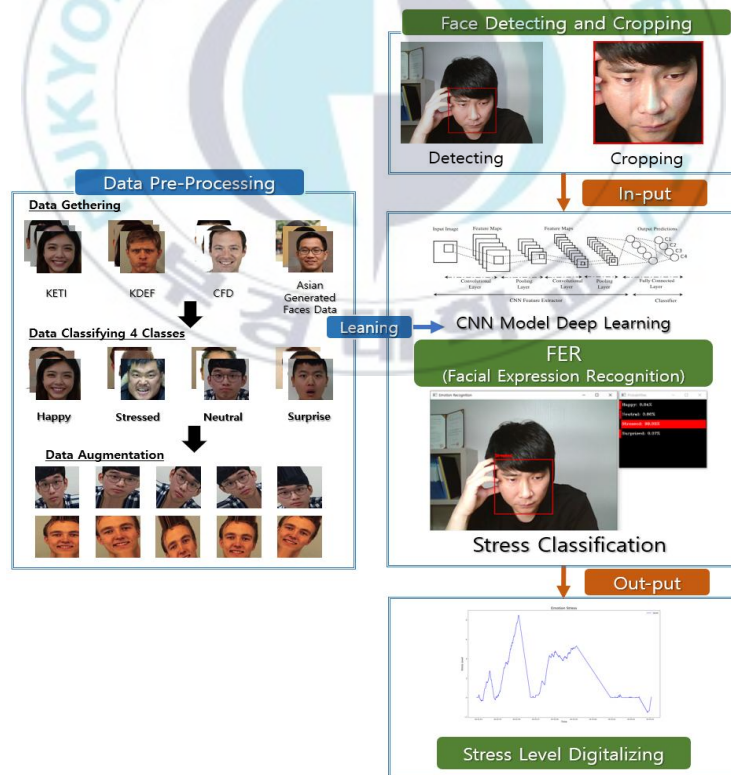
CNN을 기반으로 얼굴 표정으로부터 감정을 인식하는 연구는 얼굴의 외관적 특징을 추출하고, 얼굴의 랜드마크 포인트에서 기하학적 특징을 수집하여 두 가지의 정보를 조인트 파인튜닝(Joint Fine-Tuning)하는 기법이 있으며[23], ImageNet ILSVRC[13]에서 성능이 평가된 CNN 이미지 분류 모델의 아키텍처를 확보된 데이터셋에 맞게 변형하고 학습을 업데이트하는 기법[24]이 대표적이다. 후자의 기법은 이미지 분류 문제에서는 좋은 성능을 보이나, 감정인식에서는 검증되지 않은 경우가 있으며 데이터셋의 정확도에 영향을 많이 받는다. 얼굴 감정인식에 사용되는 대표적인 데이터 셋인 Kaggle FER2013 dataset은 2013년 개최된 Facial Expression Recognition Challenge[7]의 데이터베이스로써, 48x48 픽셀의 총 7가지의 표정 이미지가 구분된 35,887개의 이미지를 포함한다. 그러나, 해상도가 매우 낮으며, 그림 8과 같이 레이블링 오류, 워터마크 삽입, 그림, 애니메이션 등 불필요하거나 잘못 분류된 이미지를 포함하고 있다. 따라서, 후자의 기법을 이용한 얼굴 감정인식 FER 모듈의 생성 및 정확도 향상을 위해서는 정제된 고품질의 데이터를 통하여 여러 개의 CNN 이미지 분류모델을 실험할 필요가 있다.



<그림 7> 이미지 오류를 포함한 Kaggle FER2013 dataset 일부

Ⅲ. 본론

본 논문에서 제안하는 딥러닝 얼굴 감정인식(FER) 기반 스트레스 감지 시스템의 전반적인 프로세스는 그림 9와 같다. 먼저 CNN 모델에 학습할 데이터셋을 연구의 취지에 맞도록 확보하고 전처리한 후, 학습하여 FER 모듈을 생성한다. 이후 얼굴검출 (Face Detection and Cropping) 모듈을 사용하여 실시간 이미지를 추출하고, 학습된 FER 모듈을 통해 4가지의 감정으로 분류한다. 분류된 감정 상태별로 스트레스 수치를 부여하고 누적 연산하여 그 변화를 시각화한다. 3가지의 CNN 모델을 다양한 방법으로 학습하여 양질의 결과를 도출할 수 있도록 하였다.



<그림 8> 스트레스 감지 시스템 설계 프로세스 전반

3.1 학습 데이터셋 구성

Michel, Caldara, & Rossion(2006)의 연구에 따르면[17], 다른 인종의 얼굴보다 같은 인종의 얼굴을 처리할 때 얼굴 전체와 특정 특징에 대한 처리 응답시간이 더 빠르고 정확도가 더 높다.

<표 1> 백인/아시아인의 인종별 평균 얼굴인식 응답시간[17]

	<i>Caucasian participants</i>	<i>Asian participants</i>
Caucasian faces		
Whole faces	1071 (65.19)	1122 (37.15)
Isolated features	1159 (57.02)	1268 (39.89)
Asian faces		
Whole faces	1108 (63.29)	1064 (35.60)
Isolated features	1152 (53.04)	1156 (40.70)

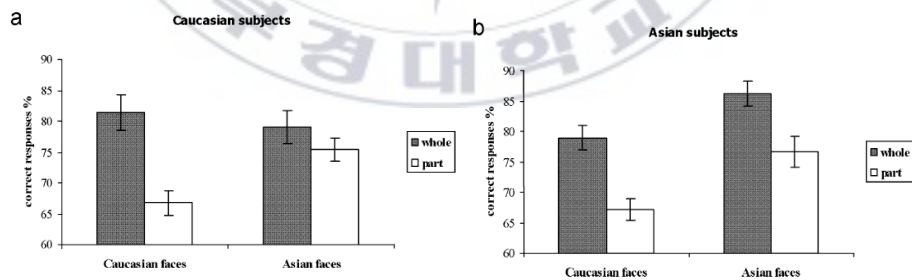
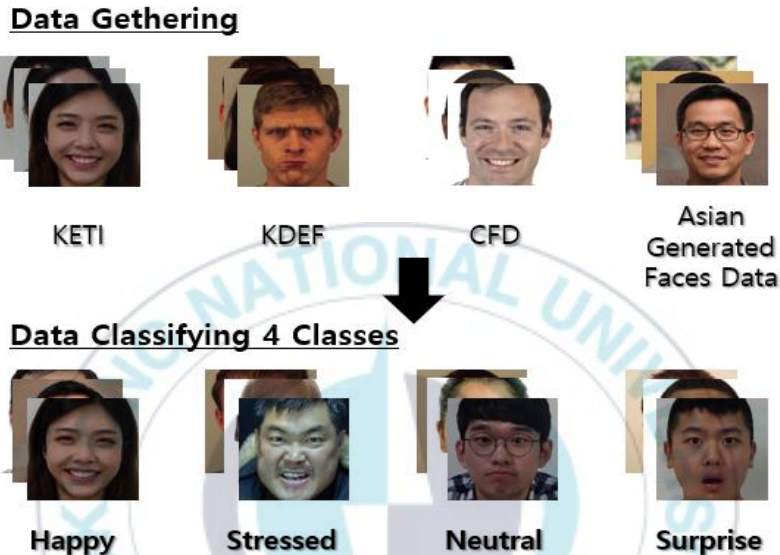


Figure 3. Recognition accuracy of (a) Caucasian and (b) Asian participants for whole faces and isolated features in the 2AFC matching test. Error bars represent standard errors of the mean.

<그림 9> 백인/아시아인의 인종별 평균 얼굴인식 정확도[17]

따라서, 국내 고객 상담사의 얼굴 감정인식 정확도 확보를 위해 한국인 및 아시아 중심의 데이터를 중심으로 고해상도 데이터를 확보하였다. 학습 데이터셋 구성을 위한 전반적인 프로세스는 그림 11과 같다.



<그림 10> 데이터 셋 구성 프로세스 전반

FER 모델을 위한 데이터셋 구성은 한국인 감정분류 데이터셋(KETI)[25]을 중심으로 Asian Generated Faces Dataset(AGFD)[26]으로 아시아인의 데이터를 추가하였고, Karolinska Directed Emotional Faces(KDEF)[27], Chicago Face Database 3종 (CFD, CFD-MR, CFD-INDIA) [28-30]을 활용하여 동서양의 다양한 얼굴 감정 데이터를 사용하였다.

- ① 한국인 감정분류 데이터셋 KETI R&D 데이터 : 한국전자기술연구원 지능정보 플래그십 R&D데이터로써 20세에서 46세까지의 연기 지망생 및 전문가 100명이 angry, fear, disgust, neutral, happy, sad, surprise 7가지 감정을 약 100번씩 발화/연기를 수행한 4,906개의 이미지로 구성된 데이터

- ② Asian Generated Faces Dataset(AGFD) : 남녀/어린이/성인/노인 등으로 분류된 아시아인 이미지로써 인공지능 GAN(Generative Adversarial Network)으로 생성된 데이터
- ③ Chicago Face Database(CFD, CFD-MR, CFD-INDIA) : 다양한 인종 597명의 이미지로 구성된 CFD, 88명의 추가데이터 CFD-MR, 142명의 인도인 데이터 CFD-INDIA 3가지를 통합한 데이터로써 4가지 감정(중립, 화남, 행복함(입 벌리고), 행복함(입 다물고), 두려움)이 포함됨
- ④ Karolinska Directed Emotional Faces(KDEF) : 20세에서 30세 사이의 아마추어 배우 70명을 대상으로 KETI R&D데이터와 같은 7가지 감정에 대하여 5가지의 다른 각도에서 촬영한 이미지 데이터

<표 2> 얼굴 감정 데이터셋 종류별 이미지 개수

KETI	AGFD	CFD	KDEF	Total
4,683	539	1,437	2,738	9,397

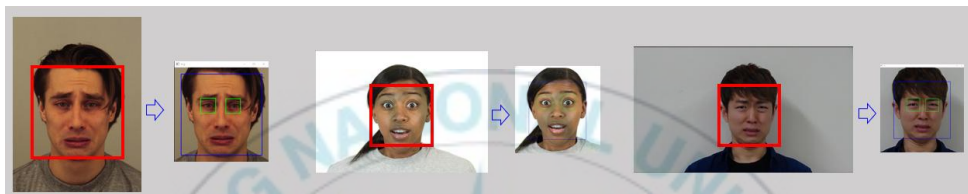
각 데이터셋 별 최대 angry, fear, disgust, neutral, happy, sad, surprise 7개의 감정분류 구성에서 본 연구에 맞도록 angry, disgust, sad는 Stress로 통합, fear와 surprise는 Surprise로 통합하였다. 결과적으로 4개 감정(Happy, Neutral, Stress, Surprise)으로 분류하고 불필요하거나 모호한 이미지 데이터를 정제하여 총 9,397개의 데이터셋을 구성하였다. 이 데이터셋에서 80%는 학습용으로, 20%는 학습 중 평가용으로 사용된다.

<표 3> 얼굴 감정 분류별 이미지 데이터 개수

Class	Happy	Neutral	Stress	Surprise	Total
DataSet	1,937	2,812	2,787	1,861	9,397

3.2 데이터 전처리 (Data Pre-Processing)

CNN 모델의 학습 효율성을 위해 학습 데이터에서 감정이 표현된 영역만을 추출하는 전처리가 필요하다. 3.3에서 서술할 얼굴검출(Face Detection and Cropping) 모듈의 Haar Cascade[31] 알고리즘을 이용하여 그림 12와 같이 얼굴 영역만을 추출하는 전처리를 수행한다.



<그림 11> 얼굴영역 검출 및 자르기 - 데이터 전처리

전처리를 마친 데이터셋은 FER 모듈의 분류성능에 문제를 일으키는 과적합 문제를 방지하기 위하여 학습데이터를 증가시킬 필요가 있다. 학습 이미지를 증가시키는 방법으로 먼저 데이터의 총 개수는 같으나, Batch 단위마다 새로 생성된 이미지를 입력하여 학습시키는 방법과 이미지 증강기법으로 총 데이터의 개수를 늘려 학습시키는 두 가지의 방법으로 나뉜다. 제안하는 시스템에서는 전자의 방법으로 코드 1과 같이 Keras에서 제공하는 이미지 증강 (Data Augmentation) 기법을 적용하였다.


```

from tensorflow.keras.preprocessing.image import
ImageDataGenerator

data_generator = ImageDataGenerator(
    rotation_range=30, width_shift_range=0.2,
    height_shift_range=0.2, rescale=1./255,
    shear_range=0.2, zoom_range=0.2,
    horizontal_flip=True, fill_mode='nearest')

```

<코드 1>

사용된 이미지 증강 기법은 30도 rotation, width shift, height shift, shear, zoom, horizontal_flip이며 이미지의 변경된 부분을 채우는 fill_mode는 nearest로 적용하고 RGB 3채널의 값을 255로 나누어 1 이하의 값으로 연산할 수 있도록 설정하였다. 마지막으로 데이터의 Shape는 100X100, 3채널로 구성하였다.

3.3 얼굴검출(Face Detection and Cropping) 모듈

감정 상태를 분류를 위해 FER 모듈에 입력되는 이미지를 추출하기 위한 단계로써 (P Viola, M Jones, 2001)에 의해 제안된 Haar Cascade[31] 알고리즘을 이용한다. 이 알고리즘은 특정 Feature에는 명암의 차이가 존재하는 것을 바탕으로 한다. 이미지에 위치를 이동하는 인접한 직사각형의 영역 내에 있는 밝기의 합차이를 통하여 특징을 찾아내고 Feature를 생성한다. 이후 Cascade Classifier로 얼굴을 검출하는데, 얼굴이 없는 영역에서는 짧은 시간에 판단을 하게 되며 얼굴 영역이라는 판단이 내려지면 현재의 위치에서 더 오래 걸리는 연산을 수행한다. 본 논문에서는 OpenCv에서 제공되는 Haar Cascade 모듈에 Web Cam에서 생성한 이미지 입력으로 사용한다. 코드 2 알고리즘은 먼저 얼굴을 검출하여 영역을 지정하고 3채널 컬러 이미지를 유지한 채 잘라낸 후 이

를 감정분류 FER 모듈에 입력하기 위하여 100X100으로 리사이징하고 255로 나누어 Shape를 일치시킨다. 이후 이미지를 전처리하여 FER 모델에 입력한다.

```
import cv2
import numpy as np
from tensorflow.keras.preprocessing.image import
img_to_array
face_croiping =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
webCam = cv2.VideoCapture(0)
faceImage = camera.read()
# Face detection
face = face_croiping.detectMultiScale(frame, 1.4, 5)
# Resize the image
resizedImage = frame[fY:fY + fH, fX:fX + fW]
resizedImage = cv2.resize(resizedImage, (100, 100))
resizedImage = resizedImage.astype("float") / 255.0
resizedImage = img_to_array(resizedImage)
resizedImage = np.expand_dims(resizedImage, axis=0)
# Emotion predict
predictEmotion = ferModule.predict(resizedImage)[0]
```

<코드 2>

3.4 Facial Expression Recognition(FER)모델

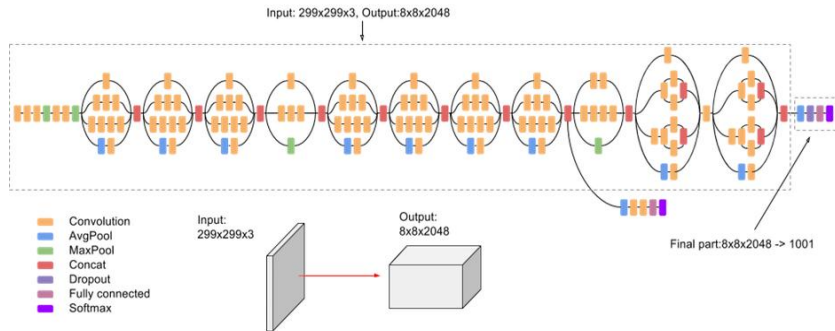
감정분류를 위한 Facial Expression Recognition 모델은 실시간 스트레스 측정에 적합해야 하며 충분한 정확도를 확보해야 한다. 따라서 ImageNet ILSVRC[20] 대회에 참가하여 검증된 CNN 기반 이미지 분류 알고리즘 모델 중에서 2015년 참가한 InceptionV3(Top-1 Acc:78.8%)[32], 2016년 참가한 Xception(Top-1 Acc:79%)[33], 그리고 2018년 참가한 MobileNetV2(Top-1 Acc:74.7%)[34], 총 3개 모델을 선정하였다.

<표 4> 3가지 CNN 모델의 Top-1 Accuracy 및 파라미터 수

Model	Top-1 Accuracy	Parameters
InceptionV3	78.8 %	23,851,784
Xception	79 %	22,910,480
MobileNetV2	74.7 %	3,538,984

가. InceptionV3

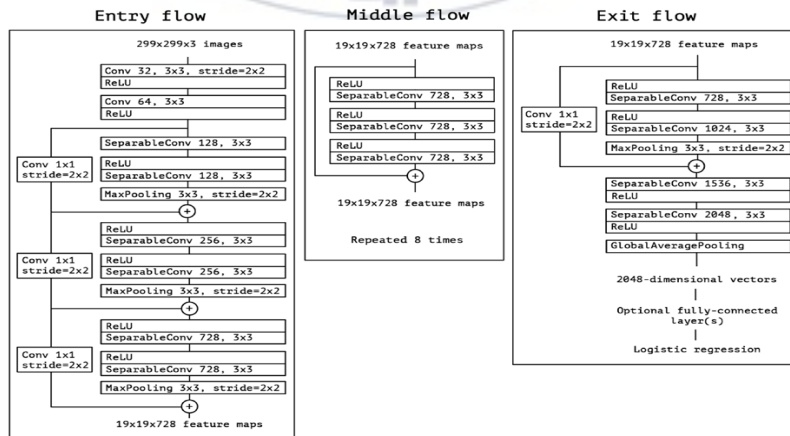
InceptionV3[32]는 2014년 ImageNet ILSVRC[20] 대회에서 우승한 GoogLeNet(InceptionV1)이 복잡한 구조로 인하여 다양한 변형이 힘들다는 점에서 Inception 모델의 연산량을 줄이는 최적화에 대한 방향성 제시를 통하여 소개되었다. 기존 모델의 큰 Filter를 3x3 Convolution으로 Factorizing convolution을 적용하여 기존 Inception 대비 구조적으로 유사하지만, Network의 깊이가 깊어지고 연산량이 줄어들었다. InceptionV3는 V2와 동일한 구조에서 Optimizer 변경, Label Smoothing, Factorized 7-7, BN-auxiliary를 적용하여 성능을 높인 모델이다.



<그림 12> InceptionV3 모델의 아키텍처[35]

나. Xception

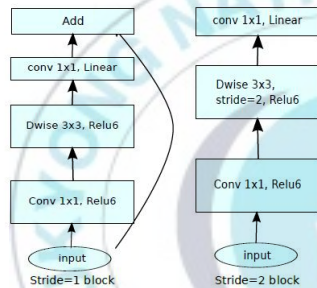
Xception[33]모델은 총 14개의 모듈로 36개의 컨볼루션 레이어가 존재하며 residual connection을 사용한다. 특히 Inception 모델 대신 작동방식은 비슷하지만, 연산량이 감소하는 Depthwise Separable Convolution을 수정하여 사용하는 Extreme Inception 모델이다. 이를 통해 더 효과적으로 cross-channel correlations와 spatial correlations를 구분하여 계산할 수 있어 Inception 모델 대비 적은 연산 처리가 가능하다.



<그림 13> Xception 모델의 아키텍처[33]

다. MobileNetV2

앞서 소개한 Xception모델에서 수정된 Depthwise Separable Convolution을 모든 Convolution에 대체하여 경량화에 집중한 모델이 MobileNetV1이며 이 Depthwise Separable Convolution을 Inverted Residuals와 Bottlenecks으로 수정한 Convolution Block을 사용한 모델이 MobileNetV2이다.[34] 전체적인 Convolution 채널은 줄이고 block 내부에서만 증가 시켜 연산량과 파라미터를 줄였다. 마지막으로 ReLU6를 활성화 함수로 사용하여 연산속도를 획기적으로 줄이면서 MobileNetV1보다 좋은 성능을 확보한 모델이다.



(d) Mobilenet V2

Network	Top 1	Params	MAdds	CPU
MobileNetV1	70.6	4.2M	575M	113ms
ShuffleNet (1.5)	71.5	3.4M	292M	-
ShuffleNet (x2)	73.7	5.4M	524M	-
NasNet-A	74.0	5.3M	564M	183ms
MobileNetV2	72.0	3.4M	300M	75ms
MobileNetV2 (1.4)	74.7	6.9M	585M	143ms

Table 4: Performance on ImageNet, comparison for different networks. As is common practice for ops, we count the total number of Multiply-Adds. In the last column we report running time in milliseconds (ms) for a single large core of the Google Pixel 1 phone (using TF-Lite). We do not report ShuffleNet numbers as efficient group convolutions and shuffling are not yet supported.

<그림 14> MobileNetV2 모델의 아키텍처 및 퍼포먼스[34]

3가지의 모델이 스트레스 측정에 충분한 정확도를 가질 수 있도록 Fine Tuning 기법을 적용하여 데이터셋을 학습한다. Fine Tuning이란 사전 학습된 Pre-Trained 모델을 활용하여 새로운 모델을 학습하면 더 적은 데이터로도 매우 높은 수준의 결과를 얻을 수 있는 전이학습(Transfer Learning)[36] 기법이다. Fine Tuning은 일반적으로 전체 모델을 새로 학습시키는 방법, Convolution Base는 고정하고 최종 분류를 담당하는 Classifier만 새로 학습시키는 방법, Convolution Base의 일부와 Classifier만 새로 학습시키는 방법 3가지로 나뉜다.

```
##### InceptionV3 #####
```

```
from tensorflow.keras.applications.inception_v3 import
```

```
InceptionV3
```

```
##### Xception #####
```

```
from tensorflow.keras.applications.xception import Xception
```

```
##### MobileNetV2 #####
```

```
from tensorflow.keras.applications.mobilenet_v2 import
```

```
MobileNetV2
```

<코드 3>

본 실험에서는 3가지의 기법을 포함해 코드 3과 같이 텐서플로우에서 제공하는 InceptionV3, Xception, MobilNetV2 모델을 이용하고 다음에서 설명하는 항목들을 변형하여 Fine Tuning 하였다.

- ① Pre-Trained Model Trainable : 사전학습된 모델의 레이어 전체학습(All), Classifier만 학습(None), 일부 학습(63 / 36 layer False)하는 Fine Tuning 기법을 적용하여 테스트한다.

```
for l in transfer_model.layers:
```

```
    l.trainable = False ## None Classifier만 학습
```

```
    l.trainable = True ## All 전체학습
```

```
for l in transfer_model.layers[:36]:
```

```
    l.trainable = False ## 36 layer False 일부학습
```

<코드 4>

```
for l in transfer_model.layers[:63]:
```

```
    l.trainable = False ## 63 layer False 일부학습
```

- ② Classify Layer : 최종 분류를 담당하는 Classifier 레이어로써 각 CNN 모델의 Out-Put을 바로 Classifier의 In-Put으로 입력하는 방법(Classify Layer #1)을 테스트한다. 그리고 과적합(Over-fitting)방지를 위해 Drop-Out 레이어를 추가하여 네트워크의 일부를 학습에서 배제한 후[37] Classifier의 In-Put으로 입력하는 방법(Classify Layer #2)을 테스트한다.

```
##### Classify Layer #1 #####
```

```
gap=tf.keras.layers.GlobalAveragePooling2D()(transfer_model.  
output)  
last_layer=tf.keras.layers.Dense(4, activation="softmax")(gap)  
model=tf.keras.Model(transfer_model.input, last_layer)
```

```
##### Classify Layer #2 #####
```

```
gap=tf.keras.layers.GlobalAveragePooling2D()(transfer_model.  
output)  
layer1 = tf.keras.layers.Flatten()(gap)  
layer2 = tf.keras.layers.Dense(1024, activation='relu')(layer1)  
drop1 = tf.keras.layers.Dropout(0.3)(layer2)  
last_layer=tf.keras.layers.Dense(4,activation="softmax")(drop1)  
model=tf.keras.Model(transfer_model.input, last_layer)
```

<코드 5>

- ③ Optimizer : 손실함수(Loss Function)의 최소화를 위해서 가중치와 편향을 갱신하는 방법이 Optimizer이다. 텐서플로우(TensorFlow)에서 제공하는 인기 있는 Adam Optimizer와 SGD Optimizer를 사용하여 테스트한다.

- ④ Learning Rate : Learning Rate란 CNN 모델이 한번 학습할 때의 학습량을 의미하는 Optimizer의 하이퍼 파라미터이다. 일반적으로 이 값이 크면 손실(loss) 값을 빠르게 내릴 수 있지만, 최적의 학습이 어렵고, 작으면 소요 시간이 오래 걸리게 된다. 따라서 Optimizer별로 Learning Rate를 0.01, 0.001, 0.0001 3가지로 분류하여 테스트한다.

```
learningRate = 0.001 # Adam 0,001, 0.0001 / SGD 0.01, 0.1
```

```
optimizer = Adam(lr=learningRate)
```

<코드 6>

```
optimizer = SGD(lr=learningRate)
```

마지막으로 학습 로그를 확보하기 위하여 텐서보드를 사용하고 모델 컴파일 및 100번의 epoch를 지정하여 학습을 실행한다.

```
tensorboard = TensorBoard(
```

```
log_dir=save_name+'_log/{}'.format(testlog))
```

```
model.compile( optimizer=optimizer,
```

```
loss='categorical_crossentropy', metrics=['accuracy'])
```

<코드 7>

```
model.fit_generator(train_data, epochs=100,
```

```
validation_data=val_data, callbacks=[tensorboard])
```

3가지 CNN 모델에 Fine Tuning 기법을 적용하기 위하여 각종 파라미터와 설정값을 달리 지정하여 총 29번의 학습모델 테스트를 진행한다. 표 5는 테스트 번호 12번까지 총 29번의 테스트에 대한 학습모델별 파라미터와 설정값을 나타낸다.

<표 5> FER 모듈 생성을 위한 CNN 모델 Transfer Learning 테스트

Test	Model	Learning Rate	Optimizer	Classfiy layer	Pre-trained Model Trainable	
1	InceptionV3 Xception MobileNetV2	0.0001	Adam	Classify layer #1	All	
2					None	
3		0.001		Adam	Classify layer #2	All
4						None
5		0.0001	Adam		Classify layer #2	All
6						None
7		0.001		Adam	Classify layer #2	All
8						None
9	Xception MobileNetV2	0.01	SDG		Classify layer #1	All
10	InceptionV3				Classify layer #2	All
11	Xception	0.0001	Adam	Classify layer #1	63 layer False	
12	Xception				36 layer False	
Classify layer #1 : Model output - Classify Layer - Output						
Classify layer #2 : Model output - Dense - Dropout - Classify Layer - Output						

3.5 스트레스 데이터 수치화(Digitalize) 모듈

FER 모델로부터 현재의 감정 상태를 분류할 때 Web Cam으로부터 실시간 이미지 데이터를 생성한다. 그러나, 회사의 업무감시 및 개인정보보호의 이슈로 이미지의 출력 및 저장 없이 감정분류 결과에 한하여 표 6의 조건에 따라 스트레스 수치 환산 후 코드 8의 알고리즘으로 누적 연산한다.

<표 6> 스트레스 데이터 수치화 및 연산기준

Emotional Status	Current Stress Level	Figure	Calculation
Happy	> 0	- 0.02	Reduction
	≤ 0	- 0.005	Reduction
Neutral	> 0	- 0.005	Reduction
	≤ 0	-	No Change
Stress	All	+ 0.02	Increase
Surprise	All	+ 0.005	Increase

```

if label== "Happy" :
    if level >0: level = level-0.02
    else: level = level-0.005
elif label== "Neutral" :
    if level >0: level = level-0.01
    else: level = level

elif label== "Stressed" : level = level+0.02
elif label== "Surprised" : level = level+0.005

data={'Time':[currentTime],'Emotion':[label],'Level':[level]}
dataframe = pd.DataFrame(data)

if not os.path.exists(save_file):
    dataframe.to_csv('./testLog/test.csv', mode='w')
else: dataframe.to_csv('./testLog/test.csv', mode='a')

```

<코드 8>

연산된 스트레스 수치는 시간 정보와 함께 Pandas 라이브러리를 이용하여 csv파일에 저장하고 관리자에 의한 모니터링을 위하여 그래프로 시각화한다.

```
save_file = './testLog/test.csv'
stress = pd.read_csv(save_file, index_col=0, engine='python')
stress.index = pd.to_datetime(stress.index)
fig = plt.figure(figsize=(20, 12))
ax = fig.add_subplot(1,1,1)
ax.plot(stress['Level'], label='Level', color='b')
ax.set_title('Emotion Stress', fontsize=20)
ax.set_ylabel('Stress Level', fontsize=14)
ax.set_xlabel('Time', fontsize=14)
ax.legend(fontsize=12, loc='best')
```

<코드 9>

Ⅳ. 실험결과 및 테스트

4.1 FER 모듈 실험결과

스트레스 감지 시스템의 핵심인 FER 모듈 생성을 위한 CNN 모델의 실험 결과 데이터는 아래 표7과 같다.

<표 7> FER 모듈 생성을 위한 CNN 모델 테스트 결과

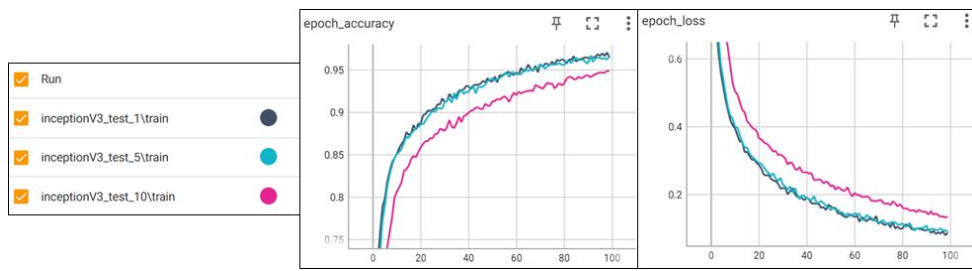
Test #	Model	Accuracy	Loss
1	InceptionV3	97.1 %	0.077
	MobileNetV2	97.5 %	0.064
	Xception	98.6 %	0.043
2	InceptionV3	54.1 %	1.077
	MobileNetV2	61.8 %	0.917
	Xception	62.7 %	0.907
3	InceptionV3	91.5 %	0.209
	MobileNetV2	87.2 %	0.707
	Xception	96.2 %	0.106
4	InceptionV3	53.8 %	1.095
	MobileNetV2	62.9 %	0.888
	Xception	64.7 %	0.853
5	InceptionV3	97.3 %	0.067
	MobileNetV2	96.7 %	0.091
	Xception	98.4 %	0.045
6	InceptionV3	60.4 %	0.950
	MobileNetV2	70.1 %	0.732
	Xception	69.6 %	0.742
7	InceptionV3	84.7 %	0.400
	MobileNetV2	87.1 %	0.665
	Xception	97.1 %	0.085
8	InceptionV3	57.2 %	0.998
	MobileNetV2	68.4 %	0.772
	Xception	68.9 %	0.736
9	Xception	95.4 %	0.121
	MobileNetV2	96.3 %	0.090
10	InceptionV3	95.3 %	0.122
11	Xception	94.1 %	0.152
12	Xception	96.7 %	0.086

총 29번의 학습 과정에서 사전 학습된 3개의 CNN 모델을 Classifier만 학습시킨 Test # 2, 4, 6, 8의 경우 최대 Accuracy가 70.1%로 매우 저조한 결과를 보였다. 해당 테스트를 제외한 나머지 결과 중에서 3개의 CNN 모델별로 Top-3 Accuracy를 선정하였다.

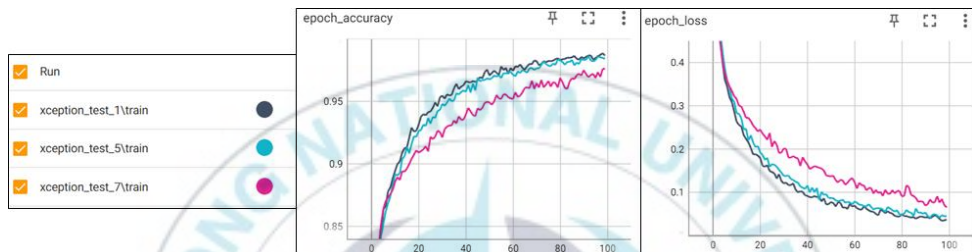
<표 8> CNN 모델 테스트 결과 중 모델별 Top 3 Accuracy

Test #	Model	Top-3	Accuracy	Loss
Test #5	InceptionV3	1	97.3%	0.067
Test #1		2	97.1%	0.077
Test #10		3	95.3%	0.122
Test #1	Xception	1	98.6%	0.043
Test #5		2	98.4%	0.045
Test #7		3	97.1%	0.085
Test #1	MobileNetV2	1	97.5%	0.064
Test #5		2	96.7%	0.091
Test #9		3	96.3%	0.090

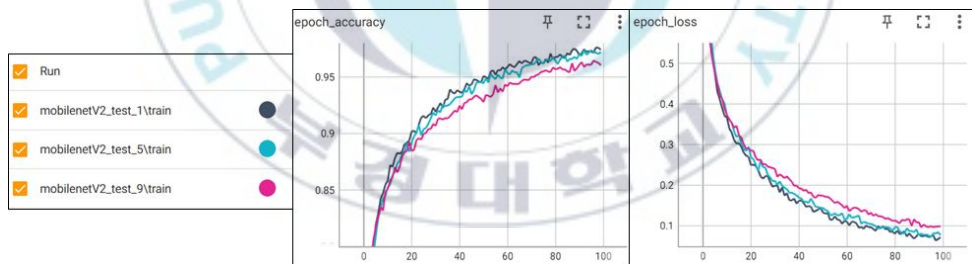
그림 16, 17, 18은 epoch 100으로 진행한 각 CNN 모델의 테스트 로그로써 정확도(Accuracy)와 손실률(Loss)의 학습 그래프이다. Accuracy의 경우 epoch 60회를 전후로 95% 이상으로 상승하고 기울기가 완만해지며 Xception 모델의 경우 최대 98.6%를 기록하였다. Loss의 경우 정확도와 반비례의 그래프를 보여준다. epoch 40회 부근에서 0.2 이하로 낮아지며 최저 0.043을 기록하였다.



<그림 15> InceptionV3 모델의 딥러닝 학습 로그 - Accuracy, Loss



<그림 16> Xception 모델의 딥러닝 학습 로그 - Accuracy, Loss



<그림 17> MobileNetV2 모델의 딥러닝 학습 로그 - Accuracy, Loss

CNN 모델의 학습 결과가 Accuracy가 95% 이상인 경우 스트레스 감지 시스템의 FER 모듈로 선정하기에 적합할 수 있다. 그러나, 앞서 구성한 데이터셋에 대한 과도한 학습으로 실제 데이터에 대한 인식오류가 증가하는 과적합 (Overfitting)이 발생할 수 있다. 따라서 선정된 9개 모델의 실시간 테스트로 과적합 여부를 확인하고 실제 인식률을 평가하여 최종 FER 모델을 선정한다.

4.2 실시간 스트레스 감지 테스트

FER 모듈의 실제 인식률을 확인하기 위하여 Windows10, AMD Ryzen7 4800H CPU 환경에서 실시간 테스트를 진행하였다. 본 시스템의 얼굴검출 모듈이 Haar Cascade 알고리즘을 기반으로 설계되어 얼굴 영역 간의 밝기 차이가 명확하지 않은 상황에서는 신뢰성 있는 테스트 결과를 확보할 수 없다. 따라서 얼굴 영역 간의 밝기차이를 확보를 위해 테스트용 조명을 추가하였다. 실시간 스트레스 감지 테스트에는 30대 남녀 각 3명씩 총 6명을 대상으로 진행하였으며, 표 9와 같이 테스트를 위한 감정표현을 통일하였다.

<표 9> FER 모듈 인식률 테스트를 위한 표정 기준

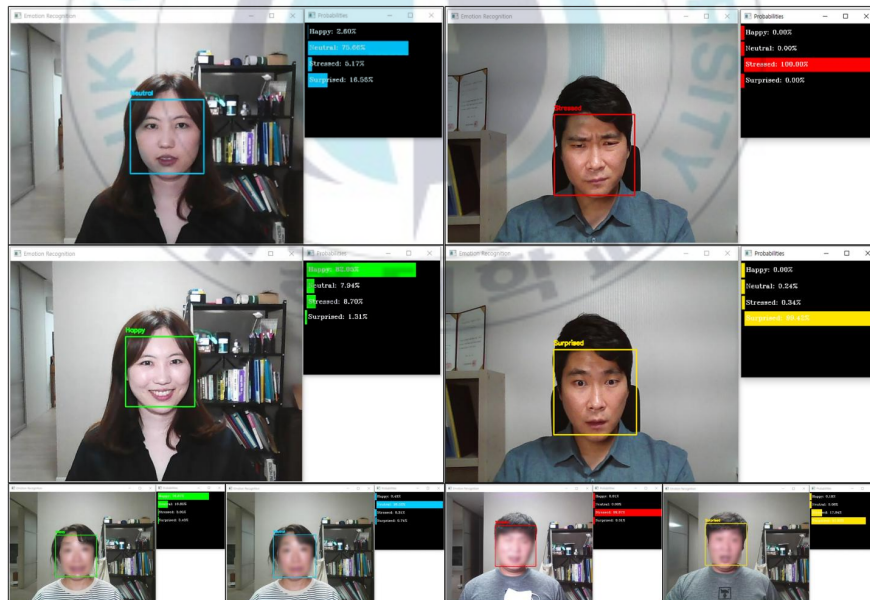
Emotion	Facial Expression
Happy	미소, 입 열고 웃음, 함박웃음
Neutral	무표정, 초점 흐리기(멍하기)
Stress	찡그리기, 입 열고 찡그리기, 울기, 완전분노(욕설)
Surprise	눈만 놀라기, 입/눈 동시에 놀라기

신뢰성 있는 테스트 데이터 확보를 위하여 명확한 인식률 판단기준이 필요하다. 표 10은 각 피실험자가 특정 표정으로 테스트를 진행할 때 FER 모듈이 해당 표정에 출력하는 인식률을 기준으로 S부터 D까지 등급화한 판단 기준이다. S와 A는 해당 표정을 검출했다고 판단할 수 있는 등급으로써 + 점수를 부여하였고, B등급 이하의 경우 해당 표정을 검출하지 못했다고 판단되므로 0 또는 - 점수를 부여하였다.

<표 10> FER 모듈 인식률 테스트 정확도 판단 기준

Grade	Score	Recognition accuracy
S	10	over 90% on correct Emotion
A	7	over 50% on correct Emotion
B	0	Oscillating between correct and Incorrect Emotion
C	-10	over 50% on Incorrect Emotion
D	-20	over 90% on Incorrect Emotion

실제 시스템에서는 검출한 얼굴 이미지를 시각화하거나 저장하지 않는다. 여기에서는 실시간 테스트를 위하여 Happy-초록색, Neutral-파란색, Stressed-빨간색, Surprised-노란색으로 표현하고 실시간 영상을 시각화하였다.



<그림 18> FER 모듈 실시간 인식률 테스트

표 9에서 정해진 표정 기준으로 실시간 인식률 테스트를 진행하며 표 10의 정확도 판단기준으로 등급화한 후 점수를 합산하여 각 FER 모듈별로 점수를 부여한다. 총 11개 표정의 최대점수는 110점이며 피실험자별 가장 높은 점수를 확보한 모델을 최종 FER 모듈로 선정한다.

<표 11> 피실험자별 실시간 인식률 테스트 결과 및 최종선정 FER 모듈

Subject	InceptionV3		Xception		MobileNetV2		Selected FER
	best	worst	best	worst	best	worst	
Man #1	71	47	101	-40	88	50	Xception Test #1
Woman #1	97	70	101	-10	107	67	MobileNetV2 Test #5
Man #2	94	-3	101	-40	104	38	MobileNetV2 Test #9
Woman #2	104	-73	68	-70	-3	-20	InceptionV3 Test #5
Man #3	48	27	107	-20	104	67	Xception Test #5
Woman #3	90	54	104	10	107	104	MobileNetV2 Test #1 / #5

최종 선정된 FER 모듈은 피실험자별로 매우 상이함을 보인다. 이는 성별, 얼굴형, 개인별 표정변화의 정도, 조명의 적절성 등에 따라 모델별로 인식률이 다름을 나타내었다. 하지만, 실시간 테스트 점수에서 100점을 초과하는 모듈을 사람별로 1개 이상 선정할 수 있었다. 이는 피실험자별로 모든 표정을 감지해 낼 수 있는 모델이 적어도 1개 이상은 존재하여 신뢰성 있는 인식률 확보하였음을 의미한다. 본 실험결과에 따라서 실제 근무환경에 시스템을 적용시 스트레스 검출 대상자별로 FER 모듈 선택할 수 있는 사전 단계를 추가할 필요가 있다.

4.3 스트레스 감지 시스템 시각화

얼굴의 감정표현이 명확한 경우 90% 이상, 비교적 불명확한 경우 65% 이상의 정확성을 보여 해당 표정을 1순위 감정으로 검출 할 수 있었다. 또한, 스트레스 데이터 수치화 모듈을 통하여 그림5와 같이 직관적으로 볼 수 있도록 스트레스 수치 변화를 그래프로 시각화하였다. FER 모듈에서 감정 상태분류를 위해 웹캠으로 이미지를 생성하지만, 개인정보와 업무감시 이슈의 회피를 위해 이미지의 출력과 저장 없이 감정분류 결과만 저장한다. 또한, 고객 상담 시에만 동작하여 상담마다 스트레스 수치를 초기화하므로 그래프를 통하여 상담 건별 스트레스 증감현황을 모니터링할 수 있다.



<그림 19> 실시간 스트레스 감지 시스템 수치변화 모니터링 그래프

V. 결론

본 논문에서 서술한 바와 같이 한국인 데이터셋을 중심으로 한 동서양의 다양한 얼굴 감정 데이터를 확보하고 3가지의 CNN 모델을 Fine Tuning하여 딥러닝한다. CNN 모델의 딥러닝 결과에 따라 각 모델별 Top-3를 선정하여 실시간 테스트를 진행하고 가장 높은 정확도를 가진 모델을 최종 스트레스 감지 FER 모듈로 선정한다. 이를 기반으로 얼굴검출 모듈과 스트레스 데이터 수치화 모듈을 통하여 대표적인 감성노동자인 고객 상담사를 위한 스트레스 감지 시스템을 설계하였다. 본 시스템은 통제 가능한 딥러닝 학습 조건에서는 최대 Accuracy 98%의 높은 학습 정확도를 보여주고 있다. 그러나, 실시간 테스트에서는 성별, 얼굴형 개인별 표정변화의 정도, 조명의 적절성등에 따라 학습된 FER 모듈별로 인식 정확도가 상이하다는 한계점이 있다. 가장 크게 작용하는 요소는 적절한 조명으로써 얼굴 영역 간의 밝기 차이가 명확한 상황에서 FER 모델의 정확도가 높게 나타난다. 따라서, 스트레스 감지 시스템 사용자의 사전 작업으로써 적절한 조명을 확보한 후 제공되는 모든 FER 모듈을 테스트하여 가장 높은 정확도를 보이는 모듈을 채택할 수 있도록 해야 한다. 이렇게 셋팅된 시스템으로 업무감시와 개인정보 제공 동의가 완료된 피실험자의 추가적인 감정 이미지 데이터를 수집할 수 있을 것이다. 향후 연구에서는 이렇게 수집된 추가 데이터를 이용하여 기존 FER 모듈의 추가 학습하거나, 본 논문에서 사용된 3가지의 CNN 모델 외 다양한 모델에 딥러닝하여 개선된 정확도와 활용성을 가진 시스템으로 발전이 가능할 것이라 예상한다. 이를 바탕으로 고객 상담사뿐만 아니라 다양한 감성노동자의 보호 시스템으로써의 발전을 기대해 볼 수 있다.

VI. 참고문헌

- [1] HOCHSCHILD, Arlie Russell, et al. The Managed Heart: Commercialization of Human Feeling. 1983.
- [2] MORRIS, J. Andrew; FELDMAN, Daniel C. The dimensions, antecedents, and consequences of emotional labor. Academy of management review, 1996, 21.4: 986-1010.
- [3] 서울특별시 감정노동종사자 권리보호센터, “서울시 감정노동 보호 가이드라인(개정판)”, 서울특별시, 2020
- [4] 김천곤, et al. 서비스경제화와 지역 서비스산업 발전전략 (Transition to the Service Economy and Development Strategies for Regional Service Industries). 연구보고서 2018-903, 2018.
- [5] 산업안전보건법 제41조, 시행규칙 제41조(고용노동부령), 시행령 제41조(대통령령)
- [6] “‘감정노동’ 콜센터 노동자 10명 중 8명 우울증 위험” Hankyoreh news. https://www.hani.co.kr/arti/society/society_general/996594.html.
- [7] “FER2013 : Facial Expression Recognition Challenge,” kaggle.com, 2013 수정, 2021-08-25 접속, <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>.
- [8] LIU, Yun; DU, Siqing. Psychological stress level detection based on electrodermal activity. Behavioural brain research, 2018, 341: 50-53.
- [9] KATAOKA, Hisanori, et al. Development of a skin temperature measuring system for non-contact stress evaluation. In: Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol. 20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No. 98CH36286). IEEE, 1998. p. 940-943.

- [10] SHAN, Yuhao, et al. Remote detection and Classification of human stress using a depth sensing technique. In: 2018 First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia). IEEE, 2018. p. 1-6.
- [11] SAKRI, Oumayma, et al. A Multi-User Multi-Task Model for Stress Monitoring from Wearable Sensors. In: 2018 21st International Conference on Information Fusion (FUSION). IEEE, 2018. p. 761-766.
- [12] LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. *nature*, 2015, 521.7553: 436-444.
- [13] RUSSAKOVSKY, Olga, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 2015, 115.3: 211-252.
- [14] ZHANG, Jin, et al. Detecting negative emotional stress based on facial expression in real time. In: 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP). IEEE, 2019. p.430-434.
- [15] ALMEIDA, Jose; RODRIGUES, Fatima. Facial Expression Recognition System for Stress Detection with Deep Learning. 2021.
- [16] CHIWANDE, Sujata, et al. Review on Deep Learning Based Stress Detection Using Facial Expressions.
- [17] MICHEL, Caroline; CALDARA, Roberto; ROSSION, Bruno. Same-race faces are perceived more holistically than other-race faces. *Visual Cognition*, 2006, 14.1: 55-73.
- [18] LECUN, Yann, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, 86.11: 2278-2324.
- [19] GUPTA, Ankita; GURRALA, Gurunath; SASTRY, P. S. An online power system stability monitoring system using convolutional neural networks. *IEEE Transactions on Power Systems*, 2018, 34.2: 864-872.

- [20] "Imagenet," IMAGENET, 2021-03-11 update, 2021-08-26, <https://image-net.org/index.php>.
- [21] HE, Kaiming, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770-778.
- [22] 이주열. 인공지능 이미지 인식 기술 동향. 한국정보통신기술협회 TTA저널 vol 187. 2020. p. 44-51
- [23] JUNG, Heechul, et al. Joint fine-tuning in deep neural networks for facial expression recognition. In: Proceedings of the IEEE international conference on computer vision. 2015. p. 2983-2991.
- [24] MOLLAHOSEINI, Ali; CHAN, David; MAHOOR, Mohammad H. Going deeper in facial expression recognition using deep neural networks. In: 2016 IEEE Winter conference on applications of computer vision (WACV). IEEE, 2016.
- [25] 한국전자기술연구원 지능정보플래그십 R&D데이터 <https://aihub.or.kr/open-data/keti-data/recognition-visual/KETI-01-001>
- [26] Copyright(c) 2018, BUPT_GWY <http://www.seeprettyface.com>
- [27] E.Lundqvist, D., Flykt, A., & Ohman, A. "The Karolinska Directed Emotional Faces - KDEF, CD ROM from Department of Clinical Neuroscience" Psychology section, Karolinska Institutet, ISBN 91-630-7164-9. 1998.
- [28] Ma, Correll, & Wittenbrink "The Chicago Face Database: A Free Stimulus Set of Faces and Norming Data." Behavior Research Methods, 47, 1122-1135. 2015.
- [29] Ma, Kantner, & Wittenbrink "Chicago Face Database: Multiracial Expansion." Behavior Research Methods. 53.3. 1289-1300. 2020.
- [30] Lakshmi, Anjana, et al. "The India Face Set: International and Cultural

Boundaries Impact Face Impressions and Perceptions of Category Membership." *Frontiers in psychology* 12 :161. 2021.

- [31] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Ieee, p. I-I. 2001.
- [32] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE conference on computer vision and pattern recognition*. p. 2818-2826. 2016.
- [33] Chollet, Francois. "Xception: Deep learning with depthwise separable convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. p. 1251-1258. 2017.
- [34] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. p. 4510-4520. 2018.
- [35] "Advanced Guide to Inception v3 on Cloud TPU," Google Cloud TPU Document, n.d. 2021-09-26, <https://cloud.google.com/tpu/docs/inception-v3-advanced>.
- [36] Howard, Jeremy, and Sebastian Ruder. "Universal language model fine-tuning for text classification." *arXiv preprint arXiv:1801.06146* (2018).
- [37] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014): 1929-1958.