2024년 2월

석사학위 논문

기계학습과 RGB 영상처리를 이용한 수어 동작 인식 방안

조선대학교 대학원

산업공학과

김 건 우

기계학습과 RGB 영상처리를 이용한 수어 동작 인식 방안

Sign language motion recognition using machine learning and RGB image processing

2024년 2월 23일

조선대학교 대학원

산업공학과

김 건 우

기계학습과 RGB 영상처리를 이용한 수어 동작 인식 방안

지도교수 박 형 준

이 논문을 공학 석사학위신청 논문으로 제출함

2023년 10월

조선대학교 대학원

산업공학과

김 건 우

김건우의 석사학위논문을 인준함

위원장 김 성 준 (인)

위 원 신 종 호 (인)

위 원 박 형 준 (인)

2023년 12월

조선대학교 대학원

목차

그림 목차

표 목차

ABSTRACT

Sign language motion recognition using machine learning and RGB image processing

Kun-Woo Kim

Advisor: Prof. Hyungjun Park, Ph.D.

Department of Industrial Engineering

Graduate School of Chosun University

This study proposes a method to recognize sign language motions using machine learning and RGB image processing. The sign language motions for recognition are selected from 24 sign language words frequently used in the living spaces of the deaf community. For this, 180 sign language motions were acquired using the National Institute of Korean Language's Korean Sign Language Dictionary and public data services. Subsequently, skeleton data was tracked and visualized using OpenCV and MediaPipe, and key points of the human body were extracted to construct the training data.

The training data consists of raw-data representing the location data (255 points) of the extracted features, and feature-data representing vector information (54 points) made based on the location data. After that, unnecessary frames were removed through data preprocessing, and the data was processed into a form suitable for the learning model. As a result of

training input data on convolutional neural network models and recurrent neural network models using two types of training data, the CNN model using raw-data showed an accuracy of 92.58%, and the LSTM model showed an accuracy of 96.64%. On the other hand, the CNN model using feature-data showed an accuracy of 98.75%, and the LSTM model achieved an accuracy of 99.85%.

To verify the usefulness of the developed model, an experimental environment was set up, and 24 new sign language words were filmed using a mobile phone camera and a webcam. As a result of evaluating the performance of the model using the filmed videos, the CNN model using raw-data showed an accuracy of 79.16%, and the LSTM model showed an accuracy of 87.5%, whereas the CNN and LSTM models using feature-data each recorded 91.66%. Through this, it was confirmed that feature-data using vector information is more effective in recognizing sign language motions than location data of features.

However, recognition errors of the model for some sign language words were confirmed, which is due to similar types of motions. It was concluded that it is necessary to construct additional training data and tune parameters to solve this. The results of this study are expected to make a meaningful advancement in promoting communication between sign language users and non-sign language users, and it is intended to be used in creating sign language avatar animations and real-time sign language interpretation systems. In the future, it is expected that the daily life of sign language users will be further improved through additional research for model performance improvement and expanding the sign language motion recognition model to recognition models that can be used in various fields.

제 1 장 서론

제 1절 연구 배경

농인들이 주로 활용하는 의사소통 수단은 주로 수어이다. 세부적으로 살펴보면, 일상적인 의사소통에서 가장 많이 사용하는 언어는 수어인 농인이 약 70%로 응답한 결과가 있다[1]. 이는 농인들이 수어를 자연스럽게 활용하고 있다는 것을 시사한다. 그러나 가족 간 의사소통에서는 수어 사용 비율이 42.7%로 낮게 나타났다. 이는 가족 구성원 전체가 수어에 능숙하지 않아서인 것으로 해석된다. 생활과 밀접한 기관인 관공서에서는 42.3%, 금융 기관에서는 45.7%가 수어가 아닌 필담으로 주로 소통하고 있음이 확인되었다. 또한, 정부에서 제공하는 수어 통역 서비스에 대한 경험 유무를 조사한 결과로는 2020년 기준으로 30대(72.9%), 40대(84.5%), 50대(75.5%)가 주로 해당 서비스를 이용한 경험이 있다고 보고되었다[2]. 《한국수화언어법》에서는 한국수어를 국어와 동등한 자격을 가진 농인의 고유한 언어로 명시하고 있어, 농인들이 차별 없이 일상생활을 영위할 수 있도록 제도적이고 정책적인 지원이 확대되어야 함을 강조하고 있다.

그러나 수어를 배우고 농인들과 자유롭게 대화하기까지는 상당한 시간과 노력이 필요하며, 수어를 습득하기 위한 방법과 수단들은 여전히 한계를 가지고 있다. 이에 대한 해결책으로, 국립국어원에서는 한국수어 교육과정 및 교재를 개발하고, 수어 교육을 위한 교사 양성에 많은 노력을 기울이고 있다. 효과적인 수어 교육의 개발과 적용에 대한 수요가 증가함에 따라, 다양한 연구들이 진행되고 있다. 기타 연구기관에서는 동작 측정 장비를 활용한 수어 해석 시스템, 컴퓨터 비전 기술과 인공지능을 접목한 수어 동작 인식 시스템, 그리고 영상 및 가상 아바타를 활용한

수어 에니메이션 제작 등의 다양한 주제로 연구가 진행되고 있다.

현재까지 수어 동작을 인식하는 기존 방법으로는 센서를 이용한 동작 인식 방법과 카메라를 활용한 동작 인식 방법이 주로 사용되고 있다. 센서를 활용한 동작 인식 방법은 데이터 글러브[3], IMU 센서[4], MYO[5] 등 가속도 및 굽힘 센서가 내장된 디바이스를 착용하여 사용자의 위치 데이터를 추적함으로써 동작을 인식하는 방법이다. 이 방법은 내장된 센서를 통해 정확한 인식이 가능하다는 장점이 있지만, 고가의 디바이스로 일반적인 사용이 어려우며, 사용자의 수어 동작에 따라 동작이 제한될 수 있는 단점이 있다. 반면에, 카메라를 활용한 동작 인식 방법은 Kinect[6], LeapMotion[7]과 같이 깊이카메라가 내장된 디바이스를 사용하여 사용자의 골격 데이터를 추적하고 이를 활용하여 동작을 인식하는 방법이다. 이 방법은 상대적으로 저렴한 가격으로 디바이스를 구입할 수 있고, 착용의 불편함 없이 사용이 가능하다는 장점이 있지만, 고정된 디바이스로 인해 동작 인식 공간이 제한되고, 수어 동작에 따라 정확한 골격 데이터 생성이 어려워 인식 정확도가 낮아질 수 있는 단점이 있다.

최근에는 센서나 깊이카메라가 내장된 디바이스를 사용하지 않고, 일반 카메라와 인공지능 기술을 병행하여 수어 동작을 인식하는 방안이 소개되면서, 이와 관련된 연구가 활발히 이루어지고 있다. Kim 등[8]은 카메라와 인공지능 모델 기반 실시간 동작 인식 모델을 개발하여 수어 동작을 인식하는 연구를 진행했으며, 손(가락), 상체, 얼굴표정으로 구성된 수어 동작 중에서 손으로만 표현 가능한 지화를 인식하는 모델을 개발하여 높은 인식 정확도를 보였다. 그러나 대부분의 수어 표현들은 상체의 움직임과 얼굴표정이 포함되어 있어 다양한 수어 동작 인식이 제한되는 단점이 있어, 보다 다양하고 정확한 수어 동작 인식 방안이 필요하다. 이에 본 연구에서는 RGB 카메라를 이용하여 손(가락), 상체, 얼굴표정의 골격 데이터를 추적하는 MediaPipe[9]를 활용하여 학습 데이터를 구축하고, 기계학습 모델을 통해 수어 동작을 인식하는 모델을 개발하여 실시간 수어 동작 인식 방안을 제안한다.

제 2절 연구 목적

본 연구의 목적은 정부에서 시행 중인 수어 서비스 확대 정책을 통해 농인과의 자유로운 의사소통이 가능하고, 저렴하면서도 편리한 사용이 가능한 RGB 카메라를 활용하여 수어 동작을 촬영하고, 이를 실시간으로 인식하는 기계학습 모델을 개발하여 수어 동작을 모르는 일반인들도 쉽고 원활한 의사소통이 가능한 수어 동작 인식 시스템을 구현하고자 한다. 이를 위해 RGB 카메라를 이용한 골격 데이터 추적 방법, 골격 데이터를 활용한 학습 데이터 구축 방법, 높은 정확도의 수어 동작 인식을 위한 기계학습 모델 개발 방법 등을 조사하고 연구한다. 본 연구에서 수행되는 세부적인 사항은 다음과 같다.

(1) RGB 카메라를 이용한 골격 데이터 추적 방안

본 연구에서는 Google에서 제공하는 AI Frameworks 중 하나인 MediaPipe를 활용하여 얼굴 특징점, 왼손/오른손 손 특징점, 상체 특징점을 정밀하게 추적한다. 이를 통해 수어 동작 인식에 필요한 주요 특징점을 분류하고, 이를 시각화하여 수어 동작의 골격 데이터를 추적하는 방법을 상세히 설명한다. 이러한 접근은 촬영된 영상 데이터를 효과적으로 분석하고, 수어 동작의 핵심 특징을 추출함으로써 의미 있는 결과를 도출한다.

(2) 골격 데이터를 활용한 학습 데이터 구축

본 연구에서는 수어 동작 인식을 위해 선별된 특징점을 기반으로 동작에 대한 벡터 정보를 생성한다. 이러한 벡터 정보는 특징점들 사이의 거리 및 각도를 계산하여 수어 동작의 핵심적인 특성을 반영한다. 이를 통해 구축된 학습 데이터는 수어 동작의 다양한 양상을 포함하며, 이를 통해 모델이 다양한 상황에서 효과적으로 동작할 수 있도록 한다 이러한 접근은 수어 동작의 특성을 정확하게 학습하는데

기여하며, 다양성과 일반성을 고려한 학습 데이터의 구축 방법에 대해 상세하게 설명한다.

(3) 기계학습 모델 개발

벡터 정보로 구성된 학습 데이터를 기반으로하여 Python의 Tensorflow[10] 패키지를 활용하여 연속된 수어 동작을 분석하기 위한 모델을 개발하기 위해 순환 신경망과 합성곱 신경망을 사용한다. 순환 신경망은 시계열적 정보를 고려하여 동작의 연속성을 학습하고, 합성곱 신경망은 이미지 기반의 특징을 추출하여 더 다양한 동작 패턴 학습이 가능하다 순환 신경망과 합성곱 신경망을 개발하여 종합적이고 효과적인 수어 동작 인식 모델을 구축한다.

(4) 제안된 인식 모델의 유용성 검증 및 모델 성능 비교

제안된 방안으로 개발된 기계학습 모델에 실제로 촬영한 수어 동작을 적용하여 도출된 결과를 비교하고 분석한다. 각 모델의 성능을 정량적으로 비교하여 제안된 방안이 어떠한 특성을 가지며, 어떠한 상황에서 더 우수한 성능을 보이는지를 규명하고, 이를 통해 효과적인 수어 동작 인식 방안의 유용성을 위한 실증적인 검증을 진행한다.

제 3절 논문 구성

본 논문의 구성은 1장 이후로 다음과 같다. 2장에서는 기존의 수어 동작을 인식하기 위한 방법으로 센서 및 카메라를 활용한 접근 방식을 소개하고, 이에 더해 최신의 인공지능 기술을 활용한 방법에 대한 고찰을 제시한다. 3장에서는 수어 단어를 인식하는 기계학습 모델을 개발하기 위한 세부적인 절차를 다룬다. 이 과정에서 수어 단어 범위의 선택, RGB 카메라 및 MediaPipe를 활용한 골격 데이터 추적 골격 데이터를 활용한 학습 데이터 구축, 그리고 합성곱 신경망과 순환 신경망을 통한 기계학습 모델 개발에 대한 설명을 포함한다. 4장에서는 3장에서 제안된 방법을 기반으로 구현된 인식 모델을 평가하고, 각 모델들을 비교/분석하여 유용성을 검증한다. 마지막으로 5장에서는 결론을 도출하고 향후 연구 방향에 대해 논의한다.

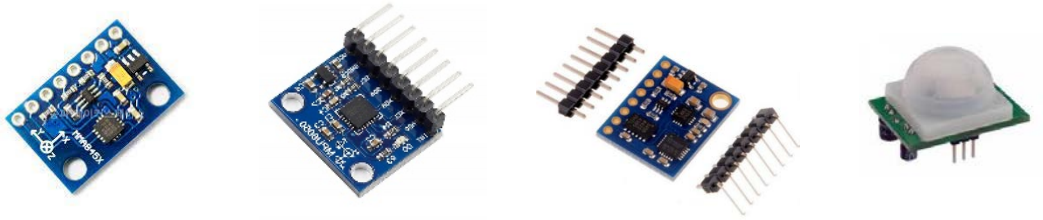
제 2 장 기존연구 고찰

제 1절 센서 및 카메라 기반 동작 인식

수어는 농인들의 소통 수단 중 핵심적인 역할을 하며, 농인들의 사회 참여 및 의사소통을 지원하는 데 중요한 도구로 작용한다. 그러나 수어를 익히는 데에는 상당한 노력과 시간이 소요되기 때문에, 농인과 일반 사회 간의 의사소통에 어려움이 존재한다. 본 연구의 주된 목적은 농인들이 보다 효과적으로 수어를 익히고, 일상적인 의사소통을 원활하게 할 수 있도록 도와주는 수어 동작 인식 방안을 모색하고자 한다. 본 절에서는 센서 및 영상 기반 수어 동작 인식 방안에 대한 간략한 설명과 기존 연구를 살펴본다.

1.1 센서를 이용한 동작 인식

수어 동작을 인식하기 위해서는 먼저 동작을 인식해야 하는데 동작을 인식하는 방법으로는 크게 두 가지로 나누어 진다. 먼저 센서를 이용한 동작 인식 방안의 경우 가속도, 자이로스코프, 자력, 모션 센서 등 다양한 센서를 통해 속도의 변화, 방향, 움직임의 변화, 축 주위의 회전, 자기장의 변화 등을 정보로 하여 동작을 추론한다. 센서를 이용한 동작 인식 방안의 경우 센서가 내장된 디바이스를 착용함으로써 센서의 위치 변화를 최소화하고, 정확한 정보 수집이 가능하다는 장점이 있다. 그림 1은 동작 인식을 위해 사용되는 센서들을 나타낸다.



(a)가속도 센서 (b)자이로 센서 (c) IMU 센서 (d) 모션 센서

그림 1. 센서를 이용한 동작 인식 방안에 사용되는 센서

[출처, ICBANQ, www.icbanq.com]

그림 2는 동작 인식을 위해 사용되는 디바이스들을 나타낸다

(a)Data Gloves (b)Motion Suit (c) Myo Armband

그림 2 센서가 내장된 디바이스 종류

[출처, PerceptionNeuron, Myo]

센서가 내장된 디바이스를 이용하여 동작을 인식하는 방안은 실시간으로 사용자의 동작 해석이 가능하고, 디바이스로부터 동작에 대한 정보들이 도출되기 때문에 정확한 동작 인식이 가능하다. 그러나 고가의 디바이스를 구입해야 하며, 동작 인식을 위해 착용의 불편함이 존재하고, 동작에 따라 사용자의 동작이 제한될 수

있다. 또한, 동작 인식을 위한 공간을 따로 구현해야 하는 경우도 존재한다. 그림 3은 동작 인식을 위해 센서가 내장된 디바이스를 사용하는 모습이다. 모션캡쳐를 이용한 수어 인식 방안 연구로 Na 등[11]은 수화 통역을 위한 VR 콘텐츠를 개발하기 위해 모션캡쳐 장비와 데이터 글러브를 사용하여 수화 애니메이션을 제작하고, 애니메이션 DB를 구축해 사용자들에게 수어 애니메이션을 보여주는 콘텐츠 제작 방안을 제안하였다. 하지만 정확한 수화 캐릭터 애니메이션을 생성하기 위해 보정 및 편집 작업에 많은 시간과 노력이 필요했으며, DB를 구축하는데 어려움이 있다는 단점이 있다.

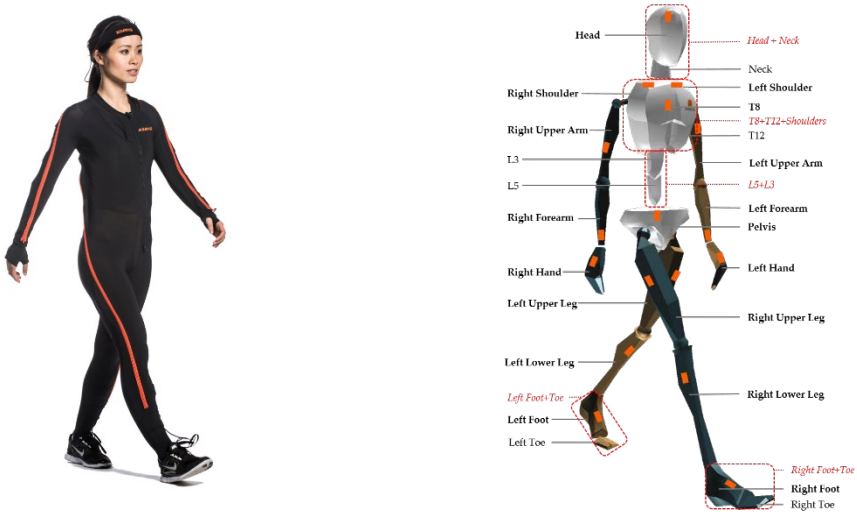


그림 3 디바이스를 이용한 동작 인식 예

[출처, XSENS, MVN inertial motion capture system]

1.2 카메라를 이용한 동작 인식

두 번째 방법은 카메라를 활용한 동작 인식이다. 이 방법은 주로 적외선 카메라와 깊이 카메라를 사용하여 동작을 감지하는데, 그 중에서 대표적인 기기로는 Leap Motion과 Kinect가 있다. Leap Motion은 2개의 적외선 카메라와 3개의 적외선 LED로 구성되어 있으며, 8입방 피트 크기의 3차원 공간 내에서 사용자의 손의 골격 데이터를 획득할 수 있다. Leap Motion은 착용할 필요가 없으며, 뛰어난 감도(1/100mm)로 손가락의 움직임까지 정확하게 측정할 수 있습니다.[12] 또한, Leap Motion은 스테레오 방식의 깊이 카메라를 사용하여 2개의 이미지 센서를 결합하여 대상과의 거리를 측정하며, 시점 차이를 이용해 이미지 차이를 통해 깊이를 인식한다. 그림 4는 Leap Motion 및 이를 통해 얻은 골격 데이터를 나타내고 있다.

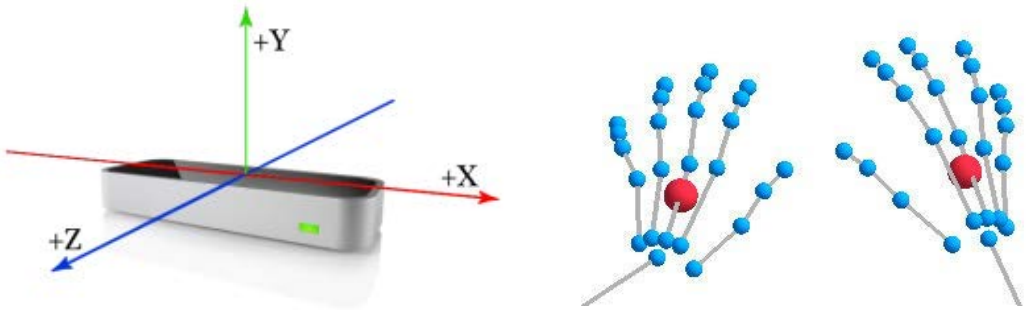


그림 4 립모션과 립모션을 통해 획득한 골격 데이터

[출처, LeapMotion, www.ultraleap.com]

수어 동작을 인식하기 위해 립모션을 이용한 다양한 연구들이 존재한다. Moon 등[13]은 수어에 해당하는 지화 동작을 인식하기 위해 립모션을 사용하여 31개에 해당하는 지문자 인식 모델을 개발하였으며, 2개의 분류 알고리즘으로 10개의 한국 수화 데이터 세트를 통해 실험한 결과 각각 67.5%, 83.6% 인식률을 보였다. 실험 결과에 대한 분석으로 립모션을 통해 표현이 어려운 지문자가 존재하였고, 그 결과

정확도가 떨어지는 문제가 있음을 확인하였다. 국내뿐 아니라 국외에서도 립모션을 활용하여 알파벳의 수어를 인식하는 연구가 진행되었다. Naglot, D[14] 등은 립모션을 이용하여 알파벳에 해당하는 수어 표현을 인식한 결과 6개 알파벳에서 정확도가 떨어지는 문제를 확인하였다.

Kinect는 Color Sensor, IR Emitter, IR Depth Sensor, Tilt Motor가 내장된 RGB 카메라로 구성되어 있으며, ToF(Time of Flight) 방식을 사용하여 적외선을 이용해 반사되어 오는 시간을 측정하여 거리를 계산하여 위치 데이터를 추적한다. Kinect는 사람의 신체 부위와 움직임을 인식할 수 있기 때문에 센서와 사람 사이의 거리, 각 관절 값 등을 정확하게 추적하는 데 높은 성능을 보인다. 최대 6명을 동시에 촬영할 수 있으며, 25개의 골격 데이터를 획득할 수 있다. 수어 동작을 인식하기 위한 Kinect 연구도 많이 이루어지고 있는데 Cheon 등[15]은 Kinect를 활용하여 24개의 수화 단어를 10번씩 수행한 결과, 22%의 오류 발생률이 있었지만 81.4%의 인식 정확도를 보였고, Dong, C 등[16]은 Kinect를 활용하여 알파벳 수어를 인식하는 연구를 진행하였지만, 어려운 손 동작이 요구되는 단어에서 인식 정확도가 떨어짐을 확인하였다. Kinect를 이용한 수어 인식은 상체 및 손의 움직임을 추적하여 다양한 수어 동작을 인식할 수 있지만, 섬세한 손 모양이나 손 동작이 몸에 가려지는 문제로 인해 정확도가 감소하는 문제가 있음을 확인하였다.

그림 5 키넥트와 키넥트를 통해 획득한 골격 데이터

[출처, Kinect, developer.microsoft.com]

립모션과 키넥트 카메라를 활용한 동작 인식 방안은 센서를 사용한 동작 인식에 비해 상대적으로 적은 비용으로 동작을 감지할 수 있으며, 사용자가 착용할 필요 없이 골격 데이터를 추적하여 위치 정보를 확인할 수 있다. 그러나 수어 동작을 인식하기 위해서는 손가락, 상체, 얼굴 표정과 같은 다양한 정보의 결합이 필요하며, 겹쳐진 수어 동작의 경우 카메라의 가려짐 현상으로 인해 정확한 골격 데이터 획득이 어려워 인식 정확도가 떨어지는 문제가 있다. 특히 수어 동작은 섬세한 동작과 다양한 표현이 필요하기 때문에 이러한 도구들이 한정된 정보만을 제공할 수 있어 한계가 있을 수 있다. 이러한 문제점을 극복하려면 동작 인식 알고리즘의 개선을 필요로 한다.

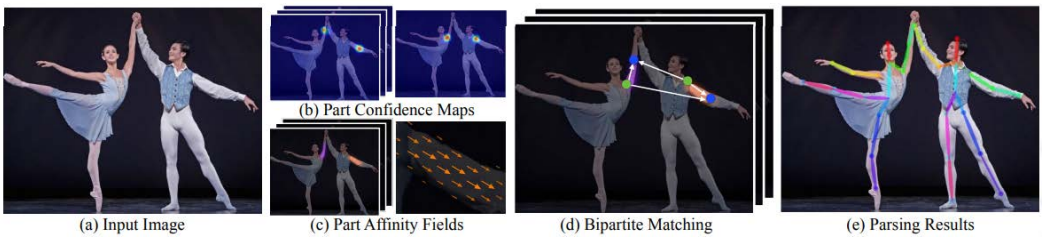
최근에는 일반적인 카메라와 컴퓨터 비전 기술을 이용하여 사람의 골격 데이터를 추적하고 동작을 인식하는 연구들이 활발히 진행되고 있다. 이 분야의 대표적인 라이브러리로는 OpenPose[17], Mediapipe, AlphaPose[18] 등이 있다.

OpenPose는 실시간으로 다중 사람 포즈 추정 및 추적 기능을 제공하는 컴퓨터 비전 라이브러리로, Carnegie Mellon University의 Perceptual Computing Lab에서 개발하였다. 심층 신경망 기반 접근 방식을 사용하여 머리, 목, 어깨, 팔꿈치, 손목, 엉덩이, 무릎, 발목과 같은 다양한 신체 특징점과 눈, 코, 얼굴과 같은 얼굴 특징점을 예측할 수 있고, 유연한 인터페이스 제공을 통해 처음 사용하는 사용자들도 쉽게 사용이 가능하고, CPU, GPU 및 특수 가속기 및 하드웨어 설정을 통해 실시간 이미지와 비디오 영상 처리가 가능하다는 장점을 가지고 있다.

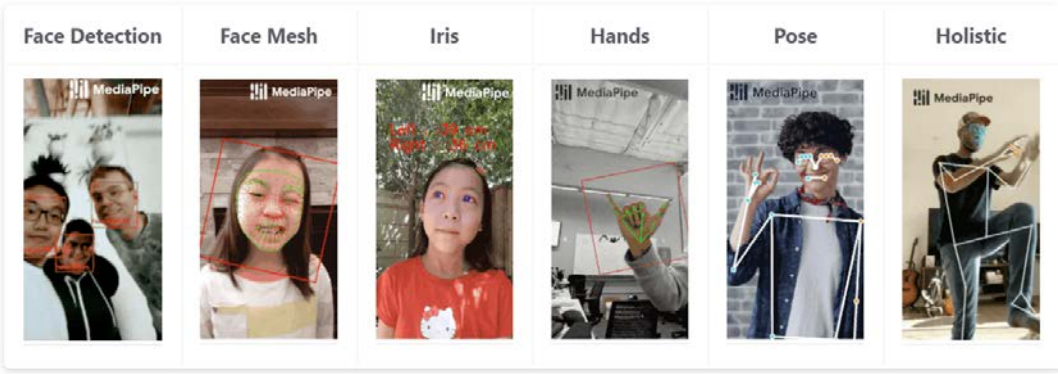
MediaPipe는 Google에서 개발한 프레임워크로, 멀티미디어 처리 파이프라인을 구축하기 위한 포괄적인 블록 빌딩 세트를 제공합니다. MediaPipe는 미디어 데이터를 실시간으로 처리하는 기능으로, 짧은 시간으로 고성능 처리가 가능하며, 이미지,

비디오, 오디오 스트림과 같은 다양한 유형의 미디어 입력을 처리하기 위한 통합 프레임워크를 제공하여 사용자 지정 처리 파이프라인을 쉽게 만들 수 있는 다양한 구조를 제공한다. 또한, 2D 및 3D 미디어 처리를 모두 지원하고, Windows, macOS, Linux, Android, iOS를 포함한 다양한 운영체제를 지원하므로 멀티미디어 애플리케이션을 개발하기 위한 다목적 도구로 활용 가능한 장점이 있다.

AlphaPose는 홍콩 중문 대학에서 개발한 라이브러리이고, 딥 러닝 기술을 이용하여 이미지나 동영상에서 사람의 포즈를 추정하도록 설계되었다. 주요 기능으로 2D 및 3D 인간 포즈를 동시에 추정하고, OpenPose와 동일하게 심층 신경망 접근 방식을 사용하여, 다양한 신체의 특징점을 예측한다. 또한 3D 위치를 추정할 수 있으므로 미터법 공간에서 3D 인간 포즈 추정이 가능하다는 장점을 가지고 있다. 그림 6은 OpenPose, MediaPipe, AlphaPose에 대한 소개 그림을 보여준다.



(a) OpenPose Example



(b) MediaPipe Examlpe

(c) AlphaPose Example

그림 6 골격 데이터 추정을 위한 컴퓨터 비전 기술의 예

[출처, Poes estimation, github.com/MVIG-SJTU/AlphaPose, github.com/google/mediapipe, github.com/Perceptual-Computing-Lab/openpose]

컴퓨터 비전 기술을 활용한 골격 데이터 추적 및 동작 인식의 연구 접근 방법은 주로 기계학습을 활용한다. 이 방법은 추적된 골격 데이터를 학습 데이터로 구축하고, 이를 기반으로 기계학습을 수행하여 동작을 인식하는 원리를 따른다. 추적된 골격 데이터의 패턴과 특징을 학습하여 새로운 데이터에 대한 동작을 인식하는 데 중점이 있다. 다음 절에서는 기계학습 기반 동작 인식에 대해 논의한다.

제 2절 기계학습을 이용한 동작 인식

본 절에서는 최근에 일반 카메라를 활용하여 디바이스 없이 골격 데이터를 추적하고, 이를 기반으로 기계학습 기술을 활용한 수어 동작 인식 방법에 대해 작동 원리와 구성에 대해 설명한다. 또한, 수어 동작을 인식하기 위한 컴퓨터 비전 방안과 신경망 모델 개발 방안에 대한 선행 연구를 살피고, 이를 통해 일반적인 환경에서의 수어 동작을 높은 정확도로 인식할 수 있는 기술에 대한 통찰을 얻고자 한다.

2.1 심층 신경망 동작 인식

심층 신경망(Deep Neural Network, DNN)[19]은 입력 계층과 출력 계층 사이에 여러 계층이 있는 인공 신경망(Artificial Neural Network, ANN) 이다. Deep Layer를 통해 입력데이터의 복잡한 기능과 패턴을 학습이 가능하여, 이미지 인식, 음성 인식, 자연어 처리 등과 같은 작업을 수행한다. 심층 신경망의 각 계층은 입력 데이터에 대한 계산을 수행하는 상호 연결된 여러 뉴런으로 구성되는데 한 계층에 있는 뉴런의 출력은 다음 계층에 있는 뉴런의 입력이 되어 네트워크가 점진적으로 학습할 수 있다. 또한, 예측 출력과 실제 출력 사이의 오류를 최소화하기 위해 뉴런의 가중치와 편향을 반복적으로 조정하는 역전파(backpropagation) 기술을 사용하여 훈련하고, 훈련 과정에는 많은 양의 데이터와 계산 리소스가 필요하지만 훈련 이후 새로운 데이터에 대해 정확한 예측이 가능한 장점을 가지고 있다. 그림 9는 심층 신경망의 대표적인 모델인 다층 퍼셉트론(Multi-Layer Perceptron, MLP) 구조를 나타낸다.

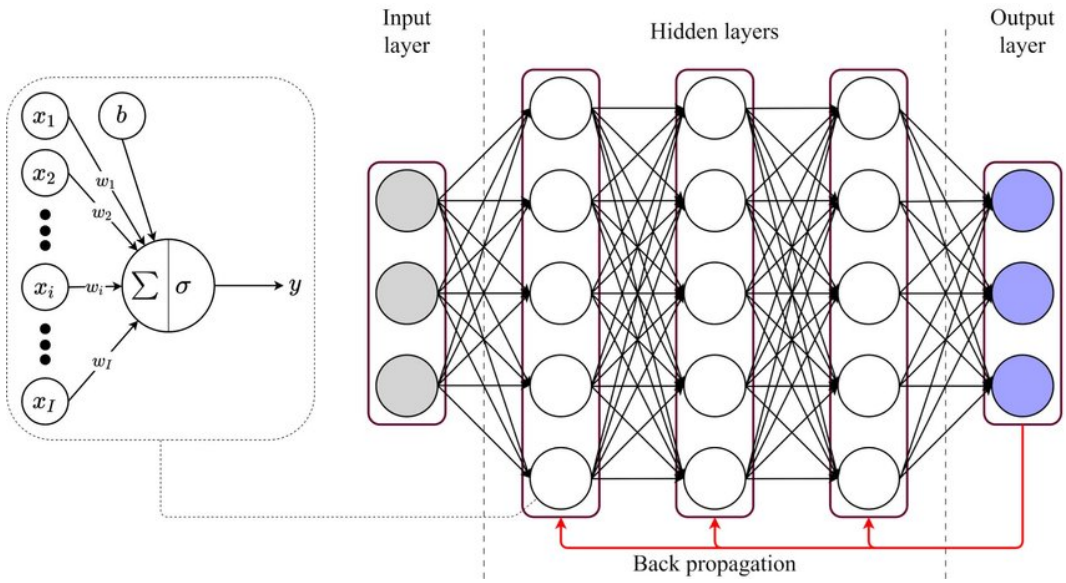
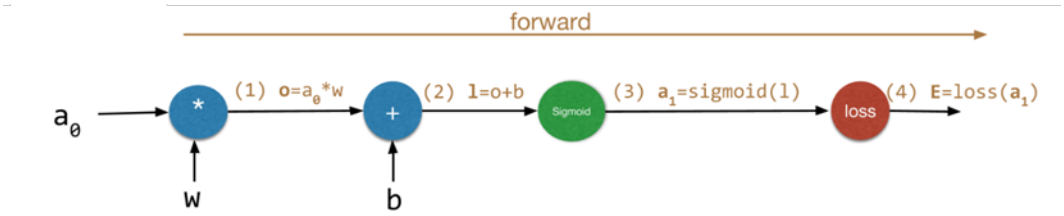


그림 7 Multi-Layer Perceptron, MLP

다층 퍼셉트론은 입력층과 출력층 사이에 하나 이상의 은닉층을 가지고 있는 신경망이며, 여기서 퍼셉트론은 두뇌의 인지 능력을 모방하도록 만든 인위적인 네트워크를 의미한다. 학습 과정은 입력 데이터와 그에 해당하는 가중치를 곱한 값들의 합을 활성화 함수(activation function)에 전달하고, 활성화 함수로부터 출력된 값을 다시 입력 데이터로 보고, 위 과정을 반복하여 출력층까지 순차적으로 계산한다. 최종 활성화 함수에서 출력 데이터로 계산되고, 이 출력 데이터와 실제 데이터를 손실함수에 넣어 연산하여 오차(cost)를 계산한다. 이러한 과정을 순전파라고 하며, 은닉층의 수가 많아지면 순전파를 지속적으로 수행하게 되면서 연산량이 높아 비효율적인 학습이 진행된다. 그림 10은 순전파 계산 과정을 나타낸다.

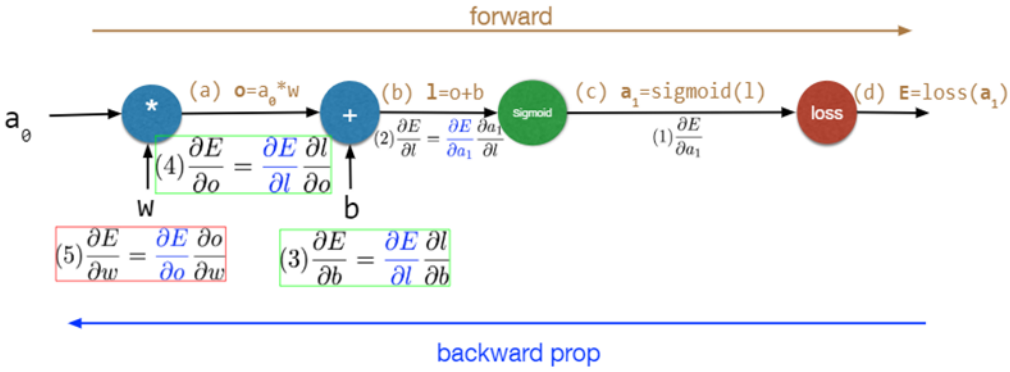


$$a1 = sigmoid\ (w*a0 + b)$$

$$loss, E(a,t) = -\sum t * log(a) + (1-t) * log(1-a)$$

그림 8 순전파(Forward Propagation) 계산 과정

비효율적인 학습을 해결하기 위해 입력데이터 와 출력 데이터를 알고 있는 상태에서 신경망을 학습시키는 지도학습(supervised learning)을 통해 각각의 가중치와 편향을 최적화 알고리즘을 통해 오차를 줄일 수 있는 가중치와 편향을 계산한다. 이러한 과정을 역전파라고 하며, 최종적으로 정확도를 높이기 위한 최적의 가중치와 편향을 가진 모델을 개발할 수 있다. 그림 11은 역전파 계산 과정을 나타낸다.



$$a = sigmoid(l) = \frac{1}{1+e^{-l}}, \quad \frac{\partial a}{\partial l} = a(1-a)$$

그림 9 역전파(Back Propagation) 계산 과정

[출처, MLP 순전파와 역전파 개념, www.velog.io/@diduya/]

하지만 다층 퍼셉트론 모델은 학습 데이터 세트에 너무 가중치와 편향이 최적화되어 과접합(Overfitting)으로 인해 새로운 데이터에 대한 정확도가 떨어지는 문제가 있고, 은닉층을 거치면서 계속된 경사하강법(Gradient descent)으로 인해 뒤로 전해지는 오차값이 현저히 작아져 학습이 제대로 이루어지지 못하는 문제점이 있다,

이러한 문제들로 인해 데이터의 형태적인 특징을 고려하고, 학습 가중치를 줄여 모델의 복잡도를 낮추고, 데이터의 특징을 추출하기 위한 방안으로 합성곱 신경망(Convolutional Neural Network, CNN)[20] 방안이 고안되었다.

합성곱 신경망은 이미지 및 비디오 같이 그리드(grid)와 같은 구조를 가진 데이터를 처리하도록 설계된 인공 신경망이다. CNN은 입력데이터에 컨볼루션 레이어를 적용하여 입력 이미지 위에 커널 또는 필터를 넣어 각 위치에서 커널과 입력 사이의 내적을 계산하고, 서로 다른 공간 위치에 있는 이미지에서 기능을 효과적으로 추출하고,

네트워크의 추가 계층으로 전달한다. 또한, CNN은 추출된 기능을 원하는 클래스에 매핑(mapping)하는 연결 레이어(Connected Layer)뿐만 아니라 인근 픽셀 그룹을 요약하여 컨볼루션 레이어에서 출력의 공간 차원을 줄이는 풀링(pooling)레이어 기능을 가지고 있고, 물체감지, 얼굴 인식 및 장면 분류를 포함한 다양한 이미지 및 비디오 인식 작업에서 많은 연구에 활용되고 있다. 그림 6은 CNN의 구조를 나타낸다.

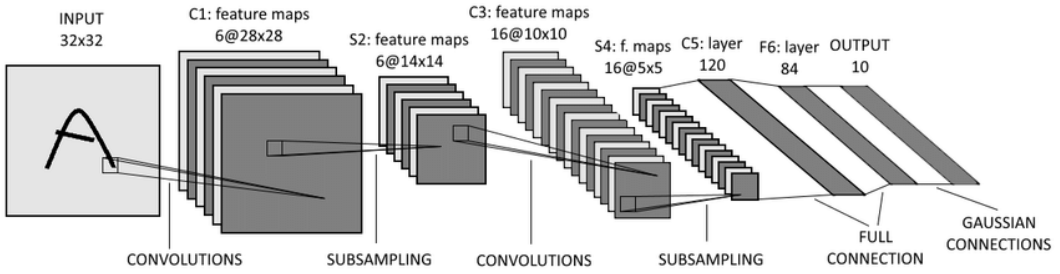


그림 10 Convolutional Neural Network, CNN

Convolutional Layer는 입력데이터로부터 특징을 추출하는 역할을 하며, 특징을 추출하는 기능은 Filter와 Filter의 값을 비선형 값으로 바꾸어 주는 활성화 함수로 이루어져 있다. Filter를 원본 이미지에 적용하는 방식이 Convolution이며, 각 픽셀별로 필터와 곱하고 그 결과를 모두 더하는 연산으로 이루어 진다. 그림 7은 Filter와 Convolution 연산 과정을 나타낸다.

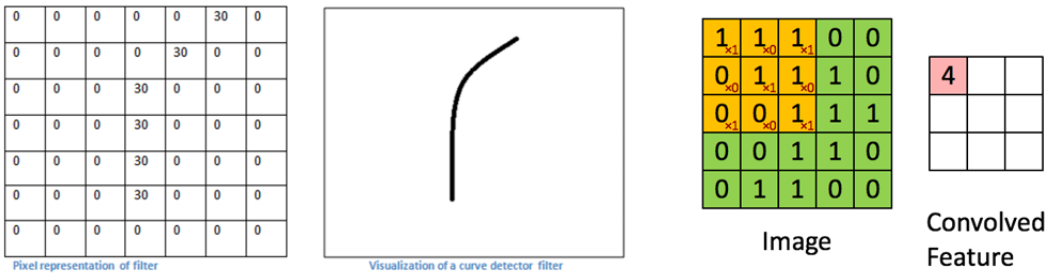
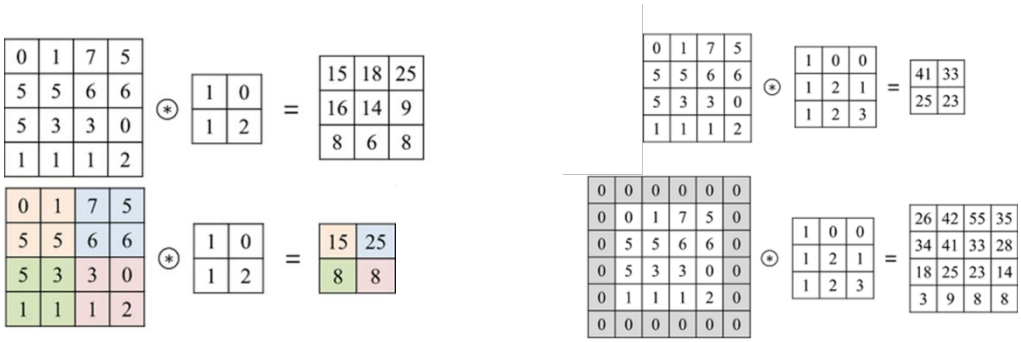


그림 11 Filter 데이터 검출 과정 및 Convolution 연산 과정

Convolutional Layer에서 Stride는 Filter를 얼마만큼 간격으로 이동시켜가며 Convolution 연산의 수행 정도를 결정하는 Parameter이며, Convolution 연산을 여러 번 수행할수록 이미지의 크기가 작아지지 않게 가장자리에 픽셀을 추가함으로써 입력 이미지와 출력 이미지의 크기를 비슷하게 만드는 Padding 기능이 존재한다. 그림 8은 Stride와 Padding 과정을 나타낸다.



(a) Stride (b) Padding

그림 12 Stride & Padding 과정

[출처, CNN, https://bong-sik.tistory.com/21]

Filter를 적용하여 얻어낸 결과를 Feature Map이다. Feature Map에서 모든 데이터가 필요로 하지 않기 때문에 Pooling Layer를 거치면서 이미지 크기도 줄이고 특징값만을 강조하게 한다. Pooling에는 여러가지 방법이 존재하는데 특징값에 따라 Max Pooling, Average Pooling, Min Pooling이 있다. 그 중 Max Pooling 기법이 가장 많이 사용되고 있다. Convolution Layer와 Pooling Layer를 통해 특징들을 추출하고, 추출된 특징값을 Connected Layer를 통해 분류 작업을 진행된다. 이러한 과정을 통해 이미지 및 비디오 영상에 담긴 사용자의 동작 변화를 통해 수어 동작을 인식한다. 관련 연구로는 CNN 기법을 활용하여 수형에 해당하는 동작인식 모델을 개발해 65개에 대한 수형 인식을 실시한 결과 98% 인식 정확도를 확인하였지만 195,000장의 학습 데이터

셋을 구축하면서 학습을 진행하기까지 오랜 시간이 소요되었으며, 몸이나 얼굴로 표현하는 수어가 아닌 손으로 표현하는 수형에 대한 동작 인식으로 다양한 수어 동작을 인식하는데 제한되는 단점이 있다.

2.2 순환 신경망 기반 동작 인식

순환 신경망(Recurrent Neural Network, RNN)[21]은 시계열 데이터 또는 자연어 처리 작업과 같은 순차 데이터를 처리하도록 특별하게 설계된 일종의 신경망 구조이다. 선형 비주기적 방식으로 데이터를 처리하는 Feedforward 신경망과 달리 순환 신경망에는 순환 방식으로 데이터를 처리할 수 있는 루프가 존재한다. 즉, 이전 입력의 컨텍스트를 고려하고 해당 컨텍스트를 사용하여 예측하거나 출력할 수 있다. 순환 신경망에서 각 노드에는 이전 시간 단계의 정보를 저장하는 메모리 또는 은닉 상태가 있으며, 이는 현재 시간 단계에서 네트워크의 출력에 영향을 미치는데 사용된다. 또한, Feedforward 신경망에 사용되는 표준 역전파 알고리즘 변형인 시간 역전파(Back Propagation Through Time, BPTT)를 사용하여 훈련이 가능하다. 순환 신경망의 일부 응용 프로그램에는 언어 모델링, 음성 인식, 감정 분석, 기계 번역 및 비디오 분석 등에 활용된다. 그림 12는 RNN의 대표적인 모델 LSTM[22] 나타낸다.

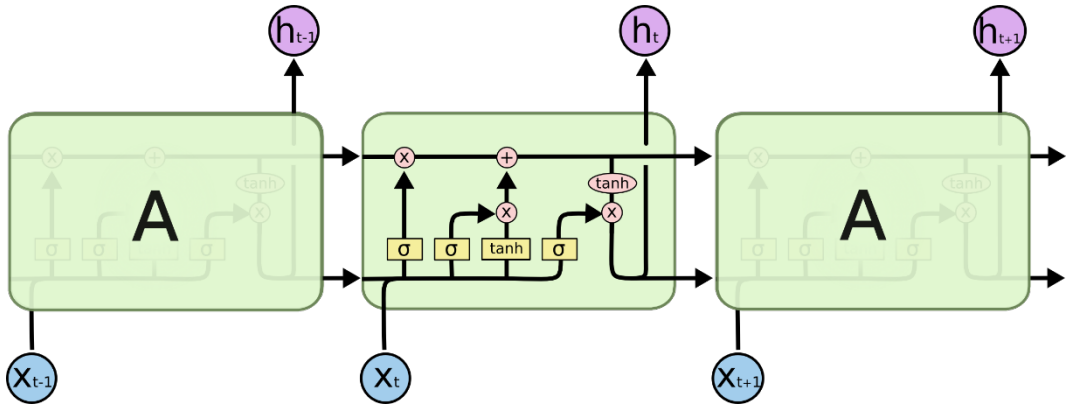
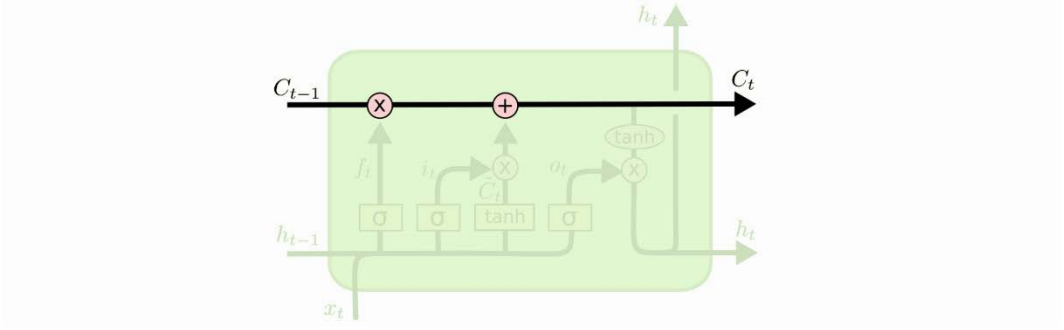
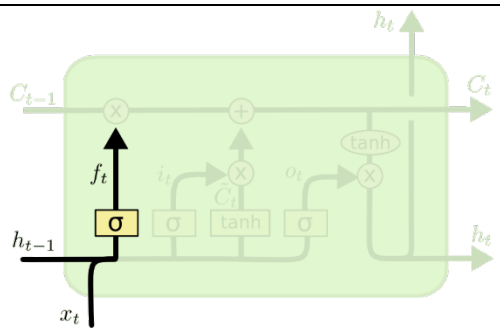


그림 13 Long Short Term Memory, LSTM

LSTM 모델은 기존의 RNN이 출력과 먼 위치에 있는 정보를 기억할 수 없다는 단점을 보완하여 장/단기 기억을 가능하게 설계한 신경망 구조이다. LSTM 모델은 RNN과 같은 체인 구조로 되어 있지만, 반복 모듈은 단순한 한개의 Tanh Layer가 아닌 4개의 Layer가 서로 정보를 주고받는 구조로 되어 있다. LSTM 모델에서는 상태(state)가 크게 두 개의 벡터로 단기 상태(Short Term state)와 장기상태(Long Term state)로 나누어 진다. 표 1은 LSTM cell을 통한 계산 과정을 나타낸다.

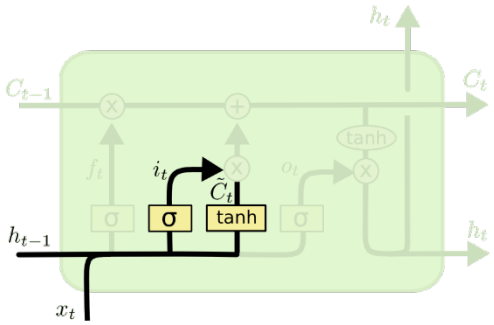
표 1 LSTM 모델 계산 과정


(1) Cell state : 정보가 바뀌지 않고 유지



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

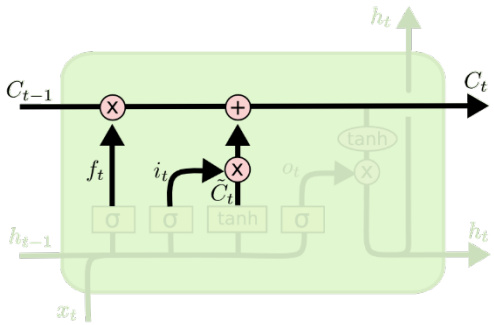
(2) Forget gate : Sigmoid Layer를 통해 정보 삭제



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

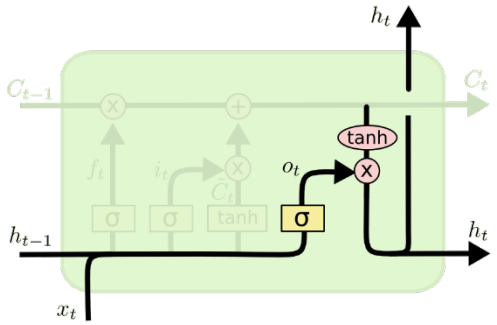
(3) Input gate : 새로운 정보 중 Cell state에 정보 저장

Sigmoid Layer를 거쳐 업데이트 결정 후 Tanh Layer에서 새로운 Vector 생성



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

(4) Cell state update : 새로운 정보 업데이트



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

(5) Output gate : sigmoid Layer에 Input data를 넣어 Output 정보 추출
Tanh Layer에 Cell state를 넣어 sigmoid Layer * Output → 최종 Output

[출처, RNN & LSTM & GRU 완벽 정리, https://hororolol.tistory.com/166]

순환 신경망의 LSTM 모델은 시계열 데이터인 연속동작을 인식하는데 적합하고, 결과에 따른 동작을 분류함으로써 높은 정확도의 장점이 있다. Jeon 등[23]은 Skeleton 정보와 LSTM을 이용하여 작업자의 동작을 인식하는 모델을 개발해 자세 검출을 실시한 결과 99.6% 정확도를 도출하였고, 유사한 동작에서 오차가 발생하였지만 심도 높은 학습을 진행하여 이를 해결할 수 있음을 확인하였다. 하지만 세밀한 동작 인식 모델이 아닌 상체 및 머리의 움직임을 이용한 동작 인식 모델로 수어 동작 인식에는 사용이 적합하지 않았다. 이에 본 연구에서는 세밀한 동작으로 구성된 수어 동작을 인식하기 위해 원활한 골격 데이터 추출 방안을 모색하고, 골격 데이터를 이용한 순환 신경망 모델을 개발하고, 심층 신경망 모델과 분석을 통해 기계학습 모델의 유용성을 검증하고자 한다.

제 3 장 수어 동작 인식을 위한 기계학습 모델

본 장에서는 수어 동작 인식을 향상시키기 위한 기계학습 모델의 제작과정을 다룬다. 우선, 학습용 수어 단어의 선정에 대한 과정과 선택된 단어에 대한 소개를 통해 학습 데이터의 구축 방법을 소개하고, 선정된 단어들에 대한 동영상 자료를 수집 및 분석하고, 동영상에서의 골격 데이터 추적을 위해 OpenCV(Open Source Computer Vision Library)와 MediaPipe를 활용하는 방법에 대해 설명한다. 이후, 수집된 골격 데이터를 학습 데이터로 효과적으로 구축하는 방안에 대해 기술하고, 이를 활용하여 기계학습 모델을 개발하는 과정에 대한 세부 사항을 상세히 기술한다. 끝으로, 학습 데이터를 이용한 기계학습의 수행 과정과 개발된 모델의 성능을 검증한다.

제 1절 학습용 수어 단어의 범위

본 절에서는 수어 동작 인식을 위한 학습용 수어 단어 조사 과정을 설명하고, 수어 단어의 선별 과정, 수어 동영상 수집 방안, 수어 동영상 분석 과정에 대해 기술한다.

1.1 수어 단어 선정

수어 동작 인식 모델을 위해 선정된 학습용 수어 단어는 농인들의 일상생활에서 가장 필수적이며, 특히 전문 인력이나 지원을 받기 어려운 상황에서 효과적으로 사용할 수 있는 은행과 병원이라는 생활 공간을 우선적으로 선정하였다. 농인들의 경우 은행이나 병원과 같은 곳을 방문할 때 수화가 필수적으로 요구되며, 금융거래 및 의료 상황에서 수화표현이 더욱 필요한 경우가 발생한다. 그러나 이러한 장소에서

수화 지원을 받거나 전문 인력을 이용하기 어려운 문제가 존재한다. 이에 은행과 병원을 선정함으로써 수어 동작 인식 모델이 더욱 현실적이고 실용적인 지원을 제공하기 위해 은행과 관련된 단어 12개, 병원과 관련된 단어 12개를 선정하였다.

은행에 관련된 수어 단어는 '빌리다', '돕다', '주다', '지불하다', '대출', '맞다', '가다', '신용카드', '예금', '심사', '없다', '어렵다' 로 선정하였고, 병원과 관련된 수어 단어는 '검사', '감기', '보건소', '소화제', '수면제', '회복', '입원', '진단서', '치료', '퇴원', '의사', '병문안'으로 선정하였다. 표 2는 선정된 24개의 단어를 나타낸다.

표 2 선정된 수어 단어

1	빌리다	**2**	돕다	**3**	주다	**4**	지불하다
5	대출	**6**	맞다	**7**	가다	**8**	신용카드
9	예금	**10**	심사	**11**	없다	**12**	어렵다
13	검사	**14**	감기	**15**	보건소	**16**	소화제
17	수면제	**18**	회복	**19**	입원	**20**	진단서
21	치료	**22**	퇴원	**23**	의사	**24**	병문안

1.2 수어 동영상 수집 및 분석

선정된 수어 단어들에 대한 수어 동작을 확보하기 위해 국립국어원 한국수어사전[24] 및 AI Hub를 통해 수어 단어들을 수집하였다. 한국수어사전은 농인과 청인이 한국수어 단어에 대한 한국어 정보를 쉽게 찾아볼 수 있도록 기존의 한국수어 웹 사전과 모바일 사전 등을 통합하여 새롭게 정비한 사전이며, 수어 정보(동영상, 사진, 설명, 원어 정보, 동형어, 반형어 등)와 한국어 정보(표제어, 품사, 뜻풀이, 용례)가 함께 제공되는 사전이다. AI Hub는 공공데이터 분야로 한국수어 인식 인고지능 기술 및 서비스 개발에 활용 가능한 대규모 데이터셋이다. 총 536,000 수어영상(.mp4)을 가지고 있으며, 영상 구성으로는 문장 2000개, 단어 3000개, 지숫자/지문자

1000개이다. 본 논문에서는 AI Hub를 통해 60개의 수어 동영상을 획득하고, 직접 수어 동작 영상을 촬영하여 120개의 수어 동영상을 제작하였다. 이후, 한국 수어사전을 통해 수어 동작을 분석한다. 수어 동영상 분석은 수어 단어 인식 모델의 성능을 최적화하고, 수어 동작의 특징과 다양성을 파악하고, 각 동작을 정확하게 학습하고 이해하는데 기여된다. 그림 14는 AI Hub를 통해 획득한 수어 동영상을 나타낸다.

그림 14 공공데이터를 이용하여 수집한 수어 동작

[출처, AIHUB, https://www.aihub.or.kr/]

(1) 동작은 오른손 손바닥을 위로 하고, 손가락 끝이 바깥쪽을 향하게 손을 펴준 상태에서 오른손을 안으로 당기면서 손 끝을 맞대는 동작이다. (2) 동작은 왼손은 주먹을 쥐고, 엄지 손가락을 펴서 밖으로 향하게 세워준 뒤 몽의 중앙으로 위치하고, 오른손을 가볍게 편 상태로 왼손 등에 오른 손바닥을 세워 두 번 대는 동작이다. (3) 동작은 오른손을 펴서 손바닥이 위를 향하게 하고 몸에서 밖으로 손을 내밀어 주는 동작이다. (4) 동작은 오른손의 1지와 5지 끝을 맞대어 동그라미 모양을 만들고, (3)동작과 동일하게 몸에서부터 밖으로 손을 내밀어주는 동작이다. (5) 동작은 오른손을 주먹을 쥐고, 2지를 위로하고 3지를 반정도 들어서 'ㄴ' 자모양을 만들어 준 뒤 몸에서부터 밖으로 손을 내밀어주는 동작이다. (6) 동작은 오른손을 편 상태에서 모든 손가락 끝을 세워준 뒤 손바닥 날 방향이 정면을 향하게 한다. 오른손을 입술 쪽으로 당기면서 2지를 입술에 두 번 대는 동작이다. (7) 동작은 오른손을 펴서

머리높이까지 위치하고 손바닥을 밑으로 향하면서 오른손을 내려주는 동작이다. (8) 동작은 손끝이 밖으로 향하게 위치하고 모로 세운 왼 손바닥에 오른손 주먹의 1지를 펴서 옆면을 댄 다음, 오른손 주먹의 1지와 5지를 펴고, 약간 구부려 끝이 밖으로 향하게 하여 아래로 내리는 동작이다. (9) 동작은 왼손 주먹의 1지와 2지를 펴서 끝이 오른쪽으로 향하게 모로 세우고, 오른 주먹의 1지와 2이를 펴서 끝이 위로하고, 등이 밖으로 향하게 세워 왼 주먹의 2지 밑면에 두 번 대는 동작이다. (10) 동작은 오른손 주먹의 1지와 2지를 약간 구부려 끝이 두 눈으로 향하게 하여 왼쪽으로 2번 돌려주는 동작이다. (11) 동작은 두 손바닥의 방향을 바꾸어 각각의 손바닥을 쓸어주는 동작이다. (12) 동작은 오른 주먹을, 바닥이 밖으로 향하게 오른뺨에 대고 돌려 손등이 밖으로 향하게 한다. (13) 동작은 오른 주먹의 1지와 2지를 약간 구부려 끝이 두 눈으로 향하게 하여 왼쪽으로 돌린 다음, 두 주먹의 엄지를 펴서 동시에 아래로 내린다. (14) 동작은 오른 주먹의 1지와 2지를 펴서 끝바닥으로 코밑을 두 번 스쳐 내린다. (15) 동작은 오른손을 펴서 손등이 밖으로 향하게 구부려 왼쪽 어깨 앞에서 오른쪽 허리로 내린 다음, 오른 주먹의 1지와 2지를 펴서 끝 바닥으로, 손등이 밖으로 손끝이 오른쪽으로 향하게 편 왼 손등을 두드리고, 손바닥이 아래로 향하게 반쯤 구부린 오른손을 가슴 앞에서 약간 내리다가 멈춘다. (16) 동작은 손바닥이 위로 향하게 편 왼손 위에 손바닥이 아래로 향하게 편 오른손을 놓고, 손가락을 두 번 부드럽게 구부렸다 편 다음 오른손 2지로, 손바닥이 위로 향하게 편 왼손 바닥을 전후로 두 번 문지른다. (17) 동작은 오른 주먹의 1지와 2지를 펴서 끝이 눈으로 향하게 하여 왼쪽에서 오른쪽으로 옮기면서 두 번 구부린 다음, 오른손의 2지로 손바닥이 위로 향하게 편 왼 손바닥을 전후로 두 번 문지른다. (18) 동작은 오른손 손가락을 모두 편 후 손 날 반대 방향을 이마에 대고, 두 주먹 1지를 펴서 오른쪽으로 눕혔다가 세운다. (19) 동작은 손등이 밖으로 손끝이 오른쪽으로 향하게 편 왼 손등을 오른 주먹의 1지와 2지 끝으로 두 번 두드린 다음, 손바닥이 위로 향하게 편 왼 손바닥에 오른 주먹의 1지와 2지를 펴서 등을 댄다. (20) 동작은 오른

주먹의 1지와 2지를 펴서 끝 바닥으로, 손등이 안으로 손끝이 오른쪽으로 향하게 편 왼 손등을 두 번 두드린 후 두 주먹의 1지를 펴서 가슴 중앙에서 배꼼으로 사각형을 만든다. (21) 동작은 손바닥이 왼쪽으로 향하게 세운 오른손의 1지 옆면을 이마 중앙에 댄 다음, 1지를 펴서 등이 옆으로 향하게 세운 두 주먹을 중앙으로 모아 두 팔이 교차하게 한다. (22) 동작은 손등이 밖으로 손끝이 오른쪽으로 향하게 편 왼 손등을, 오른 주먹의 1지와 2지를 펴서 끝으로 두드린 다음, 손바닥이 위로 향하게 편 왼 손바닥에 오른 주먹의 1지와 2지를 펴서 눕혔다가 오른쪽으로 내린다. (23) 동작은 손등이 밖으로 향하게 편 왼 손등을 오른 주먹의 1지와 2지를 펴서 끝으로 두 번 두드린 다음, 오른 주먹의 1지와 2지를 펴서 2지 옆면으로 왼 주먹이 손목을 두 번 두드린다. (24) 동작은 손끝이 위로 손바닥이 왼쪽으로 향하게 자연스럽게 편 오른손의 1지 옆면을 이마 중앙에 댄 다음, 왼 주먹의 5지를 펴서 바닥이 밖으로 향하게 세우고, 오른 손바닥으로 왼 주먹 등을 두 번 좌우로 부드럽게 스쳐 낸다.

그림 15는 국립국어원 한국수어사전을 이용해 수어 동작 분석 방법을 나타낸다.

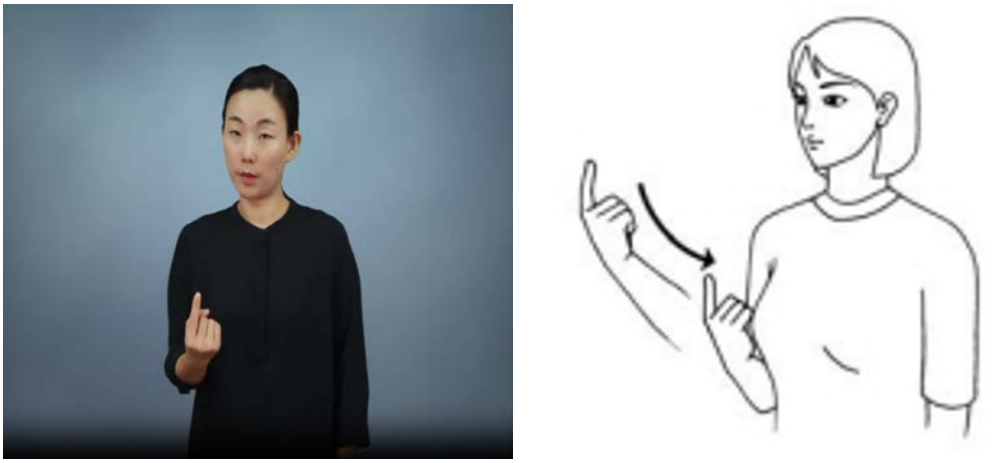


그림 15 한국수어사전을 이용한 수어 동작 분석

[출처, 한국수어사전, https://sldict.korean.go.kr/front/main/main.do]

제 2절 기계학습을 위한 골격 데이터 추적

본 절에서는 수어 동작 인식을 위한 기계학습 모델 개발의 핵심인 골격 데이터 추적 과정을 상세히 설명한다. 이 과정은 OpenCV와 MediaPipe를 활용하여 수행되며, 추적된 골격 데이터를 활용하여 효과적인 학습 데이터를 구축하는 방법에 대해 소개하고, 합성곱 신경망(Convolutional Neural Network, CNN) 및 순환 신경망(Recurrent Neural Network, RNN) 모델의 구성과 설정에 대한 내용을 소개합니다.

2.1 RGB 영상처리 및 시각화

수어 동작 인식 모델의 핵심인 RGB 영상처리와 시각화에 있어서, OpenCV와 MediaPipe의 연동은 핵심적인 역할을 수행한다. OpenCV는 오픈 소스 기반의 컴퓨터 비전 및 영상처리 라이브러리로, 강력한 기능과 다양한 알고리즘을 제공한다. OpenCV를 이용한 RGB 영상처리는 동영상 데이터를 로드하고, 각 프레임에 대한 크기 조정, 색상 변환 등의 전처리를 수행한다. 특히, MediaPipe를 통해 골격 데이터를 추적하고, 추적된 골격 데이터를 프레임 골격에 그려내어 각 관절의 위치와 동작의 흐름을 정확하게 이해할 수 있으며, 데이터의 유효성을 확인하고, 동작의 일관성을 시각적으로 파악할 수 있다. 이러한 OpenCV와 MediaPipe의 연동을 통해 RBG 영상의 처리와 시각화가 원활하게 이루어지며, 데이터 처리와 추적이 상호 보완적으로 이뤄지면서 모델 학습에 활용되는 데이터 셋은 정교한 특징을 지닐 수 있다. 이는 모델이 수어 동작을 더욱 정확하게 이해하고 학습하도록 도와주며, 최종적으로 높은 동작 인식 정확도를 달성할 수 있다.

그림 16은 OpenCV와 MediaPipe를 이용한 골격 데이터 추적 및 데이터 시각화를 나타낸다.

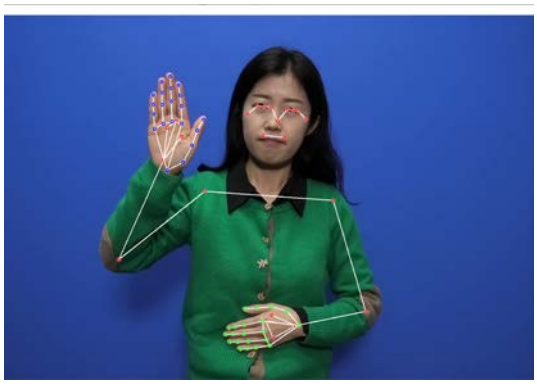



그림 16 OpenCV와 MediaPipe를 이용한 골격 데이터 추적 및 시각화

2.2 MediaPipe를 활용한 골격 데이터 추적

MediaPipe를 활용한 골격 데이터 추적에는 기계 학습 모델을 사용하여 2D 이미지 또는 동영상에서 사람 신체의 특징점 (landmark)의 3D 위치를 추정하고, 추정된 위치를 골격 데이터로 하여 동작 추적, 제스처 인식과 같은 다양한 응용 프로그램에 활용이 가능하다. MediaPipe는 신체의 특징점 위치를 추정할 수 있는 BlazePose 모델[25], 얼굴의 특징점 위치를 추정할 수 있는 BlazeFace 모델[26], 손가락 관절의 특징점 위치를 추정할 수 있는 BlazeHand 모델[27]을 가지고 있다.

BlazePose모델은 코, 눈, 귀, 어깨, 팔꿈치, 손목, 엉덩이, 무릎, 발목 등 신체의 33개의 특징점의 위치를 추정하고, BlazeFace모델은 코, 눈, 입 등 얼굴의 438개 특징점의 위치를 추정한다. BlazePose architecture를 기반으로 제작된 BlazeHand모델은 각각의 손가락 관절을 따라 21개의 특징점 위치를 추정할 수 있다. 특히, MediaPipe의 Holistic 모델은 Blaze 모델을 기반으로 얼굴, 손, 상반신, 하반신의 골격 데이터를 효과적으로 추적하는 모델이다.

그림 17은 Holistic 모델을 통해 획득 가능한 특징점 위치를 나타낸다.

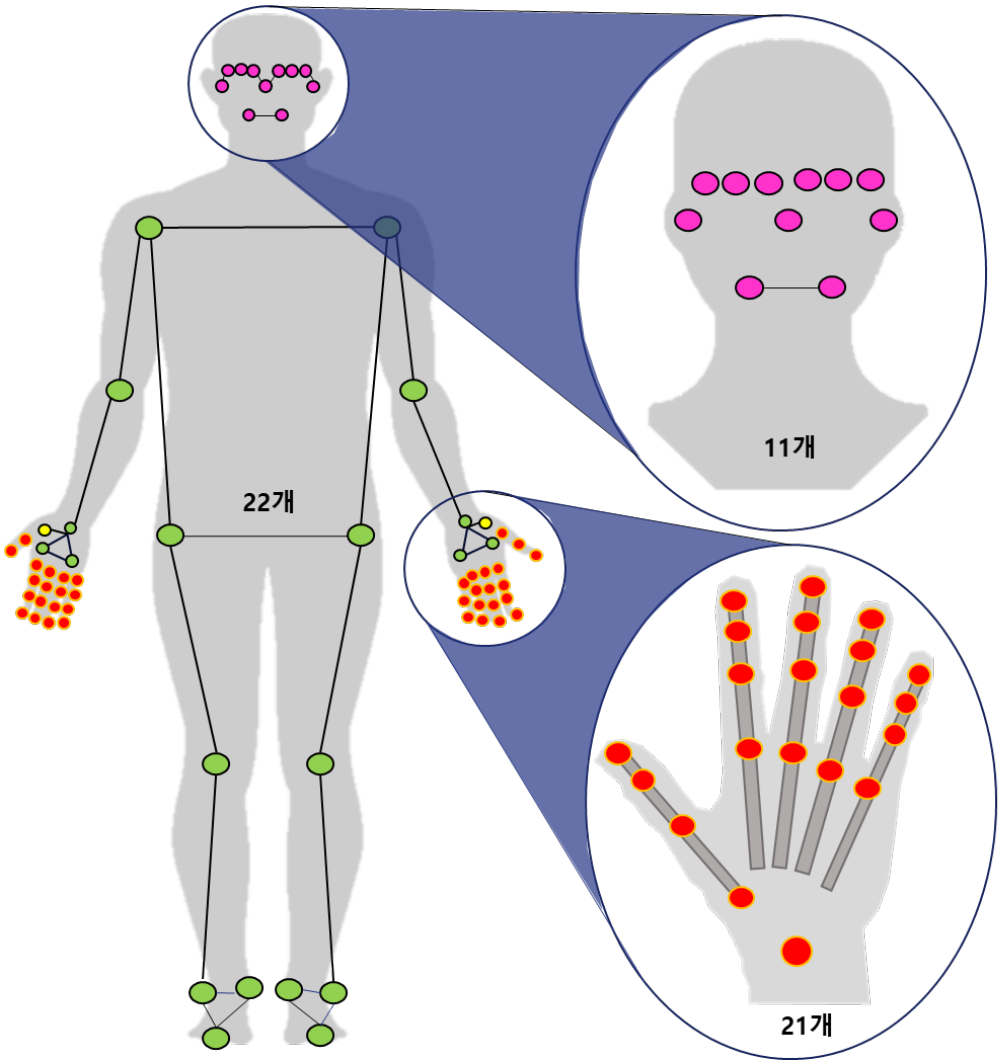


그림 17 MediaPipe Blaze 모델을 이용해 획득 가능한 특징점

MediaPipe의 사용되는 Holistic 모델은 합성곱 신경망 기반으로 제작되었고, 대규모 데이터 세트를 통한 훈련, 정규화 및 전이 학습과 같은 수준 높은 기계 학습 기술을 사용하여 이미지의 가려짐 현상 및 데이터 일반화 문제를 해결하고, 실시간으로 정확한 위치 추정이 가능하다. 또한, MediaPipe 모델의 경우 사람의 자세를 추정하는데 널리 사용되는 벤치마크 COCO 데이터 세트에서 객체 감지 모델 성능을

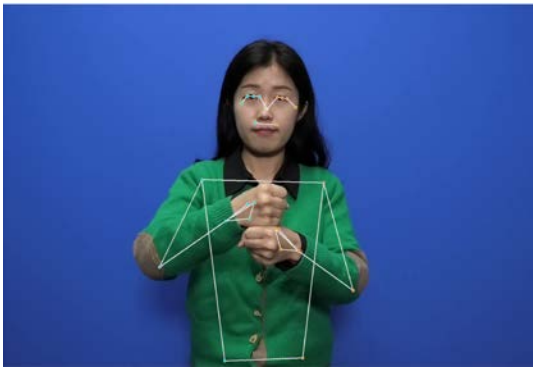
측정한 결과 1인 포즈 추정작업의 경우 0.658, 1인 이상의 포즈 추정 작업의 경우 0.547의 mAP(mean Average Precision) 점수를 달성한다. 이러한 결과는 다른 포즈 추정 모델의 결과와 비슷하거나 더 좋은 결과를 도출한다. 표 3은 모델의 품질 평가를 위해 요가, 댄스, HIIT의 서로 다른 카테고리를 3가지 다른 테스트 데이터 세트로 사용하여 각각의 추정 모델을 통해 획득한 mAP 점수를 나타낸다.

표 3 여러가지 포즈 추정 모델로 획득한 모델 평가 점수

Method	요가(mAP)	춤(mAP)	HIIT(mAP)
BlazePose	68.1	73.0	74.0
AlphaPose	63.4	57.8	63.4
Apple Vision	32.8	36.4	44.5

평가 결과 MediaPipe의 Blaze 모델이 COCO 객체 감지 모델 성능 측정에서 우수한 검출력을 가지고 있으며, 연속 동작에서도 정확한 인식 및 실시간 연산이 가능함을 확인할 수 있다. 또한, MediaPipe를 활용한 골격 데이터 추적에는 Cross-platform 지원이 가능해 컴퓨터 데스크톱, 모바일 어플리케이션, 웹을 포함하여 다양한 플랫폼에서 사용이 가능하고 호환성 문제나 성능 제한에 대한 걱정 없이 비전 응용 프로그램을 쉽게 배포할 수 있고, 다양한 오픈소스와 API를 통해 제약 없이 새로운 모델을 추가하고, 소프트웨어를 자유롭게 사용, 수정, 배포가 가능하다.

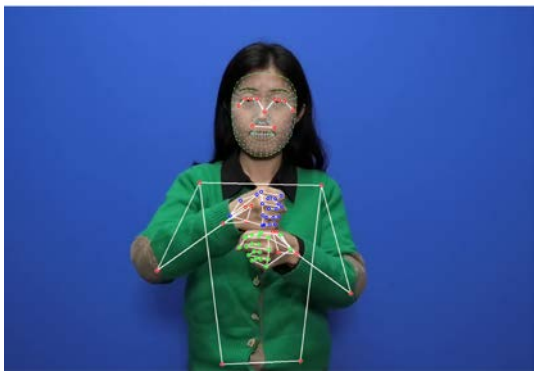
그림 18은 Blaze 모델을 통해 획득한 특징점 위치를 나타낸다.



(a) BlazePose Model

(B) BlazeFace Model

(c) BlazeHand Model



(d) Custom Model

그림 18 Blaze 모델을 통해 획득한 골격 데이터

이에 본 논문에서는 MediaPipe를 이용하여 골격 데이터를 추적하고, 추적된 골격 데이터를 활용해 학습 데이터를 구축하는 방안을 제안한다. 학습 데이터는 추적된 골격 데이터의 위치 데이터(Raw-data)와 위치 데이터를 기반으로 획득 가능한 벡터 데이터(Feature-data)로 구성한다. 추가적으로 인식 모델의 효과적인 사용을 위해 위치 변화에 따른 크기 정보를 계산하여 사용자의 움직임 추적한다.

제 3절 수어 동작 인식 모델 개발

본 절에서는 수집된 골격 데이터를 이용하여 학습 데이터를 구축하는 방안에 대해 설명하고, 골격 데이터를 이용한 기계학습 모델의 구조 및 구성, 학습 방안, 학습 모델 성능을 기술한다.

3.1 골격 데이터를 이용한 학습 데이터 구축

MediaPipe를 통해 추적된 골격 데이터는 위치 데이터를 나타내는 Raw-data와 위치 데이터를 기반으로 제작한 벡터 데이터인 Feature-data로 구성하여 학습 데이터를 구축한다.

Raw-data의 경우 원본 동작의 세부적인 특징과 움직임이 담겨 있어 동작의 고유한 패턴 및 세밀한 움직임을 포함하고, 벡터 데이터만으로는 추적이 어려운 세부 정보를 손실 없이 전달할 수 있는 장점이 있지만 데이터의 용량 및 카메라의 위치 및 방향에 따른 영향으로 발생되는 노이즈로 인하여 모델이 불필요한 정보를 학습하는 단점이 존재한다.

Feature-data의 경우 차원 축소 및 효율적인 학습이 가능하고, 실시간 처리 용이성이 있어 모델이 학습한 특징을 바탕으로 실제 동작을 빠르게 분석하고 인식할 수 있다. 특히, 적은 데이터를 가지고 동작을 파악하고, 카메라의 위치 및 방향에 따른 영향을 덜 받아 노이즈로 인한 영향을 최소화하고 모델의 정확도를 높일 수 있는 장점이 있지만 원본 데이터에 비해 세부적인 움직임이나 동작의 특징을 놓칠 수 있다. 이에 본 연구에서는 학습 데이터를 Raw-data와 Feature-data를 따로 구축하고, 모델에 적용시켜 모델의 성능 향상을 파악하고 최종적인 모델의 선택에 활용한다.

Raw-data는 Holistic 모델을 통해 몸의 해당하는 특징점 22개, 왼손에 해당하는 특징점 21개, 오른손에 해당하는 특징점 21개, 얼굴에 해당하는 특징점 11개, 위치

변화의 크기 정보 1개로 구성된다. 각각의 특징점은 X, Y, Z 축으로 동영상 프레임당 225개로 학습 데이터를 구축한다.

Feature-data는 Holistic 모델을 통해 몸의 해당하는 특징점 8개, 왼손에 해당하는 특징점 21개, 오른손에 해당하는 특징점 21개, Face 모델을 통해 얼굴에 해당하는 특징점 15개, 위치 변화의 크기 정보 1개로 구성된다. 그림 19는 Feature-data에 사용된 특징점들을 나타낸다. 표 4는 학습 데이터의 요약을 나타낸다.

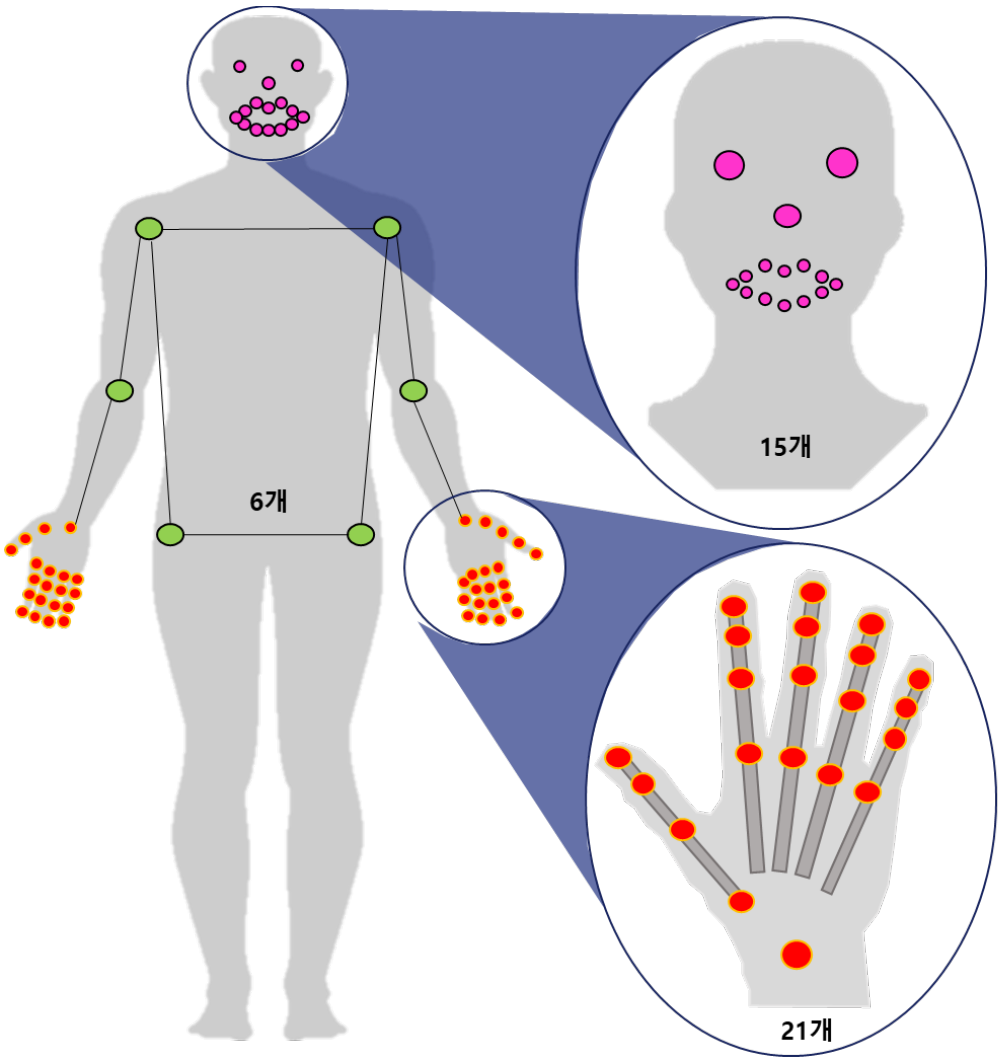
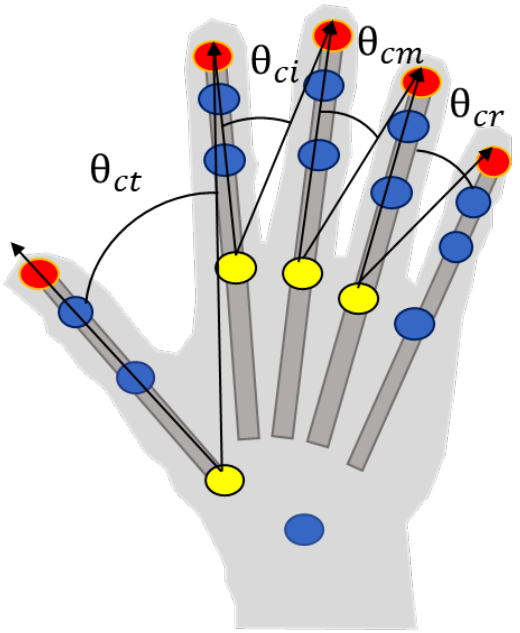


그림 19 Feature-data 구축을 위해 사용된 특징점

표 4 학습 데이터 요약

Data-Set	Property	Description	Unit
Raw-data	특징점	MediaPipe Holistic모델을 활용하여 획득 가능한 골격데이터	X : 75개 Y : 75개 Z : 75개
Feature-data	손가락 사이 각도	손허리손가락관절과 손가락 끝점의 벡터를 이용한 각도 값	왼손: 4개 오른손: 4개
	손가락 구부림 각도	위지절간관절과 손허리손가락관절의 벡터를 이용한 각도 값	왼손: 5개 오른손: 5개
	손가락 길이	손목점과 손가락의 끝점의 거리	왼손: 5개 오른손:5개
	손목 각도	손목점으로부터 손허리관절과 팔꿈치 점의 벡터를 이용한 각도 값	왼손: 1개 오른손: 1개
	팔꿈치 각도	팔꿈치점으로부터 손목점고 어깨점의 벡터를 이용한 각도 값	왼손: 1개 오른손: 1개
	어깨 벌어짐 각도	어깨점으로터 팔꿈치점과 엉덩이점의 벡터를 이용한 각도 값	왼손: 1개 오른손: 1개
	몸의 좌우 기울기	양쪽 어깨점과 양쪽 엉덩이점의 벡터를 이용한 각도 값	1개
	몸의 앞뒤 기울기	엉덩이 중앙점으로부터 얼굴의 코점과 OpenCV Camera 좌표계의 -y축 벡터를 이용한 각도 값	1개
	팔 길이	손목점과 팔꿈치점 사이의 거리	왼손: 1개 오른손: 1개
	손목 어깨 길이	손목점과 어깨점 사이의 거리	왼손: 1개 오른손: 1개
	양쪽 손목의 거리	양쪽 손목점 사이의 거리	1개
	팔뚝 길이	팔꿈치점과 어깨점 사이의 거리	왼손: 1개 오른손: 1개
	어깨 길이	양쪽 어깨점 사이의 거리	1개
	상체 길이	어깨점과 엉덩이점 사이의 거리	왼쪽: 1개 오른쪽: 1개
	엉덩이 길이	양쪽 엉덩이점 사이의 거리	1개
	손과얼굴의 길이	손목점과 얼굴의 코점 사이의 거리	왼손: 1개 오른손: 1개
	입 벌림	윗입술점과 아랫입술점 사이의 거리	1개
	입 돌출	좌우 입술 중앙점과 상하 입술 중앙점 사이의 거리	1개
	고개 좌우 돌림	투영된 양쪽 눈점과 평면의 법선벡터를 이용한 각도 값	1개
	고개 앞뒤 젖힘	투영된 눈의 중심점과 평면의 법선벡터를 이용한 각도 값	1개
	고개 좌우 기울기	투영된 양쪽 눈점과 평면의 법선벡터를 이용한 각도 값	1개

Feature-data의 데이터는 1. 손가락 사이 각도(8개), 손가락 구부림 각도(10개), 손가락 길이(10개), 손목 각도(2개), 팔꿈치 각도(2개), 어깨 벌어짐 각도(2개), 몸의 좌우 및 앞뒤 기울기(2개), 팔 길이(2개), 팔뚝 길이(2개), 몸 높이(2개), 상체 길이(2개), 양손 사이의 거리(1개), 입술 개구(1개), 입술 돌출(1개), 고개 좌우 돌림 각도(1개), 고개 앞뒤 젖힘 각도(1개), 고개 좌우 기울기(1개)로 동영상 프레임당 54개로 학습 데이터를 구축한다. Feature-data를 계산하는 식은 다음과 같다.



MCP Joint : 손허리손가락관절

MCP Joint → MCP Finger tip : Vector 1

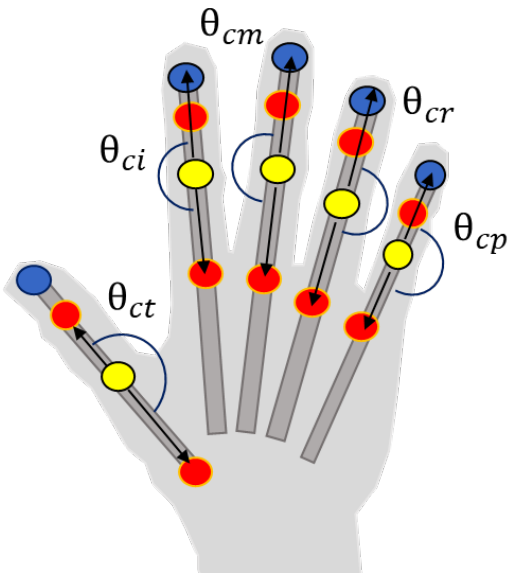
MCP Joint → Next Finger tip : Vector 2

$$\boldsymbol{\theta_{cx} = cos^{-1}\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}\right),}$$

$$\boldsymbol{x = t,\ i, m,\ r}$$

그림 20 손가락 사이 각도

손가락 사이 각도는 수어 동작에서 움직이는 손 동작을 설명하는데 용이하다 손가락의 MCP 관절과 TIP 점을 이용하여 벡터를 생성하고 생성된 벡터의 사이각을 계산한다.



PIP Joint : 위지절간관절

PIP Joint → PIP Finger DIP : Vector 1

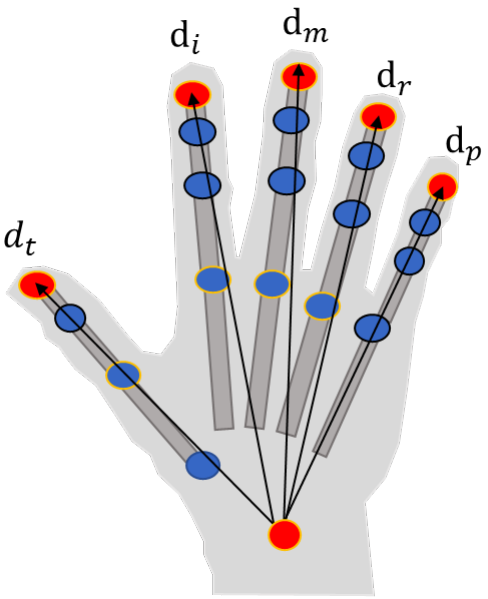
PIP Joint → PIP Finger MCP : Vector 2

$$\boldsymbol{\theta_{cx} = cos^{-1}\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}\right),}$$

$$\boldsymbol{x = t, i, m, r, p}$$

그림 21 손가락 구부림 각도

손가락 구부림 각도는 수어 동작에서 손의 주먹을 쥐는 동작이나 펴는 동작을 설명하는데 용이하다. 손의 근위지골과 중간지골 사이의 관절인 PIP 관절을 사용하여 위쪽으로 DIP 관절과 아래쪽으로 MCP 관절을 이용하여 벡터를 생성하고 생성된 벡터의 사이각을 계산한다.



D_f=Finger TIP position(5) – Wrist Position

f = Thumb, Index, Middle, Ring, Pinky TIP

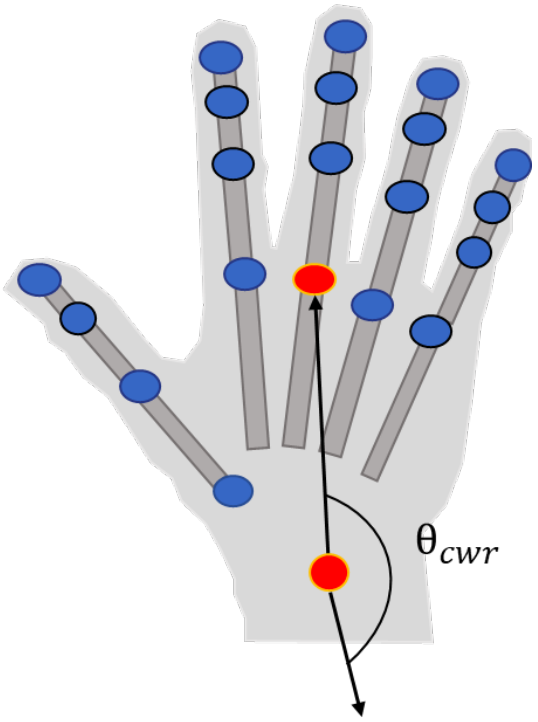
Max = D_f 중 가장 큰 값

$$D_f = \sqrt{(xk - xw)^2 + (yk - yw)^2 + (zk - zw)^2}$$

$D_f \div MAX$

그림 22 손가락 길이

손가락 길이는 신체 비율 및 수어를 표현하면서 발생되는 개인의 차이를 설명하는데 용이하다. 손목 점으로부터 각 손가락의 TIP점의 거리를 계산하고 가장 긴 거리를 나누어 주면서 정규화를 통해 길이의 비율을 계산한다.



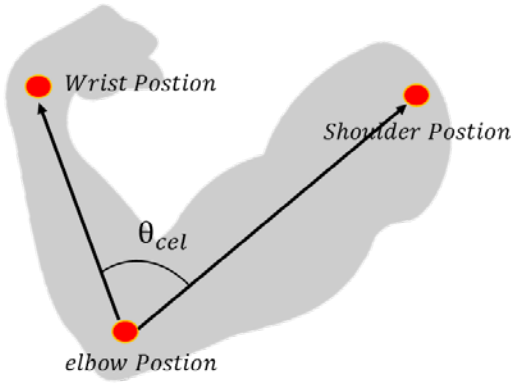
Wrist → Middle MCP : Vector 1

Wrist → Elbow : Vector 2

$$\boldsymbol{\theta_{cwr} = cos^{-1}\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}\right)}$$

그림 23 손목 각도

손목 각도는 수어 동작 시 손의 움직임을 설명하는데 용이하다. 손목점에서부터 위쪽으로 중지에 있는 MCP 관절과 아래쪽으로 팔꿈치점을 이용하여 벡터를 생성하고 생성된 벡터의 사이각을 계산한다.



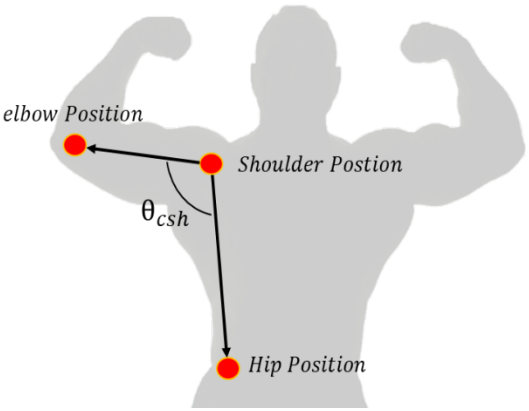
Elbow → Wrist : Vector 1

Elbow → Shoulder : Vector 2

$$\boldsymbol{\theta_{cel} = cos^{-1}\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}\right)}$$

그림 24 팔꿈치 각도

팔꿈치 각도는 수어 동작 시 팔의 움직임 및 팔꿈치 굽힘 정도를 설명하는데 용이하다. 팔꿈치 점으로부터 위쪽으로 손목점과 아래쪽으로 어깨점을 이용하여 벡터를 생성하고 생성된 벡터의 사이각을 계산한다.



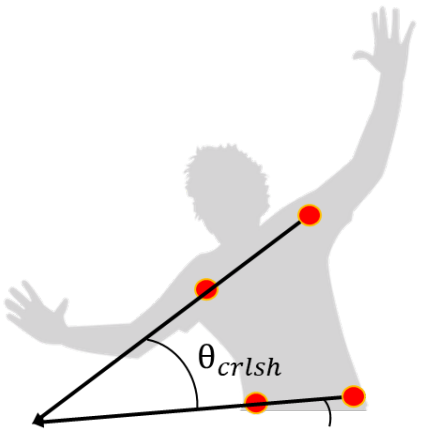
Shoulder → Elbow : Vector 1

Shoulder → hip : Vector 2

$$\boldsymbol{\theta_{csh} = cos^{-1}\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}\right)}$$

그림 25 어깨 벌림 각도

어깨 벌어짐 각도는 수어 동작 시 팔의 움직임을 설명하는데 용이하다. 어깨점으로부터 위쪽으로 팔꿈치점과 아래쪽으로 엉덩이점을 이용하여 벡터를 생성하고 생성된 벡터의 사이각을 계산한다.



Right Shoulder → Left Shoulder : Vector 1

Right hip → Left hip : Vector 2

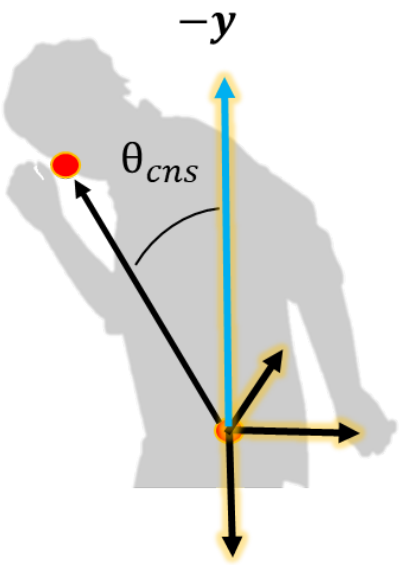
몸이 좌측으로 기울어진 경우 (+)

몸이 우측으로 기울어진 경우 (-)

$$\boldsymbol{\theta_{crlsh} = cos^{-1}\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}\right)}$$

그림 26 몸의 좌우 기울기

몸의 좌우 기울기 수어 동작 시 상체의 움직임을 설명하는데 용이하다. 오른쪽 어깨점으로부터 왼쪽 어깨점으로 향하는 벡터와 오른쪽 엉덩이 점으로부터 왼쪽 엉덩이점으로 향하는 벡터 사이의 각도를 계산하고, 왼쪽 어깨점과 오른쪽 어깨점의 높이를 비교하여 좌우를 구별한다.



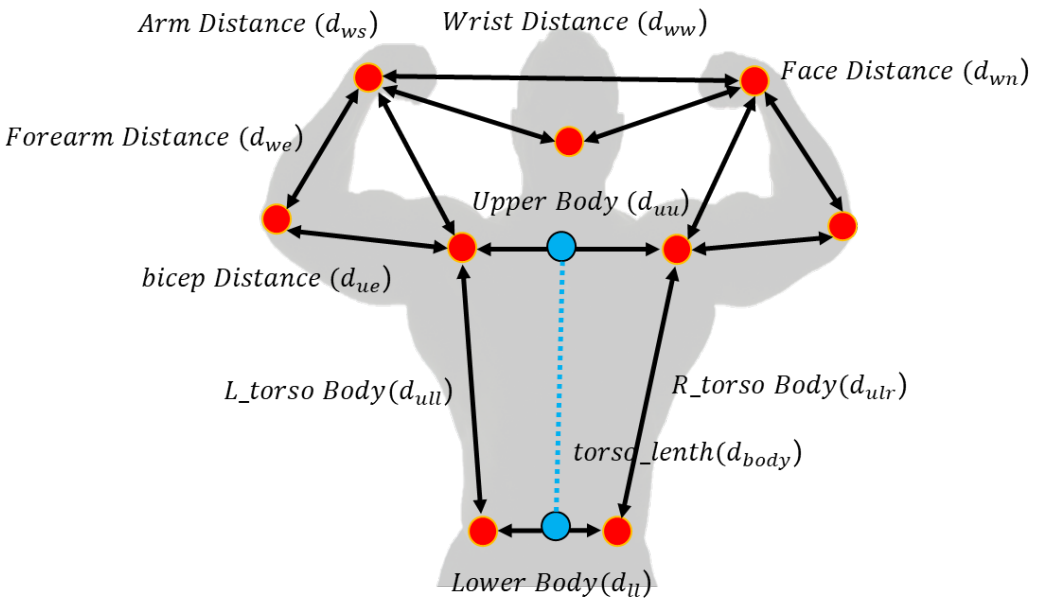
Middle hip → nose : Vector 1

OpenCV Camera (Y) : Vector 2

$$\boldsymbol{\theta_{cns} = cos^{-1}\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}\right)}$$

그림 27 몸의 앞뒤 기울기

몸의 앞뒤 기울기는 수어 동작 시 상체의 움직임을 설명하는데 용이하다. 왼쪽 엉덩이점과 오른쪽 엉덩이점을 사용하여 엉덩이에 중앙점을 계산하고 엉덩이 중앙점으로부터 코점으로 향하는 벡터를 생성한다. 이후, OpenCV Camera 좌표계를 사용하여 -y 축에 해당하는 벡터의 사이각을 계산한다.



d_{we} :Wrist position – Elbow Position, d_{ws} :Wrist position (Y) – Shoulder Position (Y)

d_{ww} : Wrist position (L) – Wrist Position (R), d_{ue} : Upper Body Position – Elbow Position

d_{uu} : Upper Body position (L) – Upper Body Position (R)

d_{ull} : Upper Body position (L) – Lower Body Position (L)

d_{ulr} : Upper Body position (R) – Lower Body Position (R)

d_{ll} : Lower Body Position (L) – Lower Body Position (R)

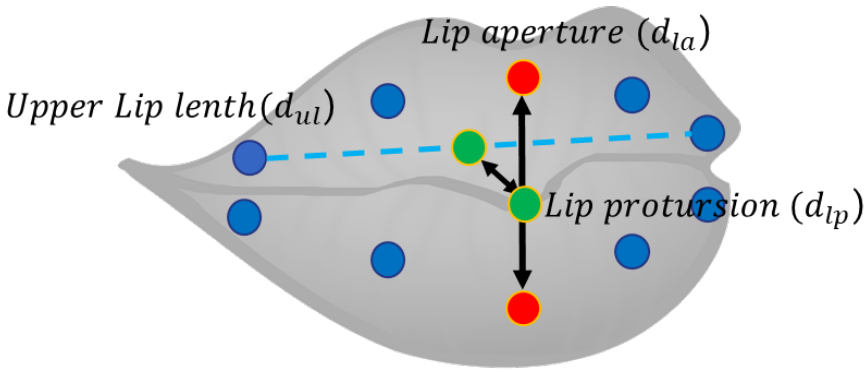
d_{wn} : Wrist position – Nose Position, d_{body} : Upper Body Middle – Lower Body Middle

$d_i = \sqrt{(xi - x)^2 + (yi - y)^2 + (zi - z)^2}$,
$i = we, ws, ww, ue, uu, ull, ulr, ll, wn$
$d_i \div (2 * d_{body})$

그림 28 팔, 팔뚝, 몸, 상체 길이

상체의 길이의 경우 신체 비율 및 수어 동작을 표현하면서 발생되는 개인의 차이를 설명하는데 용이하고, 양손을 높게 드는 동작, 양손을 모으는 동작, 팔을 올리고 내는 동작, 양손을 얼굴에 가까이 가는 동작 등을 설명하는데 용이하다. 엉덩이

중앙점과 어깨 중앙점을 연결하는 거리의 2배를 통해 길이의 비율을 계산한다.



d_{la} : Top position - Bottom position

d_{lp} : Middle position(Left-Right) – Middle position (Top-Bottom)

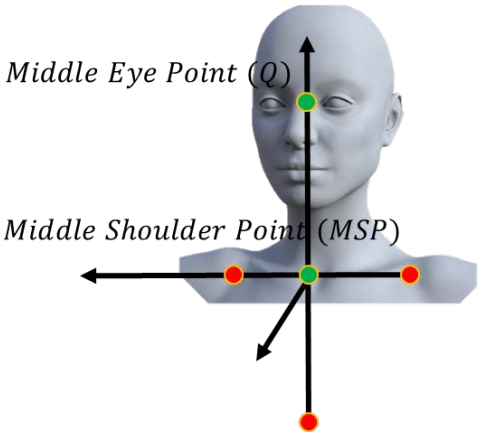
d_{ul} : Upper Lip Left Position – Upper Lip Right Postion

$d_i = \sqrt{(xi - x)^2 + (yi - y)^2 + (zi - z)^2}$, $i = la, lp$

$d_i \div d_{ul}$

그림 29 입술 개구 및 돌출 길이

입술의 개구 및 돌출은 입을 벌려 표현하는 수어 동작이나 입술을 내밀어 표현하는 수어 동작을 설명하는데 용이하다. 입술의 위쪽 끝점과 아래쪽 끝점의 거리를 계산하여 입술 개구를 계산하고, 입술의 양쪽 끝점에 중앙점과 입술의 위아래 끝점의 중앙점을 구한 후 중앙점과 중앙점 사이의 거리를 통해 입술의 돌출을 계산한다. 입술의 양쪽 끝점의 거리를 이용하여 길이의 비율을 계산한다.



$$Q: Coordinate\ Transformation$$
$$Coordinate\ [OXYZ] \rightarrow$$
$$Coordinate\ [MSP|V_{Side}|V_{Front}|V_{Top}]$$
$$New\ Coordinate\ Q\ =\ (C_X, C_Y, C_Z)$$

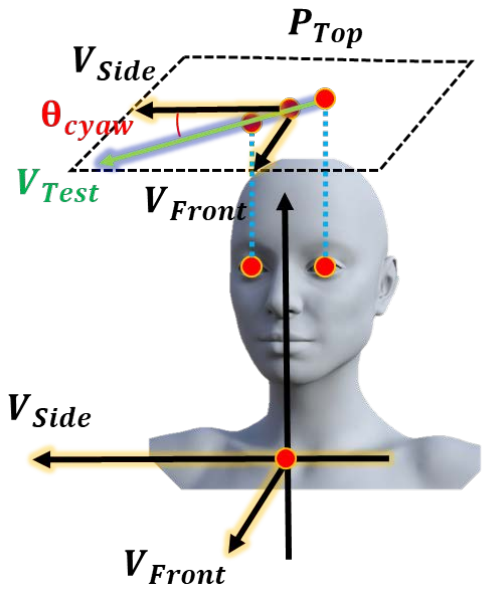
$$V = Q - MSP$$
$$C_X = V \cdot V_{Side} \ |\ C_Y = V \cdot V_{Front} \ |$$
$$C_Z = V \cdot V_{Top}$$

$$Q = (X, Y, Z)$$
$$= MSP + C_X \times V_{Side} + C_Y \times V_{Front}$$
$$+ C_Z \times V_{Top}$$

그림 30 머리자세[좌표계 변환을 이용하 투영점 계산]

머리 자세는 수어 동작 시 머리의 움직임을 설명하는데 용이하다. 엉덩이 중앙점과 어깨 중앙점을 이용하여 기준이 되는 Top 벡터를 구하고, 왼쪽 어깨점과 오른쪽 어깨점을 이용하여 기준이 되는 Side 벡터를 구하고, Top 벡터와 Side 벡터의 외적을 통해 기준이 되는 Front 벡터를 구한다. 각각의 기준이 되는 Top, Side, Front 벡터는 법선벡터로 어깨의 중앙점을 지나는 P_{Top}, P_{side}, P_{front} 평면을 생성하고, 특징점을 평면에 투영하여 새로운 벡터를 생성하고, 생성된 벡터와 평면의 기준이 되는 벡터를 이용하여 고개 좌우 돌림 각도, 고개 앞뒤 젖힘 각도, 고개 좌우 기울기를 계산한다.

고개 좌우 돌림 각도의 경우 P_{Top} 에 양쪽 눈 점을 투영시켜 벡터를 생성하고 Front 벡터와의 내적을 통해 머리의 좌우 움직임을 추적한 후, Side 벡터와의 사이각을 계산한다. 고개 앞뒤 젖힘 각도의 경우 P_{side} 에 눈의 중심점을 투영시켜 어깨중심점으로부터 투영점으로 향하는 벡터를 생성하고 Front 벡터와의 내적을 통해 머리의 상하 움직임을 추적한 후, Top 벡터와의 사이각을 계산한다. 고개 좌우 기울기의 경우 P_{front} 에 양쪽 눈 점을 투영시켜 벡터를 생성하고 Top 벡터와의 내적을 통해 좌우 움직임을 추적한 후 Side 벡터와의 사이각을 계산한다.



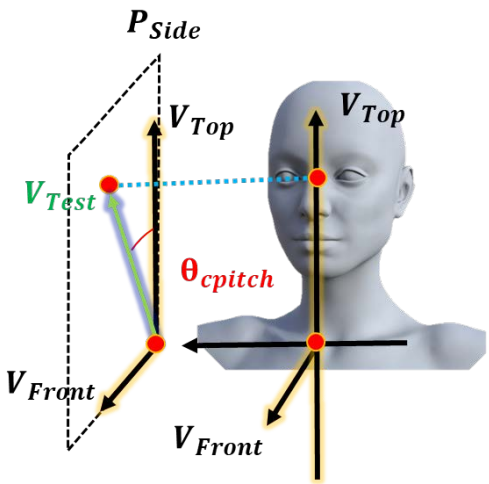
$$P_n \rightarrow P_{Top}\ [Projection]$$
$$P_n = MSP + C_X \times V_{side} + C_y \times V_{Front}$$
$$n = LEP, REP$$

$$V_{test} = REP - LEP$$
$$V_{unit} = \frac{V_{test}}{\|V_{test}\|}$$

$$Val = V_{unit} \cdot V_{front}$$
$$Val \leq 90°\ (+)\ \text{#right}$$
$$Val > 90°\ (-)\ \text{#left}$$

$$\theta_{cyaw} = cos^{-1}\left(\frac{\vec{v}_{unit} \cdot \vec{v}_{side}}{|\vec{v}_{unit}||\vec{v}_{side}|}\right) \times val(sign)$$

그림 31 고개 좌우 돌림각도



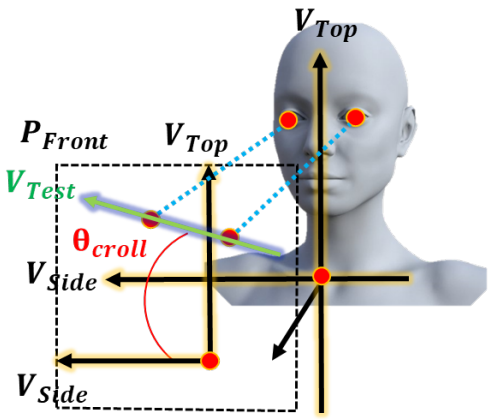
$$P_n \rightarrow P_{Side}\ [Projection]$$
$$P_n = MSP + C_Y \times V_{Front} + C_Z \times V_{Top}$$
$$n = MEP$$

$$V_{test} = P_{MEP} - MSP$$
$$V_{unit} = \frac{V_{test}}{\|V_{test}\|}$$

$$Val = V_{unit} \cdot V_{front}$$
$$Val \leq 90°(+)\ \text{#front}$$
$$Val > 90°(-)\ \text{#back}$$

$$\theta_{cpitch} = cos^{-1}\left(\frac{\vec{v}_{unit} \cdot \vec{v}_{top}}{|\vec{v}_{unit}||\vec{v}_{top}|}\right) \times val(sign)$$

그림 32 고개 앞뒤 젖힘 각도



$$P_n \rightarrow P_{Front}\,[Projection]$$
$$P_n = MSP + C_x \times V_{side} + C_Z \times V_{Top}$$
$$n = LEP, REP$$

$$V_{test} = REP - LEP$$
$$V_{unit} = \frac{V_{test}}{\|V_{test}\|}$$

$$Val = V_{unit} \cdot V_{Top}$$
$$Val \leq 90°(+) \quad \text{\#left}$$
$$Val > 90°(-) \quad \text{\#right}$$

$$\theta_{croll} = cos^{-1}\left(\frac{\vec{v}_{unit} \cdot \vec{v}_{side}}{|\vec{v}_{unit}||\vec{v}_{side}|}\right) \times val(sign)$$

그림 33 고개 좌우 기울기

3.2 학습 데이터 전처리

구축된 학습 데이터는 데이터 전처리를 통해 모델의 성능에 영향을 끼칠 수 있는 정보들을 제거한다. 모델의 성능에 영향을 주는 데이터로 수어 동작을 위해 대기하는 동작과 수어 동작을 표현한 후 멈춰 있는 동작들은 모델을 학습하는데 노이즈를 발생한다. 이를 해결하기 위해 벡터의 변화 정보량을 추가하여 동영상 속에서 사람의 움직임의 정도를 측정한다.

벡터의 변화 정보는 이전 프레임과 이후 프레임의 차이를 계산하고, 정규화 작업을 통해 벡터의 크기를 사용하였다. 이후, 실제 동영상과 벡터의 변화 정보 비교를 통해 기준이 되는 벡터의 크기 값은 0.5로 설정하였다. 이후, 학습 데이터에서 벡터의 크기가 0.5 이하인 경우 해당 프레임에서 획득한 데이터 들을 제거하였다.

모델의 성능에 영향을 줄 수 있는 다른 데이터는 수어 동작을 표현하는 속도이다. 수어 동작을 표현하는 속도의 경우 시계열 데이터를 분석하는데 어려움이 있다. 이를 해결하기 위해 키프레임에 해당하는 수어 동작을 30 프레임으로 수집하여 Raw-data,

Feature-data를 새롭게 구성하였다. 그림 34는 데이터 전처리 과정을 나타낸다.

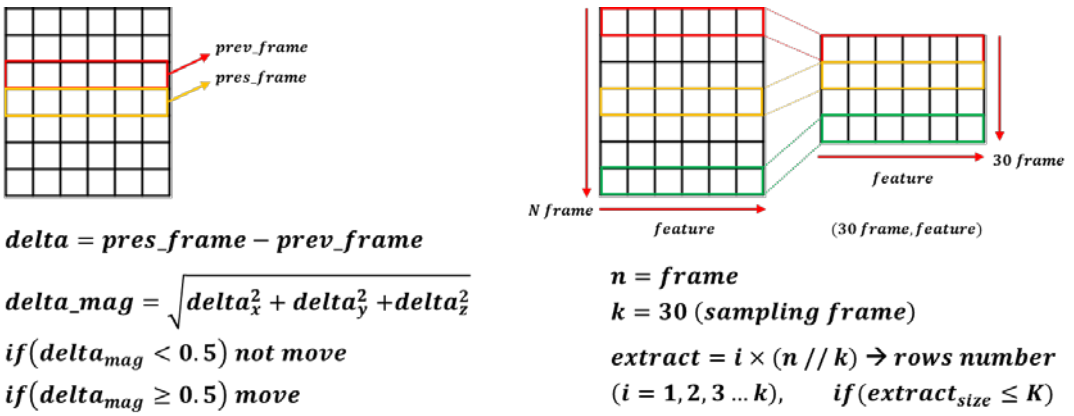


그림 34 데이터 전처리 과정

3.3 기계학습 모델 개발 및 평가

골격 데이터의 위치 데이터를 담고 있는 Raw-data와 위치 데이터를 기반으로 벡터 정보를 생성한 Feature-data를 이용하여 합성곱 신경망 모델과 순환 신경망 모델을 개발한다. 모델을 개발하기위해 Python Tensorflow keras 패키지를 사용하여 모델을 구성하고, python matplotlib[]를 사용하여 모델의 학습 과정 및 결과를 시각화 한다.

합성곱 신경망은 Sampling을 30 프레임으로 설정하여 모델에 입력데이터가 하나의 수어 단어에 해당하는 학습 데이터 전부를 데이터 셋으로 구성하였다. 모델의 구조는 4개의 콘볼루션 레이어와, 4개의 풀링 레이어, Flatten 레이어, Output 레이어로 구성되고, 필터 32개, 커널 사이즈 1개, 패딩(‘same’), 활성화 함수(‘softmax’), Optimizer(‘adam’), Loss(‘Sprase Categorical Cross Entropy’), Epoch(‘200’), 배치 사이즈(‘256’)로 학습이 진행된다.

모델 학습 과정은 프레임, 특징값(Raw-data 255개, Feature-data 54개), 단어 수(24개) * 단어당 동영상의 개수(15개)에 해당하는 전체 데이터 셋에서 Sampling으로

나눠진 (프레임 (30), 특징 값)들이 순차적으로 모델에 입력되고, 입력데이터에서 1D 콘볼루션 레이어를 통해 커널 사이즈만큼 추출하여 공간을 축약하고, max_pooling을 통해 특징일 추출한다. 출력된 데이터는 동일한 작업을 3회 수행을 통해 입력 데이터를 1까지 축소하고, Flatten 레이어와 Dense 레이어를 통해 결과를 추출한다.

그림 35는 합성곱 신경망 모델의 구조를 나타내고, 그림 36은 합성곱 신경망 모델의 결과를 나타낸다

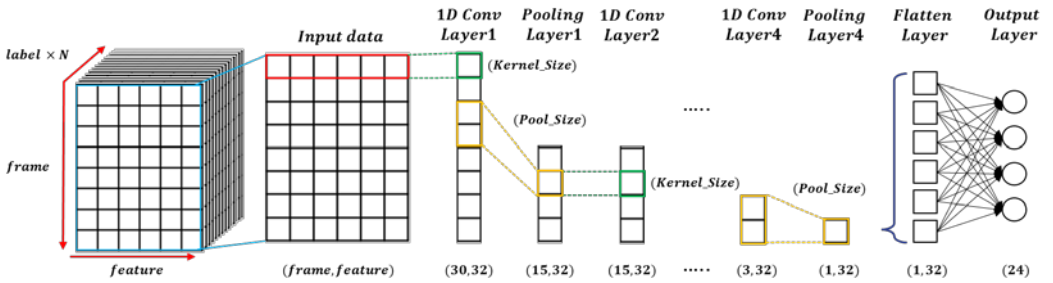
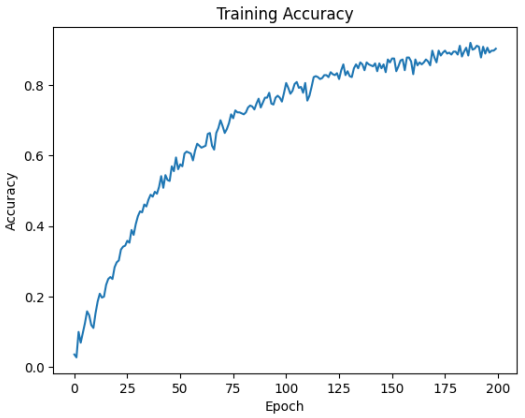
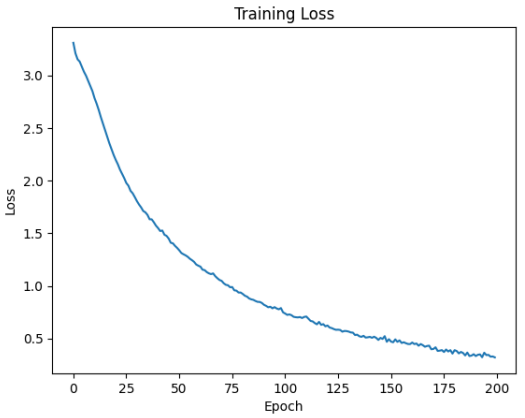


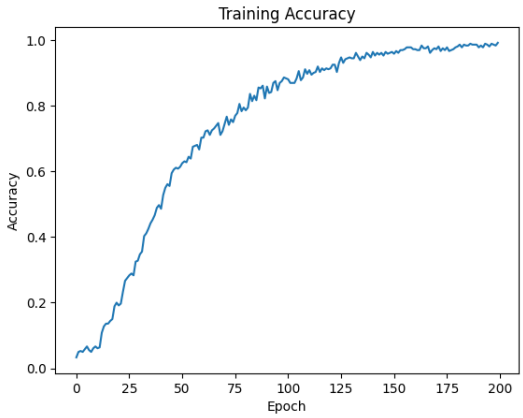
그림 35 CNN 모델 구조

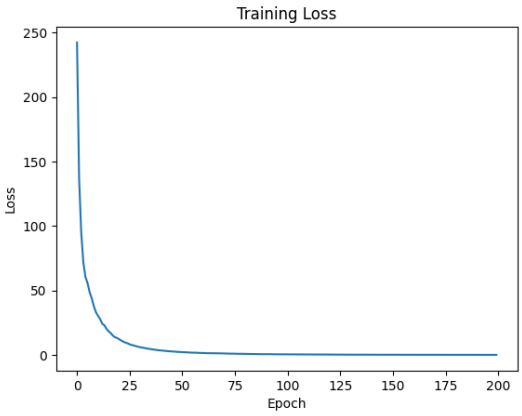


(a) Accuracy Raw-data



(b) Loss Raw-data





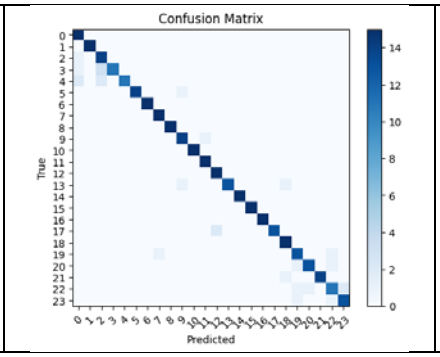
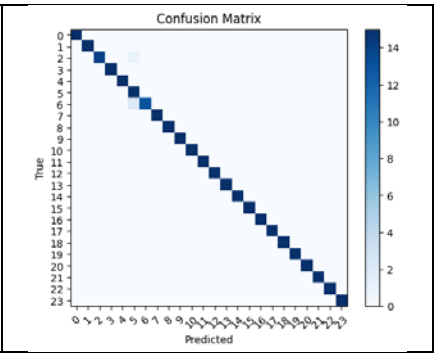
(c) Accuracy Feature-data **(d) Loss Feature-data**

그림 36 CNN 모델 결과

합성곱 신경망 모델의 Raw-data는 93.58 정확도를 나타내고, Feature-data는 99.75 정확도를 나타낸다. 합성곱 신경망 모델의 성능을 평가하기 위해 Confusion Matrix를 제작하고, precision, recall, F1-score를 측정한다. 표 5는 합성곱 신경망 모델의 성능 평가를 나타낸다.

표 5 CNN 모델 성능 분석

평가지표	Raw-data	Feature-data
Precision (정밀도)	**0.9447**	**0.9892**
Recall (재현율)	**0.9377**	**0.9958**
F1-score	**0.9375**	**0.9957**

Confusion Matrix		

순환 신경망의 경우 LSTM 모델을 사용한다. 입력데이터는 타임스텝과 특성의 수로 설정되며 이는 연속성을 반영하기 위한 입력데이터의 형태이다. LSTM 모델의 구조는 3개의 LSTM 레이어와, Output 레이어로 구성되고, 학습 노드 수는 32개, Return_Sequence=True 설정으로 다음 레이어에 시퀀스 데이터를 전달하고, 활성화 함수('softmax'), Optimizer('adam'), Loss('Sprase Categorical Cross Entropy'), Epoch('200'), 배치 사이즈('256')로 학습이 진행된다.

모델의 학습 과정은 프레임, 특징값(Raw-data 255개, Feature-data 54개), 단어 수(24개) * 단어당 동영상의 개수(15개)에 해당하는 전체 데이터 셋에서 프레임 시퀀스를 10으로 설정하여 (프레임 (10), 특징 값)들이 순차적으로 모델에 입력되고 입력 데이터에서 첫 번째 LSTM 레이어를 통해 (배치사이즈, 프레임, 노드 수) 형태의 Tensor를 생성한다 동일한 작업을 2번 시행을 통해 (프레임, 노드 수)형태의 Tensor 값을 통해 결과를 추출한다.

그림 37은 순환 신경망(LSTM) 모델의 구조를 나타내고, 그림 38은 순환 신경망 모델의 결과를 나타낸다.

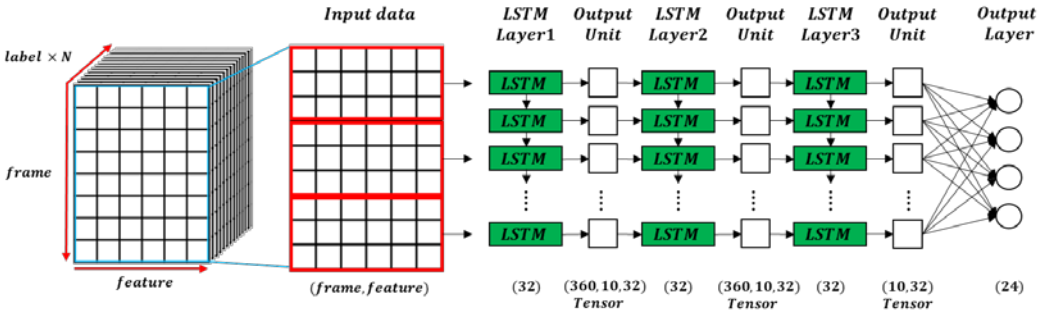
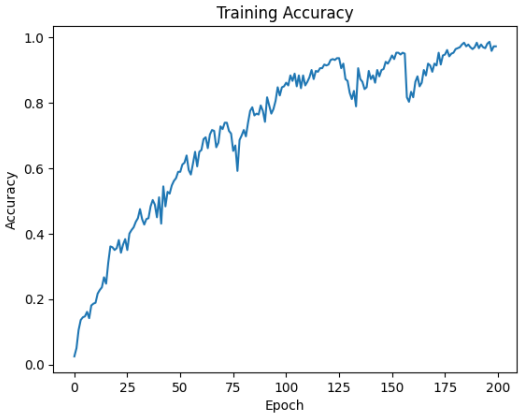
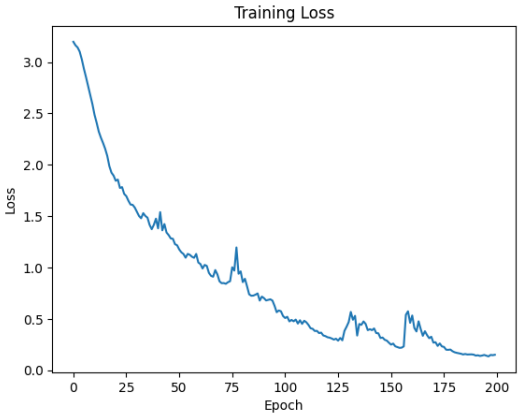


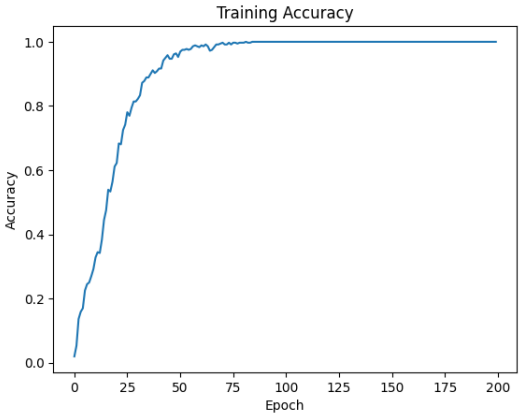
그림 37 LSTM 모델의 구조



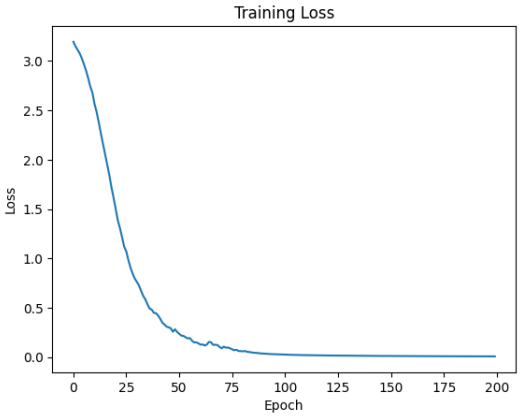
(a) Accuracy Raw-data



(b) Loss Raw-data



(c) Accuracy Feature-data

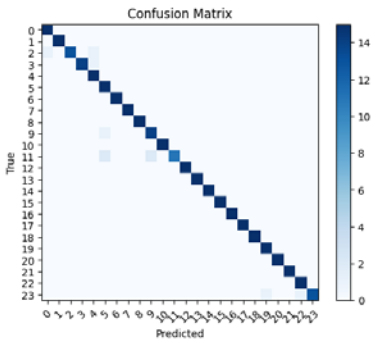
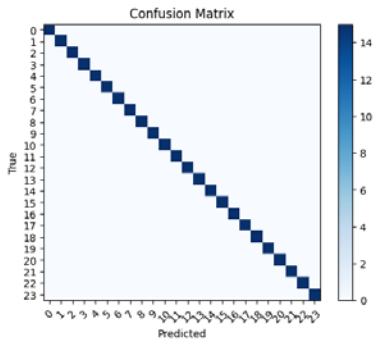


(d) Loss Feature-data

그림 38 LSTM 모델 결과

LSTM 모델의 Raw-data는 96.75 정확도를 나타내고, Feature-data는 99.92% 정확도를 나타낸다. LSTM 모델의 성능을 평가하기 위해 Confusion Matrix를 제작하고, precision, recall, F1-score를 측정한다. 표 6은 LSTM 모델 성능 평가를 나타낸다.

표 6 LSTM 모델 성능 분석

평가지표	Raw-data	Feature-data
Precision (정밀도)	**0.9781**	**0.9893**
Recall (재현율)	**0.9772**	**0.9995**
F1-score	**0.9777**	**0.9997**
Confusion Matrix		

제 4 장 수어 동작 인식 모델 구현 및 검증

제 1절 비교 및 검증

본 절에서는 제안된 수어 단어 인식을 위한 기계학습 모델의 성능 평가를 위해 실제 수어 단어 인식의 효과성을 검증하고자 한다. 이를 위해, 모델이 생성한 예측 값과 실제 수어 단어를 비교 및 분석하여 모델의 인식 정확도를 규명하고, 특정 단어에 잘못된 인식이 발생할 경우 인식 상황에 대해 조사한다. 이러한 절차를 통해 제안된 기계학습 모델의 유용성과 실용성을 보다 체계적으로 검증하고자 한다.

1.1 수어 단어 인식 모델 성능 비교

본 연구에서는 제안된 수어 단어 인식을 위한 기계학습 모델의 성능을 비교하기 위해 새롭게 수집된 24개의 수어 단어 동영상 데이터를 이용하였다. 각 동영상 데이터는 Python 환경에서 데이터 전처리를 거친 후, 모델에 입력되고, 모델이 생성한 예측 값과 실제 수어 단어를 비교하여 모델의 인식 정확도를 측정하였다. 그림 39는 24개의 수어 단어 동영상을 촬영 장면을 나타낸다.

그림 39 수어 단어 동영상 촬영 장면

촬영된 동영상의 수어 단어 의미는 '빌리다', '돕다', '주다', '지불하다', '대출', '맞다', '가다', '신용카드', '예금', '심사'이다. 표 8은 촬영된 동영상에서 획득한 학습 데이터를 이용하여 모델이 생성한 예측 값과 실제 수어 단어의 비교를 나타낸다.

표 7 수어 단어 인식 모델 성능 비교

단어	Raw-CNN	Raw-LSTM	Feature-CNN	Feature-LSTM
빌리다 [0]	Label : 0	Label : 0	Label : 0	Label : 0
돕다 [1]	Label : 1	Label : 1	Label : 1	Label : 1
주다 [2]	Label : 0	Label : 0	Label : 2	Label : 2

지불하다 [3]	Label : 3	Label : 3	Label : 3	Label : 3
대출 [4]	Label : 4	Label : 4	Label : 4	Label : 4
맞다 [5]	Label : 9	Label : 9	Label : 5	Label : 5
가다 [6]	Label : 6	Label : 6	Label : 2	Label : 6
신용카드 [7]	Label : 7	Label : 7	Label : 7	Label : 7
예금 [8]	Label : 8	Label : 8	Label : 19	Label : 22
심사 [9]	Label : 9	Label : 9	Label : 9	Label : 9
없다 [10]	Label : 10	Label : 10	Label : 10	Label : 10
어렵다 [11]	Label : 11	Label : 17	Label : 11	Label : 11
검사 [12]	Label : 9	Label : 12	Label : 12	Label : 9
감기 [13]	Label : 13	Label : 13	Label : 13	Label : 13
보건소 [14]	Label : 14	Label : 9	Label : 14	Label : 14
소화제 [15]	Label : 15	Label : 15	Label : 15	Label : 15
수면제 [16]	Label : 16	Label : 16	Label : 16	Label : 16
회복 [17]	Label : 11	Label : 17	Label : 17	Label : 17
입원 [18]	Label : 18	Label : 18	Label : 18	Label : 18
진단서 [19]	Label : 19	Label : 19	Label : 19	Label : 19
치료 [20]	Label : 17	Label : 20	Label : 20	Label : 20
퇴원 [21]	Label : 21	Label : 21	Label : 21	Label : 21
의사 [22]	Label : 22	Label : 22	Label : 22	Label : 22
병문안 [23]	Label : 23	Label : 23	Label : 23	Label : 23
일치	19	21	22	22
불일치	5	4	2	2
정확도	79.16%	87.5%	91.66%	91.66%

모델의 성능을 비교해 본 결과 Raw-data를 사용한 CNN 모델은 단어 24개 중 19개의 수어 단어를 정확하게 인식하였으며, LSTM 모델은 21개를 정확하게 인식하였다. Feature-data를 사용한 CNN 모델과 LSTM 모델은 단어 24개 중 22개의 수어 단어를 정확하게 인식하였다.

이를 통해 Feature-data를 사용한 수어 단어 인식 모델이 Raw-data를 사용한 인식

모델보다 더 높은 인식률을 가지는 것을 확인할 수 있었다.

1.2 유용성 검증

본 연구에서 제안된 방안에 대한 유용성을 검증하기 위해서 각 모델의 정확도를 계산하였다. Raw-data를 사용한 CNN 모델의 인식 정확도는 약 79.16%, LSTM 모델은 약 87.5%로 측정되었다. 반면에 Feature-data를 사용한 CNN 모델과 LSTM 모델은 각각 약 91.67%의 높은 정확도를 보였다.

그러나, 모든 모델에서 일부 수어 단어의 인식에 실패한 경우가 있었는데, 이는 대부분 비슷한 유형의 동작으로 인해 발생한 것으로 분석된다. 이러한 문제를 해결하기 위해서 더 많은 학습 데이터와 모델의 성능을 개선할 수 있는 파라미터 튜닝이 필요함을 인지하였다.

따라서, 본 연구에서 제안한 기계학습 모델은 이미 높은 인식률을 보여주었지만, 더 많은 학습 데이터와 파마리터 튜닝을 통해 더욱 개선된 성능을 보일 수 있을 것이라 예상된다. 이를 통해, 추후 심도있는 유용성 검증을 진행하고자 한다.

제 5 장 결론 및 토의

본 연구에서는 기계학습과 RGB 영상처리를 이용한 수어 동작 인식 방안을 제안하고, 제안된 수어 동작 인식 모델의 성능을 평가하였다. 이를 위해 농인들의 일상생활에서 수화가 필수적으로 요구되는 생활 공간을 찾고, 생활 공간에서 사용되는 24개의 단어를 선정하였다. 선정된 수어 동영상을 수집하기 위해 국립국어원 한국수어사전 및 공공 데이터 서비스를 활용하고, 수어 동작을 직접 촬영하여 180개의 수어 동작을 획득하였다. 이후 OpenCV와 MediaPipe를 사용하여 골격 데이터를 추적 및 시각화하고, 동영상에서 사람 신체의 특징점을 추출하여 학습 데이터를 구축하는데 활용하였다.

학습 데이터는 추출된 특징점의 위치 데이터(255개)를 나타내는 Raw-data와 위치 데이터를 기반으로 제작한 벡터 정보(54개)를 나타내는 Feature-data로 구성하고, 데이터 전처리를 통해 모델에 성능에 영향을 끼칠 수 있는 프레임을 제거하였다. 이후, Python Tensorfow keras 패키지를 사용하여 합성곱 신경망 모델과 순환 신경망 모델을 개발하였고, Raw-data와 Feature-data를 각각 학습하였다. 학습결과 Raw-data를 사용한 CNN 모델의 경우 92.78% 정확도를 보였고, LSTM 모델의 경우 97.51%의 정확도를 보였다. Feature-data를 사용한 CNN 모델의 경우 99.17% 정확도를 보였고, LSTM 모델의 경우 100% 정확도를 나타냈다.

개발된 모델을 유용성을 검증하기 위해 24개의 새로운 동영상을 촬영하고, 모델이 생성한 예측 값과 실제 수어 단어를 비교하였다. 비교 결과 Raw-data를 사용한 CNN 모델은 79.16%, LSTM 모델은 87.5%의 정확도를 보였다. 반면 Feature-data를 사용한 CNN 모델과 LSTM 모델은 각각 91.66% 정확도를 보였다. 이를 통해, 특징점의 위치 데이터인 Raw-data를 학습 데이터로 사용하는 것 보다 위치 데이터를 기반으로 제작한 벡터 정보인 Feature-data를 학습 데이터 사용하는 방안이 수어 동작을

인식하는데 보다 효과적임을 입증하였다.

그러나, 모델의 일부 수어 단어의 인식에 실패한 경우가 있었다. 이는 비슷한 유형의 동작으로 인해 발생한 것으로 분석하였고, 이를 극복하기 위해 추가적인 학습 데이터와 모델의 성능을 개선할 수 있는 파라미터 튜닝이 필요하다는 결론을 도출하였다. 본 연구를 통해 제시된 기계 학습 모델은 수어 인식의 정확성을 향상시키는 데 기여하였다. 더욱이, 이 연구를 통해 얻은 결과는 수어 사용자와 비수어 사용자 간의 의사소통을 돕는 데 의미있는 발전을 가져올 것으로 보인다.

향후, 더 많은 학습 데이터와 파라미터 튜닝을 통해 모델의 성능을 개선하고, 더 다양한 수어 단어를 인식할 수 있도록 모델을 확장하고, 수어 아바타 애니메이션 생성 및 수어 통역 시스템에 모델을 활용함으로써 수어 사용자들의 일상생활이 더욱 향상될 것으로 기대된다.

참고문헌

1. Lee, S. G., Kim, Y. J. & Park, J. H. 2017, A Study on Multilateral Conversation Support System for the Deaf using Smart Glasses and Speech Recognition Technology, pp.1-57.

2. Lee, J. W., Jeong, J. W., Oh, M. A., Jeong, H. C., Jo, J. H., Oh, J. Y. & Kim, H. S. 2020, A Study on the Usage of Korean Sign Language in [2020], Ministry of Culture, Sports and Tourism & National Institute of Korean Language, 2020.

3. Data Gloves, https://www.motioncapture.co.kr/

4. IMU Sensor, https://www.sbg-systems.com/

5. MYO, https://myomirror.com/

6. Kinect, https://dev.windows.com/en-us/kinect/

7. Leap Motion, https://www.leapmotion.com/

8. Kim, K., Kim, J., Park, H., 2021 Hand Gesture-based Interface for Navigating a Virtual Space using Leap Motion and Machine Learning, Journal of Computational Design and Engineering, Vol.26(3), pp.239-248.

9. MediaPipe, https://developers.google.com/

10. TensorFlow, https://www.tensorflow.org/

11. Na, K., Lee, B., Kim, J., Kim, J. & Jung, Y., 2009 Development of Virtual Reality Contents for Korean Sign Language Interpretation Proceedings of HCI Korea, pp.690-695.

12. Park, S.J. and Park, H., 2017, Virtual Navigation of Blood Vessels using 3D Curve-Skeletons, Korean Journal of Computational Design and Engineering, Vol.22(1), pp.89-99.

13. Moon, H. J., Kim. S. J., Jung, J. H., 2017, A Study on Sign Language Recognition System Using LeapMotion, Journal of KIPS, Vol.24(2), pp.1041-1044.

14. Naglot, D., & Kulkarni, M. (2016, August). Real time sign language recognition using the leap motion controller. In 2016 international conference on inventive computation technologies (ICICT), Vol.3, pp.1-5.

15. Cheon, W., Yang, H., 2015, Sign Language Recognition Using Kinect, Proceedings of Korean Institute of Information Scientists and Engineers, pp.984-985.

16. Dong, C., Leu, M. C., & Yin, Z., 2015, American sign language alphabet recognition using microsoft kinect. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp.44-52.

17.OpenPose, https://github.com/CMU-Perceptual-Computing-Lab/openpose/

18. AlphaPose, https://github.com/MVIG-SJTU/AlphaPose/

19. Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S., 2017, Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, Vol.105(12), pp.2295-2329.

20. Yann LeCun, L`eon Bottou, Yoshua Bengio, and Patrick Haffner., 1998, Gradient-based learning applied to document recognition. in Proceedings of the IEEE, Vol.86(11), pp.2278-2324.

21. Medsker, L. R., & Jain, L. C., 2001, Recurrent neural networks. Design and Applications, Vol.5(2), pp.64-67.

22. Yu, Y., Si, X., Hu, C., & Zhang, J., 2019, A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, Vol.31(7), pp.1235-1270.

23. Jeong, W. S. & Rhee,, S. Y., 2022, Motion Recognition of Workers using Skeleton and LSTM, Journal of Korea Multimedia Society, Vol.25(4), pp.575-582.

24. Korean Sign Language Dictionary, https://sldict.korean.go.kr/

25. Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M., 2020, Blazepose: On-device real-time body pose tracking, pp.2006-10204.

26. Brar, D. S., Kumar, A., Mittal, U., & Rana, P., 2021, Face detection for real world application. In 2021 2nd International Conference on Intelligent Engineering and Management, pp.239-242.

27. Gan, M., Lin, Y., Liu, X., Song, W., Zeng, J., & Kang, W., 2023, Ar3dHands: A Dataset and Baseline for Real-Time 3D Hand Pose Estimation from Binocular Distorted Images. In International Conference on Image and Graphics, pp.167-179.