[서식1]산학협력프로젝트 결과보고서(졸업논문)

머신러닝 기반 알파벳 수어 번역 모델

201813168 / 박주현

Alphabet Sign Language Translation Model Based On Machine Learning

Ju Hyun Park

요 약

이 보고서는 청각장애인과 비장애인 간의 원활한 의사소통을 위한 머신러닝 기반 알파벳 수어 번역 모델을 제안한다. 제안한 모델은 실시간 영상에서 데이터를 추출하는 데이터 추출 단계와 추출된 데이터를 전처리하는 전처리 단계, 데이터 기반 예측을 수행하는 모델 예측 단계, 모델 예측 결과를 시각화해주는 예측값 시각화 단계로 구분한다. 데이터 추출 단계에서는 구글에서 제공하는 AI Framework인 Mediapipe를 이용하여 손가락 관절 좌표를 데이터로 추출한다. 데이터 전처리 단계에서는 추출한 데이터를 전처리를 수행하여 모델이 예측하는데 필요한 데이터로 변형을 준다. 모델 예측 단계에서는 머신러닝 지도학습 알고리즘인 K-NN 알고리즘을 이용하여 전처리 단계에서 얻은 전처리 데이터를 29개의 Label 중에서 1개로 분류하는 연산을 수행한다. 최종적으로 데이터 출력 단계에서는 제안한 모델이 반환하는 Label 값을 출력값으로 하여 사용자에게 시각화해주는 작업을 수행한다. 제안한 모델을 훈련 시키기 위해 직접 훈련 데이터 셋을 제작하였고, 동시에 테스트 데이터 셋도 제작하여 모델의 정확도를 평가하였다. 훈련 데이터 셋에서 가장 높은 정확도를 보인 k 값인 k=5 값을 사용하여 테스트 데이터 셋 실험 결과 Accuracy가 87%를 달성하여 제안한 모델이 우수한 성능을 보임을 확인하였다.

<키워드 : 수어, 머신러닝, K-NN 알고리즘, Mediapipe >

I. 서 론

최근 인공지능 기술의 급격한 발전은 혁명적인 변화를 가져오고 있다. 이러한 변화의 흐름에서도 장애인들을 위한 기술적 지원이 충분히 보급되지 못하고 있는 것이 현실이다. 한국장애인고용공단이 발표한 2021 장애 인구 규모에 따르면, 지체 장애 119.1만 명에 이어 청각장애인의 규모는 두 번째로 많은 41.2만 명인 것으로 나타났다. 그러나 국가 공인 수어 통역사 자격증 취득자는 2,000명이 채 되지 않는 것으로 밝혀져 일상생활 속 청각장애인과 비장애인 간의 의사소통 문제는 불가피한 문제인 것으로 시사된다. 게다가 수어를 구사하기 위해서는 평균 3~5년의 시간과 노력이 필요하고, 이러한 수어의 난이도는 청각장애인과 비장애인 간의 소통에 있어 더 큰 어려움을 겪게 한다.

본 프로젝트에서는 이런 문제를 극복하기 위해 머신러닝 알고리즘을 기반으로 하는 알파벳 수어 번역 모델을 제안한다. 효과적인 수어 번역 모델은 원활한 의사소통을 위해 실시간 상호작용과 높은 정확도가 요구된다. 실시간 상호작용을 만족하기 위해 Google에서 제공하는 AI Framework인 Mediapipe를 이용하여 실시간 영상 데이터에서 손가락 관절을 빠르게 인식하여 손가락 관절 좌표를 받아들이는 방식으로 연구를 진행하였다. 그리고 높은 정확도를 만족하기 위해 손가락 관절 좌표를 모델의 입력 데이터로 사용하는 것은 데이터 분류에 있어 강인함이 떨어진다고 판단하여 손가락 마디 간의 각도를 계산하여 모델의 입력 데이터로 사용하는 것으로 연구를 진행하였다.

본 논문의 구성으로 2장에서는 인공지능 기반 수어 번역 모델 관련 연구를 소개하고, 3장에서는 본 프로젝트에서 제안하는 모델에 대해 설명한다. 4장에서는 제안한 모델을 활용한 실험 결과를 보이며, 5장에서는 모델에 대한 결론을 맺는다

II. 관련 연구

2.1 YOLO

YOLO (You Only Look Once) 모델은 객체 감지 및 객체 분류를 위한 딥러닝 모델로 입력된 이미지나 영상을 19*19개의 그리드로 분할하고 각각의 그리드에 해당하는 물체의 위치와 범위 그리고 종류를 출력하는 네트워크다. YOLO는 입력된 이미지를 한 번만 처리하므로 처리 속도가 빠르다는 장점이 있다. 이는 2-stage-detection 방식을 사용하는 네트워크에 비해 정확성에서 일부 차이가 있을 수 있지만 빠른 객체 감지가 필요한 경우에는 YOLO가 적합하다. 또한 YOLO는 후보 영역을 추출하기 위한 별도의 네트워크를 적용하지 않기에 Region-based Convolutional Neural Network (R-CNN)과 같은 다른 모델보다 처리시간 측면에서 우수한 성능을 보인다. CNN은 이미지 인식 및 분류에 사용되는 일반적인 신경망 모델이다. 반면, YOLO 모델은 CNN을 사용하여 이미지나 비디오와 같은 2D 데이터에서 공간적인 특징을 추출하고, 이러한 특징을 활용하여 객체 탐지, 분류, 위치 예측 등 다양한 작업을 수행하는 단일 인공신경망을 활용하여 학습하기에 다른 기법에 비해 인식 속도가 매우 빠르다. 이 과정에서 이미지나 비디오는 2D 데이터로 취급되며, 컬러 이미지의 경우 RGB 채널 정보가 함께 활용된다. 이러한 이유로 인식 속도가 빠른 YOLO 모델을 수어 번역 기반 모델 연구에 많이 사용한다.

2.2 UNet-LSTM 결합 모델

LSTM(Long Short-Term Memory) 모델은 오디오, 비디오 또는 텍스트와 같은 시계열 데이터를 학습 및 처리하기 위해 만들어진 Recurrent Neural Network 모델의 장기 의존성을 개선한 모델이다. LSTM 모델을 활용하여 시계열 데이터인 비디오로부터 동적 데이터인 수어 동작을 인식하는 방식으로 수어를 번역하는 연구와 UNet-LSTM 기반의 음성에서 수어 번역 모델 연구가 이루어졌다.

2.3 Conv1D-LSTM 결합 모델

Conv1D(One Dimensional Convolution)는 CNN 모델의 한 종류로 1차원 데이터를 입력 데이터로 사용하며 해당 데이터의 패턴 및 특성을 찾는 데 사용된다. 주로 시계열 데이터를 학습하는 것에 용이하여 오디오나 텍스트 데이터의 패턴 및 특성을 파악하는 데 사용된다.

Conv1D-LSTM 결합 모델은 Conv1D 모델로 입력 데이터를 가공하여 LSTM 모델로 학습시키는 모델이다. 이 결합 모델은 CNN 모델과 LSTM 모델보다 더 적은 학습 파라미터로 더 낮은 RMSE 및 MRF 오류율을 기록한다. 이처럼 결합 모델은 다른 두 모델보다 더 효율적으로 학습이 가능한 수어 번역 모델로써 연구가 이루어졌다.

2.4 Mediapipe

Mediapipe는 Google에서 제공하는 AI Framework로, 실시간 객체 감지와 추적을 지원한다. Mediapipe는 실시간 영상에서 얼굴, 손, 전신과 같은 다양한 객체를 감지할 수 있다. Mediapipe가 지원하는 다양한 객체 감지 중에서 손 객체 감지의 경우 손 객체를 감지하고, 20개의 손가락 관절을 세부적으로 감지하여 20개의 손가락 관절에 대한 좌표 (x, y)와 카메라와 손 사이의 거리 z 값을 계산하여 반환합니다. 수어 번역의 경우 실시간 영상에서의 현재 행동 인식이 매우 중요하고, 손가락의 다양한 행동을 정확하고 빠르게 인식해야 하기에 실시간 수어 번역 모델들은 주로 Mediapipe를 기반으로 개발되고 있다. 본 프로젝트에서도 알파벳 수어 동작을 수행하는 손동작을 빠르게 인식하여 결과를 도출해야 하기에 Mediapipe를 이용하여 손 객체에 대한 정보를 빠르게 가져와 실시간 상호작용의 장점을 가져왔다.

III. 주요 내용

3.1 모델 설계 구조

본 프로젝트에서 구현하고자 하는 실시간 알파벳 수어 번역 모델의 프로세스는 그림 1과 같다. 먼저 Mediapipe를 이용하여 실시간 영상에서 현재 수행되고 있는 손동작 데이터를 추출한다. 이후 추출한 데이터를 모델 예측 결과에 도움이 되는 강인한 데이터로 바꾸기 위해 데이터 전처리를 거친다. 이후 머신러닝 모델을 이용하여 현재 수행되고 있는 손동작 데이터가 어떤 Label 값을 갖는지 예측을 수행한다. 이후 예측값을 받아 이를 시각화한다.

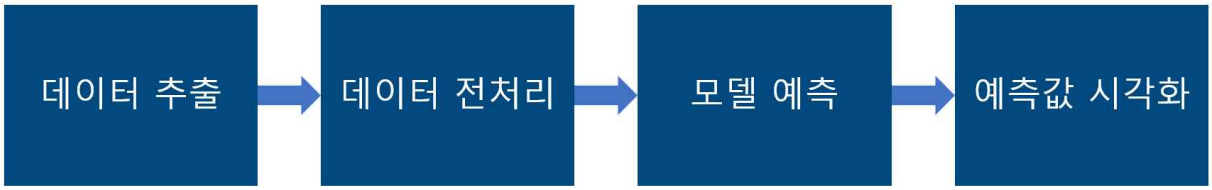
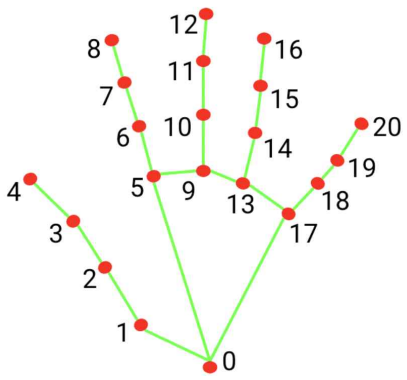


그림 1. 실시간 알파벳 수어 번역 모델 프로세스

3.2 데이터 추출

데이터 추출 단계에서는 실시간 영상에서 현재 수행되고 있는 알파벳 수어 동작의 데이터 추출을 수행한다. 실시간 영상을 활성화하기 위해 OpenCV를 이용하여 실시간 영상을 활성화하고, 이후 Mediapipe를 이용하여 수어 동작에 대한 데이터를 추출한다. Mediapipe는 Google에서 제공하는 AI Framework로 얼굴, 손, 전신과 같은 다양한 객체를 실시간 감지와 추적을 지원한다. 손 객체 감지의 경우 활성화되어 있는 영상에서 프레임 단위로 그림 2와 같이 20개의 손가락 관절을 감지하고, 20개의 관절에 대한 관절 좌표 (x, y)와 카메라와 손 사이의 거리 z 값을 추출할 수 있다. 본 프로젝트에서는 현재 수행되고 있는 알파벳 수어 동작에 대한 20개의 손가락 관절 좌표를 배열에 저장하는 방식으로 데이터 추출을 진행한다.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

그림 2. Mediapipe Hand Landmark 위치

3.3 데이터 전처리

Mediapipe를 이용하여 추출한 (x, y, z) 값은 사용자마다 손가락의 길이가 다르다는 것, 실시간 영상에서 손의 위치에 따라 관절 좌표의 값이 균일하지 못하다는 문제점이 있다. 이는 같은 Label을 갖는 데이터들이 일관되지 못한 값을 갖게 되고, 결국 모델의 예측 수행에 있어 좋지 않은 데이터라는 것을 의미한다. 따라서 어느 위치에서 수행되는 수어 동작이더라도 추출된 데이터가 일관성을 갖는 데이터로 변형해 줄 필요성이 있다. 일관성 있는 데이터를 위해 수어 동작마다 달라지는 손가락 마디의 각도가 달라지는 것에 집중하였다. 한 개의 손가락에는 4개의 마디가 있고, 한 개의 손에는 총 20개의 마디가 있다. 한 개의 손가락에는 3개의 마디 각도가 있고, 한 개의 손에는 총 15개의 마디 각도가 있다. 15개의 마디 각도는 사용자마다 관절 좌표 (x, y, z)의 값이 다르더라도 결국 벡터 연산으로 수행되기에 일관성을 갖게 된다. 마디 각도를 구하는 방법은 다음과 같다. 우선

데이터 추출 단계에서 얻은 20개의 손가락 관절 좌표를 이용하여 20개의 마디 벡터를 계산한다. 다음으로 벡터 사이의 각도 구하는 식 1을 이용하여 15개의 손가락 마디 각도 값을 계산한다.

$$\theta = cos^{-1}(\frac{\vec{\alpha}\cdot\vec{\beta}}{|\vec{\alpha}||\vec{\beta}|}) = acos(\frac{\vec{\alpha}\cdot\vec{\beta}}{|\vec{\alpha}||\vec{\beta}|})$$

식 1. 두 벡터 사이의 각도

3.4 모델 예측

모델 예측 단계에서는 머신러닝 모델을 이용하여 전처리 데이터를 알파벳 26개와 추가 기능 3개를 더한 29개의 Label 중 1개의 값으로 예측하는 결과를 얻는 단계이다. 머신러닝 모델은 지도학습 알고리즘인 K-Nearest Neighbor를 선택하였다. K-Nearest Neighbor는 새로운 데이터가 입력되면 기존의 데이터와의 거리를 계산하여 가장 가까운 k 개 데이터 중에서 수가 가장 많은 Label을 따라가는 지도학습 알고리즘이다. 프로젝트에서는 전처리 데이터가 새로운 입력 데이터이고, 기존의 데이터가 훈련 데이터 셋이다. 훈련 데이터 셋은 알파벳 수어의 마디 각도 데이터가 존재하지 않아 0~28의 값을 갖는 라벨링 작업을 직접 수행하여 15개의 마디 각도와 1개의 Label 값을 갖는 그림 3과 같은 훈련 데이터 셋과 그림 4와 같은 테스트 데이터 셋을 만들었다. K-Nearest Neighbor 알고리즘 모델은 프레임마다 추출되는 관절 좌표를 전처리하여 훈련 데이터 셋과의 계산을 통해 1개의 Label을 반환한다. 만약 1초 동안 같은 Label 값이 반환된다면 사용자가 해당 Label 수어 동작을 수행하는 것으로 판단하여 해당 Label 값을 최종 출력값으로 결정한다.

train_data

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	label
0	32.475960	25.587929	8.252770	25.815522	144.366824	20.521152	153.119167	171.395545	157.538166	143.199414	171.882595	171.408446	146.234357	166.588569	167.465896	0.0
1	31.606072	25.710666	6.519657	26.173296	143.745670	19.952463	151.712657	172.221686	156.749018	143.046840	172.012460	171.795896	146.550308	166.109924	167.001468	0.0
2	30.390004	25.734120	7.356103	27.159364	143.616866	19.383633	151.881650	172.402187	158.163917	142.416861	171.645398	172.286008	146.364281	165.717341	167.060126	0.0
3	31.614848	24.719924	8.405087	24.753390	145.194042	22.404418	154.930414	171.543602	158.543800	143.679277	172.226178	171.611250	145.433540	167.020346	168.733117	0.0
4	33.461137	25.121808	9.622665	24.826552	144.852757	22.531676	154.263942	172.320347	158.218319	143.096537	172.649381	171.384379	143.902039	167.246355	169.031408	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6797	13.637745	4.437827	18.915880	33.798126	132.594291	66.320045	135.192056	177.609946	127.396172	134.222130	168.078797	137.142940	133.899248	160.153600	150.526114	28.0
6798	12.840574	5.365858	19.456192	32.923410	133.482656	66.832332	135.573651	176.796588	127.413598	134.643829	168.857518	136.646297	131.804588	160.987585	150.974406	28.0
6799	13.415913	7.073638	18.543359	31.722039	134.383037	61.393321	137.035118	176.208448	130.875017	135.543597	168.301235	143.177745	132.778474	160.659083	152.975265	28.0
6800	13.062817	4.784622	20.063379	31.302324	134.945269	58.010853	137.097486	175.321165	131.191599	136.519820	167.937522	143.362089	133.701673	159.253567	152.183998	28.0
6801	12.412394	5.603470	17.770540	32.265286	133.054300	62.118395	139.206266	175.041751	131.779023	133.387100	169.126600	143.664203	129.204596	160.304895	154.075868	28.0

6802 rows × 16 columns

그림 3. 훈련 데이터 셋

test_data

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	label
0	33.348293	28.029244	6.434522	25.306710	147.621546	13.814457	155.484599	172.916627	162.887374	145.621626	171.760095	177.728004	152.682516	165.900899	164.643348	0.0
1	32.466970	29.608414	6.124208	25.575725	146.713341	14.558109	155.101129	172.648273	162.861676	144.831115	171.845820	177.673124	151.582057	166.197361	164.228950	0.0
2	32.925779	28.871193	6.133435	25.068904	147.867694	13.901259	155.604915	172.676099	163.159843	145.794149	171.700362	177.951648	152.718481	166.092380	164.272332	0.0
3	33.180815	28.392374	6.115724	25.575945	147.408798	13.413727	155.217442	173.227818	163.445607	145.649354	171.317936	177.942014	152.251529	165.467002	163.795323	0.0
4	32.536586	24.723108	4.172380	27.922642	143.584000	15.132717	152.086715	171.751168	159.820028	143.220558	172.040179	177.389642	148.914735	166.007331	165.654766	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1155	5.031190	23.795024	93.700175	112.125199	14.809350	128.689469	144.930526	146.179336	169.452224	17.174732	168.247162	169.966983	52.189246	157.837065	171.658622	28.0
1156	5.187651	24.100158	93.342128	113.229010	12.059875	127.895749	141.782622	146.188059	168.178965	15.367217	169.814082	170.244561	49.427543	160.408769	170.145189	28.0
1157	5.673865	22.533890	90.482508	112.674500	19.538301	125.742640	147.677755	146.004259	168.604833	15.973694	165.854424	171.382027	55.036337	158.764764	172.239115	28.0
1158	5.780321	24.947546	90.483323	113.588601	18.141783	127.310569	147.588358	148.828220	169.850730	17.919150	165.532409	171.371172	55.707218	159.798922	173.349716	28.0
1159	5.902707	24.423735	92.408199	111.589121	14.265204	128.656638	146.066064	145.662109	169.907205	17.368415	165.003917	171.406078	57.197854	159.416130	173.399774	28.0

1160 rows × 16 columns

그림 4. 테스트 데이터 셋

3.5 예측값 시각화

예측값 시각화 단계는 모델 예측 단계의 최종 출력값을 받아 사용자의 실시간 영상 화면에 시각화 해주는 기능을 수행한다. 최종 출력값을 문자열에 저장하고, openCV를 이용하여 실시간 영상 화면 좌측 하단에 문자열을 출력한다. 이와 동시에 K-Nearest Neighbor 모델이 프레임 단위로 수행하는 예측 결과를 그림 1의 0번 Landmark 아래에 출력하여 사용자의 수어 동작을 모델이 어떻게 예측하는지 보여줌으로써 사용자의 편의성을 높였다. 또한 사용자가 수어를 보고 수행할 수 있게끔 수어와 추가 기능이 들어간 이미지를 만들어 UI에 추가하였다. 최종 출력 화면은 그림 5와 같다.

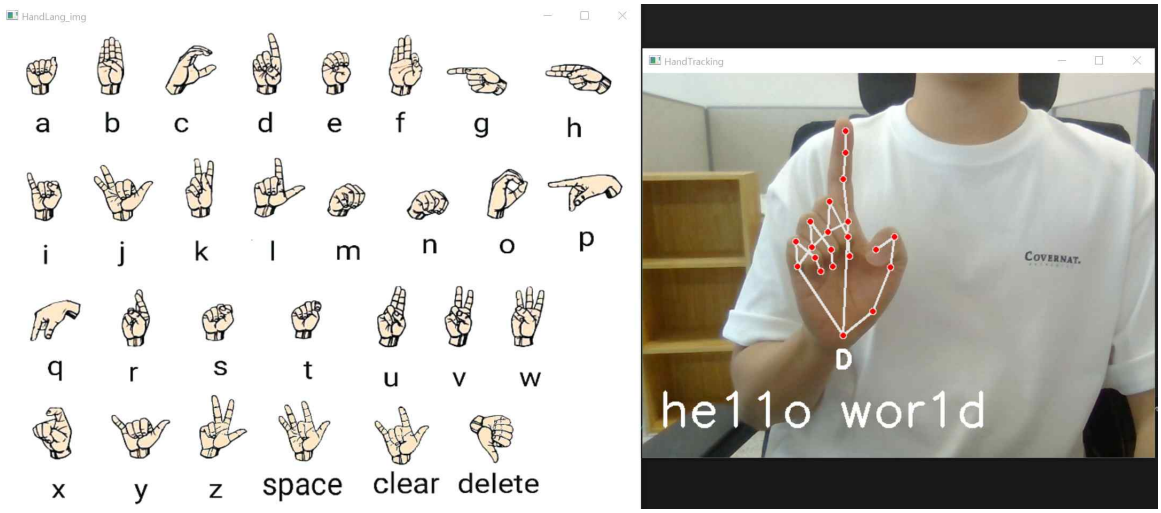


그림 5. ‘hello world’를 수어로 수행 후 시각화 결과

IV. 결과 평가

4.1 모델 예측 정확도 평가

K-Nearest Neighbor 모델은 K 값에 따라 예측하는 값이 달라진다. 따라서 그림 4와 같은 테스트 데이터 셋을 이용하여 최적의 K 값을 찾는 실험을 진행하였다. 실험은 그림 6의 (좌) 그래프와 같이 K 값의 범위를 1~100으로 설정하여 1차 확인 후, 그림 6의 (우) 그래프와 같이 Accuracy가 높은 1~10으로 재설정하여 진행했다. 최종적으로 K는 5일 때 Accuracy 87.8%로 가장 높은 Accuracy를 갖는다는 것을 확인할 수 있다.

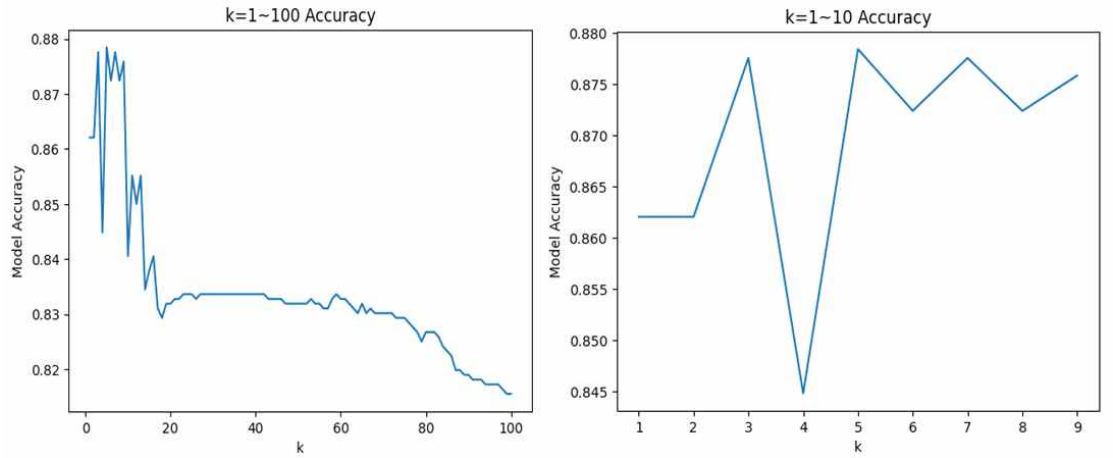


그림 6. (좌) k= 1~100일 때 모델 Accuracy (우) k= 1~10일 때 모델 Accuracy

V. 결 론

본 프로젝트에서는 실시간 알파벳 수어 인식을 위해 K-Nearest Neighbor 머신러닝 모델을 활용하였다. 실시간 영상에서의 데이터 추출을 위해 Mediapipe를 이용하였고, 마디 벡터 간의 각도를 계산하여 마디 각도를 구하고, 직접 훈련 데이터 셋과 테스트 데이터 셋을 만들고, 이를 이용하여 알파벳 수어를 분류하였다. 결과적으로 87.8%의 Accuracy를 달성하였다. 청각장애인분들이 일상생활 속에서 느끼는 의사소통의 어려움을 실시간 수어 번역 모델을 활용하여 청각장애인과 비장애인 간의 소통이 활성화되기를 기대한다. 더불어 머신러닝 기반인 본 프로젝트와 딥러닝 기반 수어 번역 모델 간의 Accuracy의 차이가 얼마나 날지 추가 실험을 진행할 예정이다.

VI. 참 고 문 헌

[1] 이동욱, 김민서, 김남호, 최광미. 실시간 수어 AI 번역 프로그램 구현. 한국디지털콘텐츠학회 논문지, pp. 2585-2591, 2023

[2] 김윤기, 지영채, 하정우. “청각장애인을 위한 UNet-LSTM 기반의 자동 음성-수어 번역 모델.” 한국컴퓨터종합학술대회 논문집, pp. 709-711, 2019.

[3] 김기찬, 하란. “딥러닝을 통한 실시간 수어 번역 프로그램 개발.” 한국소프트웨어종합학술대회 논문집, pp. 1774-1776, 2022.

* 이 보고서는 IT대학 컴퓨터공학과 2023년 2학기 졸업보고서로 심사 통과되었음[지도교수: (인)]