

프로보노 ICT멘토링 프로젝트 결과보고서

[Carrytranslator] 청각장애인을 위한 휴대용 수화 통역기

프로젝트 참여자

멘 토		유형석			더존비즈온
멘	티	황정환	인천대학교	정보통신공학과	
멘	티	안이삭	인천대학교	정보통신공학과	
멘	티	육성현	서울시립대학교	수학과	
멘	티	강지연	서울시립대학교	컴퓨터과학과	

요 약

프로젝트명	청각장애인을 위한 휴대용 수화 통역기			
주제영역	【대상】			
	<input checked="" type="checkbox"/> 장애인 <input type="checkbox"/> 1인가구 고령자 <input type="checkbox"/> 환자 <input type="checkbox"/> 기초생활수급자 <input type="checkbox"/> 기타()			
	【상황】			
택 1		택 1		
<input type="checkbox"/> 예방 <input type="checkbox"/> 사후 <input type="checkbox"/> 통합		<input type="checkbox"/> 응급() <input checked="" type="checkbox"/> 일반(일상생활)		
기술분야	<input type="checkbox"/> SW <input type="checkbox"/> 네트워크 <input type="checkbox"/> 정보보호	<input type="checkbox"/> 이동통신 <input type="checkbox"/> 블록체인	<input type="checkbox"/> 융합서비스 <input checked="" type="checkbox"/> ICT디바이스	<input type="checkbox"/> 전파위성 <input type="checkbox"/> 기반SW컴퓨팅 <input type="checkbox"/> 방송스마트미디어 <input type="checkbox"/> 디지털콘텐츠
달성성과	<input checked="" type="checkbox"/> 특허 <input type="checkbox"/> 기술이전	<input type="checkbox"/> 앱 등록 <input type="checkbox"/> 프로그램 등록	<input type="checkbox"/> 실용화 <input checked="" type="checkbox"/> 논문게재 및 발표	<input checked="" type="checkbox"/> 공모전(프로보노 공모전) <input type="checkbox"/> 기타()
프로젝트 소개	<p>농아인과 비장애인이 손쉽게 커뮤니케이션을 하기 위한 프로그램이다. 농아인은 수화를, 비장애인은 음성을 이용해 텍스트로만 대화하는 기존 상황에서 실시간으로 번역할 수 있다. 키넥트와 미니PC를 연결해 휴대성이 용이하여 수화통역사의 도움 없이 일상생활에서부터 공공기관, 은행 등 여러 방면에서 사용이 가능하다.</p>			
개발배경 및 필요성	<p>청각장애인을 위한 IT서비스 부족한 상황이며 애플리케이션을 찾아보면, 수화를 배우는 목적인 서비스가 다수이다. 청각장애인의 소통에 초점을 맞춘 IT서비스는 찾아보기 힘들다. 각 광역시마다 수화통역사 수가 부족하여 매 상황마다 수화통역사가 동반하기 어렵다. 또한, 텍스트로는 수화의 풍부한 감정 전달에 있어 부재가 발생하고, 한 손으로만 영상통화를 하여 불편함이 존재한다. 농아인들이 사회적 편견이나 인식 때문에 스스로가 제한한 생활에서 벗어나는 계기가 될 수 있다.</p>			
프로젝트 주요기능	<p>데이터 저장, 수화 번역, 음성 인식 기능이 있다. 데이터 저장에서는 수어의 신조어 직접 추가할 수 있고, 지화와 동작 모두 저장이 가능하다. 수화 번역에서는 수화(지화, 동작)를 인식하여 텍스트로 전환한다. 이때 학습한 신경망만 전달하면 프로그램을 사용할 수 있다. 음성 인식에서는 텍스트를 음성으로 송출하고, 상대방의 음성을 입력받아 디스플레이에 텍스트로 전환한다.</p>			
작품의 기대효과 및 활용분야	<p>높은 휴대성을 겸비한 수화번역 기기를 개발할 수 있고, 농아인과 비장애인 사이 의사소통의 불편함을 감소시킬 수 있다. 거기다 농아인들에 대한 일반인들의 편견을 해소할 수 있다. 금융, 동사무소, 공공 사업장 등 농아인의 의사소통이 필요한 모든 곳에서 활용이 가능하다. 후천적 농아인들을 위한 수화 교육용 프로그램으로 활용할 수 있다.</p>			

본 문

I. 프로젝트 개요

1. 프로젝트 소개

○ 기획의도

- 농아인과 비장애인이 손쉽게 커뮤니케이션하기 위한 프로그램
- 농아인은 수화를, 비장애인은 음성을 이용해 실시간으로 대화할 수 있어 수화 통역사의 도움 없이 공공기관, 은행 등 여러 방면에서 사용 가능
- 텍스트로만 번역하는 기존 서비스에서 음성이나 모션을 인식하여 실시간 번역을 할 수 있는 수화번역 프로그램 개발

○ 개요

- 농아인이 수화를 하면 키넥트를 이용해 손의 모양과 동작, 얼굴표정을 인식하여 문장으로 번역
- 번역된 문장을 미니피시에 스피커를 이용 음성으로 출력
- 비장애인이 음성으로 답하면 텍스트로 변경, 텍스트로 만들어 농아인의 디스플레이로 출력

2. 개발배경 및 필요성

○ 청각장애인을 위한 IT서비스 부족

음성을 문자화하거나 사람이 자막을 만드는 서비스, 수화통역사와 연결해주는 어플리케이션이 있다. 그러나 농인 중엔 글을 모르는 사람이 존재하고 수화통역사와 항상 연결된다는 보장이 없다. 수화와 관련해서는 문장을 치면 각 단어에 맞는 수화 영상을 보여주는 어플리케이션이 있으나 정확도가 떨어지고, 위에 서술한 소수를 위한 IT서비스라기보단 수화를 배우기 위한 서비스에 가깝다.

○ 수화통역사 수 부족

2009년 기준 용산구에서는 농아인 300명당 수화통역사 1명꼴로 미국의 150명에 1명인 비율과 비교했을 때 턱없이 부족하다. 특히 밤과 주말에 발생하는 응급상황에서 적은 수의 수화통역사로 모든 일을 해결할 수 없기에 불편함을 겪는다.

1. 전국 수화통역센터 현황

시군구지부 186개 / 수화통역센터 192개

2015.08.13.

시 도 수화 통역사 ¹⁾ (명)	시군구 ²⁾ / 지 부 / 지원본부+센터 (개)	지역지원본부 + 수화통역센터 수화통역사 (명)	수화통역센터 지역지원본부 -시·도협회 (명)	수화통역센터 시군구지부 (명)	통역사 1인 담당하는 청각장애인 숫자 (명)	등록 청각장애인 (14.12.31) (명)	인 구 (15.01.01) (천명)	면 적 (15.01.01) (km ²)
전국 ³⁾	226 186 192	772	80	692	전국 평균 327	252,779	51,250	100,212
광역시 ⁴⁾			광역시 평균 495					
서울	25/ 25/ 26	122	20 ⁵⁾	102 ⁶⁾	332	40,518	10,103	605
부산	16/ 6/ 5	28	12	16	499	13,991	3,519	769
대구	8 / 3/ 4	21	6	15	543	11,409	2,493	883
인천	10/ 2/ 1	11	0	11	1,193	13,133	2,902	1,046
대전	5 / 4/ 5	22	6	16	331	7,282	1,531	540
광주	5/ 0/ 1	16	0	16	515	8,245	1,475	501
울산	5/ 1/ 1	8	0	8	759	6,079	1,166	1060
세종	/ / 1	3	0	3	281	843	156	464
경기	31/ 31/ 32	124	4	120	377	47,197	12,357	1,0172
강원	18/ 17/ 17	65	6	59	166	10,830	1,544	16,873
충남	15/ 14/ 15	53	4	49	255	13,534	2,062	8,213
충북	11/ 9/ 10	34	4	30	290	9,867	1,578	7,407
전남	22/ 16/ 15	49	4	45	333	16,652	1,905	12,303
전북	14/ 14/ 15	61	4	57	218	13,354	1,871	8,067
경남	18/ 19/ 19	60	5	55	280	16,845	3,350	10,538
경북	23/ 23/ 24	82	5	77	225	18,679	2,700	19,029
제주 ⁷⁾	/ 2/ 1	13	0	13	332	4,321	607	1,849

1) 수화통역사 정원 : 지방자치단체 시 4명, 군 3명, (지자체 사정에 따라 인원 조정)

2) 2015년도 지방자치단체 행정구역 및 인구 현황 - 행정자치부 (2015.1.1)

3) 관리직원을 제외한 실제 활동하는 수화통역사 인원

4) 부산, 대구, 대전, 광주, 울산 - 인천광역시 제외

5) 서울 20명 = 본부8, 야간4, 자치구 파견 8명 - 은평, 마포, 종로, 구로, 관악, 서초, 노원, 강서

6) 102명 = 25구×4명 (노원 강서 5명)

7) 서울서대문농아인, 대구청각언어장애인, 대전손소리, 제주도농아인, 인천청각장애인복지관(2017년 계획)

<그림 1. 전국 수화통역센터 현황>

○ 텍스트의 감정전달 부재 및 영상통화의 불편함

농민 중에 글을 아는 사람은 문자를 이용하지만 즉시 대화와 역동적인 의사소통이 어려워 영상통화를 이용하는 사람이 많다. 그러나 영상통화를 이용하면 한 손으로만 수화를 하고 휴대폰을 들고 있어야 하기에 오래 통화를 하는데에 무리가 있다.

○ 필요성

- 수화를 못 하는 가족과의 대화에서부터 병원, 공공기관까지 수화통역사가 동행
- 긴급한 일이 아닌 일상생활까지 통역을 해주면서 수많은 농아인을 감당할 통역사 수 부족
- 통역사를 동행해도 통역하는 시간을 기다려주지 않는 상황이 많아 비장애인과 의 대화를 피하는 농아인이 많음
- 외국과 달리 한국은 수화번역기기에 대한 발달이 제대로 이루어지지 않음
- 농아인과 비장애인의 대화가 어렵다는 편견을 깨고 친숙한 감정을 갖는 효과
- 농아인들이 사회적 편견이나 인식 때문에 스스로가 제한한 생활에서 벗어나는 계기가 될 수 있음

3. 특징 및 장점

○ 미니피시를 이용한 휴대성 검비

외국의 키넥트를 이용한 수화번역기는 설치형식으로 사용되어 지지만 우리 프로젝트의 목표는 미니피시와 휴대정보조배터리 그리고 키넥트를 이용한 들고 다닐 수 있는 수화번역기이다.

○ 사용자가 데이터 저장 가능

언어에 신조어가 있는 것처럼 수화에 생기는 새로운 단어도 사용자가 직접 추가하여 사용할 수 있다. 개발자가 넣어놓은 글자나 단어만 사용할 수 있는 기존 IT 서비스와 가장 큰 차별점이다.

○ 인공지능 전달만으로 프로그램 사용

학습한 인공지능을 보유하기만 하면 수화 번역 프로그램을 사용할 수 있다.

4. 구성도

시스템 구성도	설명
<div data-bbox="132 347 895 869"> <p>The diagram illustrates the system architecture. A User interacts with a Kinect with Mini PC by using sign language. The system shows text to the user and performs Hand Detection / Recognition on hand gestures. The Kinect with Mini PC also communicates with an Opponent using TSS (Text-to-Speech) and STT (Speech-to-Text). A Translation process connects the Kinect with Mini PC to a Hand Gesture component. This component uploads recognized gestures to a Pictures (Database).</p> </div> <div data-bbox="368 969 676 1003"> <p><그림 2. 시스템 구성도></p> </div>	<div data-bbox="943 416 1377 920"> <ul style="list-style-type: none"> - 청각장애인을 위한 휴대용 수화 통역기(Carrytranslator)는 Kinect와 Mini PC로 구성되어 있다. - 사용자가 수화를 시작하면, 학습한 정보를 토대로 일치하는 글자나 단어를 불러온다. 이를 문장으로 구성하여 Mini PC에서 음성으로 상대방에게 전달한다. - 상대방의 음성이 Mini PC에 텍스트로 나타난다. </div>

II. 프로젝트 수행결과

1. 주요기능

구 분	기 능	설 명
S/W	MatLab	키넥트의 필요한 코드를 작성하는 프로그램이다. 이미지와 영상을 저장할 수 있다. 구글넷을 이용한 이미지 및 영상 인식을 하여 확률값이 가장 높은 결과를 도출한다.
	Google Speech(API)	구글에서 제공하는 오픈 api로 음성인식 및 텍스트로 바꾸어주는 역할을 한다.
H/W	Kinect	영상을 인식하여 수화데이터를 가져오는 역할을 한다.
	Mini pc	키넥트를 사용하기 위해서는 window로 사용하는 것이 편리하므로 라즈베리파이가 아닌 휴대성이 가능한 미니피시를 이용한다.

2. 개발환경

구 분	항 목	적용내역
S/W 개발환경	OS	Window 10
	개발환경(IDE)	Kinect for Window SDK v1.8
	개발도구	Matlab
	개발언어	Matlab
H/W 구성장비	디바이스	Kinect for Window v1, Mini PC

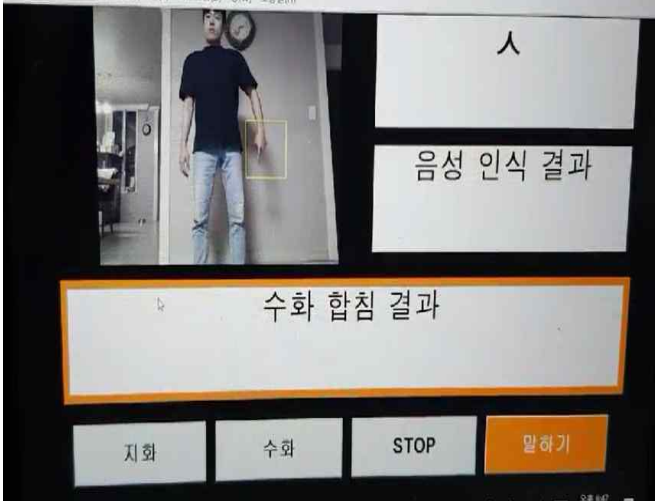
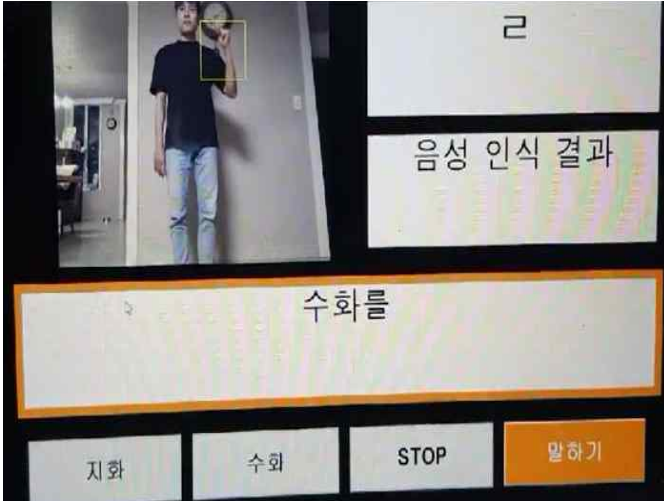
3. 장비(기자재/재료) 활용

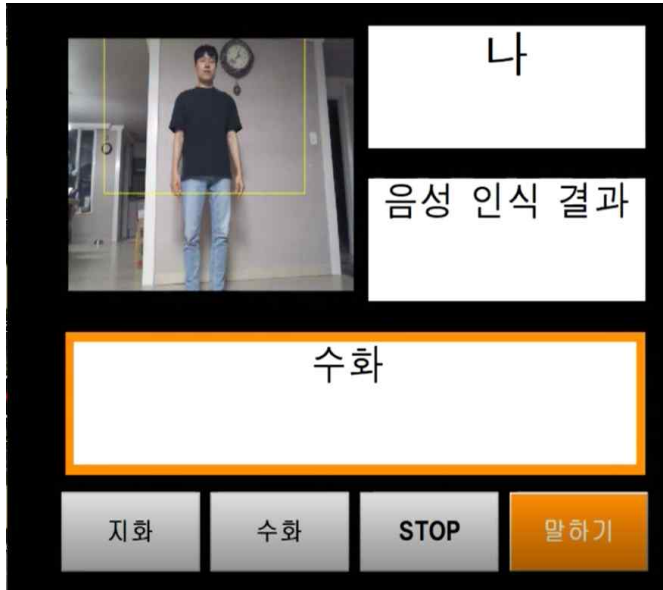
NO	품 명	작품에서의 주요기능
1	Kinect for Window v1	수화자를 촬영하고 Skeleton을 추출하는 기능
2	Mini PC	수화자에게 번역된 결과를 텍스트로 보여주고 텍스트를 음성으로 송출하는 기능

4. 프로그램 작동 동영상

- URL : <https://youtu.be/ehnBKHfFvtA>

5. 결과물 이미지

최종 결과물 이미지	설명
<div data-bbox="127 360 785 860">  <p data-bbox="327 907 584 940"><그림 3. 지화 번역></p> </div> <div data-bbox="127 1039 794 1538">  <p data-bbox="327 1590 584 1624"><그림 4. 지화 합침></p> </div>	<p>지화 실행버튼을 누르면 키넥트가 손을 추적하고 크롭한 후, 결과를 분석하여 아래에 결과를 띄워주고 음성으로 변환시켜준다.</p> <p>그 후에 합침을 뜻하는 손 모양을 인식하면 그 전의 지화가 합쳐짐으로써 단어나 문장으로 완성된다. 완성된 단어나 문장 또한 음성으로 변환되어 나온다.</p>



<그림 5. 수화 번역>



<그림 6. 수화 합침>

궤적 실행버튼을 누르면 키넥트가 척추를 기반으로 추적하여 사용자의 상체를 크롭한다. 그 후 초당 10프레임으로 2초간 영상을 촬영한 이후 결과를 분석하여 아래에 결과를 띄워주고 음성으로 변환시켜준다.

그 후 STOP버튼을 누르게 되면 그 전까지의 수화가 합쳐진 결과가 나타나게 되며 합쳐진 결과 또한 음성으로 변환 되어 나온다.

6. 달성성과

■ 논문게재 및 포스터발표	게재(발표)자명	논문(포스터)명	게재(발표)처	게재(발표)일자
	강지연	심층 학습을 이용한 실시간 한국 수화 인식 시스템	한국정보과학회	미정
□ 앱 등록	등록자명	앱명	등록처	등록일자
■ 프로그램 등록	등록자명	프로그램명	등록처	등록일자
	황정환	키넥트를 활용한 휴대용 수화번역기	한국저작권위원회	미정
■ 특허/실용신안 출원	출원자명	특허/실용신안명	출원번호	출원일자
	황정환	농아인을 위한 휴대용 수화번역기	미정	미정
□ 기술이전	기술이전기업명	기술명	금액	이전일자
■ 공모전	구분(교내/교외)	공모전명	수상여부(출품/수상)	상격
	교외	프로보노 공모전	출품	
□ 실용화	구체적인 내용			
□ 기타	구체적인 내용			

Ⅲ. 프로젝트 수행방법

1. 업무분장

역할	성명	소속	담당역할
멘토	유형석	더존비즈온	프로젝트 관리 및 개발 방법 조언
지도교수	이은규	인천대학교	데이터 테스트 방법 조언
팀장	황정환	인천대학교	구글 스피치(API) 음성인식 SW담당
팀원	안이삭	인천대학교	수화 궤적 알고리즘, 수화학습 SW담당
팀원	육성현	서울시립대학교	Kinect 모션 인식 알고리즘, 수화학습 머신러닝 SW담당
팀원	강지연	서울시립대학교	Kinect 모션 인식 알고리즘, 수화학습 SW담당

2. 프로젝트 수행일정

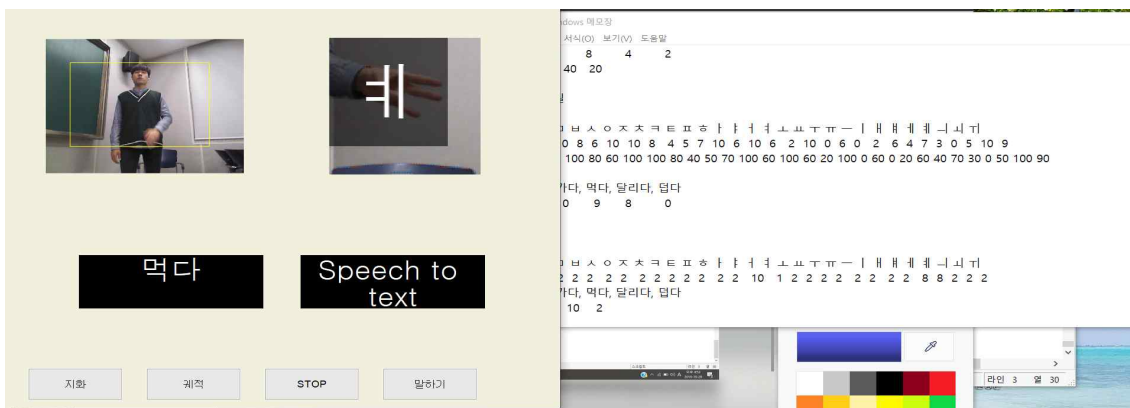
구 분	수행내용	수행일정								
		3월	4월	5월	6월	7월	8월	9월	10월	11월
계획	수행계획서 작성 및 앞으로 계획작성		V							
분석	필요한 장비물품 선택 및 수화 알고리즘 분석, 농아인들에게 기기에 대한 문제점을 알기 위한 설문조사		V							
설계	마니피셔와 키넥트 연결			V						
	미니피셔와 디스플레이, 마이크 연결			V						
개발	구글 스피치등 api이용 음성인식 기술 적용			V	V	V	V			
	키넥트를 활용한 수화 모션인식 알고리즘 개발			V	V	V	V			
	키넥트를 이용한 수화 이미지 추출			V	V	V	V			
	구글넷 이용 인식한 모션을 수화 동작과 맞는지 확인 및 결과값 추출			V	V	V	V			
테스트	기능 테스트				V	V	V			
	연동 테스트					V	V	V	V	
	통합 테스트							V	V	V
종료	보고서 작성 및 종료									V
오프라인 미팅	월 1회 멘토와 미팅 격주로 온/오프라인 팀원들과 미팅	V	V	V	V	V	V	V	V	V

3. 문제점 및 해결방안

문제점1	해결방안
HOG와 SVM을 이용해 학습한 결과의 정확도가 높지 않았다.	GoogLeNet을 이용하여 지화와 동작을 모두 학습할 수 있고, 딥러닝을 통해 정확도를 높였다.

문제점2	해결방안
정적인 이미지를 분석하는 지화 분석은 비교적 쉽고 간단하지만, 동적인 이미지를 분석하는 궤적부분에서 학습하는 데에 있어 어려움이 있다.	키넥트를 활용하여 영상을 촬영하고 그 데이터를 기반으로 다시 궤적을 학습하여 정확도를 높였다.

문제점3	해결방안
우리가 하고자하는 프로젝트는 휴대성을 가진 수화번역기이므로 농아인들이 외출 시 다른 장소에서도 사용해야한다. 그래서 실제로 다른 장소에서 지화와 수화의 성공률을 측정해보았으나 정확도는 높지 않았다.	학습 장소를 한 곳으로만 하면 다른 장소에서의 정확도가 현저히 떨어질 것으로 판단하여, 장소 두 곳을 정하여 지화와 수화에 대한 데이터 및 수화 결과를 수집하여 축적하고 학습을 통하여 정확도를 높아짐을 확인하였다.



<그림 7. 문제점3 - 해결방안>

4. 정확도 비교

수화	분석 기법 비교		장소1(휴식공간)		장소2(강의실)	
	HOG+SVM	GoogLeNet	학습 전	학습 후	학습 전	학습 후
ㄱ	96%	100%	100%	100%	100%	100%
ㄴ	5%	100%	70%	100%	90%	100%
ㄷ	56%	100%	80%	100%	60%	100%
ㄹ	97%	90%	70%	80%	70%	100%
ㅁ	86%	100%	100%	100%	100%	100%
ㅂ	52%	100%	60%	100%	80%	100%
ㅅ	30%	100%	40%	100%	60%	100%
ㅇ	52%	100%	90%	100%	100%	100%
ㅈ	36%	75%	0%	50%	100%	100%
ㅊ	8%	100%	30%	100%	80%	100%
ㅋ	97%	100%	0%	100%	40%	100%
ㅌ	69%	100%	80%	100%	50%	100%
ㅍ	9%	100%	20%	100%	70%	100%
ㅎ	82%	100%	100%	100%	100%	100%
ㅊ	96%	100%	20%	100%	60%	100%
ㅊ	75%	100%	50%	100%	100%	100%
ㅊ	7%	100%	60%	100%	60%	100%
ㅊ	64%	15%	0%	0%	20%	10%
ㅊ	47%	100%	90%	100%	100%	100%
ㅊ	50%	100%	0%	100%	0%	100%
ㅊ	30%	100%	80%	100%	60%	100%
ㅊ	28%	80%	50%	80%	0%	80%
ㅊ	92%	100%	30%	100%	20%	100%
ㅊ	95%	100%	10%	100%	60%	100%
ㅊ	72%	100%	10%	100%	40%	100%
ㅊ	34%	100%	10%	100%	70%	100%
ㅊ	66%	75%	10%	50%	30%	80%
ㅊ	23%	95%	0%	100%	0%	80%
ㅊ	48%	100%	10%	100%	50%	100%
ㅊ	33%	100%	100%	100%	100%	100%
ㅊ	39%	100%	60%	100%	90%	100%
안녕하세요	-	-	70%	60%	30%	90%
가다	-	-	0%	90%	0%	100%
먹다	-	-	80%	80%	90%	70%
달리다	-	-	40%	100%	80%	90%
덥다	-	-	20%	100%	0%	90%

<표 1. 문제점 해결 정확도 비교>

IV. 기대효과 및 활용분야

1. 기대효과

○ 1 : 1 대화 가능

기존에는 농아인과 비장애인이 대화하기 위해서 수화통역사가 필요했다. 하지만 본 프로젝트가 개발하는 수화번역 기기를 사용하여 농아인과 비장애인 사이에 제3자 없이 대화할 수 있다.

○ 농아인과 비장애인 사이 의사소통의 불편함을 감소

이전엔 농아인과 비장애인 사이 의사소통을 텍스트를 입력하며 해결했다. 하지만 텍스트를 입력하면 감정표현이 어렵고, 수화보다 느려 불편함이 따른다. 이를 해소하고자 실시간으로 수화번역을 할 수 있는 기기를 개발하고자 한다.

○ 농인의 생활 범위 확대

특수한 상황이 아니더라도 여러 사회에 참여할 수 있어 이전보다 더 넓은 생활 범위를 가질 수 있다.

○ 수화 통역 시 비용 절감

농인이 수화통역사를 무료로 이용할 수 있는 횟수와 시간이 한정되어 있어 이 시간을 넘기면 개인적으로 고용해야 한다. 이 비용이 한 번에 7만 원 정도인데, 이 기기를 이용하게 되면 이보다 적은 비용을 들여 통역할 수 있다.

○ 한국 수어 사용자를 위한 수화번역 기기 개발

외국은 농아인을 위한 수화번역 기기가 많지만, 한국은 아직 한국어를 사용한 수화번역 기기가 없다. 이에 본 프로젝트는 한국만의 수화번역 기기를 개발하여 한국에 있는 농아인들이 비장애인과 소통에 있어 한 발짝 더 나아가는 계기를 만들고자 한다.

○ 참여 멘티의 교육적 기대효과

- 미니피시 활용법, 영상처리 등의 학습과 직접 활용을 통한 학습 및 개발 역량 강화
- 농아인과 그들의 입장에 대한 전반적인 이해를 통한 맞춤 프로젝트를 통해 서로에게 한 발짝 더 다가가는 기회 마련

- 팀원들과의 의견충돌 또는 학습을 진행하는데 어려움 발생 시 팀원들과의 이견조율과 협동을 통해 서로를 위한 배려를 배울 수 있으며 이처럼 단체 생활에 필요한 점들을 경험할 수 있음

2. 활용분야

○ 농아인의 의사소통이 필요한 모든 곳

농아인이 직접 휴대하면서 수화를 번역해주는 서비스를 제공하기 위해 제작되지만, 금융이나 동사무소, 공공 사업장과 같은 곳에서 본 프로젝트의 기기를 설치해두어 농아인이 기기를 휴대하지 않아도 의사소통을 원활하게 할 수 있다.

○ 후천적 농아인들을 위한 수화 교육용 프로그램

농아인 중에서 선천적으로 장애를 갖고 태어난 사람도 있지만, 후천적으로 장애를 가지게 된 사람들도 있다. 이들은 어릴 때부터 수화를 배운 것이 아니므로 수화를 배우고 사용하는 데에 큰 어려움이 있다. 본 프로젝트에서는 수화인식프로그램을 기반으로 수화 교육용 프로그램을 만들어 활용하는 것을 기대한다.

○ 일상생활

비장애인과 대화의 원활해지므로 직업 선택의 폭이 넓어지며 생활 범위가 확대된다. 또한, 수화이미지를 활용하여 강의나 영화 같은 영상의 소리를 수화로 표현하거나 텍스트로 출력하여 농아인이 교육이나 문화에 접근하는 데 있어서 불편함을 해소하고 한 발짝 나아갈 수 있을 것이다.

3. 추후 개발 방향

○ 음성을 텍스트로 변환 후 영상으로 표현

한글을 모르는 수화자를 위해 음성을 텍스트로 변환할 때, 이를 영상으로 표현할 수 있다.

○ 클라우드를 이용한 신경망 공유

인공 신경망만 전달하면 이 프로그램을 사용할 수 있으나 한 사용자가 신경망을 업데이트 했을 때, 다른 사용자의 것은 변경되지 않는다. 이러한 점을 클라우드를 이용해 공유할 수 있도록 한다.

V. 참고자료

1. 논문

- 조선영, 변혜란, 이희경, 차지훈, "키넥트 센서 데이터를 이용한 손 제스처 인식," 방송공학회논문지, Vol.17, No.3, pp.447-457, 2015.
- 장은아, 김보연, "청각장애인 대상 사용자 니즈조사를 통한 수화 통역 앱 GUI 가이드라인 - 시립 서대문 농아인 복지관을 중심으로 -," 디자인융복합연구, Vol.12, No.6, pp. 211-225, 2013.
- 이현석, 김승필, 정완영, "키넥트의 모션 인식 기능을 이용한 수화번역 시스템 개발," 信號處理·시스템學會論文誌, Vol.14, No.4, pp. 235-242, 2013.

[별첨] 프로보노 ICT멘토링 프로젝트 산출물 증빙

※ 소스코드 등 프로젝트 수행 중 발생한 산출물 및 달성성과 증빙 첨부

[프로그램 코드]

1. 지화 이미지 저장

```
% handGestureDataset.m
% GUI 기반 지화 이미지 저장 프로그램
function handGestureDataset()

% 키넥트 초기화
% 스켈레톤 인식을 위한 depth 캠
depthVid = videoinput('kinect', 2);
triggerconfig(depthVid, 'manual');
depthVid.FramesPerTrigger = 1;
depthVid.TriggerRepeat = inf;
set(getselectedsource(depthVid), 'TrackingMode', 'Skeleton');

% color 캠 초기화
colorVid = videoinput('kinect', 1);
triggerconfig(colorVid, 'manual');
colorVid.FramesPerTrigger = 1;
colorVid.TriggerRepeat = inf;

% 함수를 위한 타이머 설정
t = timer('TimerFcn', @dispDepth, 'Period', 0.05, ...
    'executionMode', 'fixedRate');

% GUI 프레임워크 설정
window=figure('Color',[0.9255 0.9137 0.8471],'Name','Depth Camera',...
    'DockControl','off','Units','Pixels',...
    'toolbar','none',...
    'Position',[50 50 800 600]);

% 지화 이미지 저장을 시작하기 위한 버튼 설정
startb=uicontrol('Parent',window,'Style','pushbutton','String',...
    'START',...
    'FontSize',11 ,...
    'Units','normalized',...
    'Position',[0.22 0.02 0.16 0.08],...
    'Callback',@startCallback);

% 지화 이미지 저장을 멈추기 위한 버튼 설정
stopb=uicontrol('Parent',window,'Style','pushbutton','String',...
    'STOP',...
    'FontSize',11 ,...
    'Units','normalized',...
    'Position',[0.5 0.02 0.16 0.08],...
    'Callback',@stopCallback);

% 변수 초기화
i = 0;
m = 0;

% 깊이를 보여주기 위한 함수 선언
function dispDepth(obj, event)

    % 영상 출력(0~4096 로 프레임 재지정)
    trigger(colorVid);
    trigger(depthVid);
    [depthMap, ~, depthMetaData] = getdata(depthVid);
    [colorMap, ~, colorMetaData] = getdata(colorVid);
    idx = find(depthMetaData.IsSkeletonTracked);
    subplot(2,2,1);
    imshow(depthMap, [0 4096]);
```

```

% 영상 처리
% 스켈레톤 추적이 됐을 때
if idx ~= 0

    % 오른손 위치 추적
    rightHand = depthMetaData.JointDepthIndices(12, idx);

    % 오른손 데이터값 추출
    zCoord = 1e3*min(depthMetaData.JointWorldCoordinates(12, idx));
    radius = round(90 - zCoord / 50);
    rightHandBox = [rightHand-0.5*radius 1.2*radius 1.2*radius];

    % 사각형으로 오른손 크롭 후 화면에 표시
    rectangle('position', rightHandBox, 'EdgeColor', [1 1 0]);
    handColorImage = imcrop(colorMap, rightHandBox);
    result = rgb2gray(handColorImage);
    subplot(2,2,3);
    imshow(handColorImage, [0 4096]);

    % 데이터 추출이 됐을 때
    if ~isempty(handColorImage)

        % 배경 전처리
        imageSize = size(handColorImage);

        for k = 1:imageSize(1)
            for j = 1:imageSize(2)
                if handColorImage(k, j) > 2300
                    handColorImage(k, j) = 0;
                end
            end
        end

        % 지화 이미지를 폴더에 저장
        i = i+1;
        if (mod(i,5)==1)
            %원하는 문자를 넣어서 학습
            imwrite(imresize(handColorImage,[224,224]),
strcat('hangeul/ㄱ/ㄱ_', num2str(m), '.png'), 'png');
            m=m+1;
        end
    end
end
end
end
% 각 기능에 대한 callback 함수 선언
function startCallback(obj, event)
    start(colorVid);
    start(depthVid);
    start(t);
end
function stopCallback(obj, event)
    stop(t);
    stop(colorVid);
    stop(depthVid);
    m=0;
end
end
end

```

2. 지화 이미지 트레이닝

```
% handGestureTraining.m
% 변수 초기화
hangle = ['ㄱ', 'ㄴ', 'ㄷ',
'ㄹ','ㅁ','ㅂ','ㅅ','ㅇ','ㅈ','ㅊ','ㅋ','ㅌ','ㅍ','ㅎ','ㅣ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ','ㅊ'];
trainingFeatures=[];
trainlabel=[];

%hangeul 폴더에 있는 데이터 불러오기
imds = imageDatastore('hangeul', ...
    'IncludeSubfolders',true, ...
    'LabelSource','foldernames');

%학습할 데이터량 설정
[imdsTrain,imdsValidation] = splitEachLabel(imds,0.7);

%구글넷 설정 - 사용자의 환경에 따라 다른 네트워크로 설정 가능
net = googlenet;
analyzeNetwork(net);
inputSize = net.Layers(1).InputSize;

%레이어 설정
if isa(net,'SeriesNetwork')
    lgraph = layerGraph(net.Layers);
else
    lgraph = layerGraph(net);
end

[learnableLayer,classLayer] = findLayersToReplace(lgraph);
[learnableLayer,classLayer]
%라벨 설정
numClasses = numel(categories(imdsTrain.Labels));

%레이어에 따른 부가 설정
if isa(learnableLayer,'nnet.cnn.layer.FullyConnectedLayer')
    newLearnableLayer = fullyConnectedLayer(numClasses, ...
        'Name','new_fc', ...
        'WeightLearnRateFactor',10, ...
        'BiasLearnRateFactor',10);
elseif isa(learnableLayer,'nnet.cnn.layer.Convolution2DLayer')
    newLearnableLayer = convolution2dLayer(1,numClasses, ...
        'Name','new_conv', ...
        'WeightLearnRateFactor',10, ...
        'BiasLearnRateFactor',10);
end

lgraph = replaceLayer(lgraph,learnableLayer.Name,newLearnableLayer);
newClassLayer = classificationLayer('Name','new_classoutput');
lgraph = replaceLayer(lgraph,classLayer.Name,newClassLayer);
```

```

%시각화
figure('Units','normalized','Position',[0.3 0.3 0.4 0.4]);
plot(lgraph)
ylim([0,10])

%레이어 설정과 연결 설정
layers = lgraph.Layers;
connections = lgraph.Connections;

layers(1:10) = freezeWeights(layers(1:10));
lgraph = createLgraphUsingConnections(layers,connections);

%픽셀, 스케일 설정
pixelRange = [-30 30];
scaleRange = [0.9 1.1];
%이미지 처리
imageAugmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandXTranslation',pixelRange, ...
    'RandYTranslation',pixelRange, ...
    'RandXScale',scaleRange, ...
    'RandYScale',scaleRange);
augimdsTrain = augmentedImageDatastore(inputSize(1:2),imdsTrain, ...
    'DataAugmentation',imageAugmenter);
augimdsValidation = augmentedImageDatastore(inputSize(1:2),imdsValidation);
%딥러닝 학습 설정
miniBatchSize = 10;
valFrequency = floor(numel(augimdsTrain.Files)/miniBatchSize);
options = trainingOptions('sgdm', ...
    'MiniBatchSize',miniBatchSize, ...
    'MaxEpochs',6, ...
    'InitialLearnRate',3e-4, ...
    'Shuffle','every-epoch', ...
    'ValidationData',augimdsValidation, ...
    'ValidationFrequency',valFrequency, ...
    'Verbose',false, ...
    'Plots','training-progress');

%학습
net = trainNetwork(augimdsTrain,lgraph,options);
%학습된 결과 저장
save('..net','net');
%테스트
[YPred,probs] = classify(net,augimdsValidation);
accuracy = mean(YPred == imdsValidation.Labels)
idx = randperm(numel(imdsValidation.Files),4);
figure
%임의 추출한 이미지 테스트
for i = 1:4
    subplot(2,2,i)
        I = readimage(imdsValidation,idx(i));

```

```

imshow(I)
label = YPred(idx(i));
title(string(label) + ", " + num2str(100*max(probs(idx(i),:)),3) + "%");
end

```

3. 수화 영상 저장

```

% bodyGestureDataset.m
% GUI 기반 궤적 영상 저장 프로그램
function bodyGestureDataset()

% 전역 변수 선언
clear x;
global x;

% 키넥트 초기화
% 스켈레톤 인식을 위한 depth 캠
depthVid = videoinput('kinect', 2);
triggerconfig(depthVid, 'manual');
depthVid.FramesPerTrigger = 1;
depthVid.TriggerRepeat = inf;
set(getselectedsource(depthVid), 'TrackingMode', 'Skeleton');

% color 캠 초기화
colorVid = videoinput('kinect', 1);
triggerconfig(colorVid, 'manual');
colorVid.FramesPerTrigger = 1;
colorVid.TriggerRepeat = inf;

% 함수를 위한 타이머 설정
t = timer('TimerFcn', @dispDepth, 'Period', 0.1, ...
    'executionMode', 'fixedRate');

% GUI 프레임워크 설정
window=figure('Color',[0.9255 0.9137 0.8471],'Name','Depth Camera',...
    'DockControl','off','Units','Pixels',...
    'toolbar','none',...
    'Position',[50 50 800 600]);

% 궤적 영상 저장을 시작하기 위한 버튼 설정
startb=uicontrol('Parent',window,'Style','pushbutton','String',...
    'START',...
    'FontSize',11 ,...
    'Units','normalized',...
    'Position',[0.22 0.02 0.16 0.08],...
    'Callback',@startCallback);

% 궤적 영상 저장을 멈추기 위한 버튼 설정
stopb=uicontrol('Parent',window,'Style','pushbutton','String',...
    'STOP',...
    'FontSize',11 ,...
    'Units','normalized',...
    'Position',[0.5 0.02 0.16 0.08],...
    'Callback',@stopCallback);

```

```

% 변수 초기화
i = 0;
m=0;

% 깊이를 보여주기 위한 함수 선언
function dispDepth(obj, event)

    % 영상 출력(0~4096 로 프레임 재지정)
    trigger(colorVid);
    trigger(depthVid);
    [depthMap, ~, depthMetaData] = getdata(depthVid);
    [colorMetaData] = getdata(colorVid);
    idx = find(depthMetaData.IsSkeletonTracked);
    subplot(2,2,1);
    imshow(colorMetaData, [0 4096]);

    % 영상 처리
    % 스켈레톤 추적이 됐을 때
    if idx ~= 0
        % 상체 위치 추적
        body = depthMetaData.JointDepthIndices(3,;idx);
        % 상체 데이터값 추출
        radius = 300;
        bodyBox = [body(1)-0.75*radius body(2)-0.3*radius 1.5*radius radius];
        % 사각형으로 상체 크롭 후 화면에 표시
        rectangle('position', bodyBox, 'EdgeColor', [1 1 0]);
        bodyImage = imcrop(colorMetaData,bodyBox);
        % 데이터 추출이 됐을 때
        if ~isempty(bodyImage)

            m=m+1;
            x(:,;m)= imresize(bodyImage,[300,450]);

            % 30 프레임이 되었을 때 꺾적 영상을 폴더에 저장
            if(m==30)
                i = i+1;
                %원하는 동작을 입력하여 저장
                outputVideo = VideoWriter(fullfile(strcat('수화영상폴더/예시/예시','_',num2str(i))));
                outputVideo.FrameRate = 10;
                open(outputVideo)

                for ii = 1:30
                    writeVideo(outputVideo,mat2gray(x(:,;ii)));
                end

                close(outputVideo)
                m=0;
            end
        else
            m=0;
        end
    end
end

% 각 기능에 대한 callback 함수 선언
function startCallback(obj, event)
    start(colorVid);
    start(depthVid);

```

```

        start(t);
    end
    function stopCallback(obj, event)
        stop(t);
        stop(colorVid);
        stop(depthVid);
        m=0;
    end
end
end

```

4. 수화 영상 트레이닝

```

% bodyGestureTraining.m
% 궤적 영상 학습하여 netLSTM으로 저장

% 전역 변수 선언
clear;
clc;
netCNN = googlenet;
%영상 저장 폴더 설정
dataFolder = "수화영상폴더";
%영상 전처리
[files, labels] = hmd51Files(dataFolder);

%딥러닝 학습을 위한 설정
inputSize = netCNN.Layers(1).InputSize(1:2);
layerName = "pool5-7x7_s1";

%미리 영상을 처리한 파일이 있을 시 호출
tempFile = fullfile(tempdir,"kinect.mat");

%영상->데이터 처리
for i = 1:numFiles
    fprintf("Reading file %d of %d...\n", i, numFiles)

    video = readVideo(files(i));
    video = centerCrop(video,inputSize);
    sequences{i,1} = activations(netCNN,video,layerName,'OutputAs','columns');
end

%전처리 파일 저장
save(tempFile,"sequences","-v7.3");

%학습 설정
numObservations = numel(sequences);
idx = randperm(numObservations);
%학습할 데이터량 설정
N = floor(0.7 * numObservations);

```



```

%데이터 전처리
idxTrain = idx(1:N);
sequencesTrain = sequences(idxTrain);
labelsTrain = labels(idxTrain);
idxValidation = idx(N+1:end);
sequencesValidation = sequences(idxValidation);
labelsValidation = labels(idxValidation);
numObservationsTrain = numel(sequencesTrain);
sequenceLengths = zeros(1,numObservationsTrain);

for i = 1:numObservationsTrain
    sequence = sequencesTrain{i};
    sequenceLengths(i) = size(sequence,2);
end

figure
histogram(sequenceLengths)
title("Sequence Lengths")
xlabel("Sequence Length")
ylabel("Frequency")

maxLength = 400;
idx = sequenceLengths > maxLength;
sequencesTrain(idx) = [];
labelsTrain(idx) = [];

numFeatures = size(sequencesTrain{1},1);
numClasses = numel(categories(labelsTrain));

%구글넷 레이어 설정
layers = [
    sequenceInputLayer(numFeatures,'Name','sequence')
    bilstmLayer(2000,'OutputMode','last','Name','bilstm')
    dropoutLayer(0.5,'Name','drop')
    fullyConnectedLayer(numClasses,'Name','fc')
    softmaxLayer('Name','softmax')
    classificationLayer('Name','classification')];

%배치 사이즈 설정
miniBatchSize = 16;
numObservations = numel(sequencesTrain);
numIterationsPerEpoch = floor(numObservations / miniBatchSize);
numObservations
numIterationsPerEpoch

%최종 학습 설정
options = trainingOptions('adam', ...
    'MiniBatchSize',miniBatchSize, ...

```

```

'InitialLearnRate',1e-4, ...
'GradientThreshold',2, ...
'Shuffle','every-epoch', ...
'ValidationData',{sequencesValidation,labelsValidation}, ...
'ValidationFrequency',numIterationsPerEpoch, ...
'Plots','training-progress', ...
'Verbose',false);

%데이터 학습
[netLSTM,info] = trainNetwork(sequencesTrain,labelsTrain,layers,options);
%학습한 데이터를 netLSTM 파일에 저장
save('..netLSTM','netLSTM');

%기존에 학습하지 않은 데이터로 실험
YPred = classify(netLSTM,sequencesValidation,'MiniBatchSize',miniBatchSize);
YValidation = labelsValidation;
accuracy = mean(YPred == YValidation)

%비디오 불러오는 함수
function video = readVideo(filename)

vr = VideoReader(filename);
H = vr.Height;
W = vr.Width;
C = 3;

% Preallocate video array
numFrames = floor(vr.Duration * vr.FrameRate);
video = zeros(H,W,C,numFrames);

% Read frames
i = 0;
while hasFrame(vr)
    i = i + 1;
    video(:,:,i) = readFrame(vr);
end

% Remove unallocated frames
if size(video,4) > i
    video(:,:,i+1:end) = [];
end
end

```

5. 메인 코드

```
% translator.m
% GUI 기반 수화인식 프로그램

function translator(net, netLSTM)
%데이터 저장할 자바 연결리스트 생성
import java.util.LinkedList
q = LinkedList();
%초성 중성 종성의 값
cho1 = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18];
jung1 = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20];
jong1 = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27];
%전역 변수 선언
temp1 = [];
temp2 = 0;
temp3 = [""];
countT = 1;
accuracyR = ["a", "b"];
clear k;
global k;
m = 0;

% googleNet input 설정
netCNN = googlenet;
inputSize = netCNN.Layers(1).InputSize(1:2);
layerName = "pool5-7x7_s1";

% 키넥트 초기화
colorVid = videoinput('kinect', 1);
depthVid = videoinput('kinect', 2);

% 스켈레톤 인식을 위한 depth 캠
triggerconfig(depthVid, 'manual');
depthVid.FramesPerTrigger = 1;
depthVid.TriggerRepeat = inf;
set(getselectedsource(depthVid), 'TrackingMode', 'Skeleton');

% color 캠 초기화
triggerconfig(colorVid, 'manual');
colorVid.FramesPerTrigger = 1;
colorVid.TriggerRepeat = inf;

% 함수를 위한 타이머 설정
t2 = timer('Period', 0.1, 'ExecutionMode', 'fixedRate');
t2.TimerFcn = @dispDepth2;
t = timer('Period', 0.1, 'ExecutionMode', 'fixedRate');
t.TimerFcn = @dispDepth;
t3 = timer('Period', 10, 'ExecutionMode', 'fixedRate');
t3.TimerFcn = @speechfc;
```

```

% GUI 프레임워크 설정
window=figure('Color',[0, 0, 0],'Name','Depth Camera',...
    'DockControl','off','Units','Pixels',...
    'toolbar','none',...
    'Position',[50 50 800 600]);
padd = uicontrol('Parent',window,'Style','text');
set(padd,'String',' ','position',[70 120 670 140])
padd.BackgroundColor = [1, 0.55 , 0];
% 수화 번역 결과창 설정
b = uicontrol('Parent',window,'Style','text');
set(b,'String','수화 합침 결과','position',[80 130 650 120])
b.BackgroundColor = [1, 1 , 1];
b.ForegroundColor = 'black';
b.FontName = 'Dotum';
b.FontSize = 30;
b.FontWeight = 'bold';

d = uicontrol('Parent',window,'Style','text');
set(d,'String','수화,지화 번역 결과','position',[420 440 320 120])
d.BackgroundColor = [1, 1 , 1];
d.ForegroundColor = 'black';
d.FontName = 'Dotum';
d.FontSize = 38;
d.FontWeight = 'bold';

% STT 결과창 설정
c = uicontrol('Parent',window,'Style','text');
set(c,'String','음성 인식 결과','position',[420 290 320 120])
c.BackgroundColor = [1, 1 , 1];
c.ForegroundColor = 'black';
c.FontName = 'Dotum';
c.FontSize = 30;
c.FontWeight = 'bold';

% 지화를 인식하기 위한 버튼 설정
startb1=uicontrol('Parent',window,'Style','pushbutton','String',...
    '지화',...
    'FontSize',20 ,...
    'Units','normalized',...
    'Position',[0.08 0.04 0.2 0.13],...
    'Callback',@startCallback);
startb1.FontWeight = 'bold';

% 궤적을 인식하기 위한 버튼 설정
startb=uicontrol('Parent',window,'Style','pushbutton','String',...
    '수화',...
    'FontSize',20 ,...
    'Units','normalized',...
    'Position',[0.295 0.04 0.2 0.13],...

```

```

'Callback',@startCallback2);
startb.FontWeight = 'bold';

% 프로그램을 멈추기 위한 버튼 설정
stopb=uicontrol('Parent',window,'Style','pushbutton','String',...
    'STOP',...
    'FontSize',20 ,...
    'Units','normalized',...
    'Position',[0.51 0.04 0.2 0.13],...
    'Callback',@stopCallback);
stopb.FontWeight = 'bold';

% 음성 인식을 하기 위한 버튼 설정
speechb=uicontrol('Parent',window,'Style','pushbutton','String',...
    '말하기',...
    'FontSize',20 ,...
    'Units','normalized',...
    'Position',[0.725 0.04 0.2 0.13],...
    'Callback',@speechCallback);
speechb.BackgroundColor = '#ff8c00';
speechb.ForegroundColor = 'white';
speechb.FontWeight = 'bold';

% 스피치 함수 선언
function speechfc(obj, event)
    % 녹음 시작
    recObj = audiorecorder(44100, 16, 1);
    speechObject = speechClient('Google','languageCode','ko-KR');
    disp('Start speaking.')
    recordblocking(recObj, 5);
    disp('End of Recording.')

    % 녹음한 음성을 파일로 저장후 load
    filename = 'sample.wav';
    y = getaudiodata(recObj);
    audiowrite(filename, y, 48000);
    [samples, fs] = audioread('sample.wav');

    % 음성 파일을 STT로 내보냄
    outInfo = speech2text(speechObject, samples, fs);
    result = outInfo.Transcript;
    set(c,'String', result,'position',[420 290 320 120])
end

% 지화 함수
function dispDepth(obj, event)

    % 영상 출력(0~4096 로 프레임 재지정)
    trigger(depthVid);
    trigger(colorVid);
    [depthMap, ~, depthMetaData] = getdata(depthVid);

```

```

[colorFrameData] = getdata(colorVid);
idx = find(depthMetaData.IsSkeletonTracked);
ax = subplot(2,2,1);
set(ax, 'position', [0.09,0.43 0.41 0.55]);
imshow(colorFrameData);

% 영상 처리
% 스켈레톤 추적이 됐을 때
if idx ~= 0
    % 오른손 위치 추적
    rightHand = depthMetaData.JointDepthIndices(12,;idx);

    % 오른손 데이터값 추출
    zCoord = 1e3*min(depthMetaData.JointWorldCoordinates(12,;idx));
    radius = round(90 - zCoord / 50);
    rightHandBox = [rightHand-0.5*radius 1.2*radius 1.2*radius];

    % 사각형으로 오른손 크롭
    rectangle('position', rightHandBox, 'EdgeColor', [1 1 0]);
    handDepthImage = imcrop(colorFrameData,rightHandBox);

    % 데이터 추출이 됐을 때
    if ~isempty(handDepthImage)
        temp = imresize(handDepthImage, [224 224]);

        % 구글넷을 활용한 결과 예측
        YPred = classify(net,temp);
        result = string(YPred);

        accuracyR(countT) = result;
        %합침을 뜻하는 ㅁ이 들어오면 초 중 종성 계산 및 데이터 저장
        if result == 'ㅁ'
            q.add('!');
            q.add('@');
            for j=1:6
                if q.get(0)=='!'
                    break;
                elseif q.get(0)~='!'
                    temp1(j)=0;
                end
            end
            for i=0:q.size()-1
                if q.size() == 2 || q.size() == 1
                    break
                end
                if (q.get(i) == 'ㄱ' || q.get(i) == 'ㄴ' || q.get(i) == 'ㄷ' || q.get(i) == 'ㄹ'
||q.get(i) == 'ㅁ' ||q.get(i) == 'ㅂ' ||q.get(i) == 'ㅅ' ||q.get(i) == 'ㅇ' ||q.get(i) == 'ㅈ' ||q.get(i) == 'ㅊ' ||q.get(i)
== 'ㅋ' ||q.get(i) == 'ㅌ' ||q.get(i) == 'ㅍ' ||q.get(i) == 'ㅎ')&&(q.get(i+1) == 'ㅏ' ||q.get(i+1) == 'ㅑ' ||q.get(i+1)
== 'ㅓ' ||q.get(i+1) == 'ㅕ' ||q.get(i+1) == 'ㅗ' ||q.get(i+1) == 'ㅛ' ||q.get(i+1) == 'ㅜ' ||q.get(i+1) == 'ㅠ' ||q.get(i+1) == 'ㅡ' ||q.get(i+1) == 'ㅣ' ||q.get(i+1) == 'ㅚ' ||q.get(i+1) == 'ㅜ' ||q.get(i+1) == 'ㅠ' ||q.get(i+1) == 'ㅡ' ||q.get(i+1) == 'ㅣ')&&(q.get(i+2) == 'ㄱ' || q.get(i+2) == 'ㄴ' ||
q.get(i+2) == 'ㄷ' || q.get(i+2) == 'ㄹ' ||q.get(i+2) == 'ㅁ' ||q.get(i+2) == 'ㅂ' ||q.get(i+2) == 'ㅅ' ||q.get(i+2)

```

```

== 'ㅇ' ||q.get(i+2) == 'ㅈ' ||q.get(i+2) == 'ㅊ' ||q.get(i+2) == 'ㅋ' ||q.get(i+2) == 'ㅌ' ||q.get(i+2) == 'ㅍ'
||q.get(i+2) == 'ㅎ')

switch(q.get(i))
case 'ㄱ'
temp2 = temp2 + (cho1(1)*588);
case 'ㄴ'
temp2 = temp2 + (cho1(3)*588);
case 'ㄷ'
temp2 = temp2 + (cho1(4)*588);
case 'ㄹ'
temp2 = temp2 + (cho1(6)*588);
case 'ㅁ'
temp2 = temp2 + (cho1(7)*588);
case 'ㅂ'
temp2 = temp2 + (cho1(8)*588);
case 'ㅅ'
temp2 = temp2 + (cho1(10)*588);
case 'ㅇ'
temp2 = temp2 + (cho1(12)*588);
case 'ㅈ'
temp2 = temp2 + (cho1(13)*588);
case 'ㅊ'
temp2 = temp2 + (cho1(15)*588);
case 'ㅋ'
temp2 = temp2 + (cho1(16)*588);
case 'ㅌ'
temp2 = temp2 + (cho1(17)*588);
case 'ㅍ'
temp2 = temp2 + (cho1(18)*588);
case 'ㅎ'
temp2 = temp2 + (cho1(19)*588);
end
elseif (q.get(i) == 'ㄱ' || q.get(i) == 'ㄴ' || q.get(i) == 'ㄷ' || q.get(i) == 'ㄹ'
||q.get(i) == 'ㅁ' ||q.get(i) == 'ㅂ' ||q.get(i) == 'ㅅ' ||q.get(i) == 'ㅇ' ||q.get(i) == 'ㅈ' ||q.get(i) == 'ㅊ' ||q.get(i)
== 'ㅋ' ||q.get(i) == 'ㅌ' ||q.get(i) == 'ㅍ' ||q.get(i) ==
'ㅎ')&&(q.get(i+1)=='ㄷ'||q.get(i+1)=='ㅌ')&&(q.get(i+2)=='ㄱ'||q.get(i+2)=='ㄴ'||q.get(i+2)=='ㄷ'||q.get(i+2)=='ㄹ'
||q.get(i+2)=='ㅁ' ||
switch(q.get(i))
case 'ㄱ'
temp2 = temp2 + (cho1(1)*588);
case 'ㄴ'
temp2 = temp2 + (cho1(3)*588);
case 'ㄷ'
temp2 = temp2 + (cho1(4)*588);
case 'ㄹ'
temp2 = temp2 + (cho1(6)*588);
case 'ㅁ'
temp2 = temp2 + (cho1(7)*588);
case 'ㅂ'
temp2 = temp2 + (cho1(8)*588);
case 'ㅅ'
temp2 = temp2 + (cho1(10)*588);

```

```

        temp2 = temp2 + (cho1(10)*588);
    case 'ㅇ'
        temp2 = temp2 + (cho1(12)*588);
    case 'ㅈ'
        temp2 = temp2 + (cho1(13)*588);
    case 'ㅊ'
        temp2 = temp2 + (cho1(15)*588);
    case 'ㅋ'
        temp2 = temp2 + (cho1(16)*588);
    case 'ㅌ'
        temp2 = temp2 + (cho1(17)*588);
    case 'ㅍ'
        temp2 = temp2 + (cho1(18)*588);
    case 'ㅎ'
        temp2 = temp2 + (cho1(19)*588);
    end
elseif (q.get(i) == 'ㄷ'||q.get(i) ==
'ㅌ')&&(q.get(i+1)== 'ㅌ'||q.get(i+1)== 'ㄴ'||q.get(i+1)== 'ㄹ'||q.get(i+1)== 'ㄷ'||q.get(i+1)== 'ㄴ')&&(q.get(i+2) ==
'ㄱ' || q.get(i+2) == 'ㄴ' || q.get(i+2) == 'ㄷ' || q.get(i+2) == 'ㄹ' ||q.get(i+2) == 'ㅇ' ||q.get(i+2) == 'ㅁ'
||q.get(i+2) == 'ㄴ' ||q.get(i+2) == 'ㅇ' ||q.get(i+2) == 'ㅈ' ||q.get(i+2) == 'ㅊ' ||q.get(i+2) == 'ㅋ' ||q.get(i+2)
== 'ㅌ' ||q.get(i+2) == 'ㅍ' ||q.get(i+2) == 'ㅎ')
    if q.get(i) == 'ㄷ'
        switch(q.get(i+1))
        case 'ㅌ'
            temp2 = temp2 + (jung1(10)*28);
        case 'ㄴ'
            temp2 = temp2 + (jung1(11)*28);
        case 'ㄹ'
            temp2 = temp2 + (jung1(12)*28);
        end
    elseif q.get(i) == 'ㅌ'
        switch(q.get(i+1))
        case 'ㄷ'
            temp2 = temp2 + (jung1(15)*28);
        case 'ㄴ'
            temp2 = temp2 + (jung1(16)*28);
        case 'ㄹ'
            temp2 = temp2 + (jung1(17)*28);
        end
    end
    q.remove();
    q.remove();
    q.remove();
    break;
elseif (q.get(i) == 'ㄷ'||q.get(i) ==
'ㅌ')&&(q.get(i+1)== 'ㅌ'||q.get(i+1)== 'ㄴ'||q.get(i+1)== 'ㄹ'||q.get(i+1)== 'ㄷ'||q.get(i+1)== 'ㄴ')&&
q.get(i+2)=='!'
    if q.get(i) == 'ㄷ'
        switch(q.get(i+1))
        case 'ㅌ'

```


[illegible]

[illegible]

```

        case 'π'
            temp2 = temp2 + (jung1(18)*28);
        case '—'
            temp2 = temp2 + (jung1(19)*28);
        case '└'
            temp2 = temp2 + (jung1(20)*28);
        case '┌'
            temp2 = temp2 + (jung1(21)*28);
    end
    q.remove();
    q.remove();
    break;
elseif (q.get(i) == 'ㄱ' || q.get(i) == 'ㄴ' || q.get(i) == 'ㄷ' || q.get(i) == 'ㄹ'
||q.get(i) == 'ㅁ' ||q.get(i) == 'ㅂ' ||q.get(i) == 'ㅅ' ||q.get(i) == 'ㅇ' ||q.get(i) == 'ㅈ' ||q.get(i) == 'ㅊ' ||q.get(i)
== 'ㅋ' ||q.get(i) == 'ㅌ' ||q.get(i) == 'ㅍ' ||q.get(i) == 'ㅎ')&&(q.get(i+1) == 'ㄱ' || q.get(i+1) == 'ㄴ' ||
q.get(i+1) == 'ㄷ' || q.get(i+1) == 'ㄹ' ||q.get(i+1) == 'ㅁ' ||q.get(i+1) == 'ㅂ' ||q.get(i+1) == 'ㅅ' ||q.get(i+1)
== 'ㅇ' ||q.get(i+1) == 'ㅈ' ||q.get(i+1) == 'ㅊ' ||q.get(i+1) == 'ㅋ' ||q.get(i+1) == 'ㅌ' ||q.get(i+1) == 'ㅍ' ||q.get(i+1)
== 'ㅎ')&&(q.get(i+2) == ' ' ||q.get(i+2) == 'ㅏ' ||q.get(i+2) == 'ㅑ' ||q.get(i+2) == 'ㅓ' ||q.get(i+2)
== 'ㅕ' ||q.get(i+2) == 'ㅗ' ||q.get(i+2) == 'ㅛ' ||q.get(i+2) == 'ㅜ' ||q.get(i+2) == 'ㅠ' ||q.get(i+2) ==
'ㅡ' ||q.get(i+2) == 'ㅣ' ||q.get(i+2) == 'ㅚ' ||q.get(i+2) == 'ㅜ' ||q.get(i+2) == 'ㅠ' ||q.get(i+2) == 'ㅡ' ||q.get(i+2)
== 'ㅣ' ||q.get(i+2) == ' ' )
    switch(q.get(i))
        case 'ㄱ'
            temp2 = temp2 + jong1(2);
        case 'ㄴ'
            temp2 = temp2 + jong1(5);
        case 'ㄷ'
            temp2 = temp2 + jong1(8);
        case 'ㄹ'
            temp2 = temp2 + jong1(9);
        case 'ㅁ'
            temp2 = temp2 + jong1(17);
        case 'ㅂ'
            temp2 = temp2 + jong1(18);
        case 'ㅅ'
            temp2 = temp2 + jong1(20);
        case 'ㅇ'
            temp2 = temp2 + jong1(22);
        case 'ㅈ'
            temp2 = temp2 + jong1(23);
        case 'ㅊ'
            temp2 = temp2 + jong1(24);
        case 'ㅋ'
            temp2 = temp2 + jong1(25);
        case 'ㅌ'
            temp2 = temp2 + jong1(26);
        case 'ㅍ'
            temp2 = temp2 + jong1(27);
        case 'ㅎ'
            temp2 = temp2 + jong1(28);
    end
end

```

```

q.remove();
q.remove();
q.remove();
break;
elseif (q.get(i) == 'ㄱ' || q.get(i) == 'ㄴ' || q.get(i) == 'ㄷ' || q.get(i) == 'ㄹ'
||q.get(i) == 'ㅁ' ||q.get(i) == 'ㅂ' ||q.get(i) == 'ㅅ' ||q.get(i) == 'ㅇ' ||q.get(i) == 'ㅈ' ||q.get(i) == 'ㅊ' ||q.get(i)
== 'ㅋ' ||q.get(i) == 'ㅌ' ||q.get(i) == 'ㅍ' ||q.get(i) == 'ㅎ')&&q.get(i+1)=='!'&&q.get(i+2)=='@'
switch(q.get(i))
case 'ㄱ'
temp2 = temp2 + jong1(2);
case 'ㄴ'
temp2 = temp2 + jong1(5);
case 'ㄷ'
temp2 = temp2 + jong1(8);
case 'ㄹ'
temp2 = temp2 + jong1(9);
case 'ㅁ'
temp2 = temp2 + jong1(17);
case 'ㅂ'
temp2 = temp2 + jong1(18);
case 'ㅅ'
temp2 = temp2 + jong1(20);
case 'ㅇ'
temp2 = temp2 + jong1(22);
case 'ㅈ'
temp2 = temp2 + jong1(23);
case 'ㅊ'
temp2 = temp2 + jong1(24);
case 'ㅋ'
temp2 = temp2 + jong1(25);
case 'ㅌ'
temp2 = temp2 + jong1(26);
case 'ㅍ'
temp2 = temp2 + jong1(27);
case 'ㅎ'
temp2 = temp2 + jong1(28);
end
q.remove();
q.remove();
q.remove();
break;
elseif (q.get(i) == 'ㄱ' || q.get(i) == 'ㄴ' || q.get(i) == 'ㄷ' || q.get(i) == 'ㄹ'
||q.get(i) == 'ㅁ' ||q.get(i) == 'ㅂ' ||q.get(i) == 'ㅅ' ||q.get(i) == 'ㅇ' ||q.get(i) == 'ㅈ' ||q.get(i) == 'ㅊ' ||q.get(i)
== 'ㅋ' ||q.get(i) == 'ㅌ' ||q.get(i) == 'ㅍ' ||q.get(i) == 'ㅎ')&&(q.get(i+1) == 'ㅏ' ||q.get(i+1) == 'ㅑ' ||q.get(i+1)
== 'ㅓ' ||q.get(i+1) == 'ㅕ' ||q.get(i+1) == 'ㅗ' ||q.get(i+1) == 'ㅛ' ||q.get(i+1) == 'ㅜ' ||q.get(i+1) == 'ㅠ' ||q.get(i+1) == 'ㅡ' ||q.get(i+1) == 'ㅣ' ||q.get(i+1) == ' ')&&q.get(i+2)=='!'
switch(q.get(i))
case 'ㄱ'
temp2 = temp2 + (cho1(1)*588);
case 'ㄴ'

```

```

        temp2 = temp2 + (cho1(3)*588);
    case 'ㄷ'
        temp2 = temp2 + (cho1(4)*588);
    case 'ㄹ'
        temp2 = temp2 + (cho1(6)*588);
    case 'ㄱ'
        temp2 = temp2 + (cho1(7)*588);
    case 'ㅁ'
        temp2 = temp2 + (cho1(8)*588);
    case 'ㅇ'
        temp2 = temp2 + (cho1(10)*588);
    case 'ㅂ'
        temp2 = temp2 + (cho1(12)*588);
    case 'ㅅ'
        temp2 = temp2 + (cho1(13)*588);
    case 'ㅈ'
        temp2 = temp2 + (cho1(15)*588);
    case 'ㅊ'
        temp2 = temp2 + (cho1(16)*588);
    case 'ㅋ'
        temp2 = temp2 + (cho1(17)*588);
    case 'ㅌ'
        temp2 = temp2 + (cho1(18)*588);
    case 'ㅎ'
        temp2 = temp2 + (cho1(19)*588);
    end
    elseif (q.get(i) == 'ㅏ' || q.get(i) == 'ㅑ' || q.get(i) == 'ㅓ' || q.get(i) == 'ㅕ' || q.get(i) == 'ㅗ' || q.get(i) == 'ㅛ' || q.get(i) == 'ㅜ' || q.get(i) == 'ㅠ' || q.get(i) == 'ㅡ' || q.get(i) == 'ㅣ' || q.get(i) == 'ㅚ' || q.get(i) == 'ㅜ' || q.get(i) == 'ㅡ' || q.get(i) == 'ㅣ') && (q.get(i+1) == 'ㅏ' || q.get(i+1) == 'ㅑ' || q.get(i+1) == 'ㅓ' || q.get(i+1) == 'ㅕ' || q.get(i+1) == 'ㅗ' || q.get(i+1) == 'ㅛ' || q.get(i+1) == 'ㅜ' || q.get(i+1) == 'ㅠ' || q.get(i+1) == 'ㅡ' || q.get(i+1) == 'ㅣ') && (q.get(i+2) == 'ㅏ' || q.get(i+2) == 'ㅑ' || q.get(i+2) == 'ㅓ' || q.get(i+2) == 'ㅕ' || q.get(i+2) == 'ㅗ' || q.get(i+2) == 'ㅛ' || q.get(i+2) == 'ㅜ' || q.get(i+2) == 'ㅠ' || q.get(i+2) == 'ㅡ' || q.get(i+2) == 'ㅣ'))
    switch(q.get(i))
    case 'ㅏ'
        temp2 = temp2 + (jung1(1)*28);
    case 'ㅑ'
        temp2 = temp2 + (jung1(2)*28);
    case 'ㅓ'
        temp2 = temp2 + (jung1(3)*28);
    case 'ㅕ'
        temp2 = temp2 + (jung1(4)*28);
    case 'ㅗ'
        temp2 = temp2 + (jung1(5)*28);
    case 'ㅛ'
        temp2 = temp2 + (jung1(6)*28);
    case 'ㅜ'
        temp2 = temp2 + (jung1(7)*28);
    case 'ㅠ'
        temp2 = temp2 + (jung1(8)*28);
    case 'ㅡ'
        temp2 = temp2 + (jung1(9)*28);
    case 'ㅣ'
        temp2 = temp2 + (jung1(10)*28);
    end
end

```

[illegible]

```

        temp2 = temp2 + (jung1(19)*28);
    case '┘'
        temp2 = temp2 + (jung1(20)*28);
    case '┐'
        temp2 = temp2 + (jung1(21)*28);
    end
    q.remove();
    q.remove();
    break;
end

end
temp1(j) = temp1(j) + temp2;
temp2 = 0;
end

[m,n] = size(temp1);
for h=1:n
    temp1(h) = temp1(h)+44032;
end
%저장된 데이터 출력 및 tts 출력
set(b,'String', char(temp1),'position',[80 130 650 120])
tts(char(temp1))
%저장된 큐 삭제
for m=0:q.size()-1
    q.remove();
end
%전역변수 초기화
temp1 = [];
%번역 멈춤
stop(depthVid);
stop(colorVid);
stop(t);
end
%정확도 증가를 위해 5개의 값이 같으면 q에 저장
if countT == 5
    if accuracyR(1) == accuracyR(2) && accuracyR(2) == accuracyR(3) && accuracyR(3) ==
accuracyR(4) && accuracyR(4) == accuracyR(5)
        set(d,'String', char(accuracyR(2)), 'position',[420 440 320 120])
        % tts 결과 출력 및 q에 저장
        tts(char(accuracyR(2)))
        q.add(char(accuracyR(1)));
    end
    countT = 0;
end
countT = countT + 1;
end
end
end
end
end

```

% 궤적 함수 선언

```
function dispDepth2(obj, event)
```

```
    % 영상 출력
```

```
    trigger(depthVid);
```

```
    trigger(colorVid);
```

```
    [depthMap, ~, depthMetaData] = getdata(depthVid);
```

```
    [colorFrameData] = getdata(colorVid);
```

```
    idx = find(depthMetaData.IsSkeletonTracked);
```

```
    ax = subplot(2,2,1);
```

```
    set(ax, 'position', [0.09,0.43 0.41 0.55]);
```

```
    imshow(colorFrameData);
```

```
    % 영상처리
```

```
    % 스켈레톤 추적이 됐을 때
```

```
    if idx ~= 0
```

```
        % 척추 위치 기반 상반신 추적
```

```
        body = depthMetaData.JointDepthIndices(3, idx);
```

```
        % 사각형으로 상반신 크롭
```

```
        radius = 300;
```

```
        bodyBox = [body(1)-0.75*radius body(2)-0.3*radius 1.5*radius radius];
```

```
        rectangle('position', bodyBox, 'EdgeColor', [1 1 0]);
```

```
        bodyImage = imcrop(colorFrameData, bodyBox);
```

```
        % 크롭이 됐을 때
```

```
        if ~isempty(bodyImage)
```

```
            % timer 함수가 불릴 때마다 m증가
```

```
            m=m+1;
```

```
            % 이미지 리사이징 후 저장
```

```
            k(:,:,m) = uint8(imresize(bodyImage, [300, 450]));
```

```
            % 10프레임으로 3초 이후
```

```
            if(m==30)
```

```
                % 저장한 이미지를 비디오로 저장
```

```
                video = centerCrop(k, inputSize);
```

```
                % 비디오를 구글넷에 알맞은 데이터로 변환
```

```
                sequences{1} = activations(netCNN, video, layerName, 'OutputAs', 'columns');
```

```
                % 구글넷을 활용한 결과 예측
```

```
                YPred = classify(netLSTM, sequences);
```

```
                result = string(YPred);
```

```
                % tts로 결과 출력
```

```
                set(d, 'String', result, 'position', [420 440 320 120])
```

```
                tts(result)
```

```
                q.add(char(result));
```

```
                m=0;
```

```
            end
```

```
        else
```

```
            m=0;
```

```
        end
```

```
    end
```

```
end
```



```

% 각 기능에 대한 callback 함수 선언
function startCallback2(obj, event)
    start(depthVid);
    start(colorVid);
    start(t2);
end

function startCallback(obj, event)
    start(depthVid);
    start(colorVid);
    start(t);
end

function stopCallback(obj, event)
    %스탑 누르면 저장된 데이터 문장으로 연결
    if q.size()-1 == 2
        for y=0:q.size()-1
            if y==0
                temp3(y+1) = char(q.get(y)+"는");
            elseif y==1
                temp3(y+1) = char(q.get(y)+"를"+" ");
            elseif y==2
                temp3(y+1) = char(q.get(y)+" ");
            end
        end
    else
        for y=0:q.size()-1
            temp3(y+1) = char(q.get(y)+" ");
        end
    end
    %q데이터 삭제
    for m=0:q.size()-1
        q.remove();
    end
    newstr = join(temp3);
    %문장 출력
    set(b,'String', char(temp3),'position',[80 130 650 120])
    tts(char(newstr))
    %전역변수 초기화
    temp3 = [];
    newstr = "";
    %번역 멈춤
    stop(depthVid);
    stop(colorVid);
    stop(t2);

end

function speechCallback(obj, event)
    start(t3);
    stop(t3);
end

% 비디오 리사이징 함수 선언

```

```

function videoResized = centerCrop(video,inputSize)
% 비디오 사이즈 저장
sz = size(video);
% 비디오가 풍경일 때
if sz(1) < sz(2)
    idx = floor((sz(2) - sz(1))/2);
    video(:,1:(idx-1),,:) = [];
    video(:,(sz(1)+1):end,:,:) = [];
    % 비디오에 인물이 있을 때
elseif sz(2) < sz(1)
    idx = floor((sz(1) - sz(2))/2);
    video(1:(idx-1),,:,:) = [];
    video((sz(2)+1):end,:,:) = [];
end
% 비디오 사이즈 변환
videoResized = imresize(video,inputSize(1:2));
end

```

6. 실행 코드

```

% guiRun.m
% 수화 인식 프로그램 실행
% 지화 학습 결과 load
load net

% 꺾적 학습 결과 load
load netLSTM

% load한 net으로 수화 인식 프로그램 실행
translator(net, netLSTM)

```