



국민대학교
전자정보통신대학
컴퓨터공학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	<i>ToS (Touch on Screen)</i>
팀 명	<i>KoPI</i>
문서 제목	결과보고서

Version	1.3
Date	2020-JUN-09

팀원	정 형섭 (조장)
	심 유정
	이 규한
	유 성훈
	조 정근

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 수강 학생 중 프로젝트 “ToS” 를 수행하는 팀 “KoPI” 의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “KoPI” 의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


Filename	중간보고서-ToS.doc
원안작성자	정형섭, 이규한, 심유정, 유성훈, 조정근
수정작업자	정형섭, 이규한, 심유정, 유성훈, 조정근

수정날짜	대표수정 자	Revision	추가/수정 항 목	내 용
2020-06-08	조정근	1.0	최초 작성	
2020-06-09	이규한	1.1	수정	모바일 사용 메뉴얼 추가 및 세부사항 수정
2020-06-09	정형섭	1.2	추가	활용/개발기술 및 현실적 제한요소 추가
2020-06-09	전원	1.3	내용 수정	

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

목 차

1	개요	4
1.1	프로젝트 개요	4
1.2	추진 배경 및 필요성	4
2	개발 내용 및 결과물	5
2.1	목표	5
2.2	연구/개발 내용 및 결과물	6
2.2.1	연구/개발 내용	6
2.2.2	시스템 기능 요구사항	8
2.2.3	시스템 비 기능(품질) 요구사항	9
2.2.4	시스템 구조 및 설계도	10
2.2.5	활용/개발된 기술	13
2.2.6	현실적 제한 요소 및 그 해결 방안	15
2.2.7	결과물 목록	15
2.3	기대효과 및 활용방안	16
3	자기평가	17
3.1	최종 결과물	17
3.1.1	모바일 어플리케이션	17
3.1.2	PC 프로그램	18
3.1.3	서버	19
3.2	사용 가능성 여부 판단	19
4	참고 문헌	19
5	부록	20
5.1	사용자 매뉴얼(모바일 어플리케이션)	20
5.2	사용자 매뉴얼(PC 프로그램)	22
5.3	운영자 매뉴얼	24
5.4	배포 가이드	25
5.5	테스트 케이스	25

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

1 개요


1.1 프로젝트 개요

화면을 보여주는 모니터, 빔 프로젝터는 컴퓨터에 있어서 가장 기본적인 출력장치이다. 시각적인 출력장치에 매우 많은 종류가 생기고 나선 터치가 가능해지는 모니터, 터치를 인식하는 빔 프로젝터 등과 같이 출력장치에 입력장치를 추가하여 사용자와 상호작용을 하는 장치 역시 매우 많이 개발되며 상용화 되고 있다. 그러나 기존의 제품을 사용해 입력장치를 추가하려면 추가적인 기기를 구매하거나 새로운 기기를 구매해야 한다는 단점이 있다.

따라서 본 프로젝트는 빔 프로젝터, 모니터 등 PC와 연결된 스크린에서의 조작을 모바일 카메라를 통한 모션인식을 이용하여 실행하는 것을 목표로 한다. 대부분의 강의를 위한 공간은 빔 프로젝터나 큰 모니터를 사용한다. 하지만 발표를 하면서 동적인 자세 또는 특정 행동을 통해서 발표 화면 혹은 스크린을 제어하는 것은 어렵다. 이 프로젝트는 모바일 디바이스에서의 모션 인식 기능을 통해 추가적인 비용 소모 없이 화면을 터치하는 것과 같은 효과를 줄 것이다.

1.2 추진 배경 및 필요성

본 프로젝트를 추진하게 된 계기는 발표나 회의와 같은 상황에서 빔 프로젝터(모니터)의 직관적이고 편리한 입력을 사용하기 위해 시작하였다. 물론 터치를 지원하는 빔 프로젝터나 터치를 지원하는 모니터는 여러 종류가 있다. 예를 들어 가장 보편적인 터치를 지원하는 빔 프로젝터는 이노아이오 스마트빔 3, 소니의 엑스페리아 터치 등이다. 하지만 낮게는 30만원 정도부터 높게는 수 백만원 이상의 높은 가격대를 보여 실질적으로 많은 사람이 구매하여 사용하기엔 어렵다. 또한 터치 기능을 위하여 잘 사용하던 빔 프로젝터(모니터)를 제거하고 새로운 기기를 사용하는 것은 가격적으로 상당한 오버헤드가 발생한다. 새로운 기기를 사용하게 되면 기존에 사용하던 기기의 활용이 불가능하다는 단점도 존재한다. 그리하여 이 프로젝트가 필요하다고 생각하였다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

2 개발 내용 및 결과물

2.1 목표

1. 사용자의 행동을 인식하는 머신러닝 모델을 개발한다.
 - 1) 본 프로젝트에 맞는 데이터셋을 구성하기 위해 촬영한다.
 - 2) 촬영한 영상을 모션별로 분류한다.
 - 3) 데이터셋을 프로그램을 통해 머신러닝 학습에 용이하게 변환(텍스트 형태)한다.
 - 4) 변환된 데이터셋을 가지고 머신러닝 학습한다.
2. 모바일과 pc에 정보를 전달, 연결 상태를 관리하는 server를 개발한다.
 - 1) pc와 모바일을 연결하기 위해 server에서는 연결된client들의 상태와 Mapping table 정보를 갖는다.
 - 2) Mapping table정보를 가지고 모바일에서 전달하는 정보를 PC로 전송해준다.
 - 3) 연결된 사용자 정보, 사용자 별로 가지는 연결된 기기 정보를 저장하는 DB를 가진다.
 - 4) 정보의 연산과 데이터의 저장을 분리하여 클라우드 환경에서 요청에 따라 동적으로 서버를 관리한다.
3. 사용자의 모션을 인식하고 정보를 전달하는 모바일 client를 개발한다.
 - 1) 모바일의 카메라에서 손 좌표 값을 실시간으로 머신러닝 모델에게 보내준다.
 - 2) 모델은 좌표 값을 받아 알맞은 모션 값으로 리턴 해준다.
 - 3) 받은 모션 값과 모션 값이 나온 시점의 손의 좌표를 서버로 전송해준다.
 - 4) 로그인 및 회원가입 정보를 서버에 보내준다.
4. 서버로부터 전달받은 정보를 가지고 pc를 제어하기 위한 pc client를 개발한다.
 - 1) 서버로부터 모션의 정보와 손의 좌표를 입력 받으면 해당하는 작업을 수행하고 모션 정보를 출력한다
 - 2) 로그인 및 회원가입 정보를 서버에 보내준다.
 - 3) 서버에 인증번호 요청을 보내고 사용자의 사용에 발행되지 않고 명확한 UI를 가진다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

2.2 연구/개발 내용 및 결과물

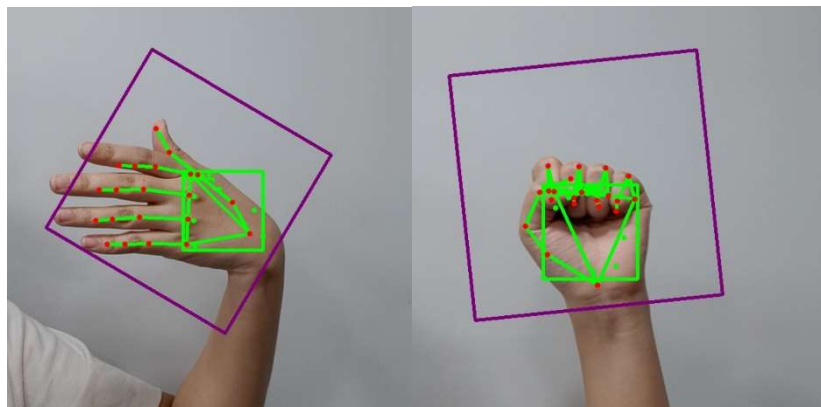
2.2.1 연구/개발 내용

모바일 Client

1. 사용자의 모션 인식

- MediaPipe

모바일, 클라우드, 웹 환경에서 가볍고 편하게 개발 및 사용할 수 있도록 Google에서 개발한 ML 솔루션이다. Mediapipe중에서도 Hand Tracking기술을 사용하여 사용자의 손을 인식하도록 하였다. Hand Tracking은 사람의 손에서 특징 점 21개를 검출한다.



- Keras(LSTM)

원하는 모션을 인식하는 모델의 개발을 위하여 학습 영상 4000개를 구성하였다. 모션 별(앞으로, 뒤로, 잠금, 클릭)별로 각각 1000개씩 제작하였다. 각 영상을 MediaPipe의 Hand Tracking application를 통과시켜 손의 움직임에 따른 좌표 값을 vectorize하여 텍스트 형태의 데이터셋으로 변경하였다.



```


harry@harry-G66-2PE: ~/mediapipe
Epoch 125/150: 8s 4ms/step - loss: 0.2473 - accuracy: 0.9222 - val_loss: 0.4895 - val_accuracy: 0.8589
Epoch 126/150: 8s 4ms/step - loss: 0.1888 - accuracy: 0.9274 - val_loss: 0.4419 - val_accuracy: 0.8768
Epoch 127/150: 8s 4ms/step - loss: 0.3189 - accuracy: 0.8983 - val_loss: 0.5891 - val_accuracy: 0.8363
Epoch 128/150: 8s 4ms/step - loss: 0.2738 - accuracy: 0.9114 - val_loss: 0.4358 - val_accuracy: 0.8664
Epoch 129/150: 8s 4ms/step - loss: 0.2111 - accuracy: 0.9199 - val_loss: 0.3477 - val_accuracy: 0.9083
Epoch 130/150: 8s 4ms/step - loss: 0.1758 - accuracy: 0.9344 - val_loss: 0.4184 - val_accuracy: 0.8871
Epoch 131/150: 8s 4ms/step - loss: 0.1522 - accuracy: 0.9452 - val_loss: 0.3241 - val_accuracy: 0.9078
Epoch 132/150: 8s 4ms/step - loss: 0.1104 - accuracy: 0.9653 - val_loss: 0.3617 - val_accuracy: 0.8993
Epoch 133/150: 8s 4ms/step - loss: 0.1189 - accuracy: 0.9578 - val_loss: 0.3615 - val_accuracy: 0.9097
Epoch 134/150: 8s 4ms/step - loss: 0.1127 - accuracy: 0.9592 - val_loss: 0.3305 - val_accuracy: 0.9031
Epoch 135/150: 8s 4ms/step - loss: 0.1534 - accuracy: 0.9470 - val_loss: 0.3608 - val_accuracy: 0.9040
Epoch 136/150: 8s 4ms/step - loss: 0.1134 - accuracy: 0.9602 - val_loss: 0.3758 - val_accuracy: 0.8956
Epoch 137/150: 8s 4ms/step - loss: 0.1582 - accuracy: 0.9433 - val_loss: 0.4103 - val_accuracy: 0.8989
Epoch 138/150: 8s 4ms/step - loss: 0.1521 - accuracy: 0.9485 - val_loss: 0.3645 - val_accuracy: 0.9022
Epoch 139/150: 8s 4ms/step - loss: 0.1157 - accuracy: 0.9602 - val_loss: 0.3878 - val_accuracy: 0.8918
Epoch 140/150: 8s 4ms/step - loss: 0.1325 - accuracy: 0.9545 - val_loss: 0.4446 - val_accuracy: 0.8815
Epoch 141/150: 8s 4ms/step - loss: 0.2526 - accuracy: 0.9133 - val_loss: 0.4677 - val_accuracy: 0.8739
Epoch 142/150: 8s 4ms/step - loss: 0.2318 - accuracy: 0.9157 - val_loss: 0.3825 - val_accuracy: 0.9083
Epoch 143/150: 8s 4ms/step - loss: 0.1347 - accuracy: 0.9536 - val_loss: 0.3577 - val_accuracy: 0.9069
Epoch 144/150: 8s 4ms/step - loss: 0.1858 - accuracy: 0.9686 - val_loss: 0.3216 - val_accuracy: 0.9163
Epoch 145/150: 8s 4ms/step - loss: 0.0961 - accuracy: 0.9686 - val_loss: 0.3548 - val_accuracy: 0.9144
Epoch 146/150: 8s 4ms/step - loss: 0.1152 - accuracy: 0.9616 - val_loss: 0.3511 - val_accuracy: 0.9022
Epoch 147/150: 8s 4ms/step - loss: 0.1875 - accuracy: 0.9639 - val_loss: 0.3389 - val_accuracy: 0.9135
Epoch 148/150: 8s 4ms/step - loss: 0.0856 - accuracy: 0.9780 - val_loss: 0.4001 - val_accuracy: 0.8984
Epoch 149/150: 8s 4ms/step - loss: 0.0928 - accuracy: 0.9785 - val_loss: 0.4376 - val_accuracy: 0.8899
Epoch 150/150: 8s 4ms/step - loss: 0.1476 - accuracy: 0.9583 - val_loss: 0.4239 - val_accuracy: 0.8899
Epoch 151/150: 8s 4ms/step - loss: 0.1799 - accuracy: 0.9311 - val_loss: 0.3803 - val_accuracy: 0.9135
test performance: accuracy=0.9135, loss=0.3683, val_accuracy=0.9135, val_loss=0.3683

```

2. 좌표 값 연산

- Image Translation

입력으로 들어오는 영상에서 Hough transform을 통해 추출된 직선으로 영상되고 있는 스크린의 영역을 찾는다. 해당하는 영역에서의 손의 상대적 위치를 계산하기 위하여 이미지 변환 기술 중 하나인 translation기술을 사용하였다. 스크린의 영역을 인식하고 손의 상대적 위치를 계산해준다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

Server

- 데이터 베이스

AWS의 EC2에 서버를 구현하고, RDS에 DB를 구현하여 DB에서 사용자의 정보와 각 사용자별로 연결된 기기의 목록을 저장하고, 로그인하여 프로그램을 사용하고자 하면, DB의 정보와 현재 연결되어 있는 기기의 목록에서 선택하여 연결해 준다.

2.2.2 시스템 기능 요구사항

ID	요구사항	결과
FR1	시스템은 스크린의 경계를 정확하게 인식한다.	달성
FR2	시스템은 손의 위치를 정확하게 인식한다.	달성
FR3	시스템은 손의 모션을 정확하게 인식한다.	달성
FR4	어플리케이션과 PC는 로그인 기능을 통해서 사용자를 인증한다	달성
FR5	어플리케이션과 PC는 비밀번호를 사용하여 사용자를 인증한다	달성
FR6	사용자는 회원가입으로 사용자의 데이터를 저장할 수 있다	달성
FR7	어플리케이션에서 PC로 모션의 정보와 좌표를 올바르게 전송한다	달성
FR8	시스템은 실시간으로 모션을 인식한다	미 달성

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

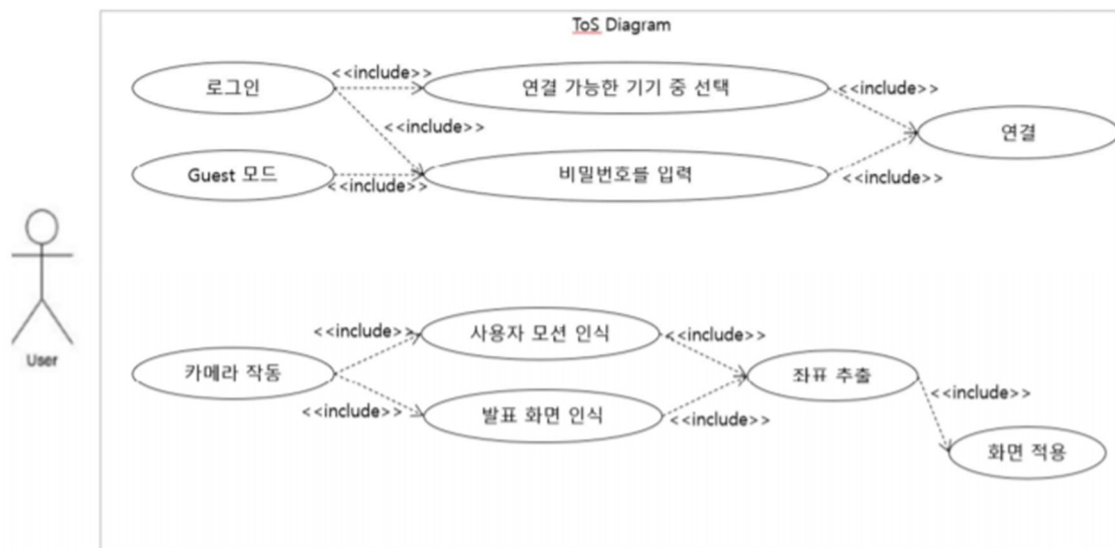
2.2.3 시스템 비 기능(품질) 요구사항

ID	업무	품질 속성	결과
NFR1	시스템은 스크린의 경계를 정확하게 인식해야 한다.	신뢰성 - 스크린의 일부가 가려져도 경계를 추출할 수 있다.	미 달성
NFR2	시스템은 손의 위치를 정확하게 인식한다.	신뢰성 - 손의 위치를 계산하여 화면에서 원하는 부분에서 작업을 수행한다.	달성
NFR3	시스템은 손의 모션을 정확하게 인식한다.	신뢰성 - 원하는 모션을 인식한다.	달성
NFR4	인식된 손의 모션과 좌표를 전송한다	성능 - 500ms 이내에 전송된 정보를 수신한다	달성
NFR5	어플리케이션과 PC클라이언트의 UI를 구성한다	사용성 - 누구나 직관적으로 사용 가능하다	달성
NFR6	사용자가 원하는 지점을 클릭한다	신뢰성 - 일반적인 빔 프로젝터를 사용한다고 가정하였을 때, 5cm이내의 좌표에서 작업을 수행한다.	미 달성
NFR7	다양한 환경에서 구동 가능하다	사용성 - 다양한 시간, 다양한 장소에서 사용 가능하다	달성
NFR8	다양한 기종의 안드로이드 기반 스마트폰에서 작동한다	이식성 - 다양한 해상도, 성능의 스마트폰에서 사용 가능하다.	미 달성
NFR9	다양한 환경의 PC에서 클라이언트를 수행한다.	이식성 - 다양한 해상도, 성능의 PC에서 사용 가능하다.	달성

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

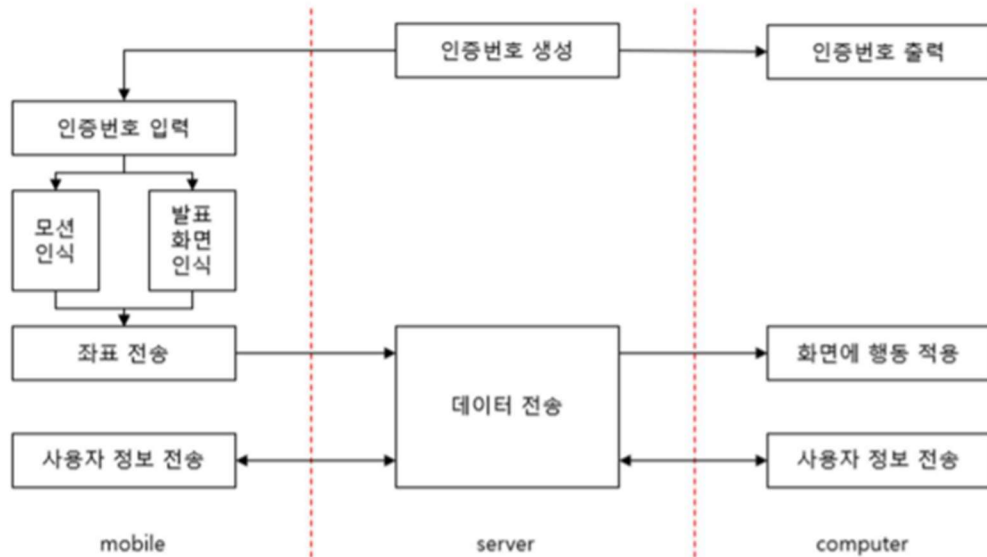
요구사항 ID	원인
NFR1	Probabilistic Hough Transform을 사용하였는데 직선을 검출하는 확률 값을 원하는 정도로 정하는데 어려움이 있다.
NFR6	스크린과의 거리를 고려하였을 때 스마트폰에 내장된 카메라의 성능이 충분하지 못해 정밀한 결과를 내지 못했다.
NFR8	수행하는 작업이 자원을 많이 사용하여 성능이 좋지 않은 스마트폰에서는 원하는 만큼의 성능을 보이지 않는다.

2.2.4 시스템 구조 및 설계도

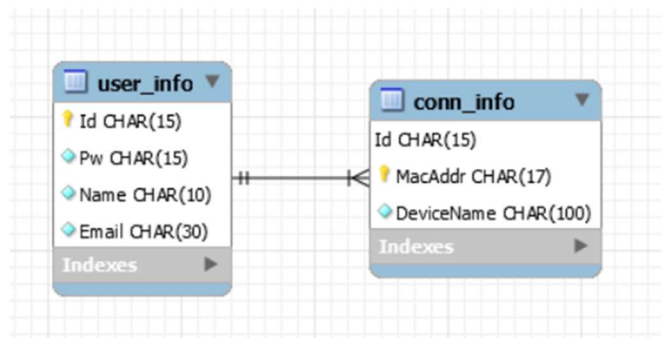


<Use-case Diagram>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09



<State Diagram>




<ER Diagram>

1. PC와 Application의 연결

1) 인증 번호

- 1-1) PC Client에서 서버에 인증번호를 요청한다.
- 1-2) 서버에서 인증 번호를 생성하고 PC로 전송한다.
- 1-3) PC에서 인증번호를 출력하고 Android Application에서 인증번호를 입력하여 서버로 전송한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

1-4) 서버에서 인증번호를 비교하여 PC와 Mobile을 연결한다.

2) 로그인

1-1) 회원가입을 통해 사용자의 정보를 서버에 저장한다. (PC, Mobile)

1-2) PC와 모바일에서 사용자의 정보로 로그인을 진행한다.

1-3) 연결된 목록을 서버에서 제공하고 제공된 목록 중 연결하고자 하는 Device를 선택하여 연결한다.

2. 스크린 인식

1) Open CV와 Hough Transform을 사용하여 스크린의 경계점을 추출한다.

2) 경계점을 이용해 직선을 찾고 스크린의 네 꼭지점을 찾는다.

3. 사용자의 손 인식

1) Google에서 개발한 MediaPipe의 Hand-Tracking을 사용하여 사용자의 손을 인식한다.

2) 학습시킨 모델을 이용하여 사용자의 모션을 인식한다.

3) 카메라로 들어온 입력 영상에 따른 손의 좌표와 모션의 정보를 출력한다.

4. 좌표 계산

1) 모션을 인식하여 나온 결과와 스크린의 좌표를 이용하여 정면에서 보았을 때 스크린에
서의 손의 상대적 위치를 계산한다.

5. 좌표 전송

1) Mobile에서 모션의 정보와 계산된 좌표의 값을 PC로 전송한다.

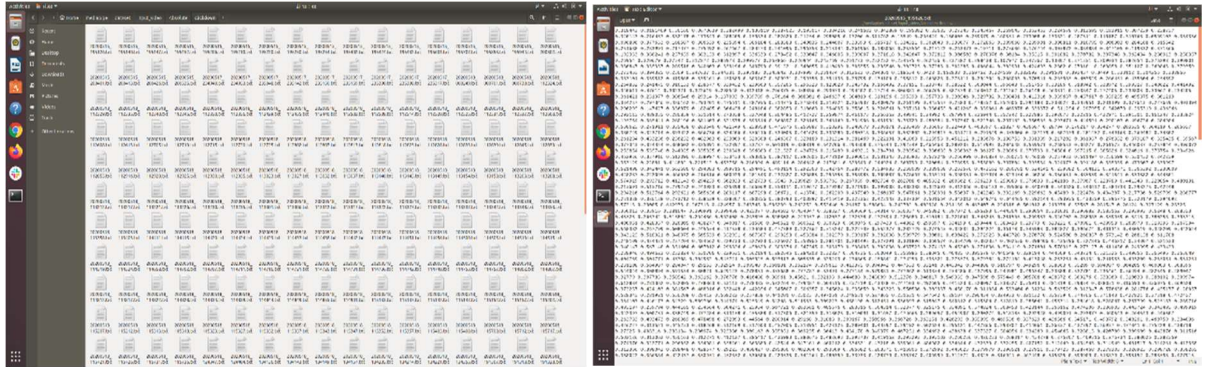
2) 입력된 모션의 정보와 좌표에 따르는 행동을 수행한다

 <div> <p>국민대학교</p> <p>컴퓨터공학부</p> <p>캡스톤 디자인 I</p> </div>	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

2.2.5 활용/개발된 기술

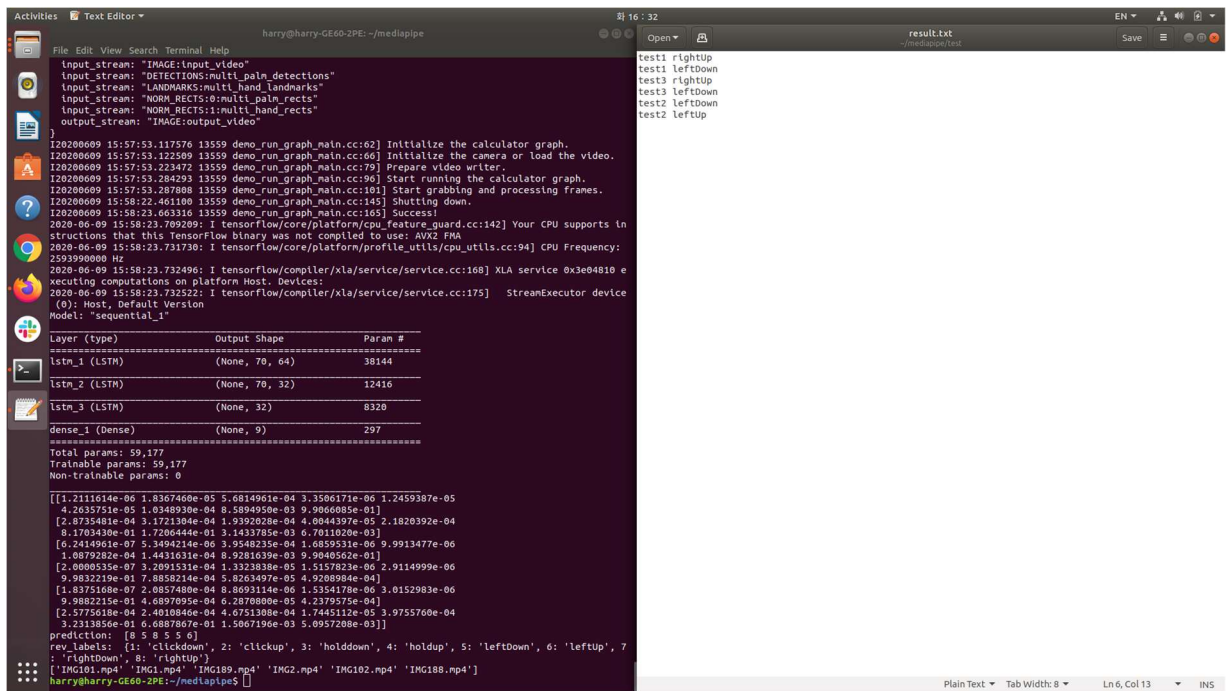
1. Mediapipe를 통한 고유 데이터셋 구성

- 영상을 학습에 용이한 텍스트 파일로 변환



2. 확보된 데이터셋을 통해 keras model 학습

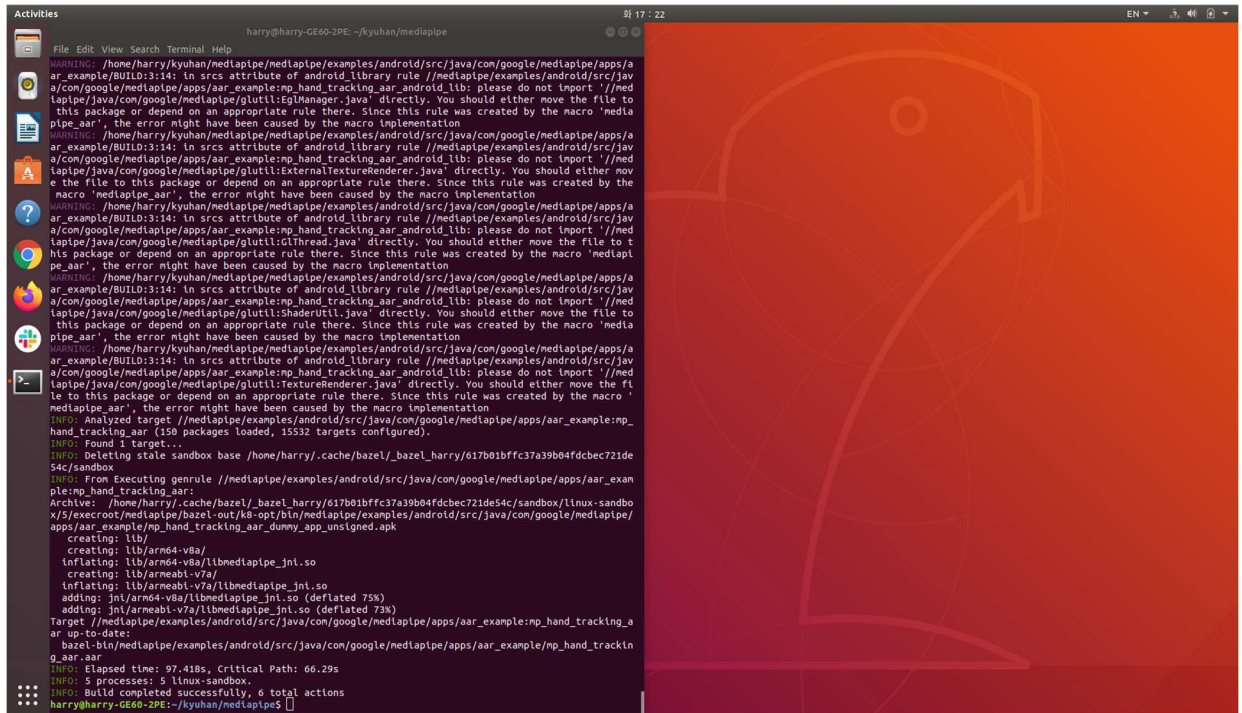
- 학습된 모델은 손의 좌표 값을 받으면 어떤 모션인지 알려준다.



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

3. mediapipe를 aar파일로 변환하여 안드로이드에 적용

- TFLite를 적용하기 용이하게 aar파일을 적용한 프로젝트 생성한다.



```

WARNING: /home/harry/kyuhan/mediapipe/mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example/BUILD:3:14: in srcs attribute of android.library rule //mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example:mp_hand_tracking_aar_android_lib: please do not import '//mediapipe/java/com/google/mediapipe/glutil:GluManager.java' directly. You should either move the file to this package or depend on an appropriate rule there. Since this rule was created by the macro 'mediapipe_aar', the error might have been caused by the macro implementation
WARNING: /home/harry/kyuhan/mediapipe/mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example/BUILD:3:14: in srcs attribute of android.library rule //mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example:mp_hand_tracking_aar_android_lib: please do not import '//mediapipe/java/com/google/mediapipe/glutil:GluManager.java' directly. You should either move the file to this package or depend on an appropriate rule there. Since this rule was created by the macro 'mediapipe_aar', the error might have been caused by the macro implementation
WARNING: /home/harry/kyuhan/mediapipe/mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example/BUILD:3:14: in srcs attribute of android.library rule //mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example:mp_hand_tracking_aar_android_lib: please do not import '//mediapipe/java/com/google/mediapipe/glutil:GluThread.java' directly. You should either move the file to this package or depend on an appropriate rule there. Since this rule was created by the macro 'mediapipe_aar', the error might have been caused by the macro implementation
WARNING: /home/harry/kyuhan/mediapipe/mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example/BUILD:3:14: in srcs attribute of android.library rule //mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example:mp_hand_tracking_aar_android_lib: please do not import '//mediapipe/java/com/google/mediapipe/glutil:ShaderUtil.java' directly. You should either move the file to this package or depend on an appropriate rule there. Since this rule was created by the macro 'mediapipe_aar', the error might have been caused by the macro implementation
INFO: Analyzed target //mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example:mp_hand_tracking_aar (150 packages loaded, 15532 targets configured).
INFO: Found 1 target...
INFO: Deleting state sandbox base /home/harry/.cache/bazel/_bazel_harry/617b01bffc37a39b04fdbc721de54c/sandbox
INFO: From Executing genrule //mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example:mp_hand_tracking_aar:
Archive: /home/harry/.cache/bazel/_bazel_harry/617b01bffc37a39b04fdbc721de54c/sandbox/linux-sandbox-5/execroot/mediapipe/bazel-out/k8-opt/bin/mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example/np_hand_tracking_aar_dummy_app_unsigned.apk
  creating: lib/
  creating: lib/arm64-v8a/
  inflating: lib/arm64-v8a/libmediapipe_jni.so
  creating: lib/arm64-v7a/
  inflating: lib/arm64-v7a/libmediapipe_jni.so
  adding: jni/arm64-v8a/libmediapipe_jni.so (deflated 75%)
  adding: jni/arm64-v7a/libmediapipe_jni.so (deflated 73%)
Target //mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example:mp_hand_tracking_aar up-to-date:
  bazel-bin/mediapipe/examples/android/src/java/com/google/mediapipe/apps/aar_example/np_hand_tracking_aar.aar
INFO: Elapsed time: 97.418s, Critical Path: 66.29s
INFO: 5 processes: 5 linux-sandbox.
INFO: Build completed successfully, 6 total actions
harry@harry-GE60-ZPE: ~/kyuhan/mediapipe

```

4. 안드로이드 프로젝트에 keras model을 import하여 적용한다

- 안드로이드에 학습모델을 import하고 실시간으로 영상을 처리한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

2.2.6 현실적 제한 요소 및 그 해결 방안

- 스마트폰의 연산 속도에 따라 다른 성능을 낸다.
 - 멀티 스레드나 멀티 프로세싱 기술을 사용한다.
- 카메라의 화질에 따라 인식하는 성능이 달라진다.
 - 인공지능을 사용하여 화질을 보정해준다.
- 데이터셋 구성이 매우 힘들다.
 - 외주를 부탁한다.
- 학습된 모델이 원하는 성능을 보이기 힘들다.
 - 데이터셋을 잘 가공해야 한다.
 - 질이 좋은 데이터를 학습시킨다.
 - 여러 모델에 학습시켜본다.

2.2.7 결과물 목록

대분류	소분류	기능	형식	비고
Algorithm	터치 위치 연산	스크린 검출, 손의 위치 파악	함수	
	Image Translate	원근법을 고려한 스크린 변환	함수	
	모션인식 딥러닝 모델	모션 인식 및 모션 분류	함수	
Mobile Application	서버와 통신	인증번호 전송 및 좌표 전송, 연결 유지	함수	
	로그인	사용자 로그인	함수	
PC Client	서버와 통신	좌표 값 수신, 연결 유지	함수	
	로그인	사용자 로그인	함수	

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

	화면 제어	입력 받은 정보를 통해 PC 제어	함수	
Data Base	사용자 정보 저장	사용자의 정보 및 연결 정보 저장	DB	
Server	AWS 서버	어플리케이션과 PC사이의 통신	함수	AWS

2.3 기대효과 및 활용방안

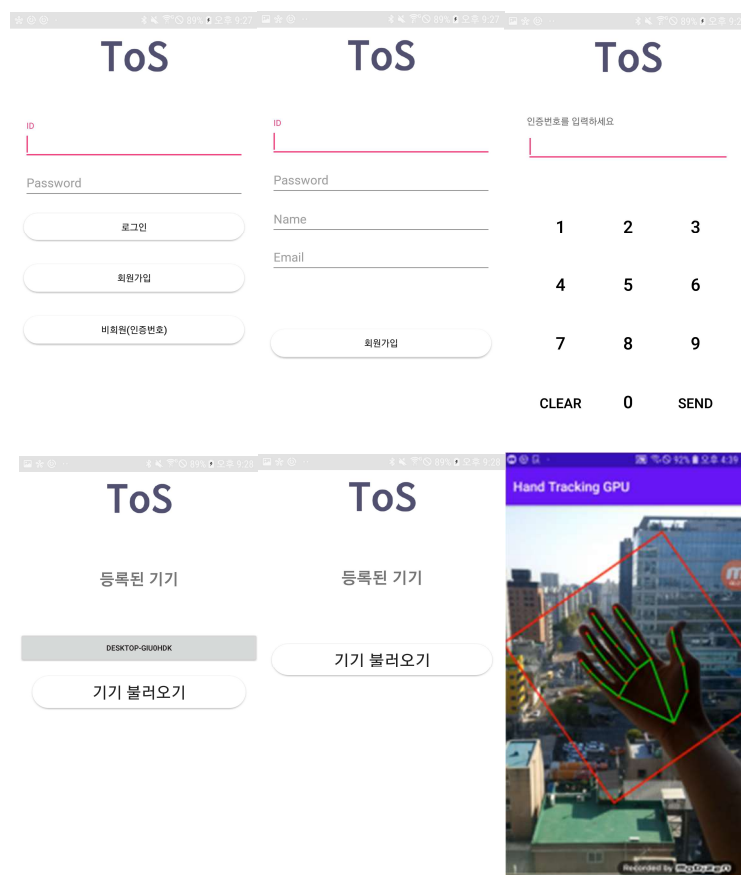
1. 빔 프로젝터를 사용함에 있어 터치를 하는 것과 같은 환상을 제공한다
일반적으로 빔 프로젝터는 발표 등에 사용된다. 원하는 지점을 클릭 가능하므로 ppt를 다채롭게 만들 수 있다. 예를 들면 ppt에 앞뒤 버튼을 추가하여 자유롭게 앞뒤로 이동할 수 있다. 혹은 행사 진행에서 버튼에 번호를 부여하여 상품 추천에도 사용될 수 있다.
2. 기존의 빔 프로젝터에서 추가적인 비용 소모가 없다.
시중에 나온 전자 칠판 혹은 터치 빔 프로젝터 등이 다수 존재한다. 하지만 여전히 회사, 공공기관 학교 등 대부분의 기관에는 일반 빔 프로젝터, 스크린이 설치되어 있다. 본 프로젝트를 통해 기존의 스크린, 빔 프로젝터의 환경에서도 터치를 하는 것과 같은 기능을 사용할 수 있다
3. 안드로이드 기기만 있으면 누구나 사용이 가능하다.
오늘날 남녀노소 할 것 없이 모두가 스마트폰을 갖고 있다. 어플리케이션을 다운로드만 받는다면 사용이 가능하기 때문에 매우 높은 접근성을 가지고 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

3 자기평가

3.1 최종 결과물

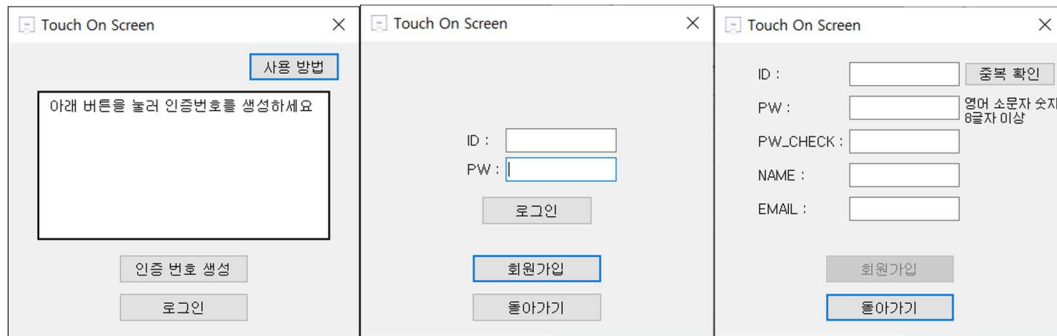
3.1.1 모바일 어플리케이션



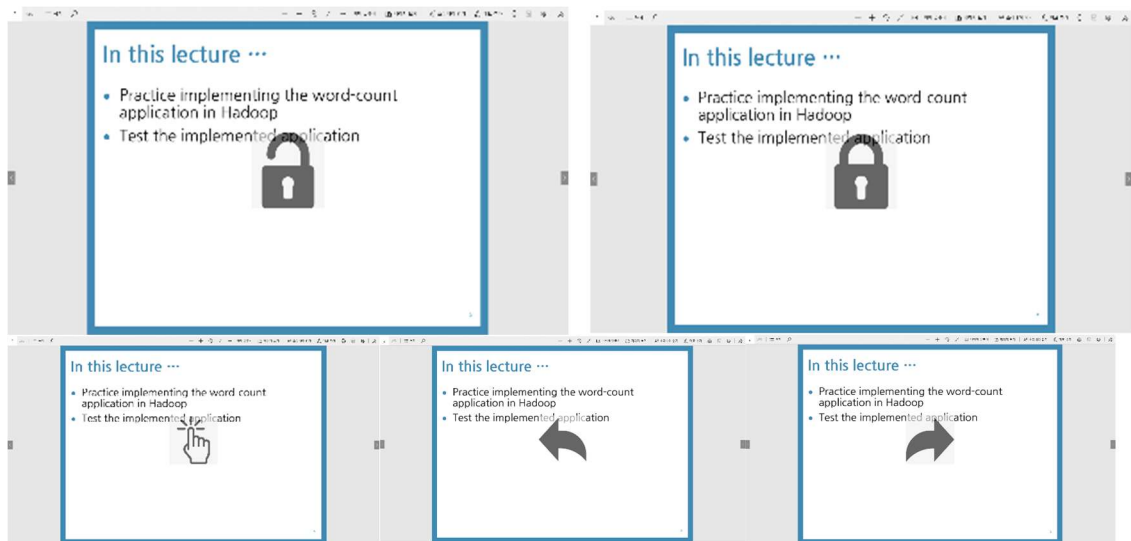
모바일 어플리케이션에서는 보는 것과 같이 UI를 통하여 회원가입, 로그인을 하고, 로그인을 한 후에는 등록된 기기 목록에서 연결하고자 하는 기기를 선택하여 특정 PC와 연결을 하거나 임시 인증번호를 입력하여 해당하는 PC와 연결한다. PC와 연결이 된 후에는 마지막 그림처럼 손을 인식하고 모션을 인식하여 Server를 통해 PC로 모션의 정보 및 좌표 값을 전달해주게 된다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

3.1.2 PC 프로그램



PC 프로그램은 인증번호를 생성하여 모바일과 연동하거나, 회원가입 후 로그인을 진행하여 DB에 정보를 저장하고 모바일로부터 연결이 되면 PC의 제어를 시작한다.



특정 모션에 대한 정보가 들어오면 해당하는 모션의 정보를 화면에 출력해 어떤 정보가 들어왔고, 행동을 하는지 보여준다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

3.1.3 서버

```
{
('223.62.8.174', 57828) connected.
}
flag: login
login {"id":"test","pw":"test"}
login end
('223.62.8.174', 57840) connected.
}
flag: login
login {"id":"test","pw":"test"}
login end
('223.62.8.174', 57842) connected.
}
('223.62.8.174', 57844) connected.
}
flag: signup
signup {} {"id":"project","pw":"project04","name":"유성훈","email":"shyoo@mail.com"}
{"id": 'project', 'pw': 'project04', 'name': '유성훈', 'email': 'shyoo@mail.com'}
signup end
('223.62.8.174', 57846) connected.
}
flag: login
login {"id":"project","pw":"project04"}
login end
('223.62.8.174', 57848) connected.
}
flag: login
login {"id":"project","pw":"project04"}
login end
('223.62.8.174', 57850) connected.
}
```


서버에서는 어플리케이션과 PC로부터 회원가입, 로그인 정보 등의 요청을 받아 데이터베이스의 데이터를 저장 및 수정한다.

3.2 사용 가능성 여부 판단

현재는 모바일 위에서 실시간 영상을 AI에게 보내어 연산을 시행하는 데에 있어 한계가 있다. 이후 더 좋은 프로세서를 가진 모바일이 개발이 된다면 본 프로젝트는 충분히 실현이 가능할 뿐만 아니라 더 많은 모션과 기능을 넣을 수 있을 것으로 보인다.

4 참고 문헌

번호	종류	제목	출처	발행 년도	저자
1	웹사이트	Docs - Mediapipe	https://mediapipe.readthedocs.io/en/latest/		
2	웹사이트	Keras Documentation	https://keras.io/ko/		

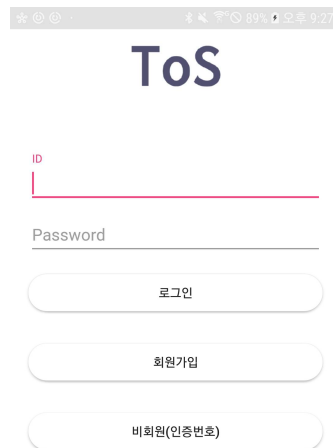
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

3	GitHub	Mediapipe	https://github.com/google/mediapipe		
4	GitHub	Sign-language-recognition-with-RNN	https://github.com/rabBit64/Sign-language-recognition-with-RNN-and-Mediapipe		


5 부록

5.1 사용자 매뉴얼(모바일 어플리케이션)

1. apk파일을 통해 어플리케이션을 설치한다.
2. 최초 시작 시 회원 가입 및 비회원을 선택한다. (회원가입 시 매번 인증번호를 보낼 필요 없이 연결된 기기 목록을 불러와서 PC와 연결이 가능하다.)



- 3-1. 회원 가입을 원한다면 필요한 항목을 입력 후 회원 가입 버튼을 클릭한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

📶 🔋 🔒 89% 🕒 오후 9:27

ToS

ID

Password

Name

Email

회원가입

3-2. 로그인 화면을 통해 로그인한다.

3-3. 로그인 시 기기 불러오기를 통해 기기 등록이 가능하다.

📶 🔋 🔒 89% 🕒 오후 9:28

ToS

등록된 기기

기기 불러오기

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

4. PC 화면에 나타난 인증번호를 입력해 PC와 연결된다.
(로그인 이용 시 기기가 등록된다.)

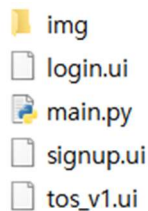


5. 카메라를 스크린이 보이는 곳에 위치하고, 다양한 제스처를 통해 프레젠테이션을 조작한다.

5.2 사용자 매뉴얼(PC 프로그램)

\$ pip install pyautogui pyqt5 win10toast

명령어를 사용하여 프로그램 실행에 필요한 모듈을 다운로드 받는다.

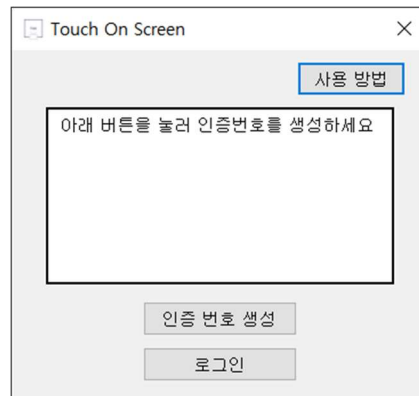


위 그림처럼 프로그램 실행에 필요한 파일들을 다운받는다.

\$ python main.py

Main 프로그램을 실행한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09



Touch On Screen

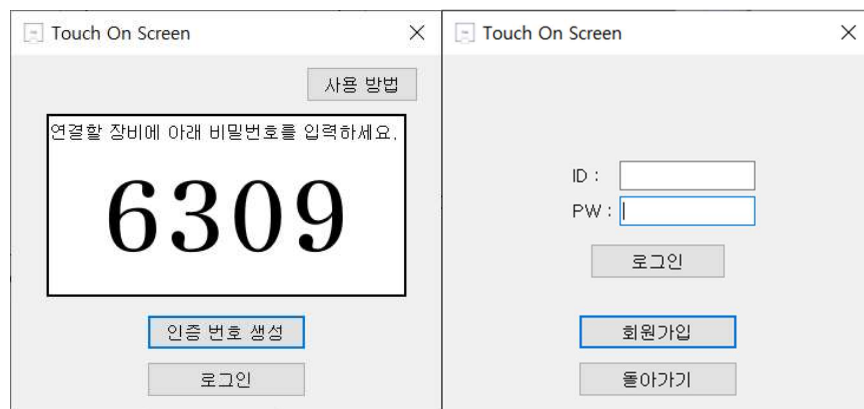
사용 방법

아래 버튼을 눌러 인증번호를 생성하세요

인증 번호 생성

로그인

메인 페이지가 열리면 로그인 버튼을 눌러 로그인 화면으로 넘어가거나 인증번호 생성 버튼을 눌러 임시 번호를 생성한다.



Touch On Screen

사용 방법

연결할 장비에 아래 비밀번호를 입력하세요.

6309

인증 번호 생성

로그인

Touch On Screen

ID :

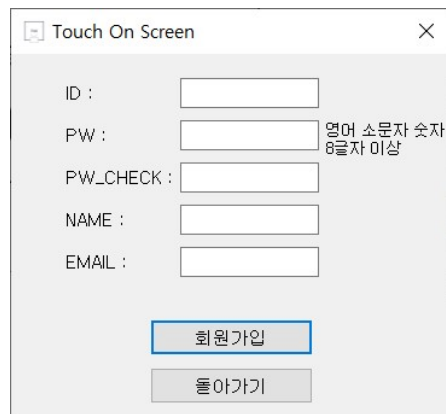
PW :

로그인

회원가입

돌아가기

왼쪽 사진은 임시 번호를 생성, 오른쪽 사진은 로그인 화면으로 넘어간 모습이다.



Touch On Screen

ID :

PW : 영어 소문자 숫자 8글자 이상


PW_CHECK :

NAME :

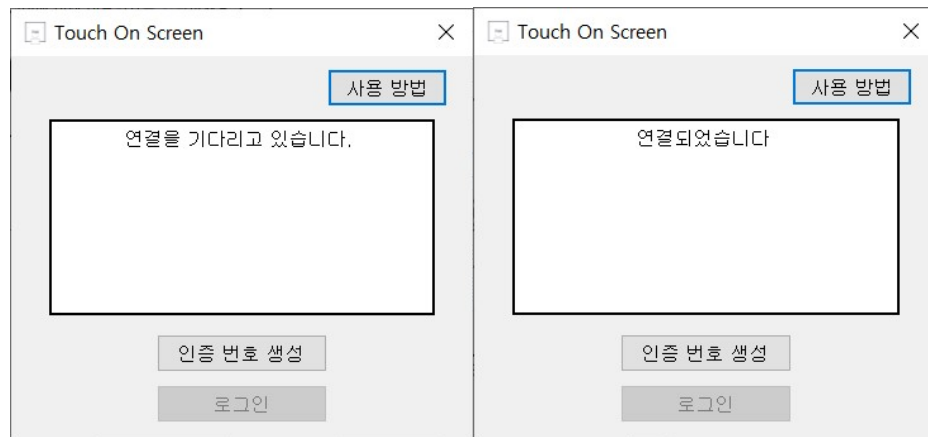
EMAIL :

회원가입

돌아가기

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

해당하는 화면은 로그인 화면에서 회원가입 화면으로 들어간 모습이다. Input box에 맞게 데이터를 입력하고 회원가입 버튼을 누르면 서버로 데이터가 전송되고, 로그인에 사용할 수 있게 된다.



로그인 하면 연결을 대기하고, 모바일에서 연결되면 연결되었다고 메시지가 나오고 모바일로부터 모션의 정보와 좌표를 받는다. 닫기 버튼을 눌러 프로그램을 종료할 시 위와 같은 toast message를 보이며, system tray Icon이 생겨 해당 아이콘으로 다시 프로그램을 복구할 수 있다. 종료하려고 할 때 역시 tray icon에서 "Exit"버튼을 눌러 종료할 수 있다.

5.3 운영자 매뉴얼

설치 가이드, 설정 및 운영 가이드 등

서버는 aws에서 구현되었다. Server 폴더에 main.py 파일을 받아 실행한다.

\$ pip install pymysql

프로그램을 실행하기 위해서 위와 같은 module을 설치하여야 한다.

\$ python3 main.py

위 명령어를 입력하여 프로그램을 실행한다. 해당하는 서버의 ip주소를 PC 프로그램과 모바일 어플리케이션에 주어 해당 서버로 접근하도록 할 수 있다.

자신의 데이터베이스를 사용하고 싶다면 main.py파일에서 데이터베이스 주소를 바꾸어 주면 된다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09


5.4 배포 가이드

다음 주소의 Tos.apk 파일을 통해 설치할 수 있습니다. (안드로이드 기종만 사용 가능하다)

https://github.com/kookmin-sw/capstone-2020-9/tree/master/src/Motion_APK

5.5 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
Algorithm	터치 위치 연산	스크린 검출, 손의 위치 파악	스크린이 나오는 영상을 넣었을 때 스크린이 검출되었음을 알리는 스크린 둘레의 사각형이 나타나는지 확인한다. 사용자의 손이 움직이는 영상을 넣었을 때 손의 움직임에 따라 좌표점이 찍히는지 확인한다.	스크린 둘레에 빨간 사각형이 나타난다. 손바닥과 관절 부분마다 좌표점이 찍힌다.	성공
	Image Translate	원근법을 고려한 스크린 변환	스크린을 정면에서 보다가 상하 좌우로 움직이는 영상을 넣었을 때 스크린 영역 검출이 유지되는지 확인한다.	변함없이 스크린 둘레에 빨간 사각형이 나타난다.	실패
	모션인식 딥러닝 모델	모션 인식 및 모션 분류	영상을 넣었을 때 모델이 올바른 output을 출력하였는지 확인한다.	올바른 output이 text파일로 저장된다.	성공
Mobile Application	서버와 통신	인증번호 전송 및 좌표 전송, 연결 유지	인증번호를 서버에 전송하고 pc에서 서버로부터 'connected' 메시지를 수신하는지 확인한다.	모바일, pc에서 서버와 연결된다.	성공

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-JUN-09

			좌표 전송 시 서버에 올바른 좌표가 오는지 확인한다.		
	로그인	사용자 로그인	로그인 버튼 클릭 시 연결된 기기 불러오는 화면으로 넘어가는지 확인한다.	사용자의 기기 목록 페이지가 보인다.	성공
	서버와 통신	좌표 값 수신, 연결 유지	서버에 client에서 좌표를 보내 보고 지속적으로 유지되는지 로그를 확인한다.	좌표 값을 수신하고 컴퓨터로 보내준다.	성공
PC Client	로그인	사용자 로그인	모바일 어플리케이션에서 회원, 비회원 로그인을 선택 후 회원이면 ID, Password를 입력한 후 그 정보를 DB의 정보와 비교하여 맞는지 틀리는지를 올바르게 판별하는지 확인한다.	DB에 회원정보가 있는 사람일 경우 비교 후 로그인 성공메시지를 준다. 없거나 틀릴 경우 로그인 실패 메시지를 준다.	성공
	화면 제어	입력 받은 정보를 통해 PC 제어	server로부터 받은 데이터를 가지고 화면을 제어해본다.	server에서 받은 데이터를 가지고 화면을 제어한다.	성공
Data Base	사용자 정보 저장	사용자의 정보 및 연결 정보 저장	임의로 사용자를 회원가입 시키고 DB에 등록되는지 확인한다.	사용자가 회원가입을 하거나 자신의 기기를 등록할 시 DB에 저장된다.	성공
Server	AWS 서버	어플리케이션과 PC사이의 통신	Server와 Client간에 통신이 되는지 지속적으로 로그를 찍어본다.	Server와 Client(pc, mobile)가 성공적으로 데이터를 송수신한다.	성공