



국민대학교  
전자정보통신대학  
컴퓨터공학부


# 캡스톤 디자인 I

## 종합설계 프로젝트

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

Version	1.3
Date	2020-APR-23

팀원	정 형섭 (조장)
	심 유정
	유 성훈
	이 규한
	조 정근

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22


#### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "ToS(Touch on Screen)"를 수행하는 팀 "KoPI"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "KoPI"의 팀원들의 서면 허락 없이 사용되거나, 재가공 될 수 없습니다.

## 문서 정보 / 수정 내역


<b>Filename</b>	수행 계획서-ToS.doc
<b>원안작성자</b>	정형섭, 심유정, 유성훈, 이규한, 조정근
<b>수정작업자</b>	정형섭, 심유정, 유성훈, 이규한, 조정근

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-04-06	이규한	1.0	최초 작성	초안 작성
2020-04-10	조정근	1.1	내용 수정	다이어그램 추가 및 전반적인 문장 수정
2020-04-15	전원	1.2	내용 수정	전반적 내용 수정
2020-04-22	전원	1.3	내용 수정	전반적 내용 수정

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

## 목 차

1	개요 .....	4
1.1	프로젝트 개요 .....	4
1.2	추진 배경 및 필요성 .....	4
1.2.1	기술 시장의 현황 .....	5
1.2.2	기술 발전 현황 .....	6
1.2.3	기 개발된 시스템 현황 .....	7
2	개발 목표 및 내용 .....	8
2.1	목표 .....	8
2.2	연구/개발 내용 .....	8
2.3	개발 결과 .....	11
2.3.1	시스템 기능 요구사항 .....	11
2.3.2	시스템 비기능(품질) 요구사항 .....	12
2.3.3	시스템 구조 .....	13
2.3.4	결과물 목록 및 상세 사양 .....	14
2.4	기대효과 및 활용방안 .....	15
3	배경 기술 .....	15
3.1	기술적 요구사항 .....	15
3.1.1	개발환경 .....	15
3.1.2	프로젝트 결과물 환경 .....	16
3.2	현실적 제한 요소 및 그 해결 방안 .....	16
3.2.1	하드웨어 .....	16
3.2.2	소프트웨어 .....	17
3.2.3	기타 .....	17
4	프로젝트 팀 구성 및 역할 분담 .....	17
5	프로젝트 비용 .....	18
6	개발 일정 및 자원 관리 .....	19
6.1	개발 일정 .....	19
6.2	일정별 주요 산출물 .....	20
6.3	인력자원 투입계획 .....	21
6.4	비 인적자원 투입계획 .....	21
7	참고 문헌 .....	22

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

# 1 개요


## 1.1 프로젝트 개요

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

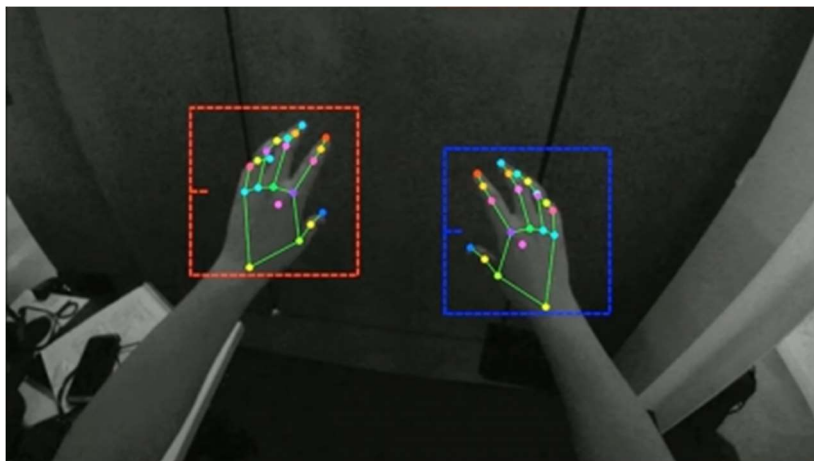
따라서 본 프로젝트는 빔 프로젝터, 모니터 등 PC와 연결된 스크린에서의 조작을 모바일 카메라에서의 모션인식을 통해 가능하게 하는 것을 목표로 한다. 대부분의 강의를 위한 공간은 빔 프로젝터나 큰 모니터를 사용한다. 하지만 발표를 하면서 동적인 자세 또는 특정 행동을 통해서 발표 화면 혹은 스크린을 제어하는 것은 어렵다. 이 프로젝트는 모바일 디바이스에서의 모션 인식 기능을 통해 추가적인 비용 소모 없이 화면을 터치하는 것과 같은 효과를 줄 것이다.

## 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. 가장 보편적인 터치를 지원하는 빔 프로젝터는 이노아이오 스마트빔 3, 소니의 엑스페리아 터치 등이 있다. 하지만 낮게는 30만원 정도부터 높게는 수 백만원 이상의 높은 가격대를 보이고 있다. 하지만 터치 기능을 위하여 잘 사용하던 빔 프로젝터(모니터)를 제거하고 새로운 기기를 사용하는 것은 가격적으로 상당한 오버헤드가 발생한다. 또한 새로운 기기를 사용하게 되면 기존에 사용하던 기기의 활용이 불가능하다는 단점이 존재한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22


### 1.2.1 기술 시장의 현황



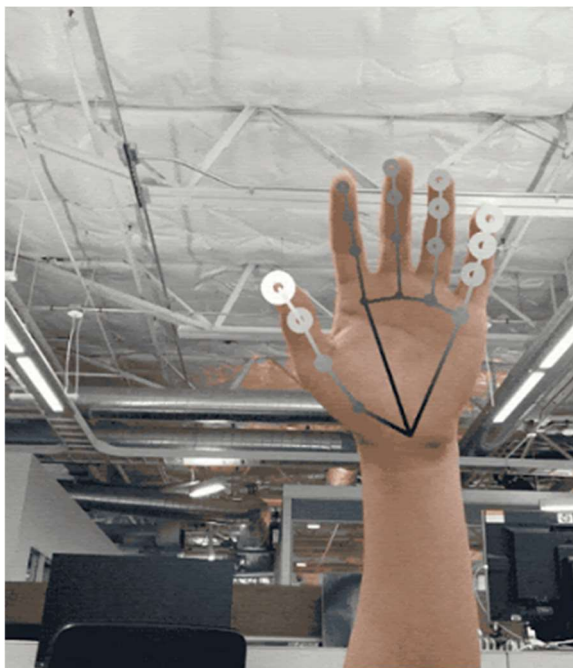
최근 오쿨러스에서 자사의 무선 헤드 마운드 디스플레이에 핸드 트래킹을 시범 도입하였다. 별도의 부착장비 없이 vr에 내장된 카메라를 이용하여 손의 인식이 가능하다. 인공지능 딥러닝을 활용하였으며 과도한 GPU 사용 없이도 원활한 핸드 트래킹을 구현하는 최적화 작업의 난이도를 낮추는데 성공했다.



소니의 엑스페리아 터치는 내장된 카메라와 적외선 센서가 초당 60프레임의 속도로 사물을 인식하며, 영상위의 물리적인 변화를 실시간으로 감지해 반응한다. 반응 속도와 오차범위는 간단한 게임이 가능한 반응 속도 정도와 무난한 정밀도를 보여준다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22


### 1.2.2 기술 발전 현황

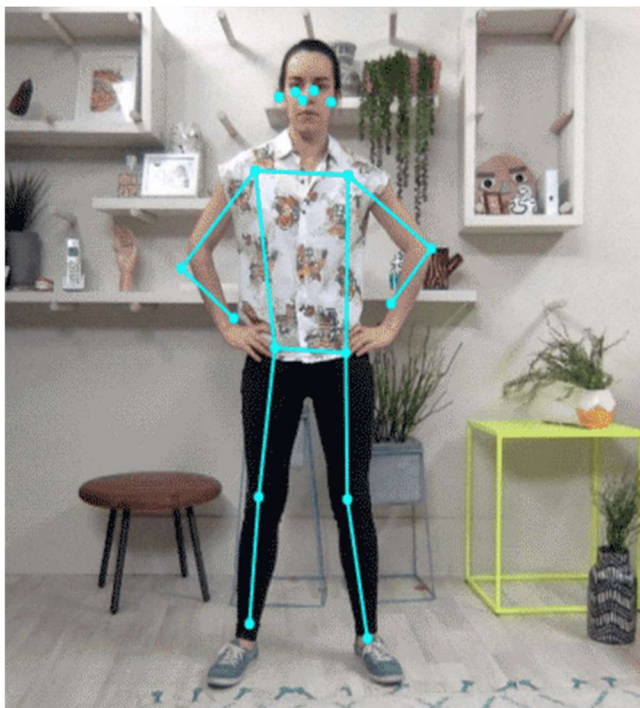


<https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html>

**Google MediaPipe:** 머신 러닝을 사용하여 이미지의 각 프레임마다 21개의 key point 점을 추론함으로써 높은 정확도를 가지고 손과 손가락을 추적한다. 한 프레임에서 여러 손을 인식하는 모듈 또한 존재하여 좀더 정확도를 높일 수 있다. 우리의 프로젝트처럼 거리가 있는 상황에서도 손의 인식을 정확하게 하는 모습을 볼 수 있다.

**TensorFlow pose estimation:** 주요 신체 관절의 위치를 추정하여 이미지나 비디오에서 사람의 자세를 추정한다. single pose, multi pose를 선택할 수 있으며 17개의 key point를 사용하여 사람의 자세를 추정한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22




[https://www.tensorflow.org/lite/models/pose\\_estimation/overview](https://www.tensorflow.org/lite/models/pose_estimation/overview)

**google teachable machine:** 인식하고자 하는 이미지 혹은 사운드를 입력으로 주어 웹상으로 모델을 학습시키고 학습된 모델을 다운받아 사용할 수 있다. 학습된 모델은 여러 이미지의 분류, 모션의 인식, 음악의 분류와 같은 기능들을 수행할 수 있다.

### 1.2.3 기 개발된 시스템 현황

현재 터치 가능한 빔 프로젝터는 저렴한 이노아이의 스마트빔3부터 100만원 이상의 높은 가격대를 형성하는 소니의 엑스페리아 터치까지 다양하게 존재한다. 하지만 저렴한 이노아이 스마트빔3는 작은 면적 투사에 적합하다. 또한 현재 개발된 시스템들 역시 기능을 사용하기 위해서는 기기들을 추가적으로 구입하여 사용하여야 한다.



 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	계획서		
	프로젝트 명	ToS (Touch on Screen)	
	팀 명	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

## 2 개발 목표 및 내용

### 2.1 목표

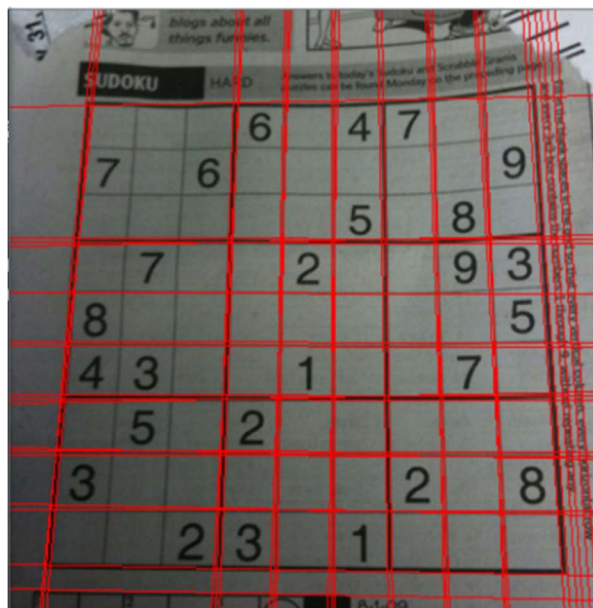
이 프로젝트는 사람의 손을 추적하고 손의 제스처를 인식하는 딥러닝 모델을 학습하고, 해당 모델을 사용하여 발표를 진행하는 사용자의 모션을 실시간으로 분석하고 화면을 제어하는 프로그램 개발을 목표로 한다. 모바일 어플리케이션에서 화면을 인식함으로 출력 장치에 무관한 화면 제어를 가능하게 한다.

### 2.2 연구/개발 내용


#### 앱 프론트 엔드

##### 1. 발표 화면의 경계 인식

- OpenCV의 Hough transform와 사용하여 스크린의 경계를 찾는다.  
영상에서  $(x,y)$  좌표공간의 픽셀들은  $(r,\theta)$ 의 공간에서는 곡선의 형태로 나타난다. 또한, 직선은 한 점의 형태로 나타난다. 이러한 특성을 이용하여 영상에서 직선을 찾아내는 방식이 hough 변환이다.





 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

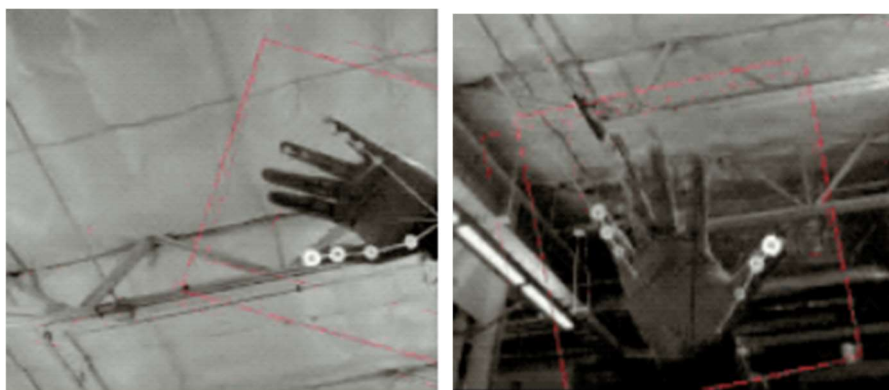
- Perspective transform

perspective transform은 대표적인 와핑(warping)기술 중 하나이다. 와핑 기술은 이미지를 왜곡시키는 기술로써, 정면이 아닌 스크린의 정규화를 위해 사용할 것이다.



## 2. 사용자의 모션 인식

- google의 hand-detection api를 활용하여 손의 위치를 찾는다.  
 학습된 ai를 활용하여 손을 인식하고, 특징점들과 각 특징점의 위치를 계산해 준다.



 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

- 모션인식

손 검출 api를 활용하여 손의 특징 점 21개를 찾아낸다. 인식하게 할 모션의 영상과 21개 점의 움직임을 학습시켜 원하는 모션을 인식하는 AI를 구성한다.

### 3. 좌표 값 연산

- 스크린의 경계와 손의 좌표를 이용해 사용자가 원하는 클릭 지점을 계산한다.
- 모바일에서 전송된 좌표 값을 통해 특정 좌표에서의 행동을 수행한다.

## 백 엔드

### 서버

- Python

### 데이터 베이스

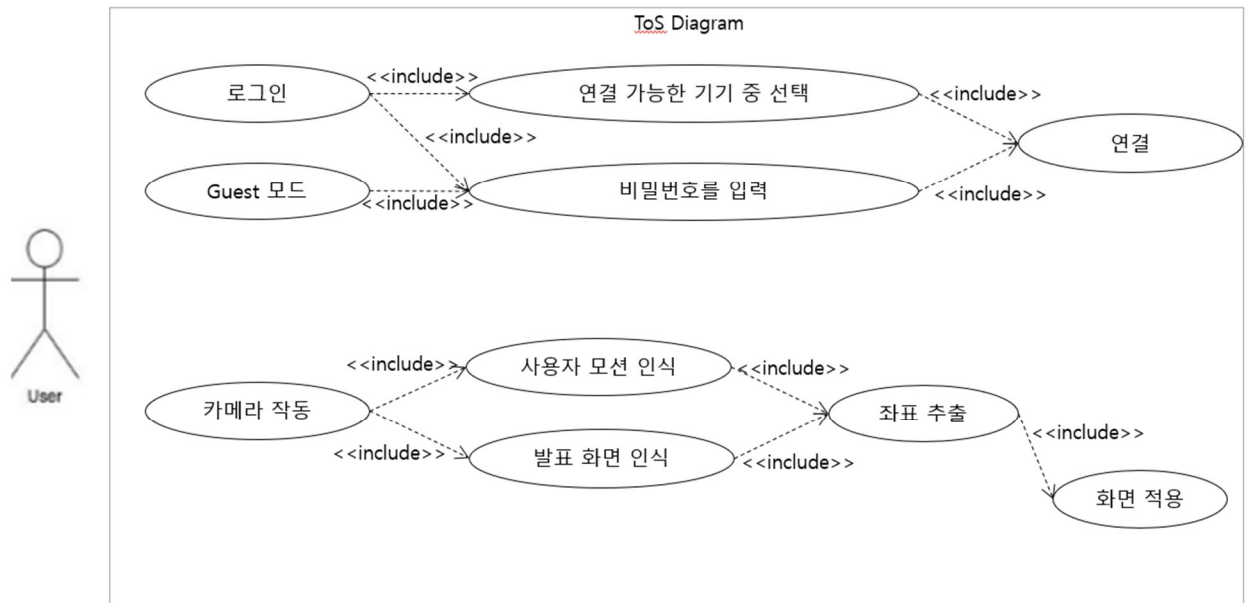
- DBMS: MySQL

사용자의 정보 및 연결 정보, 현재 서버에 연결되어 있는 PC, 모바일 어플리케이션의 정보를 가진다. 서버에서의 연산과 데이터를 분리하여 클라우드 환경의 서버를 안정적으로 구동하고자 한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22


## 2.3 개발 결과

### 2.3.1 시스템 기능 요구사항



사용자는 pc 와 모바일 어플리케이션을 연결하기 위해 로그인 / guest 모드를 선택한다. 로그인 모드는 id, password 를 입력 받아 로그인 한 후, 기존에 연결되어 있던 기기와 연결하거나 비밀번호를 생성하여 새로운 기기와 연결함으로써 pc 와 모바일을 연결한다.

사용자의 선택에 의해 연결이 된 후에는 모바일 디바이스에서 카메라를 작동시켜 사용자의 모션을 인식하고 사용자가 발표를 진행하는 화면을 인식하여 실제 카메라에서 출력된 좌표를 발표 화면에서의 좌표로 변환한다. 이후 pc 로 좌표와 입력 정보를 전달하여 화면을 제어한다.

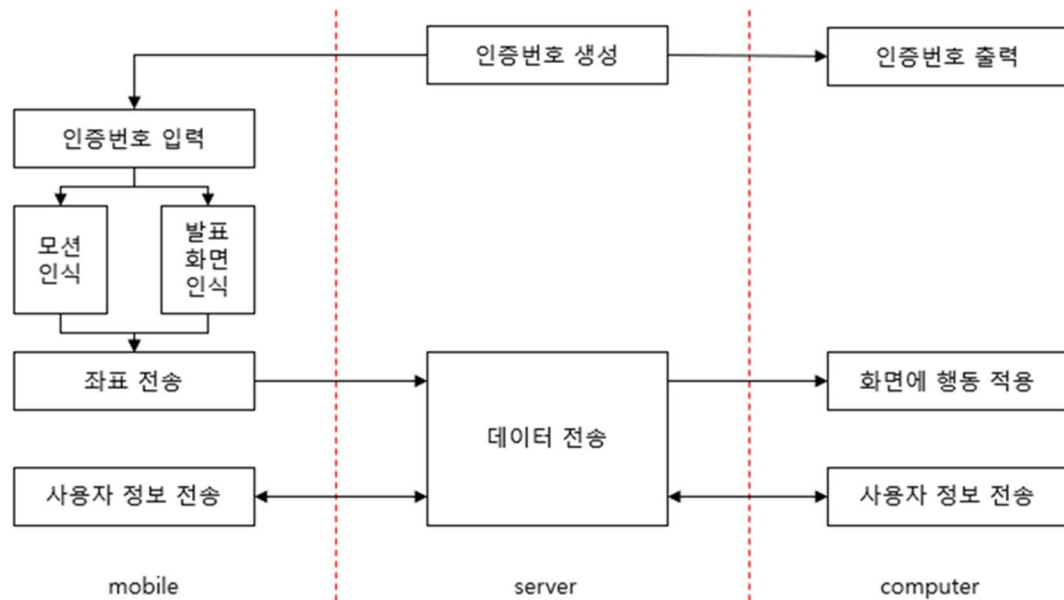
 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

### 2.3.2 시스템 비 기능(품질) 요구사항

요구사항	내용	품질 속성	중요도
NFR1	시스템은 스크린의 경계를 정확하게 인식해야 한다.	신뢰성 요구사항 - 스크린의 경계가 정확해야 클릭 지점을 정확하게 출력할 수 있다.	중
NFR2	시스템은 손의 위치를 정확하게 인식해야 한다	신뢰성 요구사항 - 손의 위치를 정확하게 인식해야 모션 인식 뿐만 아니라 좌표를 정확하게 계산할 수 있다.	중
NFR3	시스템은 손의 모션을 정확하게 인식해야 한다.	신뢰성 요구사항 - 클릭을 원하지 않는데 클릭이 발생할 수 있다.	상
NFR4	시스템은 인식된 좌표를 1초 이내에 클릭해야 한다.	성능 요구사항 - 적당한 시간 내에 클릭이 이루어져야 한다.	상
NFR5	어플리케이션과 pc클라이언트는 직관적인 UI로 구성된다.	사용성 요구사항 - 누구나 직관적으로 사용할 수 있다.	하
NFR6	사용자가 원하는 클릭 지점의 오차는 직경 5cm 이내여야 한다.	신뢰성 요구사항 - 빔 프로젝터 스크린의 크기를 고려했을 때 오차가 5cm 이내여야 사용 가능한 수준이라고 판단했다.	상
NFR7	다양한 환경에서 구동가능해야 한다.	사용성 요구사항 - 본 시스템은 다양한 시간대, 다양한 장소에서 활용될 수 있다.	중
NFR8	어플리케이션은 안드로이드에서 동작한다.	구현 요구사항	중
NFR9	다양한 기종의 안드로이드 휴대폰에서 작동한다.	이식성 요구사항 - 사람들은 다양한 스마트폰을 사용합니다. 많은 사람들이 사용하기 위해 다양한 해상도, 스펙의 스마트폰에서 구현이 가능하다.	하

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

### 2.3.3 시스템 구조



#### 1. PC와 Application의 연결


- 1) PC 클라이언트에서 서버에 인증번호를 요청한다.
- 2) 서버에서 인증번호를 생성하여 PC 클라이언트로 전송한다.
- 3) PC에서 인증번호를 출력하고 Application에서 인증번호를 입력하여 서버로 전송한다.
- 4) 서버에서 인증번호를 비교하여 PC와 Application을 연결해 준다.

#### 2. 스크린 인식

- 1) OpenCV를 이용해 스크린의 경계점을 추출한다.
- 2) 경계점을 이용해 직선을 찾고 스크린의 네 꼭지점을 찾는다.

#### 3. 사용자의 손 인식

- 1) google의 hand-tracking api를 사용해 사용자의 손을 인식한다.
- 2) 모션(ex 주먹 쥐)을 인식한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

#### 4. 좌표 계산

- 1) 모션을 인식한 경우 스크린의 좌표 값과 손의 좌표 값을 정면으로 본 시점으로 변환한다.
- 2) 손과 스크린 경계의 거리 비를 출력한다.


#### 5. 좌표 전송

- 1) application에서 서버를 통해 PC로 좌표를 전송을 해준다.
- 2) PC 클라이언트가 해당하는 지점에서 행동을 한다.

#### 2.3.4

#### 결과물 목록 및 상세 사양

대분류	소분류	기능	형식	비고
알고리즘	터치 위치 연산	스크린 검출, 손의 위치 파악	함수	
	투사변환	원근법을 고려한 스크린 변환	함수	
	모션인식 딥러닝 모델	모션인식 및 모션 분류	AI	
어플리케이션	서버와 통신	인증번호 전송 및 좌표 전송, 연결 유지	함수	
	로그인	사용자 로그인	함수	
PC 클라이언트	PC 클라이언트	인증번호 전송 및 입력 받은 좌표 값을 통한 마우스 제어	함수	
DB	사용자 정보 저장	사용자의 연결 정보 데이터베이스에 저장	DB	
서버	AWS 서버	어플리케이션과 PC간의 통신	함수	AWS

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

## 2.4 기대효과 및 활용방안

### 1. 빔 프로젝터 터치와 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 추첨에도 사용될 수 있다.

### 2. 기존의 빔 프로젝터의 활용이 가능해진다.

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. 어플리케이션과 안드로이드 기기만 있으면 되므로 누구나 사용이 가능하다.

오늘날 남녀노소 할 것 없이 모두가 스마트폰을 갖고 있다. 어플리케이션은 다운로드만 받는다면 사용이 가능하기 때문에 매우 높은 접근성을 가지고 있다.

## 3 배경 기술

### 3.1 기술적 요구사항

#### 3.1.1 개발환경

##### 1. 운영체제

- Window 10, Ubuntu 18.04 LTS, Android

##### 2. 개발 언어

- Python, Java, C++, xml



 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

### 3. 서버

- AWS

### 4. IDE

- PyCharm, Android Studio

### 5. 라이브러리

- OpenCV, MediaPipe, pyQt

## 3.1.2 프로젝트 결과물 환경

#### 1. PC 기반 서비스

- OS: Window 10
- 개발 언어: Python


#### 2. 안드로이드 기반 서비스

- OS: 안드로이드 5.0(LOLLIPOP) 이상의 환경에서 사용 가능하다.
- 개발 언어: JAVA, C++

## 3.2 현실적 제한 요소 및 그 해결 방안

### 3.2.1 하드웨어

- 모션을 인식하는 모델을 학습시키기 위해 많은 프로세스 자원이 요구된다. 모델을 학습시키는 것은 프로젝트 결과물의 성능과도 직결된 문제로, 이를 해결하기 위해 google에서 제공하는 google teachable Machine이나 학교에서 지원되는 KMUSW 딥러닝 클라우드를 활용할 계획이다.
- 어플리케이션을 구동 시키는 스마트폰의 사양에 따라 모션인식의 반응 속도가 다르고 소프트웨어적 차이가 있다. 따라서 같은 버전의 SDK로 통일하여 개발하도록 한다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

### 3.2.2 소프트웨어


1. 키넥트 없이 손을 추적하고 손의 모션을 인식하는 것은 어려운 문제이다. 머신 러닝을 통해 특정한 모션만 인식을 하도록 하고 모션 인식의 성능을 높인다.
2. 카메라가 스크린의 정면을 바라보는 경우보다 그렇지 않은 경우가 더 많다. 스크린을 측면 혹은 약간 아래에서 바라봐도 알맞은 지점을 클릭할 수 있도록 변환이 필요하다.

### 3.2.3 기타

1. 별 다른 기기 없이 어플리케이션 만으로 클릭 지점을 파악해야 한다. 그러므로 스마트폰은 화면에 빔 프로젝터의 스크린 전체가 나오는 곳에 배치해야 한다.

## 4 프로젝트 팀 구성 및 역할 분담


이름	역할
정형섭	<ul style="list-style-type: none"> <li>- Software Project 팀장</li> <li>- 모션인식 모델 개발 및 학습</li> <li>- Hand-Tracking 모델 학습</li> </ul>
심유정	<ul style="list-style-type: none"> <li>- 모션인식 모델 개발 및 학습</li> <li>- Hand-Tracking 모델 학습</li> <li>- GitHub 관리</li> </ul>
유성훈	<ul style="list-style-type: none"> <li>- Application UI 담당</li> <li>- 모바일 프로그래밍</li> <li>- Application 과 PC간의 통신</li> </ul>

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

이규한	<ul style="list-style-type: none"> <li>- 스크린 경계 추출</li> <li>- 입력 좌표 값 연산</li> <li>- 모바일 프로그래밍</li> </ul>
조정근	<ul style="list-style-type: none"> <li>- 서버 구현</li> <li>- PC 클라이언트 개발</li> <li>- Application 과 PC간의 통신</li> </ul>

## 5 프로젝트 비용


항목	예상치 (MD)
모션인식 모델 개발 및 학습	60
학습 데이터 생성	30
UI 설계	10
서버구현	20
모바일 프로그래밍	40
PC클라이언트 개발	10
Application 과 PC의 통신	30
스크린 경계 추출	20
입력 좌표 값 연산	10
합	230

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

## 6 개발 일정 및 자원 관리


### 6.1 개발 일정

항목	세부내용	3월	4월	5월	6월	비고
요구사항분석	요구 분석					
관련분야연구	주요 기술 연구					
	관련 시스템 분석					
계획서	SRS 작성					
설계	딥러닝 모델 설계					
구현	application 구현					
구현	서버 구현					
구현	PC 클라이언트 구현					
구현	시스템 구현					
AI	딥러닝 모델 학습					
테스트	시스템 테스트					

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

## 6.2 일정 별 주요 산출물

마일스톤	개요	시작일	종료일
계획서 발표	개발 환경 완성 (기본 응용 작성 및 테스트 완료) 요구사항 분석 및 관련 기술 연구 <b>산출물:</b> 1. 프로젝트 수행 계획서 2. 계획서 발표 ppt	2020-01-01	2020-03-27
설계 완료	시스템 설계 완료 개발에 필요한 기술 정보 수집 개발 환경 구축 완료	2020-03-27	2020-04-12
중간 평가	딤러닝 모델 개발 스크린 검출 PC와 어플리케이션 간의 통신 <b>산출물:</b> 1. 프로젝트 중간 보고서 2. 프로젝트 진도 점검 표 3. 1차분 구현 소스 코드	2020-03-30	2020-04-23
구현 완료	안정적인 화면의 입력 로그인 기능 추가 사용자에 대한 DB 구축 UI 개선 <b>산출물:</b> 1. 데모버전 2. 각 기능 소스코드	2020-04-25	2020-05-29
테스트	시스템 통합 테스트 <b>산출물:</b> 1. 프로젝트 최종 소스코드 2. 시연 동영상	2020-06-01	2020-06-10
최종 보고서	최종 보고 <b>산출물:</b> 1. 프로젝트 결과 보고서 2. 최종 발표 자료 3. 최종 발표 및 시연 동영상	2020-06-01	2020-06-12


 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

### 6.3 인력자원 투입계획

이름	개발항목	시작일	종료일	총개발일(MD)
정형섭 심유정	Hand-Tracking	2020-04-01	2020-04-18	30
조정근	PC 클라이언트 개발	2020-04-16	2020-04-21	10
조정근	서버 구현	2020-04-13	2020-05-15	20
유성훈 이규한	모바일 프로그래밍	2020-04-01	2020-05-15	40
유성훈	Application UI	2020-04-13	2020-04-23	10
유성훈 조정근	Application 과 PC간의 통신	2020-04-16	2020-04-21	30
이규한	스크린 경계 추출	2020-04-07	2020-04-20	20
이규한	입력 좌표 값 연산	2020-04-27	2020-05-15	10
정형섭 심유정	모션인식 모델 개발 및 학습	2020-04-27	2020-06-06	60

### 6.4 비 인적자원 투입계획


항목	Provider	시작일	종료일	Required Options
개발용 노트북 5대	본인소유	2020-03-29	2020-06-19	
AWS	국민대학교	2020-03-29	2020-06-19	
IONNOIO airxel 빔 프로젝터	본인소유	2020-03-29	2020-06-19	

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22

## 7 참고 문헌

번호	종류	제목	출처	발행 년도	저자	기타
1	웹 문서	On-Device, Real-Time Hand Tracking with MediaPipe	<a href="https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html">https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html</a>	2019.08.19	Valentin Bazarevsky and Fan Zhang	모션인식
2	웹 문서	Pose estimation	<a href="https://www.tensorflow.org/lite/models/pose_estimation/overview">https://www.tensorflow.org/lite/models/pose_estimation/overview</a>			모션인식
3	웹 문서	주성분 분석과 허프 변환을 이용한 직선 검출	<a href="https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07398223&amp;language=ko_KR">https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07398223&amp;language=ko_KR</a>	2018.02	오정수(부경대학교)	비주얼 컴퓨팅
4	논문	A Real-time Hand Gesture Recognition and Human-Computer Interaction System	<a href="https://arxiv.org/pdf/1704.07296.pdf">https://arxiv.org/pdf/1704.07296.pdf</a>	2017.04	Pei Xu	모션인식
5	웹 문서	Perspective Transform	<a href="https://www.sciencedirect.com/topics/computer-science/perspective-transform">https://www.sciencedirect.com/topics/computer-science/perspective-transform</a>	2005	TOM McREYNOLDS, DAVID BLYTHE	비주얼 컴퓨팅
6	기사	2020년, 차세대 VR U-I 시장이 온다	<a href="http://www.vrn.co.kr/news/articleView.html?idxno=11148">http://www.vrn.co.kr/news/articleView.html?idxno=11148</a>	2019.12	정우준 기자	
7	웹 문서	벽면을 터치 스크린으로! 소니의 '엑스페리아 터치'	<a href="http://www.bizion.com/bbs/board.php?bo_table=gear&amp;wr_id=1384&amp;sf_l=tags&amp;stx=%ED%94%84%EB%A1%9C%EC%A0%9D%ED%84%B0">http://www.bizion.com/bbs/board.php?bo_table=gear&amp;wr_id=1384&amp;sf_l=tags&amp;stx=%ED%94%84%EB%A1%9C%EC%A0%9D%ED%84%B0</a>	2017.02	노현	



 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>계획서</b>		
	<b>프로젝트 명</b>	ToS (Touch on Screen)	
	<b>팀 명</b>	KoPI	
	Confidential Restricted	Version 1.3	2020-APR-22