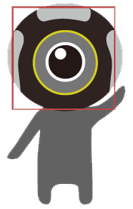
**2017 SW Capstone Design Proposal Report**

**얼굴 인식 기술 기반 출석 관리 시스템**

**I’m Here**

|  |  |  |
| --- | --- | --- |
| **Team Member** | 박세진 | 201222350 |
|  | 문세연 | 201220975 |
|  | 조동민 | 201220894 |
|  | 변우진 | 201220886 |
| **Project** | I’m Here | |
| **Professor** | 윤 대균 교수님 | |



**Team: return SUCCESS;**

목차

[개요 5](#_Toc480334575)

[서론 5](#_Toc480334576)

[문제점 5](#_Toc480334577)

[타겟 / 이해관계자 6](#_Toc480334578)

[주요 기능 6](#_Toc480334579)

[가치 8](#_Toc480334580)

[기존 사례 분석 8](#_Toc480334581)

[유사제품 8](#_Toc480334582)

[관련기술 동향 9](#_Toc480334583)

[인사이트 및 차별성 10](#_Toc480334584)

[개발내용 11](#_Toc480334585)

[목표 11](#_Toc480334586)

[주요 Features 11](#_Toc480334587)

[주요 사용 사례 12](#_Toc480334588)

[설계 17](#_Toc480334589)

[전체 시스템 구조 17](#_Toc480334590)

[SW 구조 18](#_Toc480334591)

[모듈 / 컴포넌트 19](#_Toc480334592)

[개발관리 20](#_Toc480334593)

[개발 환경 및 언어 20](#_Toc480334594)

[프레임워크 활용 및 운영 방안 20](#_Toc480334595)

[리스크 분석 및 회피 방안 21](#_Toc480334596)

[소스 및 변경관리, CI tool 확보 및 운영방안 23](#_Toc480334597)

[비용 분석 24](#_Toc480334598)

[업무분장 24](#_Toc480334599)

[Deploy 계획 25](#_Toc480334600)

[데모 계획 및 시나리오 25](#_Toc480334601)

[KPI 26](#_Toc480334602)

[Future Plan 26](#_Toc480334603)

[참고문헌 27](#_Toc480334604)

Tables

[Table 1 Revision History 4](#_Toc480332960)

[Table 2 학생 관점 주요 기능 7](#_Toc480332961)

[Table 3 교수 관점 주요 기능 7](#_Toc480332962)

[Table 4 출결 시스템 관리자 관점 주요 기능 7](#_Toc480332963)

[Table 5 개발 환경 및 언어 21](#_Toc480332964)

[Table 6 프레임워트 활용 및 운영 방안 21](#_Toc480332965)

[Table 7 리스크 분석 빛 회피 방안 22](#_Toc480332966)

[Table 8 개발 일정 23](#_Toc480332967)

[Table 9 Coding Convention 24](#_Toc480332968)

[Table 10 비용분석 25](#_Toc480332969)

[Table 11 업무분장 25](#_Toc480332970)

Figures

[Figure 1 얼굴 인식 기술의 활용 분야 9](#_Toc480334605)

[Figure 2 딥러닝 기술 활용 분야 10](#_Toc480334606)

[Figure 3 사용 사례 다이어그램 12](#_Toc480334607)

[Figure 4 전체 시스템 구조 17](#_Toc480334608)

[Figure 5 SW 구조 18](#_Toc480334609)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | 변경사항 |
| Proposal draft v1.0 | 2017.4.10 | return SUCCESS; | 초안 작성(개요 및 배경) |
| Proposal draft v1.0 | 2017.4.12 | return SUCCESS; | 기존 사례 분석, 개발 내용 작성 |
| Proposal draft v1.0 | 2017.4.14 | return SUCCESS; | 설계, 개발관리 작성 |
| Proposal draft v2.0 | 2017.4.16 | return SUCCESS; | 기획서 미작성 부분 완성, 목차 수정 |
| Proposal draft v2.1 | 2017.4.17 | return SUCCESS; | 목차 수정, 타겟, 주요기능 내용 수정 |
| Proposal draft v2.2 | 2017.4.18 | return SUCCESS; | 사용사례, 설계 부분 이미지 추가 |

Table 1 Revision History

# **개요**

## **서론**

기존의 전자 출결 관리 시스템은 강의실 문 옆의 출석기기에 학생증을 가져다 대거나, 출석기기를 통해 직접 학번을 입력하는 동시에 수강자의 얼굴 사진을 촬영하는 방식으로 출석 여부를 처리하고 있다. 그러나 전자 출결 시스템의 몇 가지 허점들과, 학생이 항상 학생증이나 학번을 인증해야 하며 이 요구사항을 잊는 경우, 출석결과 변경을 추가로 해야하는 번거로움 등이 존재한다. 본 프로젝트에서는 이러한 단점을 보완하기 위한 대안으로 얼굴 인식 기술 기반 자동 출석 시스템을 개발하기로 하였다.

## **문제점**

전자 출결 시스템의 단점으로는 다음과 같은 것이 있다.

첫째, 학생증이나 학번을 수업 시작 전에 딱 한번 인증하기만 하면 출석 처리가 완료된다. 따라서 해당 학생이 정말 강의실에서 수업을 듣고 있는지, 출석 처리만 하고 수업 참여를 하지 않았는지 여부는 강의자(교수)가 직접 확인하기 전에 알 수 없다.

둘째, 출석 관리 기기에서 사진을 촬영하나 신분 대조 작업은 전혀 하고있지 않다. 이 작업은 강의자가 직접 따로 해야하며 출석한 모든 학생이 본인과 일치하는지 확인하는 것은 매우 번거로운 일이다. 이 두가지 문제는 학생들이 대리출석을 쉽게 할 수 있는 원인이 되기도 한다.

셋째, 간혹 학생이 지각으로 구두 출석을 놓치거나, 정상 출석 처리가 가능한 시간에 전자 출결 처리를 잊었을 때 강사에게 출석 처리 변경을 요청하게 되는데, 수업이 끝난 직후가 아니면 출석 처리 변경을 요청한다 해도 자신이 수업에 참여했다는 것을 확실히 증명하기 힘들어진다.

넷째, 강사들의 입장에서는 전자 출결 사용 여부와 관계 없이 구두출석을 부르는 경우, 적지 않은 수업 시간이 학생들의 이름을 부르는 일로 소요된다는 불편함이 있다. 대형강의의 경우 약 10분 정도가 소요되며 전체 수업시간의 약 1/7정도를 차지한다.

다섯째, 출석 처리를 위해 학생증/학번 인증 과정이 필수적이다. 기존의 전자 출결 관리는 학생증을 소지하거나 학번을 입력하는 행위를 통해 출석 처리가 가능하다는 점에서 사용자 편의성이 낮고 학생 본인이 소지한 매체를 사용한다는 점에서 대리출석이 가능하다.

위와 같은 문제들로부터 본 프로젝트를 통해 기존의 전자 출결 관리 시스템의 문제를 해결하고자 얼굴 인식에 기반한 새로운 출결 관리 시스템을 구현하고자 한다.

## **타겟 / 이해관계자**

### **학생**

대학교에 재학중인 모든 학생을 타겟으로 한다. 가끔 출석 처리를 잊거나 번거롭다고 느끼는 학생들, 정직한 출석 처리를 원하는 학생들에게 특히 효과적인 서비스가 될 수 있다.

### **교수**

대학교에서 강의를 진행하는 모든 교수들 또한 타겟이 된다. 구체적으로, 출석으로 인한 시간적 비용과 대리 출석이라는 제도적 문제가 개선되길 바라는 교수라고 할 수 있다.

### **학교 정보 관리 시스템(전산 시스템/관리자)**

전산시스템은 본 프로젝트의 이해관계자(Stakeholder)가 된다. 학교 정보 관리시스템에는 출석 관리 시스템, 강의 관리 시스템, 성적 관리 시스템 등 다양한 시스템이 포함되어 있다. 출석 관리 시스템이 동작하기 위한 학생정보, 강의정보 등을 사용하기 위해 학교 정보 관리 시스템과의 상호작용이 필요하다.

## **주요 기능**

사용자에게 제공하고자 하는 주요 기능은 얼굴 인식으로 자동으로 출결 처리를 진행할 수 있도록 하고 처리된 출결 상태를 웹 페이지를 통해서 관리할 수 있도록 하는 것이다. 추가로 사용자 별 부가 기능도 제공하고자 한다. 우선 제공하려는 기능의 의미는 다음과 같다.

* **학생 얼굴 등록(검출, 정규화, 학습):** 다각도로 촬영된 학생 얼굴 사진에서 얼굴이 검출되고 딥 러닝을 통해 학습된다. 학습 된 얼굴 정보는 DB에 저장되며 필요에 따라 갱신할 수 있다.
* **출결 정보 조회/관리:** 이 기능은 인증된 사용자에게 웹 서비스로 제공 될 것이다. 사용자에 따라 다른 웹 페이지 UI를 제공한다. 출결 상태는 출석, 지각, 결석으로 구분된다. 출결 정보 관리를 통해 출결 상태 수정이나 출결 상태의 기본 값 설정 등을 수행할 수 있다.
* **수업 참여도, 집중도 정보 확인**: 인증된 사용자에게 웹 서비스로 제공된다. 사용자 권한에 따라 제한된 정보를 제공한다. 학생들의 출석률은 수업 참여도, 얼굴 인식률은 수업 집중도를 위한 데이터로 활용된다. 이 정보는 필요에 따라 통합적 또는 학생 개인별로 보여질 수 있다.
* **출결 동작 제어:** 사용자 요구에 따라 출결을 진행할 필요가 없거나, 추가로 진행 할 경우에 맞게 일종의 출결 처리 on/off 기능을 제공한다. 시스템 전체, 강의실 또는 수업 별로 제어 권한이 사용자에게 다르게 부여된다.

위의 기능들은 사용자 별로 목표 달성을 위해 아래와 같이 나뉘어 제공된다.

### **학생**

|  |  |
| --- | --- |
| 기능 | 설명 |
| 출결 정보 조회/관리 | 웹 페이지를 통해 수강 과목에 따른 본인의 일별 출결 현황과 각 출결 상태에 대한 상세 정보(얼굴 인식률, 인식 시간 등)를 조회할 수 있으며 임의로 출결 상태를 수정할 수는 없다. |
| 학생 얼굴 등록 | 학생은 추후 필요에 따라 관리자에 의해 등록되어 있던 본인의 얼굴 정보를 재등록 할 수 있다. |

Table 2 학생 관점 주요 기능

### **교수**

|  |  |
| --- | --- |
| 기능 | 설명 |
| 출결 정보 조회/관리 | 교수자는 수업을 진행하는 과목 별로 수강 학생들의 출결 현황, 상세 정보를 확인할 수 있으며 출결 상태의 default값 설정과, 완료된 출결 상태를 수정할 수 있는 권한을 갖는다. |
| 수업 참여도, 집중도 정보 확인 | 교수자 본인이 진행하는 강의에 한해서 학생들의 출결 정보에 기반한 수업 참여도와 집중도 정보를 조회할 수 있으며 이 정보는 수업 회차 단위, 강의 단위, 학생 개인 단위로 확인할 수 있다. |
| 출결 동작 제어 | 휴강 또는 보강이 발생할 경우 필요에 따라 수업 시 얼굴 인식 출석 시스템을 사용 여부를 강의, 수업 회차 단위로 설정할 수 있다 |

Table 3 교수 관점 주요 기능

### **출결 시스템 관리자**

|  |  |
| --- | --- |
| 기능 | 설명 |
| 학생 얼굴 등록  (검출, 정규화, 학습) | 학생 얼굴을 최초로 등록하고 갱신할 수 있다. |
| 출결 정보 조회/관리 | 모든 학생들의 출결 정보를 조회, 변경할 수 있다. |
| 출결 동작 제어 | 출결 동작 제어를 시스템 전체 단위까지 통제할 수 있다. |

Table 4 출결 시스템 관리자 관점 주요 기능

## **가치**

* **사용자 편의성과 데이터 정확성 증가**

얼굴 인식 출결 시스템은 수업 시작과 동시에 강의실 내에 설치된 카메라에서 수업이 끝날 때까지 주기적으로 출결 처리 작업을 자동으로 수행한다. 학생들은 수업을 듣는 동안 자연스럽게 카메라에 얼굴이 노출되 출석 처리가 가능해진다. 얼굴 인식으로 정확한 신분을 확인하게 되므로 대리출석 또한 방지된다. 사용자에게 편리함과 정확한 출결 정보를 제공할 수 있다.

* **대학강의 질적 수준 향상**

학생들의 얼굴을 강의시간 동안 여러 번 인식하는 과정에서 얻을 수 있는 인식률 & 출석률 데이터는 수업 피드백에 좋은 자료가 될 수 있다. 학생들의 수업 참여도, 집중도 같은 정보 도출에 활용 가능하다. 또한, 그것들을 데이터화 하고, 해당 데이터들을 빅데이터 처리기법으로 재구성하여 새로운 정보를 얻을 수 있다. 이렇게 얻은 다양한 정보는 교수 학습법, 강의 평가, 학생 흥미 분석 등에 중요한 데이터로 활용되고 강의 질 향상에 도움이 될 것이다.

* **다양한 기관으로의 확장 가능성**

출석 관리 시스템은 학교 외에도 다양한 기관에서 활용되고 있다. 그렇기에 대학교와 유사한 전자 출결 관리 시스템을 사용하거나 필요로 하는 기관에도 충분히 적용 가능할 것으로 기대한다. 본 프로젝트가 성공적으로 완성된다면 여러 기관에서 활용될 수 있을 것이다.

# **기존 사례 분석**

## **유사제품**

* **eXClass:**  eXClass는 출결 관리를 위한 RFID 및 학생증이 필요 없다. eXClass는 어플리케이션으로 출석 서비스를 제공하고 있어 스마트폰/패드 같은 어플 활용이 가능한 기기를 반드시 소지해야 한다.
* **LitBiz 스마트 출결 시스템:** 이 시스템에서는 강의 시간동안 출결 관리가 지속적으로 이루어진다. LitBiz 스마트 출결 시스템은 비콘을 사용한 위치기반 시스템이다. 따라서 출석 기능 외에 강의에 대한 추가적 정보 수집이 불가능하다.
* **UCheck Plus:** UCheck Plus는 웹 기반 출결 관리 시스템이다. 스마트폰을 사용한 위치 추적 기술이 사용된다. 얼굴인식을 지원하지 않지만 앱/웹을 동시지원하기 때문에 사용자의 편의성 좋은 편이다.

## **관련기술 동향**

### **바이오 인식 - 얼굴 인식**

바이오 인식이란 개인마다 다르게 가진 생체 정보(얼굴, 홍채, 지문 등)를 사용해 개인의 신분을 확인하는 방법이다. 기존의 열쇠/ID카드 등을 사용하는 시스템은 분실, 복제, 도용 등이 쉬워 보안성이 낮고, 반드시 휴대해야 하는 등의 수고가 있지만 바이오 인식은 높은 보안성과 별도의 매체 소지 등의 필요가 없는 편리성을 만족시킨다.

그 중 얼굴 인식은 사용자 편의성이 가장 탁월한 기술로, Face detection, Landmark, Normalization, Scaling, Recognition 등의 여러 단계를 거쳐 이루어진다. 얼굴 인식은 사용자의 특별한 동작이나 행위를 요구하지 않고 비접촉식으로 신분을 확인할 수 있어 경쟁력 있는 기술로 평가받고 있다. 아래 figure1은 얼굴 인식 기술이 활용되는 다양한 분야를 보여준다.



Figure 1 얼굴 인식 기술의 활용 분야

### **Deep learning**

딥 러닝은 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화를 시도하는 기계학습 알고리즘의 집합으로 정의되며, 큰 틀에서 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야라고 이야기 할 수 있다. 데이터를 컴퓨터가 알아 들을 수 있는 형태로 표현하고 이를 학습에 적용하기 위해 많은 연구가 진행되고 있으며, deep neural networks, convolutional deep neural networks, deep belief networks와 같은 다양한 기법들이 컴퓨터비전, 음성인식, 자연어처리, 음성/신호처리, 이미지 처리 등의 분야에 적용되어 최첨단의 결과들을 보여주고 있다.

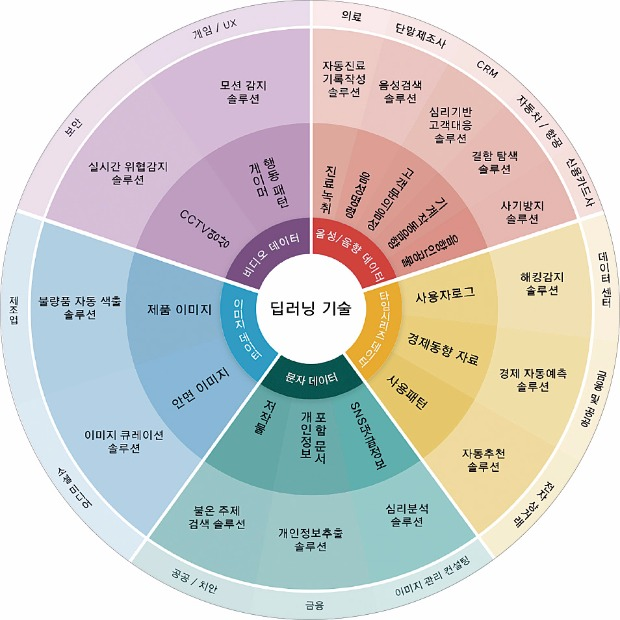


Figure 2 딥러닝 기술 활용 분야

## **인사이트 및 차별성**

기존의 전자 출결의 문제를 해결하기 위한 대안으로 개발된 NFC, 비콘을 활용한 위치 기반 출석 관리 서비스는 여전히 학생의 정확한 출석 확인에 한계가 있다. 학생을 확인하는 것이 아닌 학생이 소지한 폰과 같은 하드웨어를 통해 출석 확인 작업이 이루어지기 때문이다.

관련 기술 동향을 통해서는 얼굴인식이 보안 시스템, 출퇴근 관리에 활용되며, 딥러닝 기술을 통해 안면 이미지 데이터를 학습할 수 있음을 확인할 수 있었다. 따라서 앱 설치나 특정 물품을 소지할 필요가 없는 얼굴 인식 기반의 출석 시스템은 사용자에게 보다 나은 편의와 정확성을 제공한다고 할 수 있다.

기존의 출석 관리 서비스와 다르게 본 프로젝트에서는 학생들의 출석률, 얼굴 인식률 데이터를 활용해 강의 참여도, 집중도, 강의 평가 등에 대한 수업 피드백 정보까지 함께 제공할 것이다.

# **개발내용**

## **목표**

본 프로젝트에서 기획한 모든 기능을 구현하기에는 시간, 비용이 제한적이므로 몇 가지 주요 기능만을 선택적으로 정하여 개발 범위를 축소해 개발을 진행하는 것으로 한다. 개발 목표 및 범위는 다음과 같다.

* 다수의 학생이 찍힌 사진에서 정확한 얼굴 검출 및 인식
* 강의 중 주기적 얼굴 인식 출석 기능 수행
* 웹 서버를 통해 사용자 별 적절한 출석 관리, 전용 서비스 제공
* 학생들의 출석률과 인식률을 통한 수업 참여도, 집중도 정보 제공

## **주요 Features**

* 얼굴 인식 기술을 통한 출결 처리 수행
  + 학생 얼굴 등록 및 학습
  + 얼굴 인식 절차 수행: Face detection, Landmark, Normalization, Scaling, Recognition
  + 출석, 지각, 결석 상태 처리 및 DB 저장
* 수업 참여도 & 집중도 정보 제공
  + 강의/수업 단위로 완료된 학생 출석률, 얼굴 인식률에 따른 분석 수행
  + 분석된 데이터 시각화(그래프/도표 형태)
* 웹 페이지를 통한 출결 관리 서비스 제공
  + 사용자 인증 절차 수행(로그인 기능 포함)
  + 사용자 별 다른 UI 제공
  + 출결 상태 데이터 시각화
* 장치 연결 및 통신 제어
  + 웹캠과 라즈베리파이의 연동 상태 확인
  + 장치와 서버 간 통신 제어

## **주요 사용 사례**



Figure 3 사용 사례 다이어그램

* **얼굴 등록하기:** 얼굴 인식을 통해 출결 처리를 하기 위해 학생들의 얼굴에 대한 정보가 데이터베이스에 저장되어 있어야 한다. 최초의 얼굴 등록은 출결 시스템 관리자에 의해 수행된다. 학생이 얼굴 정보를 재등록 하는 사례는 이번 개발 범위에 포함되지 않았다.
* **출결 처리 관리하기:** 학생은 본인이 수강하고 있는 과목에 한해 자신의 출결 상태를 확인할 수 있다. 교수는 담당 강의에 한해 관련된 수강생(학생)들의 출결 상태를 확인하고 필요하다면 수정하는 작업도 진행할 수 있다.
* **수업 참여도/집중도 확인하기:** 교수는 담당 중인 강의의 학생 출석률, 얼굴 인식률에 따른 수업 참여도와 집중도에 대한 정보를 확인한다.
* **사용자 관리:** 출결 시스템 관리자는 학교 정보 관리 시스템의 지원을 통해 출석 관리 서비스를 제공 받을 사용자의 범위를 지정하고 관리한다.

### **사용사례1: 얼굴 등록하기**

|  |  |
| --- | --- |
| Primary actor | 출결 시스템 관리자 |
| Stakeholders | 학생, 학교 정보 관리 시스템 |
| Text description | 출결 시스템 관리자는 얼굴 인식 출결을 위한 학생 얼굴 정보를 등록한다. |
| Preconditions | 학교 정보 관리 시스템으로부터 학생에 대한 정보를 지원받았다.  출결 관리 웹 페이지에서 관리자 권한으로 로그인을 완료했다.  출결 시스템에 등록된 사용자 목록에서 얼굴을 등록해야 하는 학생의 정보를 조회하고 신분이 일치하는지 확인했다.  사용 중인 디바이스가 네트워크에 연결되었다. |
| Basic Flow | 1. 출결 시스템 관리자가 출결 관리 웹 페이지에서 학생 얼굴 정보 추가 기능을 선택한다. 2. 시스템에서 웹캠을 동작 시키고 촬영해야 하는 얼굴 방향과 카메라에 인식되고 있는 화면을 관리자에게 보여준다. 3. 관리자는 학생 얼굴을 정해진 방향에 맞춰 촬영한다. 4. 시스템에서 웹캠을 통해 촬영된 사진이 성공적으로 분석되면 다음 촬영을 요청한다.   *필요한 모든 방향의 얼굴 사진을 촬영할 때까지 2 - 5번을 반복한다.*   1. 관리자는 촬영 완료를 요청한다. 2. 시스템은 촬영된 사진이 앞으로 해당 학생의 얼굴 인식 출결에 사용될 수 있도록 학습하는 과정을 거치고 DB에 성공적으로 저장되었음을 관리자에게 알린다. 3. 관리자는 학생이 시스템에 성공적으로 인식되는지 테스트 요청을 보낸다. 4. 시스템에서 웹캠을 통해 학생 얼굴을 촬영하고 인식 성공 신호를 보낸다. 5. 관리자는 학생 얼굴 등록을 마친다. |
| Alternative Flows | 1. 진행 과정에서 시스템 동작이 실패했을 경우: 2. 시스템을 재시작하고 관리자 인증 후 이전 상태 복원을 요청한다. 3. 시스템은 이전 상태를 복원한다.   2a. 상태 복원에 실패하는 경우:   * 1. 에러 신호를 보내고 로그를 기록하고 Clean state로 전환한다.   2. 관리자는 학생 얼굴 등록 과정을 처음부터 수행한다.   5a. 사진 분석에 실패하는 경우:   1. 시스템에 사진 사용 불가 신호를 보내고 재촬영을 요청한다. 2. 관리자는 학생 얼굴이 잘 찍히도록 학생에게 얼굴 방향을 지시하고 필요 단계부터 촬영을 진행한다.   2a. 카메라 자체에 문제가 있을 경우   1. 관리자는 카메라를 교체하고 시스템과 장치 연결을 확인한다. 2. 시스템은 관리자에게 연결 성공 신호를 보낸다. 3. 관리자는 진행중인 시나리오를 마저 수행한다. |
| Post conditions | 학생 얼굴 정보가 출결 시스템에 등록되었다. |

### **사용사례2: 출결 처리 조회/관리하기**

|  |  |
| --- | --- |
| Primary actor | 학생, 교수, 출결 시스템 관리자 |
| Stakeholders | 학생, 교수, 출결 시스템 관리자 |
| Text description | 학생은 본인이 수강하는 과목에 대한 자신의 출결 상태를 확인하고 싶다.  교수는 담당 과목에 대한 수강생들의 출결 상태를 관리하고 싶다.  출결 시스템 관리자는 전체 학생의 출결 상태를 관리하고 싶다.  사용 중인 디바이스가 네트워크에 연결되었다. |
| Preconditions | 모든 사용자는 출결 시스템에 등록되었다.  모든 사용자는 출결 시스템에 인증 성공하였다. |
| Basic flows | 1. 교수는 담당하고 있는 1개 이상의 강의 중 하나의 강의를 선택한다. 2. 시스템은 해당 강의를 수강중인 전체 학생들의 출결 상태를 보여준다. 3. 교수는 학생 출결 상태 관리를 시스템에 요청한다. 4. 시스템은 이 강의 범위 내에서 출결 수정 권한을 교수에게 부여한다. 5. 교수는 원하는 학생의 출결 정보를 수정한다.   *교수는 수정하고 싶은 출결 상태가 없을 때까지 과정 5를 반복한다.*   1. 교수는 수정한 출결 정보 저장을 시스템에 요청한다. 2. 시스템은 변경된 정보를 DB에 갱신하고 교수에게 부여한 수정 권한을 돌려 받는다. 3. 교수는 출결 처리 관리를 마친다. |
| Alternative flows | 1. 진행 과정에서 시스템 동작이 실패했을 경우: 2. 시스템을 재시작하고 계정 인증 후 이전 상태 복원을 요청한다. 3. 시스템은 이전 상태를 복원한다.   2a. 상태 복원에 실패하는 경우:   * 1. 에러 신호를 보내고 로그를 기록하고 Clean state로 전환한다.   2. Actor는 출결 처리 조회/관리 과정을 처음부터 다시 수행한다.   1a. 학생이 출결 처리 조회를 하는 경우   * + 1. 학생은 자신이 수강 중인 과목 중 하나를 선택한다.     2. 시스템은 해당 강의에 대한 학생의 출결 현황을 보여준다.   *학생은 원하는 출결 현황 조회가 더 있다면 과정 1a를 반복한다.*   * + 1. 학생은 출결 확인을 마친다.   1b. 관리자가 출결 처리 관리를 하는 경우   * 1. 관리자는 시스템에 등록된 출결 정보 조회를 시스템에 요청한다.   2. 시스템은 요청 받은 출결 정보를 관리자에게 보여준다.   3. 관리자는 원하는 출결 처리 정보를 수정한다.   수정하고 싶은 출결 상태가 없을 때까지 과정 3을 반복한다.   * 1. 관리자는 출결 정보 저장을 시스템에 요청한다.   2. 시스템은 변경된 정보를 DB에 갱신한다.   3. 관리자는 출결 처리 관리를 마친다. |
| Post conditions | 일부 출결 상태가 변경되었다.  Actor는 출결 관리 시스템으로부터 성공적으로 로그아웃 되었다. |

### **사용사례3: 수업 참여도/집중도 확인하기**

|  |  |
| --- | --- |
| Actor | 교수 |
| Stakeholders | - |
| Text description | 교수는 담당 강의에 대한 학생들의 참여도와 집중도를 확인하려 한다. |
| Preconditions | 교수는 출결 관리 시스템에 성공적으로 로그인했다.  교수는 담당하고 있는 강의가 적어도 1개 이상 존재한다.  사용 중인 디바이스가 네트워크에 연결되어 있다. |
| Basic flows | 1. 교수는 당당중인 강의 중 하나를 선택한다. 2. 시스템은 해당 강의에 대한 학생들의 출결 정보와 전체 참여율, 전체 집중도 수치를 분석해 보여준다. 3. 교수는 전체 참여율 또는 전체 집중도 수치에 대한 상제 정보 확인을 요청한다. 4. 시스템은 해당 수치에 대해 학생 별, 수업 회차 별 기준으로 출석률, 얼굴 인식률 정보를 그래프 형태로 교수에게 제공한다.   *다른 강의에 대해 동일한 정보조회를 원할 경우 과정 1부터 반복한다.*   1. 교수는 수업 참여도/집중도 확인을 마친다. |
| Alternative flows | 1. 진행 과정에서 시스템 동작이 실패했을 경우: 2. 시스템을 재시작하고 계정 인증 후 이전 상태 복원을 요청한다. 3. 시스템은 이전 상태를 복원한다.   2a. 상태 복원에 실패하는 경우:   1. 에러 신호를 보내고 로그를 기록하고 Clean state로 전환한다. 2. 교수는 수업 참여도/집중도 확인 과정을 처음부터 다시 수행한다 |
| Post conditions | 교수에게 학생 수업 참여도, 집중도에 대한 통계가 다양한 기준으로 보여진다. |

### **사용사례4: 사용자 관리**

|  |  |
| --- | --- |
| Actor | 출결 시스템 관리자 |
| Stakeholders | 학교 정보 관리 시스템 |
| Text description | 출결 시스템 관리자는 출결 관리 서비스가 적용될 사용자를 관리하려 한다. |
| Preconditions | 출결 시스템 관리자는 성공적으로 시스템에 로그인 되었다.  출결 시스템 관리자가 사용하는 장치가 네트워크에 연결되었다.  학교 정보 관리 시스템과의 통신 연결에 성공했다. |
| Basic flows | * 1. 관리자는 학교 정보 관리 시스템에 출결 시스템 사용 대상자와 강의 정보에 대한 데이터를 요청한다.   2. 학교 정보 관리 시스템은 학생 정보, 강의 정보 등을 출결 시스템 관리자에게 제공한다.   3. 관리자는 제공받은 정보를 통해 사용자에게 적절한 출결 관리 권한을 부여한다.   4. 관리자는 출결 처리 장치가 수업시간에 작동하도록 하는 제어 작업을 수행한다. |
| Alternative flows | 1. 진행 과정에서 시스템 동작이 실패했을 경우: 2. 시스템을 재시작하고 계정 인증 후 이전 상태 복원을 요청한다. 3. 시스템은 이전 상태를 복원한다.   2a. 상태 복원에 실패하는 경우:   1. 에러 신호를 보내고 로그를 기록하고 Clean state로 전환한다. 2. 관리자는 사용자 관리 과정을 처음부터 다시 수행한다 |
| Post conditions | 출결 관리 서비스 이용자에게 적절한 권한이 부여되었다.  출결 장치의 동작 시기가 제어(설정)되었다. |

# **설계**

## **전체 시스템 구조**



Figure 4 전체 시스템 구조

* 라즈베리파이와 웹캠을 통해 학생의 얼굴 사진을 촬영하고 서버로 전송한다. 서버에서는 학생 얼굴 학습 및 저장, 얼굴 인식 작업을 수행한다.
* 서버에서 처리한 사진에 대한 데이터는 DB에 저장된다. 서버에서는 사용자 요청에 따라 DB에 저장된 데이터를 적절히 가공하여 학생 출석 및 참여도, 집중도 정보를 생성하고 제공한다.
* 출결 관리 시스템에서 필요로 하는 학생 정보, 교수 정보, 강의 정보, 강의실 정보 등은 학교 정보 관리 시스템의 지원을 받는다.
* 사용자는 네트워트 연결이 가능한 디바이스를 통해 출결 관리를 웹 서버에 요청한다. 웹 서버는 DB 정보를 사용해 사용자를 인증하고 권한에 따라 다른 UI를 제공한다.

## **SW 구조**



Figure 5 SW 구조

* **Web Page (출석 관리 서비스):** 사용자단의 웹 페이지는 스마트폰과 PC의 웹 브라우저를 통해 제공된다. 브라우저에서 교수자는 자신의 강의에 출석한 학생들과 학생들의 수업 집중도와 같은 피드백 정보를 볼 수 있다. 학생은 자신이 수강하는 강의에서 출결 상태를 확인할 수 있다. 이 웹 페이지는 HTML5/CSS3/JS/JQuery를 기반으로 하는 플러그인을 사용하여 구현된다.
* **Raspberry Pi (사진 촬영):** 라즈베리 파이에 웹캠을 연결하여 강의 중 주기적으로 사진을 촬영하도록 하고 HTTP 프로토콜을 사용하여 서버로 촬영한 사진을 전송한다.
* **Server (이미지 분석):** 서버는 JSP/Servlet을 사용해 개발한다. 라즈베리 파이로부터 사진을 받으면 서버는 얼굴인식 모듈, 딥러닝 모듈을 실행하여 학생 얼굴 등록 또는 얼굴 인식 출결 기능을 수행하고 처리된 결과를 데이터베이스에 저장한다. 또한 클라이언트의 요청이 있을 시 적절한 데이터를 데이터베이스에 요청하고, 전송받은 데이터를 가시화하여 클라이언트의 웹 브라우저로 전송한다.

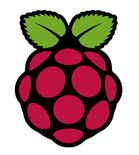
## **모듈 / 컴포넌트**

### **오픈소스**



* **OpenCV:** OpenCV는 실시간 이미지 프로세싱에 중점을 둔 라이브러리다. 이를 응용해 물체 인식, 안면 인식, 제스처 인식 등을 할 수 있고, python과 C++에 대하여 라이브러리를 제공한다.
* **Dlib:** Dlib은 머신러닝, 이미지 프로세싱, 스레딩, 네트워킹 등 여러 분야에 걸쳐 사용할 수 있는 C++, Python라이브러리다. 이 프로젝트는 여기서 이미지 프로세싱을 사용할 것이다. Dlib은 머신러닝, 이미지 프로세싱, 스레딩, 네트워킹 등 여러 분야에 걸쳐 사용할 수 있는 C++, Python라이브러리다. 이 프로젝트는 여기서 이미지 프로세싱을 사용할 것이다.
* **Caffe:** Caffe는 속도와 표현에 중점을 둔 deep learning 라이브러리다. 여러 기능 중 이 프로젝트는 deep learning에 대한 라이브러리를 사용할 것이다.

### **디바이스**



* **Raspberry Pi:** 영국 라즈베리파이(Raspberry Pi) 재단에서 만든 초소형/초저가 PC이다. 교육용 프로젝트의 일환으로 개발되었다. 이 프로젝트에서는 웹캠과 라즈베리파이를 연결해 강의 때 학생들의 사진을 찍고 서버에 보내는 역할을 한다.

# **개발관리**

## **개발 환경 및 언어**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Language | OS | Tool |
| Client | Java script  CSS  HTML | Windows  OSX | atom  eclipse |
| Server | JSP/Servlet  Python  C++ | Ubuntu  Windows | atom  vim |
| Raspberry Pi | Python | Raspbian | vim |

Table 5 개발 환경 및 언어

## **프레임워크 활용 및 운영 방안**

|  |  |
| --- | --- |
| Framework | Description |
| Node.js | Java script 기반 서버 환경 구축 |
| Bootstrap | 클라이언트 UI개발 환경 구축 |

Table 6 프레임워트 활용 및 운영 방안

## **리스크 분석 및 회피 방안**

|  |  |  |
| --- | --- | --- |
| Risk Type | Possible Risk | Solutions |
| Technology | 얼굴 인식, 딥러닝 관련 지식 부족으로 Dlib, OpenCV, Caffe 등 새로운 오픈소스, 라이브러리 사용에 서툴 수 있다. | 관련 자료를 수집하고 사용 방법을 숙지하는데 도움이 되는 튜토리얼과 샘플 코드를 실행, 분석 해 본다. |
| 얼굴 인식의 특성 상 동일한 얼굴을 다각도로 촬영한 사진이 요구되어 데이터를 수집에 어려움을 겪을 수 있다. | 이미지 라이브러리를 프로젝트 지원금으로 구매하여 작업을 수행한다. 또는 팀원 및 팀원의 지인들의 사진을 확보해 데이터로 사용한다. |
| 각각의 모듈, 컴포넌트, 디바이스를 연결하고 통신하는 과정에서 개발 일정이 지연될 수 있다. | 비슷한 사례를 조사해 활용 가능한 대안을 찾는다. 각 파트에 대한 개발 과 별개로 성공적으로 커넥션이 생성되는지 미리, 수시로 테스트 한다. |
| People | 팀원 간 일정 조율이 힘들 수 있다. | 모든 팀원은 일정을 공유하고 자주 연락할 수 있도록 한다. 메신저, Slack 등을 최대한 활용한다. |
|
| Organizational | 설계와 구현 과정에서 팀원들 간의 의견 대립이 발생할 수 있다. | 팀장이 팀원들의 의견을 수렴하여 결정을 내리는 것을 원칙으로 하고 팀원들은 해당 결정을 따른다. |
|
| 팀원마다 할당된 업무를 달성하지 못 할 수 있다. | 팀원 간 적극적인 협업을 통해 본인의 업무와 유사한 업무를 우선적으로 해결하도록 한다. |
| Tools | 처음 사용하는 툴에 어려움을 겪을 수 있다. | 해당 툴들의 공식 사이트 방문을 통해 가이드라인 및 튜토리얼을 진행하고 사용법을 숙지한다. |
| 다양한 개발 툴로 인해 통합의 어려움이 있다. | 해당 프로젝트의 개발 툴을 Vim, Eclipse, Atom 으로 통일한다. |
| Requirements | 프로젝트 진행 중에 분석되지 않은 요구사항이 존재할 수 있다. | 새로운 요구사항들에 우선순위를 정해 개발 범위에 반영하도록 한다. |
| 기존 요구사항의 변경이 필요할 수 있다. | 요구사항의 변경이 필수적인지 여부를 파악하고 필요하다면 변경 사항에 맞게 요구사항을 수정한다. |
| Estimation | 프로젝트의 난이도 및 개발 범위를 상대적으로 낮게 예상하고 있을 수 있다. | 구체적인 설계를 통해 난이도의 지표가 되는 요소들과 범위를 명백하게 설정한다. |

Table 7 리스크 분석 빛 회피 방안

**개발 일정**

* Iteration 3부터는 2주 단위로 적용한다. (J: 조동민, B: 변우진, P: 박세진 M: 문세연)
* Priority는 최소1~최대 10로 구분한다

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Iteration | Features | Priority | Task Description | Status | Owner | Supporter |
| 1 | **개발환경 설정** | 5 | Dlib, OpenCV  설치/테스트 | COMPLETE | J | P, M |
| 5 | 라즈비안 설치 | COMPLETE | J |  |
| 5 | Caffe 설치 & 구동 | COMPLETE | B |  |
| 2  (~4/21) | **이미지 분석** | 5 | 얼굴 검출 | IN PROGRESS | P | ALL |
| 5 | 랜드마크 검출 | IN PROGRESS | M | ALL |
| 7 | 정규화 | IN PROGRESS | J | ALL |
| 3 | **Feature 데이터 분석** | 6 | Feature 추출 | TODO | M | ALL |
| 6 | Feature 러닝 | TODO | J | ALL |
| 6 | Feature 매칭 | TODO | P | ALL |
| **얼굴 인식**  **동작 테스트** | 7 | 얼굴 인식 | TODO | J | ALL |
| 7 | 웹캠 & 라즈베리파이  연동 | TODO | J |  |
| 10 | 얼굴인식 모듈 수정 | TODO | P | ALL |
| 6 | 샘플데이터 테스팅 | TODO | M | J |
| 4 | **출결 시스템**  **구현** | 4 | GUI설계/구현 | TODO | B | P |
| 3 | 로그인/권한 설정 기능 구현 | TODO | B |  |
| 10 | 서버 구축 | TODO | B | P |
| 10 | 출석 처리 알고리즘 | TODO | P | ALL |
| 10 | 서버&장치 연동 | TODO | J | B |
| 9 | DB 구축 | TODO | P | B |
| 10 | DB&서버 연동 | TODO | B | P |
| 5 | **테스트 데이터 학습** | 5 | 테스트 데이터 수집 | TODO | M | P |
| 8 | 필터링 & 테스팅 | TODO | M | ALL |
| **데모 & 테스팅** | 7 | 테스팅 & 디버깅 | TODO | B | ALL |
| 7 | 데모 환경 구현 | TODO | M | ALL |

Table 8 개발 일정

## **소스 및 변경관리, CI tool 확보 및 운영방안**

### **Coding Convention**

|  |  |
| --- | --- |
| Category | Description |
| Code lay-out | 들여쓰기는 띄어쓰기 4칸으로 처리한다. |
| 한 줄은 최대 79자로 작성한다. |
| 최상위 함수와 클래스 정의는 2줄씩 띄어 쓴다. |
| 클래스 내의 메소드 정의는 1줄씩 띄어 쓴다. |
| Whitespace in Expressions and Statements | 대괄호([])와 소괄호(())안, 쉼표(,), 콜론(:), 세미콜론(;)앞의 공백은 피한다. |
| 키워드 인자와 인자의 기본값의 =는 붙여 쓴다. |
| Comments | 코드 주석은 항상 코드에 맞게 갱신한다. |
| 불필요한 주석은 달지 않는다. |
| Naming Conventions | 소문자L, 대문자O, 대문자I는 변수명으로 사용하지 않는다. |
| 모듈명은 짧은 소문자로 구성하며 필요하다면 밑줄로 나눈다. |
| 클래스명은 [카멜케이스(CamelCase)](http://en.wikipedia.org/wiki/CamelCase)로 작성한다. |
| 함수명은 소문자로 구성하되 필요하면 밑줄로 나눈다. |
| 서브클래스의 이름충돌을 막기 위해 밑줄 2개를 앞에 붙인다. |
| 상수는 모듈단위에서만 정의하며 모두 대문자에 필요하다면 밑줄로 나눈다. |
| Programming Recommendations | 코드는 가능한 한 어떤 구현에서도 불이익이 없게끔 작성한다. |
| 클래스 기반의 예외를 사용한다. |
| 모듈이나 패키지에 자기 도메인에 특화된 한 기반 예외 클래스를 빌트인 된 예외를 서브 클래싱 해 정의한다. |

Table 9 Coding Convention

### **소스 및 변경 관리**

GitHub을 사용하여 역할 분장에 맞춰 각자가 맡은 소스를 나눠서 관리한다.

### **CI tool**

Travis를 Git repository와 연동하여 한번 push를 할 때 마다 Build시 에러 여부를 확인한다.

## **비용 분석**

|  |  |  |  |
| --- | --- | --- | --- |
| 품목 | 가격(원) | 비고 | |
| Raspberry Pi 3 | 60250 | 웹캠 연동을 통한 촬영 및 서버 통신 | |
| TGCAM-T1600 | 16660 | 라즈베리파이와 연동 + 촬영 | |
| SD Card | 12500 | 라즈비안 OS 설치 | |
| 합계 | | | 89410 |

Table 10 비용분석

# **업무분장**

|  |  |
| --- | --- |
| 박세진 (팀장) | 변우진 |
| + 개발 업무   * DB 설계/구축 – 서버 연동 * Dlib 테스트 및 코드 수정 * OpenCV 테스트 및 적용 * UI 설계   + 부가 업무   * 일정 관리, 문서 검토/제출 | **+ 개발 업무**   * Caffe 설치 및 구동 * DB 구축 * 서버 구축/관리 - DB연동 * 웹 페이지 제작 |
| 문세연 | **조동민** |
| + 개발 업무   * 라즈베리파이 – 서버 연동 * Dlib 테스트 및 코드 수정 * OpenCV 테스트 및 적용 * 테스트 데이터 필터링   + 부가 업무   * 예산 관리, 기자재 주문 | **+ 개발 업무**   * 라즈베리파이 – 웹캠 연동 * 라즈베리파이 – 서버 연동 * Dlib 테스트 및 코드 수정 * OpenCV 테스트 및 적용 |
| 공통 업무 | |
| + 개발 업무   * 얼굴 인식 출석 처리 알고리즘 개발   + 부가 업무   * 문서 작성, 발표 * 멘토 커뮤니케이션 * 테스트 데이터 수집 | |

Table 11 업무분장

# **Deploy 계획**

## **데모 계획 및 시나리오**

Demo 시나리오는 프로젝트의 개발 목표를 충실하게 이행했는지 판단할 수 있도록 설정하며, 성과 평가는 KPI(Key Performance Indicators) 를 통해 측정한다. KPI는 사전에 정의한 후 프로젝트를 진행하면서 지표를 측정한다.

### **시나리오**

* 사용자 얼굴 등록
  1. 관리자 계정으로 출결 시스템을 시작한다.
  2. 사용자 얼굴 등록 기능을 수행한다.
  3. 등록할 사용자 정보를 조회해 등록 가능한지 확인한다.
  4. 웹캠을 통해 사용자 얼굴을 다각도로 촬영한다.
  5. 웹 서버로부터 사용자 얼굴 등록이 정상적으로 되었음을 확인 받는다.
  6. 얼굴 인식 테스트 기능을 수행해 사용자가 시스템에 인식되는지 보인다.
* 출결 진행 상황 확인
  1. 웹캠을 통해 강의실 또는 모형 강의실 내에 있는 사람들의 얼굴이 보이도록 임의로 지정한 강의 시간동안 사진을 촬영한다.
  2. 라즈베리파이에서 웹 서버로 전송된 사진에서 얼굴을 검출하고 DB에 저장된 사용자 얼굴과 매칭되는 것을 찾아 인식이 올바르게 이루어지는지 확인한다. 인식된 사용자의 출결 진행률을 DB에 저장한다.
  3. 강의가 종료된 후 출결 진행률에 따라 출석, 지각, 결석 처리가 DB에 저장 된 것을 확인한다.
* 출결 상태 조회/관리
  1. 학생 또는 교수 계정으로 웹 페이지를 통해 출결 상태 관리를 시작한다.
  2. 수강 중인 강의, 담당 강의 리스트를 웹 페이지에서 확인한다.
  3. 원하는 강의를 선택한다.
  4. 출결 시스템에서 학생 개인의 출결 상태 또는 수강생들의 출결 상태를 사용자에게 보여준다. 교수 계정의 경우 출결 상태를 변경하고 저장하는 기능을 진행한다.
* 학생 계정의 출결 조회 UI는 아래와 같으며 교수 계정의 경우엔 전체 학생의 출결 정보를 나타낼 수 있도록 한다.



Figure 6 출결 조회 UI 예상도(학생 기준)

### **시연방식**

데모 준비 과정 및 시연 당일 환경을 고려하여 적합한 것으로 시행한다. 학교 정보 관리 시스템에 접근하는 것에 한계가 있으므로 학생, 교수, 강의실, 강의 등의 데이터를 담고 있는 DB를 설계/구축해 사용한다.

1. 강의실에 출결 시스템 동작을 위한 장치 직접 설치 또는 데모용으로 제작한 강의실 모형, 학생 모형을 사용해 시연 진행

2. 데모 동영상을 별도로 제작한 후 상영

## **KPI**

정량적

정확도 90% 이상의 얼굴 인식이 가능하도록 한다.

매 학습마다 실제 데이터와 학습된 데이터 오차를 0.5% 미만으로 한다.

강의 시간 중 출석을 하였을 때 출석이 될 확률을 100%로 한다.

다른 사람으로 출석 될 확률을 0으로 한다.

정성적

개발 일정을 최대한 준수할 수 있도록 한다.

사전에 정의한 개발 규칙을 최대한 준수한다.

계획한 범위까지 개발을 완수한다.

역할 분담 및 업무 수행을 정한대로 수행한다.

# **Future Plan**

* **시선 추적을 통한 수업 집중도 파악:** 현재 구현 계획인 것은 얼굴의 T존 또는 정면 인식 비율을 통해 참여도를 분석하는 것인데 보다 정확한 데이터를 얻기 위해선 눈동자의 움직임을 인식하고 분석할 필요가 있다.
* **어플리케이션 개발:** 웹 페이지 외에도 스마트폰/wearable device 어플리케이션을 개발하여 기존의 서비스에서 사용 가능한 기능을 제공한다.
* **출결 시스템 사용 설정:** 휴강이나 보강에 따라 출결 진행의 필요성이 달라질 수 있다. 프로젝트 일정에 여유가 있거나 프로젝트 종료 후에 이어서 개발이 진행된다면 교수, 관리자에 한해서 정해진 단위로 출결 시스템 동작을 제어할 수 있는 기능을 제공한다. 추가로 필요에 따라 진행되지 않은 출결 상태의 기본 값을 원하는 상태로 설정할 수 있도록 한다.
* **셀프 러닝 기능 제공:** 얼굴 인식을 하기 위해서는 필수적으로 데이터 학습 과정을 거쳐야 하는데 관리자 입장에서 수많은 학생들의 사진데이터를 일일이 구하는 것뿐만 아니라 성형, 화장 등의 이유로 얼굴이 변한 학생들의 데이터를 학습시키는 일에 큰 어려움이 따른다. 그렇기 때문에 학생 본인이 스스로 사진을 가지고 학습시킬 수 있는 기능을 제공한다.

# **참고문헌**

* **기존사례분석**

eXClass: <http://www.tomatosystem.co.kr/img/dm/tomato_news1403M/eXClass.pdf>

LitBiz 스마트 출결 시스템: <http://litbiz.tistory.com/19>

UCheck Plus: <http://uattend.kr/>

* **관련 기술 동향**

문현준, 김상훈 “얼굴인식 기술동향”, 정보처리학회지 제 20권 제 3호 [2013], pp.14-15

<https://ko.wikipedia.org/wiki/%EB%94%A5_%EB%9F%AC%EB%8B%9D>

* **개발환경구축**

<https://www.raspberrypi.org/>

<https://docs.travis-ci.com/>

<https://guides.github.com/activities/hello-world/>

<http://caffe.berkeleyvision.org/>

<http://dlib.net/>

<http://opencv.org/>