

S.C.V (Supreme Checking Viewer)

Ver.1.0

2018.06.23.

한국 외국어대학교
정보통신공학과
1팀(S.C.V)

201601406(T박지훈),200903581(최은기),201402239(이대홍),
201400978(나윤호),201502533(이은지),201601433(박초희)

머리말

본 문서는 인텔에서 개발한 실시간 영상처리, 기계 학습을 위한 소프트웨어 라이브러리(open cv)와 안드로이드 장고 웹서버 HTTP 통신을 이용하여 강의실에서 이루어지고 있는 출결관리를 원활히 도와줄 수 있는 서비스이다. 파이썬에서 제공하는 library들을 통해 이미지 처리와 속도를 높였다. 얼굴인식 과정에서는 HOG 알고리즘을 사용해 전체에서 얼굴의 값을 정규화 하고 얼굴인식에서는 PCA(주성분 분석)을 사용한다. 분류 과정에서는 많이 쓰이는 ANN, KNN, SVM 중 본 프로젝트에서는 KNN을 이용하여 같은 얼굴을 가진 사람을 매칭한다.

목차

1 개요.....	5
1.1 목적.....	5
1.2 관련문서.....	5
1.3 참고사이트.....	6
1.4 용어.....	7
2. 제품소개	
2.1 제품 필요성.....	8
2.2 제품 사용법.....	9
2.2.1 회원가입.....	9
2.2.2 로그인.....	10
2.2.2.1 사용자_학생.....	10
2.2.2.2 사용자_교수.....	11
3.시스템 구성도 및 설계	
3.1 전체 시스템 구성.....	12
3.1.1- 개발환경.....	13
3.1.2 -소프트웨어.....	13
3.1.2.0 - 얼굴인식 및 선별의 과정.....	14
3.1.2.1 - 사용되는 API 및 library 구조.....	15
3.1.2.2 - 수학적 배경.....	16
3.1.2.3 - 얼굴인식에 사용되는 알고리즘.....	17
4.프로젝트 세부일정.....	20
5.팀원 담당 업무 및 Version Control.....	20
6.부록(소스코드 및 프로젝트 결론).....	22
6.1 안드로이드코드.....	22
6.2 서버코드.....	23
6.3 라즈베리파이코드.....	35
7.프로젝트 진행 후 느낀점.....	37

표목차

[Table1] 관련 문서.....	5
[Table2] 참고사이트.....	6
[Table3] 용어 및 약어.....	7
[Table4] 개발환경.....	10
[Table5] 프로젝트 세부일정.....	17

그림목차

[Figure 1] 전자출결 시스템 문제점 통계자료.....	8
[Figure 2] 전자출결 시스템 불만족 통계자료.....	8
[Figure 3] 시스템 구성도.....	9
[Figure 3] 어플사용법.....	9
[Figure 4] 어플사용법.....	10
[Figure 5] 어플사용법.....	10
[Figure 6] 어플사용법.....	11
[Figure 7] 어플사용법.....	11
[Figure 8] 시스템 구성도.....	12
[Figure 9] opencv architecture.....	14
[Figure 10] 간략화된 신경망 구조.....	15
[Figure 11] 기계학습 프로세스.....	15
[Figure 12] numpy 함수.....	16
[Figure 13] numpy 함수 실행속도.....	17
[Figure 14] HOG gradient image.....	17
[Figure 15] HOG 입력 이미지 Block Cell.....	17
[Figure 16] 정규화 식.....	17
[Figure 17] SVM 그래프.....	17
[Figure 18] 유클리드 거리.....	18
[Figure 19] 유클리드거리 task.....	18
[Figure 20] 마할라노비스 거리.....	18
[Figure 21] 화이트닝 변환 된 task.....	19
[Figure 22] 안드로이드 Version Control	20
[Figure 23] 서버 Version Control.....	21
[Figure 24] 안드로이드 프로젝트.....	22

1 개요

본 장에서는 라즈베리 파이와 안드로이드와 장고 웹서버를 활용한 HTTP 통신을 이용하여 제작한 I.S.A에 대해 소개한다. I.S.A는 기존 전자출결의 허점(대리출석, 수업 중 무단이탈)을 제거하고 얼굴인식을 통해 자신의 등록된 정보(회원 사진)와 실시간으로 비교하여 출석을 확인하는 서비스를 제공한다. 서버 DB에 저장되는 데이터를 머신 러닝으로 학습하는 GoogleNet에서 제공하는 신경망을 사용하였다. 또한 파이썬에서 제공하는 numpy와 cv2를 활용해 pillow 함수를 사용하여 효율적으로 연산할 수 있게 만들었다. 보고서를 통해 개발의 목적을 명시하고 그와 관련된 문서, 용어, 참고 자료에 대해 소개한다.

1.1 목적

최근 학생들과 교수들의 편의를 도모하기 위해 전자출결 시스템이 많이 도입되었다. 전자출결은 교수들이 인증 번호를 발행하고 출석을 열어주면 시간 내에 학생들이 번호를 입력하는 방식이다.

하지만 예상하였던 전자출결의 성과와 달리 여러 가지 문제점들이 생겨났다. 강의실이 있지 않은 친구들이 스마트 번호를 받아서 입력하면 부정 출석이 가능하다. 또한 출석하고 수업 중 무단이탈하더라도 체크할 수 없는 문제점이 있다. 이러한 문제점에 해결하기 위해 I.S.A는 학생들의 얼굴을 기반으로 하여 대리출석을 방지하고, 학생 역시 앱을 통하여 본인의 출석 여부를 손쉽게 파악할 수 있게 만드는데 그 의의가 있다.

1.2 관련문서

출처 및 출판사	문서 제목
Johnwiley&sons	KREYSZIG 공업수학1
이지스 퍼블리싱	Do it! 안드로이드 프로그래밍
한빛 미디어	장고로 배우는 쉽고 빠른 웹 개발
논문	파이썬 소개//column IT
정보과학회논문	PCA와_얼굴방향_정보를_이용한_얼굴인식
대한전자공학회	Deep Convoution Neural Networks in Computer Vision
서울대학교 컴퓨터 공학부 강의노트	기계학습개론/딥러닝
정보과학회논문	CUDA를_이용한_PCA_기반_얼굴인식의 가속화

[Table 1]

1.3 참고사이트

subject	URL
기계학습	https://medium.com/@jongdae.lim/
아핀변환	https://blog.naver.com/stareat0707/221163666490
openface	https://github.com/cmusatyalab/openface/tree/master/models/openface
딥러닝	https://www.popit.kr/openface-exo-member-face-recognition/
이미지 Processing	http://opencv-python.readthedocs.io/en/latest/doc/08.imageProcessing/imageProcessing.html
배치기울기하강	https://hbfs.wordpress.com/2012/04/24/introduction-to-gradient-descent/
dlib	http://dlib.net/
신경망	https://m.blog.naver.com/PostView.nhn?blogId=laonple&logNo=220793640991&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F
이미지의 기하학적 변형	http://opencv-python.readthedocs.io/en/latest/doc/10.imageTransformation/imageTransformation.html
유클리디안 거리/화이트닝 변환	http://yeunhwa.tistory.com/entry/%EA%B1%B0%EB%A6%AC-%EC%B2%99%EB%8F%84%EC%97%90-%EB%8C%80%ED%95%B4%EC%9C%A0%ED%81%B4%EB%A6%AC%EB%94%94%EC%95%88-%EA%B1%B0%EB%A6%AC-%EB%A7%88%ED%95%A0%EB%9D%BC%EB%85%B8%EB%B9%84%EC%8A%A4-%EA%B1%B0%EB%A6%AC-%ED%99%94%EC%9D%B4%ED%8A%B8%EB%8B%9D-%EB%B3%80%ED%99%98

[Table 2]

1.4 용어 및 약어

용어	해석
라즈베리 파이(Rasberrypi)	라즈베리 파이는 영국 학교 재단이 컴퓨터 과학 교육의 증진을 위해 만든 초소형 컴퓨터로써, 사용자가 원하는 대로 기능 확장하거나 용도를 변경 가능
ML	머신 러닝은 데이터를 보고 학습을 하는 것 즉, 경험을 통해서 다른 비슷한 유사한 일을 진행할 때 더 효율적으로 처리할 수 있도록 시스템의 구조나 파라미터를 바꾸는 것을 의미
API	Application Programming Interface 로써 운영체제와 응용프로그램 사이의 통신에 사용되는 언어나 메시지 형식을 뜻함
PIL	PIL은 Python의 가장 유명한 Image Processing 라이브러리. python3에서는 pillow로 대체

[Table 3]

2.1 제품 필요성



[Figure 1]

69.4%가 전자출결시스템에 불편함 느껴

2015년 05월 18일 (월) 14:37:45

이신후 기자 sinoo_@naver.com



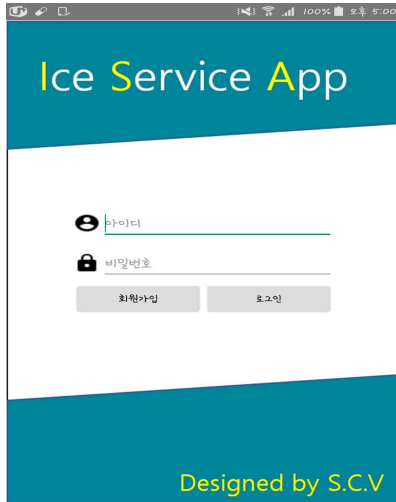
-동덕여대 학보 중

[Figure 2]

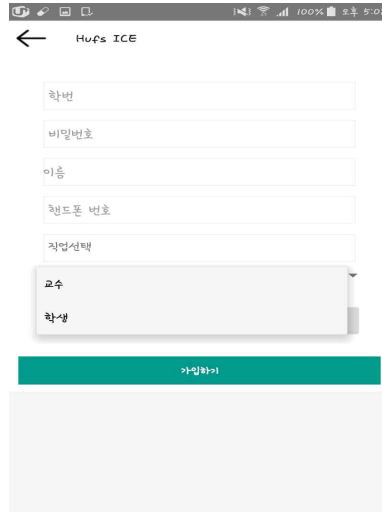
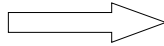
위 자료에서 볼 수 있듯이 전자출결 시스템의 본래 목적은 출석 확인 시간을 단축하고 투명한 성적관리 등을 위해 도입되었다. 전자출결 제도는 학생이 블루투스 기능을 활성화시킨 후 앱을 통해 출석하거나 교수가 수업 중 인증번호를 발행하면 학생이 일정 시간 내 전자출결 앱에 이를 입력하는 '인증번호 출석' 두 가지 방식 있다. 전자는 강의실 내에 위치한 단말기의 비콘이 인식해 이뤄진다. 그러므로 비콘은 강의실 안에 있는 단말기의 신호만을 인식해야 한다. 하지만 강의실 밖에서 들어오는 신호도 인식될 수 있으므로 완전하지 못하다. 후자는 학생들이 발행된 인증번호를 학우들에게 알려주면 누구라도 손쉽게 출석체크할 수 있다는 문제점이 있다. 이와 같은 문제점을 해결하기 위해 자신의 얼굴에서 고유한 특징(face_land_mark)을 가지고 있는 얼굴 특성을 통하여 개개인을 확인하는 I.S.A의 도입이 필요하다고 할 수 있다.

2.2 제품 사용법

2.2.1. 회원가입



[Figure 3]



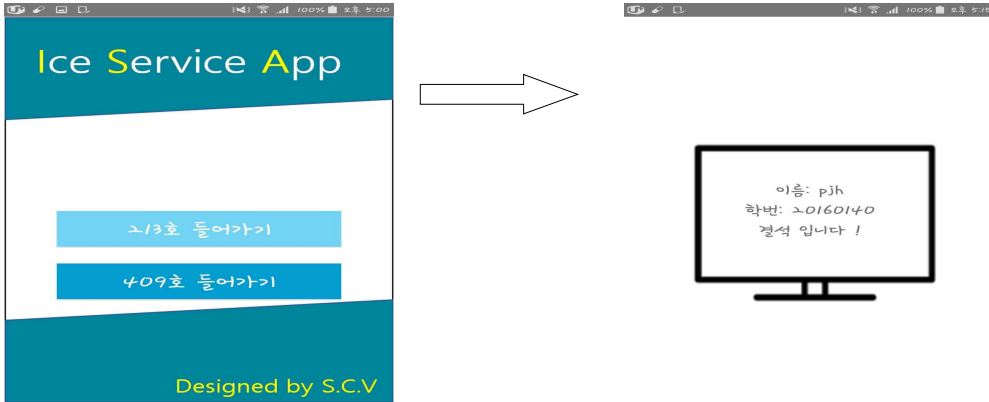
[Figure 4]

첫 화면에서는 회원가입과 로그인 버튼이 있는데, ISA어플을 이용하기 위해서는 회원가입 버튼을 클릭하여 정보를 기입해야 한다. 회원가입 버튼을 누르면 오른쪽 화면과 같이 이동하는데 이 창에서 회원의 정보를 입력하고 직업을 선택하여 교수인지 학생인지 구분한다. 아래에 있는 ADD PHOTO 버튼을 누르면 핸드폰에 있는 자신의 사진을 선택하여 등록할 수 있고 이 사진을 분석하여 회원 정보와 함께 데이터를 저장해준다.

2.2.2 로그인

사용자는 회원가입이 완료되면 가입한 정보로 로그인할 수 있는데, 이 때의 사용자는 교수와 학생의 경우로 나눠서 본다.

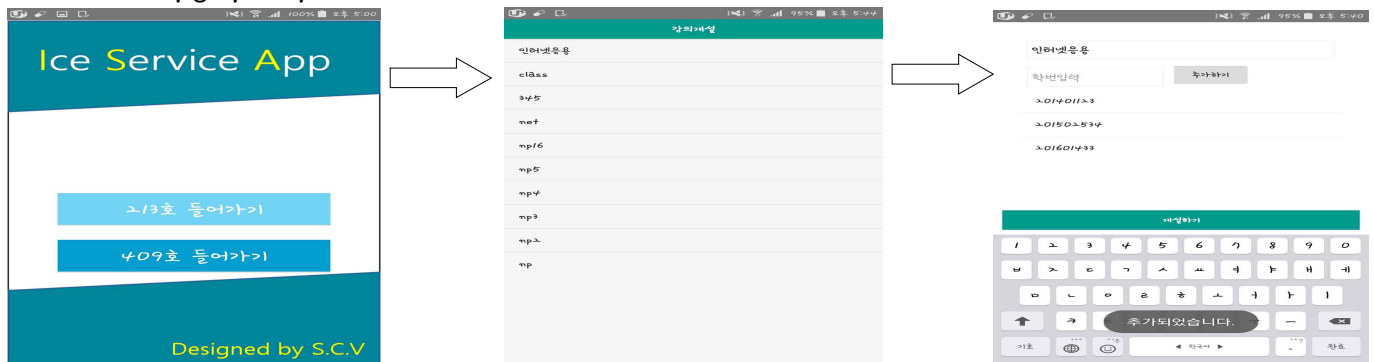
2.2.2.1 사용자_학생



[Figure 5]

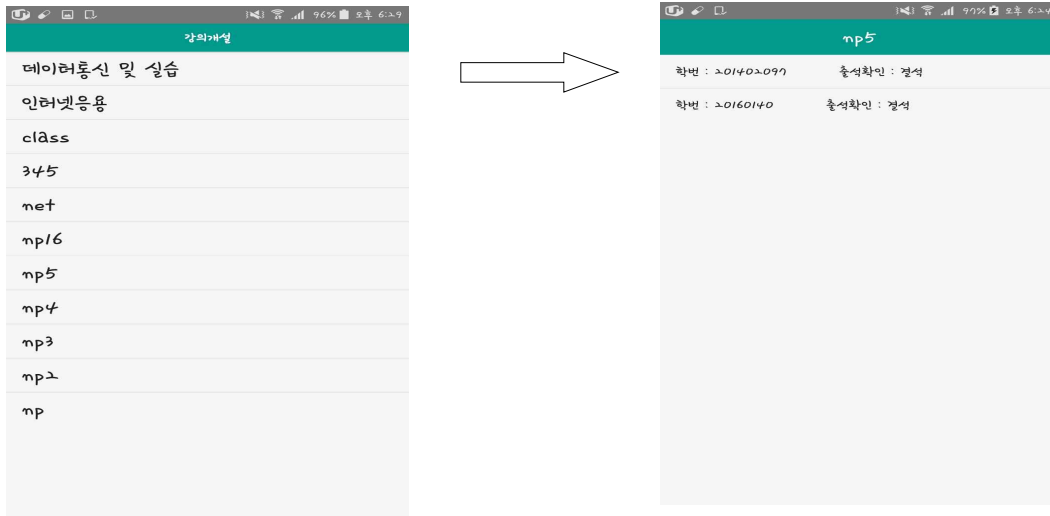
학생정보로 로그인 한 경우, 왼쪽 화면과 같이 213호와 409호로 버튼이 뜨는데 213호 강의 출석을 확인하려면 '213호로 들어가기 버튼'을 누른다. 버튼을 누르면 위의 오른쪽과 같은 화면이 뜬다. 학생의 이름과 학번, 그리고 출결을 알려주는 문구가 뜨게 된다. 이로써 간편하게 자신의 출석을 확인할 수 있다. 409호로 들어가기 버튼을 누르면 409호 스터디룸 예약을 할 수 있고, 스터디룸의 이용 현황을 알 수 있다. 409호는 추후에 마무리 할 예정이다.

2.2.2.2 사용자_교수



[Figure 6]

교수의 경우에도 학생의 경우와 같이 ‘213호 들어가기’ 버튼을 누른다. 이미 개설 한 강의가 있으면 강의목록 중 하나를 선택하고, 아직 개설된 강의가 없다면 두 번째 사진 맨 위에 있는 강의개설 버튼을 클릭하여 강의를 등록한다. 개설하기 버튼을 클릭하면 세 번째 사진처럼 강의 개설 창이 뜨는데, 여기에 강의명과 그 강의를 듣는 학생들의 학번을 간편하게 추가할 수 있다.

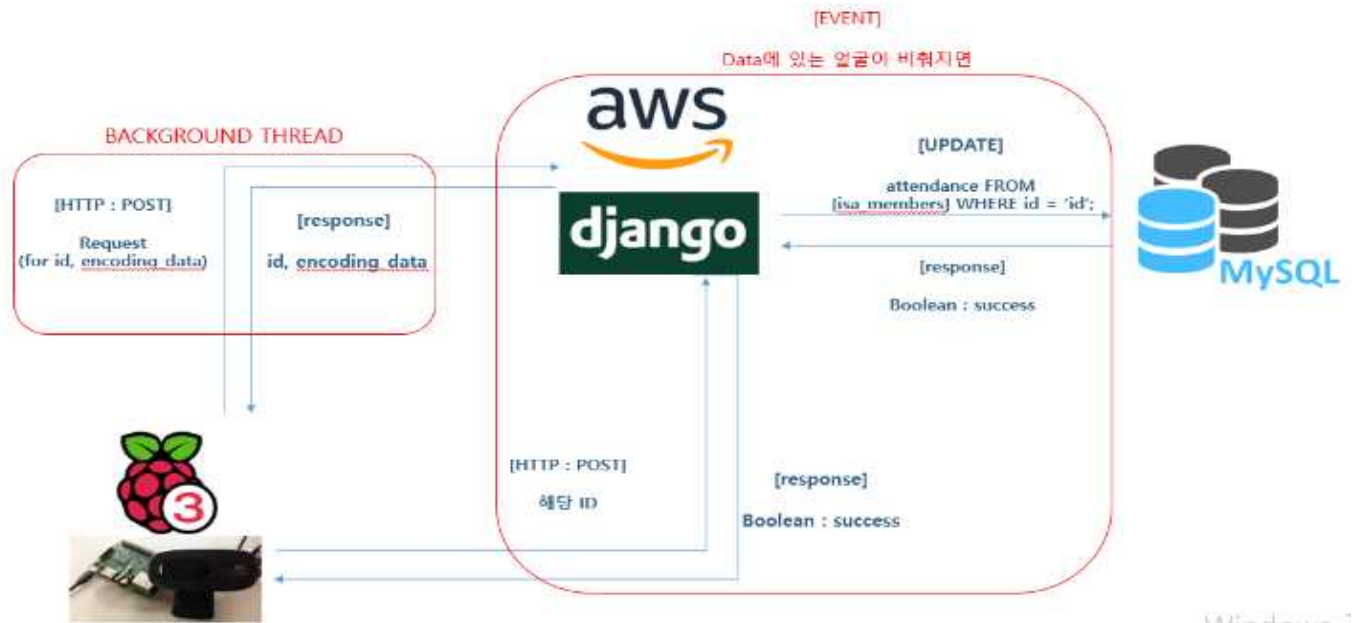


[Figure 7]

개설하기 버튼을 클릭하면 다시 이전 페이지로 돌아와서 강의 목록 중에 하나를 선택 할 수 있다. 강의를 선택 하면 그 강의를 시작하기 전에 213호에 들어오는 학생들의 얼굴을 인식하여 저장되어 있던 데이터와 비교해 일치하는 사람의 정보를 가져와 출결 확인을 한다. 학생들의 출결 여부는 두 번째 화면과 같이 뜬다.

3 시스템 구성도 및 설계

3.1 전체 시스템 구성



[Figure 8]

3.1.1- 개발환경

		상세내용
S/W	OS	Windows 7, 8, 10 , Android, 라즈비안 , Linux (Ubuntu 16.04)
	개발환경(IDE)	Android Studio, Pycharm,
	개발도구	SQL Developer, VNC Viewer , Putty ,Xshell(Xftp5) ,MySQL
	개발언어	Python3.6, JAVA
	기타사항	
프로젝트 관리환경	버전 관리	Git Hub (Source Tree)
	이슈관리	SNS (카카오톡), 문자, 전화, Team Viewer
	의사소통관리	SNS (카카오톡), 문자, 전화
	기타사항	

[Table 4]

3.1.2 - 소프트웨어

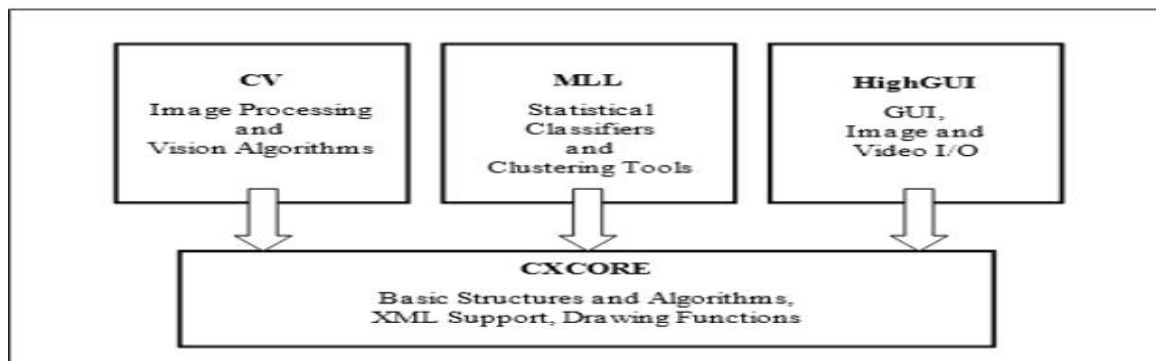
3.1.2.0 - 얼굴인식 및 선별의 과정

얼굴 분석에 앞서 들어온 정보에 대한 얼굴 찾기 위해 (HOG) 알고리즘을 사용한다. 신경망을 통해 각 얼굴의 128개의 측정값을 생성하도록 훈련시키고 복잡한 원시 데이터를 컴퓨터가 생성한 숫자 값의 목록으로 축소한다. 얼굴 분석 방법에는 여러 가지 PCA, LDA, ICA 등이 있다.

본 프로젝트에서는 PCA를 이용하여 얼굴을 분석할 것이다. 얼굴 인식에 있어서 안정적인 인식률을 얻기 위해 웹 카메라 환경에서 얼굴 영상을 입력 정보로 사용하여 향상된 인식률을 얻기 위해 영상의 사이즈를 축소하고, 밝기에 대한 정보를 정규화한 후 전처리 과정을 거쳐 얼굴 영역만을 검출할 것이다. PCA를 적용하여 고차원의 데이터에 대한 차원을 축소하여 저차원의 특징 벡터를 구한다. 구해진 특징 벡터를 이용하여 얼굴 인식을 수행한다. 얼굴과 사람 매칭 과정에서는 KNN 알고리즘을 사용한다.

3.1.2.1 - 사용되는 API 및 library 구조

①.OpenCv architecture



[Figure 9]

CV(computer Vision):구조 분석, 움직임과 추적,패턴인식,카메라 보정,영상처리,영상

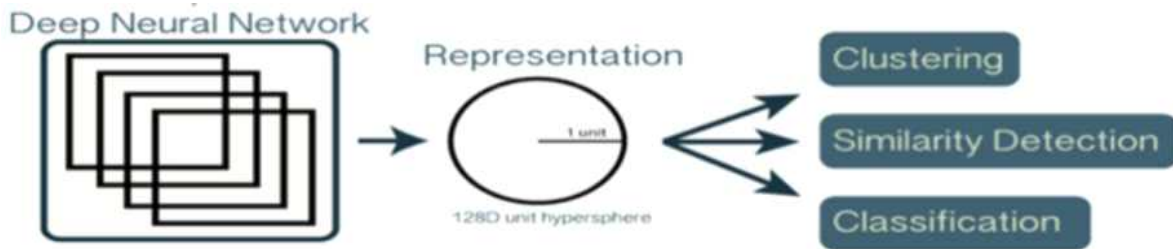
CXcore:기초 수학 관련 내용,자료구조,행렬 연산, 자료 변환, 객체 지속성

,메모리 관리, 에러처리, 코드 동적 로딩,그리기 문자열 등 담당

MLL(Machine Learning Library):자료분석,군집화(clustering),분류(classification) 담당

HighGUI:사용자 인터페이스 GUI와 영상/비디오 출력

②.간략화된 신경망 구조

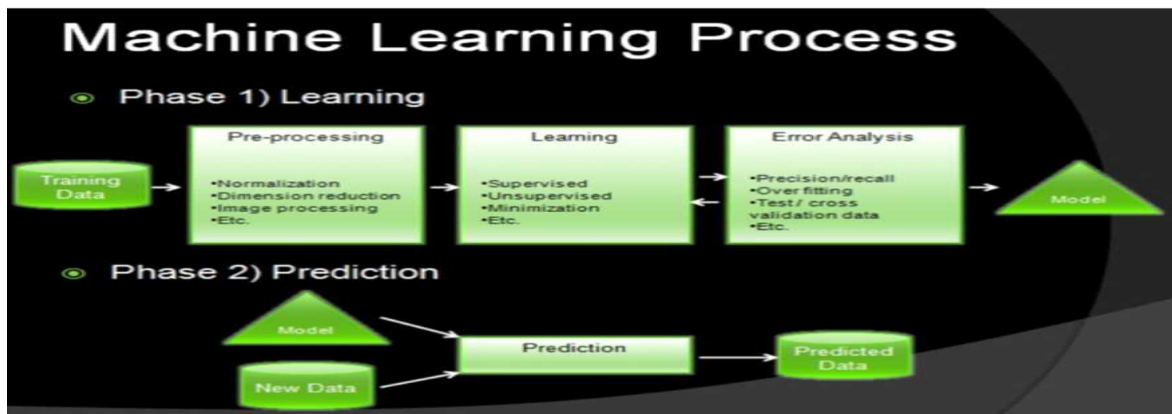


[Figure 10]

객체는 그림의 어디에서 나타나더라도 동일한 것이라는 아이디어를 활용해한다.

이미지를 중첩된 이미지 타일들로 나눈 다음에 각 이미지 타일을 작은 신경망에 제공하고 각 타일에 대한 결과를 새로운 배열에 저장한다. 따라서 커다란 이미지를 줄일 수 있다.

③.기계학습 프로세스



[Figure 11]

ML(기계학습)이란 AI(인공지능)의 한 부분으로써 컴퓨터가 학습할 수 있도록 알고리즘과 기술을 개발하는 것을 뜻한다. 컴퓨터가 스스로 패턴을 찾아내고, 새로운 분류체계를 만들어가며 데이터로 분석해 의미 있는 결과를 내놓는 일련의 과정이다. 지도학습과 비지도 학습으로 나누어지며

3.1.2.3 - API에 사용되는 수식과 수학적 개념

①. numpy 함수 활용

NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

[Figure 12]

② numpy 함수 실행속도

```
import time
import numpy as np

t1=time.time()
N=200
T=np.zeros((N+2,N+2))
for iteration in range(100):
    T[1:N+1,1:N+1]=(T[2:N+2,1:N+1]+T[0:N+0,1:N+1]\
        +T[1:N+1,2:N+2]+T[1:N+1,0:N+0]+1.)/4.
t2=time.time()
print("계산에 걸린 시간(numpy 사용):", t2-t1)

t1=time.time()
N=200
T=[[0. for i in range(N+2)] for j in range(N+2)]
for iteration in range(100):
    for i in range(1,N+1):
        for j in range(1,N+1):
            T[j][i]=(T[j+1][i]+T[j-1][i]+T[j][i+1]+T[j][i-1]+1.)/4.
t2=time.time()
print("계산에 걸린 시간(numpy 사용하지 않음):", t2-t1)
```

[그림 6] numpy 사용 유무에 따른 계산 속도 비교 프로그램

```
>>>
계산에 걸린 시간(numpy 사용): 0.07700490951538086
계산에 걸린 시간(numpy 사용하지 않음): 5.102292060852051
```

[Figure 13]

<출처:설비저널 제 45권 2016년 4월호 111page>

3.1.3.3 - 얼굴인식에 사용되는 알고리즘

HOG

2.1 Histogram of Oriented Gradient (HOG)

HOG는 Dalal [13]가 영상에서의 보행자 검출을 목적으로 제안한 특징으로서, 컴퓨터비전과 영상처리에 많이 쓰이고 있다. HOG를 만들기 위해서는 (1) 과 같이 정의된 sobel 마스크를 이용하여 입력된 영상 I 에 대해 각각 x축과 y축 gradient 이미지인 I_x, I_y 를 구한다.

$$D_x = [-1 \ 0 \ 1], D_y = [-1 \ 0 \ 1]^T \quad (1)$$

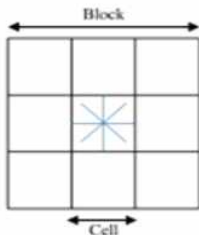
그리고 두 개의 gradient 이미지로부터 (2),(3)와 같이 각 픽셀마다의 gradient magnitude와 orientation을 구한다.

[Figure 14]

Time domain에서 sobel x을 통하여 x축방향을 미분하고 sobel y로 y축방향으로 미분한다.

I를표시한다.두개의 이미지로부터

gradient magnitude와 orientation을 구한다.



[Figure 15]

입력 이미지를 cell이라는 작은 구역으로 나누어서 cell 히스토그램을 구한다.

$$L_1-norm: f = \frac{v}{\sqrt{\|v\|_1 + e}} \quad (4)$$

$$L_2-squre: f = \sqrt{\frac{v}{\|v\|_1 + e}} \quad (5)$$

$$L_2-norm: f = \frac{v}{\sqrt{\|v\|_2^2 + e}} \quad (6)$$

[Figure16]

v는 블록안에 포함되어 있는 정규화가 되지 않은 벡터이며, e는 상수이다. $\|v\|_k$ 는 k_norm을 나타낸다. HOG 특징은 입력 영상의 모든 블록들이 히스토그램을 모아둔 정규화된 벡터이다.

종합해서 설명하면 이미지의 단순화된 버전을 만들어 주는 HOG 알고리즘을 사용해 사진을 인코딩한다.

인코딩시 단순화된 이미지를 이용해 특정부분에서 얼굴만을 뽑아 냅다.

SVM

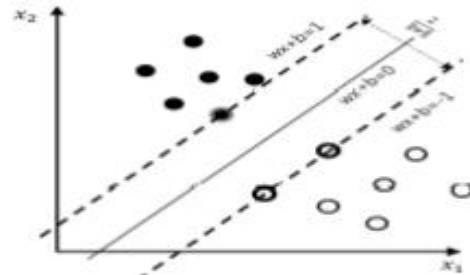


그림 2 SVM의 결정 초평면
Fig. 2. Decision Hyperplane of SVM

[Figure 17]

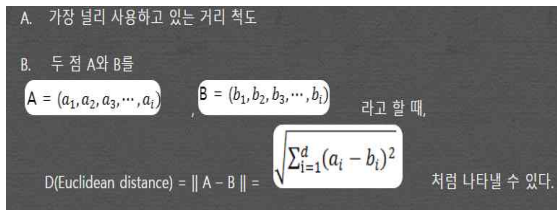
SVM의 기본원리는 그림 과 같다. 일반적인 판별 분석 방법이다. 흰색과 검은색 원이 데이터로 주어지고 두 그룹 중에서 데이터간 거리를 측정하여 중심을 구한후에 중심을 지나는 초평면을 구함으로써 흰색과 검은색 그룹을 나누는 방법을 학습하는 것이다.

KNN

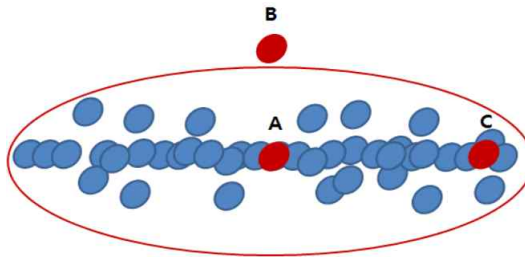
KNN 방식은 (K-nearest-Algorithm)으로 기계학습중에서 가장 간단하나 알고리즘 이다.

선행학습 없이 새로운 데이터의 요청이 올 때 분류를 수행하는 절차로 이루어진다.

이 개념은 새로 들어온 데이터가 어느 그룹에 속하는지 알아 보는 것이다. 이 때 인접한 학습 데이터의 개수가 Knn의 K를 지칭한다. K를 구하기에 앞서 표준화라는 작업이 선행 되어야 한다. 통상적으로는 유클리드 거리 개념을 이용하여 정의한다.



[Figure 18]



[Figure 19]

하지만 위 같은 경우에는 유클리드 만으로 계산하기 힘든경우가 존재한다. A를 평균이라고 할 때,유클리드 거리만으로 비교하면 B가 더 가깝지만 확률 분포를 생각해보면 확신 할 수 없다. 따라서 이 때 등장한 것이 확률 분포를 고려한 거리를 만들었다. 그것이 마할라노비스이다.

M 차원의 i번째 데이터 벡터 x_i 와 그 데이터가 속한 클래스를 y_i 라고 한다면, 여백을 최대화하는 결정 초평면을 찾는 최적화 문제는 (7)과 같이 나타낼 수 있다.

$$\begin{aligned} \min & \frac{1}{2} w^T w \\ \text{s.t. } & y_i (w x_i + b) - 1 \geq 0 \end{aligned} \quad (7)$$

(7)에서 w 과 b 는 각각 결정 초평면을 결정하는 매개변수이다. (7)은 라그랑제 승수 α 를 이용하여 다음 (8)과 같이 변환할 수가 있다.

$$L(w, b, \alpha) = \frac{\|w\|^2}{2} - \sum_{i=1}^N \alpha_i (y_i (w^T x_i + b) - 1) \quad (8)$$

KKT 조건을 이용하여 $L(w, b, \alpha)$ 를 w 와 b 에 대해 미분한 값이 0이 되는 조건을 찾으면 다음 (9)와 (10)을 구할 수 있다.

$$w = \sum_{i=1}^N \alpha_i y_i x_i \quad (9)$$

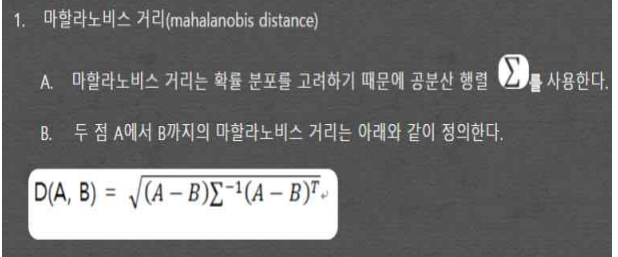
$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (10)$$

따라서 (9), (10)을 (8)에 대입을 하면, (7)의 dual 형태인 (11)을 구할 수 있다.

$$\begin{aligned} \max & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \alpha_i \alpha_j y_i y_j x_i x_j \\ \text{s.t. } & \alpha_i \geq 0 \\ & \sum_{i=1}^N \alpha_i y_i = 0 \end{aligned} \quad (11)$$

(11)과 같은 dual 형태에서 x_i 는 오직 내적형태로 밖에 나타나질 않기 때문에, (12)와 같이 벡터 내적을 커널 함수 계산으로 대체 할 수 있다.

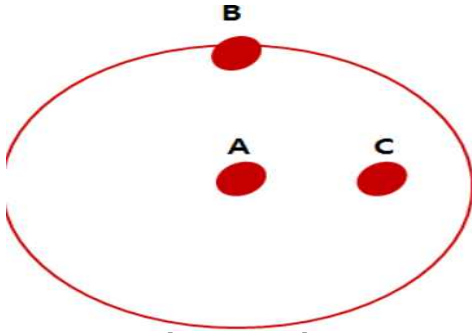
$$\begin{aligned} \max & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{s.t. } & \alpha_i \geq 0 \\ & \sum_{i=1}^N \alpha_i y_i = 0 \end{aligned} \quad (12)$$



[Figure 20]

위 그림에서 사용되는 벡터가 단위행렬일 경우, 유클리디안 거리=마할라노비스 거리가 된다. 따라서 공분산 행렬을 단위행렬로 만들면 모든 유클

유클리디안 거리=마할라노비스 거리로 측정 할 수 있다. 위와 같이 단위행렬로 만들어줄 수 있는 변환을 화이트닝 변환이라고 한다.



[Figure 21]

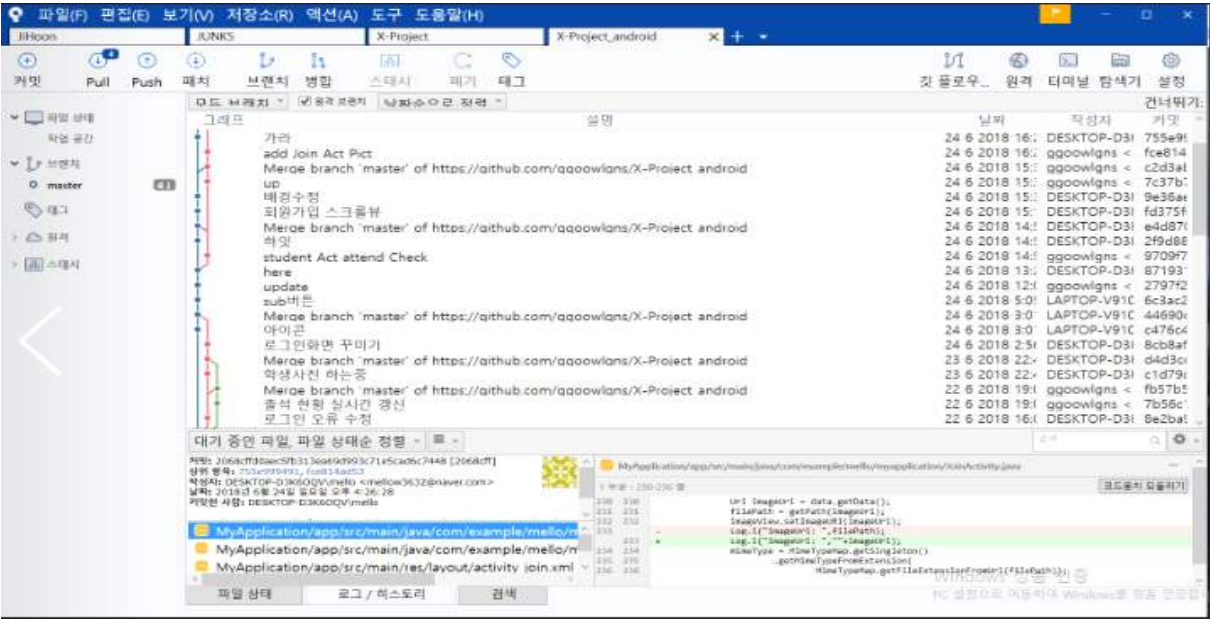
결국 그림처럼 화이트닝 변환을 통해 C가 A에 더 가까워진 것을 확인 할 수 있다.

4.프로젝트 세부일정

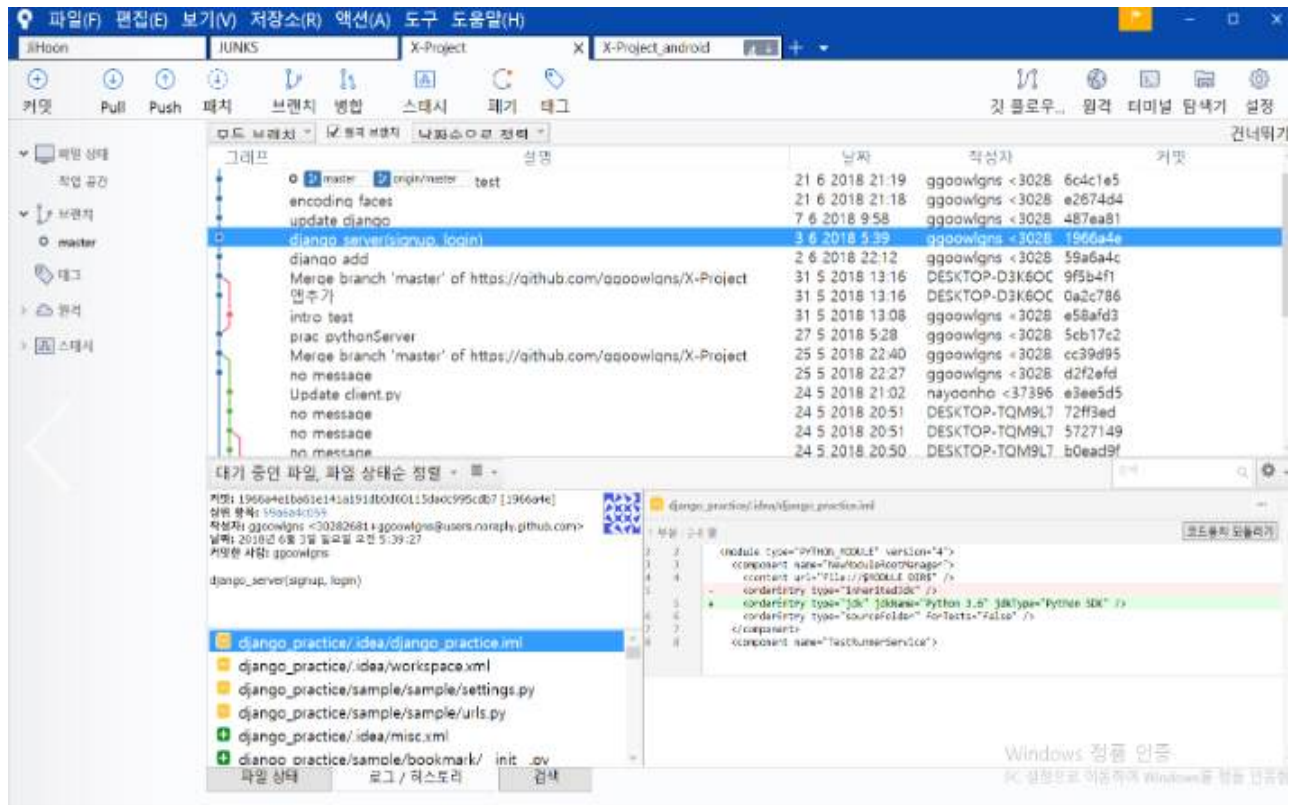
단계	개발계획	개발계획					
		1주차	2주차	3주차	4주차	5주차	6주차
기획	주제선정						
	시스템 구상						
개발	Raspberry PI (OpenCV)						
	Server & DB						
	Android App						
최종	디버깅						
	보완 및 최종결과물						

[Table 5]

5.팀원 담당 업무 및 Version Control



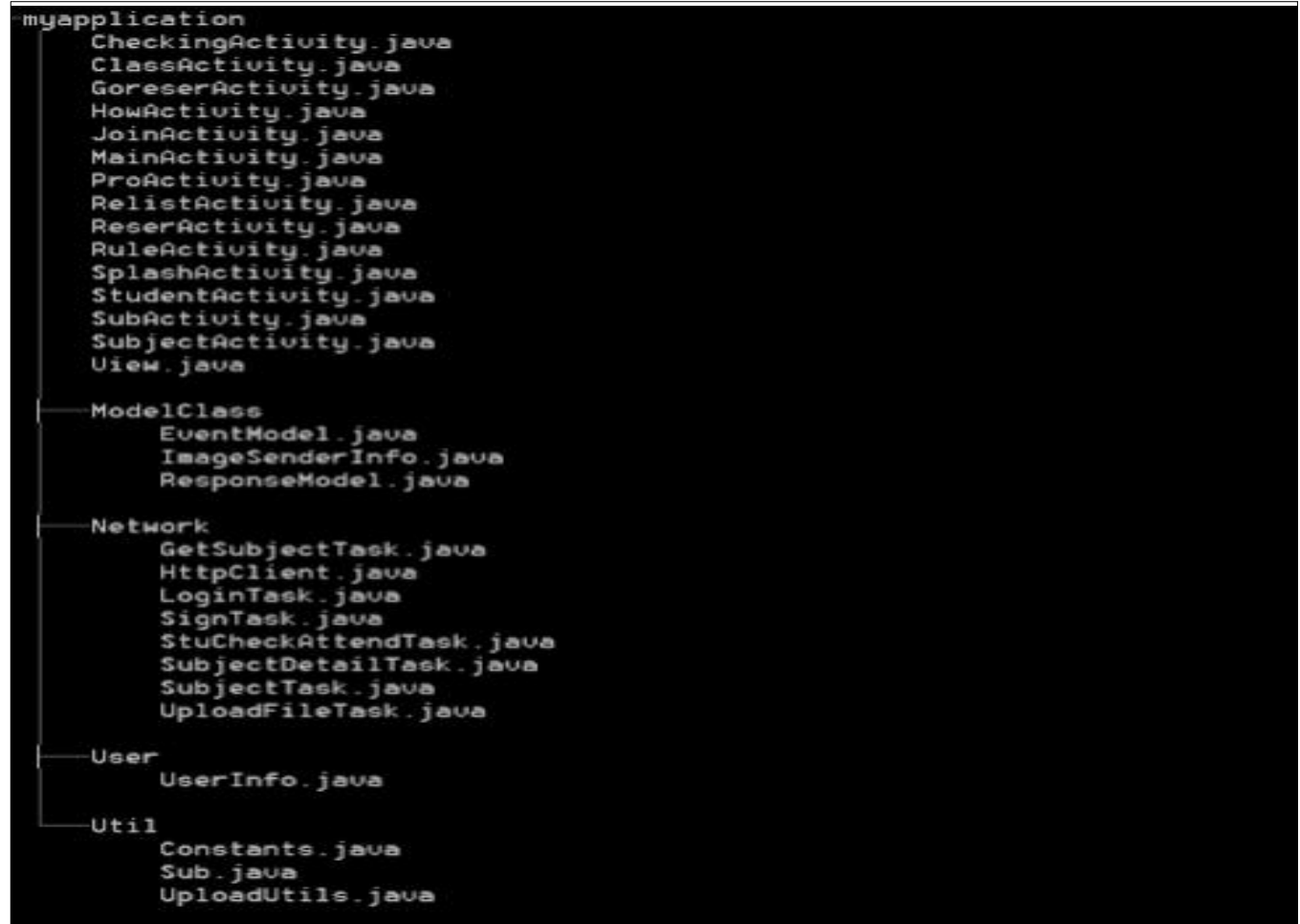
[Figure 21]



[Figure 22]

6.부록(소스코드 및 프로젝트 결론)

6.1 안드로이드



[Figure 23]

6.2 서버코드

urls.py

```
from django.contrib import admin
from django.urls import path
from . import views

urlpatterns = [
    path("", views.index),
    path("members/signup", views.signup),
    path("members/login", views.login),
    path("members/encode", views.str_encodedfile),
    path("face/get", views.give_encodingdata),
    path("photo/upload", views.upload_file),

    path("student/check_stu",views.stu_detail),

    #라즈베리파이에서 들어옴
    path("student/attend" ,views.stu_attend),
    path("student/out", views.stu_out),
    #

    path("subject/create" , views.sub_create),
    path("subject/get_subname" , views.sub_get),
    path("subject/get_subdetail" , views.sub_detail),
]
```

Views.py

```
from django.shortcuts import render,render_to_response
from django.http import HttpResponseRedirect
from .models import Members
from django.views.decorators.csrf import csrf_exempt
import MySQLdb
from django.template import RequestContext

from .forms import ImageForm
from django.urls import reverse
import json

import face_recognition
import cv2
import numpy as np
from PIL import Image

# Create your views here.

#####db연결#####
# aws_ip = "18.179.74.220"
aws_ip = "127.0.0.1"

def index(request):
    members = Members.objects.all() #해당 table의 모든 정보를 가져옴
    # no1 = Members.objects.filter(party_number = 1) #select party_number 한것만 가져옴
    str = ''
    for member in members:
        str += "id: {} <br> name: {} <br> passwd : {} <br> phone_num :  
{}".format(member.id,member.name,member.passwd, member.phone_num)

#####context만들어서 html로 보내기#####
```



```

context = {'members': members}

return render(request, 'isa/index.html',context)

@csrf_exempt
def signup(request):
    connection = MySQLdb.connect(host=aws_ip,
                                   user="root",
                                   passwd="root",
                                   db="xproj",
                                   use_unicode=True,
                                   charset="utf8")

    cursor = connection.cursor()

    if request.method == 'POST':
        id= request.POST.get('id')
        passwd = request.POST.get('passwd')
        name = request.POST.get('name')
        phone_num = request.POST.get('phone_num')
        job = request.POST.get('job')

        staff_data = [(str(id), str(passwd), str(name), str(phone_num),str(job))]
        for p in staff_data:
            format_str = """INSERT INTO isa_members (id , passwd , name ,
phone_num,job,attend, imagefile,pic_encode)
VALUES ( '{id}' , '{passwd}' , '{name}' , '{phone_num}','{job}','0','0','0');
            """

            sql_command = format_str.format(id=p[0], passwd=p[1], name=p[2],
phone_num=p[3],job=p[4])
            print("inserted")
            cursor.execute(sql_command)

        #####conection 모두 종료#####
        connection.commit()
        cursor.close()

```

```

        connection.close()

        print(id,passwd, name,phone_num)
        # return render(request,'isa/index.html', context)

    return HttpResponse()

@csrf_exempt
def login(request):
    connection = MySQLdb.connect(host=aws_ip,
                                   user="root",
                                   passwd="root",
                                   db="xproj",
                                   use_unicode=True,
                                   charset="utf8")

    cursor = connection.cursor()

    if request.method == 'POST':
        id= request.POST.get('id')
        passwd = request.POST.get('passwd')

        staff_data = [(str(id), str(passwd))]
        for p in staff_data:
            print(p[0])
            format_str = """SELECT passwd,job,name FROM isa_members WHERE id={id};
            """

            sql_command = format_str.format(id=p[0])
            login_boolean=cursor.execute(sql_command) #0 : 아이디 없을때 1 : 아이디 존재
            fetall=cursor.fetchall()
            print('login_boolean : ',login_boolean)

            if login_boolean == 1:
                if fetall[0][0] == passwd:
                    print('입력받은 password', fetall[0][0])
                    return HttpResponse(fetall[0][1]+"/"+fetall[0][2])

```

```

#####conection 모두 종료#####
connection.commit()
cursor.close()
connection.close()

return HttpResponse()

import time
@csrf_exempt
def upload_file(request):
    connection = MySQLdb.connect(host=aws_ip,
                                   user="root",
                                   passwd="root",
                                   db="xproj",
                                   use_unicode=True,
                                   charset="utf8")

    cursor = connection.cursor()

    if request.method == 'POST':
        form = ImageForm(request.POST , request.FILES)
        if form.is_valid():
            newimg = Members(imagefile = request.FILES['imagefile'])
            newimg.save()
            time.sleep(1)

            temp_path =
'/home/ubuntu/git/X-Project/django_practice/sample/media/data_analysis/uploadfile/'+str(request
.FILES['imagefile'])
            print(temp_path)
            temp_img = face_recognition.load_image_file(temp_path)
            temp_encoding = face_recognition.face_encodings(temp_img)
            print(temp_encoding[0])

            staff_data = [[str(request.FILES['imagefile']).split(".")[0], str(temp_encoding[0])]]
            for p in staff_data:

```

```

        format_str = "UPDATE isa_members set pic_encode=\'"+p[1]+\'' WHERE id
=\'"+p[0]+\';"

        print(format_str)

        sql_command = format_str
        cursor.execute(sql_command)

        print("added pic_encode")

        #####conection 모두 종료#####
        connection.commit()
        cursor.close()
        connection.close()
    return HttpResponse("pic_encode added")

@csrf_exempt
def str_encodedfile(request):
    connection = MySQLdb.connect(host=aws_ip,
                                user="root",
                                passwd="root",
                                db="xproj",
                                use_unicode=True,
                                charset="utf8")

    cursor = connection.cursor()

    if request.method == 'POST':
        staff_data = [(str(id), str(pic_encode))]
        for p in staff_data:
            format_str = "UPDATE isa_members set pic_encode='"+pic_encode+"' WHERE id =
'+id+';"

            print(format_str)

            sql_command = format_str
            cursor.execute(sql_command)

```

```

        print("added pic_encode")

        #####conection 모두 종료#####
        connection.commit()
        cursor.close()
        connection.close()
        return HttpResponse("pic_encode added")

@csrf_exempt
def give_encodingdata(request):
    connection = MySQLdb.connect(host=aws_ip,
                                   user="root",
                                   passwd="root",
                                   db="xproj",
                                   use_unicode=True,
                                   charset="utf8")

    cursor = connection.cursor()
    format_str = """SELECT id,pic_encode FROM isa_members;
    """

    sql_command = format_str
    login_boolean = cursor.execute(sql_command) # 0 : 아이디 없을때 1 : 아이디 존재
    fetall = cursor.fetchall()
    return HttpResponse(fetall)

@csrf_exempt
def stu_attend(request):
    connection = MySQLdb.connect(host=aws_ip,
                                   user="root",
                                   passwd="root",
                                   db="xproj",
                                   use_unicode=True,
                                   charset="utf8")

    cursor = connection.cursor()

    if request.method == 'POST':

```

```

        id = request.POST.get('id')
        format_str = "UPDATE isa_members set attend= \"' + str(1) + \"' WHERE id = \"' + id +
        \"';\"

        print(format_str)

        sql_command = format_str
        cursor.execute(sql_command)
        #####conection 모두 종료#####
        connection.commit()
        cursor.close()
        connection.close()

    return HttpResponse("attend changed")

@csrf_exempt
def sub_create(request):
    connection = MySQLdb.connect(host=aws_ip,
                                   user="root",
                                   passwd="root",
                                   db="xproj",
                                   use_unicode=True,
                                   charset="utf8")

    cursor = connection.cursor()

    if request.method == 'POST':
        id_num = request.POST.get('id_num')
        sub_name = request.POST.get('sub_name')
        pro_name = request.POST.get('pro_name')

        staff_data = [(str(id_num), str(sub_name), str(pro_name))]
        for p in staff_data:
            format_str = """INSERT INTO isa_subjects (id_num , sub_name , pro_name)
            VALUES ( '{id_num}' , '{sub_name}', '{pro_name}' );
            """

            sql_command = format_str.format(id_num=p[0], sub_name=p[1], pro_name=p[2])

```

```

        print("inserted")
        cursor.execute(sql_command)

        #####conection 모두 종료#####
        connection.commit()
        cursor.close()
        connection.close()

    return HttpResponse()

@csrf_exempt
def sub_get(request):
    connection = MySQLdb.connect(host=aws_ip,
                                   user="root",
                                   passwd="root",
                                   db="xproj",
                                   use_unicode=True,
                                   charset="utf8")

    cursor = connection.cursor()
    sub_names = ""
    if request.method == 'POST':
        pro_name= request.POST.get('pro_name')

        format_str = "SELECT sub_name FROM isa_subjects WHERE pro_name=\""+pro_name+"\";"
        login_boolean=cursor.execute(format_str) #0 : 아이디 없을때 1 : 아이디 존재
        fetall=cursor.fetchall()
        print('login_boolean : ',login_boolean)
        print(fetall)
        print(len(fetall))
        if login_boolean != 0:
            for i in range(0,len(fetall)):
                sub_names += (fetall[i][0]+"/")
        else:
            return HttpResponse(600)

```

```

        return HttpResponse(sub_names)
        #####conection 모두 종료#####
        connection.commit()
        cursor.close()
        connection.close()

    return HttpResponse("error")
@csrf_exempt
def stu_detail(request):
    connection = MySQLdb.connect(host=aws_ip,
                                   user="root",
                                   passwd="root",
                                   db="xproj",
                                   use_unicode=True,
                                   charset="utf8")

    cursor = connection.cursor()
    if request.method == "POST":
        stu_id = request.POST.get('id');
        format_str2 = "SELECT attend FROM isa_members WHERE id=\'" + stu_id + "\';"
        login_boolean = cursor.execute(format_str2) # 0 : 아이디 없을때 1 : 아이디 존재
        fetall2 = cursor.fetchall()
        send = fetall2[0][0]

        #####conection 모두 종료#####
        connection.commit()
        cursor.close()
        connection.close()
        return HttpResponse(send)

@csrf_exempt
def sub_detail(request):
    connection = MySQLdb.connect(host=aws_ip,
                                   user="root",
                                   passwd="root",
                                   db="xproj",
                                   use_unicode=True,
                                   charset="utf8")

```



```

cursor = connection.cursor()

if request.method == 'POST':
    sub_name = request.POST.get('sub_name')
    format_str = "SELECT id_num FROM isa_subjects WHERE sub_name=\'" + sub_name+ "\';"

    login_boolean = cursor.execute(format_str) # 0 : 아이디 없을때 1 : 아이디 존재
    fetall = cursor.fetchall()
    print('login_boolean : ', login_boolean)

    stu_names = fetall[0][0]

    send = ""
    print(stu_names.split("/")[:-1])
    for stu_name in stu_names.split("/")[:-1]:
        print("stuname: " + stu_name)
        format_str2 = "SELECT attend FROM isa_members WHERE id=\'" + stu_name + "\';"
        login_boolean = cursor.execute(format_str2) # 0 : 아이디 없을때 1 : 아이디 존재
        fetall2 = cursor.fetchall()
        print(fetall2)
        send += (stu_name+":"+fetall2[0][0]+"/")
    #####conection 모두 종료#####
    connection.commit()
    cursor.close()
    connection.close()
    return HttpResponse(send)

return HttpResponse("error")

@csrf_exempt
def stu_out(request):
    connection = MySQLdb.connect(host=aws_ip,
                                   user="root",
                                   passwd="root",
                                   db="xproj",
                                   use_unicode=True,
                                   charset="utf8")

```

```

cursor = connection.cursor()
if request.method == 'POST':
    id = request.POST.get('id')
    format_str = "UPDATE isa_members set attend= \"\" + str(0) + \"' WHERE id = \"\" + id +
\"';"
    print(format_str)

    sql_command = format_str
    cursor.execute(sql_command)
#####conection 모두 종료#####
connection.commit()
cursor.close()
connection.close()

return HttpResponse("attendance changed")

```

6.3 라즈베리 파이 코드

facerec_from_webcam_faster.py

```
import face_recognition
import cv2
import urllib.request
import requests
import numpy as np
from threading import Thread
import time

# This is a demo of running face recognition on live video from your webcam. It's a little more
# complicated than the
# other example, but it includes some basic performance tweaks to make things run a lot
# faster:
#   1. Process each video frame at 1/4 resolution (though still display it at full resolution)
#   2. Only detect faces in every other frame of video.

# PLEASE NOTE: This example requires OpenCV (the `cv2` library) to be installed only to read
# from your webcam.
# OpenCV is *not* required to use the face_recognition library. It's only required if you want to
# run this
# specific demo. If you have trouble installing it, try any of the other demos that don't require
# it instead.

# Get a reference to webcam #0 (the default one)
video_capture = cv2.VideoCapture(0)

class myThread(Thread):
    def __init__(self):
        Thread.__init__(self)
    def run(self):
        while True:
```

```

page=urllib.request.urlopen("http://18.179.74.220:8000/face/get")
response_data=page.read().decode("utf8")
##print(response_data)

#parsing data
data = response_data
temp2 = []
for temp in data.split("("):
    # print(temp)
    # print(temp.split(" ")[0])
    temp2.append(temp.split(" ")[0])

id_pic_encode = []
for temp in temp2[1:]:
    # print(temp)
    # print("id :" +temp.split(",")[0] + "          encodings :"+temp.split(",")[1])
    encodings_pre = temp.split(",")[1]
    temp4 =[]
    for en in encodings_pre.split():
        if "[" in en:
            temp4.append(en.split("[")[1])
        elif "]" in en:
            temp4.append(en.split("]")[0])
        elif "\\n" in en:
            temp4.append(en.split("\\n")[0])
        else:
            temp4.append(en)
    temp.split(",")[0].split(" ")[1]
    if "'0'" in temp4:
        continue

    temp4_arr=np.array(temp4,dtype=np.float64)
    temp4_arr = temp4_arr.astype('float64')
    print(temp4_arr)
    id_pic_encode.append([temp.split(",")[0].split(" ")[1],temp4_arr])

```

```

        #for i in id_pic_encode:
            #print(i)

        # Load a second sample picture and learn how to recognize it.

        # Create arrays of known face encodings and their names

        global known_face_ids
        global known_face_encodings

        for i in range(0,len(id_pic_encode)):
            known_face_ids.append(id_pic_encode[i][0])

        for i in range(0,len(id_pic_encode)):
            known_face_encodings.append(id_pic_encode[i][1])

        time.sleep(5)

known_face_ids = []
known_face_encodings = []

th = myThread()
th.start()

##print(known_face_encodings)
##print(known_face_ids)
##print(known_face_encodings)

# Initialize some variables
face_locations = []
face_encodings = []
face_names = []

process_this_frame = True

```

```
while True:
```

```
    # Grab a single frame of video
    ret, frame = video_capture.read()
```

```
    # Resize frame of video to 1/4 size for faster face recognition processing
    small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
```

```
    # Convert the image from BGR color (which OpenCV uses) to RGB color (which
    face_recognition uses)
    rgb_small_frame = small_frame[:, :, ::-1]
```

```
    # Only process every other frame of video to save time
    if process_this_frame:
```

```
        # Find all the faces and face encodings in the current frame of video
        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
```

```
        face_names = []
```

```
        for face_encoding in face_encodings:
```

```
            # See if the face is a match for the known face(s)
```

```
            matches = face_recognition.compare_faces(known_face_encodings,
            face_encoding, tolerance=0.37)
            name = "Unknown"
```

```
            # If a match was found in known_face_encodings, just use the first one.
```

```
            if True in matches:
```

```
                first_match_index = matches.index(True)
                name = known_face_ids[first_match_index]
```

```
            #####give server attendance#####
```

```
            data_en = {'id': str(name)}
            request_makeattendance
```

```
=
```

```

requests.post("http://18.179.74.220:8000/student/out",data = data_en)

        face_names.append(name)

process_this_frame = not process_this_frame


# Display the results
for (top, right, bottom, left), name in zip(face_locations, face_names):
    # Scale back up face locations since the frame we detected in was scaled to 1/4 size
    top *= 4
    right *= 4
    bottom *= 4
    left *= 4

    # Draw a box around the face
    cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

    # Draw a label with a name below the face
    cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
    font = cv2.FONT_HERSHEY_DUPLEX
    cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)

# Display the resulting image
cv2.imshow('Video', frame)

# Hit 'q' on the keyboard to quit!
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Release handle to the webcam
video_capture.release()
cv2.destroyAllWindows()

```

7.프로젝트 진행하면서 느낀점

교수님들이 지정해준 커리큘럼을 따라 수업을 들으면서 1학년 때부터 배우는 전공 과목들이 어디서 어떻게 쓰이는지 궁금했습니다. 그래서 때론 제가 이 전공과목을 배우고 나면 과연 무슨 일을 하고 어디에 기여할 수 있을까 하는 고민을 했습니다. 회의감이 들던 찰나에 인터넷 응용 프로젝트를 착수하게 되었습니다.

얼굴인식이라는 주제를 선정하고 자료조사를 하던 과정에 공업수학 1,2에서 배웠던 선형대수에서의 아이겐벨류,아이겐 벡터 또한 행렬의 변환 등 코딩에서도 역시 수학적인 기초 베이스가 상당히 중요하다는 것을 깨달았습니다. 나아가 저희의 전공과목을 기반으로 서버를 연동시키고 데이터베이스를 활용하는 과정에서도 수업 이외에 필요한 것들에 대해서 스스로 찾아서 공부해야 한다는 것을 인지하였습니다. 얼굴인식이라는 다소 모호한 소재를 가지고 open API, open source를 굽어서 조합하면 될 줄 알았지만 저희의 생각보다 훨씬 더 복잡하고 개념적으로 이해해야 할 부분이 많았습니다. 인터넷 자료 외에도 국내 논문, 해외 논문을 찾아가며 자료를 검색했습니다. 프로젝트를 진행하면서 저희가 쓰던 기본의 보고서 형식이 아닌 논문에서의 Abstract에 맞춰서 쓸려고 노력했습니다. 이번 프로젝트를 계기로 주제 선정 방식, 주제 분석하는 방식 업무 분담 방식 등 훗날 저희가 실무에서 일을 할 때 필요한 요소와 형식 등에 대해 알 수 있는 좋은 기회였습니다. 프로젝트 단위로 업무를 분담하면서 실무에서 팀원으로써의 역할과 책임에 대해 깊이 배울 수 있는 시간이었습니다.

참고문헌

- [1] Rabia Jafri Hamid R. Arabnia. A Survey of Face REcognition Techniques. information Processing System, 2009, 5(2):40-42
- [2] 한국정보전자통신기술학회논문지(jkiiect)'17-12, vol. 10 No 6
2017.10.6,pp. 610-613
- [3] IEIE Transactions on Smart Processing and Computing,vol,4,Febrary 2015
pp 37-39
- [4] P.K Pisharady, P. Vadakkepat, and A. P. Loh,
"Attention based dectection and recognition of hand postures against
complex backgrounds,"International Journal of Computer Vision, vol. 101,
no.3, pp.403-406,Aug. 2012.
- [5] stephen Milborrow. Multuvvew Active Shape Models with Descriptors. South Africa, Febrary
14,2016
- [6] J.Sung, "A Unified Gradient-Based Approach for Combining ASM into AAM",
International journal of computer vision, vol.75, pp. 297-308, 2007/11/01 2007.
- [7] Journal of KIISE:Computer Systems and Theory 40(1), 2013.2,8-13(6 pages)
- [8] L. I. Smith,"A Tutorial on Principal Components Analysus
," http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf,2009.