



종합설계 2차 설계서

딥러닝을 이용한 개인정보 마스크 촬영 시스템

Personal information masking shooting system using deep learning

2016150006	김강섭	정의훈교수님
2017154031	이승준	정의훈교수님
2019150031	윤예진	정의훈교수님

이승준

목차

01 종합설계 개요

02 관련 연구 및 사례

03 시스템 수행 시나리오

04 시스템 구성도

05 시스템 모듈 상세 설계

06 개발 환경 및 개발 방법

07 데모 환경 설계

08 업무 분담

09 종합설계 수행일정

10 필요 기술 및 참고문헌

지적 사항 답변 및 피드백

Q1. 실시간처리 성능 개선 필요 / RTSP library 사용 추천

A. 실시간 처리 성능을 위해 추천 사항을 참고하여 개발 시도

Q2. 마스크처리 S/W 직접 구현 / 얼굴인식 알고리즘은 API 사용

A. 구글의 얼굴인식 인공지능 모델인 facenet구조의 사전 훈련된 모델을 이용하여 얼굴인식을 구현하고 후처리를 담당하는 마스크처리 모듈은 직접 설계 및 구현

종합설계 개요

■ 초상권 침해관련 피해 신고건수

2014~2018년 9월 **39,151건**

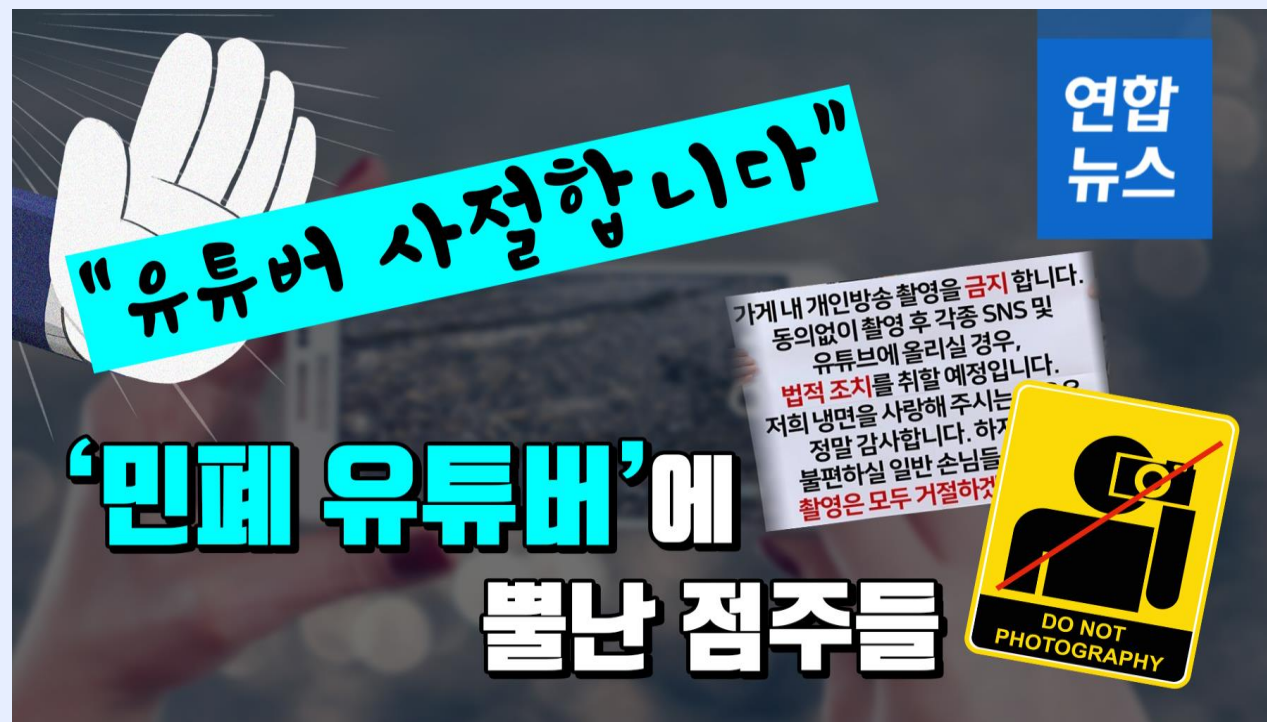
2014년 **5,017건**

2018년 (1~9월) **10,188건**

■ 유튜브 등 인터넷 방송 초상권 심의 건수

2017~2019년 5월
96건

〈자료: 방송통신심의위〉



1. 연구 개발 배경

생방송 스트리밍 중에 당사자의 동의 없이
얼굴이 노출되는 등 초상권 침해와 개인
정보가 노출되는 문제가 많이 발생

2. 연구 개발 목표

사전 등록되지 않은 얼굴을 인식하여
마스킹하는 시스템 제작, 방송 화면에 원치
않는 얼굴 노출을 방지

3. 연구 개발 효과

실시간으로 초상권 침해와 개인 정보 유출
피해를 예방

<https://www.mimint.co.kr/bbs/view.asp?strBoardID=news&bbstype=S1N12&bidx=1826601>

<https://www.yna.co.kr/view/AKR20191212155100797>

관련 연구 및 사례

I. 씨티에스의 아이마스킹

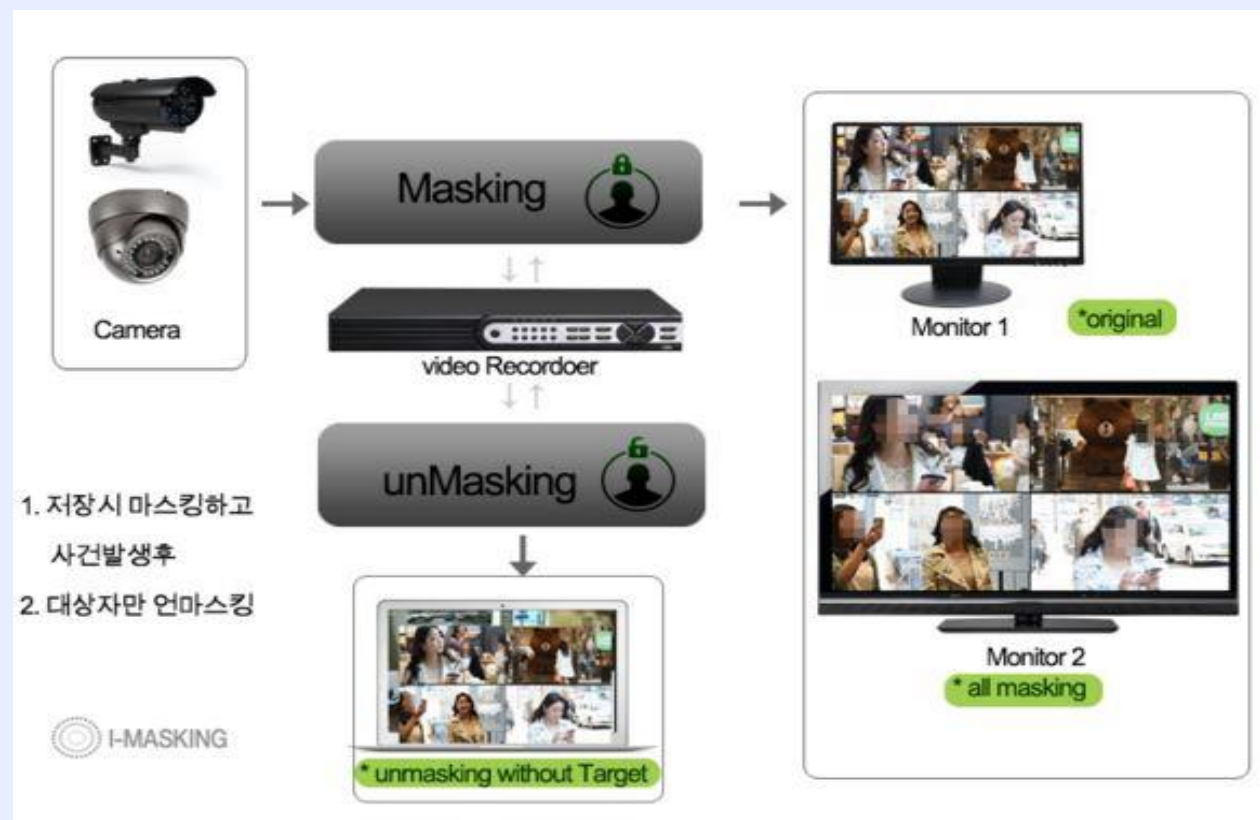


CCTV 촬영과 동시에 얼굴을 모자이크 처리
-> 아이마스킹(I-Masking) 기술 기반

개인정보 보호 취지에 맞는 영상 보안 솔루션

실시간으로 얼굴을 모자이크 처리 후 영상 저장
-> 개인정보 보호법 침해 가능성 감소

원할 경우 동의를 받아 모자이크 제거 가능



관련 연구 및 사례

II. MOMERA(모메라)

MOMERA

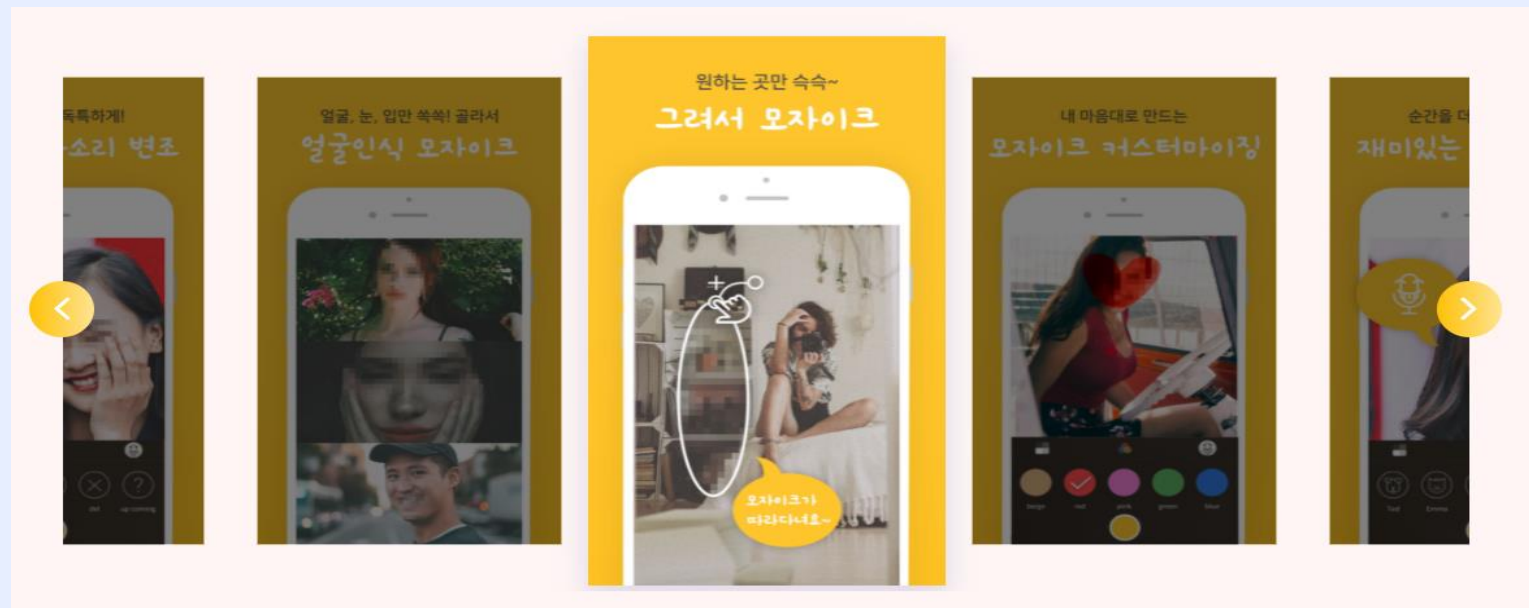


얼굴 인식 알고리즘으로 원하는 곳만
모자이크 합성되어 따라다니는
'모자이크 AR 카메라'

원하는 곳만 골라서 모자이크

디바이스의 동영상이나 사진을 불러와
모자이크 편집

독특한 모자이크 영상 공유



관련 연구 및 사례

III. 객체기반 초상권 보호 영상처리 알고리즘

객체기반 초상권 보호 영상처리 알고리즘
The Object Based Image Masking Algorithm

윤호식*, 임재혁*, 전우성**, 원치선*
*동국대학교 전자공학과
**MBC 기술 연구소

Ho-Seok Yoon*, Jae Hyuck Lim*, Woo Sung Jeon**, Chee Sun Won*

*Dept. of Electronic Eng. in Dongguk University
**MBC Technical Research Center
E-mail : cswon@cakra.dongguk.ac.kr

요약

본 논문에서는 영상 내 존재하는 의미 있는 객체단위로 초상권을 보호하는 기법을 제안한다. 제안된 방법은 초상권 보호 객체선택 단계와 객체에 마스크를 적용하는 단계 그리고 마스크가 적용된 객체를 추적하는 단계로 나누어진다. 초상권 보호 객체선택 단계에서는 블록분류(block classification) 및 워터셰드(watershed) 알고리즘을 이용하여 분할된 결과영상을 얻고 이를 이용하여 사용자가 원하는 객체를 마우스로 클릭함으로써 손쉽게 초상권 보호기법을 적용시킬 객체를 추출할 수 있다. 이렇게 정의된 객체는 다음 단계에서 마스크를 적용 받게 된다. 첫 번째 프레임에서 마스크가 적용되면 다음 프레임부터는 객체추적과정에서 연속된 화면사이의 움직임 및 밝기 정보에 의해 객체를 추적, 계속 마스크를 적용함으로써 초상권을 보호할 수 있다. 제안된 알고리즘은 초상권 보호를 위한 모자이크 처리 시 화질 저하에 따른 시청자의 화면 거부감을 최소화시키고, 반자동영상분할 알고리즘을 사용하여 객체 단위로 초상권 마스크를 적용하여 초상권 보호 대상을 놓치지 않고 추적할 수 있어 신뢰도를 높일 수 있는 장점을 가지고 있다.

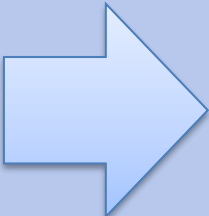
1. 서론

근래에 들어 정보매체의 발달로 말미암아 본의 아니게 개인 사생활을 침해받는 일이 많아져 개인 사생활 보호에 대한 관심이 높아지고 있다. 특히 TV나 비디오 등에서 원치 않는 신변 노출이나 공역적인 프로그램에서 특정 상표나 상품명의 노출등이 심한데, 이를 막기 위하여 방송 중에 화면에 마스크(흔히 모자이크라 불리운다)를 띄워 보호하는 기법이 많이 사용되고 있다.

현재 방송에서 사용중인 보호기법은 특정한 형태(원이나 사각형)를 갖는 일종의 마스크이다. 이는 매 프레임 단위로 사용자가 직접 위치를 지정해 주어야 하는 번거로움이 있고, 특정 객체에 적용시 객체 주변의 배경까지도 마스크로 가려져 화질의 저하로 인해 시청자가 화면에 대한 거부감을 느끼게 된다. 또한 매 프레임 단위로 직접 위치 지정이 번거로우므로 일정 시간 간격의 프레임에 대해서만 마스크를 적용하고 그 사이 프레임들은 좌우 마스크의 움직임 내삽(interpolation)을 적용하므로 객체의 움직임이 커지면 마스크 밖으로 가려져야 할 객체가 보이게 되는 경우를 종종

93

<https://scienceon.kisti.re.kr/srch/selectPORSrchArticle.do?cn=NPAP07983435&dbt=NPAP>



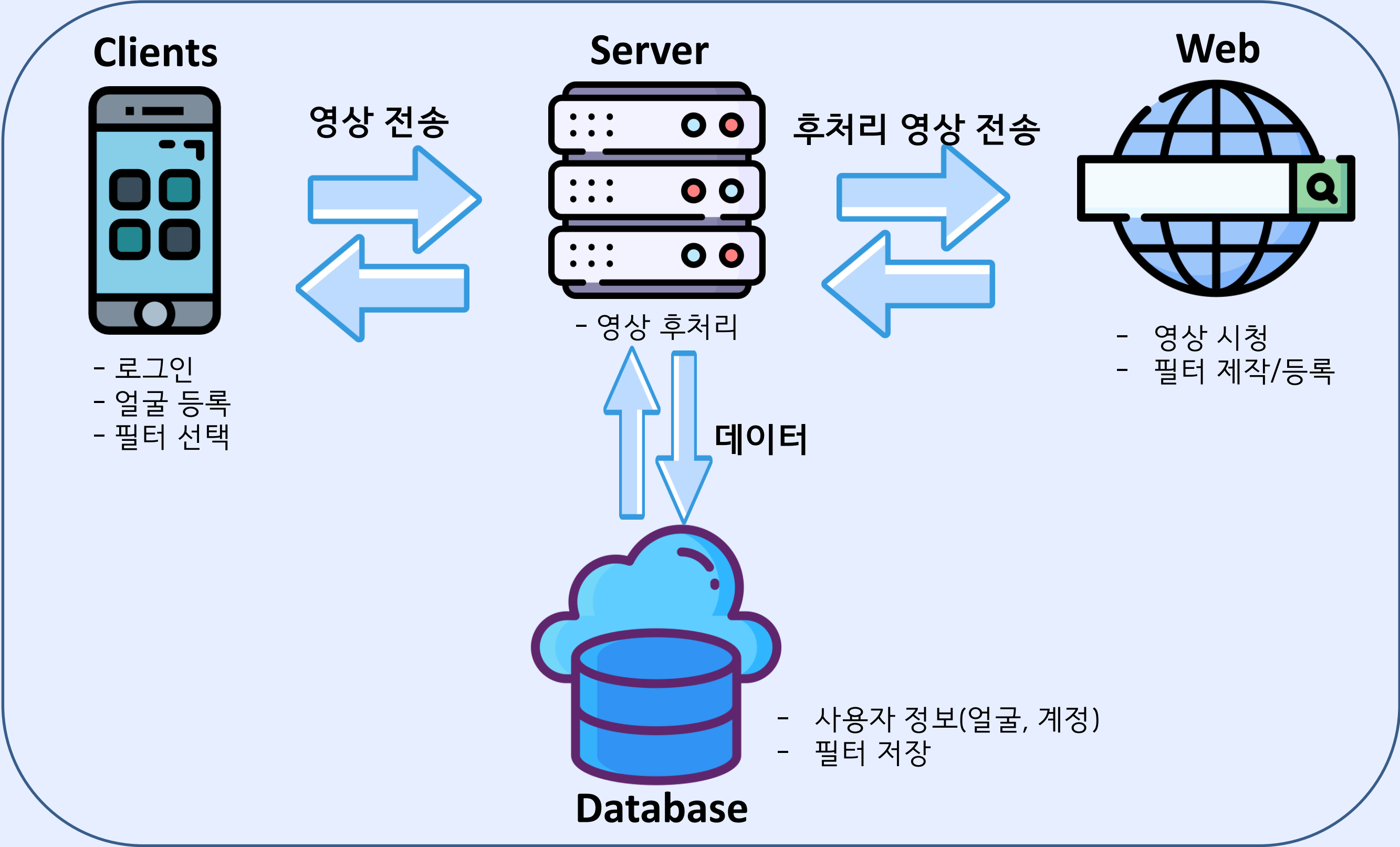
본 시스템은 FaceNet구조를 사용해 얼굴 이미지를 등록하여 생방송 스트리밍 중 모자이크 처리가 가능

영상 내 존재하는 의미 있는 객체단위로
초상권을 보호

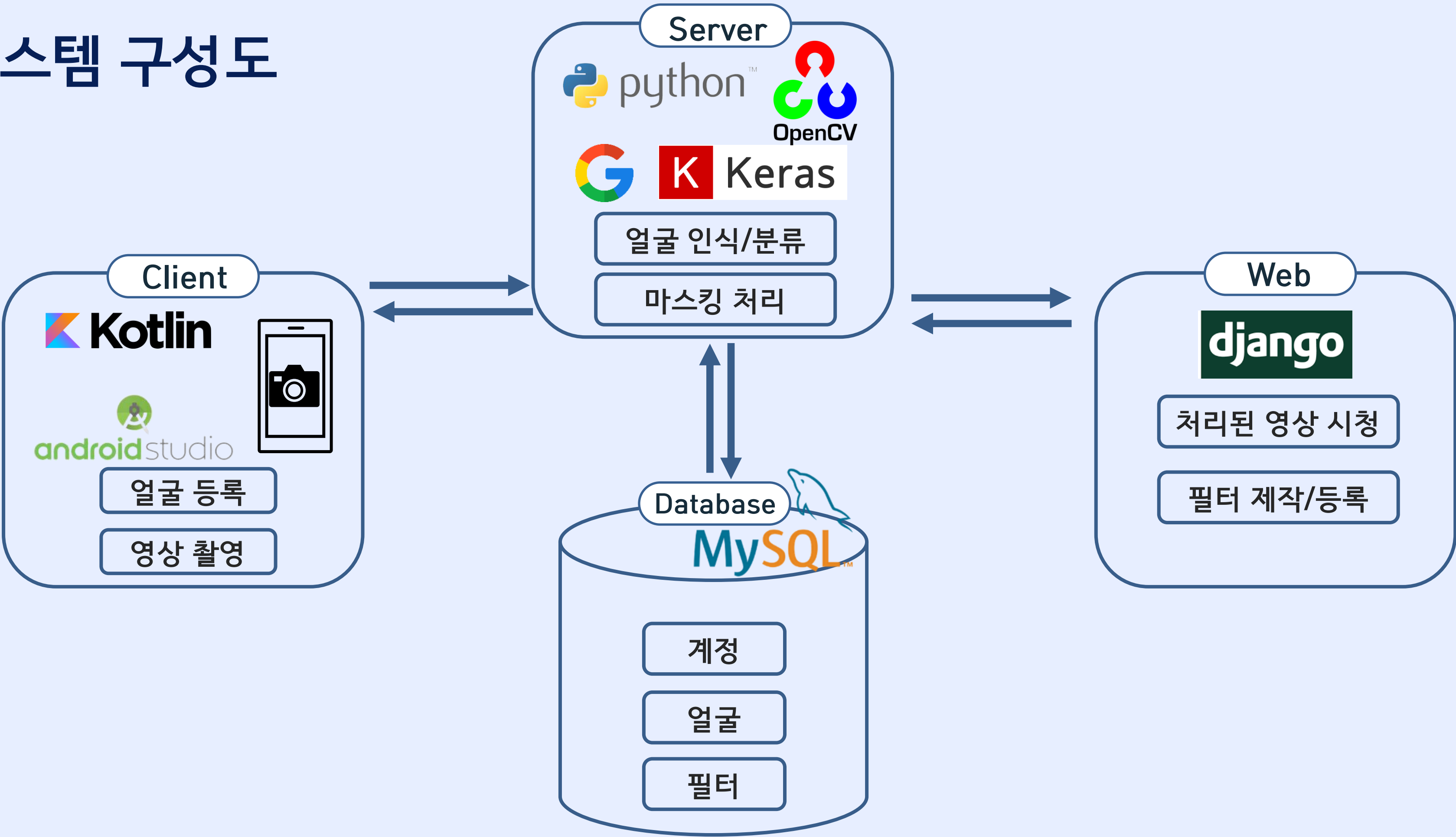
화면 내의 영상 분할은
블록단위 영상분할단계, 윤곽선 결정단계와
사용자 정의 객체생성단계로 나뉨

객체를 선택하여 지정해주면 지정된 객체의
움직임을 추적해 모자이크 처리

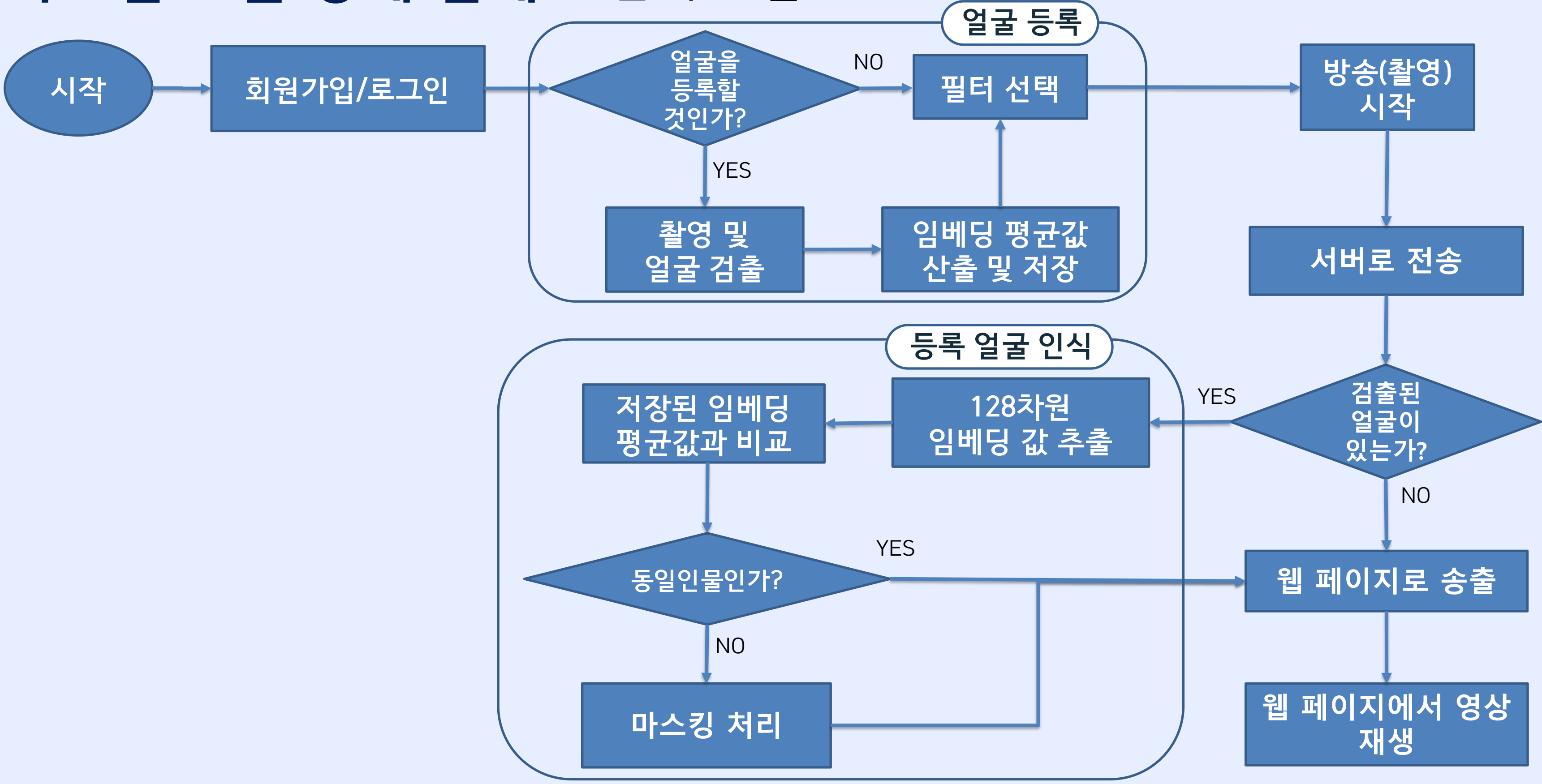
시스템 수행 시나리오



시스템 구성도



시스템 모듈 상세 설계 - 전체 흐름도



시스템 모듈 상세 설계

- DB테이블

user	
id	사용자 id
pw	사용자 패스워드
name	사용자 이름
email	사용자 email

face_vector	
id	얼굴 벡터 id
user_id	사용자 id
face_name	얼굴 주인 이름
reg_date	등록 날짜
url	얼굴 벡터 url

filter	
id	필터 id
filter_name	필터 이름
url	필터 url

user	회원 정보
face_vector	등록된 얼굴 벡터 정보
filter	등록된 필터 정보

시스템 모듈 상세 설계

- 회원가입 Register() 함수

형식	class Register : AppCompatActivity(){}
리턴 값	없음
설명	회원가입 창에서 입력한 데이터들을 변수에 저장하고 그 변수를 JSON 형 식으로 바꾸어 저장

데이터	자료형	설명
username	String	사용자 이름
email	String	사용자 이메일
id	String	사용자 아이디
password	String	사용자 비밀번호
registerForm	JSONObject	데이터 저장

시스템 모듈 상세 설계

- 로그인 Login() 함수

형식	class Login : AppCompatActivity(){}
리턴 값	없음
설명	로그인 창에서 입력한 데이터들을 변수에 저장하고 그 변수를 JSON 형식으로 바꾸어 저장

데이터	자료형	설명
userID	String	사용자 아이디
password	String	사용자 비밀번호
registerForm	JSONObject	데이터 저장

시스템 모듈 상세 설계

- 서버 통신 socket() 함수

형식	val socket = Socket(ip, port)
리턴 값	인식된 얼굴
설명	Server와 client가 특정 port를 통해 실시간으로 양방향 통신

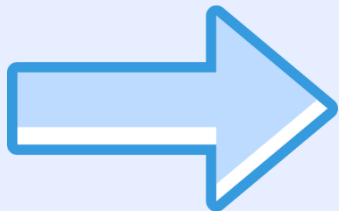
데이터	자료형	설명
client		어떤 이벤트 발생시 이벤트를 서버로 송신

시스템 모듈 상세 설계

- 얼굴 이미지에서 임베딩 값을 추출하는 calc_embs() 함수

형식	def calc_embs(imgs)
리턴 값	embs(임베딩 값 배열)
설명	전달받은 얼굴 이미지에 대해 전처리 과정을 거친 후 FaceNet모델에 전달하여 출력으로 나온 임베딩 값을 반환한다.

데이터	자료형	설명
imgs	array	전달받은 얼굴 이미지 변수
aligned_imgs	array	얼굴 이미지를 전처리한 뒤 저장
predicts	array	얼굴 이미지에 대한 모델 출력 결과 (임베딩 값 배열)
embs	array	L2 정규화 과정을 거친 predicts값을 저장



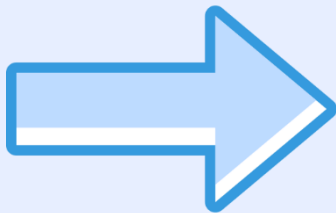
```
Embeddings = [[ 0.15712556  0.16223869 -0.07350685  0.02304433  0.06626721  0.11916567
-0.00147751  0.05570123 -0.09894355  0.00360533 -0.01800766  0.06121255
-0.08468031  0.13364638  0.01314842  0.05079055  0.19097187 -0.05928673
-0.00570582  0.06855612 -0.0720257  0.01806271  0.0992079 -0.06257208
-0.05009857  0.03702778 -0.1565432  0.06363221  0.01229365  0.01634614
-0.06443868  0.01906191 -0.11676791  0.10478934  0.04679768  0.01584635
-0.05732548  0.161399 -0.01598018  0.09414192 -0.02743223  0.17681266
-0.00460508  0.06940234 -0.09390938 -0.00829129  0.0126388  0.1288662
-0.07493604  0.15105447  0.05540227  0.03488234  0.07892174  0.05620448
-0.03895964 -0.00919146 -0.00779259 -0.02431026  0.08335686  0.03825601
-0.0792567  0.00191882  0.09594572  0.11329497  0.06353781  0.0285796
-0.07939383  0.15332824  0.16976792  0.11734817 -0.10451724  0.148393
-0.06761308  0.06074035 -0.09513466  0.04770609  0.01993667 -0.08713998
-0.05509707 -0.00522115 -0.08690462 -0.0067193  0.03004554  0.10516798
-0.01525964  0.0057779  0.1335701 -0.14762495 -0.06814617  0.04127143
-0.06367216 -0.02875072  0.01192278 -0.12360761  0.01796147 -0.0075293
-0.19511141 -0.08859263 -0.08924861  0.04904627 -0.08207577  0.00701277
-0.2439793  0.14733315 -0.07584107 -0.00099172 -0.09047384 -0.11252569
-0.03913896  0.03627214  0.00087608  0.01879008  0.02023168  0.1374438
-0.04290905 -0.08673028 -0.133026 -0.05342876  0.19645713 -0.16861248
-0.08060341  0.00891395 -0.02053973 -0.12750308 -0.08088595 -0.12381773
-0.07130403 -0.03647793]]
```

(128차원 임베딩 벡터값)

시스템 모듈 상세 설계

- 얼굴 등록 capture_images() 함수

형식	def capture_images()
리턴 값	없음
설명	전달받은 이미지에서 얼굴이 검출(dnn 이용)되면 해당 위치 좌표를 이용해 학습용 이미지로 전처리(blob, crop 등)한 뒤 따로 저장한다



데이터	자료형	설명
left, right, top, bottom	int	검출된 얼굴의 x, y좌표
img	array	프레임 상에서 얼굴 부분을 뽑아내 저장
imgs	array	img 배열을 저장할 배열
detections	array	dnn으로 검출된 얼굴의 정보 (확률, 좌표 등)
blob	array	blob된 이미지 저장

시스템 모듈 상세 설계

- 임베딩 평균값을 추출하는 calc_mean_embs() 함수

형식	def calc_mean_embs()
리턴 값	없음
설명	capture_images()함수를 통해 저장된 등록 얼굴 이미지에서 FaceNet모델을 이용해 임베딩 값을 추출해내고 평균 값을 구해 저장한다.



```
embeddings = [[ 0.15712556 0.16223869 -0.07350685 0.02304433 0.06626721 0.111916567
-0.00147751 0.05570123 -0.09894355 0.00360533 -0.01800766 0.06121255
-0.08468031 0.13364638 0.01314842 0.05079955 0.19097187 -0.05928673
-0.06578582 0.06855612 -0.0720257 0.01806271 0.0992079 -0.06257208
0.05099857 0.03702778 -0.1565432 0.06363221 0.01223965 0.01634614
-0.06443868 0.0196191 -0.11676791 0.10478934 0.04679768 0.01584635
0.05732548 0.161399 -0.01598018 0.09414192 -0.02743223 0.17681266
-0.00466508 0.06940234 -0.09398938 -0.00829129 0.0126388 0.1288662
0.07493604 -0.15105447 0.05540227 0.03480234 0.01822174 0.05629448
0.03895964 -0.00919146 -0.00779259 -0.02431026 0.08335686 0.03825601
-0.0792567 0.00191882 0.0594572 0.11324897 0.06353781 0.0285796
-0.07039383 0.15332824 -0.16070792 -0.11754617 -0.10451724 0.148393
-0.06761388 0.06074035 -0.09513466 0.04778609 0.01993667 -0.08713998
-0.05599707 -0.00522115 -0.08690462 -0.0067193 0.03004554 0.10516798
-0.01525964 0.0057779 0.1335701 -0.14762495 -0.00814617 0.04127145
0.06367216 -0.02875072 0.01192278 -0.12360761 0.01796147 -0.0075293
0.19511141 -0.08859263 -0.08924861 0.04904627 -0.06207577 0.00701277
-0.2439793 0.14733315 -0.07584107 -0.00899172 -0.0047284 -0.11252569
0.03913896 0.03627214 0.00887608 0.01879808 0.02623168 0.1374438
0.04290905 -0.08673028 -0.133026 -0.05342876 0.19645713 -0.16861248
0.00668241 0.00891395 -0.02653973 -0.12750308 -0.0060595 -0.12301773
0.07138403 -0.03647793]]
```

(n개의 128차원
임베딩 벡터값)



```
embeddings = [[ 0.15712556 0.16223869 -0.07350685 0.02304433 0.06626721 0.111916567
-0.00147751 0.05570123 -0.09894355 0.00360533 -0.01800766 0.06121255
-0.08468031 0.13364638 0.01314842 0.05079955 0.19097187 -0.05928673
-0.06578582 0.06855612 -0.0720257 0.01806271 0.0992079 -0.06257208
0.05099857 0.03702778 -0.1565432 0.06363221 0.01223965 0.01634614
-0.06443868 0.0196191 -0.11676791 0.10478934 0.04679768 0.01584635
0.05732548 0.161399 -0.01598018 0.09414192 -0.02743223 0.17681266
-0.00466508 0.06940234 -0.09398938 -0.00829129 0.0126388 0.1288662
0.07493604 -0.15105447 0.05540227 0.03480234 0.01822174 0.05629448
0.03895964 -0.00919146 -0.00779259 -0.02431026 0.08335686 0.03825601
-0.0792567 0.00191882 0.0594572 0.11324897 0.06353781 0.0285796
-0.07039383 0.15332824 -0.16070792 -0.11754617 -0.10451724 0.148393
-0.06761388 0.06074035 -0.09513466 0.04778609 0.01993667 -0.08713998
-0.05599707 -0.00522115 -0.08690462 -0.0067193 0.03004554 0.10516798
-0.01525964 0.0057779 0.1335701 -0.14762495 -0.00814617 0.04127145
0.06367216 -0.02875072 0.01192278 -0.12360761 0.01796147 -0.0075293
0.19511141 -0.08859263 -0.08924861 0.04904627 -0.06207577 0.00701277
-0.2439793 0.14733315 -0.07584107 -0.00899172 -0.0047284 -0.11252569
0.03913896 0.03627214 0.00887608 0.01879808 0.02623168 0.1374438
0.04290905 -0.08673028 -0.133026 -0.05342876 0.19645713 -0.16861248
0.00668241 0.00891395 -0.02653973 -0.12750308 -0.0060595 -0.12301773
0.07138403 -0.03647793]]
```

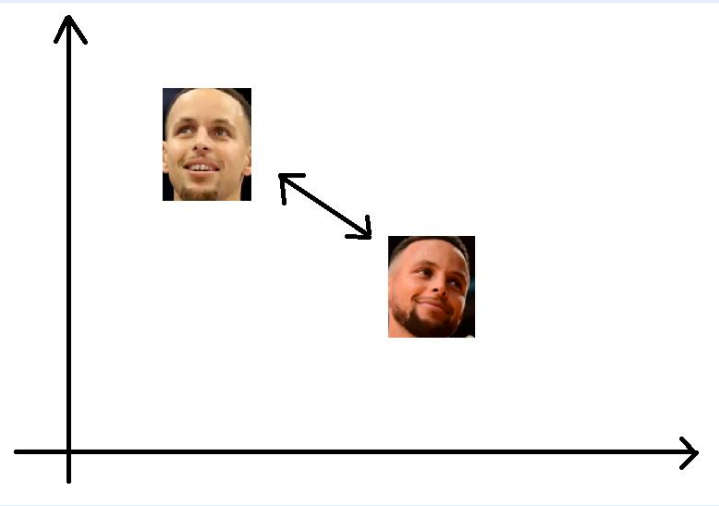
(n개 128차원 임베딩
벡터값의 평균)

데이터	자료형	설명
embs_	array	FaceNet모델을 이용해 계산된 이미지 1개의 임베딩 값을 저 장하는 배열
embs	array	캡처한 이미지 갯수만큼 embs_값을 저 장하는 배열
mean_embs	array	등록 인물 이미 지들의 임베딩 평균값을 저장하 는 배열

시스템 모듈 상세 설계

- 두 임베딩 값 사이의 유클리드 거리를 측정하는 함수
findEuclideanDistance()

형식	def findEuclideanDistance(source, dest)
리턴 값	euclidean_distance(유클리드 거리 정 수형 변수)
설명	동일 인물인지 판단하는 척도인 두 임베 딩 벡터(source, dest) 사이의 유클리드 거리를 계산하여 반환한다.

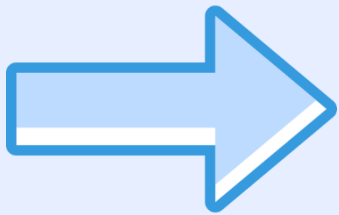


데이터	자료형	설명
source, dest	array	사전에 얼굴 이미지에 서 산출된 임베딩 벡 터값으로 비교할 얼굴 의 임베딩 벡터값과 비교대상이 되는 얼굴 (등록된 얼굴)의 임베 딩 벡터값.
euclidean_distanc e	float	Source, dest 임베딩 값 사이의 유클리드 거리를 저장

시스템 모듈 상세 설계

- 얼굴 검출 함수 detect_faces()

형식	def detect_faces(frame)
리턴 값	detected_faces_locs (검출된 얼굴의 위치좌표 배열)
설명	전달받은 프레임(frame)을 전처리하여 dnn모델을 이용해 얼굴을 검출해내고 검출된 얼굴의 위치좌표들을 저장하여 반환하는 함수



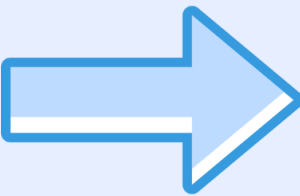
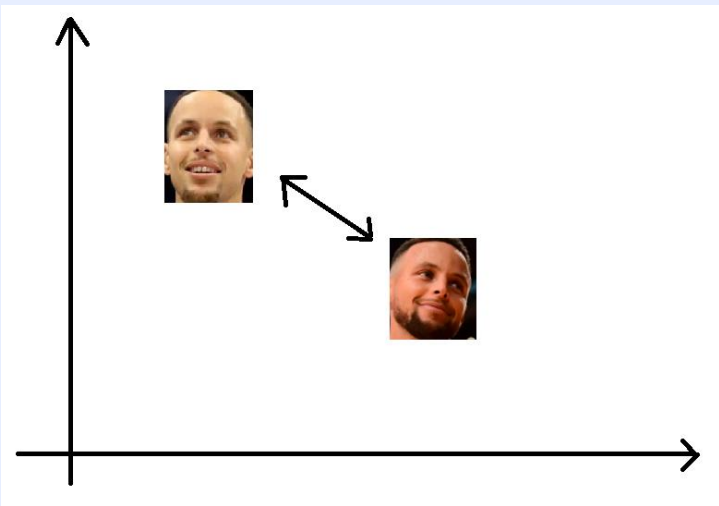
(위치 좌표)

데이터	자료형	설명
blob	array	blob된 이미지 저장
detections	array	dnn으로 검출된 얼굴의 정보 (확률, 좌표 등)
detected_faces_locs	array	검출된 얼굴들의 좌표를 저장
confidence	float	dnn이 얼굴로 예측한 확률
left, right, top, bottom	int	검출된 얼굴의 x, y좌표

시스템 모듈 상세 설계

- 얼굴 인식 함수 recognize_faces()

형식	def recognize_faces(frame, detected_faces_locs)
리턴 값	unknown_faces_locs(등록되지 않은 얼굴로 분류된 얼굴의 위치좌표 배열)
설명	검출된 얼굴들의 위치좌표를 전달받아 FaceNet모델을 이용해 임베딩 값들을 추출해내고 _findEuclideanDistance() 함수를 이용해 등록된 얼굴인지 분류하는 함수



거리와 threshold값의 크기비교

데이터	자료형	설명
left, right, top, bottom	int	검출된 얼굴의 x, y좌표
img	array	FaceNet모델의 입력 형식에 맞게 전처리된 프레임 이미지
embs	array	검출된 얼굴의 임베딩 값들을 저장
threshold	float	동일인물을 분류하는 유클리드 거리 기준값
unknown_faces_locs	array	등록되지 않은 얼굴들의 위치 좌표 저장

시스템 모듈 상세 설계

- 회원가입 signUp() 함수

형식	def signUp(request)
리턴 값	없음
설명	회원가입 페이지에서 입력한 데이터들을 이용하여 DB에 계정 생성

데이터	자료형	설명
userid	String	사용자 ID
password	String	비밀번호
username	String	사용자 이름
email	String	이메일

시스템 모듈 상세 설계

- 로그인 login() 함수

형식	def login_main(request)
리턴 값	없음
설명	로그인 페이지에서 입력한 정보와 DB에 저장된 정보를 이용하여 로그인

데이터	자료형	설명
userid	String	사용자 ID
password	String	비밀번호

시스템 모듈 상세 설계

- video() 함수

형식	def video(request)
리턴 값	StreamingHttpResponse
설명	Django에서 웹 브라우저에 영상을 스트리밍 할 수 있게 함

데이터	자료형	설명
cam	Class	비디오 캡처 클래스

시스템 모듈 상세 설계

- get_frame() 함수

형식	def get_frame()
리턴 값	jpeg.tobytes()
설명	이미지를 압축하고 크기가 조정된 메모리 버퍼에 저장, bytes형으로 반환

데이터	자료형	설명
jpeg.tobytes()	bytes	형변환된 압축 이미지
image	ndarray	이미지 프레임
frame_flip	ndarray	좌우반전된 이미지 프레임

개발 환경 및 개발 방법

I. Application

- Android Studio에서 Kotlin을 이용한 Android Application 구현
- 스마트폰 카메라를 이용해 얼굴 등록, 실시간 영상 서버로 전송
- 등록된 얼굴, 필터를 선택하는 화면 구성

II. Web

- Pycharm에서 Python Django를 이용해 구현
- 생방송 시청, 필터 등록 서비스 제공

III. Server

- AWS 등을 이용하여 서버 구축
- RTP 프로토콜을 이용한 Gstreamer 라이브러리를 이용하여 실시간 스트리밍 서버 구축
- 사전 훈련된 DNN 얼굴 검출 모델을 이용하여 실시간 영상에서 얼굴 객체 추출 및 전처리
- FaceNet 구조를 이용하여 사전 등록된 얼굴 인식 후 분류
- 선택된 필터로 미등록 얼굴 필터링 후 처리된 영상을 웹 페이지로 영상 전송

IV. DB

- MySql을 이용하여 계정 별 등록 얼굴, 필터 저장할 DB 구축

데모 환경 설계

I. Application

- 안드로이드 폰 1대와 얼굴 인식 할 사람 또는 사진 준비
- 로그인 및 회원가입이 잘 이루어지는지 확인

II. Server

- 수신하는 영상을 실시간으로 웹페이지에 송출하는지 점검
- 얼굴 검출이 제대로 되는지 확인
- 얼굴 인식 정확도 확인

III. Web

- 영상을 적은 딜레이로 수신하는지 확인
- 마스킹 처리 여부 확인
- 영상이 깨지지 않는지 확인

데모 환경 설계



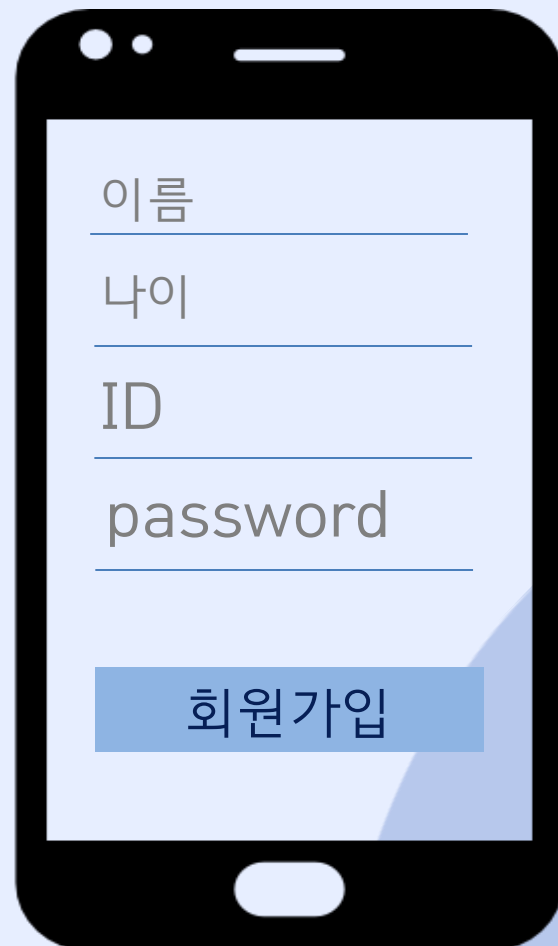
Initial login screen showing a 'LOGO' placeholder, 'ID' and 'password' input fields, a '로그인' (Login) button, and a link '회원이 아니신가요? 회원가입' (Not a member? Sign up).

회원인
경우

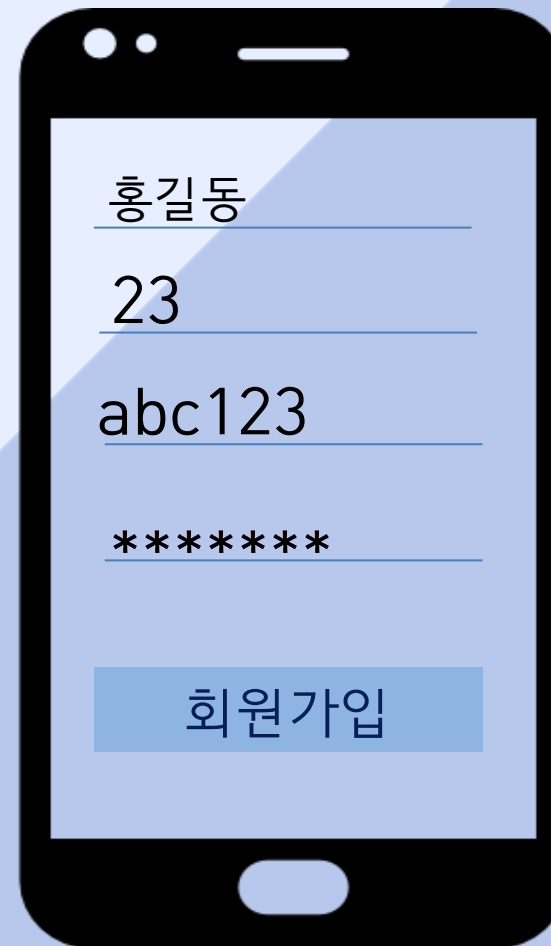


Login screen for an existing member. It shows the 'LOGO', the ID 'abc123', a masked password '*****', a '로그인' (Login) button, and the link '회원이 아니신가요? 회원가입'.

비회원인
경우



Sign-up screen for a non-member. It shows input fields for '이름' (Name), '나이' (Age), 'ID', and 'password', and a '회원가입' (Sign up) button.

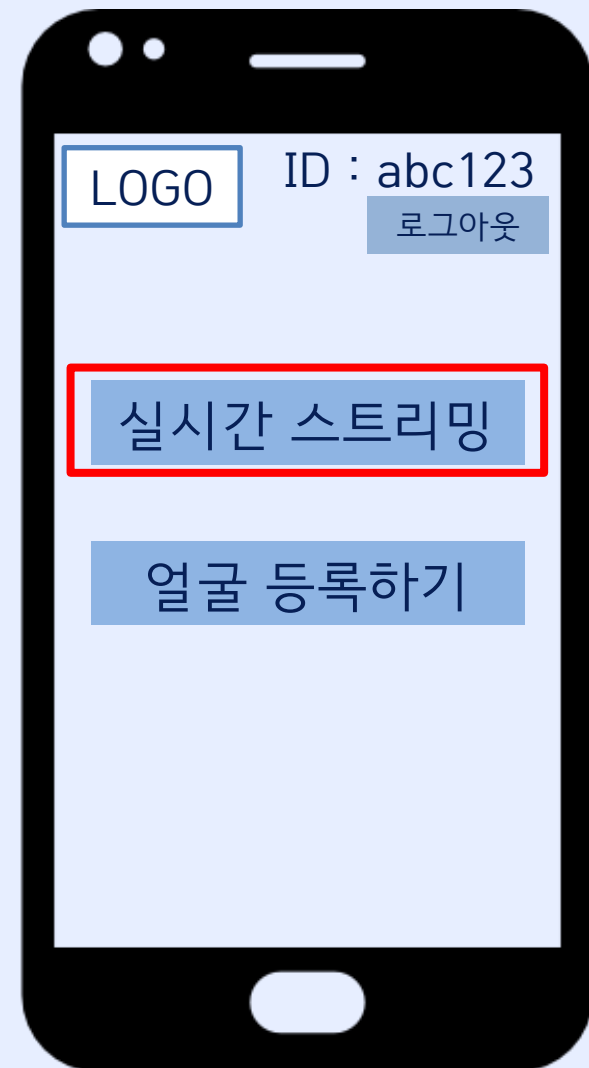


Completed sign-up screen showing the filled-in information: '홍길동' (Hong Gildong) for name, '23' for age, 'abc123' for ID, a masked password '*****', and the '회원가입' (Sign up) button.

1. 로그인 및 회원가입 시 정보가 DB에 저장됨

- 회원인 경우 : 바로 로그인
- 비회원인 경우 : 회원가입 후 로그인

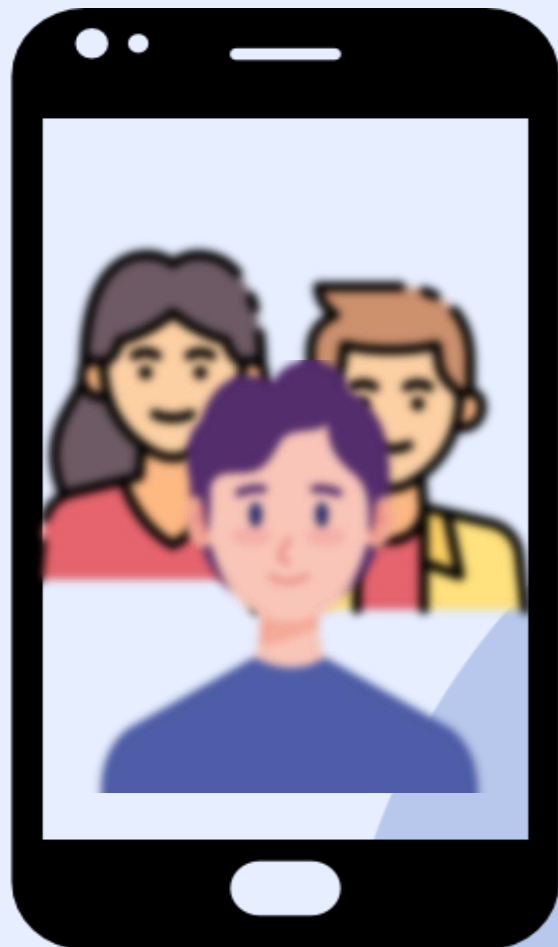
데모 환경 설계



얼굴등록이
된 경우



얼굴등록이
안된 경우



2-(1). 실시간 스트리밍을 선택한 경우

- 얼굴등록이 된 경우 : 등록된 얼굴 제외하고
나머지 얼굴 마스크 처리
- 얼굴등록이 안된 경우 : 모든 얼굴 마스크 처리

데모 환경 설계



2-(2). 얼굴 등록하기를 선택한 경우

- 직접 촬영 : 등록할 얼굴 촬영
- 사진 등록 : 갤러리에서 사진 선택

업무 분담

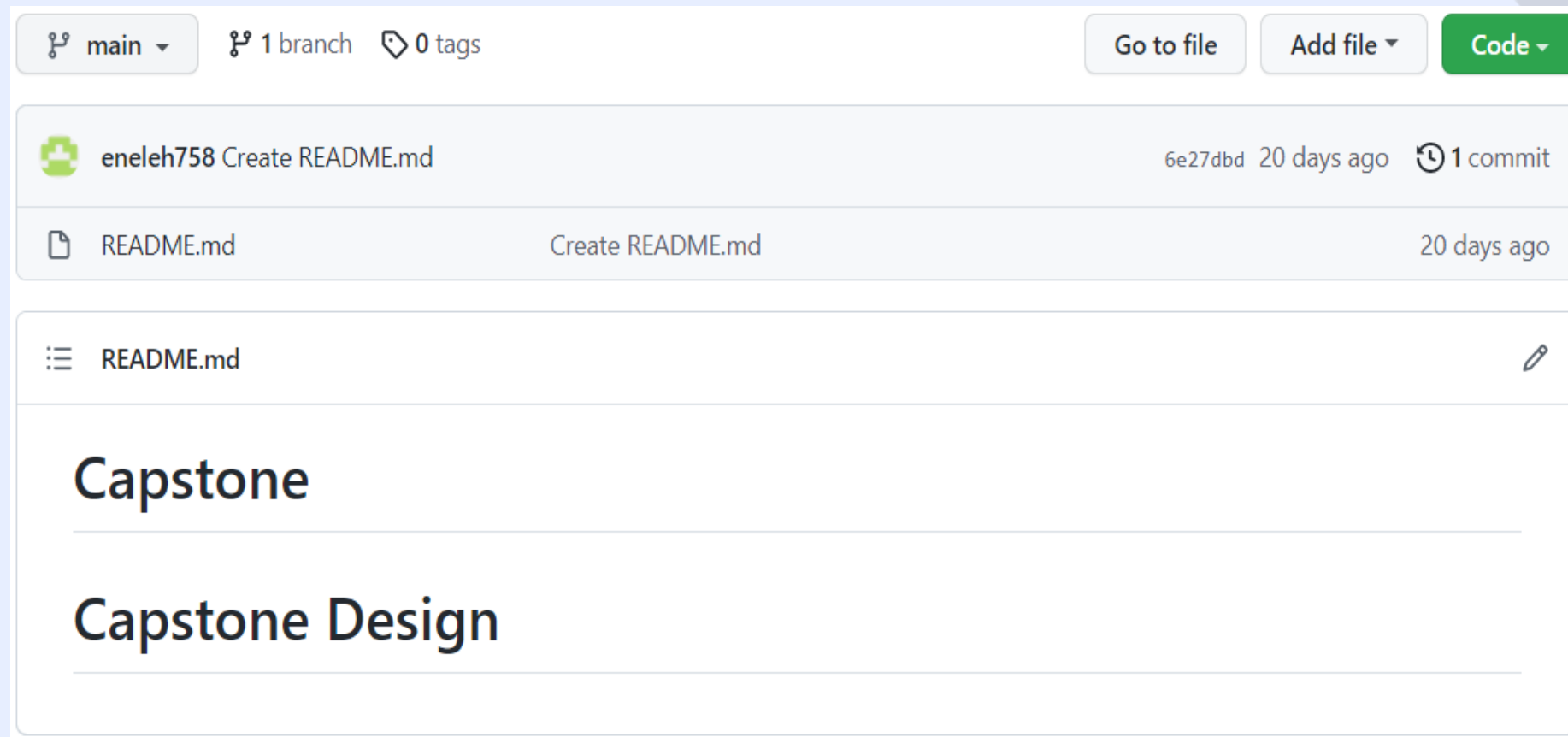
	이승준(AI, 네트워크)	김강섭(네트워크, 웹/DB)	윤예진(웹, 앱/DB)
자료수집	인공지능 모델 데이터 전/후처리 서버/클라이언트 통신	서버 구축 서버/클라이언트 통신 웹 페이지/DB연동	웹 페이지 어플리케이션 DB 연동
설 계	데이터 전/후처리 모듈 인공지능 모델 네트워크 통신	네트워크 인프라 네트워크 통신 웹 페이지	웹 페이지 어플리케이션 DB 구조
구 현	데이터 전/후처리 모듈 인공지능 모듈 네트워크 인프라 구축	웹 페이지 네트워크 인프라 구축 모듈 통합	웹 페이지 어플리케이션 DB 서버 구축, 연동
테스트	각종 버그 테스트 인공지능 모듈 정확도 테스트 실시간 영상 송출을 위한 네트워크 성능(속도) 테스트		

수행 일정

항 목	추 진 사 항	12월	01월	02월	03월	04월	05월	06월	07월	08월	09월	10월
요구사항 정의 및 분석	- 요구사항 정의 및 분석 - 요구사항 명세											
시스템설계 및 상세설계	- 시스템 설계 - 상세 설계											
구 현	- 코딩											
시험 및 데모	- 유닛 시험 - 시스템 통합시험 - 졸업작품 완전성 보강											
문서화 및 발표	- 졸업작품 중간 보고서 작성 (중간보고서, 사용자 매뉴얼 작성) - 발표(전시회, 정보과학회, 산업기술대전)											
산업기술대전	- 산업 기술대전 참가											
졸업작품 최종 보고서 작성 및 패키징	- 졸업작품 최종보고서 작성 - CD패키징 [문서, 사용법, 프로그램, 개발환경, 데모 동영상 등]											

Github 주소

<https://github.com/Shlaom/masking-shooting-system>



깃허브 ID

- 이승준 : <https://github.com/Shlaom>
- 김강섭 : <https://github.com/eneleh758>
- 윤예진 : <https://github.com/yejin603>

필요 기술 및 참고 문헌

I. FaceNet

- A Unified Embedding for Face Recognition and Clustering
- <https://github.com/davidsandberg/facenet>
- <https://arxiv.org/abs/1503.03832>

II. OpenCV

- <https://opencv.org/>

III. RTP(Real Time Transport Protocol)

IV. 기타 인공지능, 네트워크, 코틀린, Django, DB 서적



종합설계

감사합니다.