

라즈베리파이와 dlib를 이용한 얼굴인식 필터 (최종발표)

공학설계(5조) 양철관
교수님

20162911 김경빈
20162914 김민규

목차

01 1주차 2주차 요약

1. 주제 선정 이유
2. 기능 구상
3. 1주차 진행사항 요약
4. 2주차 진행사항 요약

02 3주차 진행사항

1. User Interface 구성
2. LCD 화면에 사용법 출력
3. LCD 화면 분할
4. 이메일 전송

03 3주차 실행영상

1. 상황 설정
2. 실행영상



1. 시대적 상황 반영

스마트폰과 무선 인터넷 서비스의 확장에 따른 SNS 및 카메라앱 이용자 증가
-> 다양한 카메라 기능 및 이미지 필터가 요구됨

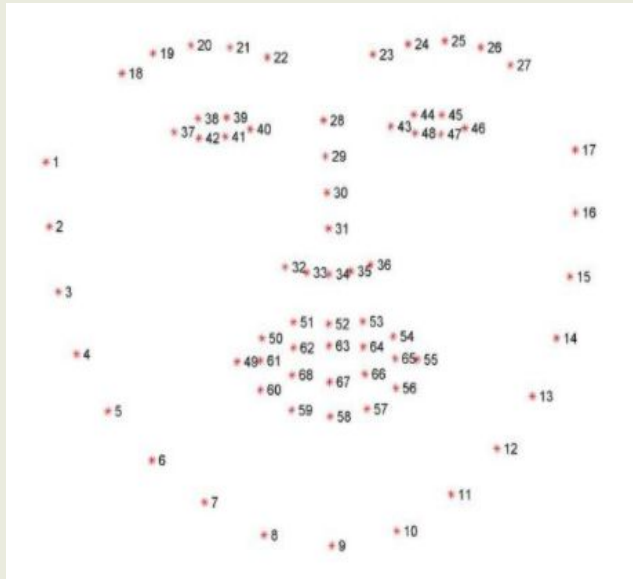
이러한 현대 요구 사항을 반영한 라즈베리파이를 이용한 얼굴인식 기능을 통해 **학교 홍보용 카메라앱**을 구현!!

01.3

1주차 진행사항 요약

dlib을 이용한 얼굴추출

미리 학습된 데이터를 통해서 아래의 사진과 같이 얼굴의 번호를
추출



```
RIGHT_EYEBROW = list(range(17, 22))  
LEFT_EYEBROW = list(range(22, 27))  
RIGHT_EYE = list(range(36, 42))  
LEFT_EYE = list(range(42, 48))  
NOSE = list(range(27, 36))  
MOUTH_OUTLINE = list(range(48, 61))  
MOUTH_INNER = list(range(61, 68))  
JAWLINE = list(range(0, 17))
```

dlib을 이용한 얼굴추출 (실행하면)



이미지 overlay 기법 사용!!

본격적인 이미지 필터를 만들기 전 실시간으로 얼굴을 추출하는 동시에 그 위에 간단한 이미지를 겹치는 과정

```
result = overlay_transparent(result,  
tmp,landmark_list[36][0],int((landmark_list[41][  
1]+landmark_list[48][1])/2))
```

보완할 사항:

학교를 좀 더 홍보할 수 있는 이미지필터의 구현이 필요함!!



01.4

2주차 진행사항 요약

01. 학교 로고



02. 학교 배경



03. 쿠민이1



04. 쿠민이2



05. 전자공학부 로고



06. 용두리



주요 사용 기법(2주차)

1주차에서 진행한 이미지 **overlay** 포함

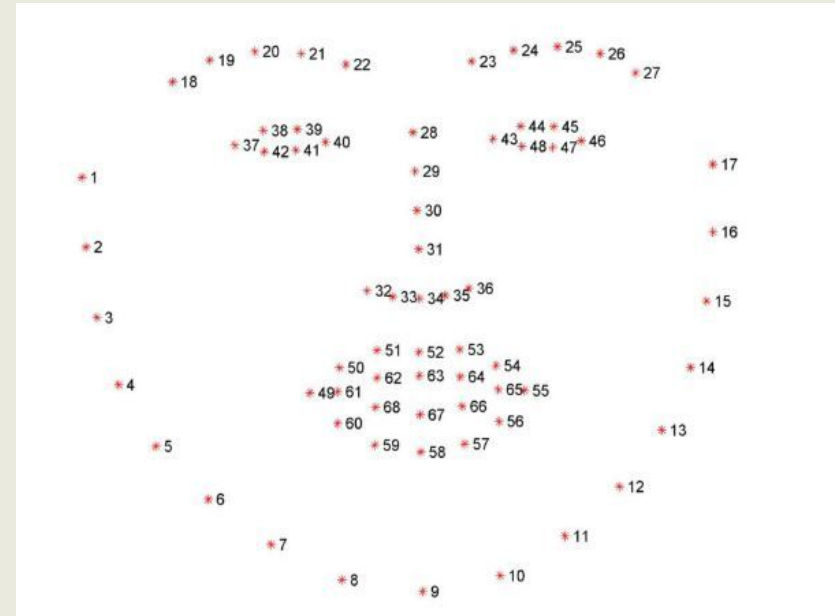
왼쪽의 사진처럼 지정된 번호(1번~27번)를 이용하여
얼굴 테두리를 이은 다각형을 뽑아내어 얼굴을 추출함

```
cv2.fillConvexPoly(feature_mask, remapped_shape[0:27],  
1)
```

```
feature_mask = feature_mask.astype(np.bool)
```

이번에는 이전에 사용하였던 각각의 이미지들이 순서대로
나타나는 이미지 필터를 구현(시간 필터링 사용)

frame 변수를 설정하여 프레임 10개가 지날 때마다
들어가는 이미지를 바꿔서 구현



02.1

3주차 진행사항 (User interface)

User interface 개발 및 이메일 전송

처음 접해본 사용자도 편리하게 사용할 수 있는 **user interface**의 필요!

01

사용법
설명

촬영을 시작하기 전
조작법에 대해 간략히
설명한 화면을 보여줌

02

촬영

사용자가 선택한 필터에
맞게
촬영 진행

03

사진
확인

촬영이 끝난 후 사진
확인 및 재촬영 여부
확인

04

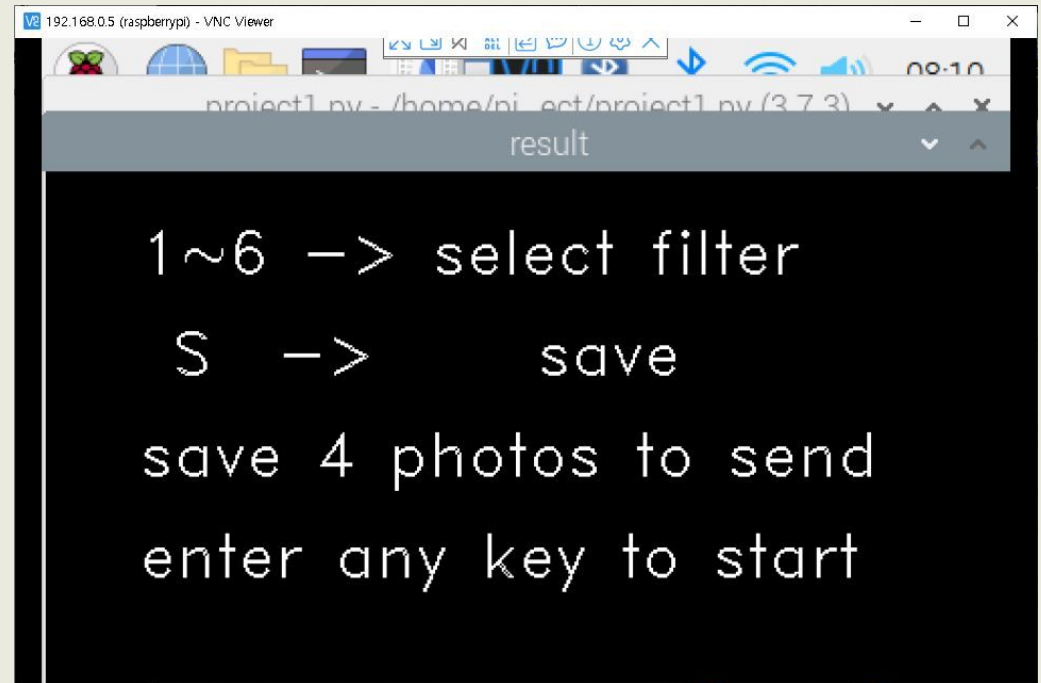
저장 및
전송

저장한 사진을 입력한
이메일로 전송

1. LCD 화면에 사용법 출력

시작 전 LCD 에 사용법 및 조작키를 설명한 화면을 출력하여 처음 접한 사용자도 편리하게 사용할 수 있도록 도와주기 위함. `cv2.putText()` 함수를 이용해서 `numpy array`에 글씨를 써서 구현했습니다.

1. 숫자키 1~6번을 클릭하여 원하는 필터를 선택
2. 필터를 선택한 후 S를 누르면 촬영 후 자동 저장
3. 만약 4개의 사진을 찍고 저장하면 촬영 종료



02.2

3주차 진행사항 (필터 선택 및 촬영 중)

2. 필터 선택 및 촬영



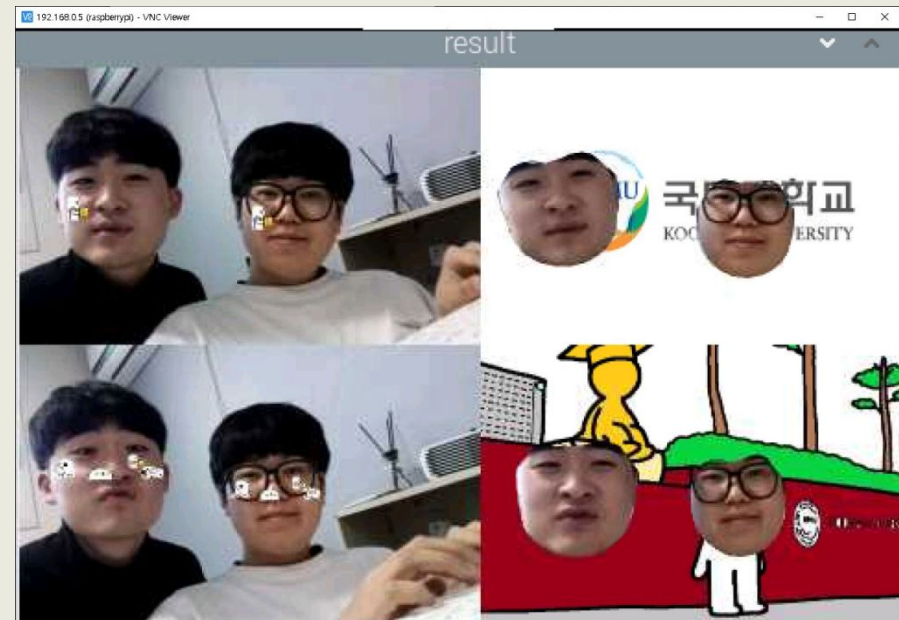
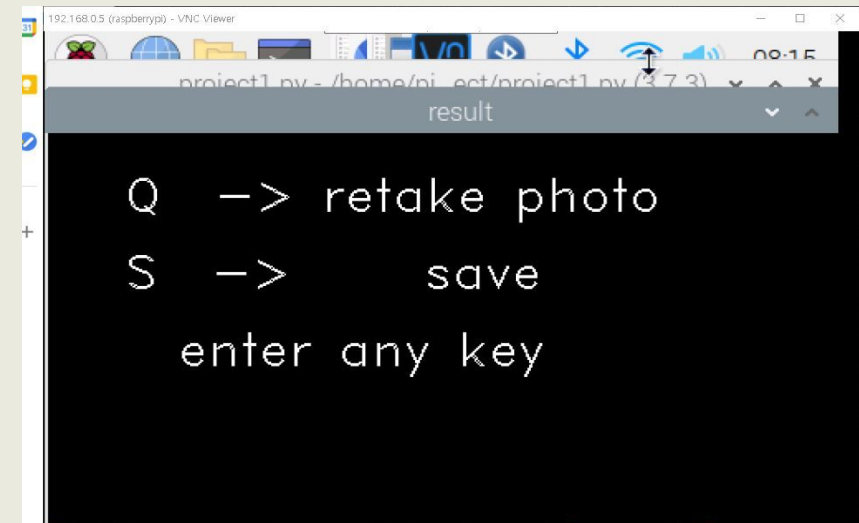
3. LCD 화면 분할

총 4장의 사진 촬영 및 저장이 된 후에는 다음과 같은 창이 띄도록 설정하여 재촬영 혹은 그대로 저장할 지의 여부를 판단

촬영한 사진 확인

촬영한 사진이 맘에 들지 않는 경우: Q 버튼 -> 재촬영

촬영한 사진이 맘에 드는 경우: S 버튼 -> 이메일 창



3. 이메일전송 (1) 라이브러리

원래 목표는 블루투스를 이용해서 사용자에게 이미지를 전송하는 것이 목표였으나, `pyhton`으로 구현하기 어려움이 있어서 이메일 보내는 것으로 바꿨습니다. `smtplib` , `email` 라이브러리를 사용해서 이메일 보냈습니다.

- (1) `kmufacefilter@gmail.com` 계정을 생성
- (2) 2차 보안을 설정한 후 앱 비밀번호를 설정
- (3) 앱 비밀번호로 `python` 코드 로그인.
- (4) 저장한 이미지를 첨부파일로 이메일 보내기

3. 이메일전송 (2) gui

Python에 내장 되어있는 `tkinter` 라이브러리를 이용해서 이메일 입력창을 만들었습니다.



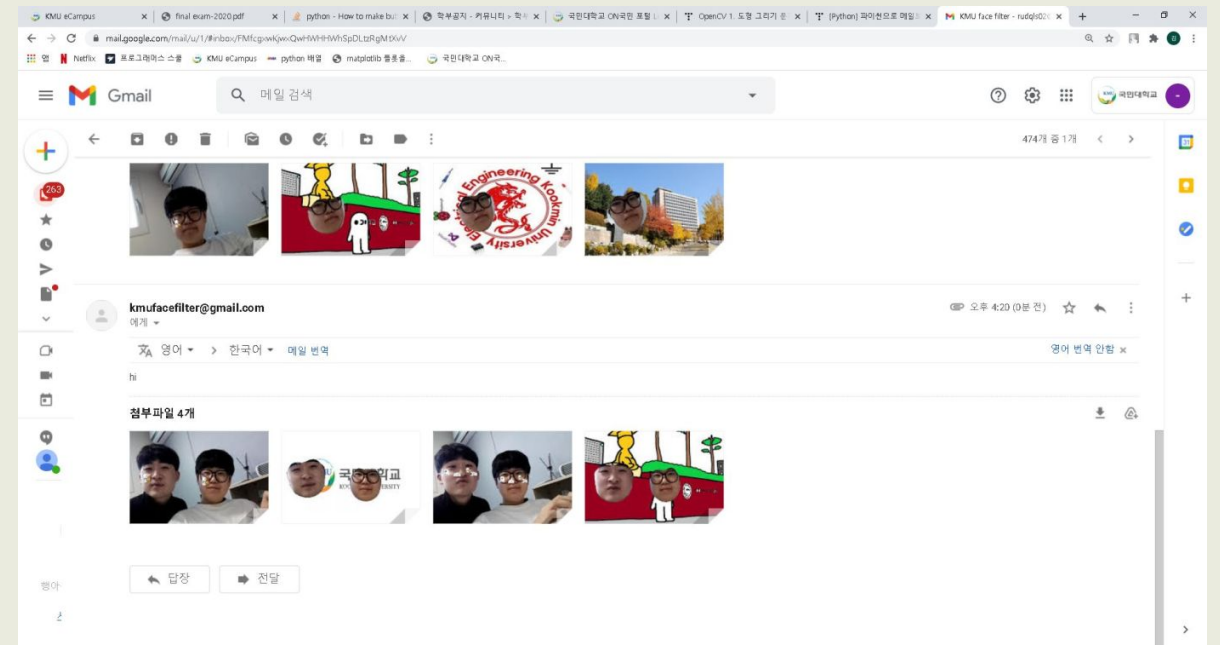
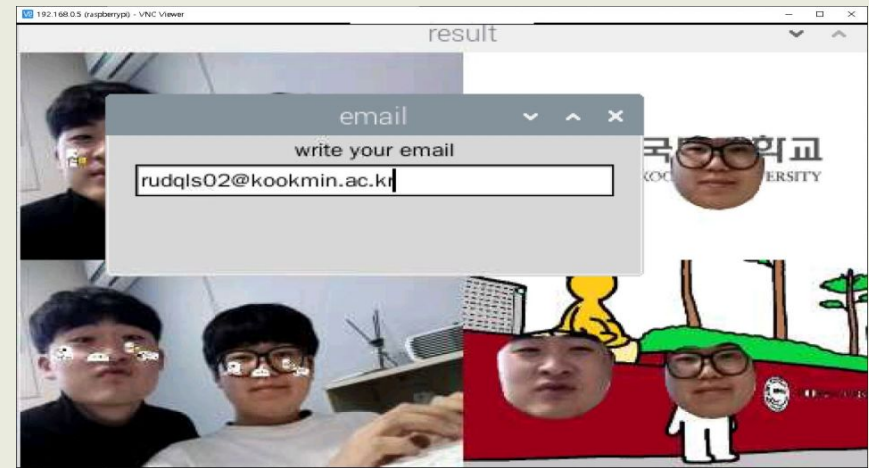
3. 이메일 전송

촬영한 사진이 맘에 들어 S 버튼을 클릭하면 다음과 같이 이메일 창이 뜨게 되고 여기에 사용자의 이메일 주소를 입력하면 해당 이메일로 저장된 사진이 발송된다.

이메일 전송 : **smtplib**, **email** 라이브러리 사용
이메일 gui : **tkinter** 라이브러리 사용

4. 이메일 확인

옆의 사진과 같이 해당 이메일에 촬영한 사진이 발송된 것을 확인 가능하다.



1. 상황 설정

Ex) 학교 홍보 행사

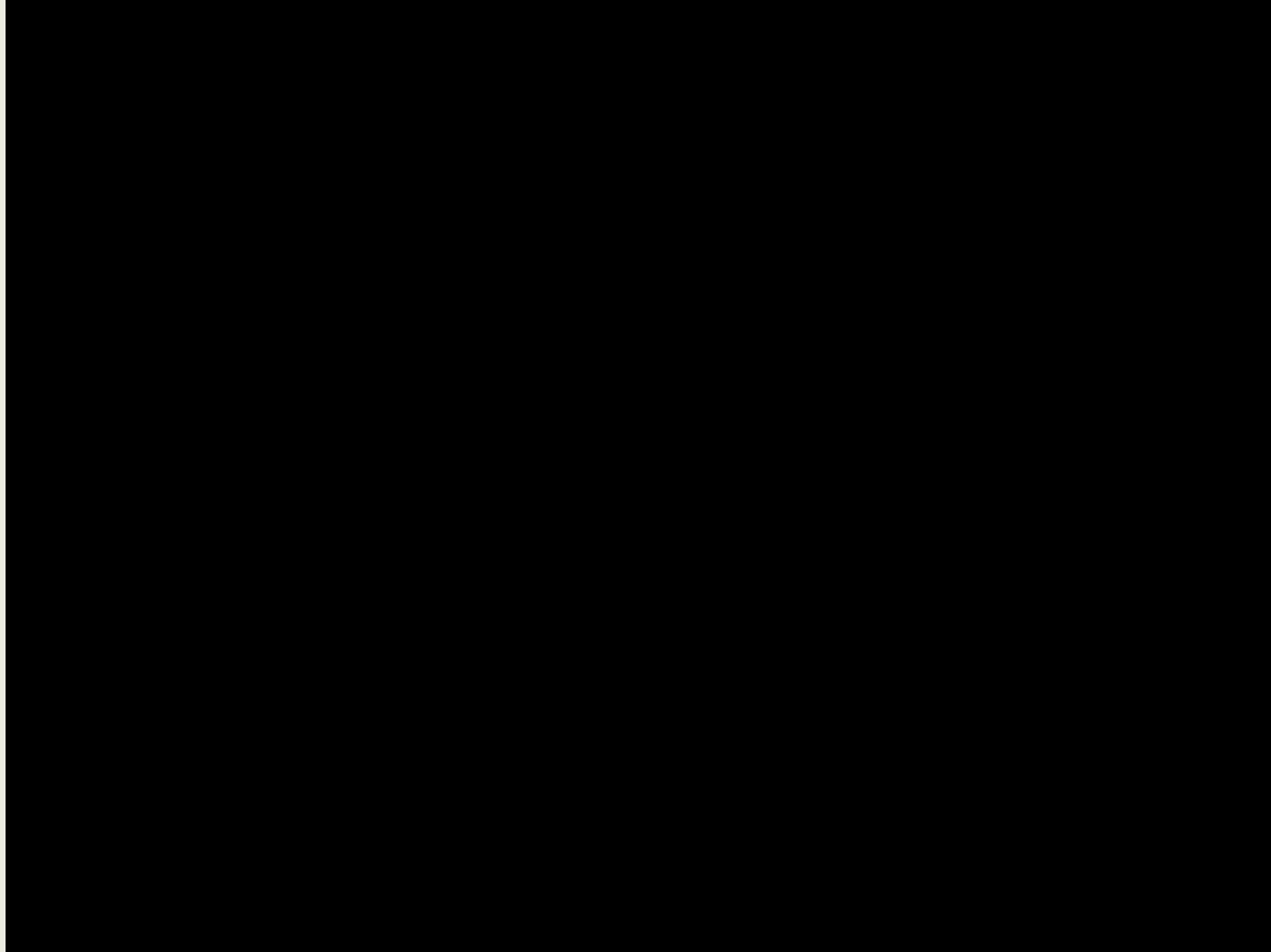
학교 홍보 행사 날 같은 경우 학교 내에 촬영 부스를 설치하여 행사 참여객들이 자유롭게 사진을 찍을 수 있도록 함

2. 기대 효과

촬영에 사용하는 이미지필터에 학교 관련 배경 및 캐릭터를 사용함으로써 사용자에게 학교 홍보 효과와 친밀감 조성을 통한 이미지 개선을 꿈꿀 수 있다.

03.2

3주차 실행영상 (실행 영상)



감사합니다