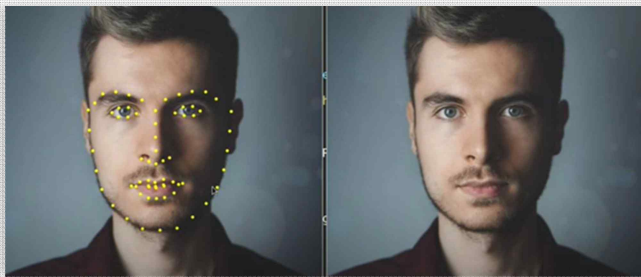


연구일시	2020년 10월 9일
연구장소	사상 투썸플레이스
수행자	류종학, 이준성, 김성훈

연구내용	
목 적	<p>1. 프로젝트 부품 작동확인</p> <p>2. 카메라 사람인식 프로그램 초안 개발</p>
연구방법	<p>1. 프로젝트 부품 작동 확인 및 불량 체크</p> <p>2. 파이썬을 이용한 카메라를 이용한 사람인식 프로그램 초안 개발</p>
도출결과	<p>1. 프로젝트 부품 작동확인 및 불량체크 결과</p> <ul style="list-style-type: none"> - Yahboom카메라 : 작동 이상 無, 컴퓨터 및 라즈베리파이와 연결하여 작동이 확인 되었습니다. - 나머지 온·습도센서, 모션인식센서는 외형에 이상이 없음을 확인하였습니다. <p>2. 파이썬을 이용하여 라즈베리파이에 들어갈 사람인식 프로그램을 파이썬으로 초안을 코딩해보았습니다.</p> <ul style="list-style-type: none"> - 인식프로그램의 실행결과 1차시부터 3차시까지 본인확인인 90%이상 성공률을 보인 반면 본인이 아닌지에 대한 인식률은 50~70%의 성공률로 아직까지는 사용자가 맞는지 틀린지에 대한 인식문제는 조금 더 해결해 나가야 할 것으로 판단이 됩니다. - 라즈베리파이 내부에서 얼굴 인식프로그램을 실행하면 성능 때문에 제대로 작동하지 않을 수도 있다고 생각이 되어 라즈베리파이로는 영상을 촬영하고 인식프로그램 자체는 컴퓨터로 처리하기로 하였습니다.
문제점 분석	<p>1. 얼굴 인식 기능에서 사용자가 아닌 대상을 사용자로 인식하는 문제점을 발견하였습니다.</p> <p>2. 카메라의 안정성 및 코드 실행의 속도가 일정하지 않은 문제 발생하였습니다.</p>
개선방안 및 향후계획	<p>1. 동영상 녹화 및 알림서비스 기능 추가예정입니다.</p> <p>2. 얼굴 인식률 개선할 예정입니다.</p> <p>3. 라즈베리파이에서 얼굴인식 코드 구현후 실행속도 시험 예정입니다.</p>

연구 노트(회의)의 붙임 자료 #1

얼굴인식
과정

#	Visual	개발사항												
	<div></div> <div><p>< 얼굴인식 과정 ></p><ol style="list-style-type: none">1. 저장된 사용자의 이미지를 불러들입니다.2. dlib 라이브러리를 사용하여 이미지를 분석하여 얼굴의 위치를 찾게 됩니다.3. 2번의 과정에서 사용자 얼굴 이미지에서 얼굴의 윤곽, 눈, 코, 입에 걸쳐 수많은 점을 찍어 좌표의 형태로 반환합니다.4. opencv 라이브러리를 이용하여 라즈베리파이 카메라영상을 불러옵니다.5. 카메라의 영상을 인코딩하여 앞의 과정과 같이 이미지를 분석하여 기존의 저장된 사용자 이미지와 대조하여 결과값을 출력합니다.</div> <div><table><tr><th></th><th>본인확인 완료</th><th>본인확인 불가</th></tr><tr><td>1차시</td><td>93.9%</td><td>69.0%</td></tr><tr><td>2차시</td><td>94.3%</td><td>71.4%</td></tr><tr><td>3차시</td><td>92.0%</td><td>58.1%</td></tr></table></div> <div><p>< 프로그램의 얼굴인식 인식률 ></p><pre>def read_img(img_path): img = cv2.imread(img_path) img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) return img def encode_face(img): dets = detector(img) if len(dets) == 0: return np.empty(0) for k, d in enumerate(dets): shape = sp(img, d) face_descriptor = facerec.compute_face_descriptor(img, shape) return np.array(face_descriptor)</pre></div> <div><p>< 얼굴인식 주요 코드 ></p><ol style="list-style-type: none">1. img_path 경로값에 파일을 할당하여 카메라 영상과 대조하게 됩니다.2. 파이썬 opencv라이브러리 함수를 사용하여 이미지를 읽은 뒤, dlib 라이브러리에 있는 함수를 사용하여 이미지를 분석, 점의 좌표값을 반환합니다.</div>		본인확인 완료	본인확인 불가	1차시	93.9%	69.0%	2차시	94.3%	71.4%	3차시	92.0%	58.1%	<div><p>■ 파이카메라 또는 웹캠을 이용하여 촬영한 영상을 토대로 기존에 가지고 있는 사용자 이미지와 비교 및 대조하여 본인확인(인식)을 할 수 있습니다.</p></div> <div><p>문제점 및 해결방안</p><p>■ 본인확인의 인식률은 높지만 본인 확인 불가의 인식률이 낮아 상대적으로 잘못된 인증을 할 가능성이 높아져 사용자가 아닌 상대방에게도 사용자라고 할 수 있어 이부분을 개선해 나갈 예정입니다.</p></div>
	본인확인 완료	본인확인 불가												
1차시	93.9%	69.0%												
2차시	94.3%	71.4%												
3차시	92.0%	58.1%												

연구 노트(회의)의 붙임 자료 #2

#	Visual	개발사항
라즈베리파이 부품 작동확인	<div data-bbox="316 250 935 725" data-label="Image"> </div> <div data-bbox="341 763 906 797" data-label="Caption"> <p>< 라즈베리파이와 카메라 초기에 연결한 모습 ></p> </div> <div data-bbox="266 801 981 882" data-label="List-Group"> <ol style="list-style-type: none"> 1. 밝기가 많이 낮아 사물을 구별하기에 많이 안 좋은 모습. 2. 밝기를 조절하는 코드를 이용하여 밝기를 조절하였습니다. </div>	<ul style="list-style-type: none"> ■ 라즈베리파이와 카메라를 연결하여 카메라가 비추는 화면을 라즈베리파이에서 확인 중인 모습입니다. ■ 라즈베리파이에서 화면을 전송받아 컴퓨터에서 프로그램으로 확인 중인 모습입니다.
	<div data-bbox="304 916 944 1393" data-label="Image"> </div> <div data-bbox="397 1431 850 1464" data-label="Caption"> <p>< 밝기 조절 후 카메라로 찍은 모습 ></p> </div> <div data-bbox="341 1536 903 1982" data-label="Image"> </div> <div data-bbox="304 2016 941 2092" data-label="Caption"> <p>< 라즈베리파이에서 촬영한 모습을 컴퓨터로 전송하여 프로그램으로 확인 중인 모습 ></p> </div>	<div data-bbox="1101 1142 1350 1176" data-label="Section-Header"> <h3 style="text-align: center;">문제점 및 해결방안</h3> </div> <ul style="list-style-type: none"> ■ 카메라의 기본 밝기 값이 적어 프로그램적으로 보정하는 코드를 넣어 밝기 문제를 해결하였습니다. ■ 카메라가 가끔 멈추거나 fps가 떨어지는 현상이 발생하여 해결하려고 합니다.