1조 발표시작합니다.(1장)

(2장)

저희는 3주 전 빠른 OpenCV 설치 및 제어, 물품 구매 및 실험으로 팀원 모두가 조금 지쳐있고, 개천절로 인한 공휴일을 맞이하여, 지난 주는 2주 전에 기획 했던 일정에서 대폭 조정된 일정을 진행 하였습니다.

지난 주는 구글 비전 사용, 모터 테스트를 최우선 과제로 하여 여러 사이드 프로젝트 조사를 병행 하였습니다.

(3장)

모터테스트는 준비되어 있는 예제를 이용하여 복잡한 스텝모터의 회전을 제어하고, 하드웨어 상 필요한 것을 파악하는데 초점을 두었습니다.

(4장)

실험은 보조배터리 5v 2a의 전력을 가지고 RPI4와 연결된 키보드, 모니터, 마우스, 모터드라이브에 전원을 공급하며, 모터를 제어하였습니다. 모터의 정격 전력은 12v 1.67a로 rpi의 5v 내부 전압과 전류로는 다소 부족하나 이번 실험은 최대홀딩토크가 실현되지 않더라도 모터의 움직임을 확인하고 방향을 제어하는데 초점을 두었기에 낮은 전력은 무시하였습니다.

(5장)

저희가 작성한 코드는 싱글 여자 코일 방식을 사용하며, 루프가 반복될 때 마다 모터의 4개선 중 한 선에만 true값을 주고 나머지는 false값을 주도록 설계되었습니다. 저희의 예상에 따르면 이 코드를 사용하면 400번의 신호 전달 시 모터의 샤프트가 왼쪽으로 회전하고 반복이 종료되면 오른쪽으로 회전 할 것으로 예상되었습니다.

(6장)

그러나 모터는 좌좌우좌, 우좌우좌 등 알 수 없는 패턴으로 제자리에서 벗어나지 않고, 드드득 소리를 내며 제대로 작동을 하지 않았습니다.

원인을 예상 해보니 4개의 선을 아두이노와 잘못 연결하거나, 모터의 전력공급이 원할하지 않아 코일의 자력을 적은 전력의 회전력이 이겨내지 못하여 드드득 소리를 내며 제자리에서 벗어나지 못할 것으로 파악되었습니다.

(7장)

그리고 다음 날 훨씬 적은 전력을 이용하는 모터를 구매하고, 동일 환경에서 실험 해본 결과

(8장)

기존 코드에서 예상한 것 같이 한 방향으로 이동 후 400번의 신호전달 후 반대로 이동함을 확인 하였습니다. 그러나 여전히 전력이 너무 낮아 힘겹게 회전하고 사람이 정말 작은 힘으로 샤프트회전 방향으로 힘을주어야지 제대로 작동 하였습니다. 그리고 다음 실험에서는 모터에 맞는 정격전압으로 실험을 진행할 예정입니다.

(9장)

다음으로는 유니티에 대한 조사와 openCV와 연동에 관한 조사입니다. 저희는 모터를 사용하지 않을 경우 openFramework를 이용하여 3d 객체의 회전 각도를 조정하고 객체 자체를 변경하여 표정을 지닌 3d이미지를 사용자에게 주려고 하였으나, c++사용자의 부재와 오픈프레임워크의 난해한 사용법으로 어려움이 따라 차안으로 플랫폼을 변경하여 보다 친숙한 유니티를 이용하여 3d객체를 조정할 계획을 수립하였습니다.

(10장)

유니티는 3가지의 가격정책을 가졌지만 저희가 사용하기에는 무료 정책인 personal만 사용해도 충분히 프로젝트를 진행할 수 있다고 판단하였습니다.

(11장)

만약 유니티를 사용하게 되는 경우 c# 스크립트는 웹 캠을 이용하여 이미지를 전달 받고 분석하여 얼굴의 위치 정보를 제공해줘야 했기에 OpenCV 연동 가능성과 사용방법에 대해 알아보았습니다.

(12장)

결과적으로 최신 버전의 openCV는 유료구매하여 편하게 연동이 가능하지만, 무료버전의 경우 rpi에서와 같이 직접 라이브러리를 pc에 설치하여 비쥬얼 스튜디오내에서 사용 가능하도록 설정해야 합니다.

(13장)

다음으로는 RPI의 OpenCV 재설치 입니다. 기존에 설치된 openCV는 latomic라이브러리 오류로 파이선 3환경을 포기하고, 파이선 2환경만 설치하였습니다. 그러나 구글비전 조사 간 구글 클라우드가 파이선 2의 지원을 2020년 1월에 종료하여 파이선 3환경에서 설치하라는 권고에 따라 기존 설치된 openCV를 재설치 해야만 했습니다.

(14장)

별다른 오류없이 설치 되었지만 opencv 컴파일 설정 중 latomic라이브러리를 명시적으로 제거함에 따라 오류를 제거하였기에 opencv를 import한 파일을 읽거나 실행하려면 실행 전 명시적으로 latomic 라이브러리를 사용한다고 명령어에 적어야 합니다.

(15장)

마지막으로 구글 클라우드 비전을 통한 표정 인식 및 성능 테스트입니다. 실험은 황세연 학생의 맥북에서 진행되었으며, 클라우드 sdk를 직접 설치하고, 원격 이미지 테스트, 로컬 이미지 테스트 순으로 진행 하였습니다.

(16장)

원격 이미지 테스트는 구글 클라우드 버킷이라는 구글 에서 제공하는 클라우드 저장소에 직접 이미지를 넣고 파이선에서 클라우드 버킷에 저장되어 있는 이미지 path를 입력 후 해당 이미지를 분석 하는 방법입니다.

사진 속 주인공이 말하기로는 아무 감정없이 찍은 사진이라 결과 또한 별 감정 없음이라고 반환되었다고 합니다.

사진은 760 x 480급 사진이고, 4g 핫스팟 환경에서 3.03s초 소요 되었습니다.

(17장)

로컬 이미지 테스트는 내 컴퓨터에 있는 이미지를 구글 api에 패스를 제공함에 따라 이미지를 분석 후 결과를 반환하는 방법을 사용했습니다.

결과는 주관적으로 정확하게 나온 듯 하였습니다.

사진은 560 x 320급 사진이고, 4g 핫스팟 환경에서 2,03s초 소요 되었습니다.

(18장)

Opencv를 대신하여 얼굴 좌표 및 표정 분석을 한번에 하려는 시도를 하였으나, 구글 클라우드 비전은 구글 서버와 네트워크 통신을 통해 이미지 분석을 하기에 로컬에서 분석하는 opencv와 속도 면에서 큰 차이를 보인다. 구글비전을 rpi에서 사용하는 경우 반환되는 레이턴시를 고려하여 3초 혹은 4초에 한번씩 요청을 보내야 할 것으로 예상된다.

(19장)

이번 주는 RPI내부에서 구글 비전을 실제로 사용해보고 openCV와 연동할 계획을 세웠습니다. 또한 구글 어시스턴트를 조사 및 사용해보고 사이드 프로젝트로 유니티로 3d이미지를 제어하는 실험을 할 예정입니다.

하드웨어는 실제 제품 용 모터를 구매하고, 외부 전력 어댑터를 선정하여 구매할 예정입니다.