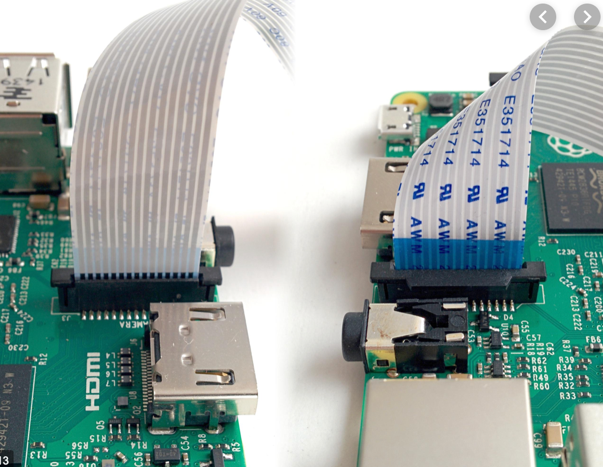
09\_27 내용정리

**파이카메라 결합**



사물의 얼굴 위치를 파악하기위해 사용되는 파이카메라는 위와 같이 리본케이블을 이용하여 라즈베리파이 본체와 연결되었다.



그러나 위의 사진처럼 리본케이블이 본체 케이스 뚜껑과 간섭을 일으켜서 제대로 닫지 못하고 강제로 닫을 경우 리본케이블이 꺾인 모습으로 사용해야 한다.

OpenCV 파이카메라 연동

테스트는 다음과 같은 순서로 진행하였다.

* 터미널에서 파이카메라를 조작하여 pi/download 경로에 test.jpg를 저장 ->
* Python2환경에서 picamera로 찍은 8메가 픽셀 사진을 detect\_multiscale로 얼굴 검출 ->
* 동일 환경에서 저장된 이미지가 아닌 py파일 실행 시 사진을 찍어 얼굴 검출->
* 파이카메라로부터 받은 영상의 1frame 사진으로 얼굴 검출 및 영상에 프레임 씌우기->
* Picamera.resolution프로퍼티를 이용하여 카메라 해상도 조절 FHD -> HD -> 640,480
* Picamera.frame프로퍼티를 이용하여 초당 프레임 조절 60 ->30 ->35
* Detect\_multiscale과 rectangle의 응답 속도와 화면 출력을 위한 시간차 조절 0.1, 0.2, 0.5, 1

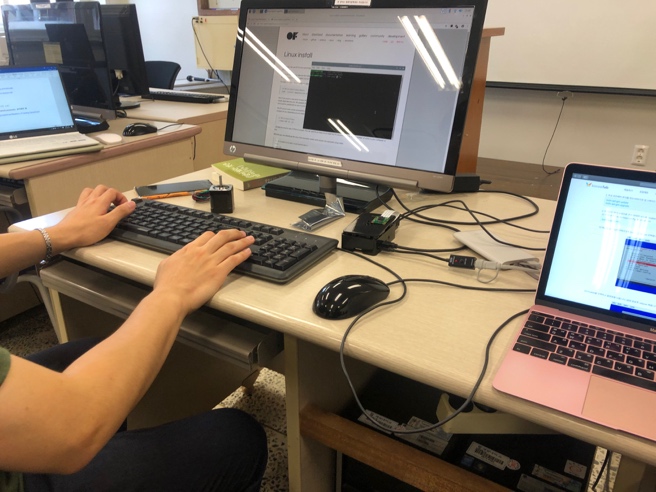
위와 같은 실험으로 우리는 사용자가 카메라를 정면으로 응시할 때 검출 확률이 증가하고 얼굴의 측면이 많이 노출될 때는 검출 확률이 감소하는 것을 알았다. 당초 예상 하였듯이 사진의 화질, 주변환경이 얼굴을 검출하는 시간에 영향을 주는 것을 확인하였고 , 카메라의 영상 화질 수는 주변환경을 고려하여 640,480으로 지정 하였다.

프레임은 얼굴 검출의 시간과 rectangle의 반응 속도 때문에 높은 프레임 수를 지정하는 것보다 적은 프레임 수를 지정하는 것이 cpu의 부하를 덜어주고 최적의 환경을 유지할 수 있게 하였다.

화면 얼굴을 따라 프레임을 씌워주는 단계까지 위와 같은 환경에서 200ms 이하를 유지 하였기에 현재 프레임 얼굴 검출과 다음 프레임 얼굴 검출과의 시간차는 0.2s로 지정하였다.

또한 얼굴 검출에는 얼굴의 크기가 일정 크기 이상부터 검출할 수 있게 지정하는 매개변수가 있었다는 것을 파악했는데, 이 min\_pixel 값을 활용하여 openCV의 성능향상을 기대할 수 있는 것으로 파악된다.

**openFrameWork**



오픈프레임워크는 이미지 , 3d , 비디오 처리를 위한 프레임워크로 모터의 움직임으로 시선을 제공하는 것을 대치하여 3d이미지가 모니터상에서 사용자에게 시선을 제공하기 위한 목적으로 사용될 것으로 예상되는 프레임워크이다.

설치는 성공하였지만, 프레임워크 내부에 있는 예제들을 실행 할 수 없었다.

에러 메시지로 x11os 는 라즈베리파이 환경에서 사용할 수 없으니 라즈베리파이 환경에서 사용할 수 있는 화면출력모드를 사용하라고 표시된다.

또 다른 예제는 libcrypto파일의 버전이 달라서 불러올 수 없다고 표시되고 있다.