중간 설계서

컴퓨터 공학과 2반

2조 (시한폭탄 조)

2020-04-28

**주제**

교차로 보행자 및 차량 검지 기술

**지도교수**

문현준 교수님, 권기학 교수님

**팀원**

14011074 조현기

15010980 구본학

15010993 조남규

15011059 이세영

목차

1. **분석**
   1. 요구사항 정의서
   2. 유스케이스 명세서
2. **설계**
   1. 클래스 설계서
   2. 컴포넌트 설계서
   3. 아키텍쳐 설계서
   4. 사용자 인터페이스 설계서
   5. 총괄시험 계획서
3. **분석**
   1. **요구사항 정의서**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **R1** | | **사용자 요구사항 정의서** | | | | | | | | | | | |
| 시스템명 | | | 보행자 및 차량 검지시스템 | | 서브시스템명 | | 교차로시스템 | | | | | | |
| 단계명 | | | 분석 | | 작성일자 | | 2020. 04. | | | 버전 | | 1.0 | |
| 요구사항ID | 요구사항 명 | | 구분 | 요구사항 설명 | 요구사항 출처 | 제약  사항 | | 중요도 | 해결방안 | | 검수기준 | | 비고 |
| SS\_IP 001 | 영상처리 | | 기능 | -카메라가 검지하고 있는 영역의 보행자를 감지하여 보행자의 위치와 수를 알 수 있어야 한다.  -카메라가 검지하고 있는 교차로 영역의 차량을 감지해야 한다 | 시한폭탄\_요구사항명세서docx.3p | 파이카메라의 영역 범위 확인 필요 | | **상** | 교차로의 전 범위가 감지되도록 카메라 범위 설정 Test 실시 | | -보행자의 위치, 수와 차량의 인식을 90% 이상 정확도를 목표로 한다. | |  |
| 성능 | -보행자와 차량의 정보에 대해 실시간으로 입력 받아 처리해야 한다.  -교차로 시스템 구현에 있어 필수 데이터이므로 정확한 인식을 해야 한다.. | 효율적인 인식 알고리즘 설계 필요 | | 오픈소스 및 패키지 활용하여, 기능 구현 | |  |
| SS\_AP 002  \_AP 002 | 인공지능 | | 기능 | -보행자의 위치로 보행자의 이동방향을 구별해야 한다.  -좌회전 대기차량, 대기차량의 수 등의 정보를 추출해야 한다. | 시한폭탄\_요구사항명세서docx.3p | 영상처리 거친 데이터 필요 | | **상** | 인식 처리가 되지 않은 데이터에 대하여 return | | -보행자의 이동방향의 정확도를 95% 이상을 목표로 한다.  -좌회전 대기차량의 구분 정확도를 99%이상을 목표로 한다 | |  |
| 성능 | -영상처리를 거친 데이터에 대해 실시간으로 입력 받아 처리해야 한다.  -교차로 신호 제어에 요구되는 데이터이므로 이동방향 및 대기상태를 정확히 파악해야 한다. | 정확한AI알고리즘 설계 필요 | | AI알고리즘과 데이터셋 학습을 통한 기능 구현 | |  |
| SS\_MS 003 | 서버 | | 기능 | -Raspberry Pi-Server와 Server-Client 간의 연결 및 파일 송신/수신 | 시한폭탄\_요구사항 명세서  docx.4p | 연구실에서만 서버구축 가능 (공간적 제약) | | **상** | 여분의 파이 서버를 활용하여 지속적인 Test 시도 및 오류 검출 | | -신호체계를 위한 클라이언트와의 응답시간을 50ms 미만을 목표로 한다. | |  |
| 성능 | -각 임베디드 기기와 서버 간의 와이파이를 이용한 무선통신이 가능해야 한다.  -각 임베디드 기기에서 들어오는 정보를 실시간으로 업데이트 해야 한다. | 데이터송수신 제한이 없을 정도의 컴퓨터 성능 필요 | | 연구실 810호 서버 사용 | |  |
| SS\_WH004 | 웹 | | 기능 | -서버에서 교차로의 보행자, 차량의 정보를 가져와야 한다.  -서버에서 얻은 정보로 신호를 적절히 조작해야 한다. | 시한폭탄\_요구사항명세서 docx.4p | 구체적인 시뮬레이션 구현 | | **상** | 유니티 프로그램을 활용하여 구현 | | -50ms 안에 서버에서 얻은 정보 표현  -신호제어는 99%의 정확도를 목표로 한다. | |  |
| 성능 | -서버로부터 실시간으로 데이터를 받아야 한다.  -가공된 정보를 활용하여 즉각적이고 정확한 신호를 제어해야 한다. | 웹 상에서 유니티 Player 구동시 Delay 발생 | | 유니티 제공 소스 활용 및 지속적인 서버, 네트워크 연결 확인 | |  |

* 1. **유스케이스 명세서**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **R2** | **유스케이스 명세서** | | | | | |
| 시스템명 | | 보행자 및 차량 검지시스템 | 서브시스템명 | 교차로시스템 | | |
| 단계명 | | 분석 | 작성일자 | 2020. 04. 28 | 버전 | 1.0 |

* + 1. 서브시스템 목록

|  |  |  |
| --- | --- | --- |
| 서브시스템 ID | 서브시스템명 | 서브시스템 설명 |
| SS\_IP\_001 | 영상처리 | 보행자 및 차량 인식을 담당하는 시스템 |
| SS\_AP\_002 | 인공지능 | 영상 처리된 정보를 AI 분석, 추출하는 시스템 |
| SS\_MS\_003 | 메인 서버 | 임베디드 기기간 연결 및 파일 관리하는 시스템 |
| SS\_WH\_004 | 웹 | 서버로부터 받은 정보를 이용하여 신호체계 조작하는 시스템 |

* + 1. 유스케이스 다이어그램(UCD)

|  |  |  |  |
| --- | --- | --- | --- |
| UCD ID | SS\_UCD\_001 | UCD 명 | 교차로시스템 |
| 관련 서브시스템 ID | SS\_IP\_001 | 관련 서브시스템명 | 영상처리 |
| SS\_AP\_002 | 인공지능 |
| SS\_MS\_003 | 메인 서버 |
| SS\_WH\_004 | 웹 |
|  | | | |

* + 1. 유스케이스 목록

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 유스케이스  ID | 유스케이스명 | 유스케이스 설명 | 관련액터 ID | 관련 UCD ID | 관련 요구사항 ID |
| SS\_UCD\_001\_01 | 횡단보도 대기, 건너기 | 보행자가 횡단보도에서 대기하고 건너는 모습이 카메라에 촬영되어 임베디드 기기에 저장된다. | SS\_AC\_001  SS\_AC\_003 | SS\_UCD\_001 | SS\_ IP 001 |
| SS\_UCD\_002\_01 | 교차로 정지, 이동 | 차량이 교차로에서 정지하고 이동하는 모습이 카메라에 촬영되어 임베디드 기기에 저장된다. | SS\_AC\_002 SS\_AC\_003 | SS\_ IP 001 |
| SS\_UCD\_003\_001 | 영상처리 | 입력으로 들어오는 영상에서 보행자 및 차량을 인식한다. | SS\_AC\_001  SS\_AC\_002  SS\_AC\_003 | SS\_ IP 001 |
| SS\_UCD\_004\_001 | 인공지능 | 영상처리에서 들어오는 보행자 및 차량의 정보를 통해 더욱 자세한 정보를 얻는다. | SS\_AC\_003 | SS\_AP 002 |
| SS\_UCD\_005\_001 | 서버 | 영상처리와 인공지능을 걸친 데이터를 교통신호제어를 하기 위한 웹으로 전송한다. | SS\_AC\_003  SS\_AC\_004 | SS\_MS 003 |
| SS\_UCD\_006\_001 | 교통신호 제어 (웹) | 서버에서 전송된 정보로 교통신호를 제어한다.. | SS\_AC\_003  SS\_AC\_004 | SS \_WH004 |

* + 1. 액터 목록

|  |  |  |  |
| --- | --- | --- | --- |
| 액터 ID | 액터명 | 액터유형 | 액터설명 |
| SS\_AC\_001 | 보행자 | 주요 | 보행자의 이동방향, 수를 판별하기 위해 필요한 요소 |
| SS\_AC\_002 | 차량 | 주요 | 차량의 대기상태, 정지유무, 수를 판별하기 위해 필요한 요소 |
| SS\_AC\_003 | 임베디드 시스템 | 보조 | 각 카메라로부터 촬영된 보행자 및 차량 영상을 저장하고, 영상처리와 인공지능 처리가 완료된 데이터를 서버에 보내는 시스템 |
| SS\_AC\_004 | 신호제어 | 보조 | 보행자 및 차량의 분석, 처리가 완료된 데이터를 받아 신호제어에 조작하는 시스템 |

* + 1. 유스케이스 시나리오

|  |  |  |  |
| --- | --- | --- | --- |
| 유스케이스 ID | SS\_UC\_001 | 유스케이스명 | 보행자 및 차량 검지를 이용한 ITS시스템 |
| 1. 주요 액터  보행자, 운전자(차량)  2. 기본 시나리오  1) 직진 - 직진이 개통되고 해당 방향 횡단보도 개통  2) 좌회전 - 좌회전이 개통되고 모든 횡단보도 폐쇄  3) 교차하는 도로의 직진 - 직진 개통되고 해당 방향 횡단보도 개통  4) 교차하는 도로의 좌회전 - 좌회전이 개통되고 모든 횡단보도 폐쇄  5) 1로 복귀  3. 대안 시나리오  <1A/3A/3EA. 횡단보도에 저속 보행자 잔류 시>  1) 횡단보도에 아직 횡단중인 보행자가 있을 경우 보행자가 횡단보도에서 없어질 때까지 횡단보도를 개통 상태로 유지한다.  <2A/4A 좌회전 없을 시 좌회전 생략>  1. 좌회전 차로에 차가 없을 시 좌회전 개통하지 않고 기본흐름 3/5으로 진행  4. 예외 시나리오  <3E. 삼거리>  1) 교차하는 도로의 좌회전 개통되고 해당 방향 횡단보도 개통  2) 기본흐름 1로 복귀 | | | |

1. **설계**
   1. **클래스 설계서**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **D1** | **클래스 설계서** | | | | | |
| 시스템명 | | 보행자 및 차량 검지시스템 | 서브시스템명 | 교차로시스템 | | |
| 단계명 | | 설계 | 작성일자 | 2020. 04. 28 | 버전 | 1.0 |

* + 1. 설계 클래스 목록

|  |  |  |
| --- | --- | --- |
| 설계 클래스 ID | 설계 클래스명 | 관련 유스케이스 ID |
| EBA001IMGProc | 영상처리 | SS\_UC\_001 |
| EBA001AIProc | 인공지능 | SS\_UC\_001 |
| EBA001MainServer | 메인서버 | SS\_UC\_001 |
| EBA001Web | 웹 | SS\_UC\_001 |

* + 1. 시퀀스도

|  |  |  |  |
| --- | --- | --- | --- |
| 시퀀스도 ID | SS\_SD\_001-01 | 시퀀스도명 | 교차로시스템 |
| 관련 유스케이스 ID | SS\_UC\_001-01 | | |
| 주요 액터 | 보행자, 운전자(차량) | 주요 객체 (또는 클래스) | 영상처리, 인공지능, 메인 서버, 웹 |
|  | | | |

* + 1. 설계 클래스도 및 클래스 정의
       1. 전체 클래스도

|  |  |  |  |
| --- | --- | --- | --- |
| 설계 클래스도 ID | SS\_DCD\_001\_01 | 설계 클래스도명 | 전체 클래스도 |
| 관련 유스케이스 ID | SS\_UC\_001\_01 | | |
| 텍스트, 지도이(가) 표시된 사진  자동 생성된 설명 | | | |

* + - 1. 영상처리

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 설계 | 클래스도 ID | | EBA001IMGProc | 설계 클래스도명 | 영상처리 |
| 관련 | 유스케이스 | ID | SS\_UCD\_001\_01, SS\_UCD\_002\_01, SS\_UCD\_003\_01 | | |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **설계 클래스 ID** | EBA001IMGProc | | **설계 클래스 명** | | 영상처리 | |
| **클래스 이름** | **클래스 설명** | **요소 이름** | **함수/변수** | **타입** | **타입**  **(반환 값)** | **설명** |
| Camera | 카메라로 영상을 촬영하는 정보를 관할하는 클래스 | Crosswalk\_info | 변수 | Double |  | 보행자가 지나가는 횡단보도를 촬영하는 정보 |
| Intersection\_info | 변수 | Double |  | 차량이 지나가는  교차로를 촬영하는 정보 |
| Processing  Car Image Processing | 교차로 영상에서 차량의 정보를 인식하는데 관할하는 클래스.  실제로 영상처리가 이루어지는 부분 | Waiting\_pedestrian\_left | 변수 | Boolean |  | 카메라가 보는 방향 기준 왼쪽 횡단보도에서 대기하는 사람 유무 |
| Waiting\_pedestrian\_right | 변수 | Boolean |  | 카메라가 보는 방향 기준 오른쪽 횡단보도에서 대기하는 사람 유무 |
| Crossing\_pedestrian\_right | 변수 | Boolean |  | 카메라가 보는 방향 기준 오른쪽 횡단보도를 건너는 사람 유무 |
| PedestrianProcessing () | 함수 | Void | Void | 보행자에 대한 영상처리가 이루어지는 함수 |
| Car Image Processing | 교차로 영상에서 차량의 정보를 인식하는데 관할하는 클래스.  실제로 영상처리가 이루어지는 부분 | Waiting\_car\_right | 변수 | Boolean |  | 카메라가 보는 방향 기준 오른쪽 도로에서 대기하는 차량 유무 |
| Crossing\_car\_right | 변수 | Boolean |  | 카메라가 보는 방향 기준 오른쪽 도로를 통해 이동하는 차량 유무 |
| CarProcessing() | 함수 | Void | Void | 차량에 대한 영상처리가 이루어지는 함수 |

* + - 1. 인공지능

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 설계 | 클래스도 ID | | EBA001AIProc | 설계 클래스도명 | 인공지능 |
| 관련 | 유스케이스 | ID | SS\_UCD\_004\_01 | | |
|  | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **설계 클래스 ID** | EBA001AIProc | | **설계 클래스 명** | | 인공지능 | |
| **클래스 이름** | **클래스 설명** | **요소 이름** | **함수/변수** | **타입** | **타입**  **(반환 값)** | **설명** |
| Object | 사물에 대한 기본 속성을 정의하는 클래스 | direction | 변수 | Integer |  | 사물의 진행 방향 |
| position | 변수 | Integer |  | 사물의 위치 |
| isWaiting | 변수 | Boolean |  | 사물이 현재 대기 상태인지 여부 |
| AI | 인공지능이 데이터를 수집하고 이를 처리하여 서버로 전송하는 부분 | Pedestrian[] | 변수 | Pedestrian |  | 보행자들의 데이터를 저장할 배열 |
| Car[] | 변수 | Car |  | 차량들의 데이터를 저장할 배열 |
| SignalState | 변수 | Char |  | 신호등의 상태를 나타내는 변수. |
| RawData | 변수 | Integer |  | 카메라에서 촬영된 가공되지 않은 데이터 |
| PedestrianDetect() | 함수 |  | Void | 보행자의 상태 및 위치, 이동방향을 식별하여 해당되는 Pedestrian 객체의 값들을 업데이트하는 함수 |
| CarDetect() | 함수 |  | Void | 차량의 상태 및 위치, 이동방향을 식별하여 해당되는 Car객체의 값들을 업데이트하는 함수 |
| SendData() | 함수 |  | Void | 처리한 데이터를 가공하여 서버로 전송하는 함수 |
| ReceiveData() | 함수 |  | Void | Raw data를 받는 함수 |
| Car | Object 클래스를 상속받아 자동차에 대한 속성을 정의하는 클래스 | isLeftStop | 변수 | Boolean |  | 차량이 좌회전 차선의 정지선에 위치해 있는지에 대한 여부 |
| carState | 변수 | Integer |  | 차량이 현재 어떤 상태에 있는지에 대한 여부. 상태에는 교차로 진입 전/통과의 상태가 있다. |
| Pedestrian | Object 클래스를 상속받아 보행자에 대한 속성을 정의하는 클래스 | inCrosswalk | 변수 | Boolean |  | 보행자가 횡단보도에 있는지에 대한 여부 |
| pedState | 변수 | Integer |  | 보행자가 현재 어떤 상태에 있는지에 대한 여부. 상태에는 횡단과 대기 상태가 존재한다. |

* + - 1. 서버

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 설계 | 클래스도 ID | | EBA001MainServer | 설계 클래스도명 | 서버 |
| 관련 | 유스케이스 | ID | SS\_UCD\_005\_01 | | |
|  | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **설계 클래스 ID** | EBA001MainServer | | **설계 클래스 명** | | 인공지능 | |
| **클래스 이름** | **클래스 설명** | **요소 이름** | **함수/변수** | **타입** | **타입(반환 값)** | **설명** |
| Raspberry Pi 1~4 | 라즈베리파이1~4에서 얻은 정보가 있는 클래스 | Walker\_Pi1~4 | 변수 | txt |  | 라즈베리파이1~4에서 얻어진 보행자 정보 |
| Car\_Pi1~4 | 변수 | txt |  | 라즈베리파이1~4에서 얻어진 차량 정보 |
| Detected\_Pi1~4\_Walker() | 함수 |  | void | 라즈베리파이1~4에 저장된 보행자의 값들을 호출하는 함수 |
| Detected\_Pi1~4\_Car() | 함수 |  | void | 라즈베리파이1~4에 저장된 차량의 값들을 호출하는 함수 |
| Main Server | 각 라즈베리파이로부터 얻은 정보를 관리하는 클래스 | Raspberry Pi 1~4 | 변수 | txt |  | 라즈베리파이1~4에 대한 정보 |
| Send\_Pi1~4\_Info() | 함수 |  | void | 라즈베리파이1~4에 저장된 정보를 보내는 함수 |
| Reciv\_Pi1~4\_Info() | 함수 |  | void | 라즈베리파이1~4에 저장된 정보를 받기 위한 함수 |
| Web | 서버로부터 얻은 가공된 정보를 관리하는 클래스 | Walker Info | 변수 | txt |  | 코너에서 얻어진 보행자들의 정보 |
| Car Info | 변수 | txt |  | 도로에서 얻어진 차량들의 정보 |

* + - 1. 웹

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 설계 | 클래스도 ID | | EBA001Web | 설계 클래스도명 | 웹 |
| 관련 | 유스케이스 | ID | SS\_UCD\_006\_01 | | |
|  | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **설계 클래스ID** | EBA001Web | | **설계 클래스 명** | | 웹 | |
| **클래스 이름** | **클래스 설명** | **요소 이름** | **함수/변수** | **타입** | **타입(반환 값)** | **설명** |
| WalkerInfo | 각 코너에 대한 정보를 관할하는 클래스 | L\_WalkNum | 변수 | Integer |  | 한 코너에 있는 왼쪽 보행자 대기소에 대기중인 보행자들의 수 |
| R\_WalkNum | 변수 | Integer |  | 한 코너에 있는 오른쪽 보행자 대기소에 대기중인 보행자들의 수 |
| L\_Direction | 변수 | Boolean |  | 왼쪽으로 진행할 보행자의 유무 |
| R\_Direction | 변수 | Boolean |  | 오른쪽으로 진행할 보행자의 유무 |
| L\_Crossing | 변수 | Boolean |  | 왼쪽 횡단보도를 건너는 보행자의 유무 |
| R\_Crossing | 변수 | Boolean |  | 오른쪽 횡단보도를 건너는 보행자의 유무 |
| CarInfo | 전체 도로에서의 차량의 정보를 관할하는 클래스 | A\_CarNum | 변수 | Integer |  | A 구역의 차량의 수 |
| A\_LCar | 변수 | Boolean |  | A 구역의 좌회전 대기차량의 유무 |
| B\_CarNum | 변수 | Integer |  | B 구역의 차량의 수 |
| B\_LCar | 변수 | Boolean |  | B 구역의 좌회전 대기차량의 유무 |
| C\_CarNum | 변수 | Integer |  | C 구역의 차량의 수 |
| C\_LCar | 변수 | Boolean |  | C 구역의 좌회전 대기차량의 유무 |
| D\_CarNum | 변수 | Integer |  | D 구역의 차량의 수 |
| D\_LCar | 변수 | Boolean |  | D 구역의 좌회전 대기차량의 유무 |
| TrafficInfo | 교통신호를 제어하는 데에 필요한 정보를 관할하는 클래스 | A\_Cros | 변수 | Boolean |  | A\_구역에서 횡단보도를 지나고 있는 보행자의 유무 |
| A\_LCancel | 변수 | Boolean |  | A\_구역에서 좌회전의 생략유무 |
| B\_Cros | 변수 | Boolean |  | B\_구역에서 횡단보도를 지나고 있는 보행자의 유무 |
| B\_LCancel | 변수 | Boolean |  | B\_구역에서 좌회전의 생략유무 |
| C\_Cros | 변수 | Boolean |  | C\_구역에서 횡단보도를 지나고 있는 보행자의 유무 |
| C\_LCancel | 변수 | Boolean |  | C\_구역에서 좌회전의 생략유무 |
| D\_Cros | 변수 | Boolean |  | D\_구역에서 횡단보도를 지나고 있는 보행자의 유무 |
| D\_LCancel | 변수 | Boolean |  | D\_구역에서 좌회전의 생략유무 |
| LCar\_CancelInfo() | 함수 | void | void | CarInfo 클래스의 요소들 중 좌회전차량 유무정보로 TrafficInfo클래스의 LCancel정보 업데이트 |
| CrossingWalkerInfo() | 함수 | void | void | WalkInfo클래스의 요소들 중 보행자가 현재 통과하고 있는 횡단보도의 정보를 얻어 Cros변수 업데이트. |
| DirectionInfo() | 함수 | void | void | WalkInfo클래스의 요소들 중 보행자의 이동방향정보를 통해 Cros변수 업데이트. |

* 1. **컴포넌트 설계서**

|  |  |  |  |
| --- | --- | --- | --- |
| 컴포넌트 ID | SS\_CO\_001 | 컴포넌트명 | 교차로시스템 |
| 관련 유스케이스 ID | SS\_UC\_001 | | |
| 1. 임베디드 시스템 컴포넌트 구조도    2. 실시간 커뮤니케이션 컴포넌트 구조도  3. 유저 컴포넌트 구조도    4. 전체 컴포넌트 구조도 | | | |

1. 컴포넌트 목록

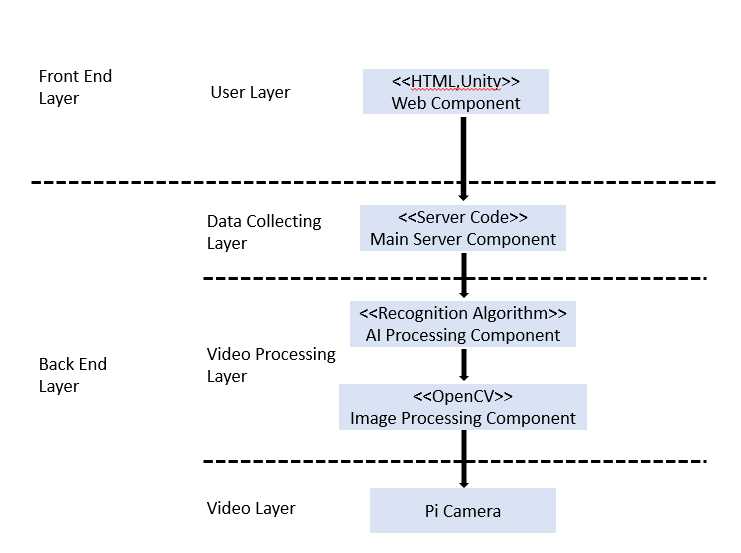
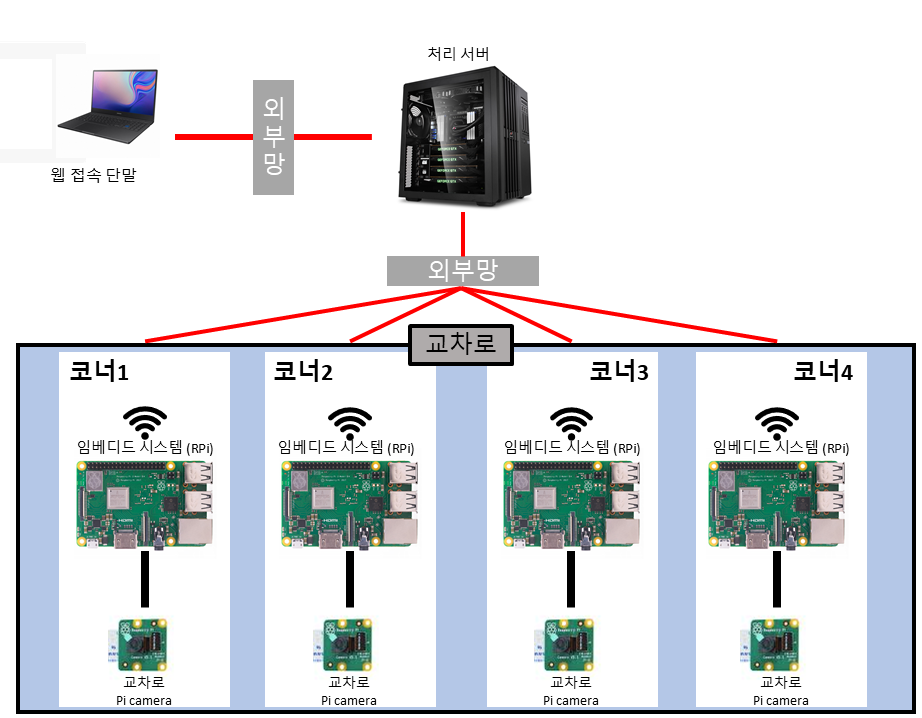
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 컴포넌트 ID | | 컴포넌트 명 | 개요 | | 관련 유스케이스 ID |
| SS\_CO | SS\_CO\_001~4 | Raspberry Pi 1~4 데이터처리 | 파이카메라1~4에 촬영된 Raw Data를 영상처리 및 인공지능 분석처리 | | SS\_UCD\_001\_001  SS\_UCD\_002\_001  SS\_UCD\_003\_001  SS\_UCD\_004\_001 |
| SS\_SP | SS\_SP\_001~4 | Raspberry Pi 1~4 파일 관리 | Raspberry Pi 1~4로부터 가공된 데이터 파일 연결 및 관리 | Web으로 보행자 및 차량 데이터 전송 | SS\_UCD\_004\_001 |
| SS\_PD | SS\_PD\_001~004 | 보행자 | 각 파이로부터 촬영된 보행자 | | SS\_UCD\_001\_001  SS\_UCD\_003\_001 |
| SS\_CD | SS\_CD\_001~004 | 차량 | 각 파이로부터 촬영된 차량 | | SS\_UCD\_002\_001  SS\_UCD\_003\_001 |

1. 컴포넌트 명세

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 컴포넌트 ID | SS\_CO\_(001~004) | 컴포넌트명 | Raspberry Pi 데이터 처리 | |
| 컴포넌트 개요 | 각 파이카메라로부터 촬영된 Raw 데이터를 영상처리 및 인공지능 분석처리를 수행한다. | | | |
| 내부 클래스 | | | | |
| ID | 클래스명 | 비고 | | Operation |
| SS\_IP\_PD\_001 | Pedestrian Image Processing | 각 파이카메라로부터 촬영되어 저장된 임베디드 기기에서 보행자에 대한 이미지 처리를 수행한다. | | -Detected\_Pi\_Walker()  -Ped\_Processing() |
| SS\_IP\_CD\_001 | CarImage Processing | 각 파이카메라로부터 촬영되어 저장된 임베디드 기기에서 차량에 대한 이미지 처리를 수행한다. | | -Detected\_Pi\_Car()  -Car\_Processing() |
| SS\_AP\_001 | AI Processing | 영상처리를 거친 보행자 및 차량 데이터에 대해 AI 분석 및 처리를 수행한다. | | -Ped\_Detect()  -Car\_Detect() |
| SS\_SP\_001 | File Management | 영상처리와 인공지능 과정을 거친 데이터를 수신 후, 파일 관리 및 웹으로의 전송을 수행한다. | | -Send\_Pi\_Info()  -Reciv\_Pi\_Info() |

* 1. **아키텍쳐 설계서**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **D5** | **아키텍처 설계서** | | | | | |
| 시스템명 | | 보행자 및 차량 검지시스템 | 서브시스템명 | 교차로시스템 | | |
| 단계명 | | 설계 | 작성일자 | 2020. 04. 28 | 버전 | 1.0 |

* + 1. 소프트웨어 아키텍쳐
    2. 시스템 아키텍처
    3. 아키텍처 요구사항 및 구현방안

|  |  |  |
| --- | --- | --- |
| 요구사항 ID | | SS\_IP\_REQ\_010 |
| 요구사항 내용 | | |
| 카메라가 검지하고 있는 영역의 보행자 및 차량을 감지하여 차량 및 보행자의 수와 개별 물체의 위치를 식별할 수 있어야 한다. | | |
| 구현방안 | | |
| 영상처리 알고리즘을 이용하여 차량 및 보행자의 수와 위치를 90%이상의 정확도로 구현하는 것을 목표로 한다. | | |
| 요구사항 ID | | SS\_AI\_REQ\_020 |
| 요구사항 내용 | | |
| 보행자의 위치와 이동방향을 식별해야 하며, 교차로상에 위치한 차량 중 좌회전 대기차량, 대기 차량의 수와 같은 정보를 얻어야 한다. | | |
| 구현방안 | | |
| 인공지능을 이용하여 보행자 정보의 식별은 95%이상의 정확도로, 좌회전 대기차량의 식별은 신호체계 연동에 있어 중요한 요소이므로 99% 이상의 정확도로 구현하는 것을 목표로 한다. | | |
| 요구사항 ID | SS\_MS\_REQ\_030 | |
| 요구사항 내용 | | |
| 서버는 교차로상에 설치된 임베디드 기기에서 처리된 후 보내진 데이터를 실시간으로 수신할 수 있어야 한다. | | |

* 1. **사용자 인터페이스 설계서**
     1. 화면

|  |  |  |
| --- | --- | --- |
| **화면ID** | **화면명** | **관련 유스케이스ID** |
| VS\_MV | 메인View | SS\_UCD\_006\_001 |

* + 1. 화면 상세설계

|  |  |  |  |
| --- | --- | --- | --- |
| 화면 ID | VS\_MV | 화면 명 | 메인View |
| 관련 유스케이스 | SS\_UCD\_006\_001(웹) | | |
| 관련 시퀀스도 ID | SS\_SD\_001-01 (9번) | | |
| 화면 유형 | 서버에서 전송된 정보 표시 | | |
| 화면 개요 | 메인 화면을 보여준다.  검지중인 구역의 횡단보도를 보행하고 있는 보행자가 있을 때, 경고를 보여준다.  현재 검지중인 한 도로의 차량의 수를 보여준다.  떠, 도로의 좌회전 대기차량을 보여준다. | | |
| 비행기, 표지판이(가) 표시된 사진  자동 생성된 설명 | | | |
| 기술적 고려사항 | | | |
| 마우스를 이용한 시점 변경, 실시간으로 업데이트되는 정보 | | | |

* 1. **총괄시험 계획서**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **D6** | **총괄시험 계획서** | | | | | |
| 시스템명 | | 보행자 및 차량 검지시스템 | 서브시스템명 | 교차로시스템 | | |
| 단계명 | | 설계 | 작성일자 | 2020. 04.28 | 버전 | 1.0 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. 시험 대상 시스템    1. 시스템 개요   보행자 및 차량 검지시스템은 영상처리, 인공지능, 서버, 웹, 총 4부분으로 구성되어 있다.  영상처리는 교차로 영상으로부터 교차로에 보행자 및 차량이 대기해 있는지 혹은 이동하고 있는지 알아낼 수 있도록 처리하는 시스템이다.  인공지능은 영상처리에서 추출된 신호를 이용해서 보행자가 어느 횡단보도를 통해서 건너는지, 차량이 대기 혹은 직진, 좌회전을 하는지 파악할 수 있도록 처리하는 시스템이다.  서버는 라즈베리파이에서 도출된 결과를 받아서 정보를 저장하는 공간이다. 이후 웹에 정보를 전송할 수 있도록 웹과 통신이 되어있다.  웹은 서버로부터 받은 정보를 표시하는 공간이다.   1. 가정 또는 제약사항  * 본 시스템에서는 실시간으로 교차로 영상을 녹화하는 것이 아닌, 교차로 영상이 이미 주어진 것으로 가정한다. * 신호 반응은 실제 신호등으로 확인하는 것이 아닌, 응용 프로그램 내에서 확인한다.  1. 시험 전략    1. 시험 범위   시험 범위는 영상처리, 인공지능, 서버, 웹, 총 4개의 부분으로 지정한다.   |  |  | | --- | --- | | **구분** | **시험 항목** | | 영상처리 | 보행자 및 차량 인식 | | 인공지능 | 보행자 및 차량 이동방향 파악 | | 서버 | 라즈베리파이 및 웹으로부터 통신 | | 웹 | 영상 정보 표시 |  * 1. 개발 단계별 수행할 시험 종류      1. 단위시험   단위시험은 크게 임베디드 시스템으로 이루어진다. 임베디드 시스템에서는 3단계로 이루어진다. 영상처리, 인공지능, 서버의 각 단위로 나누어져서 수행한다.  3.2.1.1 영상처리: 보행자와 차량의 검지가 목표이상의 인식률을 가지는지 확인한다.  3.2.1.2인공지능: 인공지능을 통한 정보의 정확도가 목표치 이상인지 확인한다.  3.2.1.3 서버: 목표하는 응답속도 안에 정보를 전송하는지 확인한다.   * + - 1. 웹: 서버에서 전송된 정보를 정확하게 표시하는지 확인한다.  |  |  |  | | --- | --- | --- | |  | 단위시험 | 통합시험 | | 시험대상 | 모듈 | 서브시스템, 시스템 구성 요소 | | 목적 | 단위 모듈이 요구사항 명세서의 요구에 부합되는지 검증 | 전체 동작이 매끄럽게 이어지는지, 모든 성능이 요구사항 명세서의 요구에 부합되는지 검증 | | 테스터 | 시험 담당자 | 시험 담당자 | | 기준선 | 요구사항 명세서 | 요구사항 명세서 | | 시험환경 | 개발환경 | 개발환경 |  * + 1. 통합시험   통합시험은 각각의 부분을 하나의 프로그램으로 통합하여 전체적으로 기능이 수행되는지 검증한다.   1. 시험 실행 계획    1. 시험 수행 절차  |  |  | | --- | --- | | 구분 | 시한폭탄 | | 총괄 시험 계획 | * 시험 전략 및 수행절차 수립 * 시험 전체 일정 수립 * 총괄시험 계획서 작성 | | 단위 시험 계획 | * 부분 시스템별 단위시험 케이스 도출 * 수행 * 오류 검토 및 수정 | | 통합시험 계획 | * 통합시험 시나리오 작성 * 수행 * 오류 검토 및 수정 |  * 1. 시험 환경   시험에 필요한 하드웨어, 소프트웨어, 시험 도구 등에 대한 환경 요구 사항은 아래와 같다.   * + 1. 하드웨어     2. 서버        - * CPU: Intel® Core™ i7-5820K          * Memory: DDR4 16GB PC4-17000          * HDD: SATA(Serial-ATA) 4TB / 7200RPM     3. 라즈베리파이 3B (Raspberry Pi 3 Model B)        - * SoC: BCN2837B0          * GPU: Dual Core Video Core IV® Multimedia Co-Processor          * Speed: 1,400MHz          * RAM: 1GB          * USB Port: 4 EA          * Ethernet: 1000Base-T          * Wireless LAN: 802.11 ac/n     4. 소프트웨어  1. 서버    * + - * OS: Linux          * Tool: PyCharm 2. 라즈베리파이 3B    * + - * OS: Raspbian    1. 시험 조직 및 역할       1. 시험 조직   시험은 캡스톤 2반 2조 시한폭탄조가 수행한다.   * + 1. 책임 및 역할        - * 14011074 조현기: 조장, 웹 시험          * 15010980 구본학: 서버 시험          * 15010993 조남규: AI 시험          * 15011059 이세영: 영상처리 시험   1. 시험 수행 일정  |  |  |  |  |  | | --- | --- | --- | --- | --- | | 순번 | 주요활동 | 시험일자 | 완료일자 | 비고 | | 1 | 총괄시험 계획 | 2020.04.24 | 2020.04.30 |  | | 2 | 단위시험 수행 | 2020.05.25 | 2020.05.29 |  | | 3 | 단위시험 결함수정 및 결과 확인 | 2020.05.29 | 2020.06.04 |  | | 4 | 통합시험 수행 | 2020.06.04 | 2020.06.11 |  | | 5 | 통합시험 결함수정 및 결과 확인 | 2020.06.11 | 2020.06.18 |  | |