요구사항 분석서

컴퓨터 공학과 2반

2조 (시한폭탄 조)

**지도교수**

문현준 교수님, 권기학 교수님

**팀원**

14011074 조현기

15010980 구본학

15010993 조남규

15011059 이세영

목차

1. **개요**
   1. 시스템의 목적
2. **요구사항**
   1. 임베디드 기기 – 영상처리
   2. 임베디드 기기 – 인공지능
   3. 서버
   4. 웹
3. **유스케이스**
   1. 유스케이스 다이어그램
   2. 유스케이스 시나리오
   3. 클래스 다이어그램
   4. 시퀀스 다이어그램
4. **구현사양**
   1. 하드웨어 인터페이스
   2. 소프트웨어 인터페이스
5. **개요**
   1. 시스템의 개요 및 개발 목적

2013~2018년 동안의 도로 유형별 사고 통계 (자료 출처: taas 교통사고분석시스템)

교차로 사고란, 교차로 내 또는 교차로 측 단 후방 30m 이내의 부분에서 발생한 사고를 말한다. 교차로는 여러 방향에서 진행하는 차들로 인하여 상충의 위험성이 높아 주의 운전이 힐요 하다. 하지만 교차로에서의 사고 현황은 줄어들지 않고 있다. 따라서, 본 시스템은 교차로에 설치되는 각 임베디드 기기를 통하여 교차로상에 위치한 개별 보행자의 이동방향 및 차량을 검출하여 교차로에서의 보행자 안전을 보장하며 차량의 검지, 특히 좌회전 차선의 차량을 검출해 좌회전 대기중인 차량이 있을 경우에만 좌회전 신호를 주는 방식 등의 방법으로 교차로의 신호체계의 효율성을 높일 것이다.

본 시스템은 임베디드 기기 내부의 영상처리 및 인공지능, 서버, 웹으로 이루어진다. 임베디드 기기는 내부의 카메라를 이용하여 영상을 촬영하며 그 영상을 활용해 교차로의 보행자와 차량을 감지한다. 인공지능은 보행자의 수, 위치, 이동방향뿐만 아니라 좌회전 차량의 대기 정보를 서버로 전달한다. 서버는 이러한 정보를 받아 기기별로 관리하며 이 정보를 웹에게 제공한다. 웹에서는 각 임베디드 기기 들에서 들어온 정보를 이용해 신호들 제어한다.

1. **요구사항**
   1. 임베디드 기기 – 영상처리
      1. 기능적 요구사항

|  |  |  |
| --- | --- | --- |
| 번호 | 요구사항 | 상세내용 |
| V\_1 | 보행자 인식 | 카메라가 검지하고 있는 영역의 보행자를 감지하여 그에 대한 위치와 수를 알 수 있어야 한다. |
| V\_2 | 차량 인식 | 카메라가 검지하고 있는 교차로 영역의 차량을 감지해야 한다. |

* + 1. 비기능적 요구사항

|  |  |  |
| --- | --- | --- |
| 번호 | 요구사항 | 상세내용 |
| V\_N1 | 보행자 정보 | 보행자의 위치, 수를 90%이상정확하게 판단해야 한다. |
| V\_N2 | 차량 인식 | 차량의 인식을 90%이상의 정확도로 인식해야 한다. |

* 1. 임베디드 기기 – 인공지능
     1. 기능적 요구사항

|  |  |  |
| --- | --- | --- |
| 번호 | 요구사항 | 상세내용 |
| A\_1 | 보행자 이동방향 구별 | 보행자의 위치로 보행자의 이동방향을 구별해야 한다. |
| A\_2 | 차량의 정보 추출 | 좌회전 대기차량, 대기 차량의 수와 같은 정보를 추출해야 한다. |

* + 1. 비기능적 요구사항

|  |  |  |
| --- | --- | --- |
| 번호 | 요구사항 | 상세내용 |
| A\_N1 | 보행자 이동방향의 정확도 | 보행자의 이동방향은 보행자의 안전을 위해 최대한 정확해야 하며 95%이상을 목표로 한다. |
| A\_N2 | 차량 정보의 정확도 | 좌회전 대기차량 구분은 좌회전 신호의 유무를 결정하는 정보이다. 따라서 99% 이상의 정확도를 목표로 구현한다. |

* 1. 서버
     1. 기능적 요구사항

|  |  |  |
| --- | --- | --- |
| 번호 | 요구사항 | 상세내용 |
| S\_1 | 임베디드 기기들 과의 무선연결 지원 | 임베디드 기기들과 서버 간의 와이파이를 이용한 무선통신이 가능해야 한다. |
| S\_2 | 실시간 업데이트 | 임베디드 기기들에게서 들어오는 정보를 실시간으로 업데이트 해야 한다. |

* + 1. 비기능적 요구사항

|  |  |  |
| --- | --- | --- |
| 번호 | 요구사항 | 상세내용 |
| S\_N1 | 빠른 데이터 처리 | 4개의 임베디드 기기에서 보내는 데이터를 빠르게 처리해야 한다. |
| S\_N2 | 클라이언트와의 응답시간 | 신호체계를 위한 클라이언트와의 응답시간은 매우 빨라야 하며, 50ms미만을 목표로 한다. |

* 1. 웹
     1. 기능적 요구사항

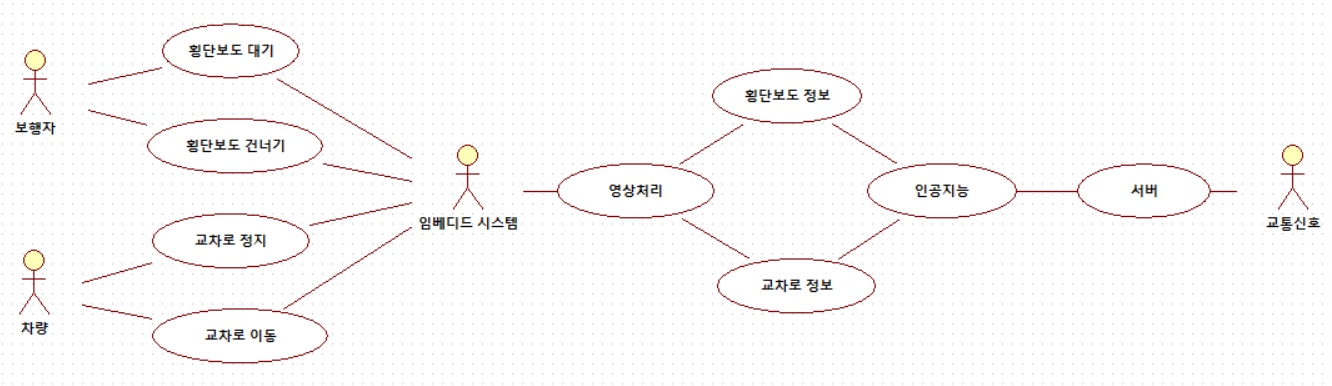
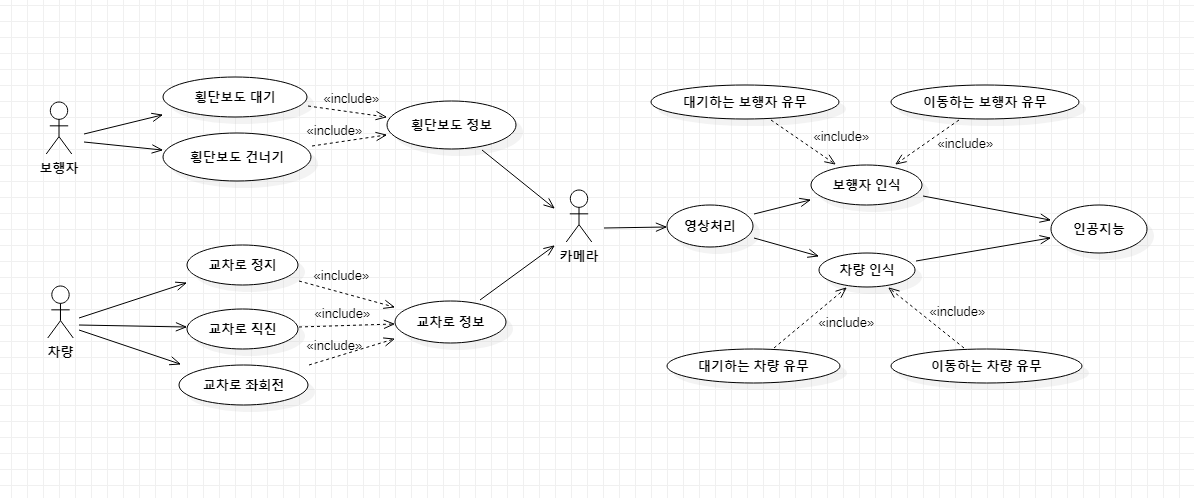
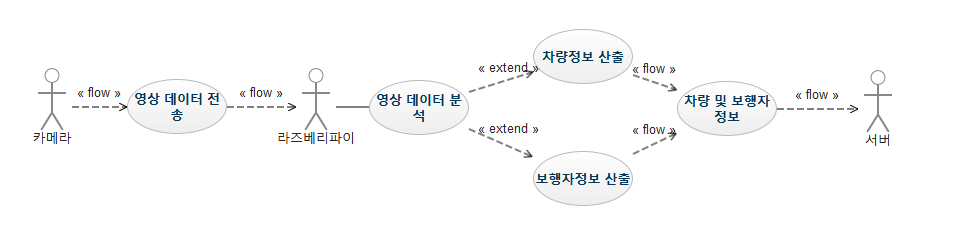
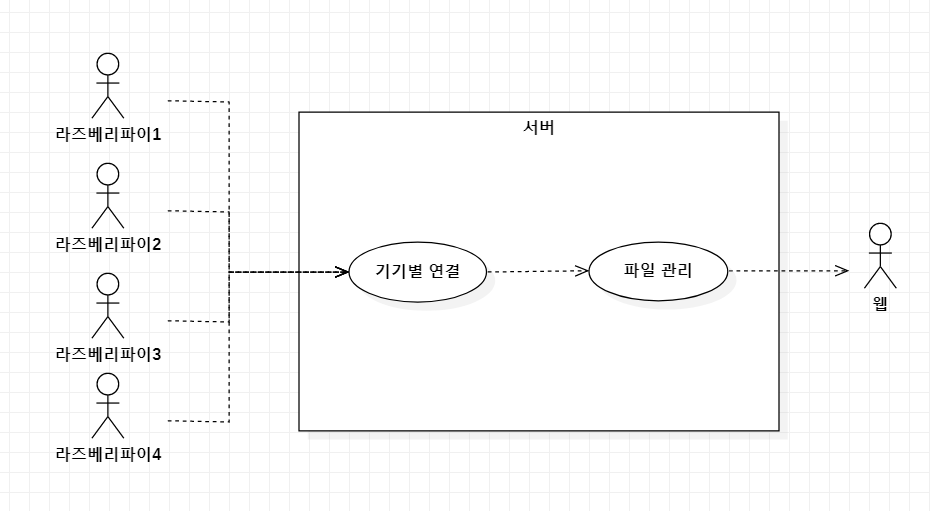
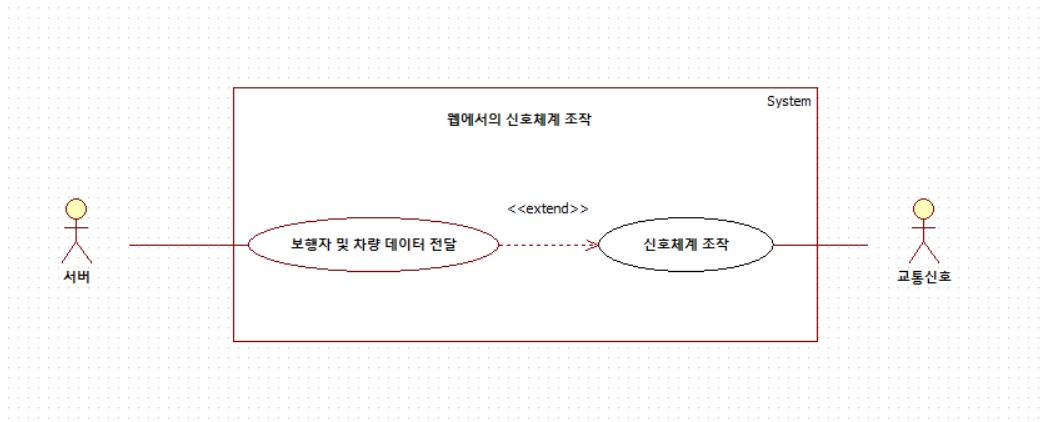
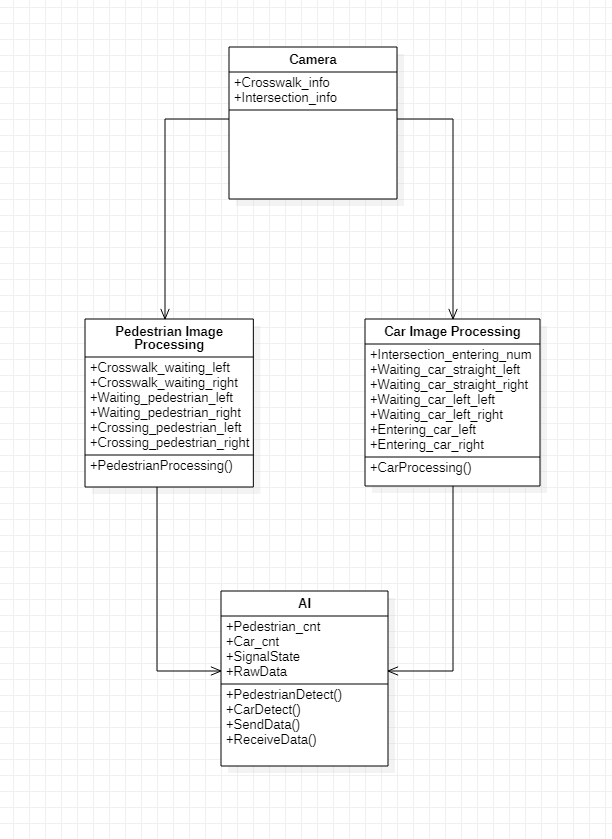
|  |  |  |
| --- | --- | --- |
| 번호 | 요구사항 | 상세내용 |
| W\_1 | 서버 통신 | 서버에서 교차로의 보행자, 차량의 정보를 가져와야 한다. |
| W\_2 | 신호체계 조작 | 서버에서 얻은 정보로 신호를 적절히 조작해야 한다. |

* + 1. 비기능적 요구사항

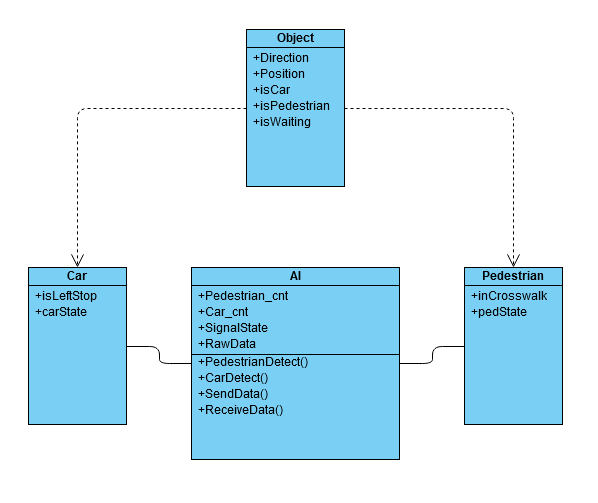
|  |  |  |
| --- | --- | --- |
| 번호 | 요구사항 | 상세내용 |
| W\_N1 | 보행자 안전 시스템 | 보행자가 횡단보도를 건널 때, 보행자 존재 신호를 준다. |
| W\_N2 | 좌회전 차량에 따른 신호부여 | 좌회전 차량이 존재할 때만 좌회전 신호를 준다. |
| W\_N3 | 차량 수에 따른 신호부여 | 차량 수가 많을수록 신호를 더 길게 준다. |

1. **유스케이스 및 다이어그램**
   1. **유스케이스 시나리오 및 유스케이스 다이어그램**
      1. 전체 시나리오

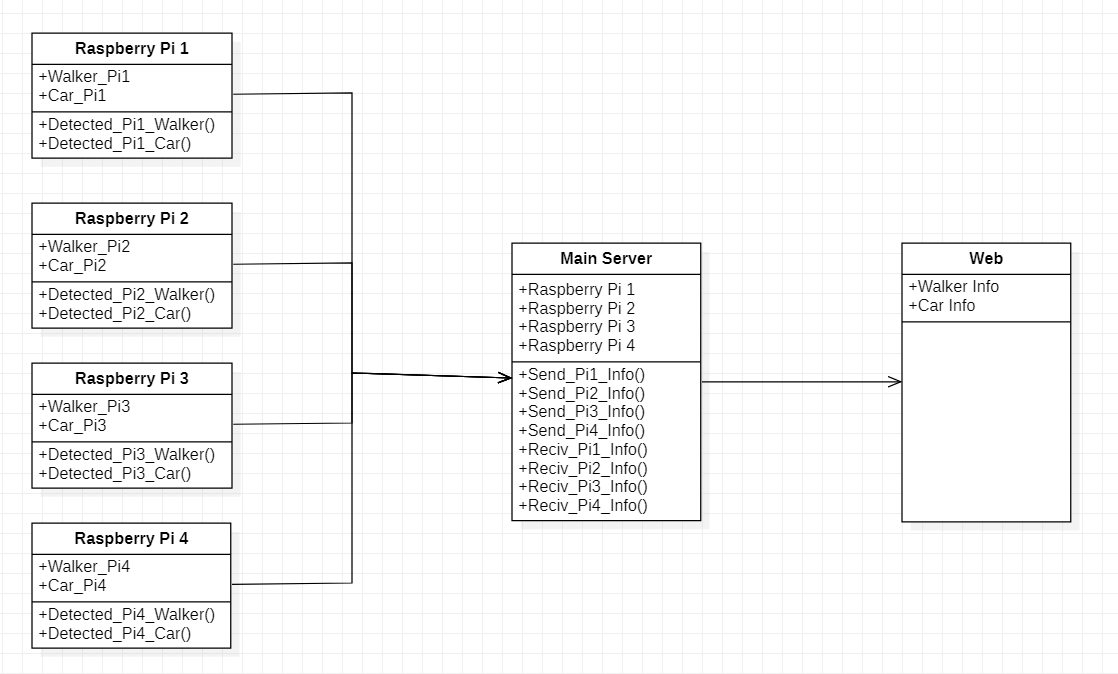
|  |  |
| --- | --- |
| 시나리오 제목 | 보행자 및 차량 검지를 이용한 ITS시스템 |
| 액터 | 보행자, 운전자(차량) |
| 기본흐름 | 1. 직진 - 직진이 개통되고 해당 방향 횡단보도 개통  2. 좌회전 - 좌회전이 개통되고 모든 횡단보도 폐쇄  3. 교차하는 도로의 직진 - 직진 개통되고 해당 방향 횡단보도 개통  4. 교차하는 도로의 좌회전 - 좌회전이 개통되고 모든 횡단보도 폐쇄  5. 1로 복귀 |
| 대안흐름 | <1A/3A/3EA. 횡단보도에 저속 보행자 잔류 시>   1. 횡단보도에 아직 횡단중인 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 |
| 예외흐름 | <3E. 삼거리>   1. 교차하는 도로의 좌회전 개통되고 해당 방향 횡단보도 개통 2. 기본흐름 1로 복귀 |

* + 1. 전체 시나리오 다이어그램
    2. 영상처리 다이어그램
    3. 인공지능 다이어그램
    4. 서버 다이어그램
    5. 웹 다이어그램
  1. **클래스 다이어그램**
     1. 영상처리

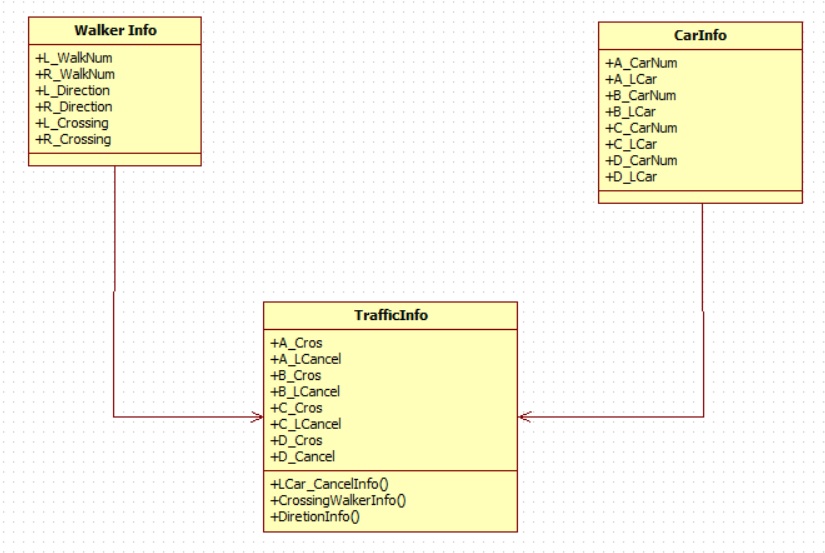
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **클래스 이름** | **클래스 설명** | **요소 이름** | **함수/변수** | **타입** | **타입**  **(반환 값)** | **설명** |
| Camera | 카메라로 영상을 촬영하는 정보를 관할하는 클래스 | Crosswalk\_info | 변수 | Double |  | 보행자가 지나가는 횡단보도를 촬영하는 정보 |
| Intersection\_info | 변수 | Double |  | 차량이 지나가는 교차로를 촬영하는 정보 |
| Pedestrian Image Processing | 교차로 영상에서 보행자의 정보를 인식하는데 관할하는 클래스.  실제로 영상처리가 이루어지는 부분 | Crosswalk\_waiting\_left | 변수 | Integer |  | 왼쪽 횡단보도의 대기 장소 수 |
| Crosswalk\_waiting\_right | 변수 | Integer |  | 오른쪽 횡단보도의 대기 장소 수 |
| Waiting\_pedestrian\_left | 변수 | Boolean |  | 왼쪽 횡단보도로 이동하려는 사람 유무 |
| Waiting\_pedestrian\_right | 변수 | Boolean |  | 오른쪽 횡단보도로 이동하려는 사람 유무 |
| Crossing\_pedestrian\_left | 변수 | Boolean |  | 왼쪽 횡단보도를 건너고 있는 사람 유무 |
| Crossing\_pedestrian\_right | 변수 | Boolean |  | 오른쪽 횡단보도를 건너고 있는 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 | Intersection\_entering\_num | 변수 | Integer |  | 차량이 진입하는 도로의 대기 장소 수 |
| Waiting\_car\_straight\_left | 변수 | Boolean |  | 카메라가 보는 방향 기준 왼쪽도로로부터 직진하려는 차량의 유무 |
| Waiting\_car\_straight\_right | 변수 | Boolean |  | 카메라가 보는 방향 기준 오른쪽 도로로부터 직진하려는 차량의 유무 |
| Waiting\_car\_left\_left | 변수 | Boolean |  | 카메라가 보는 방향 기준 왼쪽 도로로부터 좌회전하려는 차량의 유무 |
| Waiting\_car\_left\_right | 변수 | Boolean |  | 카메라가 보는 방향 기준 오른쪽 도로로부터 좌회전하려는 차량의 유무 |
| Entering\_car\_left | 변수 | Boolean |  | 카메라가 보는 방향 기준 왼쪽 도로로 들어오고 있는 차량 유무 |
| Entering\_car\_right | 변수 | Boolean |  | 카메라가 보는 방향 기준 오른쪽 도로로 들어오고 있는 차량 유무 |
| CarProcessing() | 함수 | Void | Void | 차량에 대한 영상처리가 이루어지는 함수 |

* + 1. 인공지능

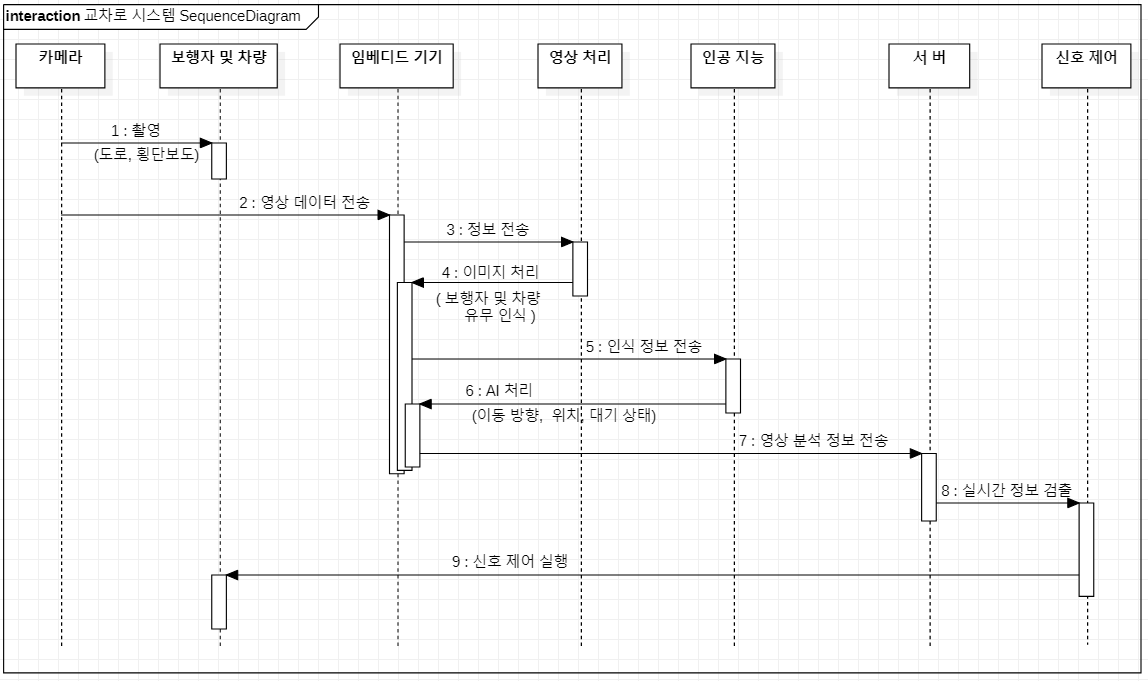
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **클래스 이름** | **클래스 설명** | **요소 이름** | **함수/변수** | **타입** | **타입**  **(반환 값)** | **설명** |
| Object | 사물에 대한 기본 속성을 정의하는 클래스 | direction | 변수 | Integer |  | 사물의 진행 방향 |
| position | 변수 | Integer |  | 사물의 위치 |
| isWaiting | 변수 | Boolean |  | 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 가공되지않은 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 위치해있는지에 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **클래스 이름** | **클래스 설명** | **요소 이름** | **함수/변수** | **타입** | **타입(반환 값)** | **설명** |
| Raspberry Pi 1 | 라즈베리파이1에서 얻은 정보가 있는 클래스 | Walker\_Pi1 | 변수 | txt |  | 라즈베리파이1에서 얻어진 보행자 정보 |
| Car\_Pi1 | 변수 | txt |  | 라즈베리파이1에서 얻어진 차량 정보 |
| Detected\_Pi1\_Walker() | 함수 |  | void | 라즈베리파이1에 저장된 보행자의 값들을 호출하는 함수 |
| Detected\_Pi1\_Car() | 함수 |  | void | 라즈베리파이1에 저장된 차량의 값들을 호출하는 함수 |
| Raspberry Pi 2 | 라즈베리파이2에서 얻은 정보가 있는 클래스 | Walker\_Pi2 | 변수 | txt |  | 라즈베리파이2에서 얻어진 보행자 정보 |
| Car\_Pi2 | 변수 | txt |  | 라즈베리파이2에서 얻어진 차량 정보 |
| Detected\_Pi2\_Walekr() | 함수 |  | void | 라즈베리파이2에 저장된 보행자의 값들을 호출하는 함수 |
| Detected\_Pi2\_Car() | 함수 |  | void | 라즈베리파이2에 저장된 차량의 값들을 호출하는 함수 |
| Raspberry Pi 3 | 라즈베리파이3에서 얻은 정보가 있는 클래스 | Walker\_Pi3 | 변수 | txt |  | 라즈베리파이3에서 얻어진 보행자 정보 |
| Car\_Pi3 | 변수 | txt |  | 라즈베리파이3에서 얻어진 차량 정보 |
| Detected\_Pi3\_Walker() | 함수 |  | void | 라즈베리파이3에 저장된 보행자의 값들을 호출하는 함수 |
| Detected\_Pi3\_Car() | 함수 |  | void | 라즈베리파이3에 저장된 차량의 값들을 호출하는 함수 |
| Raspberry Pi 4 | 라즈베리파이4에서 얻은 정보가 있는 클래스 | Walker\_Pi4 | 변수 | txt |  | 라즈베리파이4에서 얻어진 보행자 정보 |
| Car\_Pi4 | 변수 | txt |  | 라즈베리파이4에서 얻어진 차량 정보 |
| Detected\_Pi4\_Walker() | 함수 |  | void | 라즈베리파이4에 저장된 보행자의 값들을 호출하는 함수 |
| Detected\_Pi4\_Car() | 함수 |  | void | 라즈베리파이4에서 저장된 차량의 값들을 호출하는 함수 |
| Main Server | 각 라즈베리파이로부터 얻은 정보를 관리하는 클래스 | Raspberry Pi 1 | 변수 | txt |  | 라즈베리파이1에 대한 정보 |
| Raspberry Pi 2 | 변수 | txt |  | 라즈베리파이1에 대한 정보 |
| Raspberry Pi 3 | 변수 | txt |  | 라즈베리파이2에 대한 정보 |
| Raspberry Pi 4 | 변수 | txt |  | 라즈베리파이1에 대한 정보 |
| Send\_Pi1\_Info() | 함수 |  | void | 라즈베리파이1에 저장된 정보를 보내는 함수 |
| Send\_Pi2\_Info() | 함수 |  | void | 라즈베리파이2에 저장된 정보를 보내는 함수 |
| Send\_Pi3\_Info() | 함수 |  | void | 라즈베리파이3에 저장된 정보를 보내는 함수 |
| Send\_Pi4\_Info() | 함수 |  | void | 라즈베리파이4에 저장된 정보를 보내는 함수 |
| Reciv\_Pi1\_Info() | 함수 |  | void | 라즈베리파이1에 저장된 정보를 받기 위한 함수 |
| Reciv\_Pi2\_Info() | 함수 |  | void | 라즈베리파이2에 저장된 정보를 받기 위한 함수 |
| Reciv\_Pi3\_Info() | 함수 |  | void | 라즈베리파이3에 저장된 정보를 받기 위한 함수 |
| Reciv\_Pi4\_Info() | 함수 |  | void | 라즈베리파이4에 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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **클래스 이름** | **클래스 설명** | **요소 이름** | **함수/변수** | **타입** | **타입(반환값)** | **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 제어하는데에 필요한 정보를 관할하는 클래스 | A\_Cros | 변수 | Boolean |  | A\_구역에서 횡단보도를 지나고있는 보행자의 유무 |
| A\_LCancel | 변수 | Boolean |  | A\_구역에서 좌회전의 생략유무 |
| B\_Cros | 변수 | Boolean |  | B\_구역에서 횡단보도를 지나고있는 보행자의 유무 |
| B\_LCancel | 변수 | Boolean |  | B\_구역에서 좌회전의 생략유무 |
| C\_Cros | 변수 | Boolean |  | C\_구역에서 횡단보도를 지나고있는 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. **시퀀스 다이어그램**

1. 임베디드 기기의 카메라로 한 코너의 도로, 횡단보도, 보행자 대기장소를 촬영

2. 촬영되는 영상 데이터를 임베디드 기기에 전송

3 - 4. 영상처리를 통해 보행자 및 차량의 수, 유무를 인식

5 - 6. 영상처리를 거친 데이터를 인공지능을 통해 보행자의 위치 및 이동 방향, 차량의 대기 상태를 획득

7. 영상처리와 인공지능을 통해 얻은 데이터를 서버로 전송

8. 서버는 실시간으로 정보를 업데이트하며, 클라이언트에게 정보를 전달

9. 클라이언트는 얻은 정보를 통해 교통신호를 제어

1. **구현 사항**
   1. 하드웨어 인터페이스 (Hardware Interfaces)
      1. 서버
         * + CPU: Intel® Core™ i7-5820K
           + Memory: DDR4 16GB PC4-17000
           + HDD: SATA(Serial-ATA) 4TB / 7200RPM
      2. 라즈베리파이 3B+ (Raspberry Pi 3 Model B+)
         * + SoC: BCN2837B0
           + GPU: Dual Core VideoCore IV® Multimedia Co-Processor
           + Speed: 1,400MHz
           + RAM: 1GB
           + USB Port: 4 EA
           + Ethernet: 1000Base-T
           + Wireless LAN: 802.11 ac/n
   2. 소프트웨어 인터페이스 (Software Interfaces)
      1. 서버
         * + OS: Linux
           + Tool: PyCharm
      2. 라즈베리파이 3B+
         * + OS: Raspbian