|  |
| --- |
|  |
| **제18회 임베디드SW경진대회 개발완료보고서**  **[자유공모]** |
|  |

**□ 개발 요약**

|  |  |
| --- | --- |
| **팀 명** | THE FACO |
|  | |
| **작품명** | 대중교통(광역버스) 대기인원을 판별, 계산 후 더 정확한 최소시간 경로 지원 시스템 |
| **작품설명**  **(요약)** | 대중교통의 대기인원을 판별하기 위해 Open CV 를 이용하여 평균적인 대기인원의 모습을 정리하여 인원수를 판별하고, Open API 를 통해 알아낸 대중교통 정보를 대중교통 환승 장소의 대기 인원수와 함께 활용하여 경로를 추천한다. |
| **소스코드** | https://github.com/woojin1027/THE-FACO |
| **시연동영상** | < YouTube 링크 / 1개의 동영상 링크만 삽입 > |

**□ 개발 개요**

※ 프로젝트 개발 배경, 동기, 목표, 필요성 등 / 2page 이내로 작성

○ 개발 작품 개요 / 제안하는 작품에 대한 개요를 자세히 기술한다.

- 현재의 대중교통 길 찾기 앱들 중, 한국인의 대표적인 앱 두가지를 살펴보면, 약간의 경로추천 차이가 존재하기는 하지만, 최단 경로라는 것은 단지 대중교통을 이용할 때의 예상시간을 이용하여, 길 찾기 결과를 보여준다.



- 이러한 경로 계산은 평상시에는 문제가 되지 않지만, 사람들이 많이 몰리는 출 퇴근시간 특히, 신도시에서 서울로 출근하는 인원들이 광역버스를 이용할 때, 문제가 발생한다. 여러 정거장에서 이미 많은 대기 인원으로 인하여 버스들이 만원인 상태이므로 해당버스를 탑승하기 어려운 상황을 맞이하지만, 이를 길 찾기 앱에서는 확인해주지 않는다.

○ 개발 목표 / 개발 목표를 명확하게 제시한다.

- 기존의 대중교통 어플에서 제공하는 노선 안내와 대중교통 길 찾기 서비스를 넘어, 광역버스를 대기하는 인원수를 안내하고 이에 따른 경로 추천을 제공하는 것이 목표이다. 등하교 및 출퇴근 시간과 같은 탑승인원이 몰리는 시간에 발생하는 길 찾기 예상 소요시간의 오차를 줄일 수 있다.

- 코로나로 인해 사회적 거리두기 캠페인이 진행되고 있으나 출퇴근을 위한 좁은 정류장에서 사회적 거리두기를 실천하기 어렵다. 이 경우 사용자는 어플을 사용하여 대기 인원을 파악 후 대체 경로를 선택함으로써, 줄을 분산하여 설 수 있어 사람들 사이의 밀접 접촉을 줄이고자 한다.

○ 개발 작품의 필요성 / 개발하는 작품이 왜 필요한지에 대해 상세히 서술한다.

- 많은 직장인들과 학생들이, 각각 출퇴근 시간과 등하교 시간에 한 경로의 대중교통(광역버스)를 타기위해 긴 줄을 서며, 많은 시간을 줄을 서면서 보내게 된다. 실제로 대중교통을 이용하는 사용자들은 막연하게 긴 시간이 걸릴 것이라는 것만을 예상할 뿐 정확히 얼마나 기다려야 하는지, 몇 대의 버스를 대기 인원으로 인해 보내야 하는지 알지 못한다.

- 자신이 탑승할 환승 센터(정류장)에 직접 가지 않고서는 대기 인원이 얼마나 있는지도 알지 못하기에 한 정류장에서 대기할 때 감에만 의존해서, 지금 오는 버스들에 탑승 가능 한지를 판단한다. 어플을 사용한다면 직접 환승 센터(정류장)에 가지 않고서 정류장의 대기 인원을 알 수 있기 때문에 현재와 같은 코로나 사태로 인해 진행하는 사회적 거리 두기에도 긍정적 영향을 미칠 수 있다는 장점과 필요성을 설명할 수가 있다.

**□ 개발 환경 설명**

※ 최대한 자세하게 기술 / 15page 이내로 작성

○ Hardware 구성

- 라즈베리파이 4

- 카메라 센서 모듈

- paho mqtt

○ Hardware 기능 (제어 방법 등 서술)

- Raspberry PI 4 : 카메라 모듈을 통해 실시간으로 영상을 얻고 나면, 다음으로는 얻을 영상을 바탕으로 객체와 객체가 형성하는 줄을 인식하여 이를 판별해내는 과정을 라즈베리파이에서 진행한다.

앞서 설명한 바와 같이, 영상처리 과정은 PI의 CPU의 연산속도가 중요하기에 이를 해결할 수 있는 정도의 CPU 성능을 지닌 H/W 를 사용한다.

- Camera Module : 실시간 영상처리 과정을 위하여 영상을 확인할 카메라의 필요성을 얻게 되었다.

이를 위해 어느 정도의 화질을 보장하는 카메라 모듈을 이용하여 영상을 촬영해 영상에서의 객체를 판별하도록 한다.

- paho mqtt : 객체인식을 통해 얻은 데이터를 서버로 전송하기 위해 mqtt 를 이용하여 AWS 데이터베이스에 데이터를 제공함으로써 앱으로의 데이터 전송을 효율적으로 할 수 있게 한다.

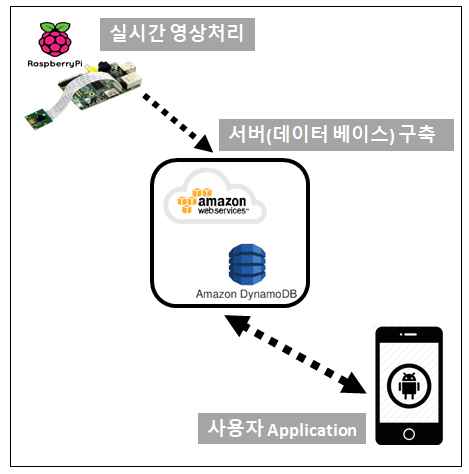
○ Software 구성

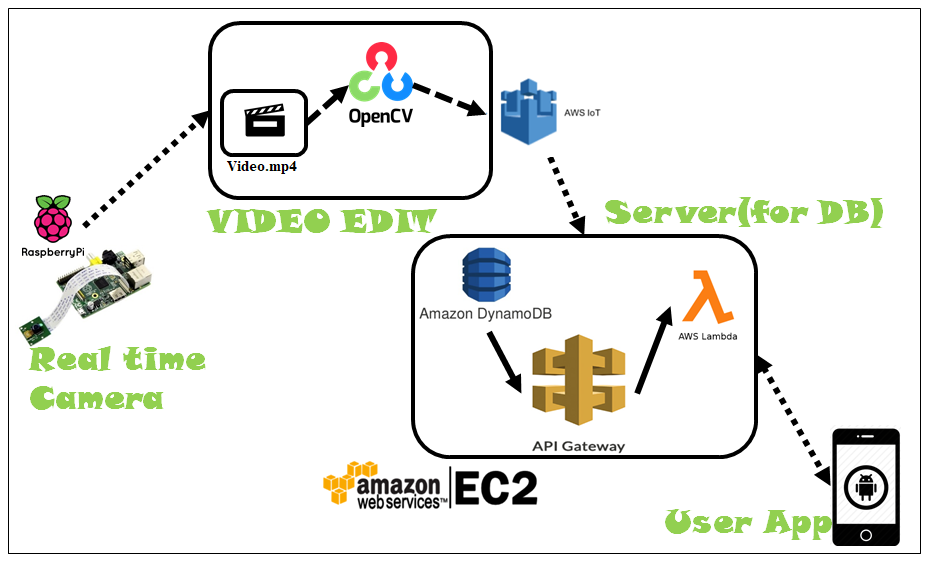
- Open CV

- Open API

- Smart Phone APP(Android APP)

○ Software 설계도





○ Software 기능 (필요 시 알고리즘 설명 포함)

- Open CV : 카메라 모듈을 통해 얻은 실시간 영상을 바탕으로 영상 내에서 사람 객체의 인식과 동시에 줄을 서고 있는 인원을 파악하여 데이터화를 진행한다.

- AWS DynamoDB : 영상처리한 데이터를 앱으로 보내기전에 데이터를 담아두는 역할을 한다.

- AWS API Gateway : AWS DynamoDB에 담겨진 대기인원에 대한 데이터를 앱에서 rest api 방식으로 호출하기 위해 사용한다.

- AWS Lambda : AWS API Gateway 를 사용하기 위한 간단한 rest api 함수를 get방식으로 앱에서 호출할 때 사용한다.

- Open API : 현재 대중교통의 실시간 파악을 위해 Open API 를 이용하여 현재 해당 노선의 정보를 받아와 H/W를 통해 얻는 자료들과 함께 연산을 진행하는데 이용한다.

- Smart Phone APP : 사용자가 편리하게 이용하기 위한 서비스를 제공할 때 UI 분야에 대한 중요성은 누구나 알고 있으리라 생각한다.

따라서 위 프로젝트 또한 구현화 과정으로 인한 APP 개발이지만 위 프로젝트에 대한 서비스를 이용할 때 가장 이용하기 편리하게 느껴질 UI를 구성한다.

○ 프로그램 사용법 (Interface)

-

-

○ 개발환경 (언어, Tool, 사용시스템 등)

- OS : 라즈비안, 안드로이드

- 개발환경(IDE) : MS Visual Studio Code, Android Studio

- 개발도구 : Android SDK, Java SDK, Maps SDK for Android

- 개발언어 : Python, Java, Node JS

**□ 개발 프로그램 설명**

※ 최대한 자세하게 기술 / 8page 이내로 작성

○ 파일 구성

-

-

○ 함수별 기능

-

-

○ 주요 함수의 흐름도

-

-

○ 기술적 차별성

-

-

**□ 개발 중 발생한 장애요인과 해결방안**

※ 개발 과정에서 나타났던 모든 장애 요인(Risk)들을 나열하고, 이러한 장애요인들이 발생했던 경우 어떻게 해결했는지 구체적으로 제시 / 1page 이내로 작성

○ 프로젝트 설계 측면

- 최초의 프로젝트를 진행할 때, 모았던 인원들 중에서 중간에 하차하는 인원들이 발생함으로써 발생하는 인원 공백과 이로 인한 프로젝트 분량의 재분배를 확인하게 되었다. 이러한 문제는 최초의 팀 구성부터 프로젝트의 분량대비 인원의 부족이 문제점으로 꼽혔던 우리 팀의 입장에서는 상당한 문제점으로 작용하였다. 하지만 이러한 문제점을 결국에는 팀원들과 분배에 대해 서로 의논하며, 최종적으로는 성공적으로 마무리 짓게 되면서 문제점을 해결할 수 있었다.

○ 작품 개발 측면

- 영상처리를 이용하기 위해, OPEN CV를 이용하였는데 이론적으로 생각하였던 알고리즘의 이용이 어려움 상황을 맞이하였다. 이유는, 하드웨어의 연산을 라즈베리파이를 이용하여 알고리즘의 연산을 진행하려 하였는데 이를 이용하기 위해서는 상당한 연산처리 속도가 필요한데, 이를 라즈베리파이로는 감당하기 어려운 상황을 맞이하였다. 이는, 실제 CCTV에서도 먼저 연산을 진행하여 검출한 인원파악만을 데이터화하여 전송하려는 프로젝트의 입장에서 문제점으로 다가오게 되었다. 이때, 연산처리가 가능한 정도의 성능의 알고리즘을 알아내어 이를 이용함으로써 프로젝트의 이론적 계획과 현실에서의 문제점을 확인, 해결할 수가 있었다.

- 공공 API의 경우, 버스의 API가 종종 업데이트되면서 해당 프로젝트로 구현화 하려던 해당 정류장의 API ID가 변화하거나, 없어지고, 특히나, 최근 들어 코로나 바이러스(C19)로 인하여 극단적으로는 노선의 변화가 생김으로써 발생하는 문제점이 발생하였다. 이러한 문제점은 프로젝트의 진행에서 구현화한 노선의 변경을 해야 할 정도의 큰 문제점이었는데, 이러한 문제점들을 해결하기 위해, 공공데이터 포털에 여러 번 문의를 하고, 최대한 많은 이용객이 이용하는 노선을 구현화하여 변경의 가능성을 최소화하는 노선으로 프로젝트의 구현화를 진행하는 등 문제점을 해결하기 위해 노력하였다.

-

○

-

**□ 개발결과물의 차별성**

※ 개발한 결과물과 기존 발표된 유사작품(제품) 간 차별성 및 우수성 설명 / 1page 이내로 작성

○

-

-

-

○

-

-

-

○

-

-

**□ 개발 일정**

※ 실제 프로젝트 개발 일정 작성 / 1page 이내로 작성

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **내용** | **2020年** | | | | | | | | | | | | | | | |
| **6月** | | | | **7月** | | | | **8月** | | | | **9月** | | | |
| 1 | 기능 설계 및 분석 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 기능 구현 및 개발 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 시험 평가 및 테스트 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**□ 팀 업무 분장**

※ 프로젝트 개발 관련 팀원의 업무 분장 위주로 작성 / 1page 이내로 작성

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **구분** | **성명** | **참여인원의 업무 분장** |
| 1 | 팀장 | ○○○ |  |
| 2 | 팀원 | ○○○ |  |
| 3 | 팀원 | ○○○ |  |
| 4 | 팀원 | ○○○ |  |
| 5 | 팀원 | ○○○ |  |