

# 제18회 임베디드SW경진대회 개발완료보고서

## [Smart Things]

### □ 개발 요약

팀 명	주차의 민족
	<div style="text-align: center; border: 1px solid black; padding: 5px; margin-bottom: 10px;">IoT 영역</div>  <p>The diagram illustrates the IoT system architecture. It is divided into three main functional areas: <b>영상분석 영역 (Image Analysis Area)</b>, <b>서버 영역 (Server Area)</b>, and <b>서비스 제공 영역 (Service Provision Area)</b>.</p> <ul style="list-style-type: none"> <li><b>영상분석 영역 (Image Analysis Area):</b> Contains three modules: <ul style="list-style-type: none"> <li><b>주차여부 판단 모듈 - M1:</b> A neural network for car presence detection, showing layers like convolution layer 1, 2, 3, and global average pooling.</li> <li><b>불법주차 차량 검출 모듈 - M2:</b> A neural network for illegal parking detection, showing a 'illegal car count' output of 2.</li> <li><b>번호판 인식 모듈 - M3:</b> A neural network for license plate recognition, showing layers like CNN, Max Pooling, and Bi-LSTM+Dense.</li> </ul> </li> <li><b>서버 영역 (Server Area):</b> Contains: <ul style="list-style-type: none"> <li><b>RealTime Database:</b> A database storing car status data (e.g., car0, car1, car2).</li> <li><b>Storage:</b> A storage system for processed frames and photos.</li> </ul> </li> <li><b>서비스 제공 영역 (Service Provision Area):</b> Contains: <ul style="list-style-type: none"> <li><b>어플리케이션 - 사용자 전용 (App - User Only):</b> A mobile application for users to check parking status.</li> <li><b>웹 페이지 - 관리자 전용 (Web Page - Admin Only):</b> A web interface for administrators to view parking photos and analysis results.</li> </ul> </li> </ul> <p>The data flow is as follows: IoT CCTV cameras capture video, which is sent to IoT Gateways. The gateways process the video into frames and photos. These are then analyzed by the image analysis modules. The results are stored in the RealTime Database and Storage. Finally, the information is provided to users via the mobile app and to administrators via the web page.</p>
<b>작품명</b>	<b>파킹 스캐너(Parking Scanner)</b>
<b>작품설명 (요약)</b>	<p>본 시스템은 IoT CCTV를 활용한 딥러닝 영상처리 기반의 스마트 주차 관리 시스템이다. 이를 위해 시스템은 4가지 영역을 통해 주차장의 현재 상태를 수집하고 분석하며 사용자에게 해당 정보를 제공한다. IoT 영역은 라즈베리파이 보드를 기반으로 구현한 IoT CCTV를 사용하여 주차장 촬영을 수행한다. 촬영된 영상은 IoT Gateway에 전송되어 해당 영상을 로컬 저장소에 저장한다. 또한 전처리 수행 후 영상분석 영역에 전처리 된 결과를 전송한다. 영상분석 영역에서는 전처리 된 프레임을 입력 받아 딥러닝 모델을 통해 주차공간별 주차 여부 판단과 불법주차 차량을 단속한다. 분석된 결과는 서비스 제공 영역을 통해 사용자 및 관리자에게 제공된다.</p>
<b>소스코드</b>	<a href="https://github.com/ParkingScanner/2020ESWContest_SmartThings_5029">https://github.com/ParkingScanner/2020ESWContest_SmartThings_5029</a>
<b>시연동영상</b>	<a href="https://bit.ly/33IVRsn">https://bit.ly/33IVRsn</a>

## □ 개발 개요

### ○ 배경 및 동기

스마트시티란 정보통신(ICT) 빅데이터 기술 융합을 통해 각종 도시문제를 해결하며 도시 자원을 효율적으로 운영하는 도시 모델을 의미한다. 특히 교통 문제는 스마트시티의 5대 중점분야 중 하나로 고려되는 만큼 주요한 문제로 다뤄지고 있다. 이에 맞춰 서울시는 'Green Parking'\* 사업을 진행하고 있다. 'Green Parking' 사업은 IoT 센서 기반의 주차 관리 시스템을 통해 시민들에게 주차장의 정보를 제공하고 있다. 하지만 'Green Parking' 사업에서 사용하는 IoT 센서는 주차장의 칸마다 설치가 필요하여 설치 및 유지·보수 과정에 있어 번거롭다는 단점이 존재한다. 이러한 단점은 광범위한 주차공간을 촬영하는 CCTV의 특성을 이용하여 개선하는 것이 가능하다. 따라서 본 시스템은 IoT CCTV를 활용한 실시간 영상분석 기반의 스마트 주차 관리 시스템인 '파킹 스캐너'를 제안한다.

### ○ 개발 작품 개요

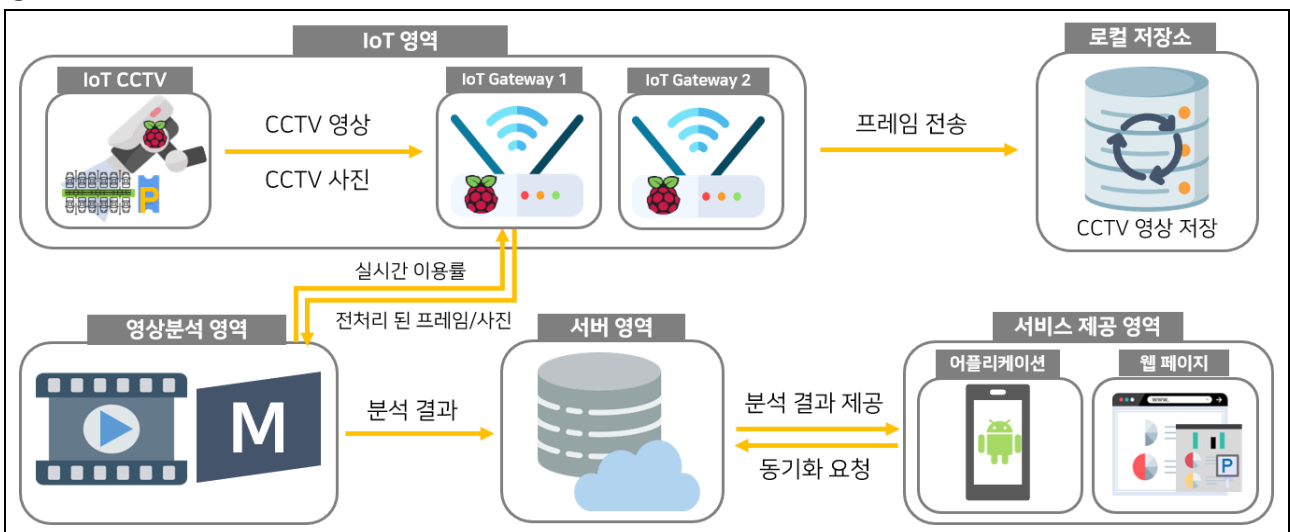


그림 1. 시스템 구조도

본 시스템은 IoT CCTV를 활용한 실시간 영상분석 기반의 스마트 주차 관리 시스템으로 분석된 결과를 주차장 사용자와 관리자에게 제공한다. 이를 통해 시민들의 주차 여건을 보장하며 교통 혼잡도를 개선한다. 또한 자동 불법주차 단속을 통해 단속에 필요한 비용 및 인력을 절감한다. 본 시스템은 4가지 영역으로 구성되며 각 영역의 기능은 다음과 같다.

- **IoT 영역** : IoT CCTV는 주차장 및 특별주차구역\*을 촬영한다. 촬영된 영상은 IoT Gateway에 전송된다. IoT Gateway는 전송받은 영상을 로컬 저장소에 전송하며 영상의 프레임을 전처리 후 영상분석 영역으로 전송한다.
- **영상분석 영역** : IoT Gateway에서 전송받은 전처리 된 프레임을 분석하여 각 주차공간의 주차 여부를 판단한다. 또한 특별주차구역과 주차구역 이외 공간에 주차된 불법주차 차량을 검출한다. 분석된 결과는 서버 영역에 전송한다.
- **서버 영역** : 영상분석 영역에서 전송받은 분석 결과와 불법주차 차량의 사진을 저장하고 해당 정보를 서비스 제공 영역과 공유한다.
- **서비스 제공 영역** : 안드로이드 어플리케이션은 주차장 사용자에게 주차장 이용률과 주차장의 기본 정보를 제공한다. 웹 페이지는 주차장 관리자에게 주차장의 이용률과 주차장 내 불법주차 차량 정보를 제공한다.

## ○ 개발 목표

### ■ IoT 영역

#### 1. IoT CCTV

##### □ IoT CCTV를 통한 실시간 영상 전송

주차장을 촬영하고 근거리 통신망을 통해 영상을 실시간으로 전송하는 IoT 기기를 개발한다.

#### 2. IoT Gateway

##### □ 병렬 연산을 통한 프레임 전처리 알고리즘

병렬 연산을 통해 프레임의 전처리 과정의 속도를 개선한 영상분석 알고리즘을 개발한다.

##### □ 주차 입력 프레임 간 유사도 비교 알고리즘

딥러닝 모델에 입력되는 영상의 입력 프레임 간 유사도 비교를 통해 딥러닝 모델의 분석 횟수를 줄인다. 이를 통해 효율적인 하드웨어 자원 운용을 위한 알고리즘을 개발한다.

### ■ 영상분석 영역

##### □ 주차장 이용률에 따른 영상분석 시점 조절 알고리즘

딥러닝 모델을 비교적 적은 빈도로 사용해 하드웨어 자원을 효율적으로 사용한다. 이를 위해 주차장 이용률에 따라 영상분석의 시점을 조절하는 알고리즘을 개발한다.

##### □ 주차공간별 주차 여부 판단 모델

주차공간별 주차 여부를 판단하기 위해 딥러닝 모델을 구현한다. 모델은 이미지 처리에 뛰어난 성능을 보이는 CNN\* 기반으로 구현한다.

##### □ 특별주차구역 번호판 인식 알고리즘

특별주차구역의 불법주차 차량 검출을 위해 번호판 검출 및 인식 알고리즘을 개발한다.

##### □ 주차구역 이외 공간 불법주차 차량 검출 모델

주차장의 혼잡도를 낮추고 정해진 주차구역 이외 공간에 주차된 차량을 검출하기 위해 불법주차 차량 검출 모델을 구현한다. 해당 모델은 YOLO 알고리즘을 이용하여 개발한다.

### ■ 서버 영역

##### □ Firebase\* 기반 데이터베이스 구축

불법주차 차량의 사진과 영상분석 영역의 처리결과를 저장하며 서비스 제공 영역의 실시간 동기화를 위해 Firebase를 사용하여 데이터베이스를 구축한다.

### ■ 서비스 제공 영역

#### 1. 어플리케이션(주차장 사용자 전용)

주차장 사용자를 대상으로 주차장의 잔여 주차공간 및 이용률을 실시간으로 제공하기 위해 Android OS를 기반으로 개발한다.

#### 2. 웹 페이지(주차장 관리자 전용)

주차장 관리자를 대상으로 실시간 주차 현황 및 불법주차 차량의 정보를 제공하기 위해 HTML/JavaScript를 기반으로 개발한다.

# □ 개발 환경 설명

## ○ Hardware 구성

### 1. IoT CCTV

본 시스템은 IoT CCTV를 통해 촬영된 주차장의 영상을 분석하고 분석 결과를 사용자에게 제공한다. 따라서 영상 전송과 주차공간 데이터를 수집하기 위한 IoT CCTV 제작이 필요하다. 이를 위해 임베디드 보드인 라즈베리파이 3B+에 카메라 모듈인 'RPI 8MP CAMERA BOARD'를 이용하여 IoT CCTV를 제작한다. IoT CCTV의 내장된 Wi-Fi 모듈을 이용해 촬영된 영상을 IoT Gateway로 전송한다. IoT CCTV의 회로도 및 외관은 아래와 같다.

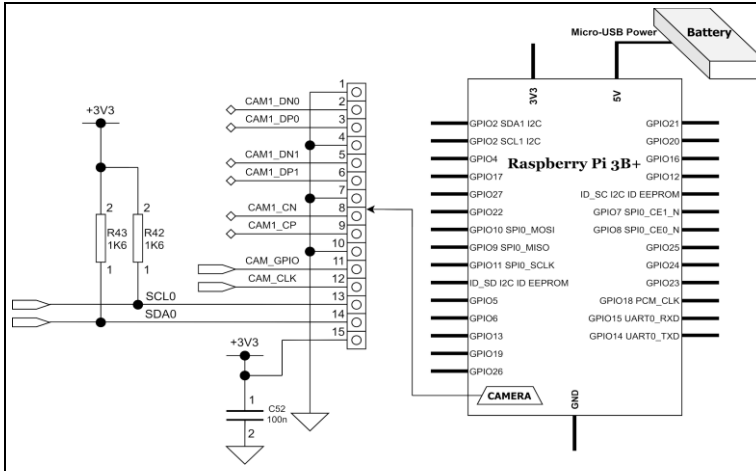


그림 2. IoT CCTV 회로도



그림 3. IoT CCTV 외관

### 2. IoT Gateway

IoT Gateway는 전송받은 영상을 로컬 저장소에 전송한다. 또한 전송받은 영상의 프레임을 전처리 후 영상분석 영역으로 전송한다. IoT Gateway는 주차구역을 촬영하는 IoT CCTV에서 전송된 영상을 처리하는 Gateway 1과 특별주차구역의 IoT CCTV에서 전송된 영상을 처리하는 Gateway 2로 구분된다. IoT Gateway는 임베디드 보드인 라즈베리파이 3B+를 이용하여 구현한다.

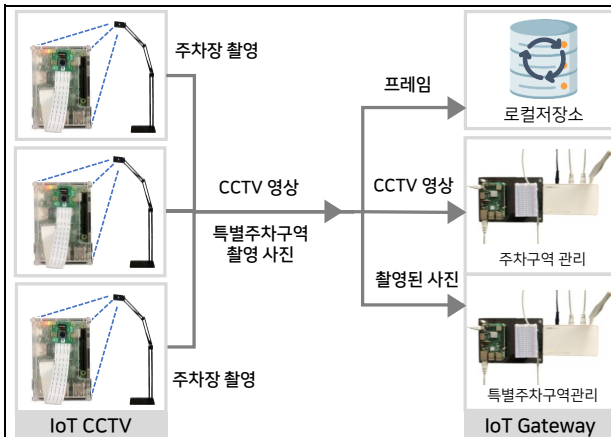


그림 4. IoT Gateway 개념도

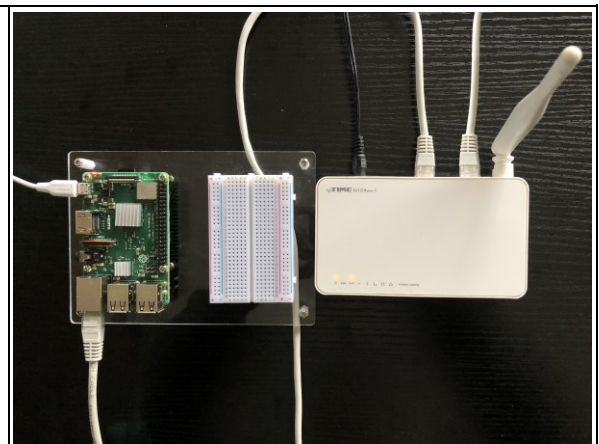


그림 5. IoT Gateway 외관

○ 데이터 수집

영상분석 영역의 딥러닝 모델 학습을 위해 IoT CCTV를 설치하여 데이터를 수집했다. 수집 대상은 주차구역과 차량 번호판 데이터이다.

첫 번째, 주차구역 데이터는 한림대학교 공학관 주차장에 각 주차구역별 1대씩 총 4대의 IoT CCTV를 설치하여 데이터를 수집했다.

두 번째, 차량 번호판 데이터는 한림대학교 주차장에 주차된 차량을 대상으로 IoT CCTV와 모바일 기기를 사용하여 수집했다. 데이터 수집은 2020년 5월부터 7월까지 약 3달간 진행했다. 수집한 데이터는 영상분석 영역의 딥러닝 모델 학습을 위해 사용되었다.

1. 주차구역 촬영 데이터 수집

- 수집 일자 : 2020년 5월 ~ 7월
- 수집 일시 : 차량의 이동이 잦은 시간대 (매주 08:00 ~ 19:00)
- 활용 모델 : 주차공간별 주차 여부 판단 모델  
주차구역 이외 공간 불법주차 차량 검출 모델

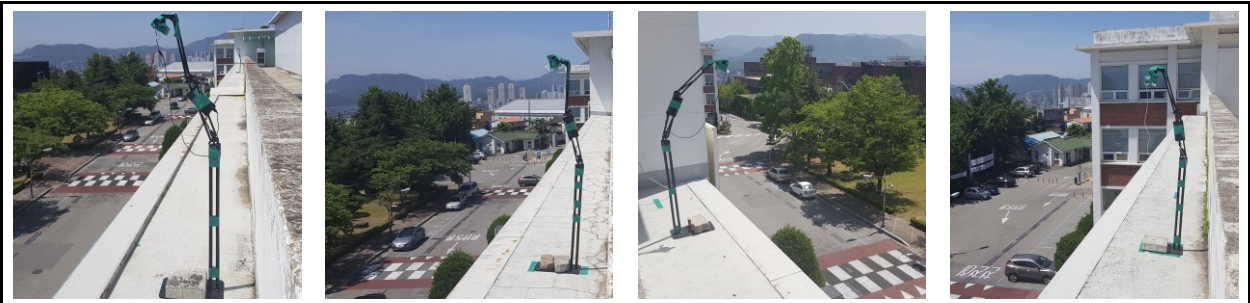


그림 6. 주차구역 IoT CCTV

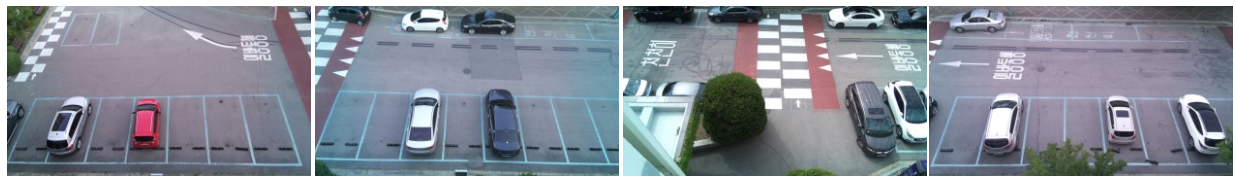


그림 7. 촬영된 주차구역별 사진

2. 차량 번호판 데이터 수집

- 수집 일자 : 2020년 5월 ~ 7월
- 수집 장소 : 한림대학교 교내 주차장
- 활용 모델 : 특별주차구역 번호판 인식 모델



그림 8. 특별주차구역 IoT CCTV



그림 9. 수집된 번호판 데이터

○ Software 구성

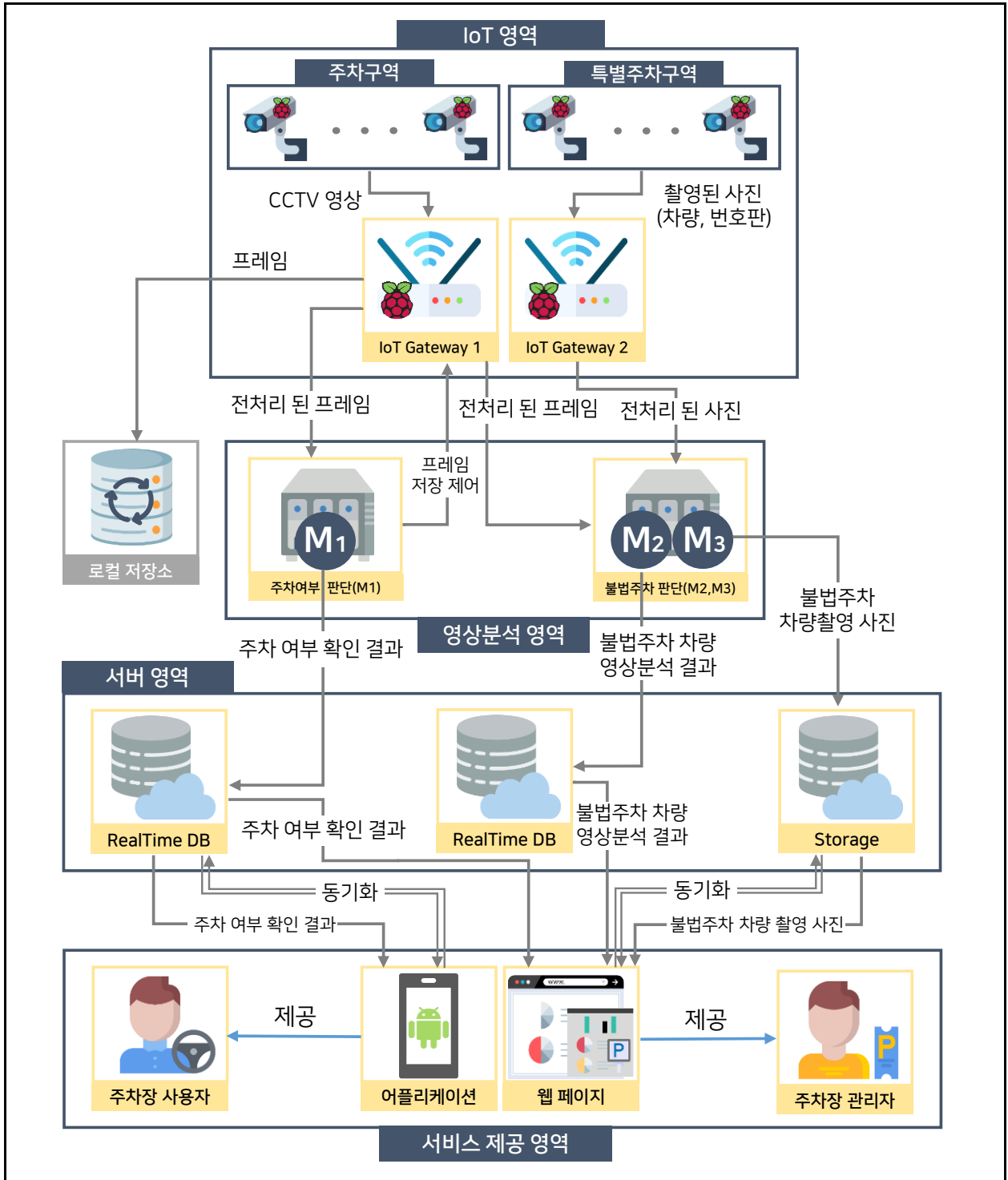


그림 10. 시스템 전체 아키텍처

본 시스템은 4가지 영역으로 구성되어 있다. 영역은 'IoT 영역, 영상분석 영역, 서버 영역, 서비스 제공 영역'으로 구성되어 있다. 영역별 기능은 아래와 같다.

영역		설명
IoT 영역	IoT CCTV (주차구역)	주차구역을 촬영하여 IoT Gateway 1로 영상을 전송
	IoT CCTV (특별주차구역)	특별주차구역을 촬영하여 IoT Gateway 2로 이미지를 전송
	IoT Gateway 1	전송받은 프레임 수신 및 로컬 저장소에 저장

		병렬처리를 활용한 전처리 수행 후 주차 여부 판단 모듈(M1)과 불법주차 차량 검출 모듈(M2)에 전송
	IoT Gateway2	전송받은 이미지에서 차량번호 검출 후 번호판 인식 모듈(M3)에 전송
영상분석 영역	주차 여부 판단 모듈(M1)	IoT Gateway 1에서 전송된 프레임을 분석해 주차공간별 주차 여부를 판단
	불법주차 차량 검출 모듈(M2)	IoT Gateway 1에서 전송된 프레임을 불법주차 차량 검출 모델을 활용해 주차구역 이외 공간에 주차한 차량 검출
	번호판 인식 모듈(M3)	IoT Gateway2에서 전송된 이미지에 대해 번호판 인식 모델을 통해 차량번호를 인식
서버영역	Firestore Database	영상분석 영역에서 분석된 주차 여부 판단 결과 저장
	Firestore Database	영상분석 영역에서 분석된 불법주차 차량 검출 결과 저장
	Firestore Storage	영상분석 영역에서 전송된 특별주차구역의 불법주차 차량 사진 저장
서비스 제공영역	어플리케이션	서버 영역에 저장된 정보를 실시간으로 동기화하여 주차장 사용자에게 제공
	웹 페이지	서버 영역에 저장된 정보를 실시간으로 동기화하여 주차장 관리자에게 제공

○ Software 설계도

■ 영상분석 영역

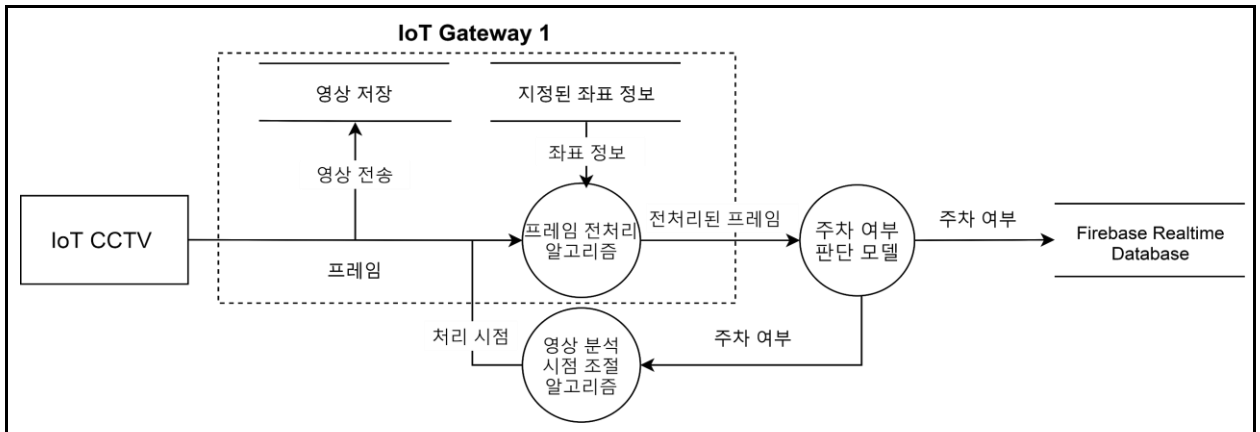


그림 11. 주차 여부 판단 모듈의 자료 흐름도

IoT Gateway 1은 IoT CCTV에서 전달받은 영상을 로컬 저장소에 전송하며 영상의 프레임을 전처리 후 영상분석 영역에 전송한다. 먼저 전처리를 수행하기 위해 지정된 좌표 정보를 불러온다. 이때, 프레임과 좌표 정보를 사용하여 병렬연산을 통한 전처리를 수행한다.

전처리 과정을 거친 프레임은 영상분석 영역으로 전송되며 해당 프레임을 통해 주차 여부를 판단한다. 판단 결과는 Firebase Realtime Database에 전송되어 저장된다. 또한 판단 결과를 통해

주차장의 이용률을 계산하고 이에 따른 주차공간별 전처리 시점을 조절한다.

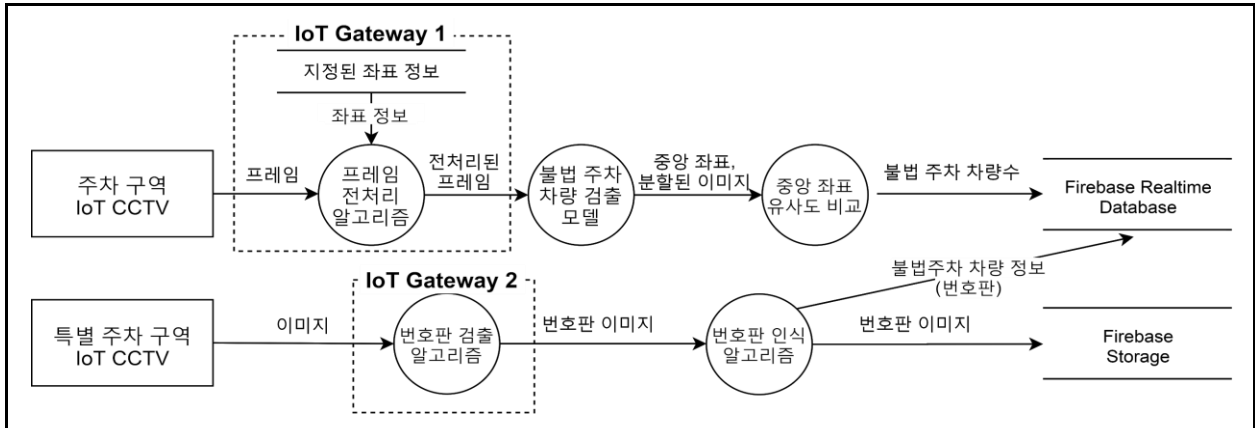


그림 12. 불법주차 차량 검출 모듈 자료 흐름도

IoT Gateway 1은 주차구역의 IoT CCTV로부터 촬영된 영상을 전송받는다. 이때, 주차공간의 지정된 좌표 값을 전달받아 영상의 프레임에 대해 전처리를 수행한다. 영상분석 영역에서는 전송받은 전처리된 프레임을 사용해 불법주차 차량을 검출한다. 검출된 불법주차 차량의 중앙 좌표와 이미지 간 유사도 비교를 수행하여 불법주차 차량의 수를 검출한다. 검출된 불법주차 차량의 수는 Firebase Realtime Database에 전송된다.

IoT Gateway 2는 특별주차구역의 IoT CCTV로부터 불법주차 차량의 이미지를 전송받는다. 이후 번호판 검출 알고리즘을 통해 전송받은 이미지에서 번호판을 검출한다. 검출된 번호판을 영상분석 영역에 전송하여 번호판 인식 알고리즘을 수행한다. 수행 결과에 따라 Firebase Realtime Database에 불법주차 정보를 저장한다. 또한 불법주차 차량의 정보인 번호판 사진을 Firebase Storage에 전송한다.

■ 서버 영역 : 데이터베이스 스키마

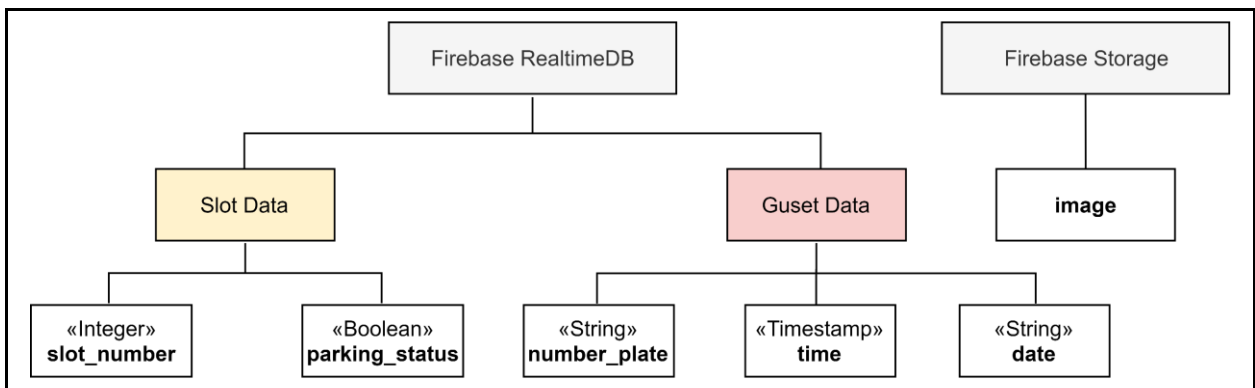


그림 13. Firebase 데이터 구조도

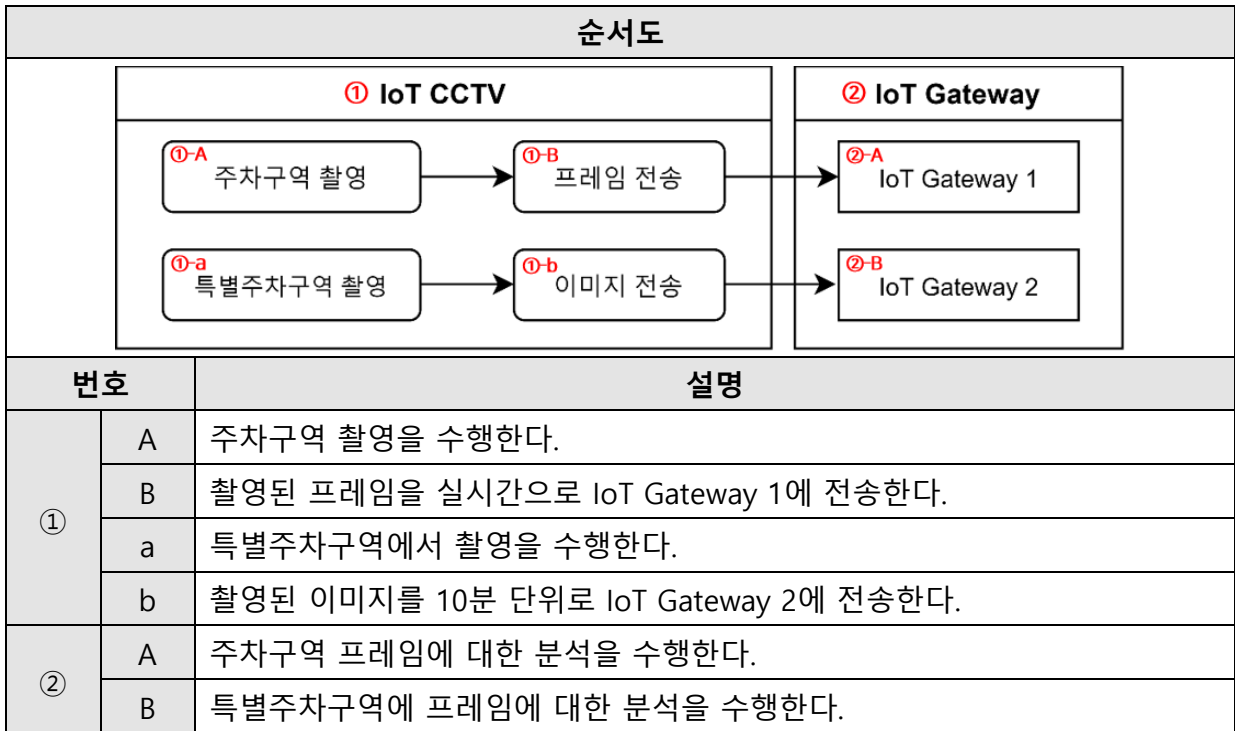
서버 영역의 데이터베이스는 Firebase RealTime Database와 Firebase Storage로 나누어진다. RealTimeDB는 영상분석 영역의 결과를 주차구역(Slot Data)과 특별주차구역(Guest Data) 객체로 나누어 저장한다. 주차구역 객체에는 영상분석 결과에 따라 각 주차공간의 주차 여부를 True(주차 가능), False(주차 불가)로 분류하여 저장한다. 분류 결과는 'parking\_status' 노드에 저장한다. 특별주차구역 객체에는 '번호판 인식 알고리즘'에서 검출된 불법주차 차량의 '차량번호(number\_plate)'와 차량이 주차한 시점의 '날짜(date), 시간(time)'을 저장한다. 촬영한 불법주차 사진은 Firebase Storage에서 별도로 관리된다.

○ Software 기능

■ IoT 영역

1. IoT CCTV

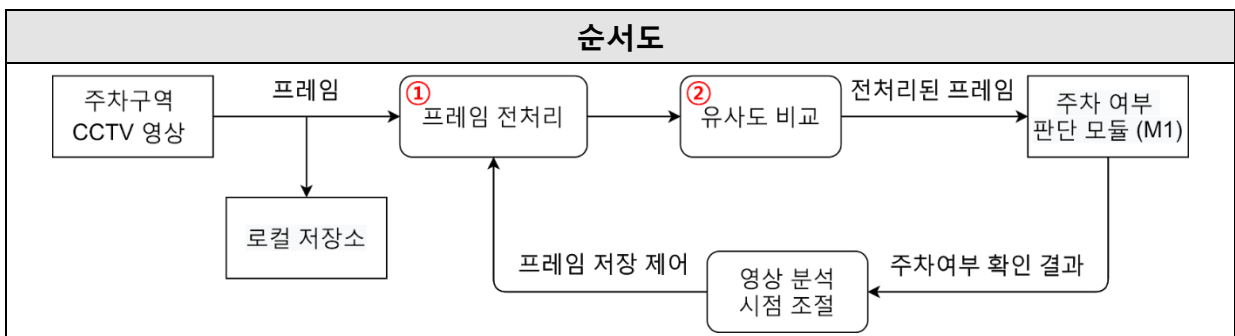
본 시스템은 주차장의 실시간 영상분석을 통해 주차 여부 판단 및 불법주차 차량 검출을 수행한다. 이를 위해 IoT CCTV를 통해 주차구역 및 특별주차구역을 촬영하고 촬영된 프레임 및 이미지는 소켓 통신을 통해 실시간으로 IoT Gateway로 전송된다. 순서도와 설명은 아래와 같다.



2. IoT Gateway 1

IoT Gateway 1은 주차구역에 설치된 IoT CCTV에서 전송된 영상을 관리한다. 먼저 주차구역의 IoT CCTV에서 전송된 영상을 수신한다. 수신된 영상을 로컬 저장소에 영상으로 전송한다. 또한 전처리 후 영상분석 영역의 모듈 M1, M2에 각각 전달한다. IoT Gateway 1의 각 전처리 순서도는 아래와 같다.

□ 영상분석 영역의 M1에 입력을 위한 프레임 전처리



1) 프레임 전처리 알고리즘

1-1) 프레임 내 주차공간 추출 알고리즘

이진 분류 모델은 다중 분류 모델에 비해 분류의 경우의 수가 적어 단순한 모델로도 좋은 성능을 낸다. 따라서 본 시스템은 이진 분류모델을 사용해 주차 여부를 판단한다. 이를 위

해 병렬연산을 통한 각 주차공간별로 프레임을 구분하는 전처리 알고리즘을 수행 후 주차 여부판단 모듈(M1)에 전송한다. 모델에 입력 전 전처리 과정의 순서와 과정의 중간 결과는 아래와 같다.

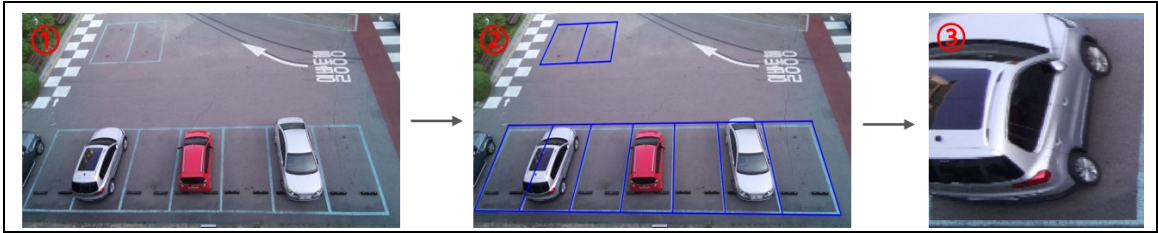
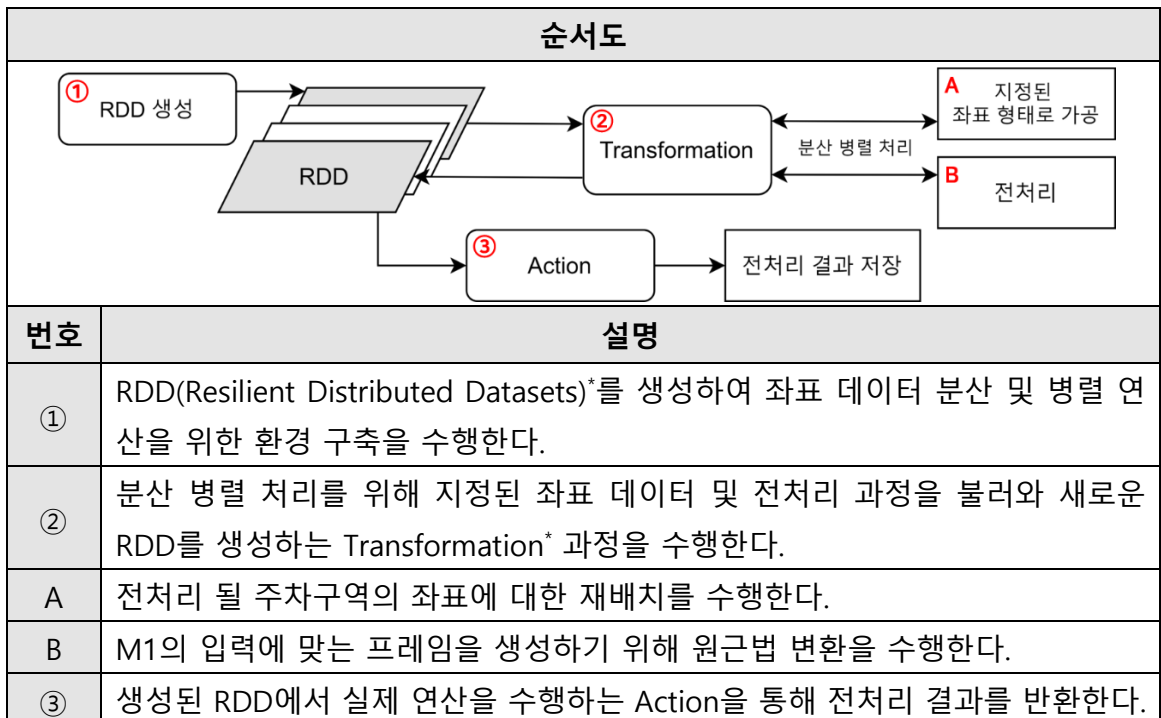


그림 14. 프레임 내 주차공간 추출 알고리즘의 중간 결과

번호	설명
①	전송된 프레임을 불러온다.
②	미리 정의된 주차공간의 위치 좌표를 통해 주차공간을 추출한다.
③	정방향 주차공간 프레임을 생성하기 위해 원근법 변환을 수행한다.

### 1-2) 병렬 연산을 통한 프레임 전처리 알고리즘

본 시스템은 실시간성 보장을 위해 병렬연산을 통한 전처리 알고리즘을 사용한다. 이를 위해 Spark를 통해 병렬연산 알고리즘을 구현한다. Spark를 통한 병렬연산의 내부 알고리즘의 순서도와 설명은 아래와 같다.



### 2) 유사도 비교 알고리즘

주차장의 특성상 같은 위치에 장시간 주차된 차량이 많다. 따라서 이전에 입력된 프레임과 현재 입력된 프레임이 중복되는 경우가 많다. 본 시스템은 딥러닝 모델의 사용 빈도를 줄여 하드웨어 자원을 효율적으로 사용하기 위해 유사도 비교 알고리즘을 수행한다. 유사도가 70% 이상이면 두 이미지가 동일하다고 판단한다.

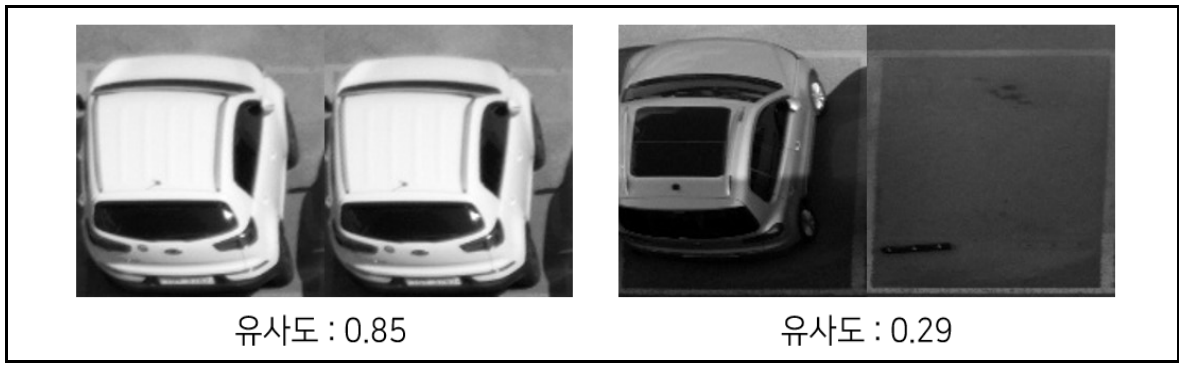
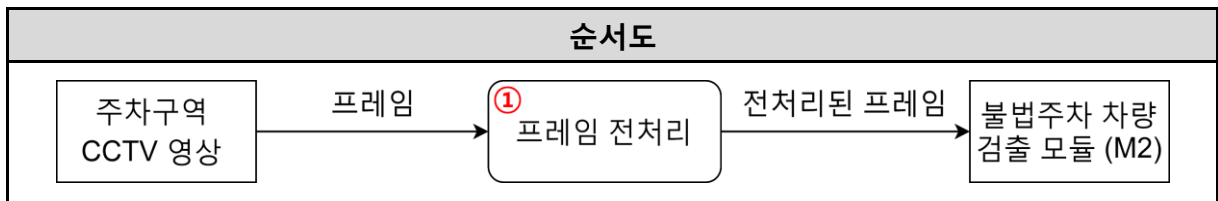


그림 15. 유사도 비교 결과

□ 영상분석 영역의 M2에 입력을 위한 프레임 전처리



1) 프레임 전처리



그림 16. 입력 프레임



그림 17. 전처리 후 프레임

본 시스템은 주차구역 이외 공간에 주차한 불법주차 차량을 검출한다. 이를 위해 주차공간을 제거한 후 불법주차 차량 검출 모듈로 전송한다. 그림 16은 모델에 입력되기 전의 입력 프레임이고 그림 17은 전처리 된 프레임이다.

3. IoT Gateway 2

IoT Gateway 2는 특별주차구역의 IoT CCTV에서 전송된 이미지를 전처리 후 영상분석 영역으로 전송한다. 10분 단위로 전송된 이미지에서 번호판을 검출해 영상분석 영역의 모듈 M3에 전송한다. 번호판 검출 과정과 설명은 아래와 같다.

□ 영상분석 영역의 M3에 입력을 위한 프레임 전처리

1) 프레임 전처리 알고리즘

특별주차구역의 IoT CCTV에 의해 이미지는 모듈 M3에 입력되기 전 10분 단위로 번호판 검출을 수행한다. 번호판 검출 과정은 아래와 같다.

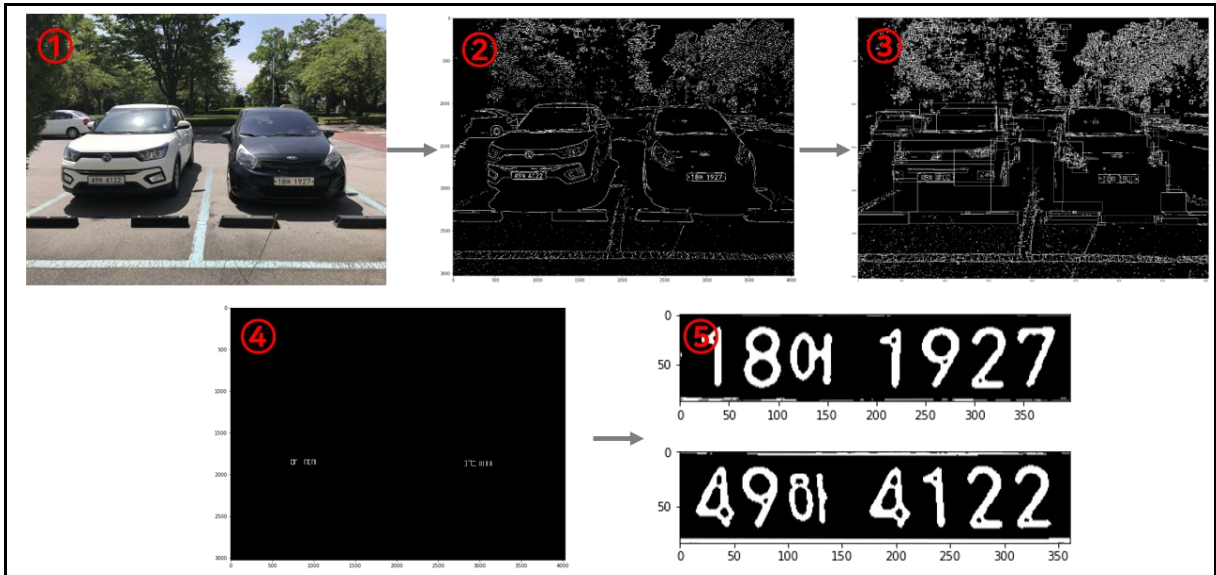


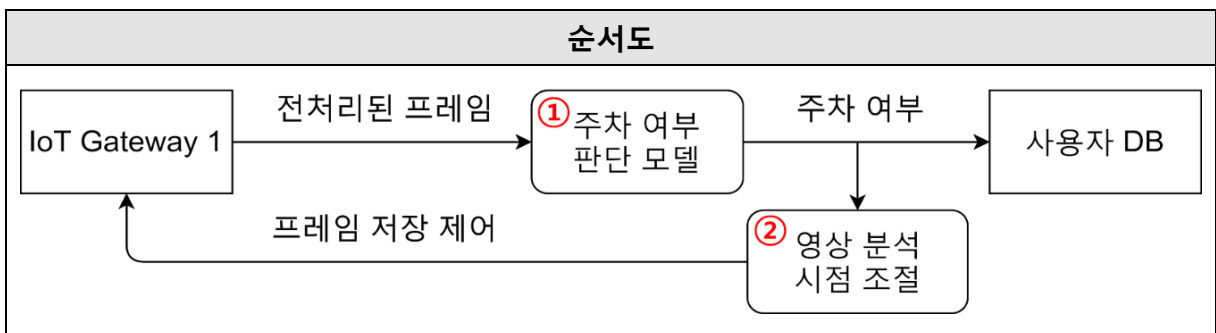
그림 18. 번호판 검출의 과정별 결과

번호	설명
①	중간값 필터를 사용해 입력 이미지의 노이즈를 제거한다.
②	연산의 복잡도를 줄이기 위해 흑백 변환을 수행한다.
③	이미지에서 모든 윤곽을 추출한다.
④	번호판 검출을 위해 모든 윤곽선 사이에서 거리, 각도, 크기를 비교한다.
⑤	검출된 번호판을 영상분석 영역으로 전송한다.

## ■ 영상분석 영역

### □ 주차 여부 판단 모듈(M1)

현재 주차장의 주차공간별 주차 여부를 사용자에게 제공하기 위해 주차 여부 판단 모듈을 통해 각 주차공간의 주차 여부를 판단한다. 주차 여부 판단 모듈의 알고리즘은 아래와 같다.



#### 1) 주차 여부 판단 모델

본 시스템은 주차공간별 주차 여부를 판단하기 위해 딥러닝 모델을 사용한다. 모델은 이미지 처리에 뛰어난 성능을 보이는 CNN 기반의 구조이다. IoT Gateway 1에서 전송된 전처리된 프레임을 분석해 주차공간의 주차 여부를 판단한다. 각 주차공간의 판단 결과는 서버 영역에 전송한다. 모델의 구조와 분석 과정은 다음과 같다.

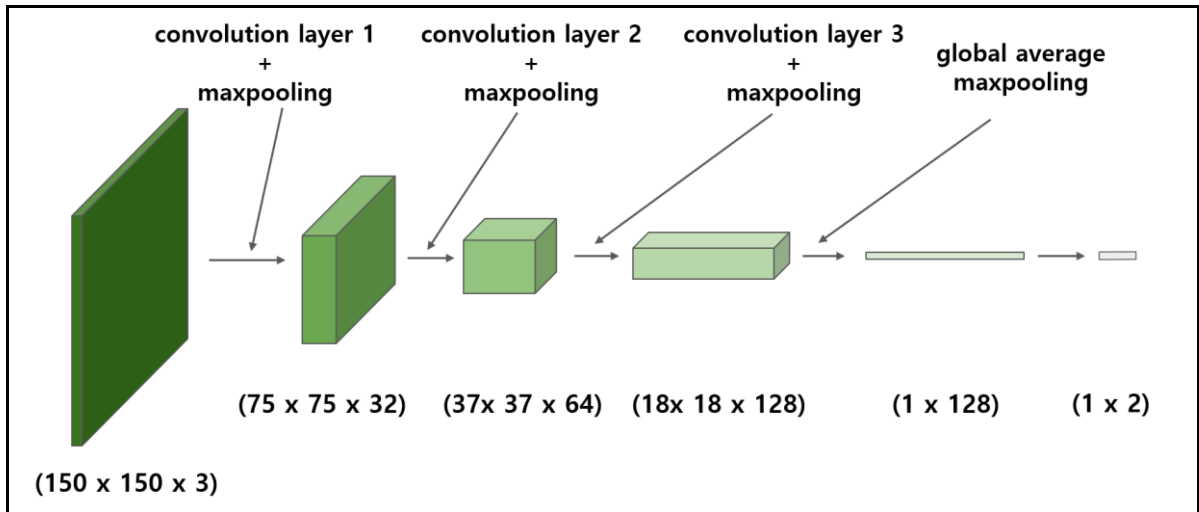


그림 19. CNN 기반 딥러닝 모델의 구조도

그림 19은 주차공간별 주차 여부를 판단하기 위한 딥러닝 모델의 구조이다. 먼저 3개의 CNN 층과 Max Pooling을 사용하여 입력된 데이터에서 특징을 추출한다. 최종 분류층에서 주차 여부를 판단한다. 판단된 주차공간별 주차 여부는 서버 영역으로 전달된다. 아래 그림은 모델의 학습 과정의 Loss와 Accuracy를 나타낸 그래프이다.

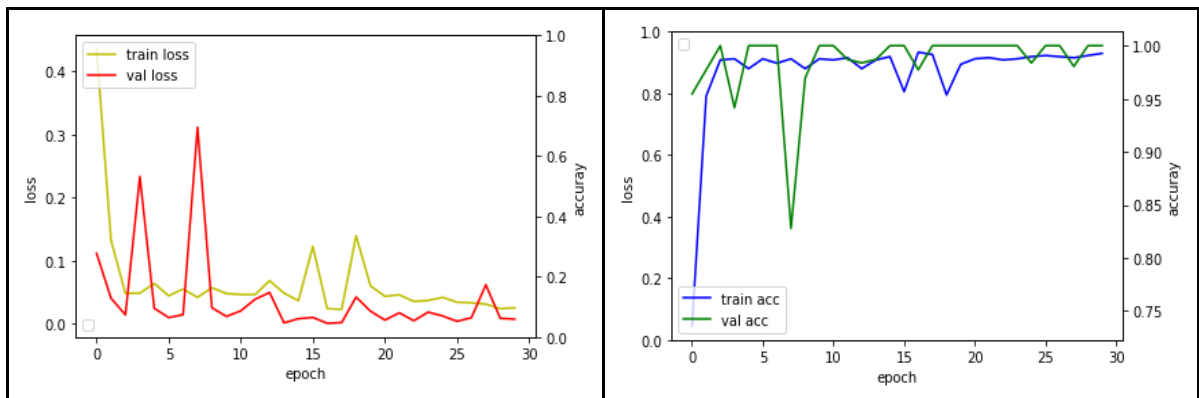


그림 20. CNN 기반 딥러닝 모델의 Loss 및 Accuracy

## 2) 영상분석 시점 조절 알고리즘

본 시스템은 주차구역의 분석 빈도를 줄이기 위해 영상분석 시점 조절 알고리즘을 구현한다. 각 주차장의 이용률을 계산하고 이에 비례해 분석 빈도를 조절한다. 이용률의 수식과 이에 따른 분석 시점의 기준 시간은 아래와 같다.

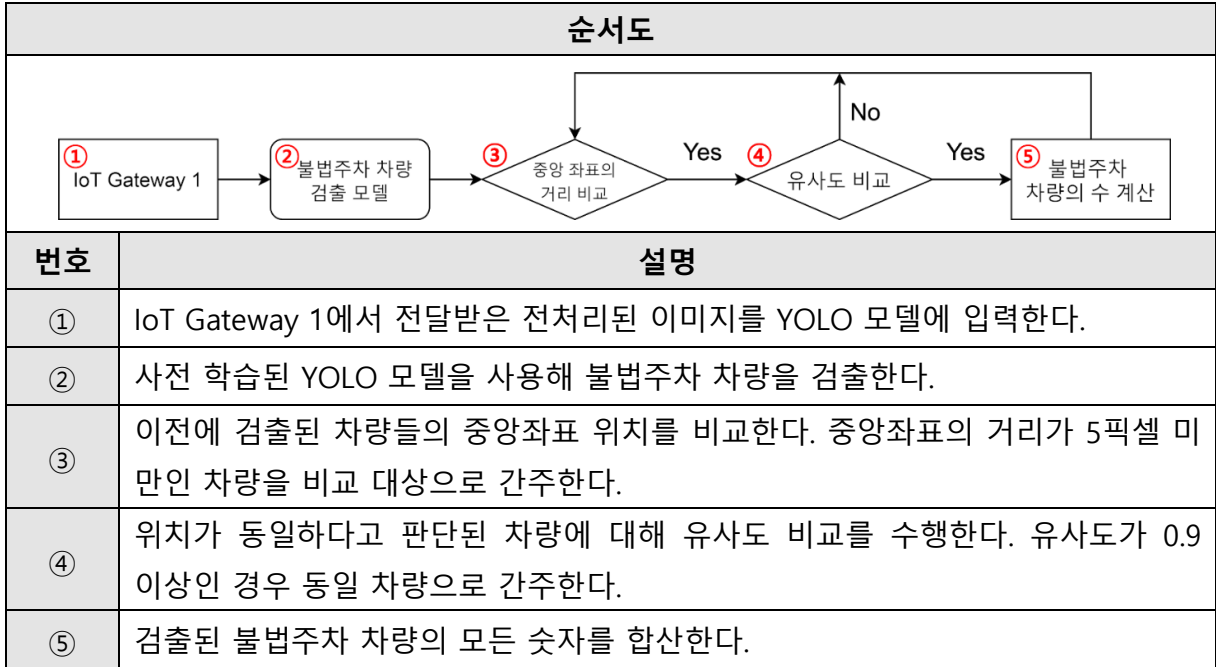
$$\text{이용률} = \frac{\text{주차된 주차구역수}}{\text{전체 주차 구역수}} \times 100(\%)$$

이용률	분석 시점 조절 시간
90% 이상	10초
70% ~ 90%	180초
50% ~ 70%	300초
50% 이하	600초

## □ 불법주차 차량 검출 모듈(M2)

주차구역 이외의 공간에서 발생하는 불법주차는 주차장 내부에 혼란을 가져온다. 주차장 내 불법주차로 인한 문제를 개선하기 위해 본 시스템은 불법주차 차량 검출 모델을 통해 불법주차 차량을 검출한다. 검출된 주차장 내 불법주차 차량의 수는 서버 영역에 전송된다.

본 시스템에서 주차구역 이외 공간에 정차한 차량들에 대해 '주차'로 판단하는 시간은 10분으로 정의한다. 따라서 10분마다 프레임을 전송받아 해당 알고리즘을 수행한다. 알고리즘의 순서도 및 설명은 아래와 같다.



**□ 번호판 인식 모듈(M3)**

본 시스템은 번호판 인식 모듈을 통해 특별주차구역에 주차한 불법주차 차량들의 번호판을 인식하여 불법주차 차량을 단속한다. 번호판 인식 모듈은 딥러닝 모델을 통해 IoT Gateway 2에서 전송된 전처리된 이미지의 차량번호를 인식한다. 차량번호는 서버 영역으로 전송된다. 차량번호 인식을 위한 딥러닝 모델의 구조와 설명은 아래와 같다.

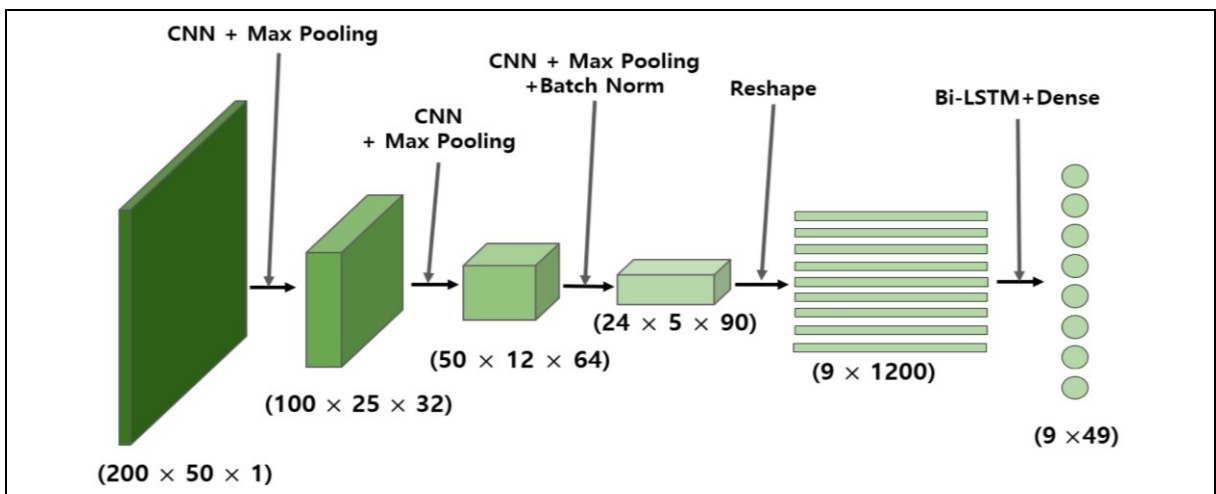


그림 21. 차량번호 인식 모델의 구조

그림 21은 검출된 번호판의 차량번호 인식을 위한 딥러닝 모델의 구조이다. 딥러닝 모델은 번호판의 고차원 특징을 추출하기 위해 CNN 기반의 층을 사용한다. 3개의 CNN 층을 통해 특징을 추출한 후 Bi-Directional LSTM\*을 사용해 차량번호를 인식한다.

○ UI 사용법

1. 안드로이드 어플리케이션 (주차장 사용자 전용)

안드로이드 어플리케이션은 주차장 사용자에게 주차장의 실시간 정보를 제공한다. 제공하는 정보는 '주차장 실시간 이용률, 세부 정보'이며 주차장의 실시간 잔여 주차공간은 사용자에게 실제 주차장과 동일한 구조로 시각화하여 보여준다. 어플리케이션의 기능은 아래와 같다.

1) 어플리케이션 전체 인터페이스

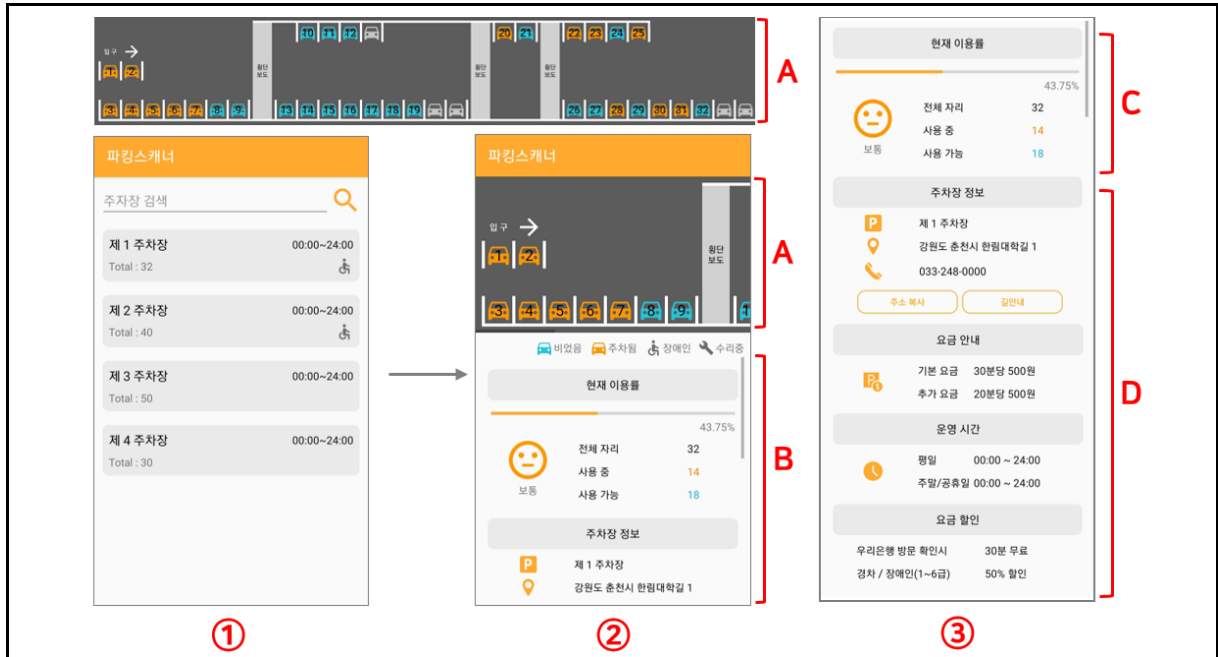


그림 22. 어플리케이션 사용자 인터페이스

번호	설명
①	어플리케이션의 접속 시 보여지는 화면으로 이용가능한 주차장의 목록이 보인다.
②	A 주차장 사용자가 원하는 주차장을 선택했을 때 보여지는 화면이다. 주차장의 '잔여 주차공간, 이용률, 기본 정보'가 보인다. 주차장의 실시간 잔여 주차공간은 실제 주차장의 구조와 동일하게 시각화 되어있다. 가로 스크롤바를 통해 주차장의 모든 주차공간을 볼 수 있다. (파란색: 잔여좌석 O, 주황색: 잔여좌석 X)
	B 주차장의 현재 이용률 및 주차장의 기본 정보가 보인다. 세로 스크롤바를 통해 주차장의 모든 정보를 볼 수 있다.
C	주차장의 현재 이용률이 실시간으로 보인다. 제공되는 정보는 '주차장의 전체 자리와 사용 중, 사용가능 자리' 이다. 좌측 이모티콘은 '여유, 보통, 혼잡' 3가지 경우로 나뉘어 보이며 이용률의 변화에 따라 표정 및 색상이 바뀐다. (여유: 파랑색 / 보통: 주황색 / 혼잡: 빨강색)
	D 주차장의 세부 정보가 보인다. 해당 화면에는 '주차장 이름, 위치, 전화번호, 요금, 운영시간, 요금 할인' 정보가 보이며 '주차장 주소복사', '길안내' 기능이 제공된다.

2) 주차장 안내 기능 - 주소 복사

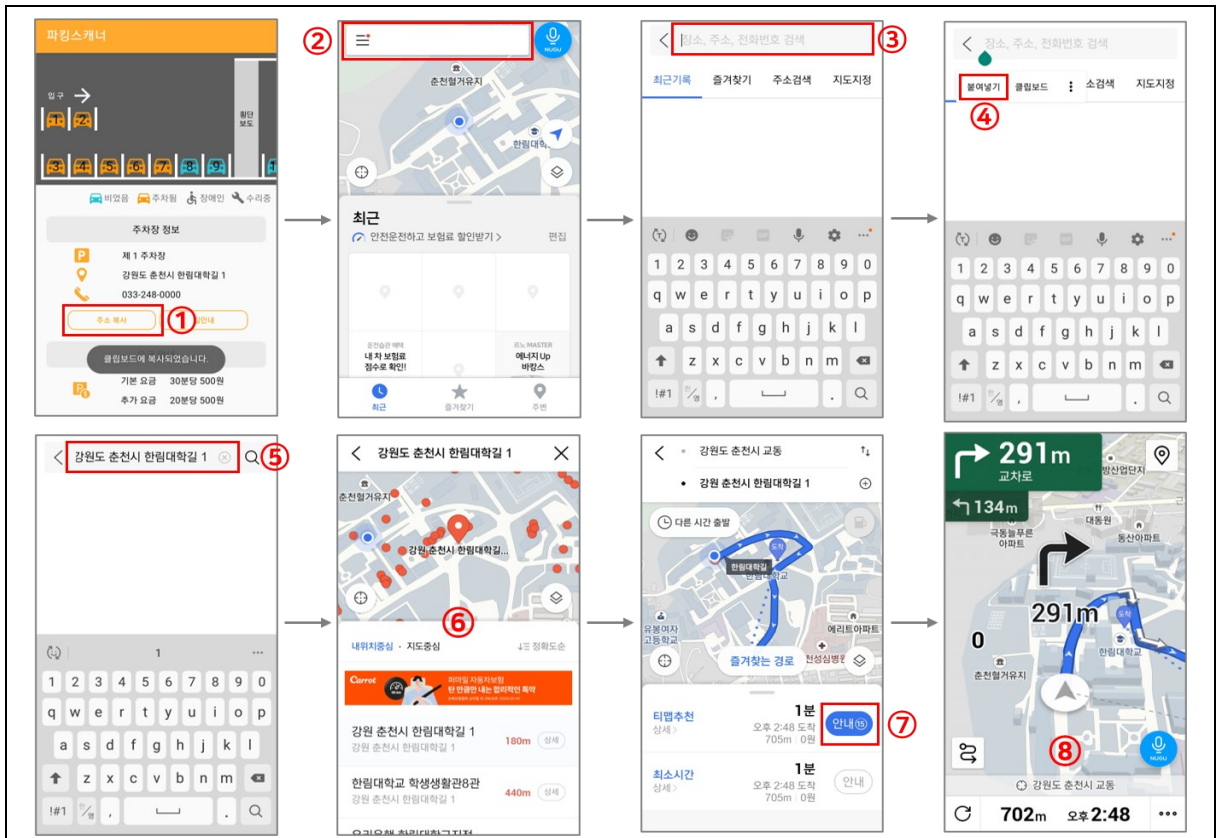


그림 23. 어플리케이션 주소 복사 기능

번호	설명
①	주차장 정보란의 '주소 복사' 버튼을 클릭하면 화면 하단에 실행 메시지가 나타나며 주차장의 주소가 복사된다.
②	길안내(내비게이션) 기능을 지원하는 지도 어플리케이션의 검색창으로 이동한다.
③,④,⑤	검색창을 길게 누른 후 '붙여넣기' 기능을 사용하여 복사한 주소를 붙여 넣는다.
⑥	해당 화면에서 잠시 대기한다. 이때 사용자의 현재 위치 정보를 받아옴으로써 목적지까지의 경로가 설정된다.
⑦	화면 우측 하단의 '안내' 버튼을 클릭한다.
⑧	사용자의 현재 위치부터 목적지까지의 길 안내 기능이 수행된다.

### 3) 주차장 안내 기능 - 길안내



그림 24. 어플리케이션 길안내 기능

번호	설명
①	'주차장 정보' 란의 '길 안내' 버튼을 클릭한다.
②	화면 하단에 실행 메시지가 나타남과 동시에 지도 어플리케이션인 'Tmap'이 자동 실행된다.
③	사용자의 현재 위치 정보를 받아옴으로써 목적지까지의 경로가 설정되고 경로가 표시된다. 이때 길 안내를 원하면 화면 우측 하단의 '안내' 버튼을 클릭한다.
④	설정된 경로대로 목적지까지 길 안내 기능이 수행된다.

## 2. 웹 페이지 (주차장 관리자 전용)

웹 페이지는 주차장 관리자에게 '주차장의 현황과 불법주차 단속 기능'을 제공한다. 제공하는 정보는 '주차장의 실시간 이용률, 주차장의 불법주차 차량 수, 만차 알림'이다. 추가로 특별주차구역의 단속 기능을 제공한다. 불법주차 차량이 주차했을 경우 관리자에게 해당 차량의 정보를 제공한다. 웹 페이지의 기능은 아래와 같다.

### 1) 웹 페이지 전체 인터페이스

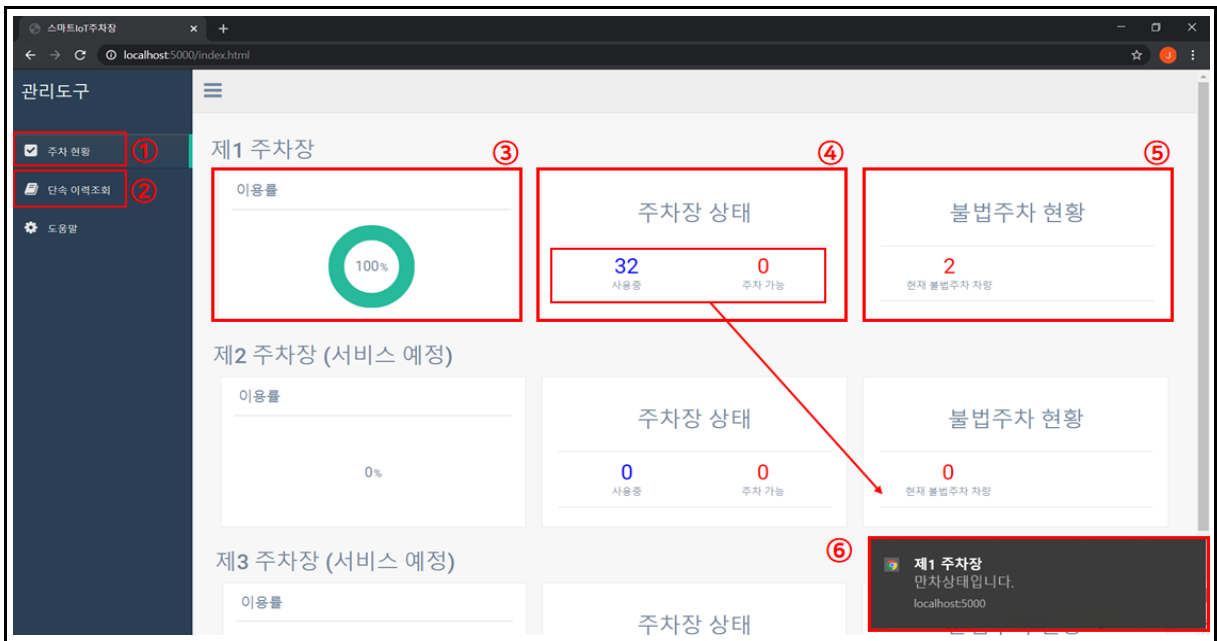


그림 25. 웹 페이지 관리자 인터페이스

번호	설명
①	'주차 현황'은 웹 페이지 접속 시 보여지는 메인 화면으로 주차장의 실시간 주차 현황을 제공한다.
②	'단속 이력조회'에는 특별주차구역의 불법주차 차량의 정보를 제공한다.
③,④	이용률은 실시간으로 주차장에 주차한 차량의 수를 기준으로 제공한다. 이용률은 원형 그래프 및 숫자 통계를 통해 시각화하며 실시간으로 업데이트된다.
⑤	주차구역 이외 공간에 주차된 차량 현황이 보인다.
⑥	관리되는 주차장이 '만차'인 경우 Windows 알림을 통해 관리자에게 전달한다.

### 2) 특별주차구역의 불법주차 관리 기능

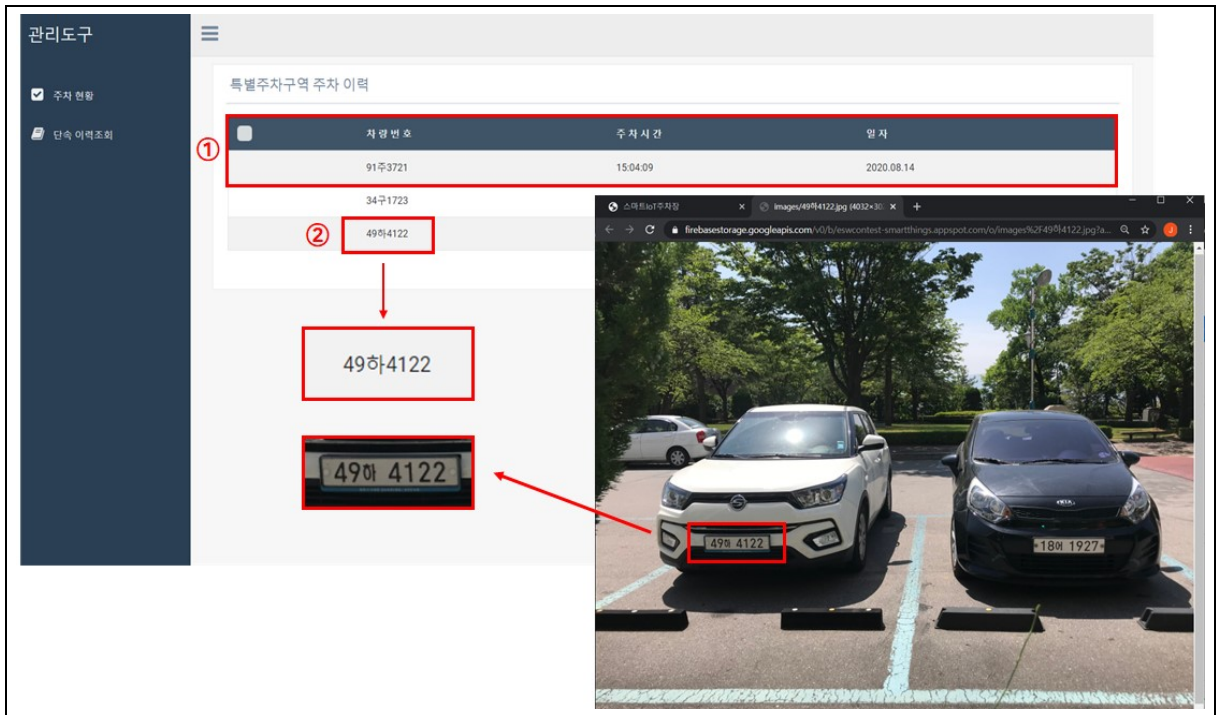


그림 26. 특별주차구역의 불법 차량 정보 제공 기능

번호	설명
①	특별주차구역의 주차이력에는 특별주차구역의 불법주차 차량 정보(주차시간, 차량 번호, 날짜)가 제공된다. 관리자는 제공되는 정보에 대한 열람이 가능하다.
②	열람하고자 하는 테이블을 선택하면 해당하는 불법주차 차량의 사진이 제공된다.

○ 개발환경 (언어, Tool, 사용시스템 등)

	IoT 영역	영상분석 영역	서비스 제공 영역	
			어플리케이션	웹 페이지
개발언어	Python	Python	JAVA	HTML, JavaScript
운영체제	Raspbian	Ubuntu 18.04	Android OS	Windows 10
프레임워크	Apache Spark			
라이브러리	Pyspark3.0 OpenCV Numpy 1.16	OpenCV Tensorflow2.1 Numpy1.16 YOLO v3	Firebase Android SDK	Firebase JavaScript SDK
개발 보드	RPI3B+			
데이터베이스	Firebase RealtimeDB, Google Cloud Storage			
버전 관리	GitHub			

## □ 개발 프로그램 설명

### ○ 파일 구성

핵심 파일들만 나열했으며 아래 나열된 파일과 전체 소스코드들은 첨부된 Github Repository URL을 통해 열람할 수 있다.

**Github URL** : [https://github.com/ParkingScanner/2020ESWContest\\_SmartThings\\_5029](https://github.com/ParkingScanner/2020ESWContest_SmartThings_5029)

### ■ IoT CCTV 영역

파일명	설명
IoT_CCTV1.py	주차구역 촬영 및 IoT Gateway 1에 실시간 프레임 전송을 위한 함수 저장
IoT_CCTV2.py	특별주차구역 촬영 및 IoT Gateway 2에 이미지 전송을 위한 함수 저장

### ■ IoT Gateway

#### 1. IoT Gateway 1

파일명	설명
cv2_compare.py	주차공간의 유사도 비교 후 임계값(70%) 미만의 주차공간을 반환
yolo_methods.py	M2 모듈 입력 전 프레임의 전처리를 위한 함수 저장
spark_methods.py	Spark를 통한 M1 모듈 입력 전 전처리를 위한 함수 저장
IoT_gateway1.py	주차구역의 IoT CCTV로부터 전송된 프레임을 로컬 저장소로 전송, 조절된 영상분석 시점에 따라 전처리 수행 후 영상분석 영역으로 전처리된 프레임을 전송하는 함수 저장

#### 2. IoT Gateway 2

파일명	설명
car_number_preprocessing.py	전송된 프레임에서 번호판을 검출하기 위한 함수 저장
IoT_gateway2.py	특별주차구역 IoT CCTV에서 이미지를 수신 및 전처리 수행 후 영상분석 영역으로 전송하기 위한 함수 저장

### ■ 영상분석 영역

#### □ 전처리된 프레임 수신 및 분석결과 전송

파일명	설명
main.py	각 모듈 제어를 위한 함수 저장
receive_client.py	IoT Gateway로부터 전처리 된 프레임을 수신하기 위한 함수 저장
firebase_update.py	영상분석 영역의 분석 결과를 서버 영역으로 전송하는 함수 저장

#### □ 주차 여부 판단 모듈(M1)

파일명	설명
car_observe2.py	IoT Gateway 1에서 전송받은 전처리 된 프레임을 사용해 주차구역

	별 주차 여부를 판단하는 함수 저장
--	---------------------

□ 불법주차 차량 검출 모듈(M2)

파일명	설명
darknet.py	darknet에서 정의한 함수 저장
park_other_place.py	차량 검출을 위한 yolo 모델을 불러오기 위한 함수 저장
yolo_methods.py	검출된 차량을 통해 주차구역 이외 공간의 불법주차 차량의 수를 측정하기 위한 함수 저장

□ 번호판 인식 모듈(M3)

파일명	설명
car_number_plate.py	IoT Gateway 2에서 전송된 번호판 이미지의 OCR 수행 함수 저장

■ 서비스 제공 영역 : 어플리케이션

파일명	설명
Car.java	주차공간에 부여되는 자리번호, 사용여부, 고유 키 값을 선언
CarAdapter.java	각 주차공간의 정보가 저장된 Listview의 viewHolder를 선언
MainActivity.java	어플리케이션의 layout 선언 및 클릭 이벤트 설정
ParkingLotActivity.java	주차장 layout과의 연동으로 어플리케이션에서 제공되는 모든 서비스 수행 / 데이터베이스와 연동해 값 변경 시 동기화 요청
activity_main.xml	어플리케이션 실행 시 가장 먼저 보여지는 인터페이스
activity_Parking_lot.xml	주차장 선택 시 보이며 어플리케이션의 모든 기능을 이용할 수 있는 메인 인터페이스

■ 서비스 제공 영역 : 웹 페이지

파일명	설명
index.html	웹 페이지 실행 시 초기 화면, '이용률, 불법주차 수' 정보 제공
numberList.html	특별주차구역에 주차된 차량 정보(차량번호, 시간, 날짜) 시각화
App.js	데이터베이스와 연동해 주차장 정보를 웹페이지와 실시간 동기화
config.js	데이터베이스에 접근하기 위한 key
Firebase-messaging-sw.js	Firebase Cloud Message를 통해 만차 여부에 따라 관리자에게 메시지 전송

○ 함수별 기능

■ IoT CCTV 영역

파일명	함수명	설명
IoT CCTV.py	video_camera	주차구역 및 특별주차구역 영상 촬영
	sending_thread	IoT Gateway로 프레임 및 이미지 전송

## 1. IoT Gateway 1

파일명	함수명	설명
IoT_gateway1.py	receive_pc	영상분석 영역의 주차구역별 '전송 시점 조절 알고리즘' 결과 수신
	control_send2pc_time	지정된 주차구역별 전송 시점에 따라 영상 분석 영역에 전송되는 주차구역의 번호 반환
	get_preprocess	주차구역 유사도 비교 결과에 따라 주차 구역 반환
	yolo_send	모듈 M2의 분석 시점에 따라 프레임 내 주차구역을 제거 후 영상분석 영역에 해당 프레임 전송
	send2pc_thread	영상분석 영역으로 전처리 된 프레임 전송
	send2db_thread	로컬 저장소로 프레임 전송
	pickle_to_data	pickle 파일을 다시 원래 데이터 형태로 변환
	receiving_thread	IoT CCTV영역에서 전송된 프레임 수신
comparing.py	compare_images	입력된 이미지의 유사도 비교 결과를 반환
	compare	유사도 비교 후 임계값 미만 주차공간 반환
yolo_methods.py	dark_box	입력 프레임의 주차공간 좌표를 통해 해당 프레임 내 전체 주차구역을 제거 후 반환
	make_dark_box	주차구역의 위치에 따라 해당 주차구역 제거

## 2. IoT Gateway 2

파일명	함수명	설명
car_num_preprocessing.py	blurring_img	중간값 · 임계값 블러링을 통한 노이즈 제거 수행
	finding_contours	입력 이미지의 윤곽선(contour)을 추출하고 해당 위치의 중앙 좌표 및 꼭지점 좌표를 반환
	removing_noise_contours	지정된 값에 따라 윤곽선의 크기 및 가로 세로의 비교를 통해 노이즈로 간주되는 윤곽선 제거
	find_chars	각 윤곽 가운데 번호판의 위치를 찾아 번호판으로 판별된 리스트를 반환
	car_number_result	영상분석 영역으로 전처리 된 프레임 전송
	search_image	검출된 번호판과 입력된 이미지를 반환
IoT_gateway2.py	pickle_to_data	데이터를 이진 값으로 변환
	sending_thread	검출된 번호판을 영상분석 영역으로 전송
	receiving_thread	특별주차구역 IoT CCTV로부터 이미지 수신

### ■ 영상분석 영역

#### □ 영상분석 및 데이터 송수신

파일명	함수명	설명
main.py	read_queue	queue에 저장된 key에 따라 데이터를 읽고 반환
	split_key_items	입력된 dictionary의 key와 value를 구분해 반환

	read_preprocessed	queue에 저장된 key에 따라 수신된 데이터 처리
receive_client.py	byte2data	binary 값을 원형 데이터로 변환 후 반환
	getkey	dictionary 값에서 키를 읽어 반환
	receive_thread	IoT Gateway에서 전송된 값을 구분해 저장하며 전송된 값을 구분하는 key를 queue에 저장
firebase_update.py	fileUpload	입력받은 이미지를 서버영역(storage)에 전송
	firebase_update	주차 여부를 입력받아 서버영역(Realtime Database)에 저장
	fire_base_carnum	불법주차 차량의 OCR 결과를 서버영역(Realtime Database)에 전송
	fire_base_illegal	불법주차 차량의 수를 서버영역(Realtime Database)에 전송

□ 주차 여부 판단 모듈(M1)

파일명	함수명	설명
car_observe2.py	process_image	처리시점이 된 섹션의 주차공간을 IoT Gateway 1에서 전달 후 car_observation을 통해 주차공간별 주차 여부를 판단하여 반환
	car_observation	정의된 주차 여부 판단모델의 구조와 가중치 경로를 입력받아 모델을 정의하며 전달받은 전처리된 프레임을 통해 주차 결과 판단 후 Realtime Database에 업데이트
	control_process_timing	각 섹션별 주차장 이용률을 계산하여 이용률에 따른 처리시점을 조절
	update_utilize	메모리에 저장된 주차공간별 주차 여부 갱신
	send_utilize	주차구역별 처리 시점을 IoT Gateway 1에 전달

□ 불법주차 차량 검출 모듈(M2)

파일명	함수명	설명
park_other_place.py	yolo_model_load	미리 학습된 YOLO 모델 빌드
	detection_park_other_place	불법차량의 수를 검출 후 Realtime Database에 전송
yolo_methods.py	dist	중앙 좌표의 거리 비교
	check_section	이전과 현재 이미지에서 추출된 각 차량의 위치와 유사도 비교 후 불법주차 차량 수를 반환

□ 번호판 인식 모듈(M3)

파일명	함수명	설명
car_number_	detection_carplate	전송받은 번호판 이미지의 OCR 수행 함수 호출

model.py	car_plate_detection	번호판 OCR 수행 후 서버 영역에 결과 전송
----------	---------------------	---------------------------

■ 서비스 제공 영역 : 어플리케이션

□ 주차공간 시각화 및 이용률

파일명	함수명	설명
ParkingLot	parkingInfo	잔여 주차공간을 통해 실시간 이용률 계산
Activity.java	carFindViewById	시각화 인터페이스에 잔여 주차공간 표시

□ 데이터베이스와 실시간 동기화

파일명	함수명	설명
ParkingLot Activity.java	InitFirebase Database	실시간 동기화를 위해 데이터베이스 값 변경에 따른 변동 값 수신
	onChildAdded	데이터베이스 항목 검색, 실시간 동기화를 위한 수신 대기, 주차 여부에 따른 이미지 적용
	onChildChanged	데이터베이스의 변화에 따른 실시간 동기화
	onChildRemoved	데이터베이스의 데이터 제거에 따른 삭제된 하위 항목의 데이터 포함

□ 주차장 길안내

파일명	함수명	설명
ParkingLot_ Activity.java	invokeRoute	사용자의 모바일 기기에 설치되어 있는 Tmap을 실행하여 길안내 수행

■ 사용자 제공 영역 : 웹 페이지

□ 웹 페이지 시각화

파일명	함수명	설명
Config.js	initializeAPP	Firebase 프로젝트 API key를 통해 웹 서버 실행
App.js	paging	주차정보 제공을 위한 테이블 자료 페이징 처리

□ 데이터베이스와 실시간 동기화

파일명	함수명	설명
App.js	getElementById	지정된 key의 하위항목을 읽고 최신 데이터 반환
	loadData	페이징 처리 후 주차 정보와 차량사진의 URL 을 테이블에 출력

□ 특별주차구역의 차량 사진 연동

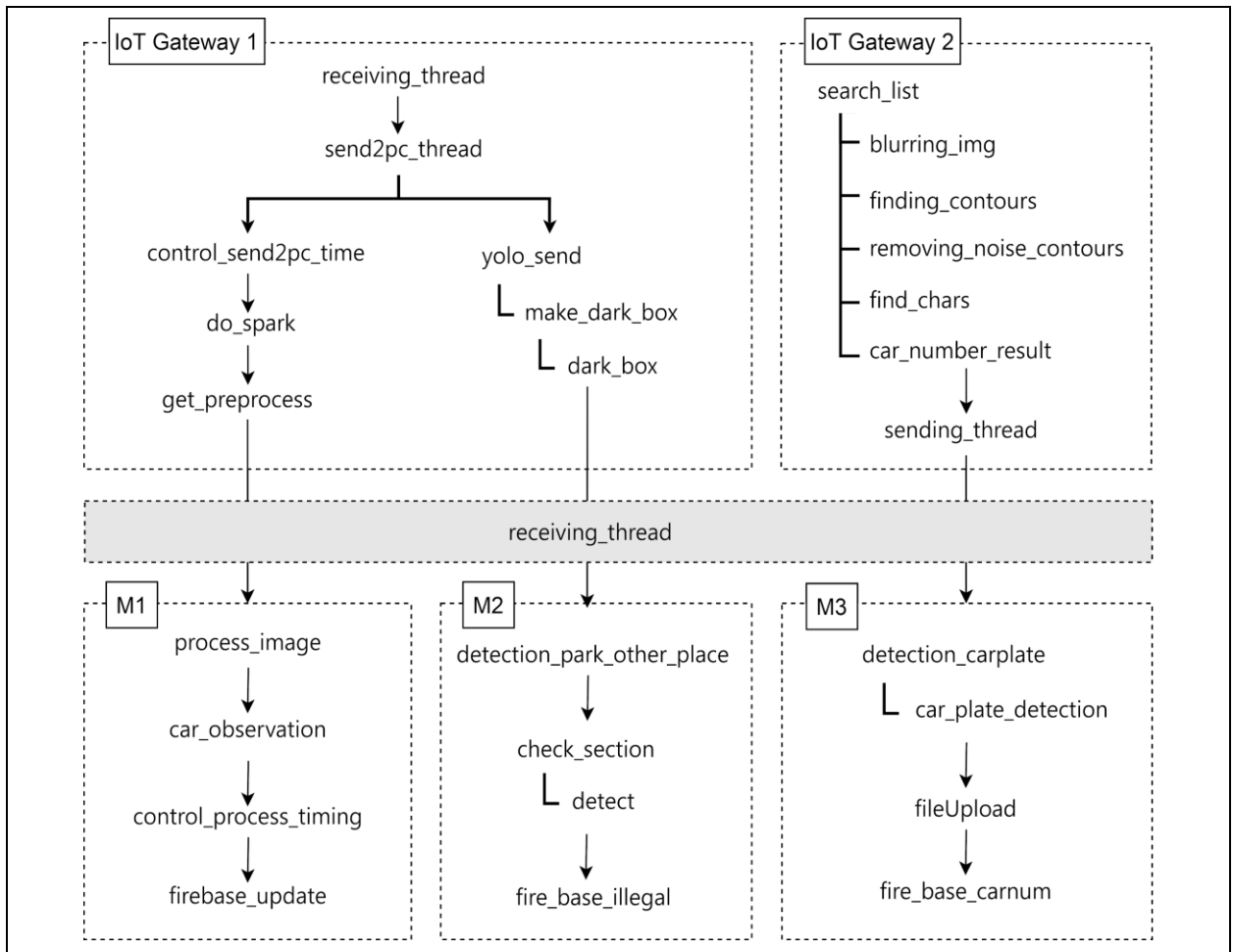
파일명	함수명	설명
App.js	getDownloadURL	차량번호와 매칭되는 사진의 URL을 호출

□ 관리자 메시징

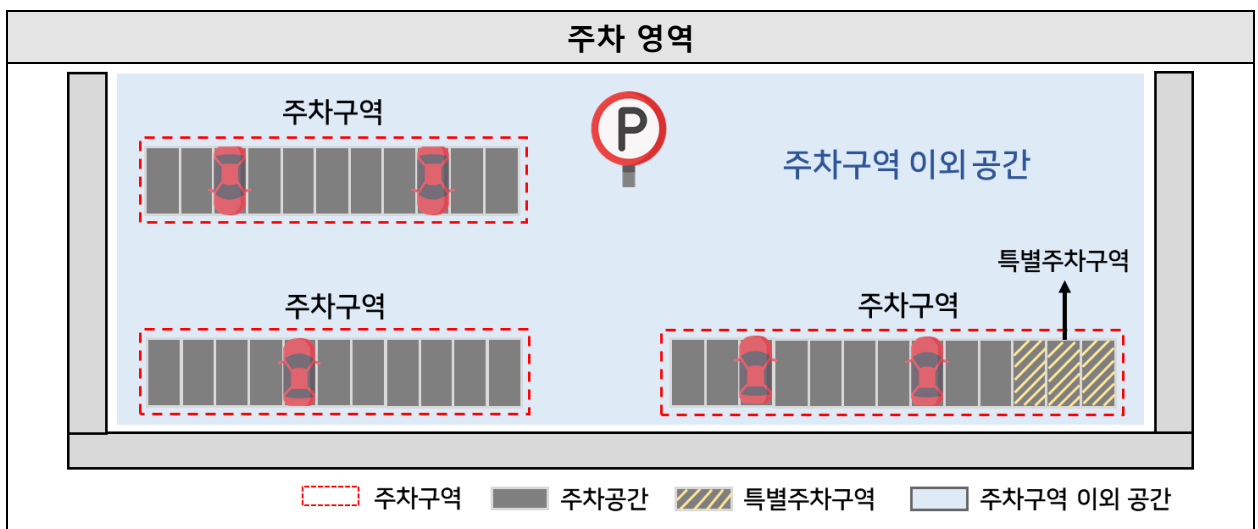
파일명	함수명	설명
-----	-----	----

firebase-messaging-sw.js	send	만차 시 관리자 페이지에 만차 상태를 전송
	messaging	관리자에게 만차알림 전송

○ 주요 함수의 흐름도



○ 각 주차 영역에 대한 정의



주차공간	주차 시설이 갖춰진 공간
주차구역	여러 개의 주차공간으로 구분해 놓은 지역
특별주차구역	특정 목적으로 지정된 주차 공간 / Ex) 임산부, 장애인 전용 주차 구역
주차구역 이외 공간	지정된 공간 이외의 구역으로서 주차 단속에 해당되는 지역

○ 기술적 차별성

본 시스템은 IoT CCTV를 통해 수신된 영상을 분석을 통해 차량들의 주차 여부를 판단한다. 이는 센서를 통해 주차 여부를 판단하는 기존 시스템에 비해 설치 및 유지·보수가 간편한 장점을 가진다. 뿐만 아니라 안드로이드 어플리케이션과 웹 페이지를 통해 실시간으로 사용자와 관리자에게 주차장의 정보를 제공함으로써 편의성을 높인다. 이외에도 기존 시스템에서 제공하지 않는 불법주차 정보를 관리자에게 제공한다.

타 주차 관리 시스템과의 차별성			
시스템	파킹 스캐너 (Parking Scanner)	지자기 센서 주차 감지 시스템	초음파 주차유도 시스템
IoT 기기	주차장의 IoT CCTV	주차공간 바닥의 지자기 센서	주차공간 상단부의 초음파 센서
기술 예시			
기술적 특징	타 시스템에 비해 설치·유지·보수 용이	설치가 비교적 번거로움	설치가 비교적 번거로움
	실내/외 주차장 관리 가능	비교적 높은 유지·보수 비용	비교적 높은 유지·보수 비용
	어플리케이션을 통한 잔여 주차공간의 개수 표기 및 시각화	실내/외 주차장 관리 가능	실내 주차장 관리 가능
	웹 페이지를 통한 불법주차 차량 정보 제공	잔여 주차공간의 개수 표기	잔여 주차공간의 개수 표기
	CCTV 기능과 더불어 주차장 관리 가능	추가적 기능 구현의 어려움	추가적 기능 구현의 어려움
	소프트웨어 기능 추가를 통한 시스템 업그레이드 가능		

## □ 부록

No	구분	설명
1	CNN (Convolutional Neural Network)	가중치 행렬인 필터를 사용해 입력된 데이터를 순환하며 분석한다. 따라서 데이터 내의 지역적 정보를 고려하는 것이 가능한 인공지능망이다. 이러한 특징으로 이미지 처리에 우수한 성능을 보인다.
2	YOLO (You Only Look Once)	이미지를 동일한 크기의 그리드로 구분 후 각 그리드마다 위치한 객체를 탐지해 다수의 객체를 탐지한다. CNN 기반의 모델링 기법으로 실시간 객체 탐지를 수행한다.
3	Bi-Directional LSTM	LSTM은 내부 은닉 상태를 유지하며 시간 순서에 따른 데이터를 처리하는 순환신경망 가운데 하나이다. Bi-Directional LSTM은 시간적 측면에서 순방향과 역방향을 고려하기 위한 순환신경망 구조이다.
4	OCR (Optical Character Recognition)	'광학 문자 인식'은 인쇄물 또는 사진 상의 글자와 이미지를 디지털 데이터로 변환해주는 자동인식기술이다.
5	Spark	오픈소스 분산 병렬 처리 프레임워크이다. In-memory 처리방식을 지원해 대용량 데이터를 고속 처리하며 성능 향상을 위한 다양한 인터페이스를 제공한다.
6	RDD (Resilient Distributed Datasets)	Spark에서 사용되는 핵심 데이터 구조로서 데이터를 여러 서버에 분산한 구조이다. RDD 생성 과정을 기록해 데이터 처리 과정 중 문제가 발생한 경우 생성된 RDD를 추적해 복구가 가능하다.
	Transformation	생성된 RDD를 변환해 반환하는 연산 방식이다. 연산 수행 과정만을 기록해 메모리 자원의 낭비를 방지한다.
	Action	생성된 RDD를 기반으로 필요한 연산을 수행하여 값을 반환한다
8	Firebase	모바일 및 웹 어플리케이션 개발 플랫폼이다. 실시간 데이터 동기화를 지원하며 클라우드 기반 Realtime Database를 지원한다.

## □ 참조

### \* Green Parking

출처 : [https://newsis.com/view/?id=NISX20190325\\_0000597432&clD=10201&pID=10200](https://newsis.com/view/?id=NISX20190325_0000597432&clD=10201&pID=10200)

### \* Firebase

출처 : <https://firebase.google.com/products>

## □ 개발 중 장애요인과 해결방안

### ○ 딥러닝 모델의 학습을 위한 학습데이터 수집

딥러닝 모델의 설계 및 학습에 있어서 학습 데이터의 양은 딥러닝 모델의 성능과 직접적인 관계가 있다. 본 시스템의 구성에 앞서 사전 학습에 사용할 학습 데이터가 부족했다. 이를 보완하기 위해 그림 6과 같이 차량 이동이 많은 시간에 카메라를 설치해 3개월 간 데이터를 수집했다. 수집된 학습 데이터의 기하학적 변환을 통해 데이터의 수를 증폭했다. 이 후 해당 데이터를 전처리 후 딥러닝 모델 학습에 사용했다. 구현 과정에서 학습된 딥러닝 모델의 성능을 확인해 시스템에서 사용 가능 여부를 확인했다.

### ○ 시스템에서 실시간성 정의

본 시스템은 주차장의 정보를 분석해 주차장 사용자와 관리자에게 실시간으로 분석된 정보를 제공한다. 본 시스템은 주차장의 정보를 분석하기 위해 딥러닝 모델을 사용하여 구현했다. 딥러닝 모델의 특성상 많은 연산량이 요구되어 실시간으로 데이터를 분석하는 것이 어렵다. 따라서 두가지의 알고리즘을 구현하여 실시간성을 보장했다.

1. 딥러닝 모델의 성능을 유지하며 간단한 구조의 모델 설계를 지향했다. 이를 위해 데이터 프레임을 주차공간별로 분리하여 이진분류를 수행하는 모델을 설계했다. 또한 CNN 기반 모델에서 'Pooling'을 통해 연산 파라미터의 수를 감소시켰다.
2. 스레드 방법을 통해 영상분석 영역의 모듈들이 독립적으로 수행되도록 구현했다. 이때, 멀티프로세싱 방법을 통해 프로세스를 병렬적으로 처리하도록 구현했다. 이를 통해 각 모듈의 처리 속도를 개선했다.

### ○ 제한된 하드웨어 자원의 효율적 운영의 필요성

본 시스템은 주차관리 시스템의 특성상 다수의 주차장을 통합적으로 관리해야 한다. 따라서 관리하는 주차장의 수가 증가함에 따라 처리해야 하는 데이터의 양이 증가하게 된다. 이는 실시간성 보장에 장애요소가 된다. 따라서 증가하는 데이터의 양에 따라 한정된 하드웨어 자원을 효율적으로 활용해 실시간성 보장을 위한 방법이 필요하다. 따라서 각 주차장의 IoT CCTV에서 촬영된 영상을 통합적으로 관리하는 IoT Gateway를 구현했다. 이를 통해 영상분석 영역에서 수행되는 연산을 IoT Gateway와 분담해 영상분석 영역의 연산량을 감소시켰다. 또한 하드웨어 자원의 효율적 사용을 위한 두가지 알고리즘을 구현했다.

1. 주차장의 특성상 같은 위치에 장시간 주차된 차량이 많다. 따라서 동일한 차량이 연속적으로 모델에 입력되는 경우가 많다. 이를 보완하기 위해 입력 프레임 간 유사도 비교를 수행했다. 유사도 임계 값을 넘지 못한 프레임을 모델에 입력 후 분석하여 불필요한 연산을 감소시켰다.
2. 주차장 이용률에 따른 '영상분석 시점 조절 알고리즘'을 구현했다. 주차장에 주차된 차량이 많을 수록 주차공간별로 주차 여부가 변동될 가능성이 많다. 주차장 이용률이 높은 경우 해당 주차장의 영상분석 시점을 짧게 설정해 높은 빈도로 주차 여부를 분석한다. 반대로 주차장 이용률이 낮은 경우 영상분석 시점을 길게 설정해 낮은 빈도로 주차 여부를 분석한다. 이를 통해 이용률에 따라 딥러닝 모델의 분석 빈도를 조절해 하드웨어 자원을 효율적으로 사용하도록 구현했다.

## □ 개발결과물의 차별성

### ○ 기존 주차 관리 시스템에 대한 보완

기존의 센서 기반 주차관리 시스템은 각 주차공간마다 별도의 센서를 필요로 하여 다수의 센서 설치가 필요하다. 따라서 설치 및 유지·보수가 번거롭고 센서 설치가 가능한 공간이 제한적이라는 단점이 존재한다.

본 시스템은 위와 같은 단점을 IoT CCTV를 활용한 딥러닝 영상처리 기반의 스마트 주차 관리 시스템을 통해 개선한다. IoT CCTV를 설치해 다수의 주차공간별 주차 여부를 동시에 판단한다. 이는 주차공간별로 센서 설치가 필요한 기존 시스템에 비해 설치가 간단하다. 이외에도 하나의 IoT CCTV로 여러 개의 센서의 역할을 대체함으로써 유지·보수의 요소를 감소시켰다는 장점이 존재한다.

본 시스템의 IoT CCTV는 카메라를 통해 감시 및 녹화하는 CCTV의 역할을 수행한다. 따라서 CCTV와 주차 관리 시스템의 기능을 통합적으로 수행할 수 있으므로 부가적인 CCTV를 설치할 필요가 없다. 따라서 기존에 CCTV와 주차 관리 시스템을 각각 설치해야 하는 기존 시스템에 비해 설치 비용 절감되는 긍정적 효과가 있다.

### ○ 기대효과

#### ✓ 비용 절감의 효과

기존 센서 기반의 주차관리 시스템은 주차공간별 센서 설치가 필요하다. 본 시스템은 IoT CCTV를 도입해 촬영되는 주차구역의 주차공간별 주차 여부 판단이 가능하다. 따라서 센서 설치 및 유지·보수 비용이 절감이 가능하다.

#### ✓ 주차장 근처 도로 및 골목 정체 감소

본 시스템은 안드로이드 어플리케이션을 통해 주차장의 현재 이용률 및 잔여 주차공간 정보를 제공한다. 이를 통해 사용자는 자신의 주변에 위치한 주차장 중 이용률이 낮은 주차장을 이용하는 것이 가능하다. 사용자는 주차장의 이용률을 미리 확인함으로써 주차장을 찾아 배회하는 시간을 절약할 수 있다. 따라서 주차장 사용자의 불편을 해소할 수 있으며 근처 도로 및 골목의 정체가 감소하게 됨으로써 주차 및 교통 문제가 개선이 가능하다.

#### ✓ 효율적 인력 관리

관리자는 웹 페이지를 통해 주차장의 현재 상태와 불법주차 차량에 대한 정보를 제공받는다. 따라서 관리자는 불법주차 차량 단속을 효율적으로 수행할 수 있다. 이를 통해 단속에 필요한 인력 및 비용 절감의 효과를 받을 수 있다.

#### ✓ 특별주차구역의 대상 차량에 대한 주차 권리 보장

특별주차구역에 주차한 차량의 번호판을 인식하여 해당 차량의 불법주차 여부를 판단한다. 이 결과를 관리자에게 제공한다. 이를 통해 관리자는 특별주차구역의 불법주차 차량을 단속하며, 특별주차구역에 주차가 가능한 주차장 사용자의 주차 권리를 보장한다.



## □ 팀 업무 분장

No	구분	성명	참여인원의 업무 분장
1	팀장	정영석	개발 총괄, 알고리즘 설계
			불법주차 차량 검출 모델 제작
			번호판 인식 알고리즘 개발
			프레임 전처리 알고리즘 개발
			데이터 수집
			보고서 작성
2	팀원	이연희	사용자 서비스 개발, 안드로이드 기획 및 총괄
			안드로이드 어플리케이션 개발
			시스템 아키텍처 제작
			전체 구조도 및 시각자료 제작
			동영상 프로젝트 설명 제작
			발표자료 제작
			보고서 작성
3	팀원	노재영	관리자 서비스 개발, 웹 페이지 기획 및 총괄
			웹 페이지 개발
			데이터베이스 및 서버 구축
			시연 영상 편집
			보고서 작성
4	팀원	손주형	IoT H/W 제작
			라즈베리파이 제작
			프레임 전처리 알고리즘 개발
			유사도 비교 알고리즘 개발
			보고서 작성
5	팀원	김도경	딥러닝 모델 제작, 알고리즘 설계
			주차 여부 판단 모델 제작
			프레임 전처리 알고리즘 개발
			데이터 수집
			시연 영상 편집
			보고서 작성