

# 실시간 대중교통 모니터링 시스템 구현

오은서<sup>1</sup>, 권소령<sup>1</sup>, 오정민<sup>1</sup>, 평보<sup>2</sup>, 김태국<sup>3\*</sup>

<sup>1</sup>국립부경대학교 컴퓨터·인공지능공학부 학생, <sup>2</sup>국립부경대학교 인공지능융합학과 석사과정,

<sup>3</sup>국립부경대학교 컴퓨터·인공지능공학부 교수

## Implementation of a real-time public transportation monitoring system

Eun-seo Oh<sup>1</sup>, So-ryeong Gwon<sup>1</sup>, Jung-min Oh<sup>1</sup>, Bo Peng<sup>2</sup>, Tae-kook Kim<sup>3\*</sup>

<sup>1</sup>Student, Computer and Artificial Intelligence Engineering, Pukyong National University

<sup>2</sup>Master's Student, School of Artificial Intelligence Convergence, Pukyong National University

<sup>3</sup>Professor, School of Computer and Artificial Intelligence Engineering, Pukyong National University

**요약** 본 논문에서는 실시간 대중교통 모니터링 시스템을 제안하였다. 제안한 연구는 대중교통 앱(App)을 제작하고, 광센서, 압력 센서, 객체검출 알고리즘을 활용하여 구현하였다. 또한, 버스 모형을 제작하여 동작을 검증하였다. 제안한 실시간 대중교통 모니터링 시스템은 다음과 같이 3가지 특징을 가진다. 첫째, 광센서와 압력 센서의 값의 변화에 따라 좌석의 착석 여부와 총 승객 인원을 파악하여 앱에서 대중교통 내부의 혼잡도를 확인할 수 있도록 구현하였다. 둘째, 다수의 승객이 동시에 승하차할 때 발생할 수 있는 광센서의 오차를 방지하기 위해, 객체검출 알고리즘인 YOLO를 활용하여 CCTV 승객 수 확인 가능성을 확인하였다. 셋째, 별도의 화면에서 탑승할 버스 내의 좌석이 착석된 경우를 색깔로 표시함으로써 편의를 제공한다. 승객의 현재 위치 확인, 현 위치에서 탑승 가능한 대중교통 및 도착 잔여 시간도 확인 가능하다. 따라서 제안한 시스템은 대중교통 이용객들에게 보다 높은 편의성을 제공할 수 있을 것으로 기대한다.

**주제어** : 아두이노, 대중교통, 실시간 모니터링 시스템, 객체 검출, YOLO(You Only Look Once)

**Abstract** In this paper, a real-time public transportation monitoring system is proposed. The proposed system was implemented by developing a public transportation app and utilizing optical sensors, pressure sensors, and an object detection algorithm. Additionally, a bus model was created to verify the system's functionality. The proposed real-time public transportation monitoring system has three key features. First, the app can monitor congestion levels within public transportation by detecting seat occupancy and the total number of passengers based on changes in optical and pressure sensor readings. Second, to prevent errors in the optical sensor that can occur when multiple passengers board or disembark simultaneously, we explored the possibility of using the YOLO object detection algorithm to verify the number of passengers through CCTV footage. Third, convenience is enhanced by displaying occupied seats in different colors on a separate screen. The system also allows users to check their current location, available public transportation options, and remaining time until arrival. Therefore, the proposed system is expected to offer greater convenience to public transportation users.

**Key Words** : Arduino, Public Transport, Real-time Monitoring System, Object detection, YOLO (You Only Look Once)

## 1. 서론

평소 대중교통을 이용하는 승객들은 애플리케이션을 통해 목적지까지 가는 방법을 검색한다. 이때, 목적지를 향해 가는 대중교통의 노선이 여러 가지인 경우를 본 경험이 있을 것이다. 특히 출퇴근 시간대에 승객이 가득 차 있는, 혼잡한 버스를 이용하는 것과 승객이 얼마 없어 여유로운 버스에 앉아 가는 것은 탑승객이 느끼는 피로도의 차이가 크다는 것을 체감할 수 있다. 이와 같은 불편함을 겪지 않기 위해 출근 시에는 조금 더 이른 시간에, 퇴근 시에는 조금 더 늦은 시간에 대중교통을 이용하는 승객들이 다수 있다. 수도권 이외의 지역에서는 여전히 애플리케이션에 노선을 검색하면, 해당 목적지까지 걸리는 시간과 대중교통의 도착 잔여 시간정도만 알려주는 경우가 많다[1].

본 연구에서는 이를 개선하기 위해 해당 대중교통 내부의 혼잡한 정도와 좌석의 착석 여부를 알려주는 기능을 추가하여 보다 편리하게 대중교통을 이용할 수 있는 환경을 만드는 것을 목적으로 한다. 직접 버스 모형을 제작하여 아두이노(Arduino)를[2] 설치하고, 승하차를 위한 출입구에는 적외선 광센서를, 좌석에는 FSR(force sensitive resistor) 센서를 부착한다. 그리고 많은 승객이 동시에 승하차할 때 발생할 수 있는 센서 측정 오차 문제를 보완하기 위해, 객체 검출 알고리즘 YOLO(You Only Look Once)를 활용하였다. 이를 통해 CCTV로 승객 수의 확인 가능성을 확인하였다.

그리고 MIT 앱 인벤터(App Inventor)를[3] 이용해 만든 대중교통 앱과 아두이노를 블루투스 통신을 통해 연결하여 실제 대중교통을 이용하는 것과 유사한 상황을 구현하였다.

## 2. 관련 연구 동향

대중교통을 이용하는 승객의 수가 많아지고 기존의 대중교통 시스템을 개선하려는 많은 연구가 이루어지고 있다. 특히, 버스 차내 혼잡도 파악을 통한 정보 제공과 좌석 정보를 개선하기 위한 많은 연구가 이루어지고 있다.

이백진 등의 연구에서는 선호의식 조사를 통한 버스 차내 혼잡도 정보제공이 버스선택에 미치는 영향을 분석하였다. 운행정보 등의 교통정보 위주의 기존 시스템과 달리 버스 차내 혼잡도 정보 제공이 이용자들의 버스선택 행태에 미치는 영향을 분석하였다[4].

정양록 등의 연구에서는 수집된 교통카드 데이터의 전처리 과정을 통해 버스의 승하차 인원을 추정하고 버스 내 재차인원 및 혼잡수준을 추정하는 것을 목적으로 한다. 부산광역시에서 운행되는 버스 데이터를 수집 및 분석하며 검증을 위해 현장 조사를 실시하여 연구의 정확도를 높였다[5].

진유진 등의 연구에서는 버스 내 CCTV 영상데이터를 분석한 결과를 인공지능과 접목시켜 혼잡도 산출 모델을 개발하였다. 이를 통해 이용자의 편의성과 안전성을 기대하며, 대중교통 이용의 증가로 인한 온실가스 배출량 감축의 효과도 기대한다[6].

임정환 등의 연구에서는 시계열 예측을 사용한 버스 잔여좌석 예측 시스템을 연구하였다. 제안된 연구에서는 ARIMA(Autoregressive Integrated Moving Average) 기법을 이용하여 시계열 분석을 하였다[7].

한편, 딥러닝의 발전은 센서 기반 응용 시스템에 새로운 개선 및 보완 방법을 제공하였다. 그중 객체 검출 알고리즘은 입력 이미지에서 객체 특징을 학습하여 객체를 실시간으로 인식하고 표시할 수 있으며 이는 여러 분야에 널리 응용되고 있다.

최우석 등은 YOLO-NAS 모델을 활용하여 산업 현장 내 작업자의 안전모 착용 여부를 확인하는 연구를 진행하였다. 해당 연구에서 임베디드 시스템과 객체 검출 기술의 결합 가능성과 필요성을 입증하였다[8].

박민호 등의 연구에서는 선박을 감지하기 위해 YOLOv3 및 YOLOv5를 동시에 사용하여 두 모델의 성능을 비교 분석하였다. 결과적으로 두 모델 모두 우수한 감지 성능을 보였지만, 모델 크기 및 학습 속도 등 평가 지표를 고려했을 때 YOLOv5가 더 좋은 종합 성능을 보인다고 밝혔다[9].

이도현 등의 연구에서는 횡단보도에서 휠체어 및 목발을 사용하는 교통 약자들을 효율적으로 탐지하는 시스템을 개발하였다. 해당 연구에서 YOLO 모델을 앙상블 기법과 결합하여 더 높은 검출 성능을 달성하였다[10].

김덕원 등의 연구에서는 YOLO와 OpenCV를 활용해 불법 현수막을 자동으로 검출하고 분류하는 시스템을 제안하였다. CCTV 영상에서 현수막 영역을 YOLOv5로 검출하고, OpenCV로 색상을 분석하여 규제 준수 여부를 판단한다. 이를 통해 인력과 비용을 절감하고 효율적인 관리가 가능하게 만들었다[11].

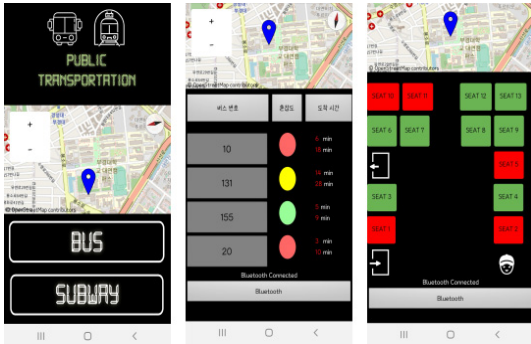
이승원 등은 딸기 수확 로봇 시스템을 제안하였다. 해당 연구에서 YOLO를 활용하여 딸기를 검출하고, 개발한 딸기 성숙도를 분류하는 알고리즘으로 딸기 수확을

위한 자르는 지점을 계산하는 기능을 구현하였다. 제안한 팔기 수확 로봇 시스템은 다양한 재배 상황에서 성능 평가를 수행하였으며, 익은 과일에 대한 수확 성공률이 90%에 도달하였다[12].

### 3. 실시간 대중교통 모니터링 시스템 구현

본 논문에서는 실시간 대중교통 모니터링 시스템의 구현을 연구하였다. 제안한 시스템은 대중교통의 도착 잔여시간 확인, 차내 혼잡도 확인, 좌석의 착석 여부 등을 확인할 수 있도록 구성된다.

#### 3.1 대중교통 시스템 앱 화면 구성

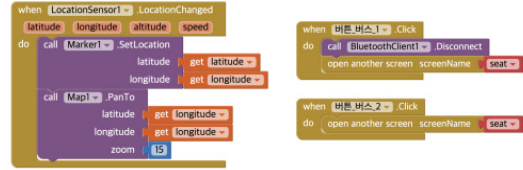


[Fig. 1] Formation of public transportation system

[Fig. 1]은 본 연구의 최종 결과물인 대중교통 시스템 애플리케이션 화면 구성을 나타내며, 화면은 크게 세 가지로 구성된다. 최초 화면에서는 버스와 지하철 중 사용자가 원하는 대중교통의 종류를 선택할 수 있도록 구현한다. 화면에 나타나는 지도는 사용자의 현재 위치를 마커로 표시하여 주변 지리를 한눈에 볼 수 있도록 한다. 탑승할 대중교통의 종류를 선택한 후 두 번째 화면에서는 사용자의 현재 위치와 가장 가까운 버스 정류장에 도착할 버스의 번호와 해당 버스의 차내 혼잡도 정도를 색깔로써 확인할 수 있다. 이때, 버스 차내 혼잡도는 버스 좌석이 남아있는 경우 초록색, 빈 좌석이 없고 입석 승객의 수가 적은 경우 노란색, 입석 승객이 많아 혼잡한 경우는 빨간색으로 나타낸다. 각각 버스 번호마다 두 대의 버스 도착 잔여 시간을 함께 확인할 수 있도록 한다. 탑승할 버스를 지정한 후 해당 버스 번호를 선택하면 마지막 화면으로 이동하고, 차내 좌석의 배치와 좌석마다의 착석 여부를 시각적으로 파악할 수 있다.

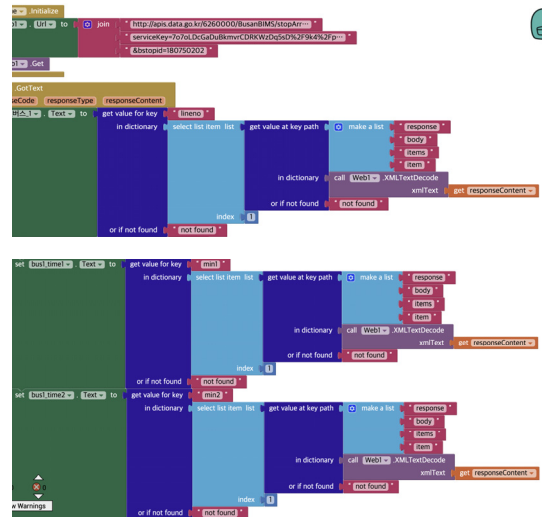
#### 3.2 대중교통 시스템 동작 구현

본 연구에서는 실시간 대중교통 모니터링 시스템 제작을 위해 MIT 앱 인벤터(App Inventor)를 사용한다.



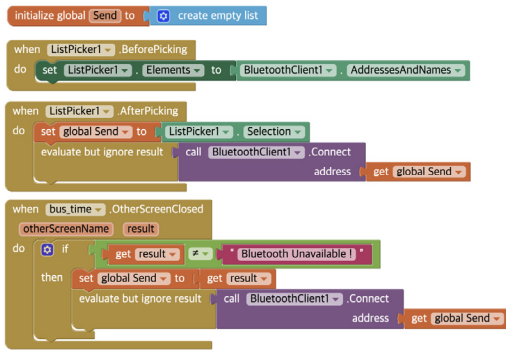
[Fig. 2] Composition of GPS and screen transition

[Fig. 2]는 대중교통 시스템 앱 화면에서 나타나는 GPS 기반의 지도와 버튼을 통한 화면 전환을 구현한 코드이다. 스마트폰에 내장된 GPS 기능을 통해 사용자의 현재 위치를 위도와 경도로 파악하여 지도를 조정하고 마커로써 표현한다. 버스와 지하철을 선택하는 버튼, 각각의 버스 번호 버튼을 눌러 다른 화면으로 이동할 수 있으며, 화면 하단의 뒤로가기 버튼을 통해 이전 화면으로 이동할 수 있도록 한다.



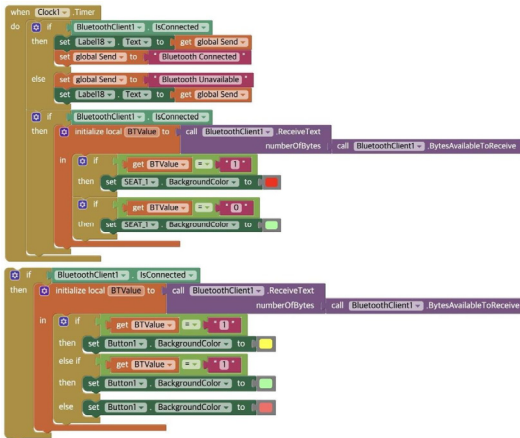
[Fig. 3] Composition of OpenAPI

[Fig. 3]은 버스 도착 잔여시간을 앱에 나타내기 위한 부분으로, 공공 API를 사용한다. 공공데이터 포털에서 [13] 대중교통정보시스템의 OpenAPI를 활용 신청하여 승인을 받은 후 사용하였으며, 인증키와 버스 정류장 ID, 도착 버스의 번호에 따른 잔여 시간 API를 활용한다.



[Fig. 4] Composition of Bluetooth connection

[Fig. 4]은 본 연구에서 아두이노와 앱을 블루투스로 연동하기 위해 작성한 코드 화면이다. 앱 화면 중 블루투스 연결을 위한 버튼은 아두이노와 연동을 확인하기 위한 용도이며, 목록에서 아두이노를 선택한 후 연결할 수 있도록 한다. 이를 위해 아두이노에서는 블루투스 센서 HC-06 모듈을 사용한다.



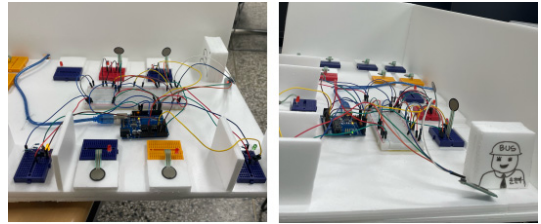
[Fig. 5] Composition of connection with Arduino's sensor

[Fig. 5]는 아두이노에 설치된 센서와 연동되는 부분을 작성한 코드이다. 본 연구에서는 차내 총 승객의 수를 파악하기 위해 승하차를 위한 앞문, 뒷문 위치에 각각 적외선 광센서를 설치한다. 앞문의 광센서가 임계값 이상이 되면 총 승객의 수가 1 증가하고, 뒷문의 광센서가 임계값 이상이 되면 총 승객의 수가 1 감소한다. 각 좌석이 착석된 여부에 따라 승객이 착석한 상태의 좌석은 화면 상에서 빨간색으로, 비어있는 좌석은 초록색으로 나타내도록 한다.

## 4. 버스 모형 제작 및 아두이노 구현

본 연구에서는 실제 대중교통과 유사한 환경에서 아두이노 센서의 동작을 확인하기 위해 버스 모형을 제작하여 검증하였다.

### 4.1 버스 모형 및 아두이노 제작



[Fig. 6] Model of Bus in Public Transportation

[Fig. 6]은 우드락을 사용하여 직접 제작한 버스 모형의 모습이다. 실제 버스와 유사한 환경을 만들기 위해 미니 브레드보드를 좌석마다 설치하고, 노약자석은 노란색, 일반 좌석은 파란색, 임산부석은 빨간색 보드를 사용한다. 앞문, 뒷문을 나타내기 위해 작은 가벽을 설치하고, 두 개의 문에는 적외선 광센서를 설치하여 승하차 승객을 감지한다. 버스의 모든 좌석에는 FSR 압력 센서를 부착하여 착석 승객의 수와 좌석의 착석 여부를 파악한다. 이때, 사람을 대신하여 여러 가지 무게의 무게추를 이용해 연구를 진행한다.

### 4.2 아두이노 기능 구현

<Table 1>은 아두이노에서 블루투스 연결이 가능하도록 하며 적외선 광센서로부터 입력값을 받아 승객의 수를 카운트하는 코드이다. 'sensorValue1'과 'sensorValue2'에는 각각 승차 출입문, 하차 출입문에 부착된 적외선 광센서 2개와 연결되어 있어, 이 값을 읽어온다. 적외선 광센서의 결과값은 아날로그 값이므로 'analogRead()' 함수를 통해 정수값으로 변환하여 두 변수에 저장한다. 이후 'cnt' 변수를 통해 탑승객의 수를 저장한다.

본 연구에서 진행한 테스트에서는 적외선 광센서가 탑승객을 인지하는 기준을 800으로 정하여 센서의 값이 800 초과이면 승차, 800 미만이면 하차로 판단하여 각각 승객의 수를 카운팅한다.

〈Table 1〉 Code about counting people from infrared optical sensor

```
void loop(){
  while(BTSerial.available()){
    BTValue = BTSerial.read();
    Serial.println(BTValue);
  }

  int fsrValue1 = analogRead(fsrPin1);
  int fsrValue2 = analogRead(fsrPin2);

  sensorValue1=analogRead(sensorPin1);
  sensorValue2=analogRead(sensorPin2);

  if(sensorValue1>800)
    cnt++;
  //----- disembark passenger cnt -----
  if(sensorValue2>800)
    cnt--;
```

〈Table 2〉 code for LED display according to number of passengers

```
if(cnt >= 0 && cnt <= 10) {
  digitalWrite(ledPin2, LOW);
  digitalWrite(ledPin1, HIGH);
}
else if(cnt >10 && cnt <= 18) {
  digitalWrite(ledPin1, LOW);
  digitalWrite(ledPin2, HIGH);
}
else if(cnt >18) {
  digitalWrite(ledPin1, LOW);
  digitalWrite(ledPin2, LOW);
}
```

〈Table 2〉는 승객의 수에 따라 LED 표시를 달리 작동시키는 코드이다. 본 연구에서는 13인석 버스를 기준으로 혼잡도의 기준을 여유, 보통, 혼잡의 3가지로 분류하였고, 이는 각각 초록색, 노란색, 빨간색 LED와 연결되어 있다. 연구를 진행하는 동안 시각적으로 센서의 작동 결과를 확인하기 위해 앱의 실행과 별도로 버스 모형 내부에 센서를 설치하였다. 소스 코드에서와 같이, 승객이 0-10명 사이는 여유, 10-18명 사이는 보통, 18명 이상은 혼잡으로 설정하였다.

〈Table 3〉 Code about FSR sensor

```
if(fsrValue2 > 700) {
  digitalWrite(fsrled2, HIGH);
}
else {
  digitalWrite(fsrled2, LOW);
}
```

〈Table 3〉은 FSR 압력 센서의 구동에 대한 코드이다. 버스 모형 내부에 13개의 좌석마다 각각 압력 센서를

설치하였고, 이 중 2개의 FSR 센서를 연결하여 위치를 번갈아가며 연구를 진행하였다.

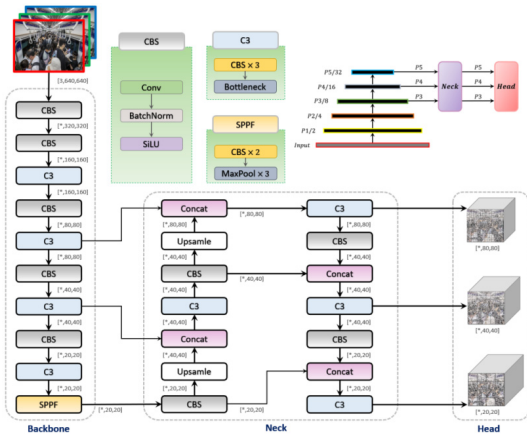
〈Table 1〉에서 'fsrValue1', 'fsrValue2'의 변수는 각각의 압력 센서의 아날로그 값을 읽어온 결과이며, 적외선 광센서와 마찬가지로 정수 값으로 변환하여 나타낸다. 압력 센서는 값이 700 이상일 때 사람이 착석했다고 기준을 정하고, 해당 좌석과 연결된 LED를 ON한다. LED가 켜진 좌석은 대중교통 시스템의 앱 상에서 해당 좌석이 착석되었다고 확인할 수 있도록 한다.

### 4.3 객체 검출 알고리즘 YOLO 활용

기존의 적외선 광센서를 사용하여 승객의 승하차를 감지하고, 이를 통해 총 승객수를 계산하는 방식은 버스 내의 혼잡함이나 다수의 승객이 동시에 승하차하는 상황에서 오차가 발생할 수 있다. 이러한 문제를 방지하기 위해 본 연구에서는 YOLOv5(You Only Look Once version 5)[14] 객체 검출 알고리즘을 활용하여 CCTV 영상을 통해 실시간으로 승객수를 파악하는 방안의 가능성을 확인하였다. YOLOv5를 활용한 승객수 파악은 PC에서 검증하였고, 라즈베리 파이(Raspberry Pi) 등의 임베디드 시스템으로 구현 가능하다[15-17].

YOLO는 단일 단계 검출(One-stage detection) 알고리즘으로 RCNN[18], Fast-RCNN[19], Faster-RCNN[20]과 같은 두 단계 검출(Two-stage detection) 알고리즘에 비해 우수한 검출 정확도를 유지하면서도 매우 빠른 검출 속도를 갖추고 있다. YOLOv5는 YOLO 시리즈의 다섯 번째 버전으로 이전 버전들의 개선과 최적화를 통해 더 높은 검출 정확도와 속도를 가지고 있으며, 자율주행, 얼굴 인식 및 침입 탐지 등 실시간 응답이 필요한 감지 작업에 적합하다. YOLOv5의 네트워크 구조는 Backbone, Neck 및 Head의 세 부분으로 구성된다. [Fig. 7]은 YOLOv5의 네트워크 구조 및 각 모듈의 상세 구성을 나타낸다.

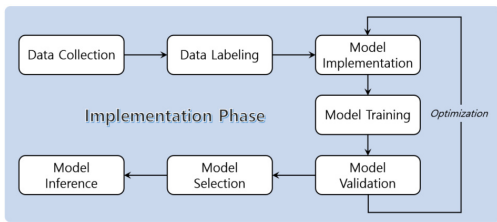
Backbone의 주요 구성 요소는 CBS, C3 모듈 및 SPPF 모듈이다. CBS 모듈은 기본적인 합성곱 연산 모듈로 하나의 합성곱 연산층과 Batch Normalization 층 및 SiLU 활성화 함수를 포함한다. C3 모듈은 YOLOv5의 핵심 특징 추출 모듈로, BottleneckCSP 구조를 사용하여 단계 간 부분 연결(Cross Stage Partial connections)을 통해 네트워크의 표현 능력과 특징의 재사용성을 높인다. 마지막인 SPPF(Spatial Pyramid Pooling-Fast)는 SPP 모듈을 개선한 것으로 다중 스케일(Multi-scale) 최대 풀링 연산을 사용하여 특징을 추출함으로써 특징 추출 능력을 강화하고 다양한 크기의 입력 이미지를 처리할 수 있다.



[Fig. 7] The Network Architecture of YOLOv5

Neck는 주로 특징의 융합과 전달 역할을 하며, FPN (Feature Pyramid Network)과 PANet(Path Aggregation Network)의 장점을 결합한다. Neck 부분은 Backbone 부분에서 추출된 특징 맵을 상향 경로(Bottom-up Pathway)를 통해 고차원(깊은) 특징을 저차원 특징으로 전달하여 저차원 특징의 표현 능력을 강화한다. 그런 다음 하향 경로(Top-down Pathway)를 통해 저차원 특징을 고차원 특징으로 전달하여 서로 다른 수준의 특징 간 정보 융합을 더욱 강화함으로써 검출의 정확도와 견고성(Robustness)을 높인다.

마지막, Head는 검출과 분류를 수행한다. Backbone과 Neck 부분에서 추출 및 융합된 특징을 추가로 처리하여 최종 예측 결과를 생성한다. [Fig. 8]은 YOLOv5를 활용한 승객수 계산 기능을 구현하는 단계를 나타낸다.



[Fig. 8] Implementation Process of Passenger Counting Using YOLOv5

#### 4.3.1 데이터 수집

모델의 일반화 능력과 적용성을 보장하기 위해 대중교통 내부의 혼잡 상황을 포함한 이미지를 수집하였다. 그 중에 다양한 승객 밀도의 버스과 지하철 내부, 사람 많이 모이는 쇼핑몰, 학교, 식당 등 환경이 포함된다. 전체 이

미지는 구글, Pixabay, Pexels 이미지 라이브러리를 이용하여 수집하였고, [Fig. 9]은 수집된 이미지들의 일부를 보여준다. 이러한 데이터는 총 400장으로 모델의 학습 데이터로 사용될 것이다.



[Fig. 9] A Portion of the Collected Data

#### 4.3.2 레이블링 및 분류

수집된 데이터를 학습할 수 있도록 LabelImg를 이용하여 수동으로 레이블링 작업을 수행하였다. [Fig. 10]은 LabelImg에서 예시 그림을 레이블링하는 과정을 나타낸다. 좁은 차내 공간에서 승객 밀도가 높은 경우 심각한 가림 현상이 발생할 수 있다. 이러한 문제를 해결하기 위해 데이터 레이블링 단계에서는 주로 승객의 머리 부분에 초점을 맞추어 레이블링을 수행하였다. 이는 승객이 서로 겹쳐있는 상황에서도 머리 부분은 비교적 쉽게 식별할 수 있기 때문이다. 이로써 모델이 승객의 머리 부분을 주요 특징으로 인식하도록 하여, 높은 밀집도 상황에서도 더 정확하게 승객 수를 추정할 수 있도록 한다. 레이블링 된 데이터를 학습용 70%, 검증용 15% 및 테스트용 15%로 분류하였다.



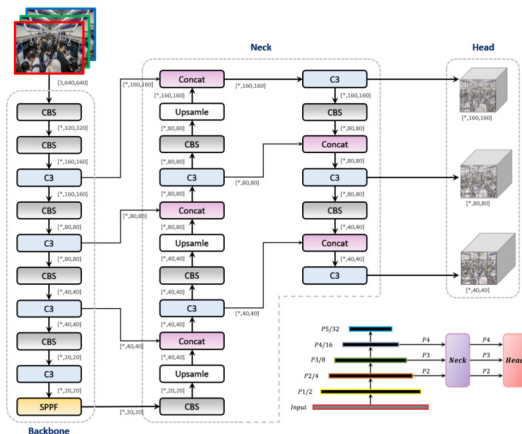
[Fig. 10] Labeling Work Using LabelImg

#### 4.3.3 모델 구현 및 학습

데이터셋 분석을 통해 검출 대상(머리)의 크기가 현재 승객이 위치한 장소에 따라 달라진다는 사실을 알 수 있

다. 승객이 차내 CCTV에서 멀리 떨어져 있을 경우, 포착되는 부분(특징)이 매우 작아진다. 그리고 차내 승객이 많고 밀집되어 있는 경우 검출 대상 간의 가림 현상이 발생할 수 있다.

[Fig. 11]과 같이 YOLOv5의 원래 구조에서는 Head 부분에 작은 목표(P3/small), 중간 목표(P4/medium), 큰 목표(P5/large)를 검출하는 검출 헤드가 포함되어 있으며, 이들은 각각 원본 입력 크기의 8배, 16배, 32배로 다운샘플링된 크기를 가진다. 그러나 더 작은 목표(멀리 있는 승객의 머리)의 경우 특징이 적고 불명확하여 일련의 합성곱 연산으로 인해 특징이 소실되기 쉽다. 또한, YOLOv5는 추론 과정에서 NMS(Non-Maximum Suppression) 알고리즘을 사용하여 가장 높은 신뢰도의 검출 상자 B를 유지하고, B와 IoU(Intersection over Union) 값이 지정된 임계값을 초과하는 검출 상자의 신뢰도 점수를 0으로 설정한다. 일반적으로 이러한 방법은 모델이 다수의 검출 상자 중에서 중복된 검출 상자를 제거하는 데 효과적이지만, 밀집되고 객체 간 가림이 발생하는 상황에서는 NMS 알고리즘이 여러 목표의 검출 상자를 동일한 객체의 검출 상자라 잘못 인식할 수 있다.



[Fig. 11] Improved YOLOv5 Network Architecture

이러한 문제를 해결하기 위해 본 연구에서 더 작은 객체를 검출할 수 있도록 새로운 검출 헤드 P2를 추가하였다. 이 헤드는 원본 입력 크기의 4배로 다운샘플링된 크기를 가지며, 기본 모델에서 큰 객체를 검출하는 P5 헤드는 제거하였다. [Fig. 11]은 개선된 모델의 네트워크 구조를 나타낸다. 이러한 개선을 통해 모델은 작은 목표를 검출하는 능력을 향상시키는 동시에 필요한 파라미터 양을 줄일 수 있다. 추가로 추론 과정에서 기존의 NMS

방법 대신 Soft-NMS를 사용하였다. NMS와 달리 Soft-NMS는 B와 IoU가 임계값을 초과하는 검출 상자의 신뢰도 점수를 0으로 설정하는 대신, 이를 감소시킨다. 이러한 메커니즘은 밀집된 객체와 가림 현상이 발생하는 환경에서 더 나은 처리 효과를 제공하여 누락 검출 문제를 줄일 수 있다.

특징을 충분히 학습시키기 위해 Batch size는 4로 설정하여 200 epoch의 학습을 진행하였다. 실험환경은 <Table 4>에 나타난 것과 같다.

<Table 4> Experiment environment

SW	OS	Windows 11
	Python	3.9.13
	CUDA	11.5.2
	cuDNN	8.9.6
	Pytorch	1.12.1
HW	CPU	Intel Core i9-12900K
	GPU	NVIDIA GeForce RTX 4060 Ti

#### 4.3.4 모델 검증

검증 단계에서 YOLOv5의 nano(n), small(s), medium (m), large(l), extra large(x)이 5가지 모델 크기를 각각 학습시키고, <Table 5>에 나타난 바와 같이 이들의 종합 성능을 비교하였다. 이 중에 YOLOv5-person는 YOLOv5m을 기반으로 개선한 모델이다. 이 모델은 기존의 P5 검출 헤드를 제거하고, 작은 객체를 검출할 수 있는 P2 검출 헤드를 추가했으며 중첩된 객체의 누락 검출 문제 해결하기 위해 Soft-NMS를 적용하였다. 이러한 개선 방법이 수집된 데이터셋에 매우 잘 적용될 수 있음을 보여주었다.

<Table 5> Performance comparison between models

모델	Precision	Recall	mAP <sub>50</sub>	mAP <sub>50-90</sub>
YOLOv5-n	0.781	0.624	0.699	0.332
YOLOv5-s	0.796	0.650	0.705	0.335
YOLOv5-m	<b>0.809</b>	0.634	0.711	0.331
YOLOv5-l	0.785	0.642	0.698	0.327
YOLOv5-x	0.783	0.602	0.663	0.306
YOLOv5-person	0.774	<b>0.665</b>	<b>0.750</b>	<b>0.384</b>

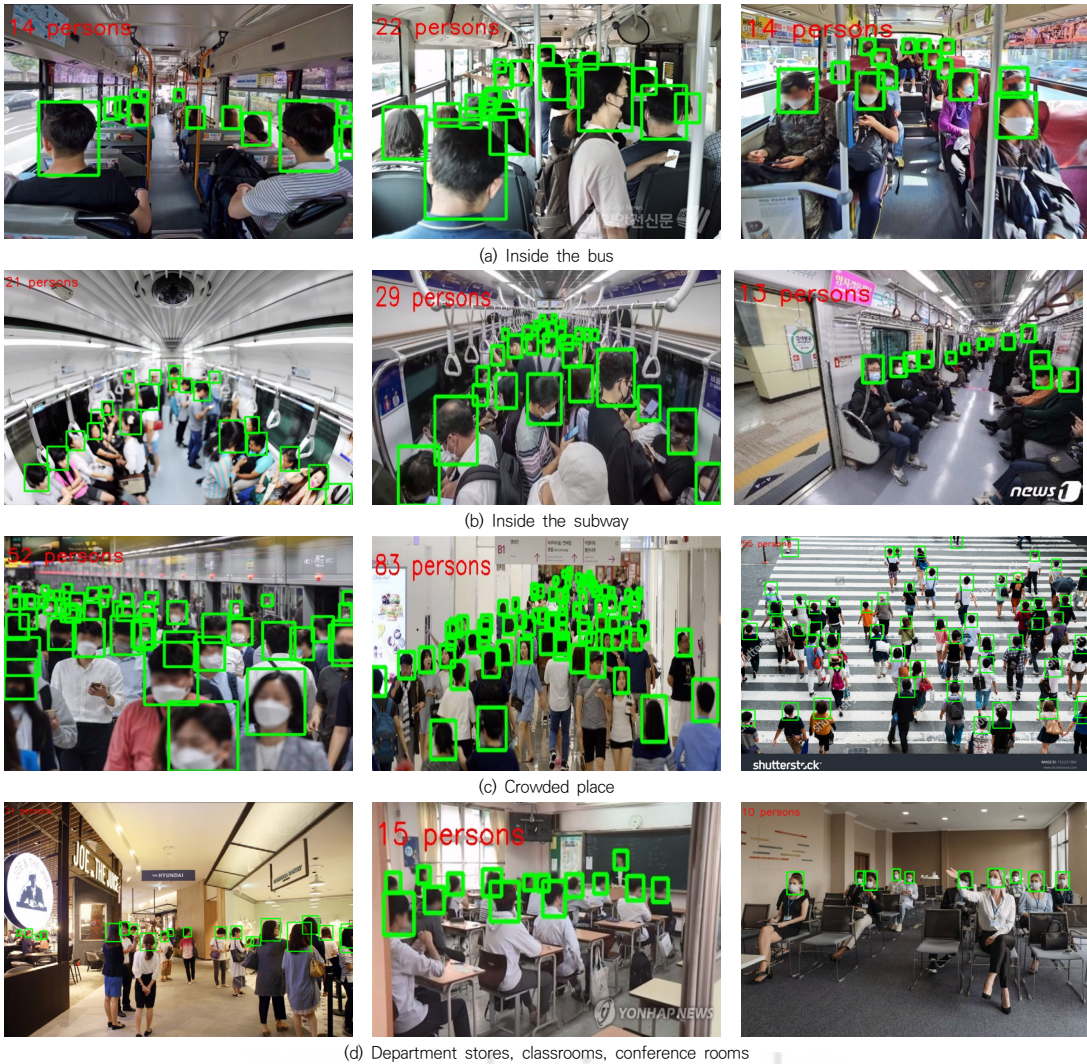
기존 모델 YOLOv5와 비교했을 때, 개선 모델 YOLOv5

-person는 mAP<sub>50</sub>(Mean Average Precision at 50%) 및 mAP<sub>50-90</sub>(mean Average Precision at IoU thresholds from 0.50 to 0.95)은 각각 3.9%와 5.3%를 향상하였다. 그리고 Recall는 3.1% 증가하여 작고 증척된 객체에 대한 누락 검출 문제를 완화할 수 있음을 확인할 수 있다.

〈Table 6〉는 각 학습된 모델의 층 수, 파라미터 수 및 모델 메모리 크기를 나타낸다. YOLOv5m에 비해 YOLOv5-person의 전체 파라미터 수와 메모리 크기는 각각 약 24%와 23%가 감소하였다. 이러한 감소는 모델의 검출 능력을 향상시키는 동시에 모델의 크기를 줄여 경량화하고 실제 응용에 있어 더욱 편리하게 만든다.

〈Table 6〉 Size comparison between models

모델	Layers	Parameters (Million)	memory (MB)
YOLOv5-n	157	1.765	3.9
YOLOv5-s	157	7.022	14.4
YOLOv5-m	212	20.871	42.2
YOLOv5-l	267	46.138	92.8
YOLOv5-x	322	86.218	173.1
YOLOv5-person	229	15.809	32.6



[Fig. 12] Detection results of YOLOv5-person

〈Table 5〉 중 모든 모델의 검출 정확도가 상대적으로 낮게 나타난 것은 주로 데이터셋의 부족 때문이다. 수집된 데이터셋은 총 400장의 이미지로, 이는 모델이 특징을 충분히 학습시키기에 부족한 양이다. 이러한 제한된 데이터셋의 이유는 주로 버스과 지하철 내부에서 CCTV를 통해 이미지를 수집하는 것이 어렵기 때문이다. 대중교통 수단 내부의 촬영은 프라이버시 문제, 접근성 문제 등의 제약이 따른다.

#### 4.3.5 모델 이용

[Fig. 12]는 YOLOv5-person를 사용하여 테스트 데이터셋에서의 검출 결과를 나타낸다. 비록 데이터셋이 제한적이지만 테스트 데이터에 대한 검출 결과를 분석해본 결과, 개선된 모델은 여전히 대중교통과 같은 복잡한 환경에서 검출에 적합함을 확인할 수 있다. 그리고 백화점, 교실, 길거리, 회의실 등 사람들로 붐비는 다양한 장소에서도 우수한 성능을 보였다.

## 5. 결론 및 기대효과

본 연구에서 개선하고자 참고했던 기존의 대중교통 시스템 애플리케이션에는 버스의 도착 잔여 시간을 확인할 수 있는 기능이 있다. 해당 기능을 대중교통 시스템에서 사용하기 위하여 OpenAPI를 통해 받아온 정보를 사용한다. 기존의 기능에서 혼잡도를 알 수 있는 기능을 추가하기 위해, 아두이노와 FSR 압력 센서, 적외선 광센서를 이용하여 센서값을 측정하고 블루투스 통신을 통해 애플리케이션에 표시되도록 구현하였다. 대중교통 내부의 좌석에 승객이 착석하여 FSR 압력 센서에서 무게가 감지되면, 애플리케이션에 있는 버스 좌석도가 적색으로 표시되면서 착석 여부를 나타낸다. 적외선 광센서는 출구와 입구에 각각 하나씩 설치하여 총 인원수를 파악할 수 있도록 한다.

더 나아가 차내 혼잡도 측정의 정확성을 높이기 위해 YOLOv5 객체검출 알고리즘을 적용하여 차내 CCTV를 통해 승객 수를 확인하는 기능의 가능성을 확인하였다. YOLOv5 작은 객체 및 중첩된 객체의 검출 문제를 해결하기 위해 NMS 대신 Soft-NMS 알고리즘을 적용하고, 작은 객체 검출 헤드를 추가하여 모델을 개선하였다. 개선된 모델은 기존 모델보다 성능을 향상시키는 동시에 더 작은 모델 크기로 최적화되어 테스트 데이터에서 뛰어난 검출 능력을 나타냈다.

향후 연구에서는 데이터셋 부족 문제를 보완하기 위해 더 많은 데이터를 수집하고, 학습 데이터의 다양성을 확보해야 한다. 이를 통해 모델의 성능을 한 단계 향상시키고, 보다 더 정확하고 신뢰성이 있는 대중교통 혼잡도 측정을 가능하게 할 것이다. 그리고 버스 모형에서 검증하였으므로, 실제 대중교통 환경에서의 적용 시 다양한 변수들로 인해 시스템의 성능이 다를 수 있다. 실제 대중교통 환경에서의 적용 및 검증도 필요하다.

이러한 연구를 통해 대중교통 시스템의 운영 효율성을 향상시키고, 승객들에게 안전하고 편리한 서비스를 제공할 수 있을 것으로 기대한다.

## REFERENCES

- [1] S.R.Gwon, E.S.Oh, J.M.Oh, T.K.Kim, "A Study on Real-Time Public Transportation Information," *Korea Multimedia Society Spring Conference*, Vol.26, No.1, pp.107-108, 2023.
- [2] Arduino[Internet], <https://www.arduino.cc>.
- [3] MIT App Inventor[Internet], <https://appinventor.mit.edu>.
- [4] B.J.Lee, J.K.Kim, K.S.Kim, S.H.Oh, "Stated Preference Analysis of the Impacts of Bus Crowdedness Information on Bus Choice," *Journal of Korean Society of Transportation*, Vol.26, No.6, pp.61-70, 2008.
- [5] Y.R.Jeong, S.H.Bae, "Estimation of Bus Passenger Occupancy and Degree of Congestion by Using Bus Card Data in Busan," *Journal of Transport Research*, Vol.22, No.3, pp.13-24, 2015.
- [6] Y.J.Jin, Y.M.Kim, D.H.Lee, S.H.Bae, "CCTV video data to estimate the number of people in the bus and determine the congestion level," *Journal of Korean Society of Transportation*, No.78, pp.297-302, 2018.
- [7] J.H.Lim, S.S.Park, S.B.Go, J.T.Kim, D.H.Kim, "Bus Seat Availability Prediction System based on Time Series Forecasting," *Journal of KIISE(Korean Institute of Information Scientists and Engineers)*, pp.1873-1875, 2017.
- [8] W.S.Choi, J.H.Jo, W.C.Park, S.H.Choi, "Design of YOLO-NAS Model for Safety Helmet Detection in Embedded Systems," *Journal of the Korea Contents Association*, Vol.24, No.3 pp.14-24, 2024.
- [9] M.H.Park, J.H.Choi, W.J.Lee, "Object detection for various types of vessels using the YOLO algorithm", *Journal of the Korean Society of Marine Engineering*, Vol.48, No.2 pp.81-88, 2024.
- [10] H.D.Lee, S.G.Kim, S.C.Na, J.Y.Ham, C.H.Kwak, "A Study on the Detection of Traffic Disadvantaged Persons Using Ensemble YOLOv5 Model," *Journal of the Korea Computer Information Society*, Vol.29,

No.1, pp.61-68, 2024.

- [11] D.W.Kim, J.H.Lee, "Automated Banner Crackdown System Using YOLO and OpenCV Technology," *Journal of the Society of Semiconductor and Display Technology*, Vol.22, No.4, pp.48-52, 2023.
- [12] S.W.Lee, K.H.Yoo, Y.K.Kang, J.M.Kim, C.S.Lee, "A Study on Strawberry Harvesting Robot System Using YOLO-based Object Recognition," *Journal of the Korean Society of Manufacturing Engineering*, Vol.32, No.2, pp.101-108, 2023.
- [13] Data Portal[Internet], <https://www.data.go.kr>.
- [14] YOLOv5, YOLOv5 by Ultralytics[Internet], <https://github.com/ultralytics/yolov5>.
- [15] J.H.Moon, B.Peng, J.H.Kwon, T.K.Kim, "Implementation of Smart Umbrella Stand Based on IoT," *Journal of Internet of Things and Convergence*, Vol.9, No.1, pp.57-64, 2023.
- [16] S.B.Park, Y.J.Jeong, D.E.Lee, T.K.Kim, "A Study on the Elevator System Using Real-time Object Detection Technology YOLOv5," *Journal of Internet of Things and Convergence*, Vol.10, No.2, pp.103-108, 2024.
- [17] T.K.Kim, "Spatial Crowdedness Measurement System using IoT and Amazon Web Services," *Journal of Internet of Things and Convergence*, Vol.9, No.4, pp.15-20, 2023.
- [18] R.B.Girshick, J.Donahue, T.Darrell, J.Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.580-587, 2014.
- [19] R.B.Girshick, "Fast R-CNN," *Proceedings of the IEEE international conference on computer vision*, pp.1440-1448.
- [20] S.Ren, K.He, R.B.Girshick, J.Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in neural information processing systems*, 2015.

오 은 서(Eun-seo Oh)

[준회원]



- 2020년 3월 ~ 현재 : 국립부경대학교 컴퓨터·인공지능공학부

<관심분야>

사물인터넷(IoT), 안드로이드(Android), 인공지능(AI), 데이터베이스(DataBase)

권 소 령(So-ryeong Gwon)

[준회원]



- 2020년 3월 ~ 현재 : 국립부경대학교 컴퓨터·인공지능공학부

<관심분야>

사물인터넷(IoT), 인공지능(AI), 딥러닝(Deep Learning)

오 정 민(Joung-min Oh)

[준회원]



- 2016년 3월 ~ 현재 : 국립부경대학교 컴퓨터·인공지능공학부

<관심분야>

사물인터넷(IoT), 인공지능(AI), 빅데이터(Big Data)

펑 보(Bo Peng)

[준회원]



- 2023년 2월 : 국립부경대학교 컴퓨터·인공지능공학부(공학사)
- 2023년 3월 ~ 현재 : 국립부경대학교 인공지능융합학과 석사과정

<관심분야>

사물인터넷(IoT), 컴퓨터 비전(CV)

김 태 국(Tae-kook Kim)

[종신회원]



- 2004년 8월 : 고려대학교  
전기전자전파공학부(공학사)
- 2006년 8월 : 고려대학교  
메카트로닉스학과(공학석사)
- 2014년 8월 : 고려대학교  
모바일솔루션학과(공학박사)

- 2016년 3월 ~ 2022년 2월 : 동명대학교 AI학부 교수
- 2022년 3월 ~ 현재 : 국립부경대학교 컴퓨터공학부 교수

<관심분야>

사물인터넷(IoT), 콘텐츠 전송 네트워크(CDN), 이동성,  
인공지능(AI), 빅데이터, 모바일 서비스