

# 경기도자율주행센터 데이터 활용 경진대회 :

## 도로 별 교통 혼잡도 분류

### Gyeonggi-do Self-Driving Center Data Utilization Competition : Traffic congestion classification by road

KyungHee Univ. 박건우 KyungHee Univ. 하재정

#### 요 약

본 연구는 경기도 자율주행센터 데이터 활용 경진대회의 4개의 소주제 중 도로 별 혼잡도 분류 모델 설계에 대한 내용을 담고 있다. 대상 데이터는 판교 제로시티 내 교차로에 설치된 3개의 CCTV 영상이며, 각 CCTV는 신호대기 교차로, 일반 교차로의 2개의 도로 영역을 가리키고 있다. 따라서 본 연구에서는 총 6개의 도로 영역에 대한 혼잡도 분류를 수행하기 위해 영상 영역 추출, 객체 탐지, 움직임 패턴 탐지 등의 영상 분석을 진행하여 특성 추출 및 로지스틱 회귀 모델을 설계하였고, 그 성능을 검증하였다.

#### 1. 서 론

본 연구는 경기도 자율주행 센터 데이터 활용 경진 대회의 도로 별 교통 혼잡도 분류에 대한 연구로, 교차로에 설치된 3개의 CCTV 영상을 분석해 총 6개 도로의 혼잡도를 원할, 서행, 정체 3가지로 분류하는 모델을 설계한다.

연구가 진행된 flowchart는 [그림 1]과 같다. 먼저 [그림 2]에 주어진 데이터 라벨링 기준에 맞춰 특성을 추출하기 위해 CCTV 영상 업스케일링 및 프레임 보간을 적용하여 객체 탐지를 수행하였다. 해당 객체 탐지 데이터를 활용하여 신호 대기 교차로 (도로 1)의 경우에는 차량 수(평균, 최소, 최대)를 특성으로 추출했고, 일반 교차로 (도로 2)의 경우에는 차량 속도 증감 계수와 차량 밀집도를 특성으로 추출했다.

이후 해당 특성들을 활용하여 혼잡도를 분류하는 단순 회귀 모델인 Logistic Regression을 각 도로마다 설계했고, 최종 정확도 94.51%를 달성했다.

#### 2. 관련 연구 및 라이브러리

##### 2.1. Open cv (Open-Source Computer Vision)

Open CV는 컴퓨터 비전 오픈소스 라이브러리로, 영상 처리를 위한 여러 알고리즘을 가지고 있다. 주요 기능으로는 이미지 및 비디오 입출력, 이미지 이진화, 노이즈 제거, ROI (Region of Interest) 설정 등이 있다. 본 연구에서는 Open CV의 알고리즘을 활용하여 ROI를 통한 도로 별 영상 및 정적 배경 차분, 원근 변환 등을 수행한다.

##### 2.2. waifu2x(업스케일링 및 프레임 보간)

waifu2x는 영상 업스케일링 및 프레임 보간 서비스를 제공하는 위한 오픈 소스 소프트웨어이다.

업스케일링이란 이미지나 동영상의 픽셀 사이 사이에 근처 픽셀의 색 데이터를 참조하여 만든 새로운 픽셀을 끼워 넣어 영상의 해상도를 향상하는 기법이다. 프레임 보간은 연속하는 두 개의 프레임 사이에 그 중간 프레임을 생성하여 끼워 넣는 기법으로, 프레임률을 늘림으로써 영상에서의 모션을 더 부드럽게 만들거나, 슬로우 모션의 효과를 만들어낼 수 있다. 본 연구에서는 해당 오픈 소프트웨어를 통해 CCTV 영상 업스케일링 및 프레임 보간을 진행하였다.

##### 2.3. YOLOv7 (You Only Look Once)

YOLOv7는 대표적인 2D 객체 탐지 알고리즘으로, 이미지 또는 영상에서 특정 객체를 분류하고, 해당 객체의 위치 정보를 반환해 주는 기능을 가지고 있다. 같은 기능을 하는 알고리즘으로 R-CNN (Region-Convolutional Neural Network)이 있다. R-CNN의 경우 학습 및 분류 시 하나의 이미지(프레임)를 여러 장으로 나누어서 사용하는 반면, YOLO는 이미지를 단 한 번 사용하기 때문에 소요 시간의 측면에서 월등히 앞선다. 따라서 본 연구에서는 후속 연구를 통한 실시간 혼잡도 분류 모델로의 확장을 고려하여 R-CNN이 아닌 YOLO를 선택했다.

2.4. 원근변환

원근변환(Perspective Transformation)은 직선의 성질만 유지가 되고, 선의 평행성은 유지가 되지 않는 변환 함수로서, 4 개의 점의 이전 위치와 변환 후의 위치를 지정하여 마치 시점을 바꾼 것과 같은 변환을 발생시킨다. 이전과 이후의 8 개의 점을 이용하여 3\*3 투시변환행렬을 생성한 후 이를 각 픽셀에 적용하여 이미지를 변환시킨다.

2.5. Gunner Farneback 알고리즘

Gunner Farneback 알고리즘은 광학 흐름 알고리즘의 일종이다. 광학 흐름(Optical)이란 이전 프레임과 다음 프레임 간 픽셀이 이동한 방향과 거리로 영상 내 물체의 움직임 패턴을 추적하는 알고리즘을 말한다. 따라서 광학 흐름은 영상 내 물체가 움직임을 분석하거나, 나아가 예측하는 데에 사용된다. Gunner Farneback 알고리즘은 2003 년 Two-Frame Motion Estimation Based on Polynomial Expansion 에서 Gunner Farneback 이 제시한 알고리즘으로, 확장 다항식을 기반으로 두 프레임 사이의 움직임을 계산한다. Dense Optical Flow 에 속하며 연산량이 크지만 정확도가 높다는 장점이 있다.

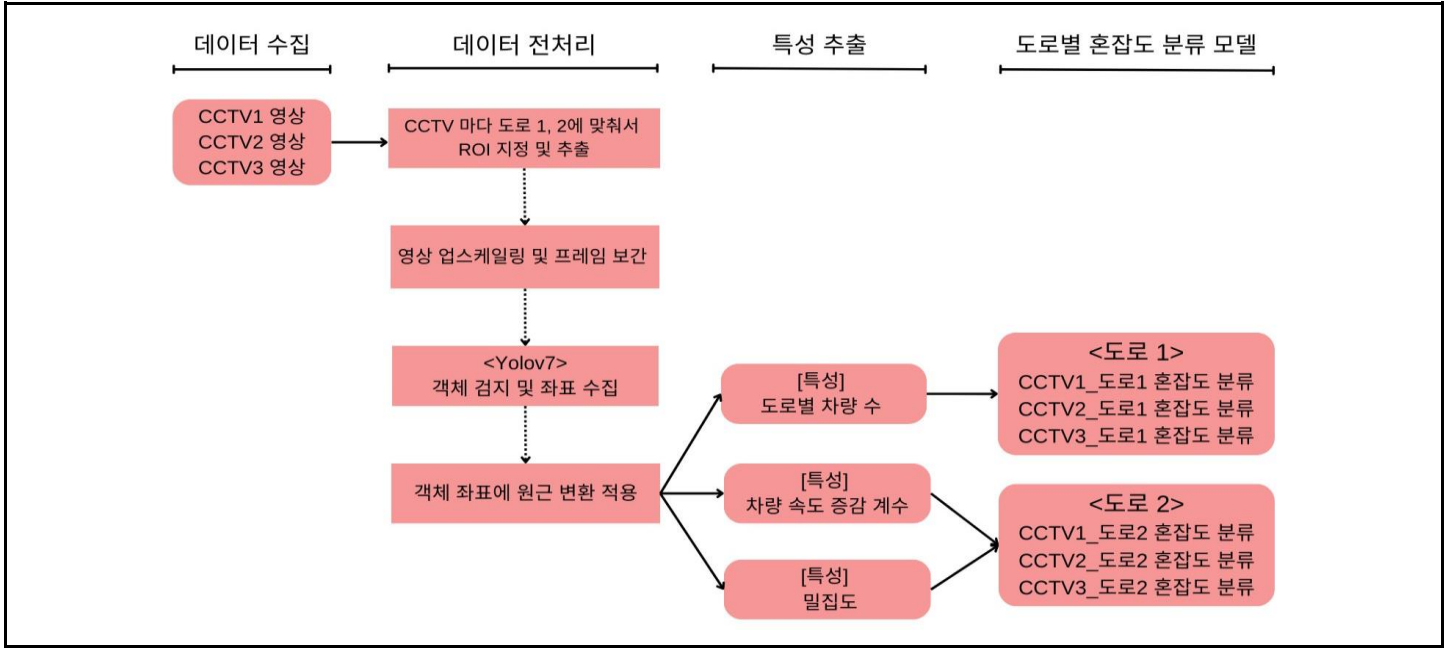


그림 1. Flow Chart

	신호 대기 교차로	일반 교차로
원활	대기하는 차량이 4대 미만인 경우	차량 속도 유지, 차량 간 간격이 2m 이상
서행	대기하는 차량이 4대 이상 8대 미만인 경우	차량 속도 감속, 차량 간 간격이 1m 이상 2m 미만
혼잡	대기하는 차량이 8대 이상인 경우	차량 속도 감속, 차량 간 간격이 1m 미만

그림 2. 혼잡도 라벨링 기준



그림 3. CCTV 별 도로 영역

3. 영상 데이터 전처리

3.1 영상 영역 추출 (ROI)

ROI (Region of Interest)란, 말 그대로 영상에서 관심 있는 영역만을 추출해 내는 기법이다. 이번 연구의 목적은 각각 2개의 도로 영역을 가리키는 3개의 CCTV 영상을 분석해 총 6개 도로의 혼잡도를 각각 분류해 내는 것이다. 따라서 6개의 도로마다 세밀한 영상 분석을 진행하기 위해 OpenCV의 bitwise, fillpoly 등의 함수들을 사용하여 세 종류의 CCTV 영상을 6개의 도로 별 CCTV 영상으로 분할하였다.[그림3,4,5]



그림 3. CCTV 1 원본



그림 4. CCTV1\_도로1 ROI 지정



그림 5. CCTV1\_도로1 ROI 추출 및 확대

3.2 차영상, 배경차분 (Background Subtraction)

차영상 기법이란, 인접한 프레임 사이의 픽셀 별 밝기 변화를 측정하고, 변화가 있는 영역을 객체의 이동 영역이라고 추정하는 간단한 객체 검출 기법이다. Ali Tourani[1]은 차영상 기법을 활용하여 이미지에서의 차량 수를 counting하는 연구를 진행했고, 약 95%의 정확도를 달성했다.

하지만 이 기법은 이동 중인 물체는 손쉽게 탐지할 수 있지만, 이동 물체와 배경의 색상이 비슷하거나, 장기간 정지한 물체에 대해서는 탐지가 불가능하다는 단점이 있다. 본 연구에서 다루는 영상 데이터는 교차로에 설치된 CCTV 영상으로, 신호 대기라는 변수로 인해 전체 영상 (15분) 기준 1분가량 정체하는 경우가 많다. 따라서 장시간 멈춰 있는 객체도 탐지해 내기 위해 차영상 기법이 아닌 정적 배경 차분 기법을 선택했다.

정적 배경 차분이란, 사전에 등록된 배경 이미지와 현재 입력 프레임과의 차영상을 이용하여 객체를 검출해 내는 객체 검출 기법이다. 특정 이미지를 영상의 모든 프레임에서 제거하기 때문에 영상에서의 조명 변화가 나타날 경우 객체 검출 능력이 떨어지는 단점이 있다. 하지만 본 연구에서는 15분이라는 짧은 시간의 연속된 CCTV 영상을 다루기 때문에 해당 문제는 발생하지 않았다. 배경 이미지의 경우 차량이 아예 없을 때의 이미지를 기존 영상에서 추출하거나, 불가피한 경우 ai 지우개 프로그램을 통해 도로마다 객체가 없는 순수한 배경 이미지를 만들어 사용하였다. [그림 6,7,8]



그림 6 . 차량이 없을 때의 이미지 추출 (CCTV\_1\_1)



그림 7. 차량이 없는 순간이 없는 경우 (CCTV\_3\_1)



그림 8. 그림 7에서 AI 지우개로 객체를 제거한 이미지

이를 통해 멈춰 있는 객체도 감지할 수 있는 차영상을 제작하는 데에 성공했다. [그림 9] 하지만, 신호 대기로 인하여 차가 매우 근접하거나, CCTV 화각 상 각 차량들이 매우 가깝게 위치해 있는 경우에는 객체가 하나로 인식되는 문제가 발생했다. 이에 더해 도로 밖의 사람도 객체 수로 인지되는 문제가 발생하여 객체 수의 정확한 탐지가 중요한 본 연구에 차영상 및 정적 배경 차분 기법은 부적절하다는 판단과 함께 최종 모델에 사용하지 않았다. [그림 10, 11]





그림 9. 멈춰있는 객체도 탐지해 낸 이미지



그림 10. 정차한 차들이 하나로 인식되는 이미지



그림 11. 사람도 인식된 이미지

### 3.3 객체 추적(YOLOv7)

차영상의 대체 알고리즘으로써 YOLOv7 알고리즘을 사용하였다. YOLOv7의 경우 차영상, 배경차분과는 달리 이전 프레임과는 관계없이 현재 프레임만을 고려하여 객체를 탐지하기 때문에 3.2에서와 같이 정차한 차들을 하나로 인식하는 등의 문제는 발생하지 않았다. [그림 12]



그림 12. 정차한 차들도 인식하는 데에 성공한 이미지

또한 YOLO는 탐지된 객체들의 위치를 사각형 형태의 바운딩 박스로 표현하고, 해당 박스의 좌상단, 우하단의 좌표를 반환한다. 통상 해당 좌표들의 중점(사각형의 중앙)을 객체의 위치로 가정하는 경우가 많지만, 본 연구에서 다루는 객체는 도로 위의 차량이므로, 도로에서 가장 가까운 아래쪽 모서리 좌표의 중점을 객체의 위치로 가정하고 연구를 진행하였다.

### 3.4 영상 업스케일링, 프레임 보간

YOLO를 통해 높은 수준의 객체 탐지 및 추적 성능을 확인할 수 있었다. 하지만 영상 중간중간 객체를 놓치는 경우가 존재했고, 그 원인이 CCTV 영상의 낮은 화질과 낮은 프레임률이라고 판단했다. 따라서 객체 수 탐지 능력 향상과 추후에 특성으로 활용할 객체 사이 거리를 보다 정확히 계산하기 위해 CCTV 영상의 업스케일링과 프레임 보간을 진행했다. 업스케일링 및 프레임 보간 프로그램은 waifu2x-extension-GUI를 사용했다. [그림 13, 14]을 보면 업스케일링 후 객체 탐지 능력이 기존보다 향상된 것을 확인할 수 있다.



그림 13. 업스케일링, 프레임 보간 전



그림 14. 업스케일링, 프레임 보간 후

### 3.5 YOLO 전이학습

영상 업스케일링, 프레임 보간을 진행하자 YOLO를 통한 객체 탐지 및 트래킹 성능이 전반적으로 향상되었다. 하지만 CCTV3의 도로1과 도로2의 경우 CCTV로부터 도로가 너무 멀리 있었기 때문에 영상 업스케일링과 프레임 보간을 진행했음에도 탐지하지 못하는 객체가 다수 존재했다. 그중에서도 도로2의 가장 우측 차선의 경우 객체 탐지의 신뢰성이 매우 낮았다. [그림 15]

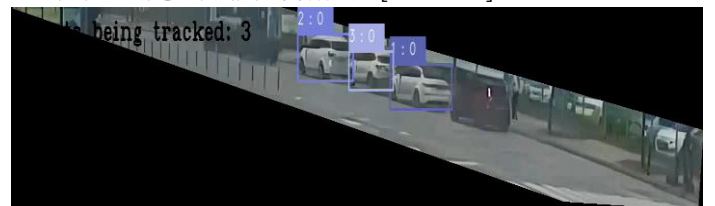


그림 15. 일부 개체를 탐지하지 못하는 이미지

해당 차선의 경우 다수의 차량이 무단 정차하고 있는 것으로 확인되었는데 혼잡도 라벨링과 비교했을 때 혼잡도에는 영향을 주지 않은 것으로 확인되었다. 따라서 해당 차선은 ROI에서 제거하더라도 현재 test 영상들에 대한 혼잡도 분류 성능에는 영향이 없을 것으로 예상된다. 하지만 실제 모델로서의 활용도를 생각했을 때 해당 차선에 무단 정차 차량이 없는 test data가 들어왔을 때 혼잡도를 분류해낼 수 있어야 한다고 판단했다. 따라서 본 연구에서는 CCTV3 도로 2에서 정차한 차량을 포함한 모든 차량의 좌표를 YOLO로 검출해 낸 후 영상 동안 좌표변화가

임계값 이하일 경우 객체 정보를 추출하지 않도록 프로그램을 설계했다.

우선 CCTV3 도로 2의 객체들을 모두 탐지해 내기 위해 해당 도로의 영상을 1초마다의 이미지들로 변환하여 총 900장 중 450장을 ROBOFLOW 웹사이트에서 이미지 라벨링을 진행하였다.

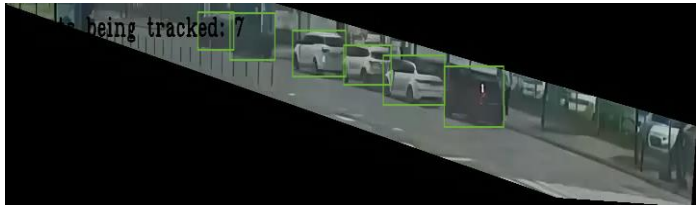


그림 16. 라벨링한 이미지

이후 해당 데이터를 YOLOv7 모델을 전이 학습시켜 CCTV3의 도로2 객체도 탐지하는 데에 성공했다.[그림 16]

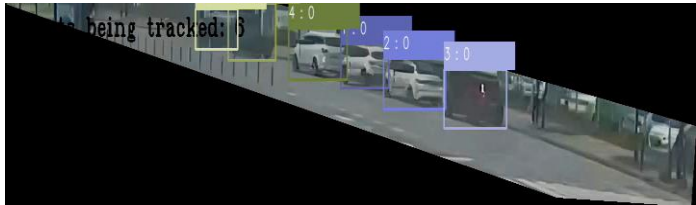
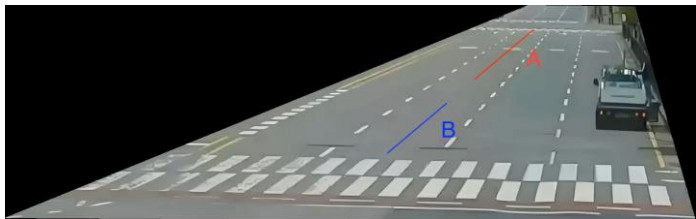


그림 17. 전이 학습 후 객체 탐지에 성공한 이미지

### 3.6 원근변환

업스케일링 및 프레임 보간과 전이학습을 통해 각 도로에서 객체의 좌표를 보다 정확하게 탐지하는 데에 성공했다. 이후 일반 교차로 상황에서 “서행”과 “정체”를 분류 기준인 객체 사이 거리를 계산하기 위해 YOLO로 구한 객체들의 좌표에 원근 변환을 적용하여 실세계에서의 좌표를 계산했다. 원근 변환을 적용하기 전의 객체들의 좌표는 화면에 표시되는 사각형 바운딩 박스의 좌표로, top-view (항공 이미지처럼 상공에서 수직 아래로 내려다봤을 때의 좌표)가 아니기 때문에 실세계에서의 객체의 좌표 및 객체 사이의 거리를 대변할 수 없다.

예를 들어, 그림 A에서 선분 A와 B는 길이가 같지만, CCTV에서의 거리, 각도를 고려한다면 실세계에서는 A의 길이가 B보다 길 것이다. [그림 18]



<그림 18. 원근 변환 전 이미지>

즉 YOLO의 좌표는 실제 도로에서의 객체의 위치를 대변하지 못한다는 것이다. 따라서 본 연구에서는 YOLO를 통해 알아낸 객체의 좌표를 실세계에서의 좌표로 변환하기 위해 각 도로 영상마다 원근 변환을 적용했다. [그림 19]

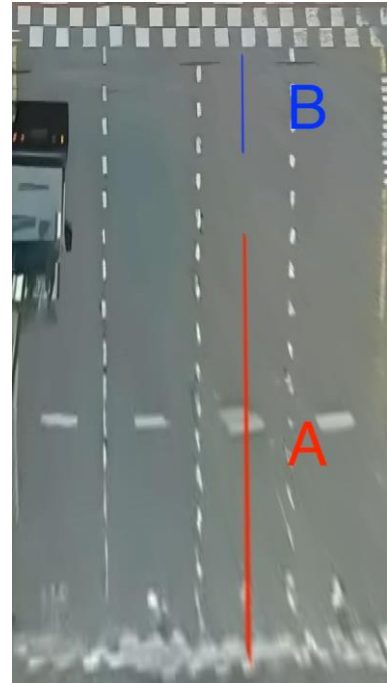


그림 19. 원근 변환 후 이미지

[그림 19]을 보면, 동일한 길이였던 선분 A와 B가 원근 변환을 통해 실제로 top-view에서 봤을 때의 길이로 변환되자, 상대적으로 멀리 있던 선분 A의 길이가 선분 B보다 더 길어진 것을 확인할 수 있다. 결론적으로 원근 변환을 통해 CCTV 화면에서의 객체의 좌표를 실세계에서의 좌표로 변환하는 데에 성공했고, 이를 활용하여 도로 내 객체의 밀집도를 더욱 정확하게 계산하여 특성으로 활용하였다.

## 4. 특성 추출

특성의 경우 [그림 2]와 [그림 3]의 혼잡도 분류 기준 및 CCTV 도로 영역에 따라 추출하였다. 각 CCTV 영상의 도로1의 경우 일반 교차로이므로 차량의 수 (평균, 최대 최소값)를 특성으로 사용하였고, 도로2의 경우 신호 대기 교차로 이므로, 속도 증감 계수와 밀집도 (평균, 최대, 최소)를 특성으로 사용하였다.

### 4.1. 도로1 특성

신호대기 교차로인 도로1 영역의 혼잡도 분류 기준은 차량의 수이다. [그림2]

#### 4.1.1. 차량의 수

차량의 수는 YOLOv7을 통하여 검출하였다. ROI를 통해 각 도로에 대해 비교적 세밀한 차량의 수 탐지가 가능했으나, CCTV 영상의 화각으로 인해 반대편 차선의 차량, 직교하는 차선의 차량, 좌회전 차량이 추가로 탐지되는 등의 문제가 발생했다. [그림20, 21, 22]

[그림 20]과 같이 반대편 차선의 차량이 탐지되거나, 그림 [그림 21]과 같이 직교하는 차선의 차량이 차량의 수에 포함되지 않도록 하기 위해, 원근 변환된 프레임 상에서

주어진 차선 좌표에서 벗어난 차량을 차량 수에서 제거하였고, [그림 22]와 같이 좌회전 차량이 차량수에 포함되는 것을 방지하기 위해 횡단 보도 구역에서 x축 방향으로의 좌표 변화량이 큰 차량을 차량 수에서 제거하였다.



그림 20. 반대편 차선의 차량이 같이 검출되는 이미지

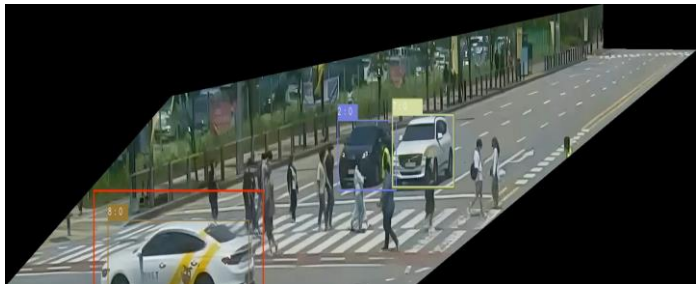


그림 21. 직교하는 차선에서 주행하는 차량

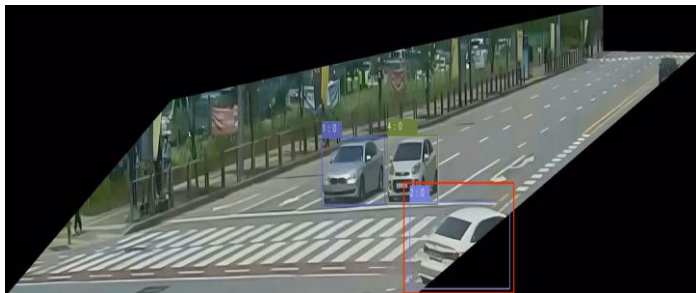


그림 22. 좌회전 차량이 차량 수에 포함되는 것

CCTV의 화각으로 인한 차량 수 탐지 오류 외에도 하나의 객체에 두개 혹은 그 이상의 바운딩 박스가 형성되어 복수의 객체로 인식되는 YOLO 알고리즘의 한계로 인한 차량 수 탐지 오류가 발생하였다. [그림 23]



그림 23. 하나의 객체에 두 개 이상의 바운딩 박스 형성 (보라색, 연보라색)

이러한 객체들을 하나의 객체로 카운팅할 수 있도록 좌표가 매우 가까운 (도로마다 임계값 설정) 객체들은 하나로

카운팅되도록 조건을 추가했다. 이후 각 CCTV의 도로1 영상을 대표할 수 있는 차량 수 지표를 만들기 위해 위의 조건 아래에 10프레임마다 계산한 차량 수의 평균인 mean, 최대인 max, 최소인 min을 특성으로 사용하였다.

## 4.2. 도로 2 특성

일반 교차로인 도로2의 혼잡도 분류 기준은 차량 속도 유지 혹은 감소 여부, 차량 간 간격이다. [그림 2]

### 4.2.1. 차량 속도 증감 계수 (Gunner Farneback)

Gunner Farneback 알고리즘을 활용하여 각 개체마다 매 프레임에서의 움직임을 알 수 있었다. 그러나 차량 수 탐지에서와 마찬가지로 직교하는 차선 방향으로 움직이는 차량, 반대 차선에서 주행하는 차량의 속도 증감 계수가 같이 검출되는 오류가 발생했다.

이를 해결하기 위해 원근변환이 적용된 프레임 상에서 각 움직임을 벡터의 +y 방향의 벡터만을 모두 합하여 ‘속도’로 계산하였다. [그림 24]

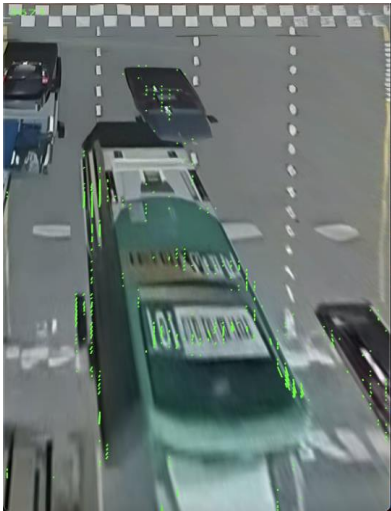


그림 24. 원근 변환된 차량의 이동 방향 및 속도의 크기

이후 각 프레임에서 발생하는 속도 값을 회귀직선으로 구현했을 때의 회귀 계수를 속도 증감 계수로 활용했다. [그림 25]

부호가 양일 경우 가속, 음일 경우 감속이며 어노테이션에 주어진 차량 속도 유지 또는 감소의 기준이 모호하기 때문에 속도의 방향과 크기가 그대로 특성에 반영되도록 하였다.

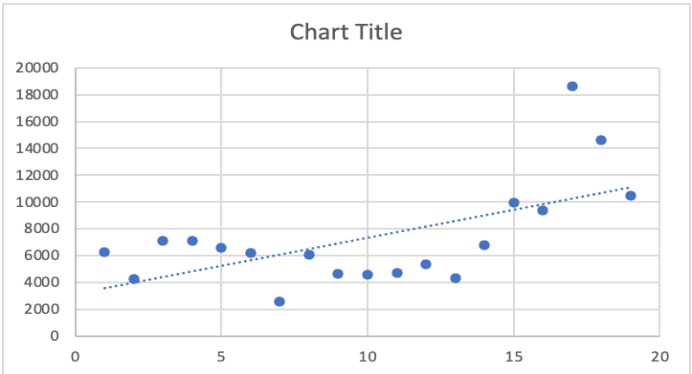


그림 25. 각 프레임마다의 속도 값의 합계로 구한 회귀직선

#### 4.2.2. 차량 밀집도

일반 교차로에서 주어진 혼잡도 분류 기준은 차량 간의 간격이다. 일반적으로 도로 교통에서의 차량 간격은 앞차와 뒤차의 간격을 의미한다. 하지만 “혼잡도”의 측면에서 앞, 뒤차의 차량 간격만을 고려하는 것보다는 앞뒤 양옆을 모두 고려하는 것이 일반화 성능에 도움이 될 것이다. 이를 대변하는 값이 바로 객체의 밀집도이다.

다시 말해 “개별 차선의 앞차와 뒤차의 거리”보다는 “차량이 전체적으로 몰려있는 정도”가 혼잡도의 지표로서 더 올바르다는 것이다. 따라서 본 연구에서는 혼잡도 분류 기준으로 차량 간격이 아닌, 객체 좌표의 밀집도를 특성으로 활용하였다. [그림 25, 26]를 보면 원할 시 밀집도가 낮고, 서행 시 원할에 비해 밀집도가 높은 것으로 보아 객체의 밀집도가 혼잡도 분류 기준으로서 타당하다는 것을 알 수 있다.

밀집도의 경우, 표준편차가 동일한 상황에서 더 많은 차량이 있을 때 밀집도가 더 높다는 통계적 사실에 근거하여 객체(차량) 좌표의 표준편차를 차량의 수로 나눠줌으로써 영상 프레임마다의 객체 밀집도를 계산했다. (D값과 밀집도는 반비례 관계이다.)

$$D = \frac{\text{Std. dev}}{\text{Number of cars}}$$

예외적으로 차가 한 대도 없거나, 차량이 한 대인 경우에는 밀집도를 계산할 수 없으므로, 아주 작은 밀집도를 나타내기 위해 100,000이라는 매우 큰 값을 부여하였다. 차량 수와 마찬가지로 각 CCTV의 도로2 영상을 대표할 수 있는 차량 밀집도에 대한 지표를 만들기 위해 위의 조건 아래에 10프레임마다 계산한 차량 밀집도의 평균인 mean, 최대인 max, 최소인 min을 특성으로 사용하였다.

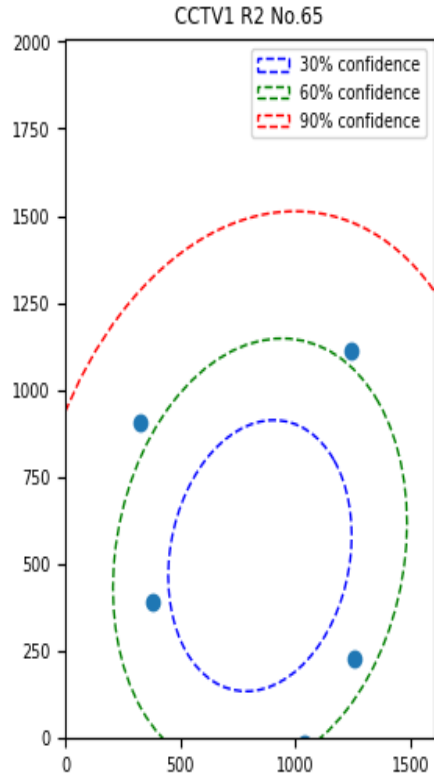


그림 25. 원할 시 밀집도 그래프

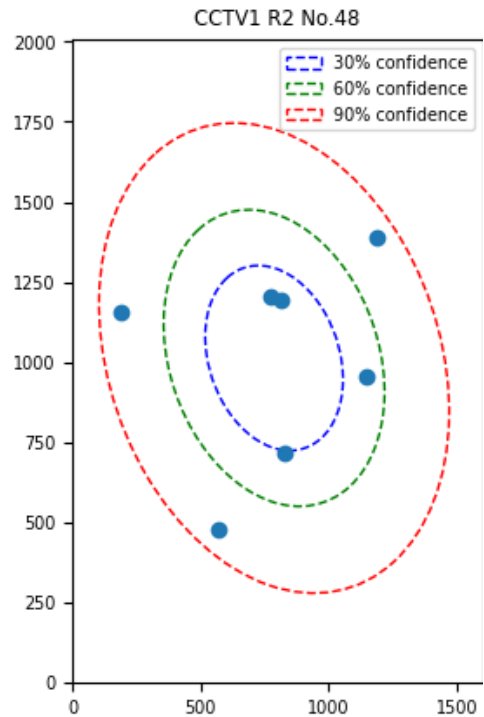


그림 26. 서행 시 밀집도 그래프



4. 모델링

앞서 추출한 도로 별 특성을 활용하여 각 도로 별 혼잡도 분류 모델을 설계하였다. 차량 수 (평균, 최소, 최대), 차량 속도 증감 계수와 차량 밀집도 등 특성들이 모두 혼잡도와 선형적인 관계가 있는 변수들로 이루어져 있다는 점에서 Base Model로 로지스틱 회귀 모델을 선택하였다. CCTV 1, 2, 3의 도로 1의 경우 현재 영상의 차량 수와 함께 이전 영상의 차량 수를 특성으로 추가했다. 15분이라는 연속된 영상이라는 점에서 이전 영상의 교통흐름이 현재 영상의 교통흐름에 영향을 줄 것으로 판단하였다. 도로 2의 경우에는 속도 증감 계수와 차량의 밀집도의 평균, 최소, 최댓값을 특성으로 활용했다.

또한 6개의 각 도로 영역 마다 우회전 및 좌회전 대기, 유턴 차선으로 인한 차선 개수 변동 등 각기 다른 특징을 가지고 있었기에 6개의 도로 마다 서로 다른 모델을 설계하여 혼잡도 분류를 수행했다.

모델 성능 평가의 경우 각 모델마다 사용할 수 있는 데이터가 매우 적기 때문에 RandomStratifiedKfold를 사용하여 5-fold 교차 검증을 수행하였다. 교차 검증 결과는 다음과 같다. [그림 27]

	Logistic_Regression	XGBoost	LightGBM
CCTV_1_1	0.96	0.97	0.95
CCTV_2_1	0.78	0.78	0.78
CCTV_3_1	0.82	0.83	0.85
CCTV_1_2	0.95	0.95	0.95
CCTV_2_2	Nan	Nan	Nan
CCTV_3_2	0.97	0.97	0.97
평균	0.89	0.90	0.90

그림 27. 교차 검증 결과 (정확도)

부스팅 알고리즘인 XGBoost와 LightGBM과 선형 회귀 기반의 Logistic\_Regression의 성능차이가 거의 없다는 것을 확인하여 가장 간단한 모형이자 해석이 쉬운 로지스틱 회귀 모델을 최종 모델로 선택하였다. 해당 모델로 주어진 Test Data를 예측한 후 제출했을 때, 최종 94.51%의 테스트 정확도를 얻을 수 있었다. 예외적으로 CCTV 2의 2번 도로의 경우 train set에 있는 타겟 라벨이 원할밖에 없었기에 모델 학습을 진행하지 않았고, CCTV3의 도로 2 모델로 학습한 모델로 예측한 결괏값을 제출하였다.

<제출 후 받은 성능>

정확도: 94.51%  
소요 시간: 0.44초

5. 결론

본 연구에서는 CCTV 영상 데이터를 각 도로 별 영상으로 나누어 각종 라이브러리를 통해 전처리하고, YOLO 모델, Gunner Farneback 기법을 통해 객체를 탐지했다. 이후 탐지된 객체들의 정보(좌표, 속도 증감 계수 등)을 활용하여 혼잡도 분류를 위한 특성을 추출했고 해당 특성들을 활용하여 6개 도로에 대한 혼잡도 분류 모델 6개를 각각 설계했다. 그 결과 교차검증 성능 약 90%, 실제 test 성능 94.51%를 달성했다.

하지만 아직 모델의 성능을 확신할 수 없다. 각 도로마다 학습용 데이터는 약 60개의 영상으로 적은 편이었고, CCTV2의 2번 도로의 경우 훈련용 영상이 모두 “원할” 영상일 정도로 대체로 불균형한 데이터셋이었다. 무엇보다 CCTV 화질 및 화각의 한계로 업스케일링, 전이 학습 후에도 놓치는 객체들이 존재하는 등의 한계점이 있었다. 후속 연구로 데이터 추가 학습 및 객체 탐지 성능 개선이 이어진다면 더 높은 신뢰도를 가진 혼잡도 분류 모델을 설계할 수 있을 것이다.



<컴퓨터 성능>

컴퓨터1 : 영상 업스케일링, 프레임 보간, 전이학습

현재 날짜/시간: 2023년 11월 24일 금요일, 오전 2:48:52

컴퓨터 이름: DESKTOP-J7URJAU

운영 체제: Windows 11 Home 64비트 (10.0, 빌드 22621)

언어: 한국어 (국가별 설정: 한국어)

시스템 제조업체: ASRock

시스템 모델: H610M-HDV/M.2 R2.0

BIOS: 5.04

프로세서: 13th Gen Intel(R) Core(TM) i5-13400F (16 CPUs)

메모리: 32768MB RAM

페이지 파일: 22280MB 사용됨, 12364MB 사용 가능

DirectX 버전: DirectX 12

디바이스

이름: NVIDIA GeForce RTX 3060 Ti  
제조업체: NVIDIA  
칩 유형: NVIDIA GeForce RTX 3060 Ti  
DAC 유형: Integrated RAMDAC  
디바이스 유형: 전체 디스플레이 장치  
전체 메모리 크기: 24336 MB  
디스플레이 메모리: 8038 MB

드라이버

주 드라이버: nvidumdx.dll,nvidumdx.dll,nvidumdx.dll  
버전: 31.0.15.4617  
날짜: 11/9/2023 09:00:00  
WHQL 서명됨: 예  
Direct3D DDI: 12  
기능 수준: 12\_2,12\_1,12\_0,11\_1,11\_0,10\_1,10\_0,9\_1  
드라이버 모델: WDDM 3.1

컴퓨터2 : 분류 모델 학습 및 실행

컴퓨터 이름: DESKTOP-16I05S8

운영 체제: Windows 11 Home 64비트 (10.0, 빌드 22621)

언어: 한국어 (국가별 설정: 한국어)

시스템 제조업체: LG Electronics

시스템 모델: 16Z90P-GA7LK

BIOS: T2ZF0320 X64

프로세서: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz (8 CPUs), ~2.8GHz

메모리: 16384MB RAM

페이지 파일: 14239MB 사용됨, 7740MB 사용 가능

DirectX 버전: DirectX 12

디바이스

이름: Intel(R) Iris(R) Xe Graphics  
제조업체: Intel Corporation  
칩 유형: Intel(R) Iris(R) Xe Graphics Family  
DAC 유형: Internal  
디바이스 유형: 전체 디스플레이 장치  
전체 메모리 크기: 8174 MB  
디스플레이 메모리: 0 MB

드라이버

주 드라이버: igdumd64.dll,igd10iumd64.dll,igd10iumd64.dll  
버전: 27.20.100.8853  
날짜: 10/14/2020 09:00:00  
WHQL 서명됨: 예  
Direct3D DDI: 12  
기능 수준: 12\_1,12\_0,11\_1,11\_0,10\_1,10\_0,9\_3,9\_2  
드라이버 모델: WDDM 2.7

컴퓨터 3 : YOLO모델 데이터 추출 및 특성 추출

Hardware

ATA  
Apple Pay  
Audio  
Bluetooth  
Camera  
Card Reader  
Controller  
Diagnostics  
Disc Burning  
Ethernet  
Fibre Channel  
FireWire

Hardware Overview:

Model Name: MacBook Pro  
Model Identifier: MacBookPro17,1  
Chip: Apple M1  
Total Number of Cores: 8 (4 performance and 4 efficiency)  
Memory: 16 GB  
System Firmware Version: 8422.141.2  
OS Loader Version: 7459.141.7.01.1  
Serial Number (system): C02FM52VQ05P  
Hardware UUID: 3B54528F-4C75-536B-90BF-F01CE7A01236  
Provisioning UDID: 00008103-0014111211D1001E  
Activation Lock Status: Enabled