# Research on Multi-class Road Obstacle Recognition and Decision Based on YOLOP Combined YOLOv5 Algorithm

Ganrong Dong, Tiancong Han, Chenxiao Feng

SCH Univ.

Dept. of AI and Bigdata

Gyeongseon Baek

baekgs011109@sch.ac.kr

# Contents

# 1. Introduction

Autonomous driving is considered a popular trend

BUT

The intelligence level is not high

The complexity of road obstacle recognition is relatively high

Conducting research on road obstacle recognition

## 2. Dataset

- S2TLD (Small to Traffic Light Dataset)
  - ➢ A traffic signal dataset presented by Shanghai Jiao Tong University
  - ➢ Comprising 10,000 traffic signal data captured under various environmental conditions and time periods
- CCTSDB (Chinese Traffic Sign Detection Benchmark)
  - ➢ A Chinese traffic sign dataset created by Jiangsu University of Science and Technology
- nuScenes
  - ➢ A large-scale open dataset for autonomous driving developed by the Motional team
- BDD100K (Berkeley DeepDrive 100K)
  - ➢ A vehicle detection and lane dataset developed by Berkeley DeepDrive, including 100,000 images

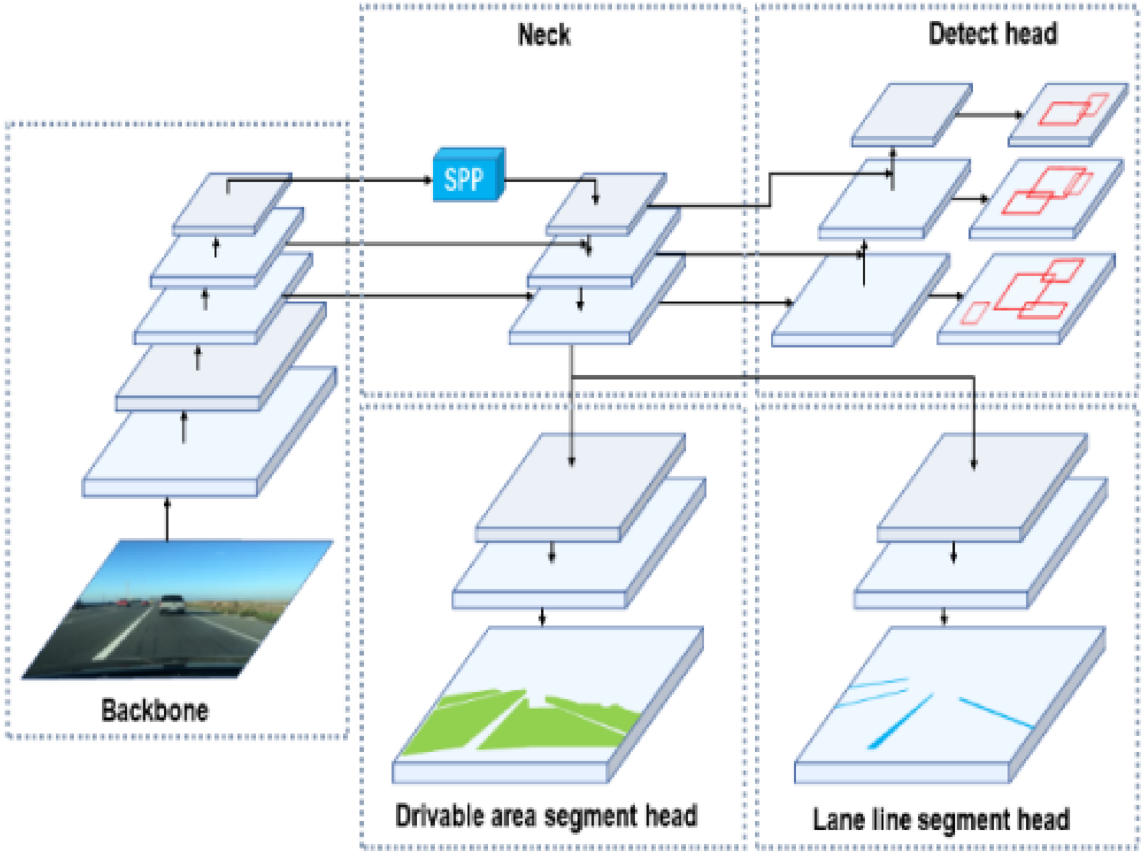
## 2. Dataset

- S2TLD (Small to Traffic Light Dataset)
  - ➢ A traffic signal dataset presented by Shanghai Jiao Tong University
  - ➢ Comprising 10,000 traffic signal data captured under various environmental conditions and time periods
- CCTSDB (Chinese Traffic Sign Detection Benchmark)
  - ➢ A Chinese traffic sign dataset created by Jiangsu University of Science and Technology
- nuScenes
  - ➢ A large-scale open dataset for autonomous driving developed by the Motional team
- BDD100K (Berkeley DeepDrive 100K)
  - ➢ A vehicle detection and lane dataset developed by Berkeley DeepDrive, including 100,000 images

# 3. Method

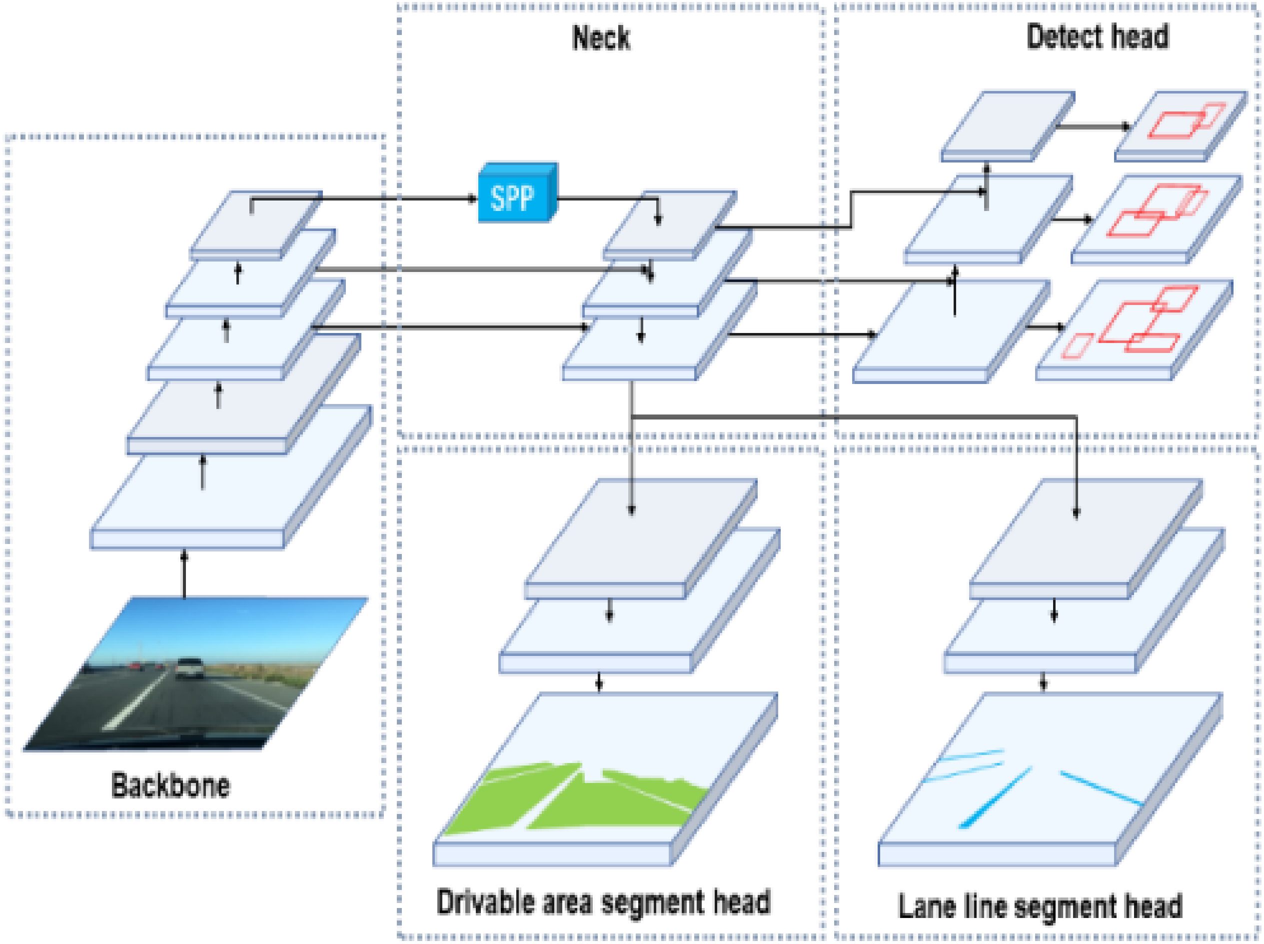
## 3.1. YOLOP (You Only Look Once for Panoptic Driving Perception)

- A deep learning model that simultaneously performs object detection and segmentation
- A method that processes three tasks simultaneously through a network
- The capability to build multi–tasks that can share information between multiple tasks



# 3. Method

## 3.1. YOLOP (You Only Look Once for Panoptic Driving Perception)




### Encorder

- Backbone
- Neck
- Extracts features from input images and transforms them into refined information

### Decorder

- Detect head
- Lane line segment head
- Drivable area segment head

# 3. Method

## 3.1. YOLOP (You Only Look Once for Panoptic Driving Perception)

$$L_{det} = \alpha_1 L_{class} + \alpha_2 L_{obj} + \alpha_3 L_{box}$$

- Loss function used exclusively in vehicle detection
- $\alpha 1$, $\alpha 2$, $\alpha 3$ are adjustable parameters for controlling the weights of each loss component

$$L_{all} = \gamma_1 L_{det} + \gamma_2 L_{da-seg} + \gamma_3 L_{ll-seg}$$

- The comprehensive loss function used in this model
- $\gamma 1$, $\gamma 2$, $\gamma 3$ are adjustable parameters for controlling the weights of each loss component

# 3. Method

## 3.1. YOLOP (You Only Look Once for Panoptic Driving Perception)

TABLE I. CAR DETECTION RESULT

| Network | Recall(%) | mAP50(%) | Speed(fps) |
|---|---|---|---|
| Faster R-CNN | 77.2 | 55.6 | 5.3 |
| DLT-Net | 89.4 | 60.2 | 8.6 |
| YOLOP | 89.2 | 76.5 | 41 |

TABLE II. DRIVABLE AREA SEG RESULT

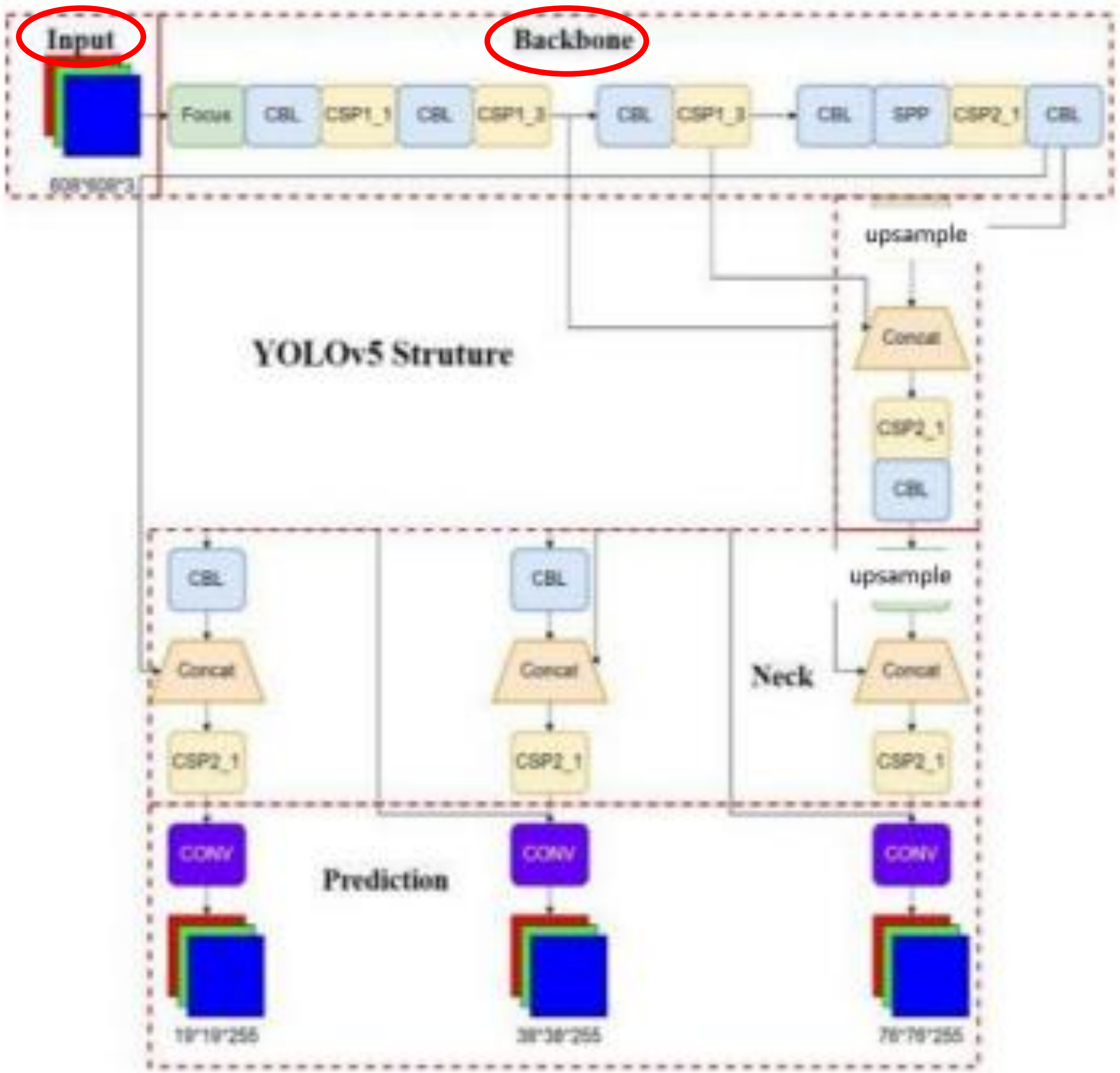
| Network | mIoU(%) | Speed(fps) |
|---|---|---|
| DLT-Net | 71.3 | 9.3 |
| PSPNet | 89.6 | 11.1 |
| YOLOP | 89.2 | 41 |

TABLE III. LANE DETECTION RESULT

| Network | IoU(%) | Accuracy(%) |
|---|---|---|
| SCNN | 15.8 | 35.79 |
| Enet-SAD | 16.0 | 36.56 |
| YOLOP | 26.2 | 70.5 |

# 3. Method

## 3.2. YOLOv5 (You Only Look Once version 5)




### Input

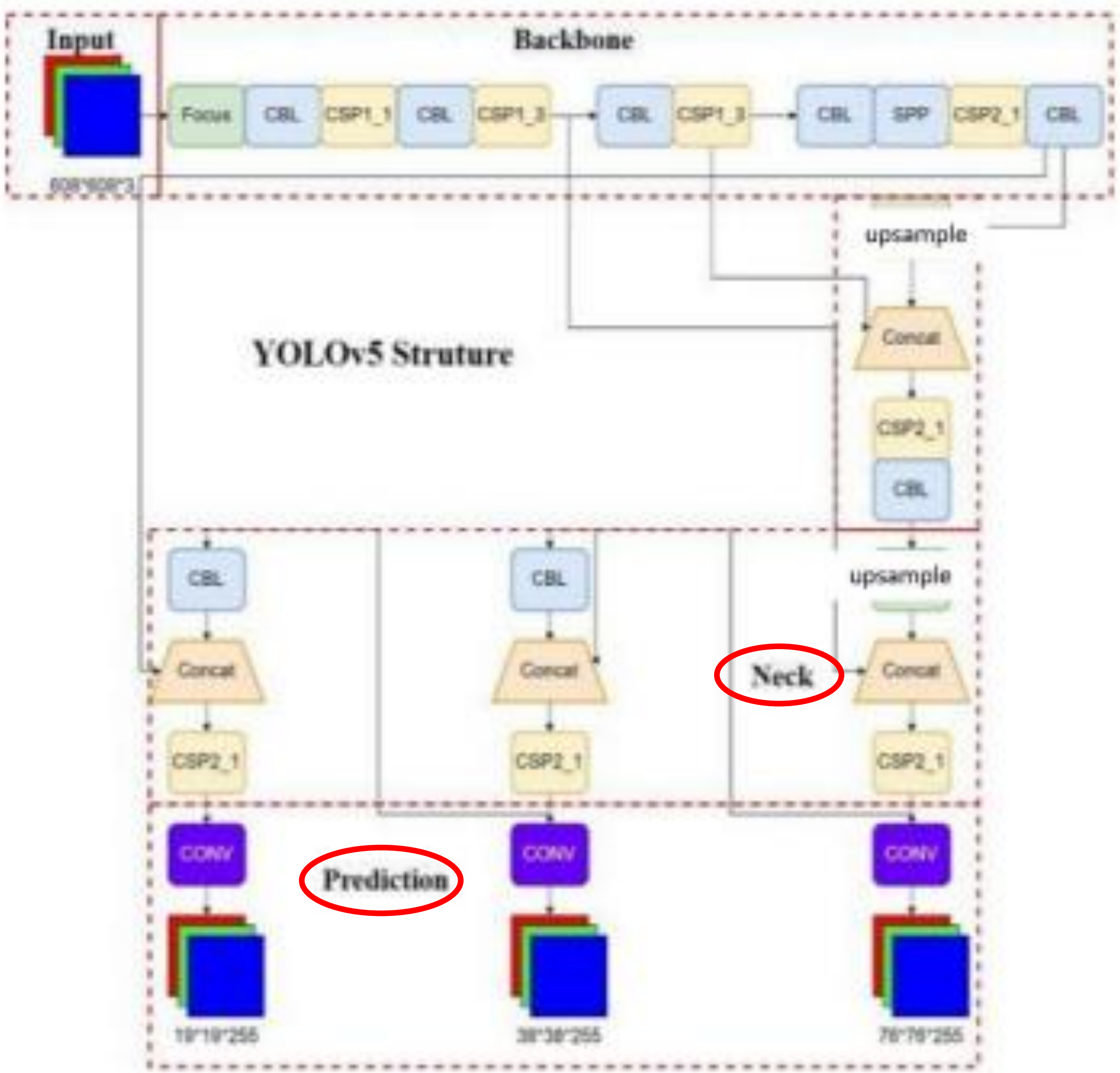
- Input data is received

### Backbone

- Extracts feature maps from the input data
- Includes Focus structure, CSP structure, which enhances inference speed

# 3. Method

## 3.2. YOLOv5 (You Only Look Once version 5)




### Neck

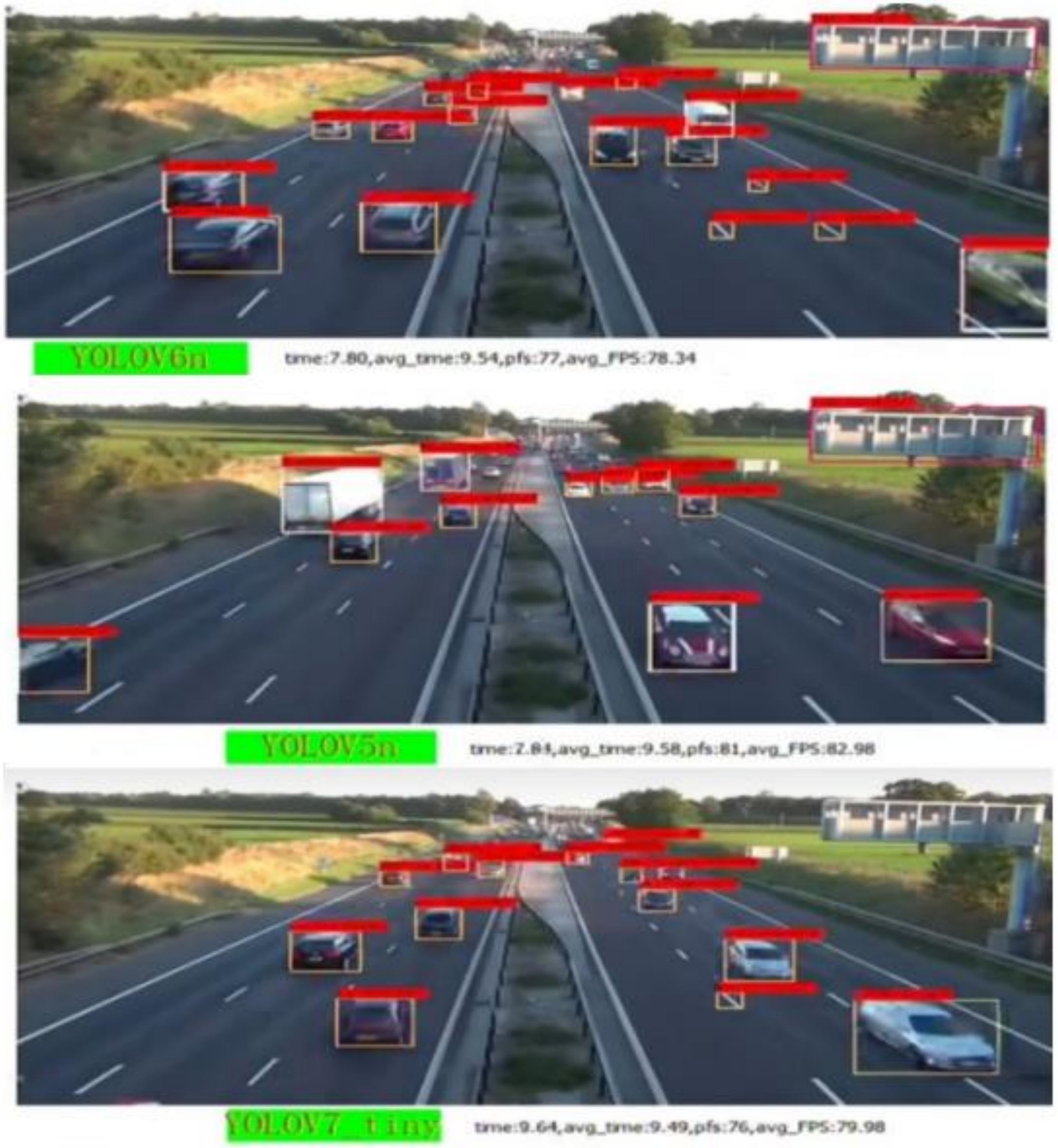
- Mixes and combines feature maps extracted from the backbone

### Prediction

- Object detection and prediction
- Preservation of boxes with the highest probability

# 3. Method

## 3.2. YOLOv5 (You Only Look Once version 5)

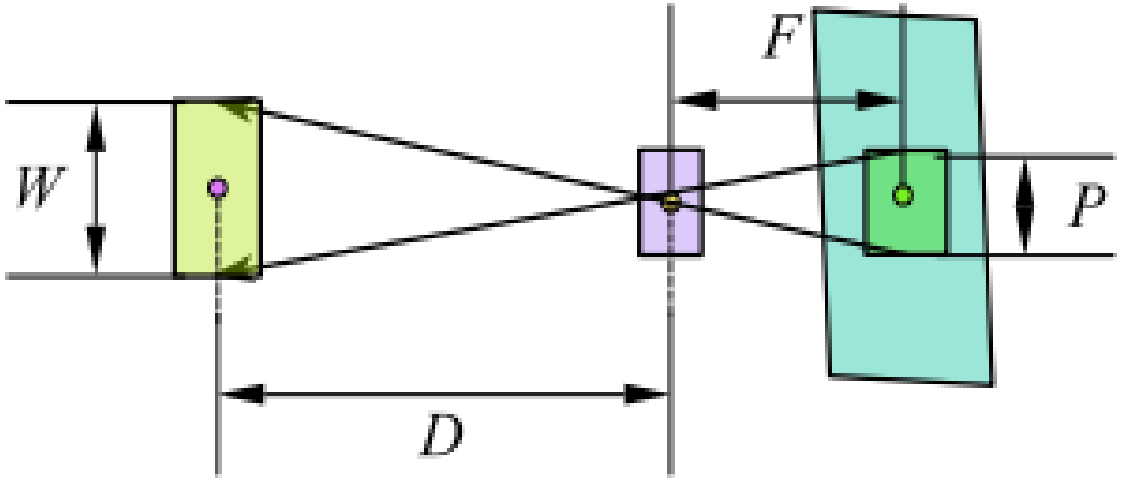



| | Average processing time | Average recognition time | Average frame rate |
|---|---|---|---|
| YOLOv6 | 7.80ms | 9.54ms | 78.34FPS |
| YOLOv5 | 7.85ms | 9.58ms | 82.98FPS |
| YOLOv7 | 9.64ms | 9.49ms | 79.98FPS |

- YOLOv5 exhibits fewer false positives and missed detections, providing a more balanced and stable result

# 3. Method

## 3.3. Similar triangle method



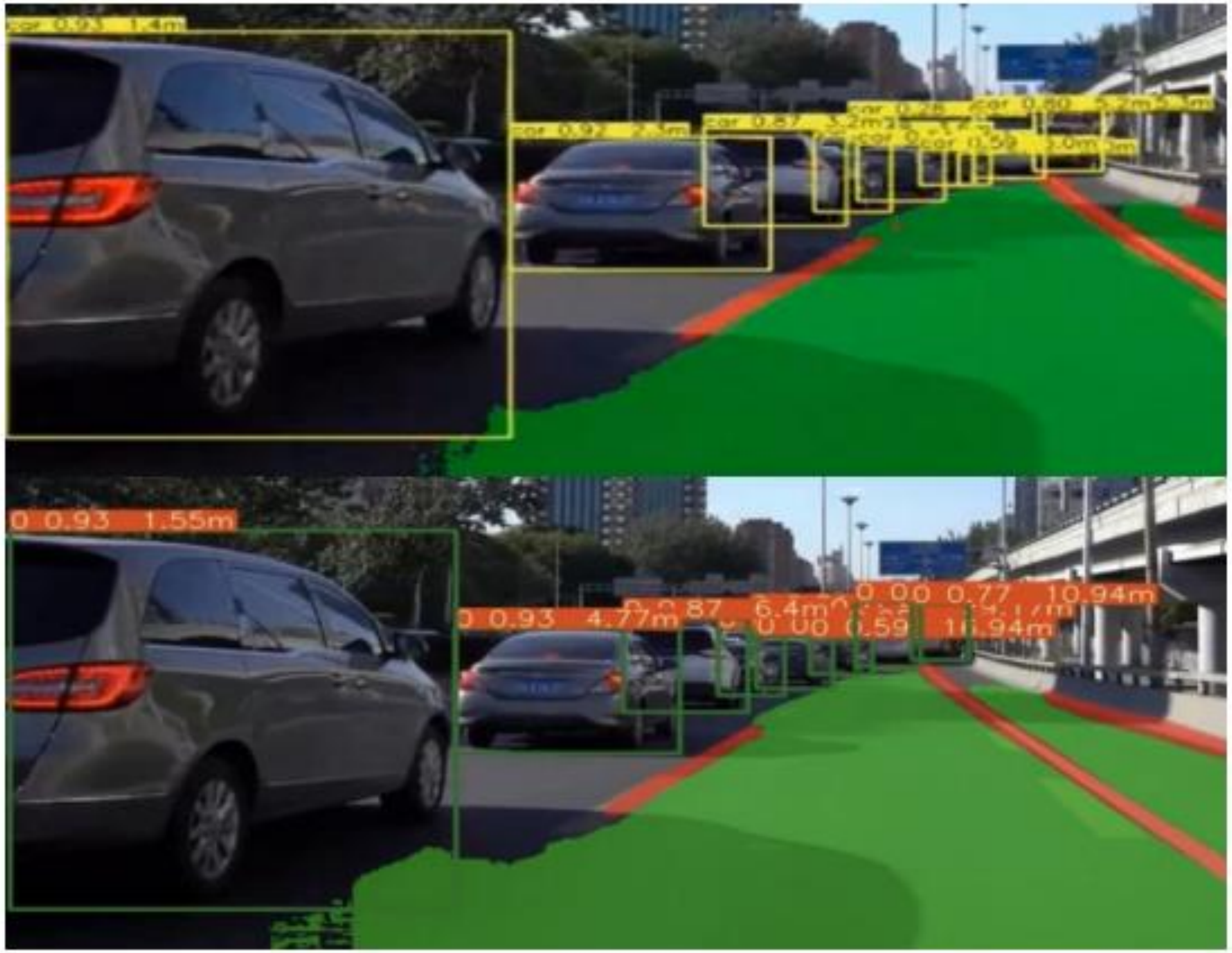

- Using the similar triangle method
- Calculating the actual distance between the object and the camera

$$D = (F * W)/P$$

- D : Distance from the target to the camera
- F : Given camera's focal length
- W : Actual height of the target
- P : Pixel width in the x-direction or pixel height in the y-direction occupied by the target in the image
  - ➢ Obtained from YOLOv5 target detection results

# 3. Method

## 3.3. Similar triangle method




**Depth estimation method**

- More precise tracking of object positions
- Requires extensive computation and may lead to longer processing times

**Similar triangle method**

- Due to its simple calculations, it is fast and efficient
- Economical and practical
- The drawback is that it requires prior knowledge of the actual object's size

# 3. Method

Lane Detection → YOLOP

Object Detection → YOLOv5

# Experimental Procedure

1. Data Conversion
   - Convert data from .xml format to YOLO format or text format
2. Preparation of YOLOv5 Format Data
   - Maintain a ratio of 9:1 for training and test sets while preparing data in YOLOv5 format
3. Cloud Training
4. Data and File Transfer
5. Virtual Environment Setup
   - Configure a virtual environment for running the YOLOv5 code
6. Model Training

TABLE IV. TRAINING SERVER PARAMETERS

| | |
|---|---|
| GPU | RTX 3080 |
| CPU | 12 vCPU Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz |
| Memory | 10GB |
| RAM | 43GB |

TABLE V. TRAINING PARAMETERS

| | |
|---|---|
| Weight | yolov5s.pt |
| Batch size | 16 |
| Workers | 8 |
| Epoch | 100 |
| Other parameters | Default |

# 4. Experiment and Result

Fig. 6. Result of a traffic lights

Fig. 7. Result of traffic signs

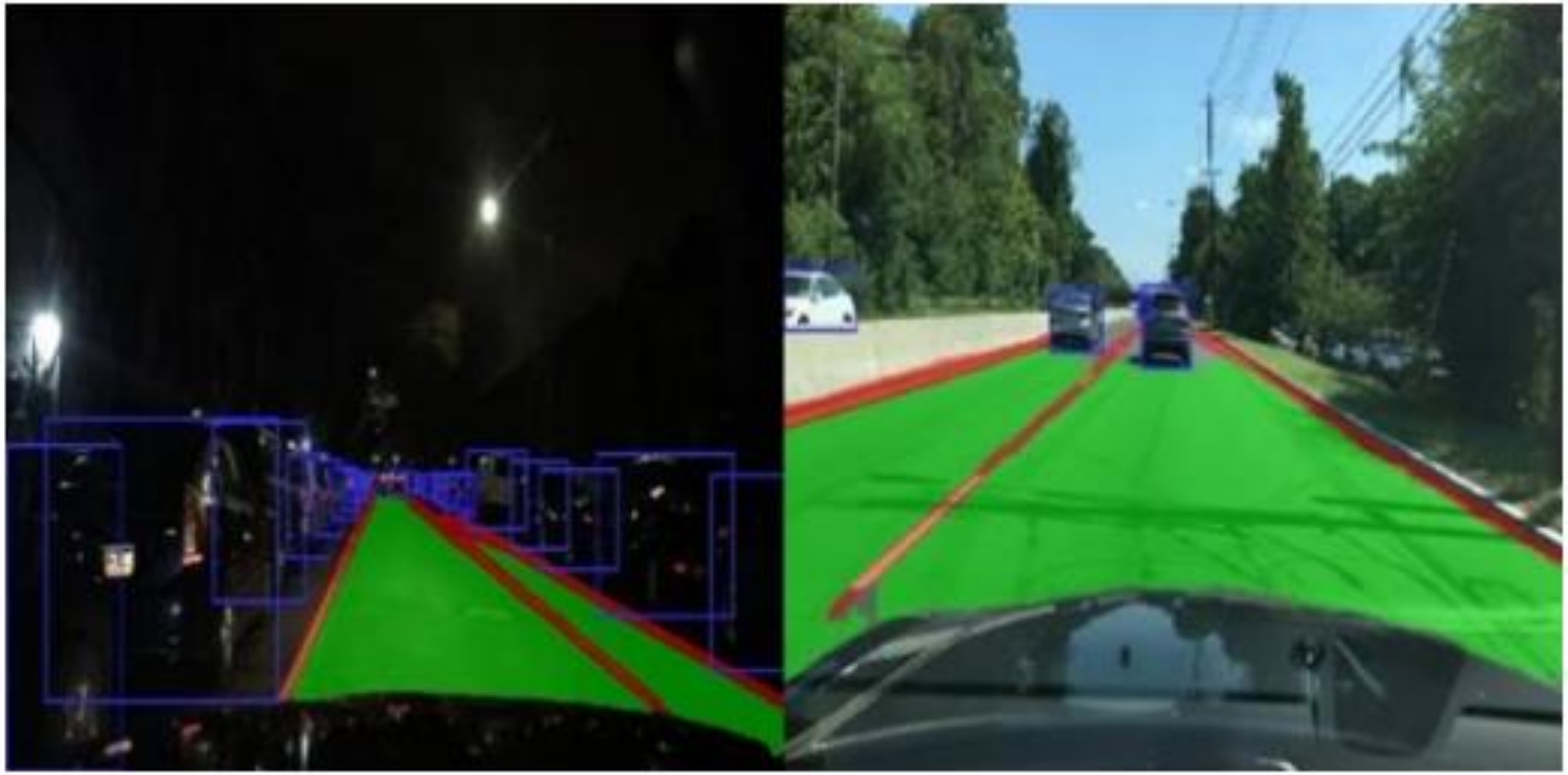
Fig. 8. Result of pedestrians

# 4. Experiment and Result

TABLE VI. TRAFFIC LIGHT, TRAFFIC SIGN, PEDESTRIAN MODEL TRAINING RESULTS

| Class | Precision (%) | Recall (%) | mAP@0.5(%) |
|---|---|---|---|
| red | 95.8 | 97.2 | 97.7 |
| yellow | 99.9 | 87.7 | 93.7 |
| green | 94.3 | 94.9 | 96.4 |
| Off | 90.7 | 92.2 | 91.9 |
| warning | 98.9 | 99.5 | 99.6 |
| prohibitory | 97.7 | 98.2 | 99.2 |
| mandatory | 99.0 | 98.5 | 99.6 |
| Pedestrian | 83.2 | 68.2 | 68.0 |

# 4. Experiment and Result



# 4. Experiment and Result


[ In the case of applying it to autonomous driving ]

# 4. Experiment and Result

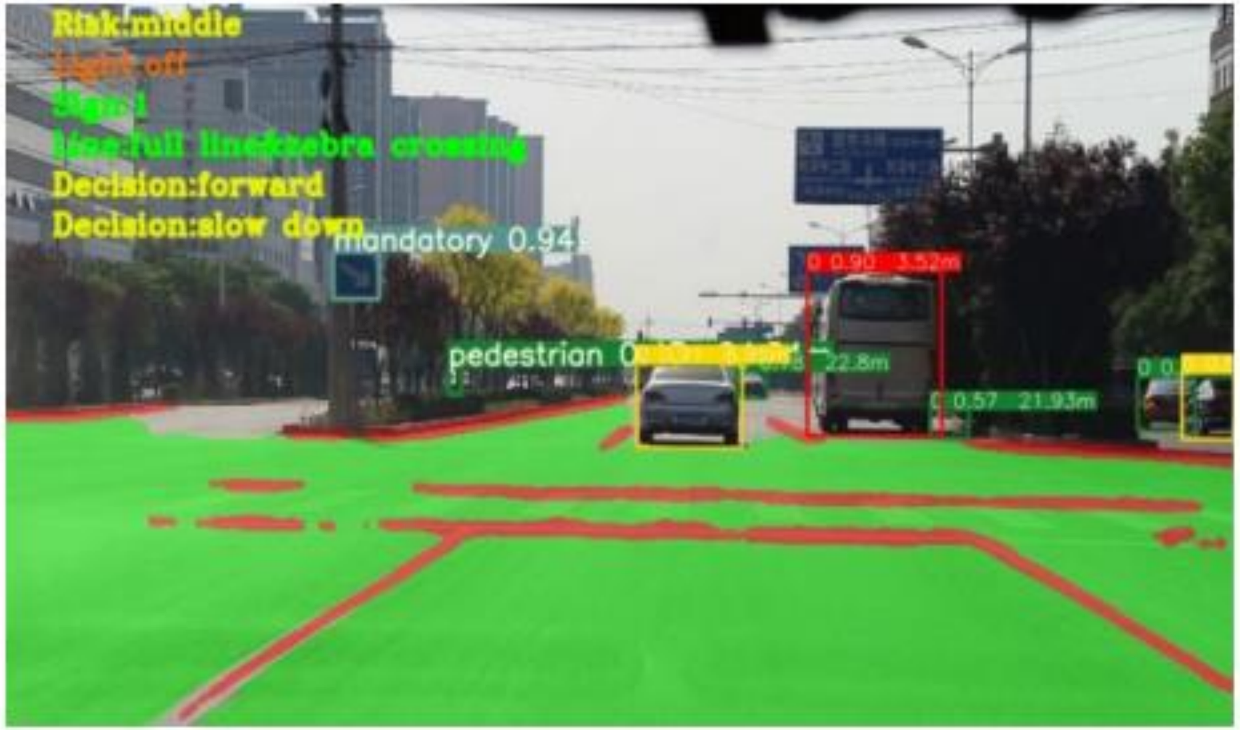
TABLE VII. PRIORITIES AND DECISIONS

| Priority | Decision |
|---|---|
| Distance | The distance between the self-driving car and the car or person in front of it is divided into three risk levels. Less than 5 meters is high risk, between 5 meters and 10 meters is moderate risk, and greater than 10 meters is low risk. According to different risk levels, make different decisions on speed. For high risk, you must slow down to a stop. If the risk is medium, it can be slowed down appropriately. If the risk is low, the speed can be maintained. |
| Traffic light | When the red light is detected, the vehicle stops and waits. If the light is green, keep going. |
| Traffic sign | There are three kinds of signs that can be recognized by the driving system: indicating signs, prohibiting signs and warning signs. The system adjusts the driving state of the vehicle according to the identification result. For example, if the no left turn sign is recognized, the vehicle is forbidden to turn left on the current road. |
| Lane line | Based on the type of lane identified by the driving system, plan the next ride. For example, if the lane line is identified as a sidewalk, reduce the speed to pass. If the dotted line is recognized, the vehicle can change lanes. |
| Driveable area | The driveable zone ensures that vehicles remain in a safe zone. It replaces lane lines when there are no lane lines. According to the results identified by the system, constantly revise the driving route. |


## Decision Priority Example 1

- If the distance to the vehicle or person in front is less than 5 meters, distance priority is the highest
- In the example shown in the adjacent figure, although the green light is recognized, since the distance from the car in front is less than 5 meters, "stop" is selected

# 4. Experiment and Result




## Decision Priority Example 2

- If the distance between the vehicle and the vehicle in front is greater than 5 meters and less than 10 meters, it is considered as medium risk

- In the example shown in the adjacent figure, a zebra crossing is recognized, so the driving system decides to go straight and pass slowly

# 5. Conclusion

- By selecting a combination of YOLOP and YOLOv5 models, excellent results are obtained
- An accuracy of over 90% is achieved in recognizing all parts except pedestrian
- When applied to actual vehicles, it yielded good results

→ To optimize the methods used in this study and make them applicable in a wider range of scenarios, it seems necessary to explore various directions for improvement

# 6. How to apply

- 드론으로 경로 탐색을 할 때, 차선 및 여러 객체를 감지하고 거리를 추정하는 데에 도움이 될 것이라고 생각함

# Thank you