ITP4514 Artificial Intelligence and Machine Learning

EA Assignment

Traffic Signs Detection and Recognition

Student Name: MAN Hok Hin

Table of Contents

1.Introduction ***3***

2. Problem Formulation ***5***

3. Methodology ***7***

4. Findings & Results ***13***

5. Summary ***19***

6. References ***20***

1.Introduction

In recent years, Hong Kong has witnessed a troubling upward trend in illegal parking incidents and traffic accidents.

Between 2020 and 2024, Hong Kong has faced increasing challenges related to traffic safety and illegal parking. According to the Hong Kong Police Force, traffic accidents involving personal injuries have shown a concerning upward trend. In 2020, there were approximately 18,360 traffic accidents, resulting in 97 fatalities, 2,021 serious injuries, and 16,242 minor injuries. Over the subsequent years, the number of incidents fluctuated but remained high, with 2024 alone recording 12,850 such incidents from January to September across various police regions (Hong Kong Police Force) [1].

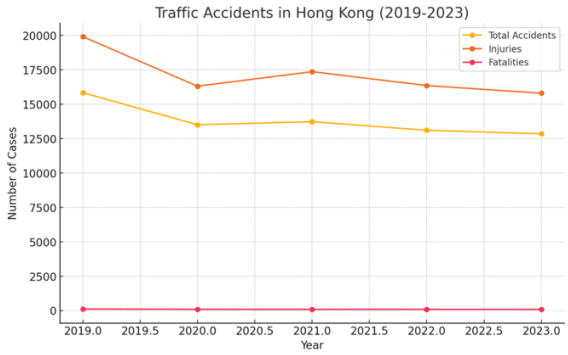


Figure 1. Hong Kong Traffic Accidents between 2019 and 2023 [1]

Illegal parking has also been a persistent issue from 2020 to 2023. In 2020, authorities issued around 2.7 million fixed penalty tickets for parking violations. This number saw a gradual decrease in the following years, with approximately 2.5 million tickets in 2021, 2.4 million in 2022, and about 2.3 million in the first three quarters of 2023 (Hong Kong Police Force) [2]. Despite these reductions, illegal parking continues to contribute significantly to traffic congestion and poses risks to public safety by obstructing emergency vehicle routes.

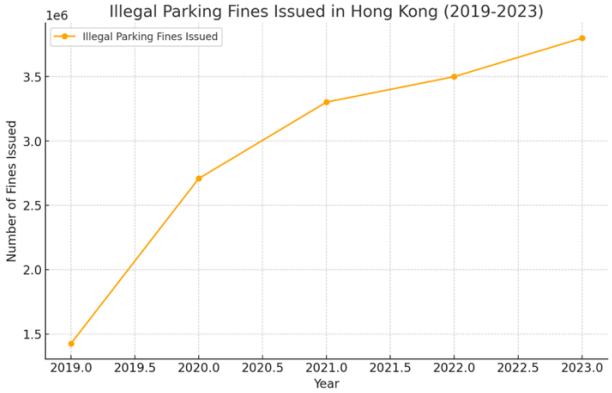


Figure 2. Hong Kong Illegal Parking Fines Issued between 2019 and 2023 [2]

To address these pressing issues, I developed Smart Car Cam, a user-friendly website designed to improve road safety and enforce traffic rules in real time. Smart Car Cam uses camera technology to quickly detect important traffic signs, such as dangerous warning signs and no-parking signs specific to Hong Kong. When a sign is detected, the system gives an immediate audio alert to remind drivers to pay attention and follow the rules. Additionally, Smart Car Cam records the detected signs, ensuring that drivers can see the signs even in poor weather conditions.

At the heart of Smart Car Cam is YOLOv8 [4], a powerful artificial intelligence tool known for its excellent ability to recognize objects. YOLOv8 allows our system to accurately and quickly identify traffic signs, ensuring real-time monitoring and prompt responses. By using this advanced AI technology, Smart Car Cam can reliably help reduce the number of accidents and illegal parking incidents.

By combining easy-to-use technology with smart safety features, Smart Car Cam aims to make Hong Kong's roads safer and more efficient for everyone.

2. Problem Formulation

Issues with Current Situation:

1. The driver has not seen that roadworks was in progress.
   Description:
   Drivers often miss noticing roadwork signs, especially in busy or congested areas, which can lead to sudden lane changes, confusion, or even accidents. [9] This issue becomes worse during bad weather, nighttime, or when the signs are not clearly visible. Missing these warnings can put both the driver and road workers at risk, leading to traffic disruptions and increasing the likelihood of collisions. Without a reliable way to alert drivers to these changes in real-time, the safety of everyone on the road is compromised.
   Solution:

   1.1 Real-time Detection of Traffic Signs:

   We build a website that can quickly and accurately find important traffic signs. For example, '300mtoleftexit', 'dangerous', 'roadnarrowsblocked', 'roadnarrowsboth', 'roadnarrowsright', 'speed50km', 'speed70km', 'speed80km', 'speed100km', 'speed110km', 'roadworks'… It has 32 object class can be detect. It can alert the user to drive be carefully and help them follow the rule.

   1.2 Recording Traffic Signs for Review
   We are driving very fast in the highway. The system can be saving detected signs automatically. The drivers can review them later. This can help them remember and follow traffic rules better.

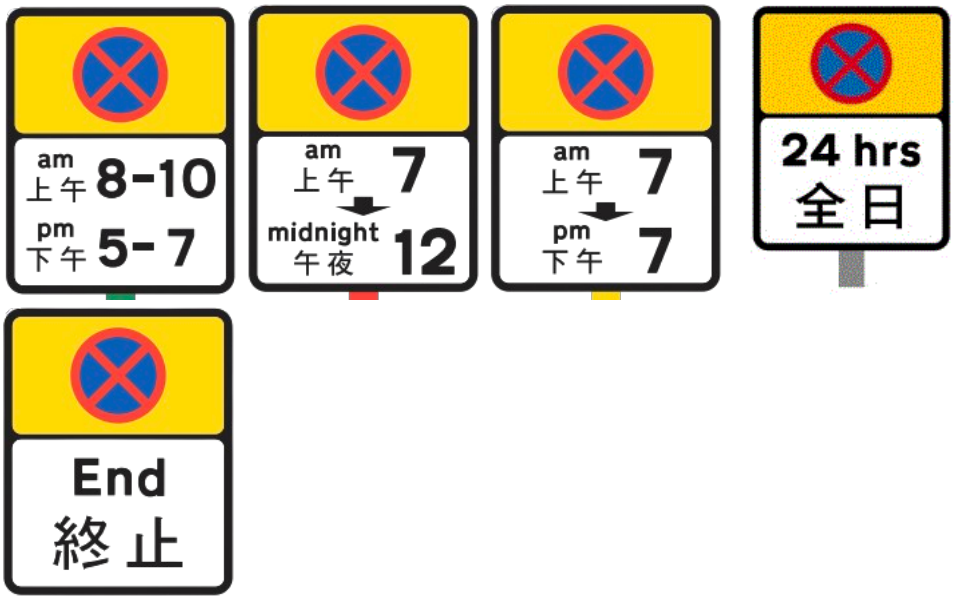
   1.3 Instant Alerts for Drivers
   The system has added an audio alert system to immediately warn drivers when the system detects a traffic sign. It can help them stay alert and drive safely.
   1.4 Live GPS location
   The system can be providing the current location. It can let them know where they are. In the most import thing is can be knowing the current speed.
   1.5 Car distance detection
   The system can be detecting the font car distance. It can alert the drive to keep the distance when they are following the car. This should reduce accidents by making drivers more aware of their surroundings.

2. The drive doesn't know here can't parking
   Description:
   Currently, many drivers are unaware of designated no-parking zones, which results in frequent parking violations. This issue contributes significantly to traffic congestion, blocks emergency routes, and leads to fines or penalties for drivers. The lack of clear visibility or understanding of no-parking areas, especially in high-density regions like Hong Kong.

   Solution:

   2.1 Real-time Detection of No parking Traffic Signs:

   To address the issue of illegal parking, we develop a website that can detect Auto detect the no parking Traffic Signs. In Hong Kong, it has 4 types of the no parking traffic sign. Such as, "8-10am 5-7pm", "7am-12am", "7am-7pm","24 hrs'. It will alert the drive where cannot parking and can help them to avoid parking in restricted areas.



3. Methodology

3.1 Software

I. Ultralytics YOLOv8

Yolov8m model

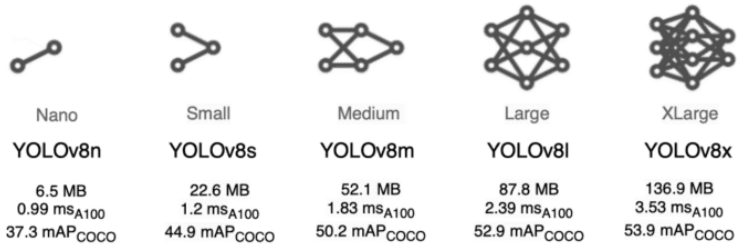


Figure 3. Different model of YOLOv8

The Smart Car Cam system is powered by YOLO (You Only Look Once) [3][4] which is a cutting-edge deep learning model designed for fast and accurate object detection. YOLO has become one of the most popular AI models for real-time object detection because of its impressive speed and efficiency.

In the project, I have collected about 2300 images to label and train the AI. After we trained more than 800 times with YOLOv8m model. The Box loss and the class loss drop to nearly 0.25 and 0.16. The trained model can detect 32 traffic signs.

However, I use the YOLOv8m original model to detect the car and bus. Using the **Monocular Distance Estimation** [10] to calculate the font car distance. We assume the real car width is 2m. The real bus width is 2.5m. The Focal length is 700 to calculate the car object distance.

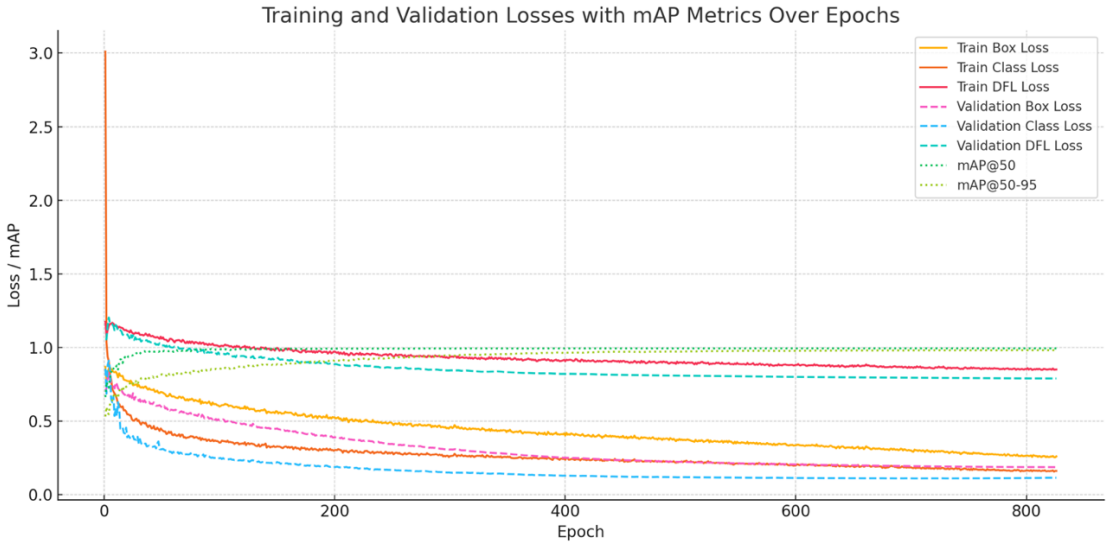


Figure 4. The training result 1

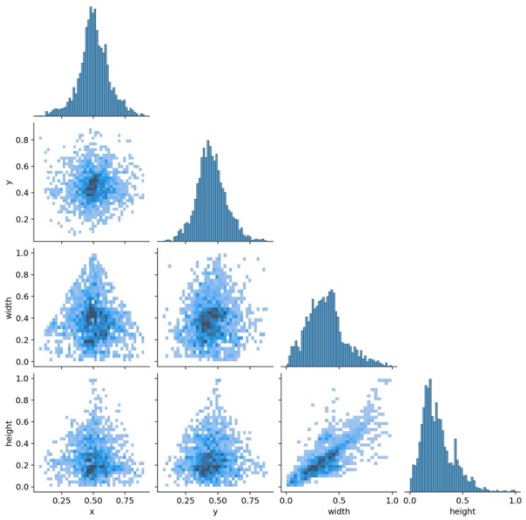


Figure 5. The training result 2

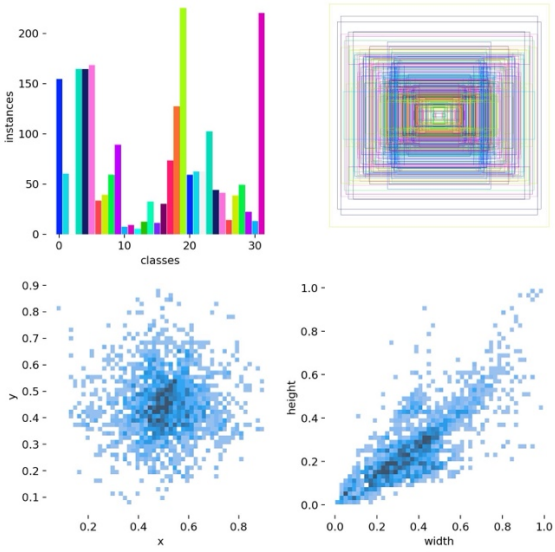


Figure 6. The training result 3

II. OpenCV Python Library

OpenCV (Open Source Computer Vision Library)[5] is a powerful, open-source library designed for computer vision and image processing tasks. It is widely used in various fields such as robotics, surveillance, self-driving cars, medical imaging, and more. The Python interface of OpenCV (cv2) provides tools to handle real-time image processing, making it ideal for projects that involve computer vision.

In the provided code, the OpenCV library (cv2) plays a crucial role in several aspects of image processing, detection, and rendering.

Firstly, I used the OpenCV library to decode the image sent by the client. The client sends images in base64 format. OpenCV is used here to convert this encoded data back into an image format (frame) that can be processed.

```
# Decode the image from the client
        img_data = base64.b64decode(data_image)
        np_arr = np.frombuffer(img_data, dtype=np.uint8)
        frame = cv2.imdecode(np_arr, cv2.IMREAD_COLOR)
```

Then, it will draw bounding boxes and labels on detected objects after detecting objects. Such as the traffic sign, car and bus.

```
cv2.rectangle(frame, (x_min, y_min), (x_max, y_max), color, thickness)
                        label_text = f"{class_name} {distance:.2f} m"
                        cv2.putText(frame, label_text, (x_min, y_min - 10),
```

```
                                    cv2.FONT_HERSHEY_SIMPLEX, 0.9, color, 2)
```

Next, it will be extracting the region of interest (ROI). As I want to use the OpenCV slicing the extract a portion of the frame containing the detected sign. This extracted region (ROI) is then passed to the EasyOCR library for text recognition.

```
roi = frame[y_min:y_max, x_min:x_max]
```

Finaly, we will be encoding the annotated frame back to JPEG. The encoded image is then converted into base64 text to be transmitted over the WebSocket connection.

```
# Encode the frame back to JPEG
        _, buffer = cv2.imencode('.jpg', frame)
        jpg_as_text = base64.b64encode(buffer).decode('utf-8')
```

III. EasyOCR Python Library

EasyOCR[6] is an open-source Optical Character Recognition (OCR) library that is widely used for extracting text from images. It is built using deep learning techniques and supports over 80 languages. EasyOCR is known for its ease of use, flexibility, and impressive accuracy, especially in real-world scenarios like detecting text on street signs, documents, and even images with non-standard fonts.

The primary advantage of EasyOCR over traditional OCR libraries (like Tesseract) is that it uses pre-trained deep learning models, making it highly effective at recognizing text in complex backgrounds, distorted images, and low-light conditions. It is especially useful in applications where text is not perfectly aligned or has various fonts and sizes.

In the my code, EasyOCR is used to recognize text from detected traffic signs.

Firstly, Initialization of the EasyOCR Reader to ‘en’. Then, it will be extracting the region of interest (ROI). After that, it will be using EasyOCR to detect text. Finally, it will be post-processing and matching with expected words.

We use the “expect_word” list to pair the “no-stopping” traffic sign as the “no-stopping” traffic sign is very similar. It can verify the traffic sign change the class name to correctly “no-stopping” traffic sign.

```
# Define expected words for traffic signs
expected_words = {
    "NoParking24h": ["24hrs", "24 hrs", "24", "hrs"],
    "NoParkingEnd": ["End"],
    "NoParkingGreen": ["8", "10", "8-10"],
    "NoParkingRed": ["12", "7"],
    "NoParkingYellow": ["7", "7"],
```

```
    "ExceptHolidays": [ “General” , "Holidays"],
    "ExceptTaxi": ["Except” , "taxi"]
}
```

IV. Flash API Python Library

In my code, two Python libraries are primarily used to create a web application that can handle real-time data processing and communication.
Fristly, the “Flask” is lightweight web framework for building web applications and APIs in Python.
The other one “Flask-SocketIO” is an extension of Flask that adds support for WebSocket communications, enabling real-time, bidirectional communication between the client and server.

V. SpeechSynthsis API

The SpeechSynthesis API [12] is a built-in feature of modern web browsers that allows developers to convert text into spoken words. This API is part of the broader Web Speech API and provides a way to add text-to-speech capabilities to web applications without requiring external libraries or tools. It is supported by most modern browsers, making it a versatile and widely accessible option for enhancing user interaction, particularly in accessibility-focused applications.

In my font-end JavaScript code, the SpeechSynthesis API is used to provide real-time audio alerts to drivers when certain traffic signs are detected. This feature helps keep drivers informed and attentive without requiring them to take their eyes off the road, thereby promoting safer driving.

VI. GPS Geo Location

In my font-end JavaScript code, the **Geolocation API** is a feature of modern web browsers that allows web applications to access the user’s geographic location, which includes details such as latitude, longitude, speed, and accuracy. This information can be used for location-based services, navigation, and other applications that require real-time position tracking.

Firstly, it will be using the watchPosition to Continuously Track Location. It provides the latest position data (position.coords) whenever the user moves, including, the user’s current latitude , the user’s current longitude. The timestamp of when the position was recorded.

Then, it will display the Latitude, Longitude, and Speed on the webpage.

```
// Display latitude and longitude
locationElement.textContent = `緯度: ${latitude.toFixed(4)}, 經度:
${longitude.toFixed(4)}`;
// Handle speed
if (speed !== null && !isNaN(speed)) {
```

```
    // If the device directly provides speed, use it
    const speedKmh = (speed * 3.6).toFixed(2); // Convert m/s to km/h
    speedElement.textContent = `速度: ${speedKmh} km/h`;
}
```

After that, it will be calculating speed manually if not provided by the Device.

```
if (previousPosition && previousTimestamp) {
    // If the device doesn't provide speed, calculate it manually
    const timeDelta = (timestamp - previousTimestamp) / 1000; // seconds
    const distance = getDistanceFromLatLonInKm(
        previousPosition.latitude,
        previousPosition.longitude,
        latitude,
        longitude
    );
    const speedKmh = (distance / timeDelta) * 3600; // km/h
    speedElement.textContent = `速度: ${speedKmh.toFixed(2)} km/h`;
}
```

However, it will reverse Geocoding to get street name. The code uses the **Nominatim API** to perform reverse geocoding, converting the latitude and longitude coordinates into a human-readable street name.

```
// Use Nominatim API for reverse geocoding to get street name
const response = await
fetch(`https://nominatim.openstreetmap.org/reverse?format=jsonv2&lat=${latitu
de}&lon=${longitude}`);
if (!response.ok) {
    throw new Error('Reverse geocoding failed');
}
const data = await response.json();
const address = data.address;
const street = address.road || address.pedestrian || address.cycleway ||
address.street || '未知街道';
locationElement.textContent += ` | 街道: ${street}`;
```

3.2 Hardware

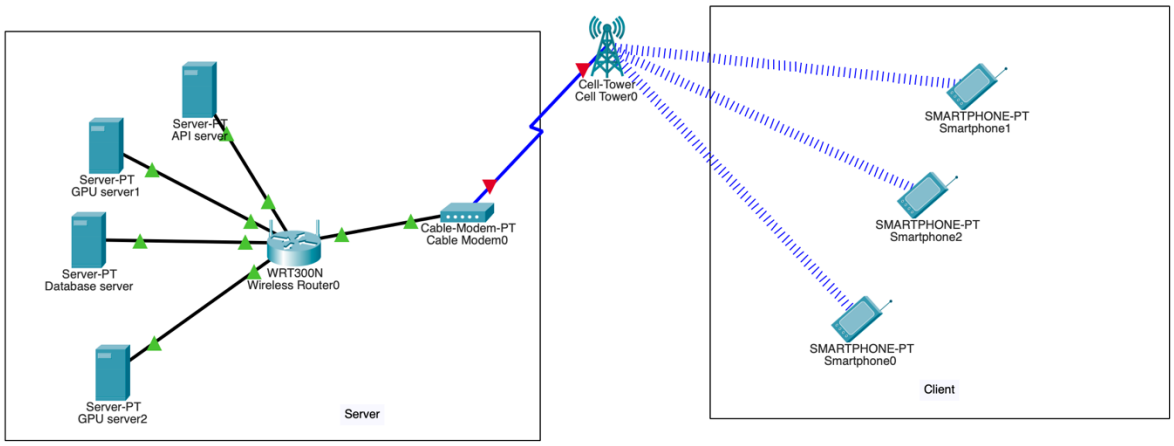


Figure 7. network architecture diagram

This image depicts a network architecture diagram that how the system run. In the server-side (left section), it has the GPU and API and database server to handles general processing tasks. The client-side (right section) will send and receive the real-time data though mobile wireless network.

4. Findings & Results

The Smart Car Cam system can be clearly identifying different traffic sign. It can be detected 32 traffic signs. However, it can detect the bus and car distance using the Monocular Depth Estimation [10] to calculate the distance. When system detected the traffic sign, the system uses an audio alert system to immediately warn drivers. And then, it will add to the right section stark to record traffic signs for review. Furthermore, it provides the current Geo location and the Street name.

The result show below:

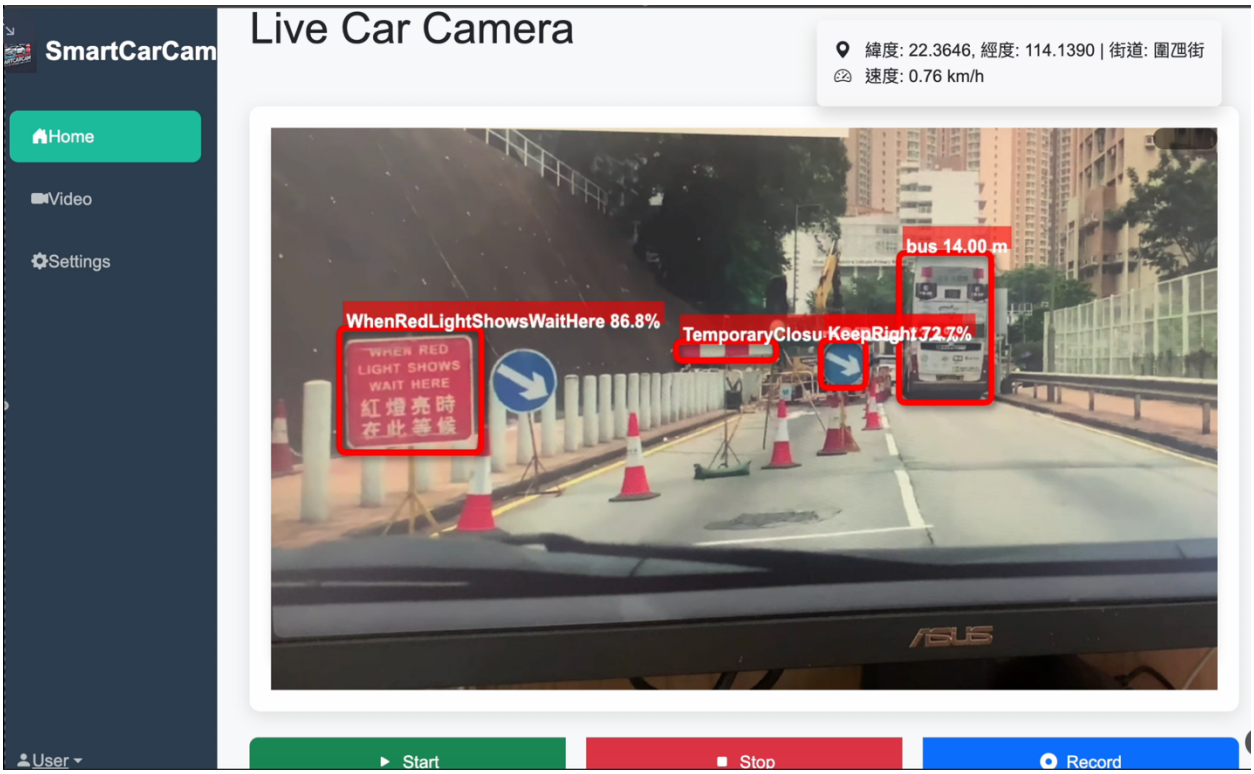


Figure 8. the result of detected a road work and the bus distance

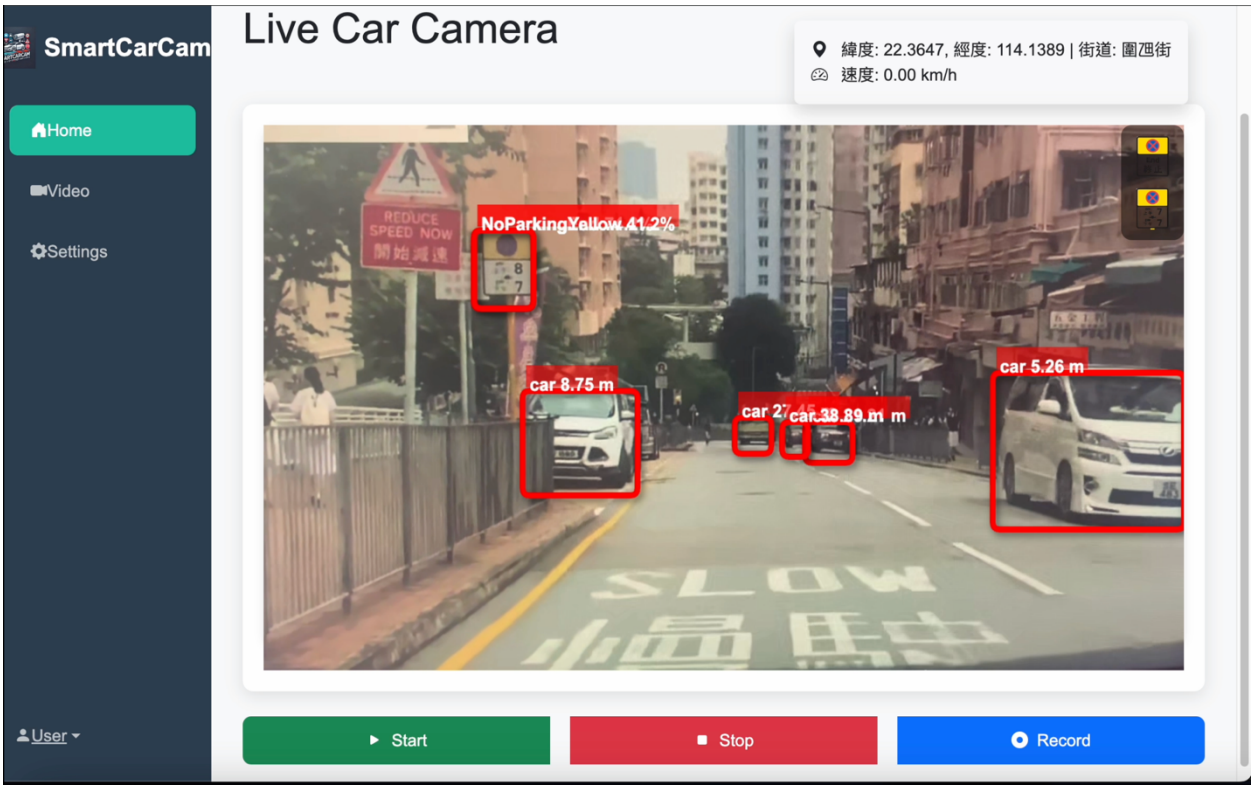


Figure 9. the result of detected a No-Parking traffic sign(Yellow) and the car distance

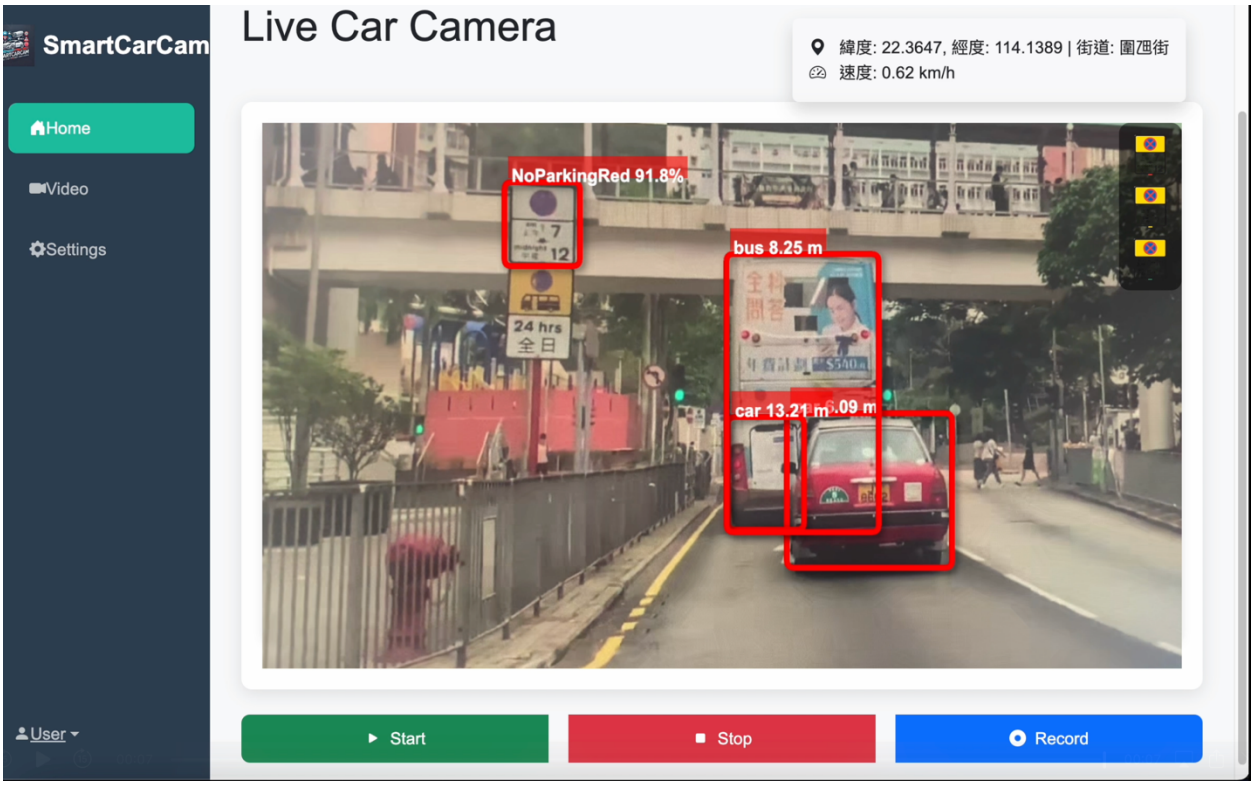


Figure 10. the result of detected a No-Parking traffic sign(Red) and the car & bus distance

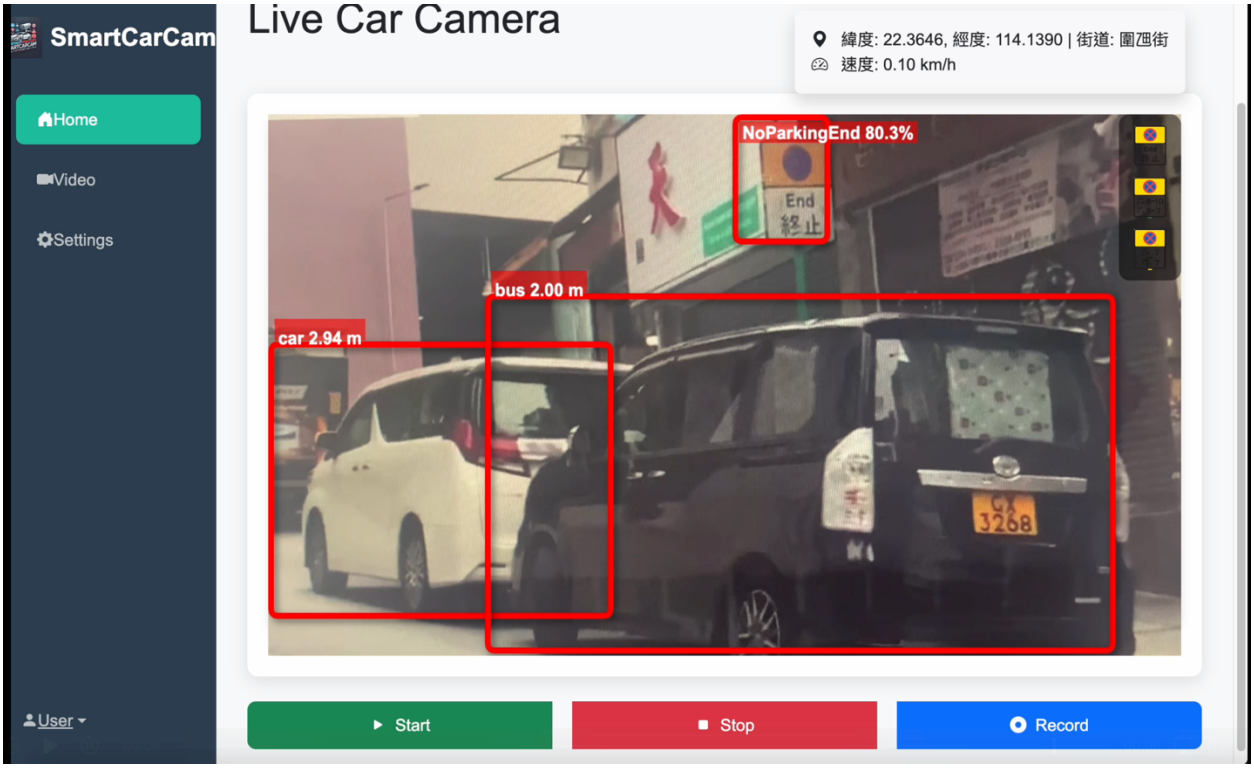


Figure 11. the result of detected a No-Parking traffic sign (End) and the car distance

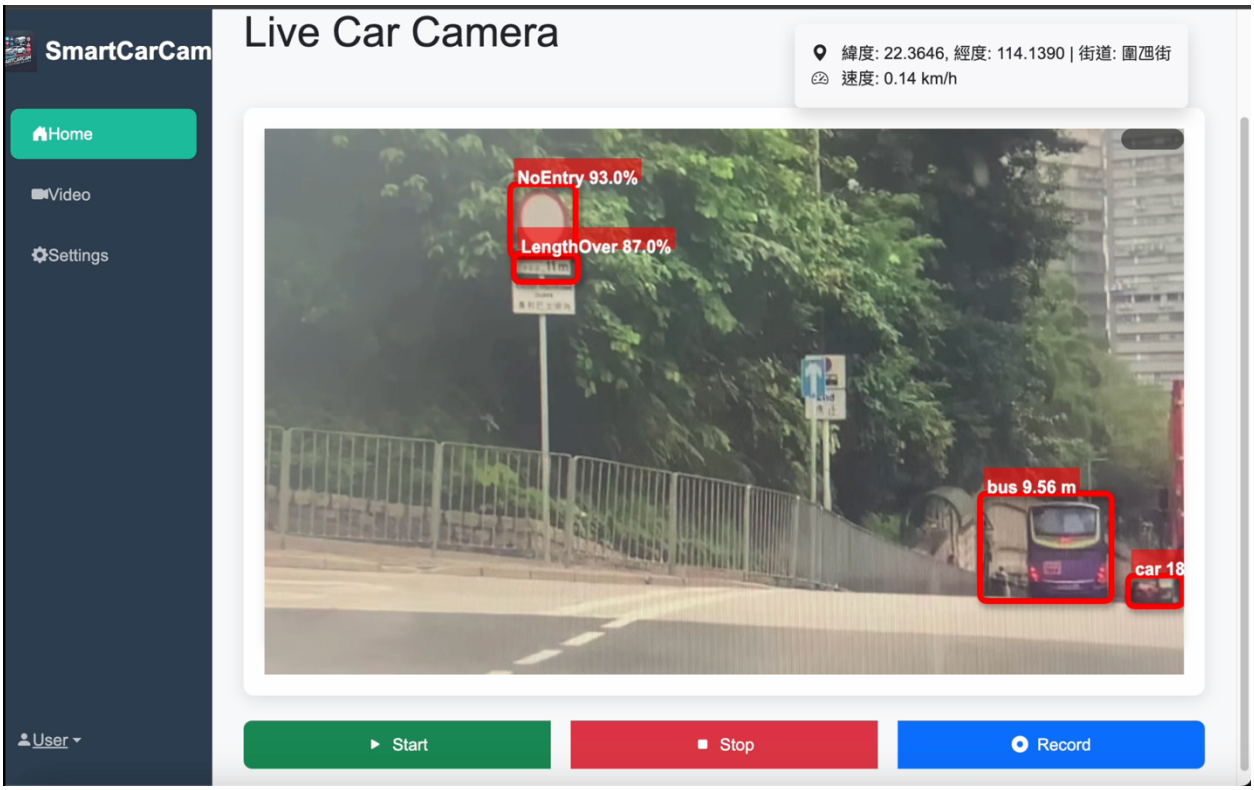


Figure 12. the result of detected a No entry with the length over 11m traffic sign and the car distance

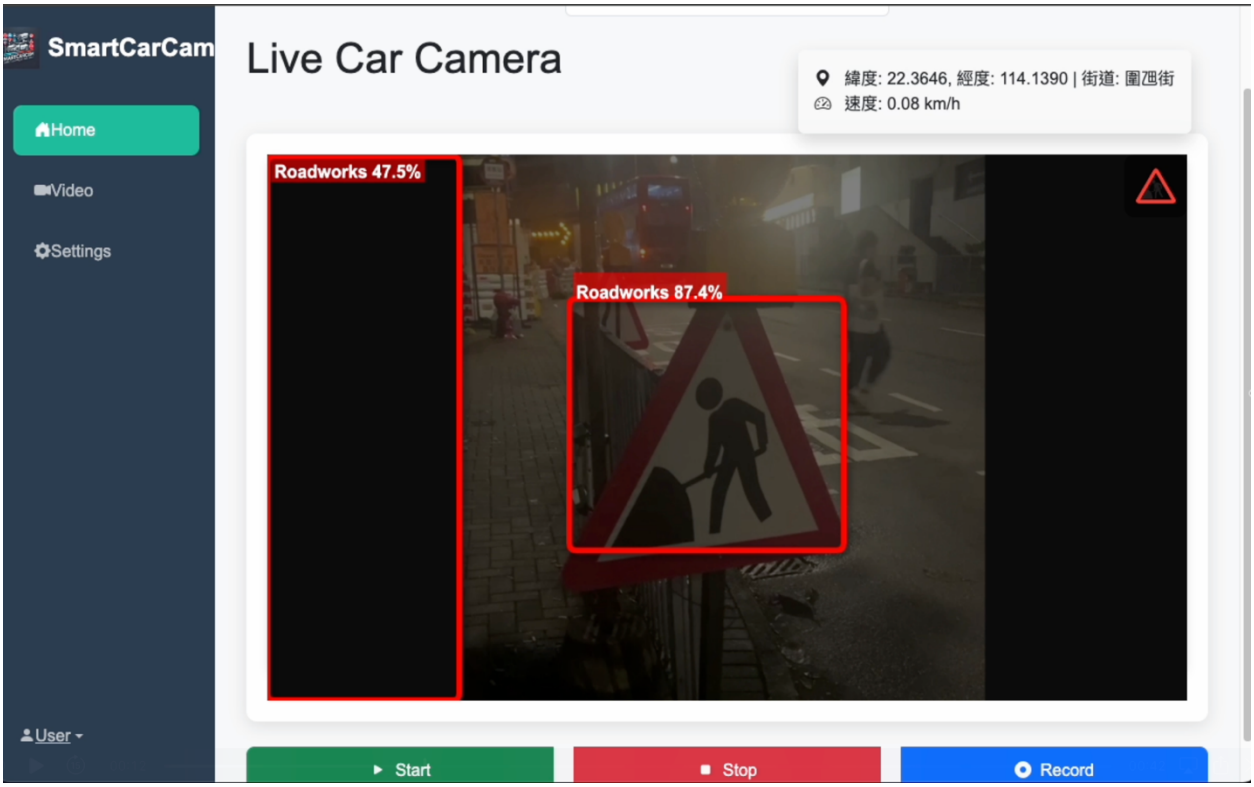


Figure 13. the result of detected a raod work traffic sign in the mid night

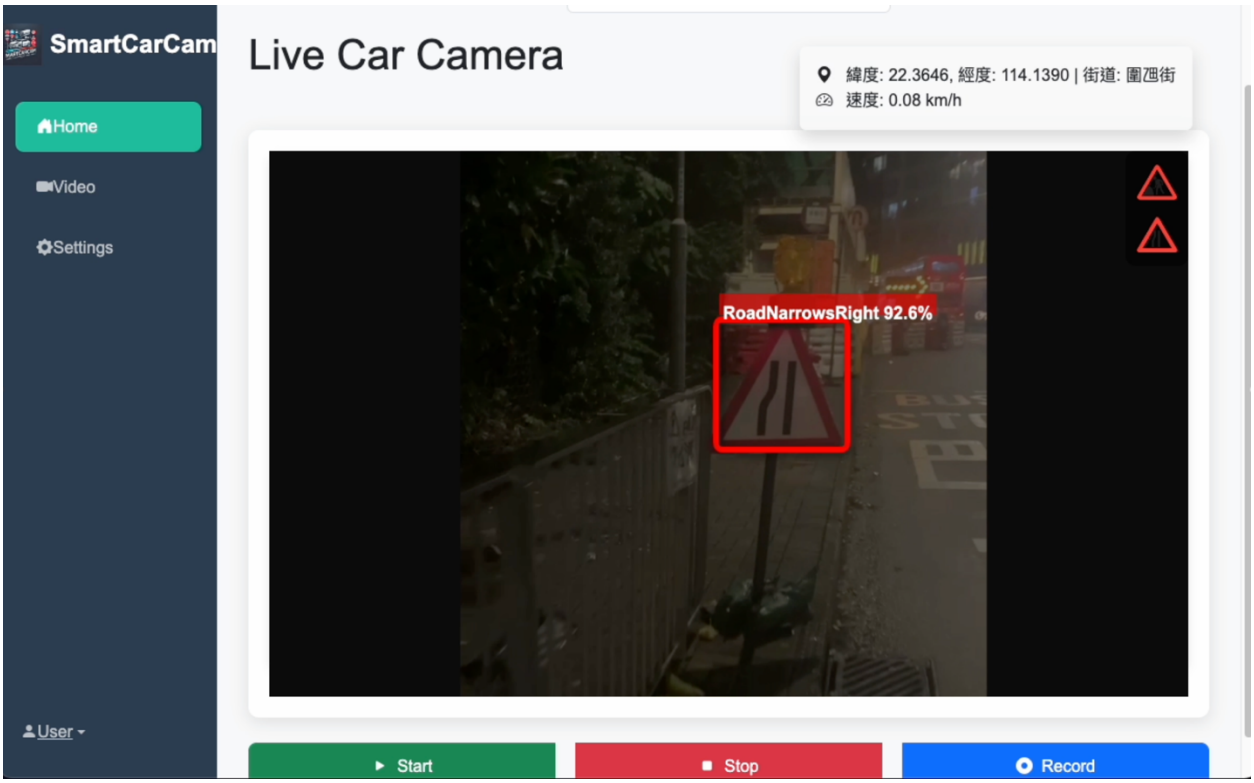


Figure 14. the result of detected a Road Narrows traffic sign in the mid night

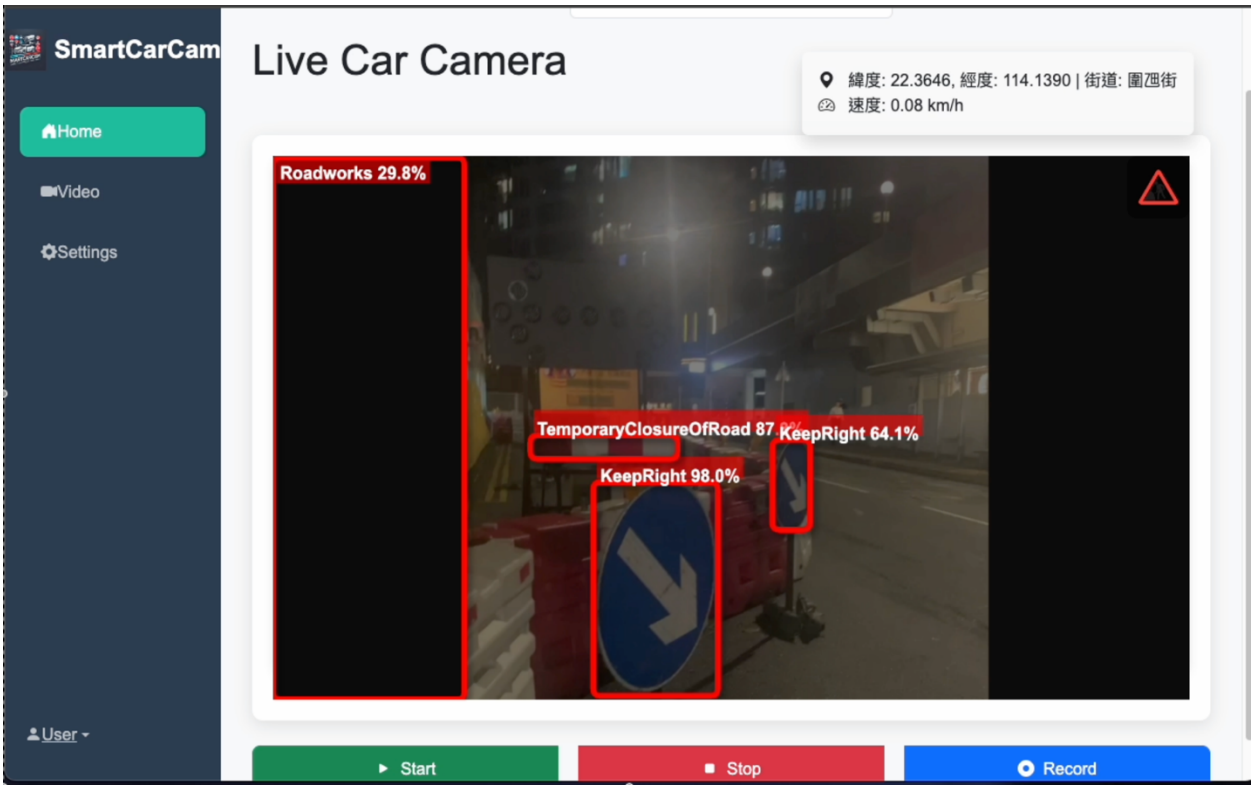


Figure 15. the result of detected many road work traffic signs in the mid night

5. Summary

The Smart Car Cam system helps make Hong Kong's roads safer and reduces illegal parking. It uses technology called YOLOv8 and EasyOCR which is recognize traffic signs and gives quick audio warnings. This helps prevent accidents when drivers miss important signs. The system can be detecting no-parking signs in real time, which helps stop illegal parking, especially in busy areas.

However, it has GPS tracking and can measure distances, which makes drivers more aware and encourages them to drive more safely. The project successfully combines AI and web technologies, showing that it can grow and be used in real-life situations.

In the future, the system could be improved by adding more different traffic signs. Maybe can use the data.gov.hk Traffic Aids Drawings (2nd generation) API [11] to verify the traffic signs more correctly. Overall, the Smart Car Cam is a practical and useful tool for improving road safety and ensuring that traffic rules are followed.

6. References

1. Number of Traffic Accidents with Person Injury (TAPI) by Police Region 2024

https://www.police.gov.hk/ppp_en/09_statistics/tapi_tcpr_2024.html

2. Number of Traffic Enforcement by Quarter in 2024

https://www.police.gov.hk/ppp_en/09_statistics/tes_2024.html

3. Ultralytics YOLO8 code

https://github.com/ultralytics/ultralytics

4. Ultralytics YOLO8 Document

https://docs.ultralytics.com/models/yolov8/

5. OpenCV

https://opencv.org/

6. EasyOCR

https://github.com/JaidedAI/EasyOCR

7. Python Flask API

https://flask.palletsprojects.com/en/stable/api/

8. Pytorch

https://pytorch.org/

9. 吐露港公路大工程四線收剩一線　疑切線致三車相撞鐵騎士倒地傷

https://www.hk01.com/article/1056931?utm_source=01articlecopy&utm_medium=referral

10. Monocular Depth Estimation

https://paperswithcode.com/task/monocular-depth-estimation

https://iopscience.iop.org/article/10.1088/1742-6596/1168/3/032040/pdf

11. Traffic Aids Drawings (2nd generation) API

https://data.gov.hk/tc-data/dataset/hk-td-tis_16-traffic-aids-drawings-v2

12. SpeechSynthesis API

https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis