A Real-Time Intelligent Traffic Alert System Based on Multimodal Sensing and Edge Computing

Yu-Ping Liao
Department of Electrical Engineering
Chung Yuan Christian University
Taoyuan, Taiwan
lyp@cycu.edu.tw

Chien-Yu Chen
Department of Electrical Engineering
Chung Yuan Christian University
Taoyuan, Taiwan
vivian920903036615@gmail.com

Priscilla Yang
Department of Electrical Engineering
Chung Yuan Christian University
Taoyuan, Taiwan
xunzhenyang@gmail.com

Yiqi Cai
Department of Electrical Engineering
Chung Yuan Christian University
Taoyuan, Taiwan
yiqicai7@gmail.com

***Abstract*—We propose an intelligent traffic alert system that integrates multimodal sensing with edge computing to improve safety at intersections. The system incorporates a CNN-based siren recognition and direction detection module, an image preprocessing module that handles scene classification and enhancement, and a YOLO-based module for detecting pedestrians at crosswalks. The system is designed to operate in various urban environments, including low visibility, environmental disruptions, and unpredictable pedestrian behavior. The image enhancement module improves visibility in low-light and rainy conditions, achieving up to +24 points improvement in quality score while maintaining stable detection performance. Additionally, tests on image datasets verify that pedestrian detection at crosswalks is reliable, with an accuracy of 91%. Furthermore, the sound recognition accuracy of 92% shows its capability to identify emergency vehicle sirens. Implemented on a Raspberry Pi 5 with an LCD interface for driver alerts, the proposed system demonstrates that lightweight models and multimodal mixture can be effectively deployed on edge devices to provide instant and reliable assistance for smart transportation applications.**

Keywords—traffic safety, real-time detection, multimodal sensing, sound recognition, image enhancement, pedestrian detection, edge computing, smart transportation.

I. INTRODUCTION

According to statistics from the National Police Department, there has been a significant increase in incidents where cars fail to yield to pedestrians in Taiwan, reaching 139,086 in 2023 alone, an increase of 1.7 times compared to the previous year, and leading to a continuous increase in pedestrian deaths and injuries, as shown in Fig.1 [1]. It reflects the obvious shortcomings of the current transportation system in ensuring pedestrian safety in real time. Statistics from the Fire Department of Taoyuan City, Taiwan, show that 47.62% of ambulance accidents involve collisions at intersections [2]. This is primarily because drivers find it difficult to accurately determine the direction of the siren.

The above phenomenon highlights two core problems: first, the static nature of traffic signal systems today makes it difficult to respond to dynamic road conditions, such as pedestrian crossings, in real time. Second, traditional single-audible warnings lack the ability to provide precise directionality in complex traffic environments. To address these issues, we propose an intelligent traffic warning system based on sound and visual recognition to enhance pedestrian protection and accident response, corresponding to the United Nations SDG 3 and SDG 11.

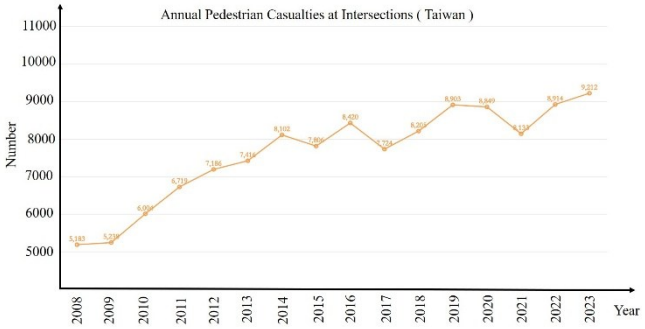

Fig. 1. Annual pedestrian casualty at intersections.

II. METHODOLOGY

A. System Design

1) Hardware Architecture

The system hardware architecture is shown in Fig. 2 and there are three main components: image and sound input, edge device computing, and LCD output display. First of all, the camera continuously captures real-time video streaming. The microphone array processes ambient audio and can also determine the direction of the sound source. Next, these images and sound data are fed into pre-trained YOLO and sound recognition models for detection. Finally, the LCD display dynamically updates based on the warning signals received from the edge computing platform.

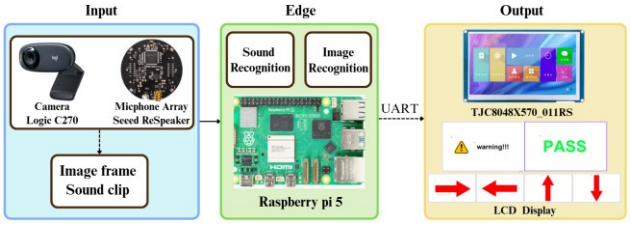

Fig. 2. System architecture.

2) Processing Flow

The system processing flow is illustrated in Fig. 3. It uses multithreading with two threads in total. One thread handles sound recognition, while the other handles image recognition. First, let us focus on the sound recognition thread. The input from the microphone array is processed by the Sound

Recognition and Direction Module, which detects the siren sound, determines its direction, and generates high-priority UART messages. Next is the image recognition thread. For this thread, the process begins with a webcam video feed input. The input image frames are first detected and classified into four scenes by the Image Preprocessing module. Then the images are optimized according to the classification and passed to the Pedestrian Crosswalk Detection module, which detects whether pedestrians are on the crosswalk and generates low-priority UART messages. Both sets of UART messages will be sent to the UART Manager Module, which processes them based on priority. Finally, the final UART message is sent to the LCD display, which shows the corresponding scene.

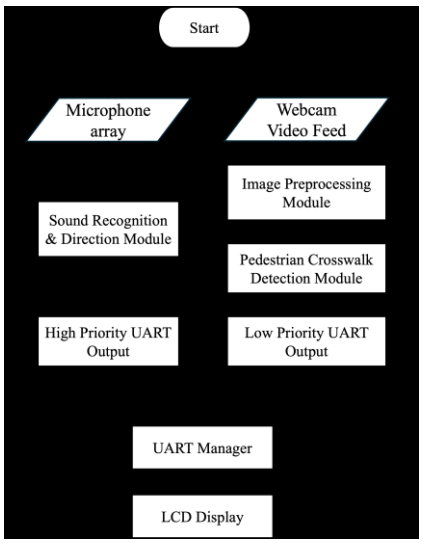

Fig. 3. A real-time intelligent traffic alert system processing flow.

B. Sound Recognition

As shown in Fig. 4, the diagram is the flow chart for sound recognition. When the microphone receives an external sound, it gains the sound direction. If a Siren sound is detected, an icon indicating the specific direction will be displayed on the screen.

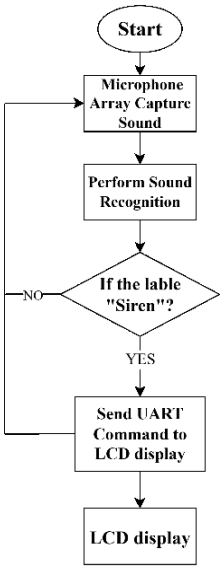

Fig. 4. The flowchart of sound recognition.

1) Sound Model Architecture

Sound recognition adopts a simple CNN architecture, and the details of this architecture are as shown in Table I. The model takes the Mel-Spectrogram as input, and the tensor size being (C, M, T), where C is monaural sound, M is the number of Mel bands, and T is the number of time steps. The network backbone consists of four repeated convolutional blocks (Blocks 1–4) that are connected to each other. Each block contains two layers of 3×3 convolutions (stride=1, padding=1) + Batch Normalization + ReLU, and uses 2×2 maximum pooling to reduce the time and frequency dimensions by half. Since convolutions use the parameter padding=1, each block can maintain the same size before pooling.

TABLE I. SOUND MODEL ARCHITECTURE

Convolution block	Sound Model Architecture			
	Convolutional layer	*Output channel*	*Kernal size*	*Pooling layer*
Block 1	Conv2d×2	32	3×3	MaxPool2d (2×2)
Block 2		64		
Block 3		128		
Block 4		256		

After convolution, the feature map is flattened and three fully connected layers are connected. We also use a dropout layer to reduce the risk of co-adaptation and overfitting. The final linear layer outputs logits for each class. Because cross-entropy loss is used during training, the softmax function is implicitly handled by the loss function. During model construction, the model performs a forward pass with a dummy tensor that matches the actual preprocessing. This dynamically calculates the flatten_dimension to automatically accommodate the time-step variations caused by different audio lengths or time-frequency parameters, thus avoiding inconsistencies in manually calculated dimensions.

We developed our model architecture to achieve an ideal balance of parameter count, inference speed, and recognition performance. Key design decisions include: (i) a small 3×3 convolutional stack that enables a deeper receptive field with fewer parameters; (ii) using Batch Normalization during training to stabilize gradients and improve robustness to class imbalance; (iii) applying 2x2 pooling operations that lead to a 16-fold down sampling of the time-frequency dimensions, and balancing the modulation patterns required for recognition with computational efficiency. (iv) two fully connected layers with dropout, which help suppress overfitting while preserving high-level semantics, allow for better generalization of background noise in real-world environments.

2) Training process and convergence analysis

To evaluate the model's performance during training, we use the MATLAB module to record and plot the trends of training epochs with respect to Training Loss, Validation Loss, Training Accuracy, and Validation Accuracy. The loss curve is as shown in Fig. 5 (a), the training loss (red line) declined rapidly in the early stages of training and stabilized after about the 25th epoch, eventually converging to a lower value of around 0.1, which means that the model achieved a good fit on the training dataset. However, validation loss (blue line) fluctuated significantly in the early stages, which may be related to the instability of the learning rate at the initial stage

of training, but after the 25th epoch, the validation loss gradually converged and stabilized at around 0.3. The training loss remains consistently lower than the validation loss, indicating a slight overfitting phenomenon. The accuracy curve is as shown in Fig. 5 (b), training accuracy (red line) steadily improved during training and achieved a stable performance of approximately 92% after 30 epochs. The validation accuracy (blue line) also fluctuated greatly in the early stages, but in the middle and later stages of training, its trend was highly consistent with the training accuracy. It eventually stabilized between 92% and 93%, indicating that the model has good generalization ability.

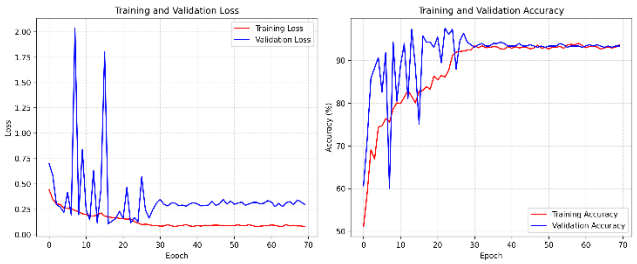

Fig. 5. The plot of loss and accuracy of the sound model.

C. Image Preprocessing

An image scene detection and optimization module is designed for preprocessing to improve the performance of the pedestrian detection model in different environments. It supports both single-image input (for annotation/analysis) and real-time video capture. The process includes four steps: scene classification, non-rainy judgment, enhancement, and quality assessment.

1) Scene Classification

We design a deep learning model for the "rainy day" and "non-rainy day" scene classification problems. Since rainy days have the most significant impact on detection, identification needs to be prioritized. To ensure both training efficiency and real-time deployment, we need a fast, lightweight, and accurate model for this task. While large networks such as VGG, ResNet, and EfficientNet achieve high accuracy, their size and latency are impractical. On the other hand, MobileNet is specifically designed for mobile and embedded vision. Hence, it was chosen as the backbone for transfer learning. As shown in Fig. 6, We selected MobileNetV3-Large due to its superior balance of accuracy (75.2%) and latency (51 ms), outperforming V2 (72.0%, 64 ms) [3].

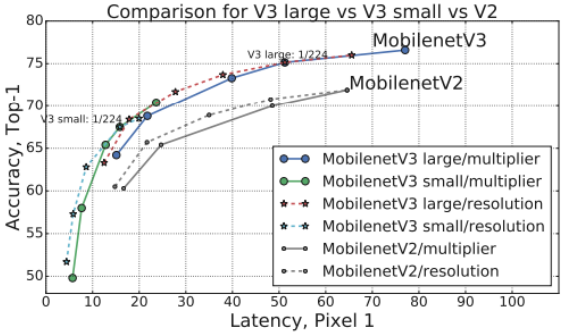

Fig. 6. Comparison of MobileNetV3 and V2 in terms of accuracy and latency [3].

The dataset consists of surveillance and public images[4][5][6], annotated into rainy/non-rainy, with a 7:2:1 split(training/validation/testing). Training stabilizes as shown in Fig. 7, where both loss keeps decreasing and accuracy levels off.

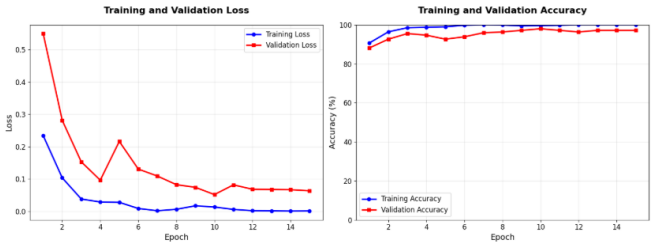

Fig. 7. The plot of loss and accuracy.

2) Non-rainy scene judgment

If an image is classified as "non-rainy" and has a confidence level above 0.95, the system further relies on traditional image characteristics (brightness, contrast, and edge density) for scene segmentation.

Except for rainy, the following are the scenes classified:

- normal: Brightness, contrast, and edge density are all within the normal range.
- night: The average brightness, the contrast, and the edge density are all low.
- overexposed: Excessive highlight pixel ratio and imbalance in dynamic range.

3) Image Enhancement

According to the scene judgment results above, the system applies corresponding optimization strategies:

- Night: Gamma correction or CLAHE.
- Overexposed: Highlight suppression and dynamic range adjustment.
- Rainy: Simple brightness/contrast compensation (no rain-removal).
- Normal: No processing.

4) Quality & Stability Assessment

After image optimization, the system introduces a **Quality Score(Q)** to quantify enhancement effects, combining brightness, contrast, edge density, entropy, dynamic range, and overexposure ratio, normalized to [0–100]. The formula is as follows:

$$Q = 100 \times [w_b \cdot f_b(B) + w_c \cdot f_c(C) + w_e \cdot f_e(E) + w_c \cdot f_c(C) + w_h \cdot f_h(H) + w_d \cdot f_d(D) - w_s \cdot f_s(S)] \quad (1)$$

Where B = Average image brightness (0–255)/ C = Contrast ratio/ E = Edge density/ H = Entropy/ S = Saturation ratio / D = Dynamic range. All the feature values above are normalized to the range of 0–1, and the score difference (ΔQ) before and after processing provides direct evidence of improvement.

D. Pedestrian Crosswalk Detection Module

The pedestrian crosswalk detection module is designed to warn drivers and improve the safety of pedestrians when crossing the road. To implement it in real-world applications on resource-limited edge devices like the Raspberry Pi 5, processing speed and accuracy are the main performance values we prioritize. To meet these objectives, we've

employed experiments to validate and optimize our selection across critical metrics. The image test set used in the following tests, including the YOLO Module Selection test, IoA Threshold Selection test, and Confidence Threshold Selection test, is a dataset we collected that consists of 66 manually annotated images. The hardware used for the tests is a Raspberry Pi 5 with 4GB of RAM.

1) Module Process Flow

The process flow of the module, shown in Fig. 8, is straightforward; it consists of two phases: an initial setup phase, which involves manually annotating the crosswalk since the camera has a fixed view of the crosswalk. Once the module knows the boundary of the crosswalk, it is ready to work with the following steps.

- It takes the video feed as input, uses YOLOv11n (ONNX format) to detect and mark the pedestrian bounding boxes.
- Calculate the intersection between the crosswalk area and the pedestrian bounding box.
- Calculate the IoA (Intersection over pedestrian bounding box Area) value to determine if the pedestrian is within the crosswalk.
- If the module determines there are pedestrians in the crosswalk, it sends out a low-priority UART message.

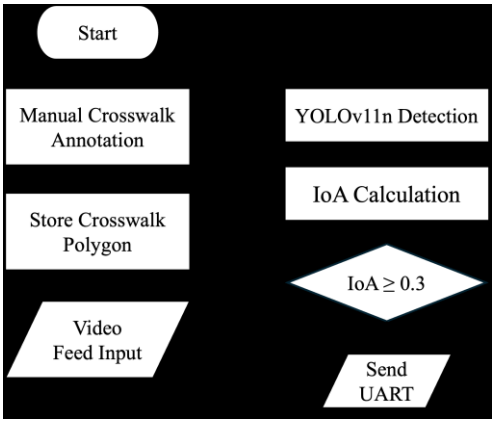

Fig. 8. Pedestrian crosswalk detection module process flow.

2) YOLO Model Selection

Instead of training our own object detection model, we use a pre-trained YOLOv11 model from Ultralytics to focus on the implementation of the model and the module algorithm. For the YOLO model selection, we conducted two tests. First, we compared the processing speed and accuracy of three YOLOv11 variants (n, s, m). As shown in Table II, YOLOv11n has a slightly lower recall (0.89) and mAP (0.88) but the highest FPS (3.36). Although larger models provide better accuracy, YOLOv11n was selected because the system requires near real-time processing. Additionally, we evaluated two YOLOv11n deployment frameworks, PyTorch and ONNX. The results shown in Table III indicate that ONNX outperformed PyTorch on all metrics, offering higher precision, recall, mAP, and FPS. Therefore, we chose to implement the YOLOv11n model using the ONNX format in the module.

TABLE II. YOLOv11 VARIANT PERFORMANCE COMPARISON

Model	Performance Metrics			
	Avg FPS	***mAP***	***Precision***	***Recall***
yolo11n.pt	3.36	0.88	0.94	0.89
yolo11s.pt	1.36	0.91	0.92	0.92
yolo11m.pt	0.41	0.93	0.92	0.94

TABLE III. DEPLOYMENT FRAMEWORK PERFORMANCE COMPARISON

Framework	Performance Metrics			
	Avg FPS	***mAP***	***Precision***	***Recall***
PyTprch	3.62	0.85	0.92	0.86
ONNX	5.70	0.87	0.94	0.97

3) IoA Threshold Selection

We tested IoA thresholds (0.05 to 1) with 0.05 increments to determine the suitable value for our algorithm. The YOLO model we tested on was YOLOv11n (ONNX format). The results are shown in Fig. 9. When IoA is at 0.3, the model has the highest overall accuracy and F1 score, with excellent Recall and Precision. Aside from that, it has the lowest MEA. Therefore, 0.3 was selected as the threshold value for the IoA.

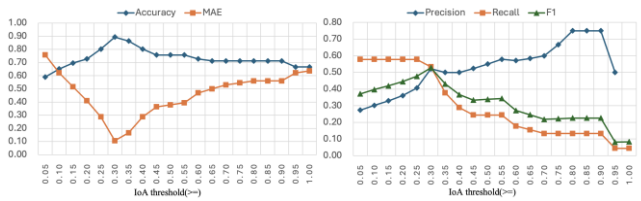

Fig. 9. IoA threshold sensitivity analysis.

4) Confidence Threshold Selection

The confidence threshold sets the minimum confidence for detections. Only confidence values higher than the confidence threshold will be considered as valid predictions. In the confidence threshold test, we evaluated the threshold values from 0.05 to 0.95. As shown in Fig. 10, the highest overall accuracy with excellent precision is achieved at 0.6. Therefore, we use 0.6 as the confidence threshold for our module.

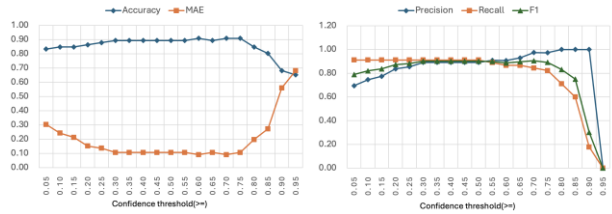

Fig. 10. Confidence threshold sensitivity analysis.

E. LCD displays

A TAOJINGCHI's display TJC8048X570_011RS display device is used for the output display of the results. The

display interface is designed and developed using the USART HMI software tool. We created several graphic pages based on the functional requirements of the system, compiled them into .tft files, exported them to the LCD display in the end.

This device mainly implements the following two functions:

1) Pedestrian Alert Display

When the Pedestrian crosswalk detection module detects a pedestrian in the zebra crossing area, the Raspberry Pi sends specific commands to the display. At the same time, the screen quickly switches from the regular traffic situation screen (Fig. 11 (b)) to the pedestrian warning screen (Fig. 11 (a)), alerting surrounding vehicles in real time.

(a) (b)

Fig. 11. Warning and safe display.

2) Emergency Vehicle Direction Indication

When the sound recognition detects the siren sound of an emergency vehicle (e.g., police car, fire truck), it will determine its source direction. Subsequently, the display will immediately update the screen to show the direction of the official vehicle with intuitive icons as shown in Fig. 12, assisting drivers to yield independently, thereby improving road traffic efficiency and safety.

Fig. 12. Direction display.

III. EXPERIMENTS

A. Sound recognition

1) Experiment Objective

The experiment is divided into two parts: the first part tests sound recognition and signal transmission accuracy, and the second part is to verify the responsiveness and page-switching accuracy on the LCD display.

2) Experiment setup

Hardware device: Raspberry Pi 5, Seeed ReSpeaker, and TAOJINGCHI's display TJC8048X570_011RS.

Dataset: A self-constructed siren and background sound dataset, along with the public open-source dataset (UrbanSound8K), which covers a variety of emergency vehicle siren sounds (such as ambulances, fire trucks, police cars) and various common household sounds.

3) Sound Recognition Performance Evaluation

To quantify the accuracy of the sound recognition module, we employ the following steps and evaluation metrics.

We will randomly input the test audio into the trained model. The recognition results output by the model are compared with the real labels of the audio, and the accuracy, precision, recall, and F1-score of the model are calculated in different sound categories.

When testing the model, it will generate a confusion matrix that visualizes the model classification across categories. This helps us analyze which sounds the model most often misjudges as other categories and optimize the model accordingly. To evaluate the accuracy of sound source positioning, siren sounds will be played from multiple angles near the microphone array and record the direction page displayed on the LCD.

4) Experiment Procedure

The experimental process is to play a set of pre-recorded test scenario audio, captured by the microphone, and observe the real-time reaction of the system on the LCD display and the page output results.

B. Image enhancement

1) Experiment Objective

The objective is divided into two parts:

- **Scene Classification**: Test the system's ability to distinguish four scene types (normal, night, rainy, overexposed).
- **Enhancement Impact**: Verify whether the enhancement module improves image quality and maintains detection stability, with emphasis on night scenes.

2) Experiment Setup

The experiments were conducted on both surveillance and non-surveillance images [7]-[10], tested on a Raspberry Pi platform with the proposed system (MobileNet backbone for scene classification and YOLO-based pedestrian detection). Condition-specific enhancement (e.g., gamma correction) was applied before detection.

3) Dataset and Ground Truth

A total of 40 non-roadway images (10 per scene: normal, night, rainy, overexposed) were tested for classification. Truth labels were manually assigned, and representative images were processed in both original and enhanced forms for enhancement assessment, with quality scores (Q) and persons detected in the region of interest (ROI) recorded. ROI counts reflected relative consistency rather than absolute accuracy.

4) Experiment Procedure and Metrics

- **Scene Classification**: Each image was classified and compared with truth. After this, we summarized the results with confusion matrix and per-class results.
- **Enhancement Impact**: The images were then processed in original/enhanced form. Metrics included quality score change (ΔQ) and ROI person counts (ΔPersons), and FPS was monitored but not a primary metric.

C. Pedestrian Crosswalk Detection Module Test

1) Experiment Objective

This experiment assesses the final integrated pedestrian crosswalk detection module. The goal is to quantify the accuracy and performance under the final setup.

2) Experiment setup

- Hardware: Raspberry Pi 5 (4GB RAM)
- YOLO Model: yolo11n.onnx (Ultralytics)
- Input: Camera frames resized to 640×640 pixels

- Parameters: Confidence threshold (0.6), IoA threshold (0.3).

3) Dataset and Ground Truth

The test dataset consists of 66 manually annotated images that cover a wide range of conditions, including different lighting (daylight, darkness, artificial illumination), pedestrian quantity (single, multiple, and crowds), and camera angles. For the ground truth annotation, each image is manually annotated and includes verified pedestrian bounding boxes, crosswalk areas, and the number of pedestrians on the crosswalk.

4) Experiment Procedure and Metrics

The experiment follows these four steps. First, images are processed through YOLOv11n (conf $\geqslant$ 0.60) to obtain pedestrian bounding boxes. Next, the algorithm determines whether pedestrians are on or off the crosswalk using IoA $\geqslant$ 0.30. Then, the predictions are matched with ground truth using greedy IoU matching (IoU $\geqslant$ 0.7). The program will calculate TP/FP/FN, exact-count accuracy, average FPS, and MAE.

IV. Results and Discussion

A. Sound recognition

To comprehensively evaluate the final performance of the sound model, we calculated the precision, recall, and F1-score for each class on separate test datasets. As shown in Fig. 13, the model performed well in both the "siren" and "background noise" categories.

Siren: The accuracy, recall, and F1-score in this category are around 99%, 92%, and 96%, respectively. This means that most of the samples predicted by the model as siren sounds are correct (high accuracy); And 92% of all actual siren sound samples were successfully recognized by the model (high recall).

Background Noise: Accuracy, recall, and F1-score are around 97%, 99%, and 98% in this category, respectively. This shows that the model also excels in recognizing background noise, rarely incorrectly identifying it as a different category.

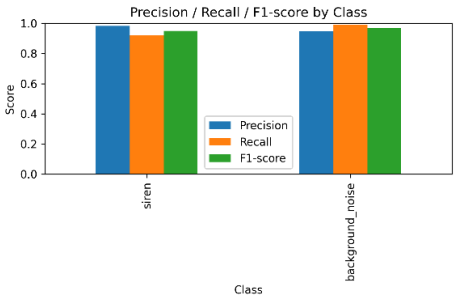

Fig. 13. Model precision, recall, and F1 score performance in “siren” and “background noise”.

From the confusion matrix (Fig. 14), the overall recognition performance of the model is excellent. Out of a total sample of 310, the model had only 12 misclassifications (10 false positives + 2 false negatives). The accuracy of the model reaches (115+183) / (115 + 10 + 2 + 183) = 0.961 (96.1%).

Moreover, the model's false siren rate for siren sounds is extremely low (only 2 out of 125, approximately 1.6%), which shows that it performs well in critical siren sound recognition and significantly reduces the risk of missing important events.

Although there are a small number of false detections, the model is still able to have excellent detection capability compared to its overall performance.

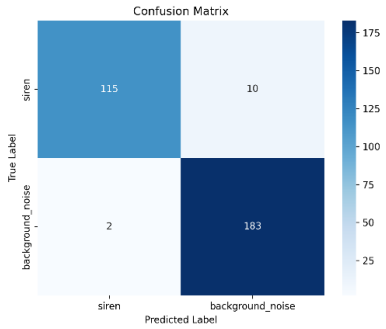

Fig. 14. Confusion matrix of sound recognition.

In order to visually demonstrate the real-time operation behavior of the intelligent sound recognition and direction indication system developed in this study, Fig. 15 shows a schematic diagram of the LCD interface display and the ReSpeaker microphone array under different sound inputs.

Fig. 15(A) and (B) show the system in silent and background noise environments, respectively. At this point, the LCD display will display "PASS" word, indicating that the system is in a safe and alarm-free state.

Fig. 15(C) to (F) show how the system determines the display page based on the direction of the sound source in the case of an alarm sound. When the alarm sound is successfully recognized, the LCD screen switches to the corresponding page based on the Direction of Arrival (DOA) that has been determined.

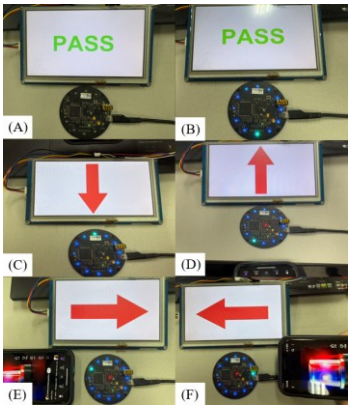

Fig. 15. The results of LCD display.

TABLE IV. Real-Time Rrcognition and Display Page Rrcords

Figure 15	CNN Based Sound Detection Model Performance				
	DOA Angle	*DOA Quadrant*	*Page*	*Detected Label*	*True Label*
(A)	235	N/A	1	Background Noise	Background Noise
(B)	78	N/A	1	Background Noise	Background Noise
(C)	346	Front	5	Siren	Siren
(D)	195	Back	4	Siren	Siren
(E)	293	Left	2	Siren	Siren
(F)	87	Right	3	Siren	Siren

According to Table IV, we can draw the following conclusions:

1) Idle and background noise handling

When the sound detected in the environment is background noise, the model correctly recognizes it as "Background noise," as shown in the first and third records. At this point, the LCD remains on the default page 1 (Pass), indicating a non-alert state, which aligns with design expectations. Although the ReSpeaker microphone continues to detect DOA angles (e.g., 235 degrees and 78 degrees), page 1 does not require specific directionality; therefore, the DOA Quadrant is shown as N/A.

2) Siren detection and precise direction indication

When receiving siren sound, the system not only accurately classifies it but also provides a detailed direction indication based on the DOA angle and switches to the corresponding LCD page.

3) Consistency between detection and true label

The "Detected Label" and "True Label" of the model are highly consistent across all recorded samples, indicating improved accuracy in model judgments and accurate sound classification in real-world environments.

B. Image processing

1) Scene Classification

We tested 40 non-roadway images (10 per scene: normal, night, rainy, overexposed). As shown in Fig. 16, normal and rainy scenes achieved 100% accuracy, while night (80%) and overexposed (70%) were more challenging, often confusing with rainy. Although only 10 samples per class are reported here, this subset is sufficient to demonstrate overall performance; additional tests were conducted but omitted for brevity.

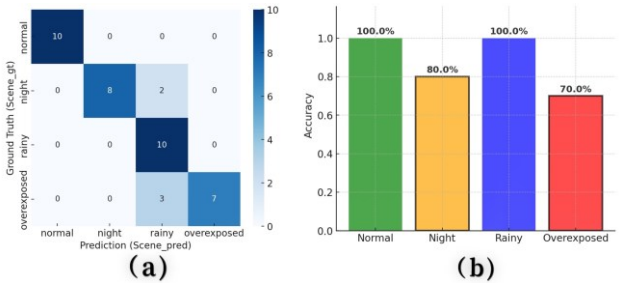

Fig. 16. (a)Confusion matrix of scene classification. (b) Per-class accuracy.

2) Image enhancement

As shown in Table V, night and rainy scenes showed clear quality improvement (ΔQ +7–24) while maintaining detection stability. Normal scenes remained unchanged, avoiding over-processing. Overexposed cases were inconsistent, included only for completeness. Representative examples in Fig. 17 and Fig. 18 further illustrate the night visibility improved.

TABLE V. EFFECTS OF IMAGE ENHANCEMENT

Actual Scene	Quality Score (Q) and ROI Detection Per Scene		
	Detection Scene	*ΔQuality*	*ΔROI Persons*
1. night	night	23.8	0
2. night	night	5.8	-1
3. night	rainy	10.8	1
4. night	night	12.9	-1
5. night	rainy	11.1	0
6. normal	normal	0	0
7. rainy	rainy	22.5	0
8. overexposed	Over.	-27.4	-3

Overall, the results confirm that the enhancement strategy is most effective in night scenarios, improving visibility without harming detection, and leaving room for future refinement.

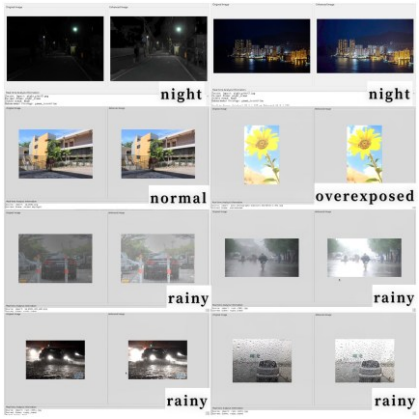

Fig. 17. Visual examples of original vs. enhanced images in different scene.

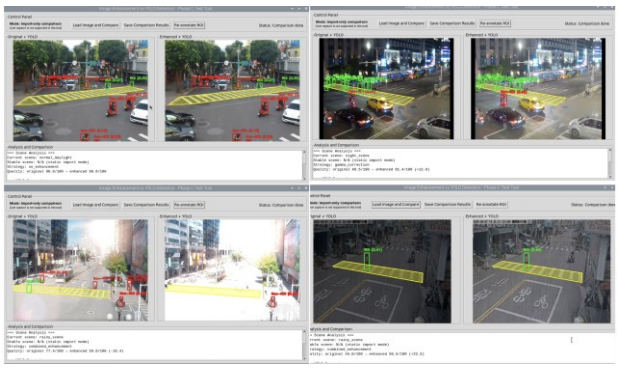

Fig. 18. Representative intersection surveillance cases with YOLO detection before and after enhancement.

C. Pedestrian Crosswalk Detection Model Experiment

The results in Table VI show that the model offers great accuracy (0.91), precision (0.91), recall (0.87), F1 score (0.89), and MAE (0.09). Meaning that the detection is highly reliable with minimal false warnings. Missed detection mainly occurs in poor lighting conditions and crowded situations. However, the processing speed makes it hard for real-time processing. Only achieving an average FPS of 6.03 with an average inference time of 197.44 milliseconds. This means that we will need to try a

different object detection model, change the hardware, or apply frame skipping strategies.

TABLE VI. PEDESTRIAN CROSSWALK DETECTION MODEL PERFORMANCE

Model	Performance Metrics					
	Accuracy	***Precision***	***Recall***	***F1***	***MAE***	***Avg FPS***
YOLO11n (ONNX)	0.91	0.91	0.91	0.89	0.09	6.03

V. CONCLUSION

The Real-Time Intelligent Traffic Alert System offers a solution for the shortcomings of pedestrian safety at crosswalks and the lack of directional information in traditional emergency vehicle sirens. The system incorporates three main modules. The first is the sound recognition module, which uses a self-trained CNN-based sound recognition module to identify the siren of emergency vehicles, and a microphone array to determine the direction. The second module is the image preprocessing module, which identifies different scenes and applies the corresponding enhancement strategy to improve the image for better recognition. The final module is the pedestrian crosswalk detection, which uses a YOLOv11n object detection model and an IoA mechanism to detect pedestrians on crosswalks. Combinedly, these modules offer clear and intuitive visual guidance for drivers. This research shows the potential of traffic safety edge-based AI detection systems.

VI. REFERENCES

[1] Department of Highway Administration and Road Safety, Ministry of Transportation and Communications, "Traffic Accidents - Road Safety Information Inquiry Website. [Online]. Available: https://roadsafety.tw/Dashboard/Custom?type=%E7%B5%B1%E8%A8%88%E5%BF%AB%E8%A6%BD%E5%9C%96%E8%A1%A8. (Accessed 19 Aug. 2025).

[2] Taoyuan City Government Fire Department, "Redirect Notice." *Google.com*, 2025, https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.tyfd.gov.tw/cht/index.php%3Fact%3Ddownload%26ids%3D9854&ved=2ahUKEwjrzJOtx6eNAxUBk68BHfLGACwQFnoECBkQAQ&usg=AOvVaw1CO6f8Z3se55Q31N5FTGQZ. (Accessed 19 Aug. 2025).

[3] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," *arXiv preprint arXiv:1905.02244*, 2019

[4] zly19980718, "[Image Rain and Fog Removal Dataset] Introduction to the Outdoor-Rain Dataset," *Csdn.net*, 10 Dec. 2024, blog.csdn.net/zly19980718/article/details/144382671. (Accessed 19 Aug. 2025).

[5] Tamim Ahasan Rijon, "Weather Detection Image Dataset," *Kaggle.com*, 2023, www.kaggle.com/datasets/tamimresearch/weather-detection-image-dataset?select=fogsmog. (Accessed 19 Aug. 2025).

[6] Ajayi, Gbeminiyi, "Multi-Class Weather Dataset for Image Classification," *Data.mendeley.com*, vol. 1, 13 Sept. 2018, data.mendeley.com/datasets/4drtyfjtfy/1, https://doi.org/10.17632/4drtyfjtfy.1.

[7] Merva Editors, "Photos are always too dark or too bright? First learn about the 3 elements of photography exposure," Merva Note | Selected e-book interactive platform | Explore the interesting things you want to learn, Jun. 09, 2021. [Online]. Available: https://mervanote.com/post/photography-exposure-20210609/ (Accessed Aug. 19, 2025).

[8] "Free Photo," *Photo-amodelc.com*, 2025. [Online]. Available: https://zh-tw.photo-ac.com/search/Late%20night?per_page=100&page=4&orderBy=popular&color=all&modelCount=-2&shape=all (Accessed Aug. 19, 2025).

[9] Canon Hong Kong Company Limited, "Night Photography Tutorial for Beginners," - *[Canon Hongkong Company Limited]*, 2025, [Online]. Available: https://www.canon.com.hk/tc/club/article/itemDetail.do?itemId=10376&page=1. (Accessed 19 Aug. 2025).

[10] Real-time video monitor, "Real-time video monitor: Real-time video of road conditions in Taiwan, weather observation at tourist attractions." [Online]. Available: https://tw.live/, 2025. (Accessed 19 Aug. 2025).