IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 August 2023, accepted 23 August 2023, date of publication 28 August 2023, date of current version 5 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3309410

RESEARCH ARTICLE

Contrast Limited Adaptive Histogram Equalization for Recognizing Road Marking at Night Based on Yolo Models

RUNG-CHING CHEN[1], (Member, IEEE), **CHRISTINE DEWI**[2], **YONG-CUN ZHUANG**[1], AND **JEANG-KUO CHEN**[1]

[1]Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan
[2]Faculty of Information Technology, Satya Wacana Christian University, Salatiga, Central Java 50711, Indonesia

Corresponding author: Christine Dewi (christine.dewi@uksw.edu)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant NSTC-111-2221-E-324-020, Grant NSTC-111-2622-E-324-002, Grant NSTC-112-2221-E-324-003-MY3, and Grant NSTC-112-2221-E-324-011-MY2.

ABSTRACT In recent years, artificial intelligence has led to rapid development and application across various industries. One of the significant developments is the improvement of transportation methods. Accidents involving vehicles frequently result in a high number of fatalities as well as economic damage. Road detection is one of the applications that can be used by self-driving cars. Traffic accidents happen, but artificial intelligence is used in many nations to construct smart cities and apps for self-driving cars. Since public road sign datasets have been used in significant research for road sign identification and analysis, these datasets are particularly significant for training autonomous vehicles. This study records the roads of various cities in Taiwan through road driving. It manually collects traffic signs in Taiwan to create a data set of road signs in Taiwan in daytime environments as well as nighttime environments. Since there is currently no data set of road signs in Taiwan, this study is necessary to create a data set of road signs in Taiwan. The YOLO model is utilized in this work to design road signs in Taiwan for mark detection. The techniques of Contrast Stretching (CS), Histogram Equalization (HE), and Contrast Limited Adaptive Histogram Equalization (CLAHE) are evaluated in a nighttime setting and compared to the original image captured at night. The experimental results show that the best model during the day is YOLO V4 (no flip), the test set mAP is 86.77%, the Precision is 82%, the Recall is 87%, the F1-score is 84%, and the IoU is 63.92%. At night, the CLAHE image method works best in the YOLOv5x model, with a mAP of 86.40%. YOLOv5 can be used in mobile devices or embedded devices, so this study recommends using CLAHE's YOLOv5x as the best model at night and used to improve the effect of road sign detection at night.

INDEX TERMS YOLO, road sign detection, CLAHE, CNN, deep learning.

I. INTRODUCTION

Road marking identification is an essential task in computer vision and autonomous driving. These tasks aim to identify and track road markings, such as lane lines, crosswalks, and stop lines, in real time using sensors and cameras mounted on vehicles. This information is used to aid in navigation, obstacle avoidance, and overall safety of the vehicle and its passengers [1], [2]. Road marking detection involves identifying road markings in an image or video frame. This can be done using various computer vision techniques, such as edge detection, Hough transforms, and deep learning-based methods. Once the presence of road markings is detected, the next step is to recognize the type of road marking, such as a solid or dashed line or a crosswalk.

The following step in road marking recognition categorizes the detected road marking. Support vector machines, decision trees, and convolutional neural networks (CNNs) are common machine learning methods used for this purpose. The classification algorithm is trained on a large dataset of labeled

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

road markings and can accurately recognize the marking in real time [3].

Both road marking detection and recognition are critical components of autonomous driving systems, as they provide valuable information for the vehicle to navigate safely. However, these tasks can be challenging due to variations in lighting conditions, weather, and road surface conditions. Thus, ongoing research is being conducted to improve the accuracy and reliability of road marking detection and recognition algorithms [4].

Machine learning, deep learning, image processing, and computer vision algorithms are only a few approaches to road sign detection. Some common techniques used in road sign detection include color-based segmentation, shape analysis, edge detection, and feature extraction. There are several places where the ability to recognize road signs may be useful, including ADAS, driverless cars, and ITS. There will be fewer accidents and more efficient traffic flow as drivers can better recognize speed limits, stop signs, yield signs, and other crucial road indicators [5].

Road marking sign identification has several benefits, including: (1) Improving road safety: Accurately identifying road marking signs can help drivers to follow traffic rules and regulations, such as speed limits and lane markings, which can reduce the risk of accidents and enhance road safety. (2) Enhancing traffic efficiency: Road marking sign identification can help drivers to navigate through complex road networks, including highways and intersections, which can improve traffic flow and reduce congestion. (3) Enabling autonomous driving: Road marking sign identification is an essential component of autonomous driving systems, which rely on computer vision algorithms to interpret real-time traffic data and make driving decisions. (4) Supporting intelligent transportation systems (ITS): Road marking sign identification can provide valuable data for ITS, such as traffic flow patterns and congestion levels, which can help traffic engineers optimize traffic management and reduce travel time [6].

YOLO (You Only Look Once) is a popular object detection system in computer vision and image processing. Moreover, YOLO is known for its real-time performance, as it can detect objects in an image with high accuracy and speed [7], [8]. Some of the benefits of the YOLO (You Only Look Once) object detection system are: (1) Real-time object detection: YOLO is designed to process images in real-time, which means it can detect objects in a video stream at a high frame rate. This makes it a suitable algorithm for applications that require real-time object detection, such as autonomous vehicles, security cameras, and robotics. (2) High accuracy: YOLO has achieved state-of-the-art results in object detection benchmarks, demonstrating high accuracy in detecting objects of various sizes and orientations. (3) End-to-end system: YOLO is an end-to-end system that combines object detection and classification into a single network, which simplifies the training and deployment process. (4) Low memory requirements: YOLO is designed to operate on a single network, which reduces its memory requirements compared to other object detection systems that require multiple networks. (5) Flexibility: YOLO can be trained on custom datasets and can be fine-tuned for specific applications, making it a versatile algorithm that can adapt to different use cases [9], [10].

The following is a brief overview of the study's most important contributions: (1) This study collects actual driving videos. It establishes the Taiwan Road Marking Sign Dataset at night (TRMSDN) for road markings. (2) Different YOLO models are tested to find the best version for identifying Taiwan's traffic signs on the road. (3) Under nighttime driving conditions, the study compares and discusses changes in original images, Contrast Stretching (CS), Histogram Equalization (HE), and Contrast Limited Adaptive Histogram Equalization (CLAHE). (4) The effectiveness of detecting actual road markings in Taiwan is tested using the recognition system established in this study across different YOLO versions. (5) Experiments show that CLAHE can improve all models' performance.

This paper will proceed as described below. In Part II, we discuss some similar work. In Section III, we detail our proposed methodology. The procedure and findings are presented in Section IV. Section V presents a comprehensive analysis and discussion of our results. In Section VI, we draw some conclusions and suggest further research.

II. RELATED WORKS

A. ROAD SIGN DETECTION

Recent years have seen the effective implementation of real-time traffic sign recognition in autonomous vehicles and driver-aid systems. YOLO-based road sign recognition has also shown significant interest, and several papers have been written on the subject. Warning, speed limit, directional, and prohibited signs comprise the CSUST (Chinese Traffic Sign Detection Benchmark) dataset, which was used to evaluate YOLOv3 and YOLOv4 by Yang and Zhang [11]. YOLOv4 did better in the experiments in target detection than YOLOv3, especially when recognizing road signs and finding little things. Another technique for traffic sign identification utilizing YOLOv3 and a unique image dataset was proposed by Miji et al. [12]. To assist management in meeting social demands for traffic safety, Gatelli et al. [13] suggested a vehicle classification approach applicable in Brazil. They used YOLOv4 as the target detector. Mohd-Isa et al. [14] used the YOLOv3 framework, which includes Spatial Pyramid Pooling (SPP), to detect Malaysian traffic signs and recognize minor signs in real-world environments. Ye et al. [15] used YOLOv2, RM-Net, Faster R-CNN, and SSD to detect road signs and proposed a two-stage network based on YOLOv2 to address distortion in road detection, as well as mean average Precision and Recall.

Tai et al. [16] used DCGAN, LSGAN, and WGAN to synthesize high-quality images of prohibited road signs. They used synthetic images to improve the IoU (Intersection over Union) and performance of the YOLOv3 and YOLOv4 models. Dewi et al. [17] proposed a deep learning-based method for detecting and recognizing various prohibitory signs using YOLO and YOLOv3 SPP. The best average classification accuracy achieved was 99.0%. Dewi et al. [18] developed a CNN-based traffic sign classification solution, combining synthesized and original images to enhance the dataset and verify the effectiveness of synthetic data.

B. IMAGE ENHANCEMENT

Image enhancement refers to improving the visual quality of a digital image. It involves applying various techniques to an image to make it more visually appealing or extract more information. Image enhancement techniques can be broadly classified into two categories: spatial domain techniques and frequency domain techniques [19]. Spatial domain techniques involve manipulating the pixel values of an image directly. In contrast, frequency-domain techniques include transforming the image into its frequency domain using techniques such as Fourier transforms and manipulating the frequency components to enhance the image.

Some common techniques used in image enhancement include brightness and contrast adjustment, histogram equalization, noise reduction, sharpening, and color correction. These methods can be implemented either by hand via picture editing tools or automatically via custom-built algorithms. Image enhancement is widely used in various fields, such as photography, medical imaging, satellite imaging, and video processing [20].

C. CONTRAST STRETCHING

Contrast Stretching (CS) is a simple image processing technique used to enhance the contrast of an image by increasing the dynamic range of pixel intensities. The goal of contrast stretching is to improve an image's brightness and contrast to appear clearer and more vibrant. Further, CS known as normalization, is a simple image enhancement technique that tries to increase the dynamic range of gray intensity by using piecewise linear transformation to stretch the gray levels, sacrificing the gray levels from 0 to r1 and from r2 to L-1 [21], [22].

CS increases the range of gray levels from s1 to s2 to improve image contrast and quality. The formula for contrast stretching is shown in Formula (1) [23], where J is the original gray level image value, *J_min* is the minimum gray level image value in the entire Image, *J_max* is the maximum gray level image value in the whole image, and *J_new* is the final output new gray level image value.

$$\boldsymbol{J_{new}} = \frac{\boldsymbol{J - J_{min}}}{\boldsymbol{J_{max} - J_{min}}} \boldsymbol{\times 255} \tag{1}$$

D. HISTOGRAM EQUALIZATION (HE)

Histogram equalization (HE) is a technique used in image processing to enhance the contrast of an image. The basic idea behind HE is to transform the pixel values of an image so that they are more evenly distributed across the entire range of possible values [24]. In HE, the image's histogram is first computed, a graphical representation of the frequency of occurrence of each pixel value in the image [25]. Then, the histogram is equalized by redistributing the pixel values in a way that results in a more uniform distribution of values across the entire range of possible values [26].

The HE processes involves two steps: (1) computing the cumulative distribution function (CDF) of the image and (2) mapping the pixel values of the image to a new range using the CDF [27]. The CDF is computed by summing up the histogram values from the leftmost bin to the rightmost bin. Then, the pixel values are mapped to the new range by multiplying the CDF value by the maximum pixel value and rounding off to the nearest integer. HE can be applied to grayscale and color images. However, it may not be suitable for all types of images, especially those with a narrow range of pixel values or with extreme brightness or contrast variations. In such cases, other image enhancement techniques may be more appropriate. The HE f1ormula is as follows: Let I be the input image with M rows and N columns and let h(k) be the histogram of I with k in the range of 0 to L-1 (L is the number of intensity levels). (1) Compute the normalized histogram as seen in Formula (2).

$$p(k) = \frac{h(k)}{M \times N}, for\ k = 0, 1, \ldots, L-1 \tag{2}$$

(2) Compute the cumulative distribution function (CDF) of the normalized histogram with Formula (3).

$$CDF(j) = SUM(p(k)), for\ k = 0, 1, \ldots, j, \quad and\ j = 0, 1, \ldots, L-1 \tag{3}$$

(3) Compute the new intensity values for each pixel with Formula (4).

$$s = round\,(L-1) \times CDF(I)) \tag{4}$$

(4) Replace each pixel in the input image with its corresponding new intensity value. The HE formula maps the original image's intensity values to new values based on the cumulative distribution function of the histogram. This spreads the intensity values over the entire range of intensities, resulting in an image with improved contrast.

E. ADAPTIVE HISTOGRAM EQUALIZATION (AHE)

Adaptive Histogram Equalization (AHE) is a variation of the histogram equalization technique used to enhance an image's contrast, particularly in areas of the image that have low contrast or are affected by uneven illumination [28]. Unlike traditional histogram equalization, which applies the same equalization function to the entire Image, AHE applies different equalization functions to different image regions based on the local image statistics. The AHE process involves dividing the image into small regions or tiles, computing each tile's histogram, and equalizing each tile's histogram independently. This ensures that the contrast enhancement is applied only to the regions that require it rather than to the entire image. One of the main advantages of AHE is that it preserves the local contrast of the image while enhancing the overall contrast. This makes it particularly useful for images with complex textures, such as medical or satellite images [29].

However, AHE has some limitations. It can introduce artifacts in regions where the local histogram is very narrow, such as in uniform image areas. This can lead to a visible grid-like pattern in the output image, known as the "halo effect." Various modifications of AHE have been proposed to overcome this limitation, such as contrast-limited adaptive Histogram Equalization (CLAHE), which limits the contrast enhancement in each tile to avoid the halo effect.

F. CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION (CLAHE)

Contrast Limited Adaptive Histogram Equalization (CLAHE) is a modification of the Adaptive Histogram Equalization (AHE) technique used to enhance the contrast of an image while avoiding artifacts such as the "halo effect" that can occur in AHE [30]. In traditional AHE, the contrast enhancement can be excessive in image regions with a narrow histogram [31]. This can lead to the over-amplification of noise and artifacts in those regions, resulting in a visible grid-like pattern around the edges, known as the "halo effect". CLAHE overcomes this limitation by limiting the contrast enhancement in each image tile. The limiting is done by clipping the histogram at a predefined value called the "clip limit". This ensures that the contrast enhancement is not excessive in regions with a narrow histogram and avoids the over-amplification of noise and artifacts.

The CLAHE process involves dividing the image into small regions or tiles, computing the histogram of each tile, and then clipping the histogram at the clip limit. The clipped histogram is then equalized, and the resulting pixel values are used to reconstruct the image. The clip limit is chosen empirically based on the characteristics of the image and can be adjusted to achieve the desired level of contrast enhancement. CLAHE has been widely used in medical imaging, particularly in X-ray and CT scans, where it can enhance the image's contrast and reveal subtle details that the human eye may miss. It is also useful in other fields, such as satellite imaging and computer vision.

G. YOLO

The YOLO algorithm has gone through several versions, including YOLOv1, YOLOv2, YOLOv3, and YOLOv4 [32]. Each version has introduced improvements to the original algorithm, such as better accuracy, faster processing speed, and improved handling of small objects and object occlusion [34].

YOLOv2 was proposed by Joseph Redmon and Ali Farhadi in 2017 [33]. It uses a new architecture, Darknet-19, which includes 19 convolutional layers and 5 max-pooling layers, designed similarly to VGG16, using 3 × 3 convolutional layers and 2 × 2 max-pooling layers. Compared with YOLOv1, the main improvements are higher recall and localization capabilities. This version no longer uses dropout but instead adds batch normalization after each convolutional layer to improve the convergence speed of the model and reduce overfitting. This version increases the resolution of image classification (High-Resolution Classifier) and introduces the Anchor box from Faster RCNN. The former uses a 224 × 224 image classifier in YOLOv1, which has low resolution and is not conducive to model detection, so YOLOv2 increases the resolution to 448 × 448. The latter is removed from the previous generation's fully connected layer (Fully Connected Layer) and uses Anchor Box to predict bounding boxes, as YOLOv1 is more difficult to adapt to different shapes, resulting in less accurate detection positions. To find more suitable Anchor Boxes and enable the model to predict accurate bounding boxes more easily, YOLOv2 uses k-means clustering to analyze the bounding boxes in the training set.

YOLOv3 was proposed by Joseph Redmon and Ali Farhadi in 2018 [40]. This version introduced a new backbone called Darknet-53, which consists of 53 convolutional layers and additional ResNet (Residual Network) layers from layer 0 to layer 74. Compared to the previous version, Darknet-19, Darknet-53 removed all the max pooling layers and added many 1 × 1 and 3 × 3 convolutional layers. However, increasing the number of layers in the network can lead to the problem of vanishing or exploding gradients, so Darknet-53 incorporated the ResNet network to solve the gradient problem. YOLOv3 uses the Feature Pyramid Network (FPN) method and multiple scales of feature maps to detect objects of different sizes, enhancing the prediction ability for small objects. This version also uses logistic regression to predict the confidence of bounding boxes and uses the IoU between the bounding box and the ground truth as the evaluation criterion.

Complete YOLOv4 specifications are explained as follows: (1) Bag of Freebies (BoF) [35] Backbone: CutMix [36] and Mosaic data augmentation, DropBlock [37] regularization, and Class label smoothing. CIoU-loss, CmBN, and DropBlock regularization are used in the detector. Enhancing mosaic data, self-adversarial instruction, getting rid of the grid, Sensitivity, relying on various sources to establish a common ground or reality, Timetabler for Cosine Annealing [38], Optimal hyperparameters, and Random training shapes. (2) The Bag of Specials, also known as BoS Mish activation, Cross-stage partial connections (CSP), and Multi-input weighted residual connections (MiWRC) make up the backbone of the algorithm. Detector components are mish activation, SPP-block, SAM-block, PAN path-aggregation block, and DIoU-NMS.

YOLO algorithm [34] is a typical end-to-end network construction. This algorithm is shorter than the R-CNN [39], [40]. The YOLOv4 head employs the YOLOv3 model as a one-stage dense prediction. In addition, YOLOv3 segments the input image into squares of the same size by dividing it into S×S grids [41], predicts bounding boxes and possibilities to every grid cell. In addition, YOLOv3 uses multiscale fusion to make predictions about the entire image.

The whole image is preprocessed using a single CNN using this technique. Clusters are used to evaluate the boundary lines.

YOLOv5 is the latest YOLO object detection system version, released in 2020 by Glenn Jocher and developed by Ultralytics. YOLOv5 is based on the EfficientDet object detection framework and uses a single-shot detector (SSD) architecture, like previous versions of YOLO. One of the critical improvements of YOLOv5 over previous versions is its speed and accuracy. YOLOv5 is faster and more accurate than previous versions due to the use of a more efficient backbone network and improved training techniques. The new architecture is also more flexible, allowing for easier customization and transfer of learning. YOLOv5 can be trained on a smaller dataset and still achieve high accuracy, making it more accessible to researchers and developers with limited data [42], [43].

YOLOv5 is available in several versions, including YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, which differ in model size and performance. YOLOv5s is the smallest and fastest version, while YOLOv5x is the most extensive and accurate [44], [45].

III. METHODOLOGY

A. RESEARCH WORKFLOW AND EXPERIMENT SETTING

In this experiment, we classify our dataset into 4 groups: Dataset 1, Dataset 2, Dataset 3, and Dataset 4, as seen in Table 1. Dataset 1 consists of the original image without an image enhancement method. The original image and the CS contrast stretching improvement are included in Dataset 2. In addition, Dataset 3 combines the original and HE images. Dataset 4 is the final one, including the CLAHE-enhanced original image.

TABLE 1. Dataset combination setting.

Dataset	Original Image	Image Enhancement		
		CS	HE	CLAHE
Dataset 1	✓			
Dataset 2	✓	✓		
Dataset 3	✓		✓	
Dataset 4	✓			✓

An overview of system approaches can be seen in Figure 1 (a) Research Workflow. After preparing the dataset with the image enhancement, we trained our dataset with the YOLO model. The YOLO model includes YOLOv2, YOLOv3, YOLOv4, YOLOv4-csp, YOLOv4-tiny, YOLOv4-mish, YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Then, we compare the outcomes of the various models.

The BBox mark tool [46] was adopted to create a bounding box for all signs. The labeling procedure is carried out for each class. Multiple marks may be assigned to a single image. During the detection phase, only one class detector model was used, and each class label was associated with a separate training model. The bounding box labeling tool's return values are object coordinates $(\boldsymbol{x_1}, \boldsymbol{y_1}, \boldsymbol{x_2}, \boldsymbol{y_2})$. These coordinates of items are distinct from the input value of Yolo. Instead, the Yolo input value is the center point, width, and height *(x, y, w, h)*. As a result, the system must adjust the bounding box coordinates in the Yolo input format. The modification process is based on Formula (5) – (10).

$$dw = 1/W \tag{5}$$

$$x = \frac{(x_1 + x_2)}{2} \times dw \tag{6}$$

$$dh = 1/H \tag{7}$$

$$y = \frac{(y_1 + y_2)}{2} \times dh \tag{8}$$

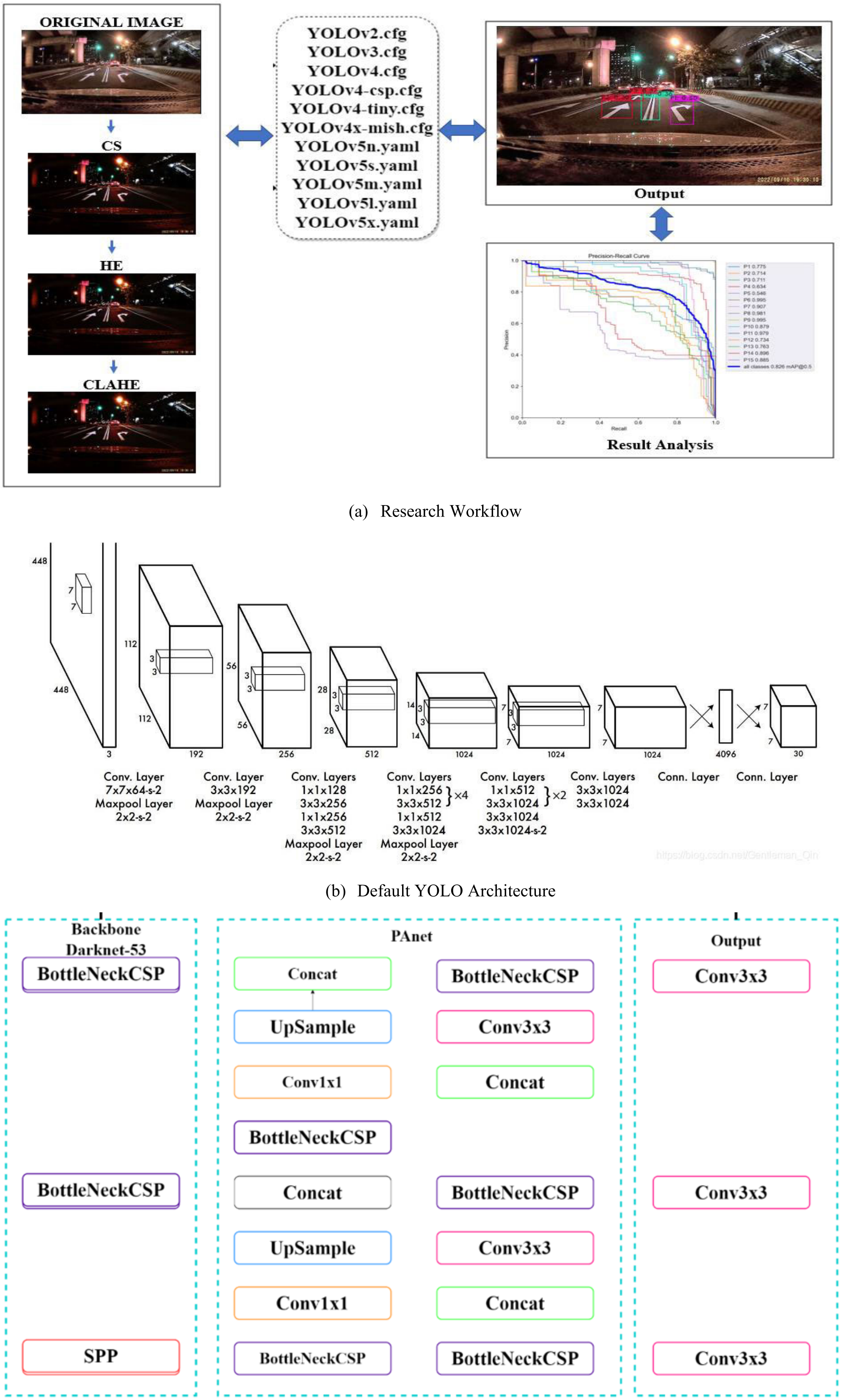
$$w = (x_2 - x_1) \times dw \tag{9}$$

$$h = (y_2 - y_1) \times dh \tag{10}$$

Image dimensions are as follows: $H =$ image height, $dh =$ absolute image height, $W =$ image width, and $dw =$ absolute image width. This means that the range for the picture's width and height (*dw, dh*) can be from 0.0 to 1.0 in float form.

The YOLO architecture is shown in its most basic form in Figure 1(b). One-stage object detection was implemented in YOLOv1 as a response to the slow calculation performance of the two-stage object detection. One-stage object detection can achieve a processing speed of 45 FPS (frames per second), which satisfies real-time requirements. The GoogLeNet model serves as the foundation for YOLOv1's convolutional neural network architecture, which includes 24 convolutional layers and 2 fully connected layers. The former is primarily utilized for feature extraction, whilst the latter is principally utilized for the prediction of probability and coordinate placements. The tensor formula for YOLOv1 is written as S S (B 5 + C), where S denotes the input dimension, B denotes the number of bounding boxes in each grid, 5 denotes the five prediction values for each bounding box, specifically the center point (x, y), width and height (w, h), and confidence, and C denotes the number of classes that are associated with the object.

The architecture of YOLOv5 is depicted in Figure 1(c). The input, the backbone, the neck, and the output are the four primary components that make up YOLOv5. The major responsibility of the Backbone Model is to identify significant pieces for analysis from inside the input image. Cross Stage Partial Networks (CSP) and Spatial Pyramid Pooling (SPP) are the fundamental building blocks that Yolov5 uses when it comes to extracting rich and crucial attributes from input photographs. This is accomplished with the help of Yolov5. When it comes to the accurate generalization of a model for object scaling, it is essential to correctly recognize the same item in numerous sizes and scales. SPP is helpful in this endeavor. The development of the neck network makes use of the feature pyramid architectures of the Feature Pyramid Network (FPN) and the Path Aggregation Network



(a) Research Workflow

(b) Default YOLO Architecture

(c) YOLOv5 Architecture

FIGURE 1. An overview of the system (a) Research workflow, (b) Default YOLO architecture, and (c) YOLOv5 architecture.

TABLE 2. Training parameter value for YOLO v2, YOLO v3, and YOLO v4.

Parameter	YOLO v2	YOLO v3	YOLO v4	YOLO v4-csp	YOLO v4-tiny	YOLO v4x-mish
Size of Input images	512x512	512x512	512x512	512x512	512x512	512x512
Batch	64	64	64	64	64	64
Subdivisions	8	16	32	16	8	16
Learning Rate	0.001	0.001	0.0013	0.001	0.003	0.001
Maximum number of batches	30000	30000	30000	30000	30000	30000
Filters	85	51	51	51	51	51

TABLE 3. YOLO v5 Series model training parameter values.

Parameter	YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
Size of Input images	512x512	512x512	512x512	512x512	512x512
Batch	16	16	16	16	16
Epoch	100	100	100	100	100

(PANet). The FPN structure contains potent semantic features dispersed across its entirety, beginning at the top feature maps, and making their way down to the lower feature maps. These features progress through the structure from top to bottom. During this time, the PAN structure must ensure that trustworthy localization features are sent from lower feature maps to higher feature maps. Yolo version 5 makes use of PANet as a neck, which enables the development of a feature pyramid.

Table 2 shows the training parameter values for various models, including YOLOv2, YOLOv3, YOLOv4, YOLOv4 csp, YOLOv4-tiny, and YOLOv4x-mish. During the setup, the input size was adjusted to improve the detection accuracy of small objects, and the number of subdivisions was adjusted to provide enough memory space for training under the memory capacity limit of the graphics card. The remaining parameters were set to their default values. Our experiment set the input image size to 512 × 512 and batch 64.

Moreover, Table 3 shows the training parameter values for YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, and other models. Due to limited memory size, the batch size was set to 16, and the epoch size was determined to be the best training iteration through experimentation. We implemented 100 epochs in the experiment during training. Figure 2 contains information that provides a summary of all YOLOv5's models. This summary includes the inference speed on both the CPU and the GPU and the number of parameters with an image size of 512 × 512 pixels. Moreover, YOLOv5s (Small): This is the smallest variant of YOLOv5, designed for faster inference and deployment on resource-constrained devices. It has fewer layers and parameters compared to other variants, making it faster but potentially sacrificing some accuracy. Further, YOLOv5m (Medium): YOLOv5m strikes a balance between speed and accuracy. It has a moderate number of layers and parameters, making it suitable for a wide range of applications. Next, YOLOv5l (Large): YOLOv5l is a larger variant that offers higher accuracy but is slower than the smaller variants. It has more layers and parameters, allowing it to capture more intricate details in the detected objects. YOLOv5x (Extra Large): YOLOv5x is the largest variant of YOLOv5. It has the highest accuracy but is the slowest in terms of inference speed. YOLOv5x is suitable for applications where accuracy is paramount and inference speed is not the primary concern.

Furthermore, Table 4 describes our Experiment Specifications. The hardware specifications used in the experiment include an Intel Core i7-11700 CPU with 8 cores and an RTX 3080 GPU accelerator, along with 32GB DDR4-3200 RAM.

TABLE 4. Experiment specifications.

Name	Specifications
OS	Windows 10
CPU	I7-11700 (2.50 GHz, max 4.90 GHz)
GPU	RTX 3080 (10GB)
RAM	DDR4-3200 (32GB)

B. IMAGE ENHANCEMENT RESULT

Nighttime environments have low brightness and contrast, making distinguishing traffic signs from the background difficult. This reduces the quality of photos used for training and recognition. As a result, before running image recognition, the original nighttime road dataset is preprocessed with image enhancement to make the target area in the photographs easier to recognize. Adjustment results utilizing three different Image-enhancing techniques are displayed in Figure 3. This experiment's primary goal is to detect road surface markings better using three different picture-enhancing

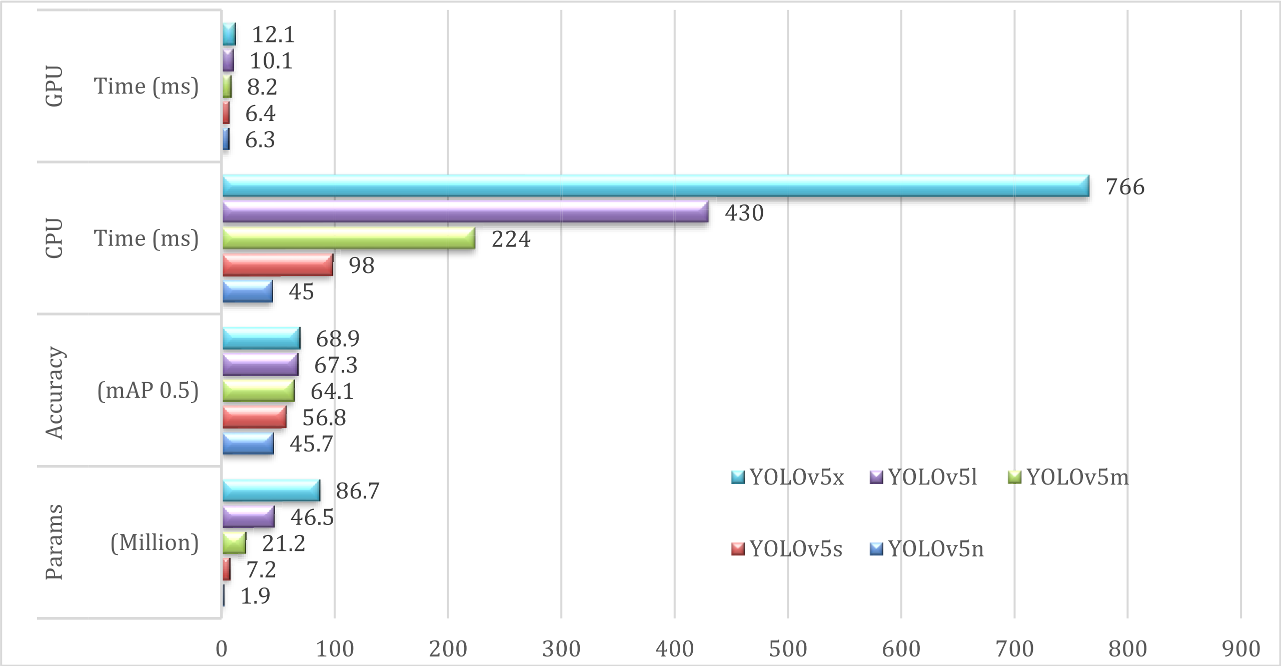
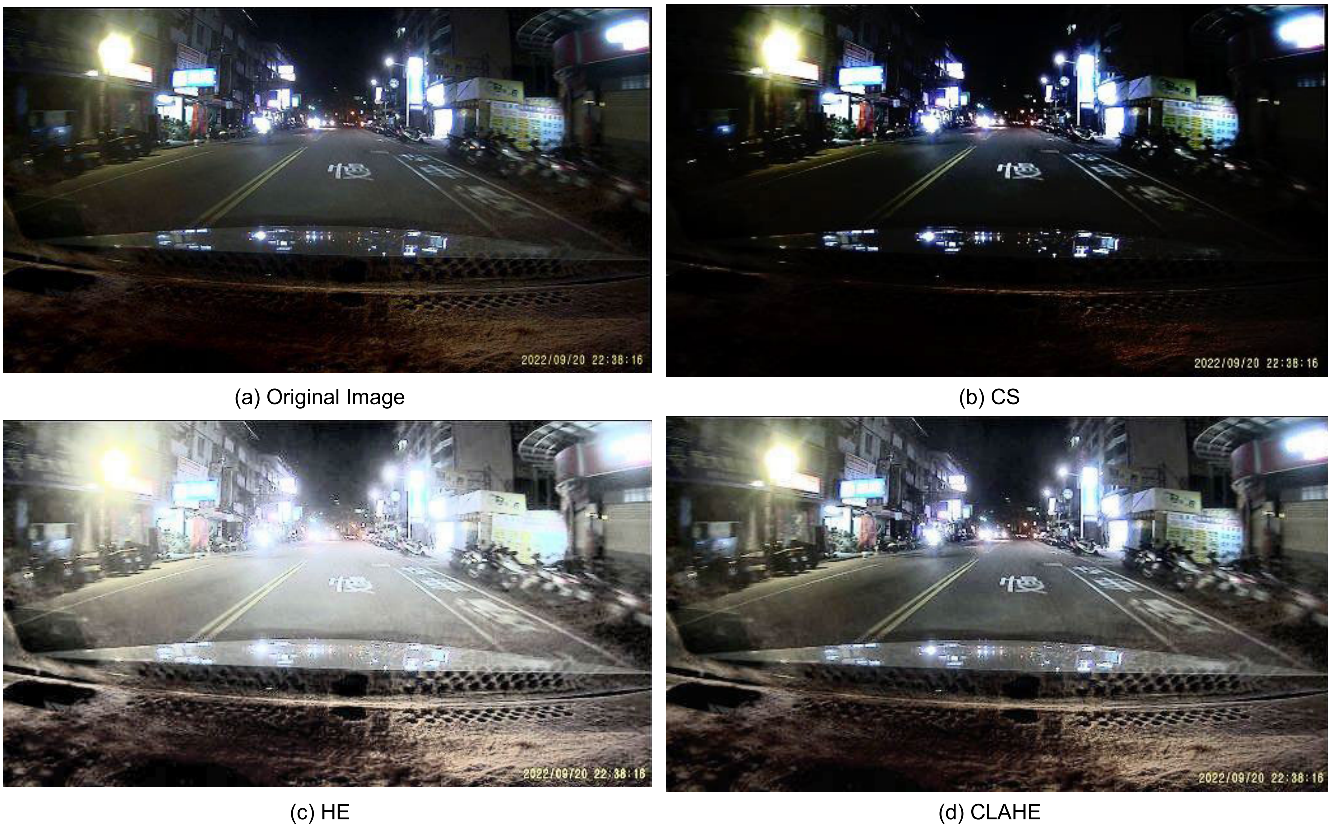


FIGURE 2. Overview of the Yolo V5 model.


(a) Original Image

(b) CS

(c) HE

(d) CLAHE

FIGURE 3. Example of the image enhancement methods with (a) . Original image, (b) CS, (c) HE, and (d) CLAHE.

algorithms and evaluate which is the most effective. The three image enhancement methods are: (1) Contrast Stretching exhibits in Figure 3(b). (2) Histogram Equalization is shown in Figure 3(c). (3) CLAHE, shown in Figure 3(d).

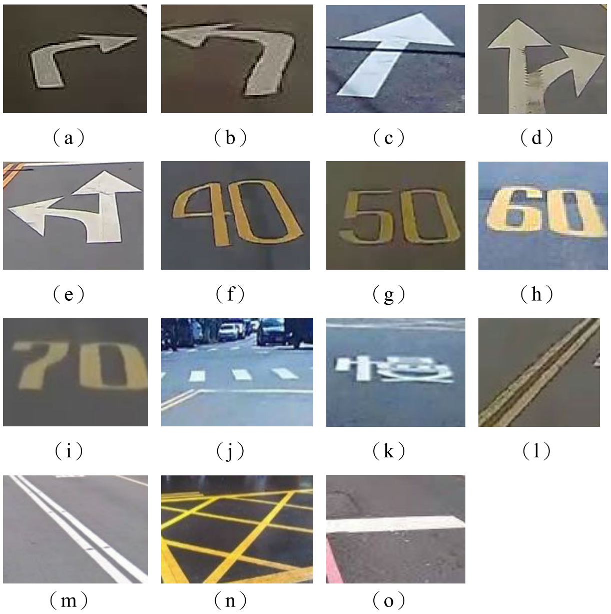


FIGURE 4. Taiwan road marking sign dataset at night (TRMSDN) sample class. (a) Turn right, (b) Turn left, (c) go straight, (d) Turn right or go straight, (e) Turn Left or Go Straight, (f) Speed limit(60), (g) Speed limit(60), (h) Speed limit(60), (i) Speed limit(70), (j) Zebra crossing, (k) Slow sign, (l) Overtaking prohibited, (m) Barrier line, (n) Cross hatch, and (o) Stop line.

IV. EXPERIMENT RESULTS

A. DATASET

Furthermore, we conducted our experiment using actual traffic signs on general highways in Taiwan, including those for turning right or left, speed restrictions, zebra crossings, and stop lines. Vehicles should modify their actions or be limited when they see these signs. In this experiment, we addressed the challenge of road sign recognition by recording the roads of different cities in Taiwan from the driver's perspective and manually collecting Taiwan's traffic signs on the road to create a unique dataset, namely Taiwan Road Marking Sign Dataset at Night (TRMSDN). We amassed 4386 individual photos for training purposes, splitting them between 15 distinct classes and allocating 80% to the training set and 20% to the testing set. A 512 × 288 pixel picture is used. The collection includes pictures of traffic signs, as shown in Figure 4. Moreover, these speeds were not included due to a lack of 40, 50, and 70 mph speed limit signage on the highways from which the night dataset was compiled. The training and testing labels for each road marking sign type are displayed in Figure 4 and Table 5, respectively. Besides, Figure 5 describes the Taiwan Road Marking Sign Dataset at Night (TRMSDN) instances. The TRMSDN dataset has an average of 399 to 409 instances per class. Our dataset can be accessed online through this link: https://drive.google.com/drive/u/1/folders/1U2qlw-ViqdW1pny77aJGLUIEf0B-HKzZ.

TABLE 5. Taiwan road marking sign dataset at night (TRMSDN).

Class	Name	Training	Testing	Total Image
P1	Turn Right	277	69	405
P2	Turn Left	301	75	401
P3	Go Straight	283	71	407
P4	Turn Right or Go Straight	307	77	409
P5	Turn Left or Go Straight	272	68	403
P6	Speed limit(60)	276	69	400
P7	Zebra Crossing (crosswalk)	327	82	401
P8	Slow Sign	275	69	399
P9	Overtaking Prohibited	310	77	404
P10	Barrier Line	295	74	409
P11	Cross Hatch	302	76	398
P12	Stop Line	283	71	403
	Total	3509	877	4386

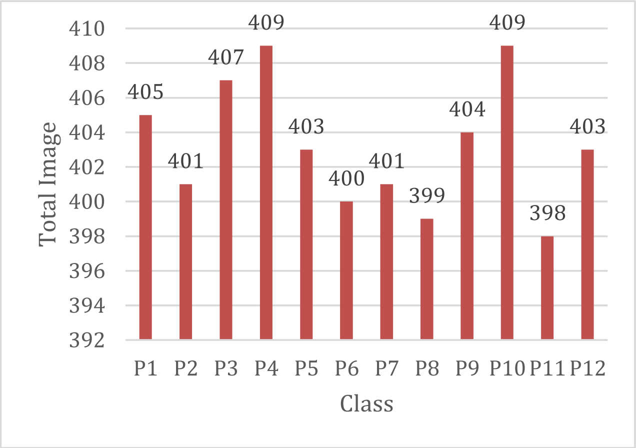
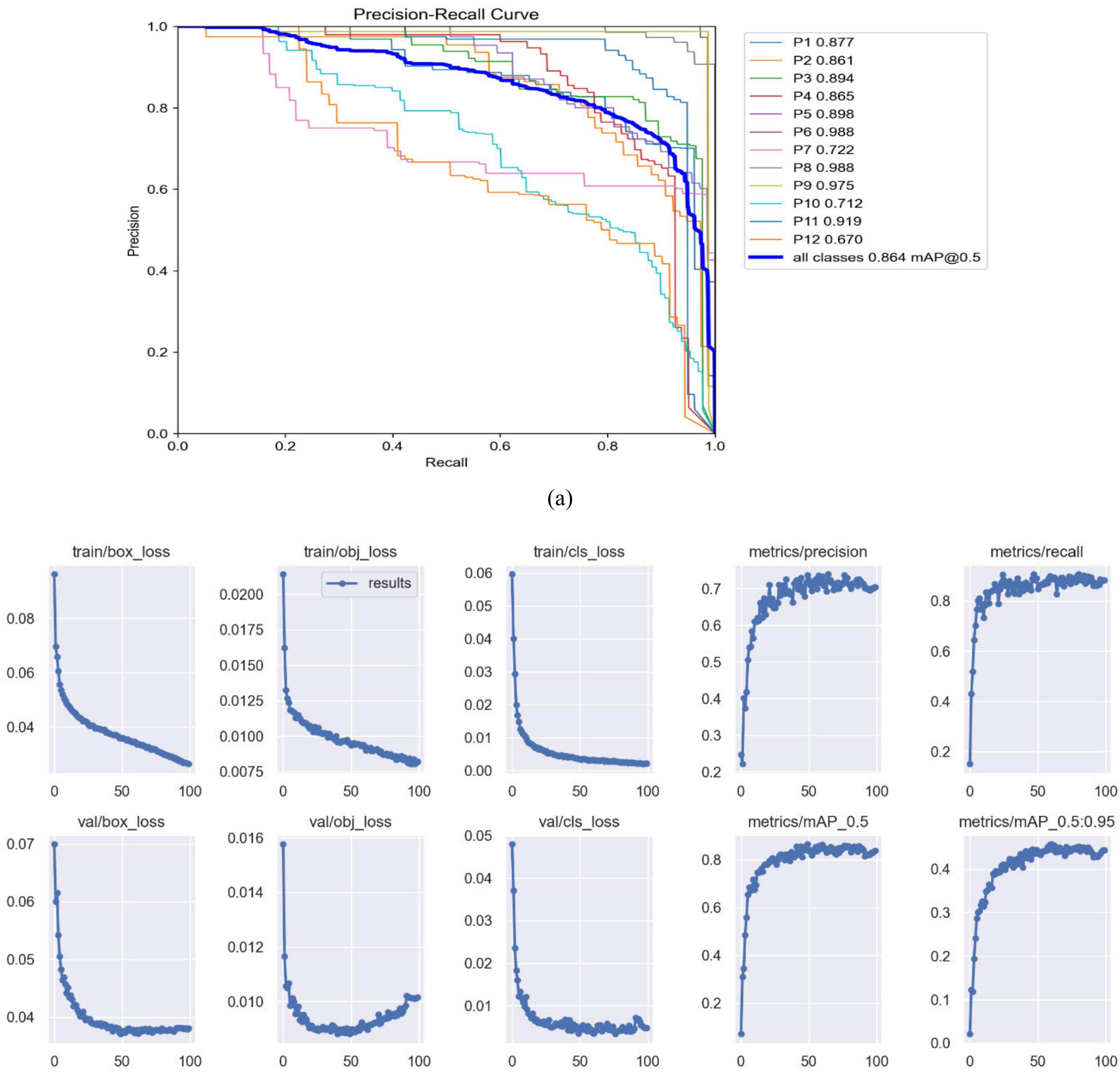


FIGURE 5. Taiwan road marking sign dataset at night (TRMSDN) instances.

B. TRAINING RESULTS

Our research improves the YOLOv2, YOLOv3, and YOLOv4 models during training by employing a 0.001 learning rate for analysis, a 0.1 learning rate decay at each epoch, and a momentum of 0.9. Our experiment used cross-validation and early halting to address the over-fitting issue. The out-of-sample prediction error is often obtained by the tried-and-true method of 5-fold cross-validation. The number of iterations that can be performed before the learner over-fits can be determined with the help of early stopping criteria. This experiment applies *max_batches* = 24000 iterations, policy = steps, scales = 0.1,0.1, momentum = 0.949, decay = 0.0005, and mosaic = 1. Further, the scale (0.1, 0.1) and the current iteration number 0.001 batches are used in the training process. Calculating the current learning rate becomes learning rate × scales [0] × scales [1] = 0.00001, and the learning rate value will be updated regularly.

The training results for one of the YOLOv5 models that were developed using PyTorch are presented in Figure 6. This model is known as the YOLOv5x model. The training results for each category are displayed as a function of the



(a)

(b)

FIGURE 6. **YOLOv5x nighttime CLAHE training results. (a) Precision and recall curve, and (b) Training and validation graph.**

number of times the neural network was iterated. This includes the results of the box loss function, the target recognition accuracy loss function, the classification error loss function, test accuracy, and test omission rate. The mean absolute percentage for this model is 86.4%, indicated in Figure 6(a). After more than 30 training cycles, the various loss functions began to decline, indicating that the training of the network's parameters had converged. All the loss functions went in the same direction and gradually got smaller, indicating that the network parameters were trained correctly. This is illustrated in Figure 6(b).

IoU calculates the overlap ratio between the boundary box of the prediction (*pred*) and ground truth (*gt*), as shown in Equation (11) [47], [48].

$$IoU = \frac{Area_{pred} \cap Area_{gt}}{Area_{pred} \cup Area_{gt}} \tag{11}$$

Nevertheless, the output examples can be classified into three classes. True positive (TP) is the number of correctly recognized samples; false positive (FP), which assigned to the number of samples with incorrect identification; true negative (TN) is the number of unrecognized samples.

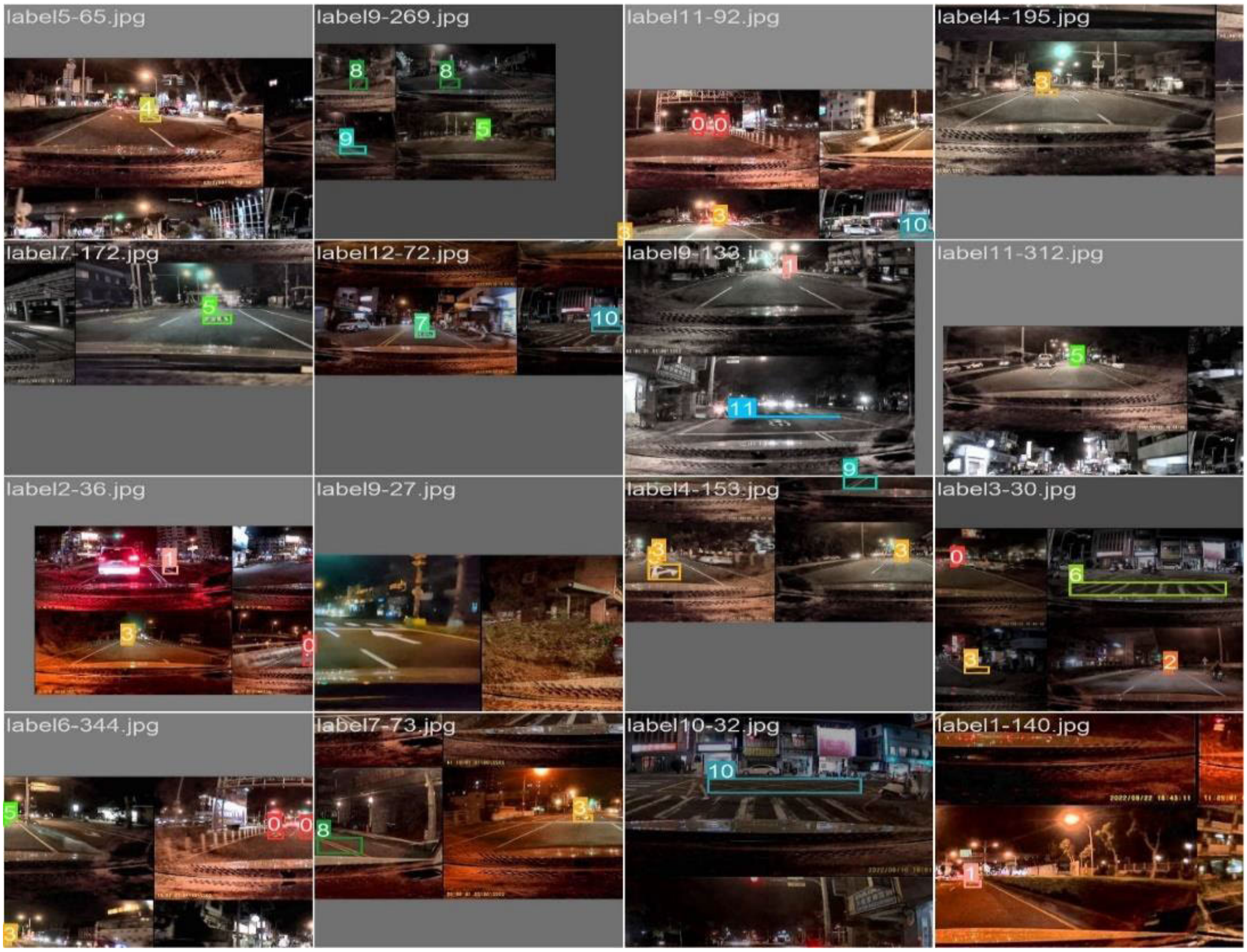


FIGURE 7. Training batch 0 with dataset 3.

Precision and Recall are represented by [49] and [50] in Equation (12)-(13).

$$Precision(P) = \frac{TP}{TP + FP} \tag{12}$$

$$Recall(R) = \frac{TP}{TP + FN} \tag{13}$$

Another evaluation index, F1 [51], [52], [53], is shown in Equation (14).

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

Yolo loss function based on Equations (15) [33].

$$\begin{aligned}
&\lambda_{coord} \sum\nolimits_{i=0}^{s^2} \sum\nolimits_{j=0}^{B} l_{ij}^{obj} \left[\left(x_i - \hat{x}_i\right)^2 + \left(y - \hat{y}_i\right)^2 \right] \\
&+ \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \boldsymbol{l_{ij}^{obj}} \\
&\times \left[\left(\sqrt{\boldsymbol{w_i}} - \sqrt{\boldsymbol{\hat{w}_i}}\right)^{\mathbf{2}} + \left(\sqrt{\boldsymbol{h_i}} - \sqrt{\boldsymbol{\hat{h}_i}}\right)^{\mathbf{2}} \right] \\
&+ \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 \\
&+ \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{ij}^{noobj} \left(C_i - \hat{C}_i\right)^2 \\
&+ \sum\nolimits_{i=0}^{s^2} l_i^{obj} \sum\nolimits_{c \epsilon classes} \left(p_i(c) - \hat{p}_i(c)\right)^2
\end{aligned} \tag{15}$$

where $\boldsymbol{l_{ij}^{obj}}$ denotes if the object appears in cell i, and $\boldsymbol{l_{ij}^{obj}}$ denotes that the $\boldsymbol{j^{th}}$ bounding box predictor in cell i is responsible for the prediction. Next, $\left(\boldsymbol{\hat{x}, \hat{y}, \hat{w}, \hat{h}, \hat{c}, \widehat{and\ p}}\right)$ are represented as the predicted bounding box's center coordinates, width, height, confidence, and category probability. Those symbols without the cusp are true labels. Furthermore, our works set the $\boldsymbol{\lambda coord}$ to 0.5, indicating that the width and height errors are less effective in calculating. Then, $\boldsymbol{\lambda noobj}$ = 0.5 is introduced to weaken the influence of many grids without objects on the loss value.

A machine learning model absorbs new information during the training phase by studying previously collected data [54]. During the training phase of the process, the

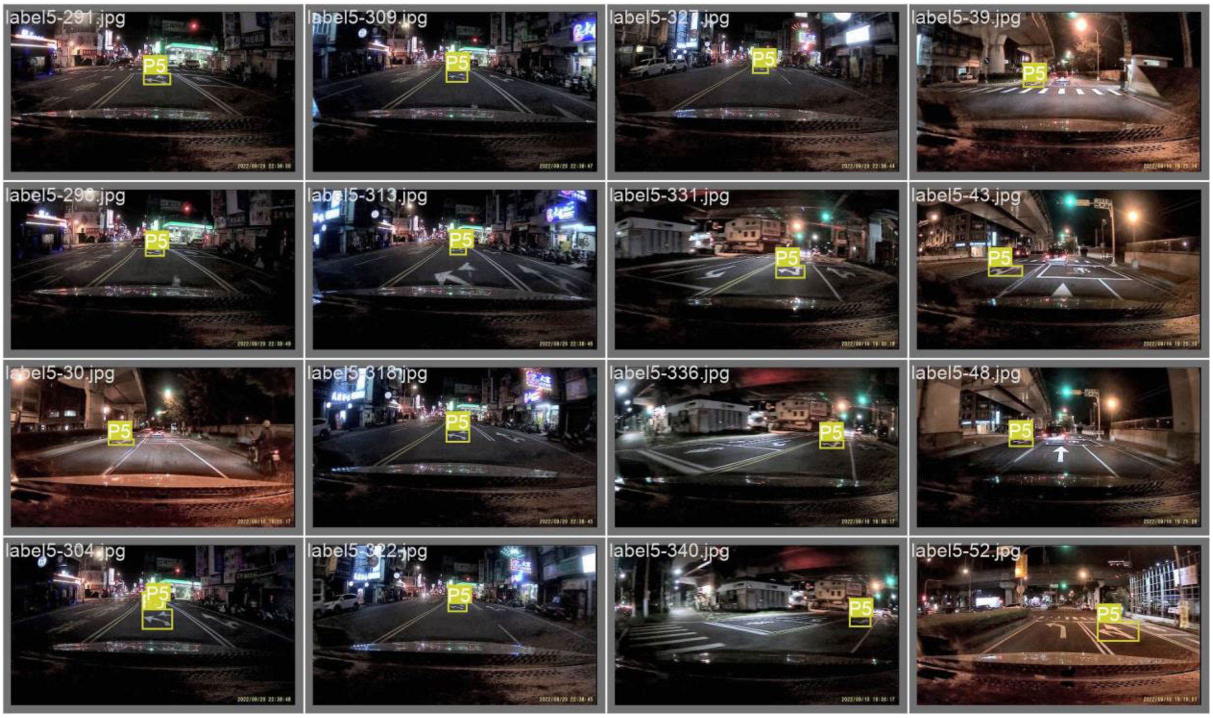


FIGURE 8. Validation process batch 0 with dataset 3.

TABLE 6. Comparison of the YOLO model results on TRMSDN with CLAHE during training.

MODEL	Training				
	Precision (%)	Recall (%)	F1- score (%)	IoU (%)	mAP (%)
YOLOv2	65	88	75	48.90	86.11
YOLOv3	77	82	79	59.19	87.73
YOLOv4	71	90	79	54.12	90.74
YOLO v4(No flip)	77	94	85	59.28	92.21
YOLOv4-csp	72	78	75	55.96	85.26
YOLOv4-csp (No Flip)	77	90	83	60.81	94.1
YOLOv4-tiny	62	78	69	47.07	78.17
YOLOv4-tiny (No Flip)	71	82	76	53.85	86.43
YOLOv4x-mish	72	86	78	56.01	87.2
YOLOv5n	73	91	81	NaN	89.2
YOLOv5s	75	90	82	NaN	89.1
YOLOv5m	74	91	82	NaN	87.5
YOLOv5l	74	92	82	NaN	90
YOLOv5x	76	93	84	NaN	89.8

model is exposed to a substantial quantity of labeled data, and it makes incremental adjustments to its internal parameters to reduce the number of errors in its prediction. The available data are typically partitioned into training and validation sets before the training process begins. The model is educated with the training set, while the validity set is used to assess how well it functions. YOLOv5 experiment during training batch 0 with dataset 3 is shown in Figure 7.

Validating a model entail evaluating it using a distinct data collection, the validation set. It is possible to assess how well the model will generalize to data that has not been observed before using the validation set. During the validation process, the predictions made by the model are compared to the actual values contained within the validation set, and various performance metrics such as accuracy and mean squared error are computed. Next, the validation process batch 0 with dataset 3 is described in Figure 8.

TABLE 7. Comparison of the YOLO model results on TRMSDN CLAHE during testing.

MODEL	Testing				
	Precision (%)	Recall (%)	F1- score (%)	IoU (%)	mAP (%)
YOLOv2	60	80	69	43.79	78.94
YOLOv3	70	74	72	52.07	80.11
YOLOv4	66	84	74	48.75	79.74
YOLO v4(No flip)	71	84	77	52.08	81.6
YOLOv4-csp	67	72	70	49.88	79.84
YOLOv4-csp (No Flip)	68	79	73	50.96	83.66
YOLOv4-tiny	58	73	65	43.80	79.13
YOLOv4-tiny (No Flip)	67	78	72	50.17	82.46
YOLOv4x-mish	67	78	72	49.62	79.99
YOLOv5n	70	85	77	NaN	82.6
YOLOv5s	72	86	78	NaN	83.7
YOLOv5m	70	90	79	NaN	85.1
YOLOv5l	71	89	79	NaN	86
YOLOv5x	73	89	80	NaN	86.4

TABLE 8. Training Result on TRMSDN with Image Enhancement.

MODEL	Training (%)			
	Ori	CS	HE	CLAHE
YOLOv2	86.26	83.91	85.27	86.11
YOLOv3	89.11	88.65	91.75	87.73
YOLOv4	87.36	88.37	88.78	90.74
YOLO v4(no flip)	91.54	89.42	88.25	92.21
YOLOv4-csp	87.45	89.18	89.2	85.26
YOLOv4-csp (No Flip)	89.12	90.15	89.42	94.1
YOLOv4-tiny	81.48	85.32	83.08	78.17
YOLOv4-tiny (No Flip)	88.91	90.71	90.41	86.43
YOLOv4x-mish	89.07	90.44	86.6	87.2
YOLOv5n	85.2	85	82.4	89.2
YOLOv5s	87	87.3	88.1	89.1
YOLOv5m	87.1	86.7	90.2	87.5
YOLOv5l	88.6	87	90.4	90
YOLOv5x	89.7	89.7	86.1	89.8
Average	87.707	87.989	87.854	88.110

Training performance result of YOLO models with TRMSDN Dataset and CLAHE Image shown in Table 6. Further, YOLOv4-csp (No Flip) achieves the highest performance with 94.1% mAP during training, followed by YOLOv4 (No Flip) with a mAP of 92.21%. For the YOLOv5 series, YOLOv5x got the maximum mAP of 89.8%. A comparison of the YOLO model during testing is exhibited in Table 7. YOLOv4-csp achieves the highest performance of 83.66% mAP. Furthermore, YOLOv5x got the maximum mAP of 86.4% in the YOLOv5 series.

V. DISCUSSIONS

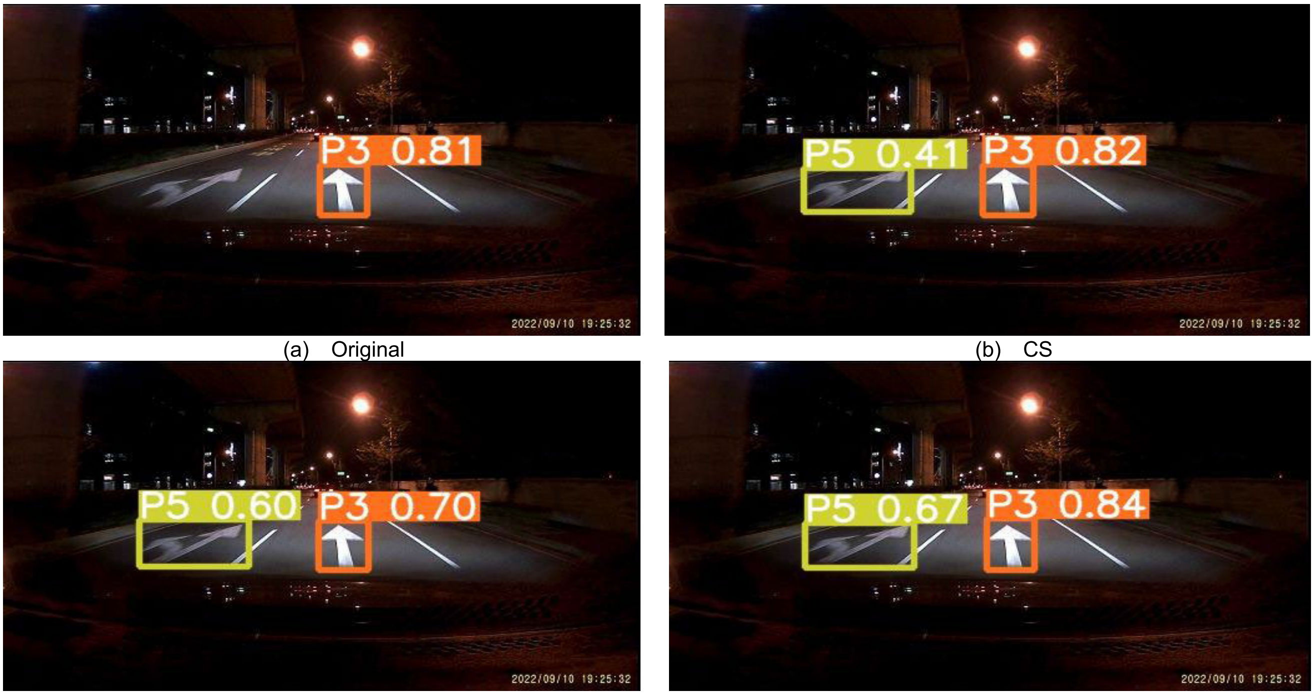
The enhanced images at night and the originals are compared in Table 8. Table 5 demonstrates how the values of the training set and testing set can be improved by employing image enhancement techniques, including CS, HE, and CLAHE. Further, CLAHE exhibits the best mAP during training, with an average mAP of 88.10%, followed by CS with 87.98%. Next, HE got an average mAP of 87.85%, and the original image got 87.707% of mAP.

Table 9 shows the testing result on TRSMDN with image enhancement. The experiment result shows a similar trend. Image enhancement methods can improve the performance of the original images. CLAHE exhibits the best average mAP with 82.09%, followed by HE with an mAP of 81.94%.

CLAHE is a popular image enhancement technique that offers several benefits, including: (1) Improved Contrast: CLAHE enhances the contrast of an image,

TABLE 9. Testing result on TRMSDN with image enhancement.

MODEL	Testing (%)			
	Ori	CS	HE	CLAHE
YOLOv2	75.79	77.55	76.24	78.94
YOLOv3	81.06	81.03	80.1	80.11
YOLOv4	81.67	81.15	80.59	79.74
YOLO v4(no flip)	85.15	80.67	85.22	81.6
YOLOv4-csp	80.98	80.25	81.05	79.84
YOLOv4-csp (No Flip)	82.6	83.87	83.31	83.66
YOLOv4-tiny	72.3	80.3	77.96	79.13
YOLOv4-tiny (No Flip)	85.71	86.21	83.99	82.46
YOLOv4x-mish	81.14	80.39	80.2	79.99
YOLOv5n	80.9	81.9	81.3	82.6
YOLOv5s	83	83	82.7	83.7
YOLOv5m	85.3	81.3	84.8	85.1
YOLOv5l	85.8	82.8	84.8	86
YOLOv5x	85.6	85.6	85	86.4
Average	81.928	81.858	81.947	82.090



(a) Original

(b) CS

(c) HE

(d) CLAHE

FIGURE 9. The recognition results using YOLOv5x.

making it easier to see details that may be difficult to discern in the original image. (2) Local Contrast Enhancement: CLAHE enhances the contrast of an image in small regions rather than globally, allowing it to preserve local details. (3) No Over enhancement: Unlike standard histogram equalization, CLAHE uses a limiting function to prevent over-enhancement of the image, which can result in noise and artifacts. (4) Customizable Parameters: CLAHE offers several parameters that can be adjusted to fine-tune the enhancement for a particular image, such as the block size and the clip limit. (5) Widely Used: CLAHE is a well-established technique widely used in various applications,

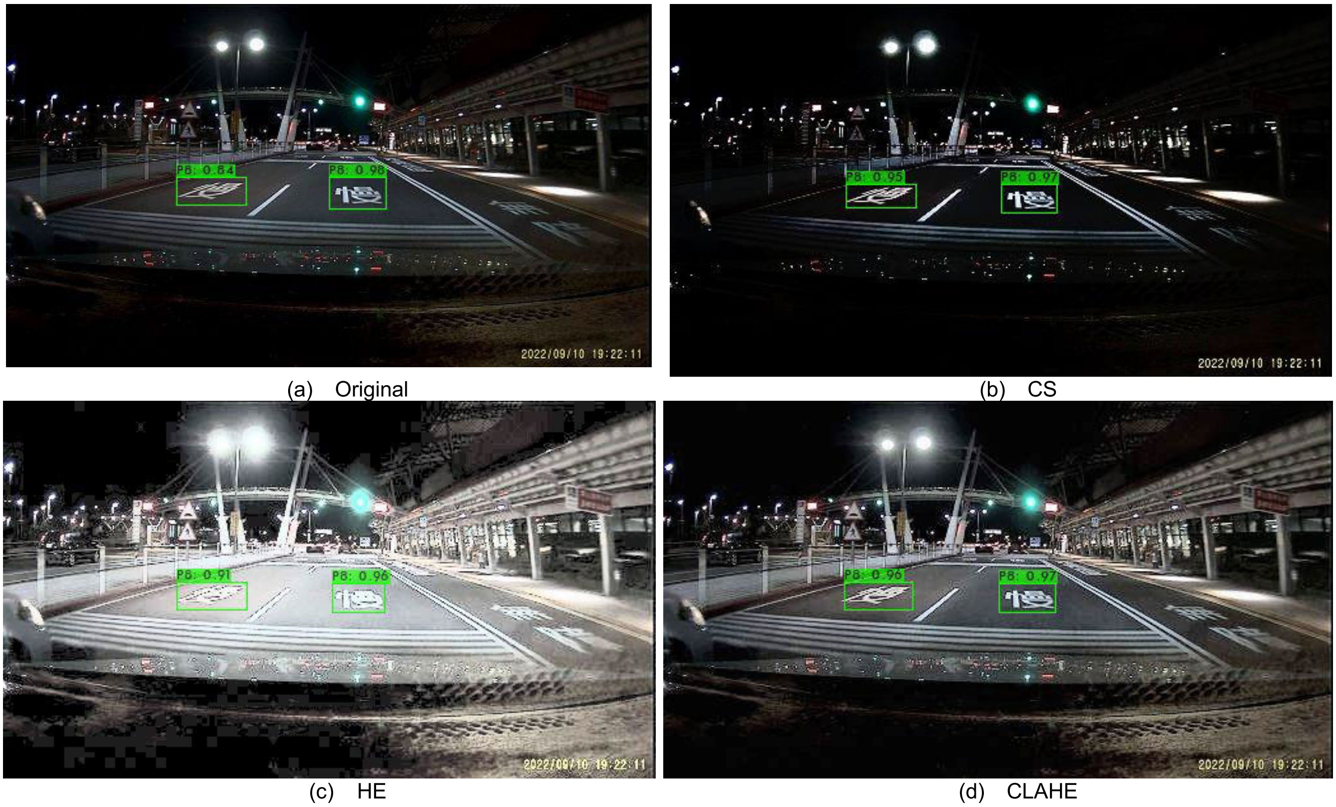


FIGURE 10. The recognition results using YOLOv4-csp (No Flip).

including medical imaging, remote sensing, and computer vision.

Overall, CLAHE can be a powerful technique for enhancing an image's contrast and visual quality while preserving local details and avoiding over-enhancement.

Figure 9 represents the result of YOLOv5x. Furthermore, the original image in Figure 9(a) cannot detect all the signs in the image. It only detects one class P3 with 70% accuracy. CLAHE result is shown in Figure 9(d) and exhibits the optimum accuracy for all classes. YOLOv5x can detect all the classes in the image very well, with an accuracy of 67% (Class P5) and 84% (Class P3).

Moreover, Figure 10 explains the recognition result using YOLOv4-csp (No Flip). As a result, we can conclude from the test results shown in Figure 10 that every model can correctly identify all classes despite differences in the bounding box coordinates and degrees of accuracy. CLAHE recognition results with YOLOv4-csp (No Flip) are shown in Figure 10(d) with 96% and 97% accuracy for class P8. The mAP may be reduced due to the direction category (such as turning left and right). In addition, In the test environment, the performance of the model may be affected by weather conditions (such as rain and fog).

The research only focuses on road sign detection in Taiwan and may not be comparable to other countries and regions because their road conditions and sign designs may differ. Further, only limited image enhancement techniques are considered; we can add more image processing methods in future works. The impact of weather conditions (such as rain and fog) on model performance is not considered. In real scenarios, weather conditions can significantly affect the visibility of road signs, and additional preprocessing steps may be required to improve the model's performance.

VI. CONCLUSION

This paper mainly discusses how the image enhancement method can improve the performance of the original image. Our work combines the original image with image enhancement methods such as CS, HE, and CLAHE. We use different numbers and sizes of images for training. Our work analyzes and examines CNN models combined with various backbone architectures and extractor features, specifically YOLOv2, YOLOv3, YOLOv4, and YOLOv5, for road marking recognition in the night. In this experiment, we examine key characteristics of the detector, such as its detection time, workspace size, and number of BFLOPs. Our results demonstrate that the road marking sign recognition performance can be improved by employing image enhancement of the original photos in the dataset for training.

Further, we drew the following conclusions from our experiments: (1) Experimentally, the best dataset is Dataset 4, which is the original image enhanced by CLAHE. (2) This research supports using CLAHE's YOLOv5x as the optimal model. (3) Increasing the noise during training will lengthen the training process and reduce the number of general errors.

Therefore, improving object identification performance can be accomplished by combining the dataset with CLAHE photos and the original pictures.

In the future, one of our goals is to improve our dataset in Taiwan, particularly by collecting data in a variety of environments than just at night. To highlight the benefits of image enhancement, we shall evaluate it compared to other road marking sign standards. In the future, researchers plan to investigate the combination of Explainable AI (XAI) and other picture enhancement methods with additional detection methods.

REFERENCES

[1] M. R. Bachute and J. M. Subhedar, ''Autonomous driving architectures: Insights of machine learning and deep learning algorithms,'' *Mach. Learn. Appl.*, vol. 6, Dec. 2021, Art. no. 100164, doi: 10.1016/j.mlwa.2021.100164.

[2] C. Dewi, R.-C. Chen, Y.-T. Liu, and H. Yu, ''Various generative adversarial networks model for synthetic prohibitory sign image generation,'' *Appl. Sci.*, vol. 11, no. 7, p. 2913, Mar. 2021.

[3] C. Dewi, A. P. S. Chen, and H. J. Christanto, ''Deep learning for highly accurate hand recognition based on YOLOv7 model,'' *Big Data Cognit. Comput.*, vol. 7, no. 1, p. 53, Mar. 2023, doi: 10.3390/bdcc7010053.

[4] C. Dewi, R.-C. Chen, Y.-C. Zhuang, X. Jiang, and H. Yu, ''Recognizing road surface traffic signs based on YOLO models considering image flips,'' *Big Data Cognit. Comput.*, vol. 7, no. 1, p. 54, Mar. 2023, doi: 10.3390/bdcc7010054.

[5] C. Dewi, R.-C. Chen, H. Yu, and X. Jiang, ''XAI for image captioning using SHAP,'' *J. Inf. Sci. Eng.*, vol. 39, pp. 711–724, Jul. 2023. [Online]. Available: https://www.airitilibrary.com/Publication/alDetailedMesh?docid=10162364-N202303030013-00001

[6] M. A. Farooq, W. Shariff, and P. Corcoran, ''Evaluation of thermal imaging on embedded GPU platforms for application in vehicular assistance systems,'' *IEEE Trans. Intell. Vehicles*, vol. 8, no. 2, pp. 1130–1144, Feb. 2023, doi: 10.1109/TIV.2022.3158094.

[7] V. A. Adibhatla, H.-C. Chih, C.-C. Hsu, J. Cheng, M. F. Abbod, and J.-S. Shieh, ''Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once,'' *Math. Biosci. Eng.*, vol. 18, no. 4, pp. 4411–4428, 2021, doi: 10.3934/mbe.2021223.

[8] J. Chen, K. Jia, W. Chen, Z. Lv, and R. Zhang, ''A real-time and high-precision method for small traffic-signs recognition,'' *Neural Comput. Appl.*, vol. 34, no. 3, pp. 2233–2245, Feb. 2022, doi: 10.1007/s00521-021-06526-1.

[9] C. Liu, Y. Wu, J. Liu, Z. Sun, and H. Xu, ''Insulator faults detection in aerial images from high-voltage transmission lines based on deep learning model,'' *Appl. Sci.*, vol. 11, no. 10, p. 4647, May 2021, doi: 10.3390/app11104647.

[10] K. M. Kwayu, M. Toth, and A. Himmelein, ''A scalable deep learning framework for extracting model inventory of roadway element intersection control types from panoramic images,'' *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2676, no. 5, pp. 630–642, May 2022, doi: 10.1177/03611981211069066.

[11] W. Yang and W. Zhang, ''Real-time traffic signs detection based on YOLO network model,'' in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery*, 2020, pp. 354–357, doi: 10.1109/CyberC49757.2020.00066.

[12] D. Mijic, M. Brisinello, M. Vranješ, and R. Grbic, ''Traffic sign detection using YOLOv3,'' in *Proc. IEEE 10th Int. Conf. Consum. Electron. (ICCE-Berlin)*, Nov. 2020, pp. 1–6, doi: 10.1109/ICCE-Berlin50680.2020.9352180.

[13] L. Gatelli, G. Gosmann, F. Fitarelli, G. Huth, A. A. Schwertner, R. De Azambuja, and V. J. Brusamarello, ''Counting, classifying and tracking vehicles routes at road intersections with YOLOv4 and DeepSORT,'' in *Proc. 5th Int. Symp. Instrum. Syst., Circuits Transducers*, 2021, pp. 1–6, doi: 10.1109/INSCIT49950.2021.9557244.

[14] W. N. Mohd-Isa, M. S. Abdullah, M. Sarzil, J. Abdullah, A. Ali, and N. Hashim, ''Detection of Malaysian traffic signs via modified YOLOv3 algorithm,'' in *Proc. Int. Conf. Data Analytics Bus. Ind., Way Towards Sustain. Economy*, 2020, pp. 1–5, doi: 10.1109/ICDABI51230.2020.9325690.

[15] X.-Y. Ye, D.-S. Hong, H.-H. Chen, P.-Y. Hsiao, and L.-C. Fu, ''A two-stage real-time YOLOv2-based road marking detector with lightweight spatial transformation-invariant classification,'' *Image Vis. Comput.*, vol. 102, Oct. 2020, Art. no. 103978, doi: 10.1016/j.imavis.2020.103978.

[16] S.-K. Tai, C. Dewi, R.-C. Chen, Y.-T. Liu, X. Jiang, and H. Yu, ''Deep learning for traffic sign recognition based on spatial pyramid pooling with scale analysis,'' *Appl. Sci.*, vol. 10, no. 19, p. 6997, Oct. 2020, doi: 10.3390/app10196997.

[17] C. Dewi, R.-C. Chen, and H. Yu, ''Weight analysis for various prohibitory sign detection and recognition using deep learning,'' *Multimedia Tools Appl.*, vol. 79, nos. 43–44, pp. 32897–32915, Nov. 2020, doi: 10.1007/s11042-020-09509-x.

[18] C. Dewi, R.-C. Chen, Y.-T. Liu, X. Jiang, and K. D. Hartomo, ''YOLOv4 for advanced traffic sign recognition with synthetic training data generated by various GAN,'' *IEEE Access*, vol. 9, pp. 97228–97242, 2021, doi: 10.1109/ACCESS.2021.3094201.

[19] Z. Chen, K. Pawar, M. Ekanayake, C. Pain, S. Zhong, and G. F. Egan, ''Deep learning for image enhancement and correction in magnetic resonance imaging—State-of-the-art and challenges,'' *J. Digit. Imag.*, vol. 36, no. 1, pp. 204–230, Nov. 2022, doi: 10.1007/s10278-022-00721-9.

[20] Y. Jiang, L. Li, J. Zhu, Y. Xue, and H. Ma, ''DEANet: Decomposition enhancement and adjustment network for low-light image enhancement,'' *Tsinghua Sci. Technol.*, vol. 28, no. 4, pp. 743–753, Aug. 2023, doi: 10.26599/TST.2022.9010047.

[21] R. Rosnelly, B. S. Riza, L. Wahyuni, S. Suparni, A. Prasetio, and R. Rahim, ''Improvement of hybrid image enhancement for detection and classification of malaria disease types and stages with artificial intelligence,'' *TEM J.*, vol. 11, pp. 535–542, May 2022, doi: 10.18421/TEM112-06.

[22] P. Civicioglu and E. Besdok, ''Contrast stretching based pansharpening by using weighted differential evolution algorithm,'' *Exp. Syst. Appl.*, vol. 208, Dec. 2022, Art. no. 118144, doi: 10.1016/j.eswa.2022.118144.

[23] A. Asokan, D. E. Popescu, J. Anitha, and D. J. Hemanth, ''Bat algorithm based non-linear contrast stretching for satellite image enhancement,'' *Geosciences*, vol. 10, no. 2, p. 78, Feb. 2020, doi: 10.3390/geosciences10020078.

[24] S. Agrawal, R. Panda, P. K. Mishro, and A. Abraham, ''A novel joint histogram equalization based image contrast enhancement,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1172–1182, Apr. 2022, doi: 10.1016/j.jksuci.2019.05.010.

[25] B. S. Rao, ''Dynamic histogram equalization for contrast enhancement for digital images,'' *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106114, doi: 10.1016/j.asoc.2020.106114.

[26] S. Doshvarpassand, X. Wang, and X. Zhao, ''Sub-surface metal loss defect detection using cold thermography and dynamic reference reconstruction (DRR),'' *Struct. Health Monitor.*, vol. 21, no. 2, pp. 354–369, Mar. 2022, doi: 10.1177/1475921721999599.

[27] S. H. Majeed and N. A. M. Isa, ''Adaptive entropy index histogram equalization for poor contrast images,'' *IEEE Access*, vol. 9, pp. 6402–6437, 2021, doi: 10.1109/ACCESS.2020.3048148.

[28] J. Murugachandravel and S. Anand, ''Enhancing MRI brain images using contourlet transform and adaptive histogram equalization,'' *J. Med. Imag. Health Informat.*, vol. 11, no. 12, pp. 3024–3027, Dec. 2021, doi: 10.1166/jmihi.2021.3906.

[29] S. F. M. Radzi, M. K. A. Karim, M. I. Saripan, M. A. A. Rahman, N. H. Osman, E. Z. Dalah, and N. M. Noor, ''Impact of image contrast enhancement on stability of radiomics feature quantification on a 2D mammogram radiograph,'' *IEEE Access*, vol. 8, pp. 127720–127731, 2020, doi: 10.1109/ACCESS.2020.3008927.

[30] U. Kuran and E. C. Kuran, ''Parameter selection for CLAHE using multi-objective cuckoo search algorithm for image contrast enhancement,'' *Intell. Syst. with Appl.*, vol. 12, Nov. 2021, Art. no. 200051, doi: 10.1016/j.iswa.2021.200051.

[31] M. R. Islam and M. Nahiduzzaman, ''Complex features extraction with deep learning model for the detection of COVID19 from CT scan images using ensemble based machine learning approach,'' *Exp. Syst. Appl.*, vol. 195, Jun. 2022, Art. no. 116554, doi: 10.1016/j.eswa.2022.116554.

[32] J. Redmon and A. Farhadi, ''YOLOv3: An incremental improvement,'' 2018, *arXiv:1804.02767*.

[33] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ''You only look once: Unified, real-time object detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[34] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[35] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of freebies for training object detection neural networks," 2019, *arXiv:1902.04103*.

[36] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031, doi: 10.1109/ICCV.2019.00612.

[37] G. Ghiasi, T. Y. Lin, and Q. V. Le, "DropBlock: A regularization method for convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10727–10737.

[38] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–16.

[39] S. Ren, K. He, and R. Girshick, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99, doi: 10.1016/j.nima.2015.05.028.

[40] Q. Xu, R. Lin, H. Yue, H. Huang, Y. Yang, and Z. Yao, "Research on small target detection in driving scenarios based on improved YOLO network," *IEEE Access*, vol. 8, pp. 27574–27583, 2020, doi: 10.1109/ACCESS.2020.2966328.

[41] H. Chen, Z. He, B. Shi, and T. Zhong, "Research on recognition method of electrical components based on YOLOv3," *IEEE Access*, vol. 7, pp. 157818–157829, 2019, doi: 10.1109/ACCESS.2019.2950053.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[43] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[44] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[45] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.

[46] (2019). *Bbox Label Tool*. [Online]. Available: https://github.com/puzzledqs/BBox-Label-Tool

[47] Á. Arcos-García, J. A. Álvarez-García, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems," *Neurocomputing*, vol. 316, pp. 332–344, Nov. 2018, doi: 10.1016/j.neucom.2018.08.009.

[48] C. Dewi, R. C. Chen, and H. Te Hung, "Comparative analysis of restricted Boltzmann machine models for image classification," in *Proc. Asian Conf. Intell. Inf. Database Syst.*, vol. 12034, 2020, pp. 285–296, doi: 10.1007/978-3-030-42058-1_24.

[49] H. Yang, L. Chen, M. Chen, Z. Ma, F. Deng, M. Li, and X. Li, "Tender tea shoots recognition and positioning for picking robot using improved YOLO-V3 model," *IEEE Access*, vol. 7, pp. 180998–181011, 2019, doi: 10.1109/ACCESS.2019.2958614.

[50] Y. Yuan, Z. Xiong, and Q. Wang, "An incremental framework for video-based traffic sign detection, tracking, and recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1918–1929, Jul. 2017, doi: 10.1109/TITS.2016.2614548.

[51] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Comput. Electron. Agricult.*, vol. 157, pp. 417–426, Feb. 2019, doi: 10.1016/j.compag.2019.01.012.

[52] R. Shi, T. Li, and Y. Yamaguchi, "An attribution-based pruning method for real-time mango detection with YOLO network," *Comput. Electron. Agric.*, vol. 169, Feb. 2020, Art. no. 105214, doi: 10.1016/j.compag.2020.105214.

[53] H. Kang and C. Chen, "Fast implementation of real-time fruit detection in apple orchards using deep learning," *Comput. Electron. Agricult.*, vol. 168, pp. 1–10, Jan. 2020, doi: 10.1016/j.compag.2019.105108.

[54] A. Persson, H. Jonasson, I. Fredriksson, U. Wiklund, and C. Ahlström, "Heart rate variability for classification of alert versus sleep deprived drivers in real road driving conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3316–3325, Jun. 2021, doi: 10.1109/TITS.2020.2981941.

RUNG-CHING CHEN (Member, IEEE) received the B.S. degree from the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, in 1987, the M.S. degree from the Institute of Computer Engineering, National Taiwan University of Science and Technology, in 1990, and the Ph.D. degree in computer science from the Department of Applied Mathematics, National Chung Hsing University, in 1998. He is currently a Distinguished Professor with the Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan. He is listed in the Top 2% Scientists Worldwide in A.I. by Stanford University. His research concerns include network technology, pattern recognition, knowledge engineering, the Internet of Things, data analysis, and artificial intelligence.

CHRISTINE DEWI received the B.S. degree (S.Kom.) from the Informatics Engineering Study Program, Faculty of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia, in 2010, and the M.Cs. degree from the Master of Information Systems Study Program, Faculty of Information Technology, Satya Wacana Christian University, in 2012. In 2021, she finished her Ph.D. and was a Postdoctoral Researcher with the College of Informatics, Chaoyang University of Technology, Taiwan, until 2022. She is currently an Assistant Professor with the Department of Information Technology and the Head of the Associate Degree Informatics Engineering Study Program at Satya Wacana Christian University. Her current research interests include image processing, computer vision, object detection and recognition, artificial intelligence, and machine learning. In 2022, she got the Author With the Highest Sinta Score and Research Publications in three years award from Satya Wacana Christian University. She received the Excellent Dissertation Award from the Taiwanese Association for Consumer Electronics (TACE), Taiwan, in 2021. This award aims to honor a dissertation project funded by MOST Taiwan. The participants were all students from various universities in Taiwan. She is active as a reviewer and guest editor in many journals.

YONG-CUN ZHUANG received the B.I.M. degree from the Department of Information Management, Chaoyang University of Technology, in 2021, where he is currently pursuing the master's degree. His research focuses on object detection and recognition, artificial intelligence, image processing, and machine learning.

JEANG-KUO CHEN received the B.S. degree from the Department of Information Engineering, Tatung Institute of Technology, Taipei, Taiwan, in 1985, the M.S. degree from the Institute of Information Engineering, Tatung Institute of Technology, in 1989, and the Ph.D. degree from the Department of Information Engineering, National Tsinghua University, Hsinchu, Taiwan, in 1997. He is currently an Assistant Professor with the Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan. His research interests include database design, concurrency control, XML applications, big data, and NoSQL databases.