applied sciences

MDPI

Article

Edge-AI-Based Real-Time Automated License Plate Recognition System

Cheng-Jian Lin [1,2,*], Chen-Chia Chuang [1] and Hsueh-Yi Lin [1]

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan; 3a617090@gmail.com (C.-C.C.); lhy@ncut.edu.tw (H.-Y.L.)
[2] College of Intelligence, National Taichung University of Science and Technology, Taichung 404, Taiwan
* Correspondence: cjlin@ncut.edu.tw

Abstract: The rapid development of urban intelligence has turned intelligent transport system (ITS) development into a primary goal of traffic management. Automated license plate recognition (ALPR) for moving vehicles is a core aspect of ITS. Most ALPR systems send images back to a server for license plate recognition. To reduce delays and bandwidth use during image transmission, this study proposes an edge-AI-based real-time ALPR (ER-ALPR) system, in which an AGX XAVIER embedded system is embedded on the edge of a camera to achieve real-time image input to an AGX edge device and to enable real-time automatic license plate character recognition. To assess license plate characters and styles in a realistic setting, the proposed ER-ALPR system applies the following approaches: (1) image preprocessing; (2) You Only Look Once v4-Tiny (YOLOv4-Tiny) for license plate frame detection; (3) virtual judgment line for determining whether a license plate frame has passed; (4) the proposed modified YOLOv4 (M-YOLOv4) for license plate character recognition; and (5) a logic auxiliary judgment system for improving license plate recognition accuracy. We tested the proposed ER-ALPR system in selected real-life test environments in Taiwan. In experiments, the proposed ER-ALPR system achieved license plate character recognition rates of 97% and 95% in the day and at night, respectively. Through the AGX system, the proposed ER-ALPR system achieves a high recognition rate at a low computational cost.

Keywords: license plate recognition; deep learning; embedded system; You Only Look Once (YOLO)

check for updates

Citation: Lin, C.-J.; Chuang, C.-C.; Lin, H.-Y. Edge-AI-Based Real-Time Automated License Plate Recognition System. *Appl. Sci.* **2022**, *12*, 1445. https://doi.org/10.3390/app12031445

Academic Editors: Teen-Hang Meen and Chun-Yen Chang

Received: 13 December 2021
Accepted: 27 January 2022
Published: 28 January 2022

1. Introduction

Governments worldwide are rushing to integrate artificial intelligence (AI) technology into urban management. In particular, large cities have many cars on their roads, and vehicle management automated license plate recognition (ALPR) systems have been developed to manage this traffic. Many ALPR systems incorporate information technology in urban settings; these technologies include interval speed tracking [1,2], unmanned parking lot management [3], and roadside illegal traffic enforcement [4].

Existing ALPR systems can be divided into two main categories, namely multi- and single-stage methods. Most existing ALPR systems employ the multi-stage method, which is divided into three main stages as follows: (1) location of license plate, (2) character segmentation, and (3) identification of license plate characters [5]. Each step employs different image processing or machine learning methods. A single-stage method uses deep learning technology to locate and recognize a license plate. In a traditional ALPR system, the multi-stage method uses techniques such as binarization [6], edge detection, character detection, texture detection, color detection, template matching technology, and feature extraction [7]. However, the aforementioned techniques are easily affected by environmental factors such as license plate tilt, blurring, and lighting-related changes [8]. Compared with the multi-stage method, the single-stage method is less affected by the environment and provides better recognition accuracy.

ALPR systems usually use image binarization or grayscale to locate candidate license plates and characters, after which manual methods are applied for feature extraction and traditional machine learning methods are employed for character classification [7]. With the recent development of deep learning, numerous studies have adopted the convolutional neural network (CNN) for license plate recognition because of its high accuracy with respect to general object detection and recognition [9,10]. Selmi et al. [11] used a series of preprocessing methods (based on morphological operators, Gaussian filtering, edge detection, and geometrical analysis) to locate license plates and characters. Subsequently, they used two different CNNs to: (1) group a set of license plate candidates for each image into a single positive sample; and (2) recognize license plate characters. In this method, a single license plate is processed for each image, and the resulting distortion of a license plate is primarily caused by poor lighting conditions. Laroca et al. [12] proposed the use of a CNN for training and fine-tuning each ALPR stage, such that the system remains robust even under dynamic conditions (e.g., different cameras, lighting, and backgrounds); in particular, this strategy allows for a good recognition rate to be obtained in character segmentation and recognition.

License plate styles, sorting rules, and compositions of license plate characters differ between countries. When real-time image recognition is applied to traffic, blurred images due to vehicle movement are captured. Generally, public data sets mostly contain static images of parking lots; therefore, they cannot be used for training. Variations in style and weather increase the difficulty of collecting data. Han et al. [4] used a generative adversarial network to generate multiple license plate images. Their You Only Look Once V2 (YOLOv2) [13] performed well on a generated synthetic data set. To address the challenges posed by changing scenery and differently styled license plates, the recognition accuracy of image preprocessing technology should be improved. In machine vision, which is a character recognition technology, comparisons with color or grayscale images are conducted to enable the binarization method to more easily separate the pixels belonging to license plate characters in an image. However, even with an enhanced image function, a single threshold is still difficult to obtain. To solve this problem, Niblack [14] proposed the application of an adaptive binarization technology.

Most ALPR systems still use network connection servers or cloud infrastructure for image recognition. Edge AI involves the installation of embedded systems directly on the edge to avoid the use of complex centralized AI systems [15], such as Raspberry Pi Pico (RPI), Jetson AGX Xavier (AGX), and JETSON TX2. Compared with methods that involve the sending of images back to a server or the cloud, Edge AI saves many images that occupy bandwidth during transmission, reduces image transmission delays [16], and increases capability in terms of frames-per-second (FPS) during execution. When a computer is installed on the edge, it can also directly connect to a camera without using a network. An embedded system is cost-efficient, easy to disassemble and assemble, and small in size.

For real-time images, dozens of frames are combined every second. The key task is to determine the correlation between a frame and subsequent frames. Wojke et al. [17] proposed a simple, online, and real-time tracking (SORT) algorithm that uses the Hungarian algorithm to perform Kalman filtering frame by frame in image space. In the SORT algorithm, Intersection over Union (IoU) is used as the basis for association. Objects are numbered and labeled to enable the tracking of objects. Different solutions are applied based on the given situation. In a study on road real-time image license plate recognition, the reduction of recognition time and increase of model recognition rate are highly crucial points that designers aim to improve on [18,19].

In the present study, an edge-AI-based real-time automated license plate recognition (ER-ALPR) system is proposed. This system incorporates an AGX XAVIER embedded system that is installed on the edge of a camera to achieve real-time image input to an AGX edge device and to enable real-time automatic license plate character recognition. The major contributions of this study are as follows:

(1) An embedded system is embedded on the edge of a camera to enable real-time automatic license plate character recognition.
(2) You Only Look Once v4-Tiny (YOLOv4-Tiny) is used to detect the license plate frame of a car.
(3) A virtual judgment line is used to determine whether a license plate frame has passed the judgment line.
(4) A modified YOLOv4 (M-YOLOv4) for recognizing license plate characters is proposed.
(5) A logic auxiliary judgment system for improving the accuracy of license plate recognition is proposed.

The present study is organized as follows. Section 2 describes the proposed approach, Section 3 discusses the experimental results, and Section 4 provides the conclusions and future research directions.

2. Materials and Methods

Most ALPR systems return images to a server for license plate character recognition. To reduce delays and bandwidth use during image transmission, an ER-ALPR system is proposed in this study. This section introduces the flow of the ER-ALPR system and its related hardware. In addition, the system's license plate detection approach, license plate character recognition approach, and logic auxiliary judgment system are also described in this section.

2.1. Flow of Proposed ER-ALPR System

A flow chart of the proposed ER-ALPR system is presented in Figure 1. Specifically, the ER-ALPR system first sets the virtual license plate judgment line and subsequently acquires images from the camera then performs license plate recognition after image processing in each frame. Next, the latest license plate distance change is calculated to determine whether a license plate has passed the judgment line. Once a license plate passes the judgment line, the entire image is used for license plate character recognition, and only the license plate character recognition results obtained in the license plate frame position are used. Thereafter, the license plate character recognition results are used to determine whether the license plate rules are consistent. In addition, the auxiliary judgment of character logic is used to verify whether a license plate meets the standard formats. Finally, the recognition results are obtained. Moreover, YOLOv4-Tiny model as a one-stage model is able to detect multiple objects at a time which means it is possible to detect multiple license plate frames in one image. Furthermore, a virtual judgment line checks whether a license plate frame has passed a specific area which means license plate character recognition can be performed in order and will not reidentify the license plate.

2.2. Hardware Specification of Edge Embedded System

In the present study, JETSON AGX XAVIER is used as the edge embedded system because of its small size and energy-saving feature. It also provides a greater temperature range for applications at the edge. The system was run on a 100 × 87 mm^2 computer powered by a 512-Core NVIDIA Volta GPU with Tensor Cores and an 8-Core ARM v8.2 64-Bit CPU with 8 MB of L2 cache and 4 MB of L3 cache (Nvidia Corporation, CA, USA). The development kit uses low-power double-data-rate synchronous dynamic random access memory (DDR SDRAM, Samsung Corporation, Seoul, Korea) and runs ubuntu18.04 as the operating system (Canonical Ltd., London, UK). The camera supports a maximum resolution of 1920 × 1080, is equipped with auto focus functionality, and has a maximum frame rate of 60 FPS. Infrared mode is used at night. In addition, a night license plate auxiliary light is applied to ensure that a license plate is sufficiently bright in a night image.

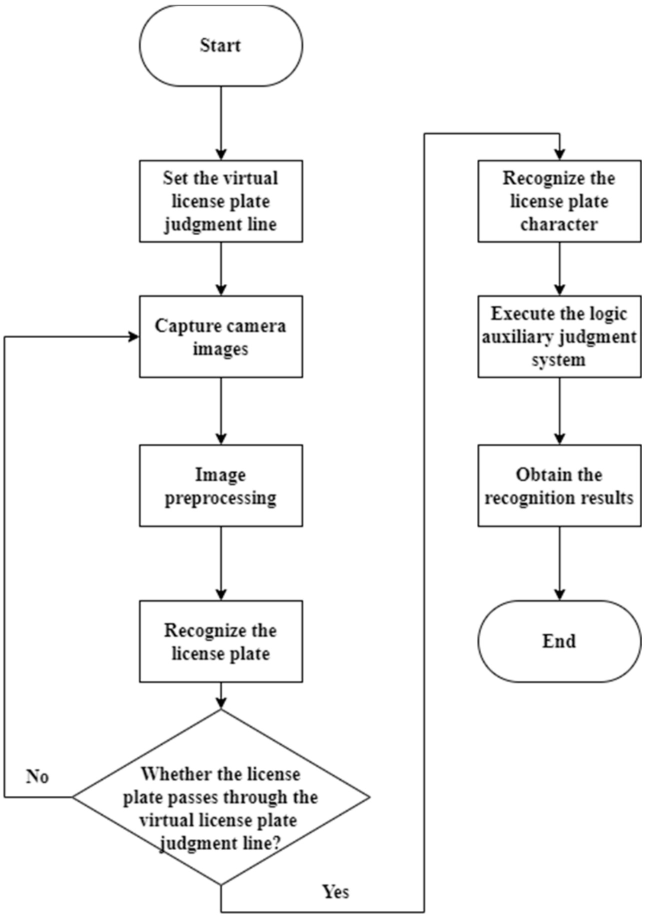
Start
Set the virtual license plate judgment line
Capture camera images
Image preprocessing
Recognize the license plate
Whether the license plate passes through the virtual license plate judgment line?
No
Yes
Recognize the license plate character
Execute the logic auxiliary judgment system
Obtain the recognition results
End

Figure 1. Proposed ER-ALPR system.

2.3. Image Preprocessing

The rules governing license plates differ between countries. In the present study, we tested our system by implementing it on roads and license plates in Taiwan. Taiwan's car license plates come in one of six styles: black characters on a white background, green characters on a white background, red characters on a white background, black characters on a yellow background, white characters on a green background, and white characters on a red background. A license plate contains Latin letters (A to Z), Arabic numerals (0 to 9), and the hyphen (-). Among the rules for these six styles, the rules for white characters on a green background and white characters on a red background are almost identical; therefore, we applied image preprocessing to a license plate in the manner as demonstrated in Figure 2. First, an original red–green–blue (i.e., RGB) image (Figure 2a) is converted into an International Commission on Illumination XYZ (CIE XYZ) image [20] and, subsequently, into a blue–green–red (BGR) image (Figure 2b). Next, the BGR image is converted into a grayscale image (Figure 2c). This step converts the colors green and red to black and the colors yellow and white to white. Therefore, through the preprocessing methods, the six license plate formats are converted into two license plate styles, namely black characters on a white background (originally black characters on a white background, green characters on a white background, or red characters on a white background) and white characters on a black background (originally white characters on a green background or white characters on a red background). This strategy addresses the difficulty associated with collecting license plate data sets for an excessive number of license plate color patterns.

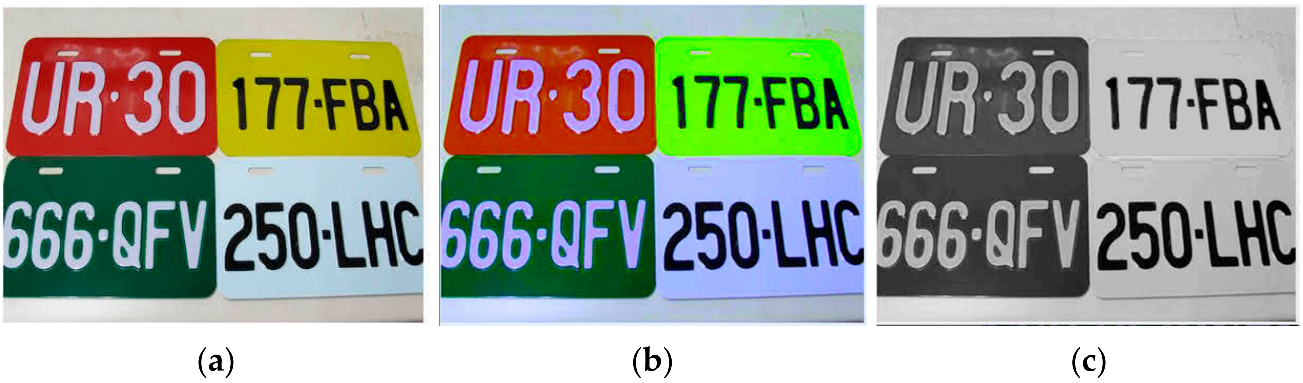

Figure 2. Image preprocessing. (**a**) Original image. (**b**) Blue–green–red image. (**c**) Grayscale image.

2.4. License Plate Frame Detection and License Plate Character Recognition

This subsection describes the use of YOLOv4-Tiny to detect the license plate frame of a car and the use of a virtual judgment line to determine whether a license plate frame has passed a specific area. An M-YOLOv4 is proposed for recognizing license plate characters. To effectively obtain the characteristic parameters of license plate characters, the present study uses the proposed M-YOLOv4 to recognize license plates. YOLOv4 [21] does not have a fully connected layer, and it is compatible with input images of all sizes. Furthermore, it uses a convolution layer instead of a pooling layer and performs up sampling. Therefore, during the training process, YOLOv4 can effectively prevent information loss. In real-time images, each frame of YOLOv4 is more stable relative to that of YOLOv3; hence, recognition frames are not affected by flickering.

2.4.1. License Plate Frame Detection through YOLOv4-Tiny

The colors of license plates in Taiwan represent different car categories and different number arrangements which are able to assist rule judgment; therefore, positional information of a license plate frame is provided by integrating color features into the proposed system. After the completion of image processing, license plates either contain black characters on a white background or white characters on a black background (Figure 2c). A license plate is detected through YOLOv4-Tiny (Figure 3), which is a lightweight network and a simplified version of YOLOv4. YOLOv4-Tiny uses convolutional layers and maximum pooling layers to extract the features of objects; it then uses upsampling and concat layers to fuse the features of the previous layer with the aim of expanding feature information. Subsequently, the license plate detection effect is further improved. In an AGX system, YOLOv4 is executed at a speed of 0.06 to 0.09 s/frame, whereas YOLOv4-Tiny is executed at a speed of 0.02 to 0.03 s/frame. In other words, YOLOv4-Tiny is three times as fast as YOLOv4 and achieves an FPS of approximately 36. When a license plate frame passes the virtual license plate judgment line, license plate character recognition is performed more slowly but more accurately.

2.4.2. Virtual Judgment Line of License Plate Frame

The correlation between the front and rear images of a real-time image is crucial. The purpose of the virtual judgment line is to determine whether images should be captured for license plate character recognition. Figure 4 illustrates how the distance between the center point of a license plate frame and the virtual judgment line is calculated. If the center point of the license plate frame has not passed and has passed the virtual judgment line, it is represented by the "−" and "+" symbols, respectively. To prevent repeated identification and the recording of the same license plate in each frame, the virtual judgment line allows for only one license plate to be processed at a time.

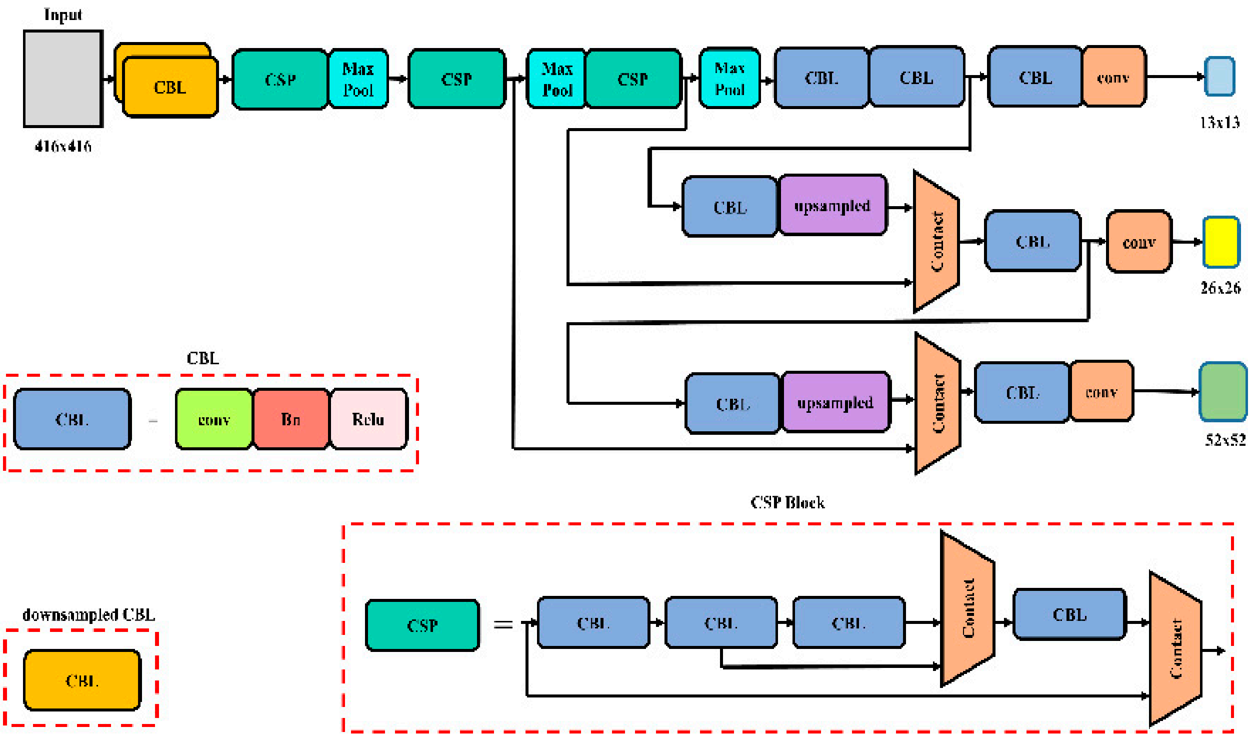

Figure 3. YOLOv4-Tiny model. The red rectangular represents several modules commonly used in YOLOv4.

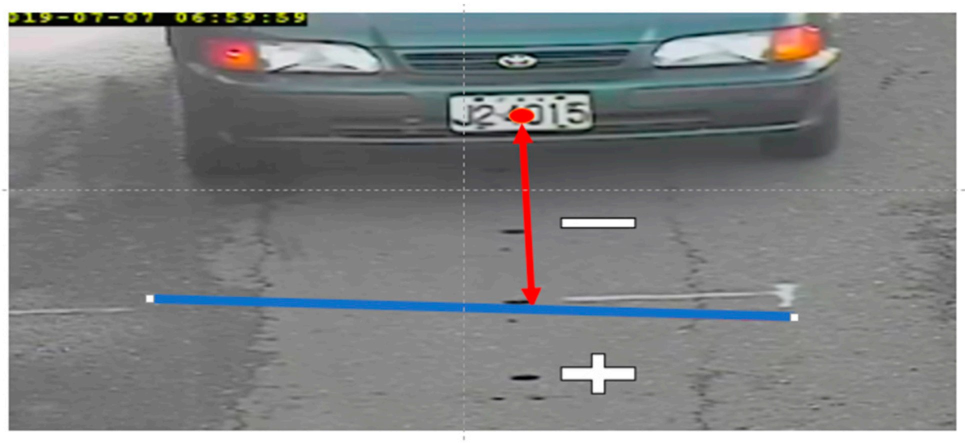

Figure 4. Distance (red line) between center point of license plate frame and virtual judgment line (blue line).

2.4.3. License Plate Character Recognition through Proposed M-YOLOv4

In deep learning, small and large objects are defined based on the proportions of an object in an image rather than on the size of the object in an image. To avoid any misjudgment of license plate characters or the inclusion of erroneous judgments in the recognition results, two approaches are applied. The first approach retrieves the license plate frame as an image and subsequent application of character recognition for the license plate; the second approach involves the identification of a license plate frame and direct implementation of license plate recognition for a full image. In the first approach, the cut license plate frame turns the characters in the license plate into a large object, which cannot be converged during training through YOLOv4. Therefore, the present study applied the second approach to select only characters within the license plate frame in the full image for recognition. With this strategy, only the characters within a license plate frame are selected for identification and the proportions of the license plate characters in the image are small. The original YOLOv4 allows for the recognition of license plate characters. However, the characteristics of small objects are lost after the multilayer convolution of the backbone layer is performed. Therefore, in the system of the present study, the original YOLOv4 is modified to improve the recognition of license plate characters. As illustrated in Figure 5, to avoid losing the characteristics of small objects through multiple cross-stage-partial (CSP) layers during training, all Conv + BN + Leaky_relu (CBL) blocks are moved up to form connections with the CSP layer and the final CSP convolutional layer of the original YOLOv4 backbone layer is removed (Figure 5 gray background area). CBL is the smallest component in the YOLOv4 network structure. This modified YOLOv4 model is called M-YOLOv4, which performs well in the recognition of license plate characters and is executed as fast as YOLOv4.

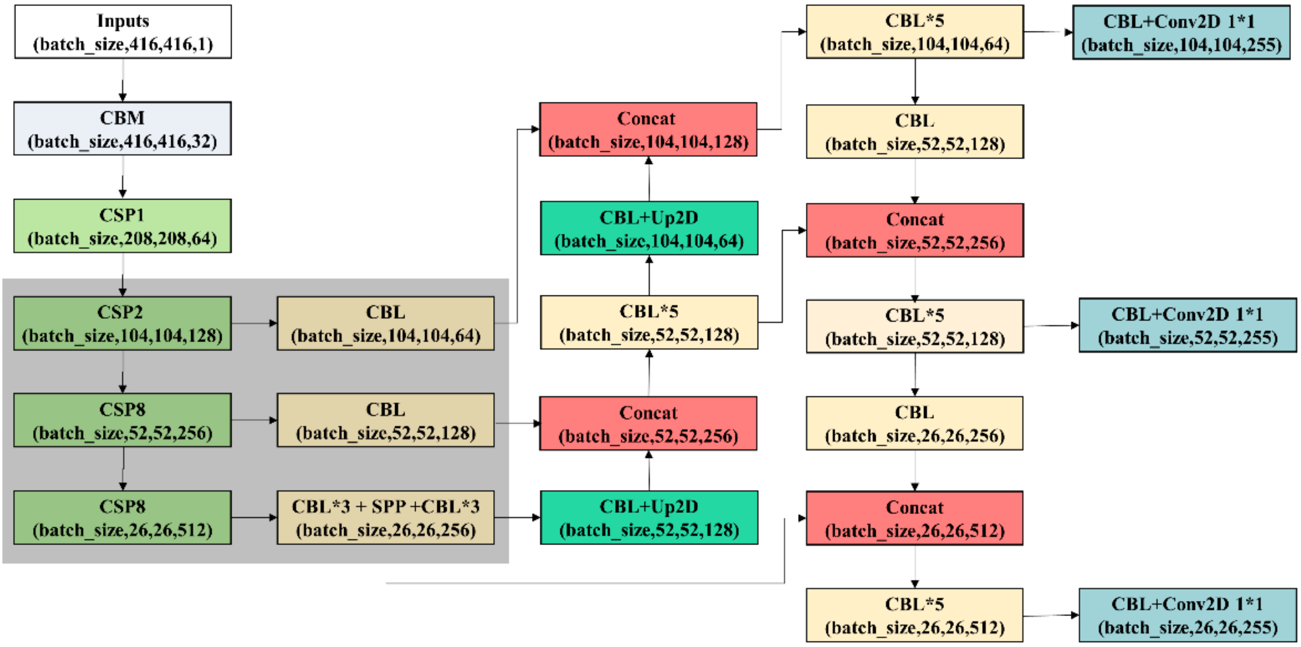
Inputs (batch_size,416,416,1)
CBM (batch_size,416,416,32)
CSP1 (batch_size,208,208,64)
CSP2 (batch_size,104,104,128)
CBL (batch_size,104,104,64)
CSP8 (batch_size,52,52,256)
CBL (batch_size,52,52,128)
CSP8 (batch_size,26,26,512)
CBL*3 + SPP +CBL*3 (batch_size,26,26,256)
Concat (batch_size,104,104,128)
CBL+Up2D (batch_size,104,104,64)
CBL*5 (batch_size,52,52,128)
Concat (batch_size,52,52,256)
CBL+Up2D (batch_size,52,52,128)
CBL*5 (batch_size,104,104,64)
CBL+Conv2D 1*1 (batch_size,104,104,255)
CBL (batch_size,52,52,128)
Concat (batch_size,52,52,256)
CBL*5 (batch_size,52,52,128)
CBL+Conv2D 1*1 (batch_size,52,52,255)
CBL (batch_size,26,26,256)
Concat (batch_size,26,26,512)
CBL*5 (batch_size,26,26,512)
CBL+Conv2D 1*1 (batch_size,26,26,255)

Figure 5. Proposed M-YOLOv4 model. (a*b means that the number of a is b).

The proposed M-YOLOv4 (Figure 5) comprises several blocks (Figure 6). Conv refers to a convolution layer that serves to slide the convolution kernel over the input image to obtain a feature map. Bn refers to batch normalization, and Mish denotes the activation function. Compared with the commonly used ReLU, Mish has a smoother function and superior generalization ability. Leaky ReLU is an improved activation function that allows ReLU to solve the problem of a gradient disappearing when the input is negative. ADD is the addition of a tensor, and a dimension is not expanded after an addition. Concat is the splicing of a tensor, and a dimension is expanded after splicing. CBM is the smallest component in a YOLOv4 structure, and it comprises Conv, Bn, and Mish. CBL consists of Conv, BN, and Leaky ReLU. Res unit imitates a ResNet model architecture, which comprises CBM, CBM, and ADD, to enable the deeper construction of the model. CSPX imitates the CSPNet model structure, which comprises CBM, Res unit, and Concat.

In addition, for the proposed M-YOLOv4 training process, Distance-based IoU–Non-Maximum Suppression (DIoU-NMS) is adopted. DIoU-NMS not only considers the IoU between two object frames but also the distance between the center points of two object frames. The original YOLOv4 produces numerous overlapping recognition frames. Because license plate characters do not overlap, the DIoU-NMS used in the present study is reduced to suppress the overlapping of multiple recognition frames.

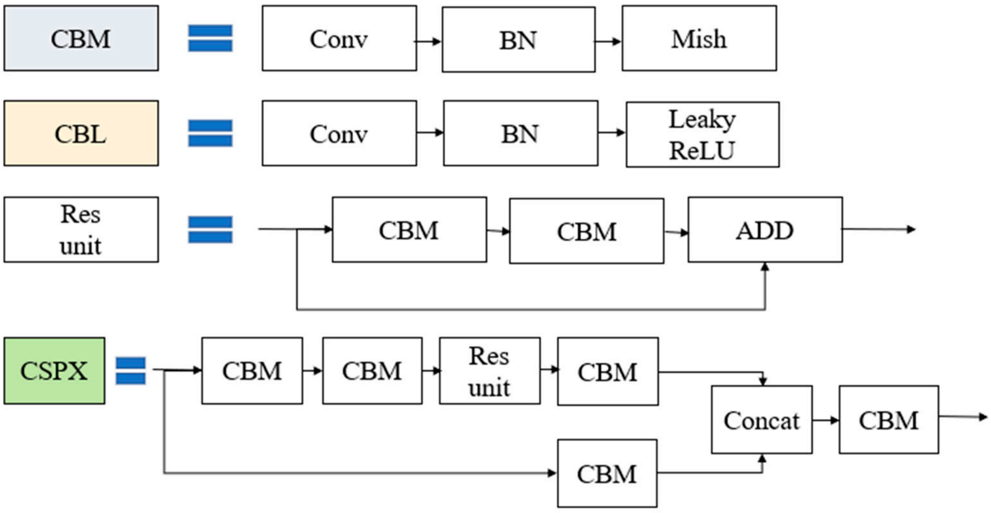

Figure 6. Internal block composition of proposed M-YOLOv4.

2.5. Logic Auxiliary Judgment System

When license plate characters are recognized, the next step is to verify the accuracy of the recognition results. In the single-stage method, object detection allows for multiple objects to be recognized in a full image. In this scenario, the problems that may occur are as follows. First, an object that is not a license plate character is recognized as a license plate character. Second, an object is a license plate character but is not recognized. Third, an object is a license plate character but contains more than two recognition frames that overlap. Fourth, and finally, an object is a license plate character but is erroneously recognized. In the present study, the M-YOLOv4 first completes the recognition of license plate characters and then compares them by applying standard license plate rules to obtain a higher recognition rate.

In Taiwan, license plates use a hyphen as the separation point for the length of all license plates. Only one hyphen appears on a license plate. However, the hyphen mark is easily confused with a stain, which means that a license plate may be recognized to have more than two hyphens. To overcome this problem, the present study also adds several features of the characters on the left and right side of the hyphen as a class for training (Figure 7).

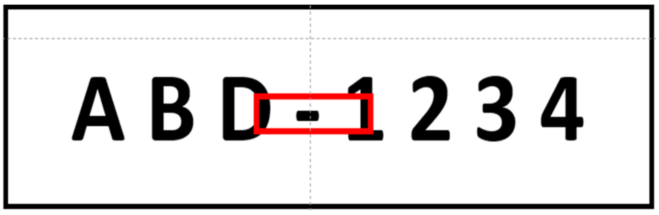

Figure 7. Use of character features on left and right sides of "-" in the red rectangular as a class for training.

In addition, a common rule is imposed for most national license plates; specifically, license plate characters do not overlap and the spacing between each character is fixed (Figure 8). If characters are not recognized during the license plate recognition process, the positions of recognized license plate characters are used to calculate the spacing between the center points of each character. At the same time, the total average width of the license plate characters is calculated. If the character spacing is greater than the total average character width, the system determines that there are missing characters that must be recognized. The license plate characters on both sides are compared with the frame on both sides of a license plate.

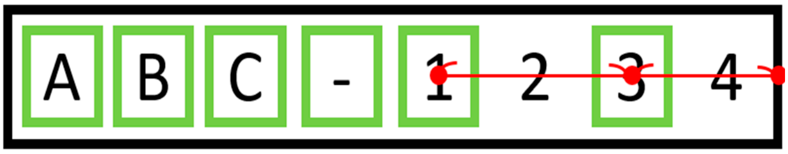

Figure 8. Unrecognized license plate characters. Characters 2 and 4 are missing characters because the character spacing in the red line is greater than the overall average character width.

During the recognition process, a scenario may occur in which more than two recognition frames overlap license plate characters (Figure 9). Because license plate characters do not overlap, the IoU of each license plate character is calculated to determine whether multiple recognition frames have overlapped. When the presence of unrecognized characters or overlapping of multiple recognition frames is detected, recognition failure is established with respect to the recognition results. This approach can improve the credibility of the recognition results.

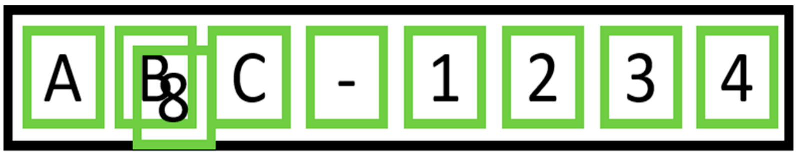

Figure 9. Overlapping of multiple recognition frames.

The proposed system also has a repair function. Specifically, the system observes a large amount of license plate recognition results to identify relevant rules that are then applied to perform repairs. For example, in Figure 9, the two overlapping recognition frames are "B" and "8." When most of the multiple recognition frames overlap, one of the recognition frames usually contains the correct license plate characters. In Figure 10, all overlapping recognition frames are disassembled. After each character is disassembled, the standard license plate rules are applied. Only reasonable standard license plate rules are selected and applied for repairs to obtain accurate recognition results. If a character is not recognized or the recognition error or only reasonable rule is not identified, then a repair cannot be performed; in this situation, the next image is captured for recognition.

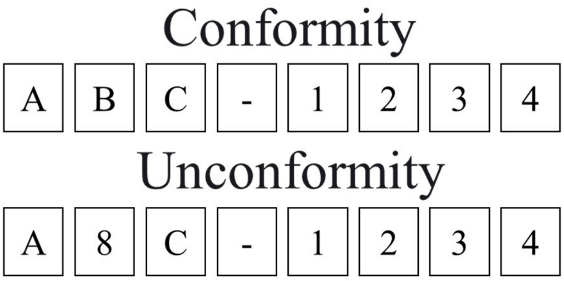

Figure 10. Disassembly of overlapping recognition frames.

3. Experimental Results

Most public data sets of license plates contain static images, including those of license plates obtained from parking lots [22]. Therefore, in the present study, a camera with edge AI was set up in the central region of Taiwan to collect license plate and license plate character data for use as training and test data sets. The LY-AN2616D camera embedded system, which uses AGX, is used. Its installation height is approximately 3 m. If the camera height is too high, the license plate characters in captured images become deformed. If the camera height is too low, vehicles on a given lane would be positioned too close to each other in captured images, such that the license plate of the rear vehicle in an image is ignored. Figure 11 illustrates the implementation of the virtual judgment line for license plate frames. In Figure 11, two virtual judgment lines (i.e., green and red lines) are used for license plate frames. Because more accurate recognition results can be obtained when a frame is closer to the camera, the red virtual judgment line is used in the present study.

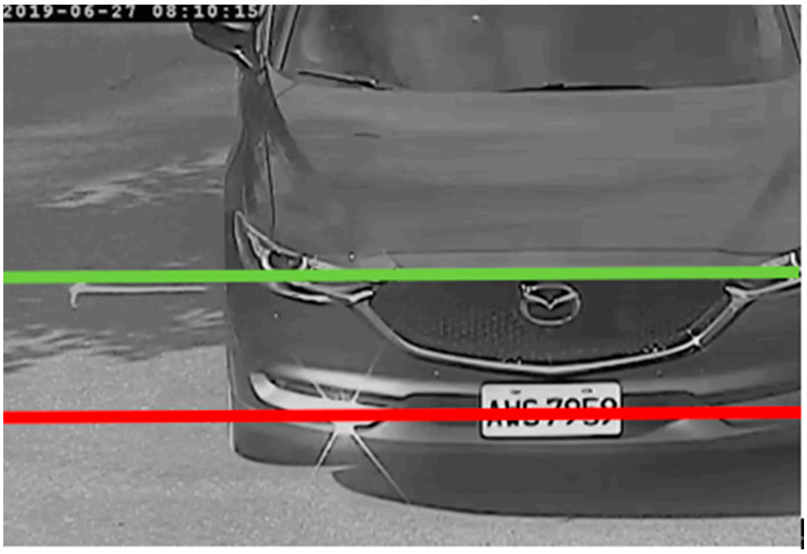

Figure 11. Virtual judgment line setting for license plate frames. Two virtual judgment lines (i.e., green and red lines) are used for license plate frames.

3.1. Collection of License Plate Data Sets

The training data set comprises the real-time images of two road sections in central Taiwan (specifically, Dapi Road and Guangming Road). The images have a dimension of 704 × 480 and were taken under a variety of conditions (i.e., daytime, nighttime, and rain). Almost 2800 license plates were captured in images. License plate characters were divided into 37 categories (A–Z, 0–9, and "-"), and more than 15,000 license plate characters were recognized. The basic details of the license plate character training data set are presented in Table 1.

Table 1. License plate character training data set.

Location	Time or Condition	Number of License Plates
Dayi Road	Daytime, nighttime, and rain	1800
Guangming Road	Daytime, nighttime, and rain	1000

To verify the proposed M-YOLOv4, three evaluation data sets were collected. The first evaluation data set contained 300 images that were taken under daytime and nighttime conditions on Dayi Road and Guangming Road. The second evaluation data set comprised 150 license plate images taken under rainy conditions on Dayi Road and Guangming Road. The third evaluation data set comprised 300 images that were taken under daytime and nighttime conditions on Zhongxing Road; this data set was used to verify the versatility of the proposed approach for other road sections. The basic details of the evaluation data sets are presented in Table 2.

Table 2. Evaluation data set used to verify the proposed approach.

Location	Time or Condition	Number of License Plates
Dayi Road/Guangming Road	Daytime and nighttime	300
Dayi Road/Guangming Road	Rain	150
Zhongxing Road	Daytime and nighttime	300

3.2. Evaluation Results of Proposed M-YOLOv4

To evaluate the output of the model, the category with the highest value in terms of model output (i.e., top-1) was used as the results of the classification. The mean average precision (mAP) [23] was used as the evaluation index. To determine mAP, the average precision (AP) of each category was calculated. The calculation formula is as follows:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

$$\mathrm{AP} = \int_0^1 \mathrm{p(r)dr} \tag{3}$$

$$\mathrm{mAP} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{AP_i} \tag{4}$$

where TP represents the positive samples that are judged as positive, FP represents the negative samples that are judged as negative, FN represents the positive samples that are judged as negative, TN represents the negative samples that are judged as negative, N represents level type, and AP represents the average precision value on a precision–recall curve. Briefly, TP means the license plate characters are detected and recognized correctly; TN represents that the background is detected and recognized as a background; FP indicates the background is detected but recognized as a character; FN denotes the character is detected but recognized as a background.

In the present study, mAP was calculated on the basis of the characters of a single license plate. Through the application of the proposed M-YOLOv4 for testing, the mAP obtained from the experimental results of the three evaluation data sets was 96.1%, 94.5%, and 93.7% (Table 3). The experimental results indicate that favorable recognition results can be obtained on rainy days and on various road sections.

Table 3. Recognition results obtained through proposed M-YOLOv4.

Location	Time or Condition	mAP (%)
Dayi Road/Guangming Road	Daytime and nighttime	96.1
Dayi Road/Guangming Road	Rain	94.5
Zhongxing Road	Daytime and nighttime	93.7

3.3. Real-Time Image Testing of Proposed ER-ALPR System on Actual Roads

To verify the proposed ER-ALPR system, two experiments (one involving six time periods at one location and the other involving four time periods at two locations) were conducted to test the system's real-time image license plate recognition performance on actual roads. For the experiments, when one character on a license plate is incorrectly recognized, the entire license plate is regarded as being incorrectly recognized.

In the first experiment, six time periods and a single location were selected for the testing of real-time image license plate recognition on an actual road. The weather during the six time periods was sunny. Table 4 summarizes the actual verification location, time, and number of license plates obtained. The test was carried out using the proposed ER-ALPR system, and the experimental results are presented in Table 4. The average accuracy rates under daytime and nighttime conditions were 97% and 95.5%, respectively. At night, accuracy rate was reduced because of poor lighting conditions. After the application of the proposed logical assisted judgment system method for license plate characters, the average accuracy rates under daytime and nighttime conditions increased to 98.2% and 97.3%, respectively, representing an overall accuracy rate improvement of approximately 1.5%. The experimental results indicate that the proposed ER-ALPR system is highly accurate when performing real-time image license plate recognition on actual roads under daytime and nighttime conditions.

Table 4. Experimental results on real-world road data at six periods and one location.

Location/Time	Number of License Plates	Proposed ER-ALPR System without Logic Auxiliary Judgment		Proposed ER-ALPR System	
		Number of Correctly Recognized License Plates	Accuracy	Number of Correctly Recognized License Plates	Accuracy
Zhongxing Road 27 June 2021 7–8 A.M.	200	195	97.5%	197	98.5%
Zhongxing Road 27 June 2021 0–1 p.m.	238	230	96.6%	233	97.8%
Zhongxing Road 7 July 2021 8–9 A.M.	69	67	97.1%	68	98.5%
Zhongxing Road 7 July 2021 6–7 P.M.	104	100	96.1%	101	97.1%
Zhongxing Road 7 July 2021 8–9 P.M.	74	70	94.5%	72	97.2%
Zhongxing Road 24 October 2021 0–1 A.M.	221	212	95.9%	216	97.7%

In the second experiment, four time periods and two locations were selected for the testing of real-time image license plate recognition on actual roads. These four time periods were characterized by rainfall of less than 40 mm/h. Because atmospheric transmittance cannot penetrate the rain screen when rainfall is more than 40 mm/h, the license plate characters captured by the camera under such rainfall conditions cannot be visually identified. Table 5 lists the test location, time, and number of license plates obtained. The proposed ER-ALPR system was used for testing, and the experimental results are presented in Table 5. Under rainy daytime conditions, the average accuracy of license plate recognition was 95.3%. Under rainy nighttime conditions, the average accuracy of license plate recognition was 93.1% because of reduced lighting conditions and rain screen interference. After the implementation of the proposed logical assisted judgment system for license plate characters, the average accuracy rates achieved under rainy daytime and rainy nighttime conditions were 96.2% and 94%, respectively, an overall improvement in accuracy rate of approximately 0.9%. The experimental results indicate that the proposed ER-ALPR system is highly accurate when performing real-time image license plate recognition on actual roads during rainy daytime and rainy nighttime conditions.

Table 5. Experimental results on real-world road data at four periods and two locations.

Location/Time	Number of License Plates	Proposed ER-ALPR System without Logic Auxiliary Judgment		Proposed ER-ALPR System	
		Number of Correctly Recognized License Plates	Accuracy	Number of Correctly Recognized License Plates	Accuracy
Guangming Road 28 June 2021 7–8 A.M.	166	158	95.1%	160	96.3%
Dayi Road 27 June 2021 3–4 P.M.	134	128	95.5%	129	96.2%
Guangming Road 27 June 2021 8–9 P.M.	112	105	93.7%	107	95.5%
Dayi Road 27 June 2021 7–8 P.M.	41	38	92.6%	38	92.6%

Figure 12 presents the real-time image recognition results obtained through the application of the proposed ER-ALPR system on actual roads. Figure 12a,e present the license plate recognition results obtained under daytime conditions, Figure 12b,f present the license plate recognition results obtained under nighttime conditions, Figure 12c,g present the license plate recognition results obtained under rainy daytime conditions, and Figure 12d,h present the license plate recognition results obtained under rainy nighttime conditions.

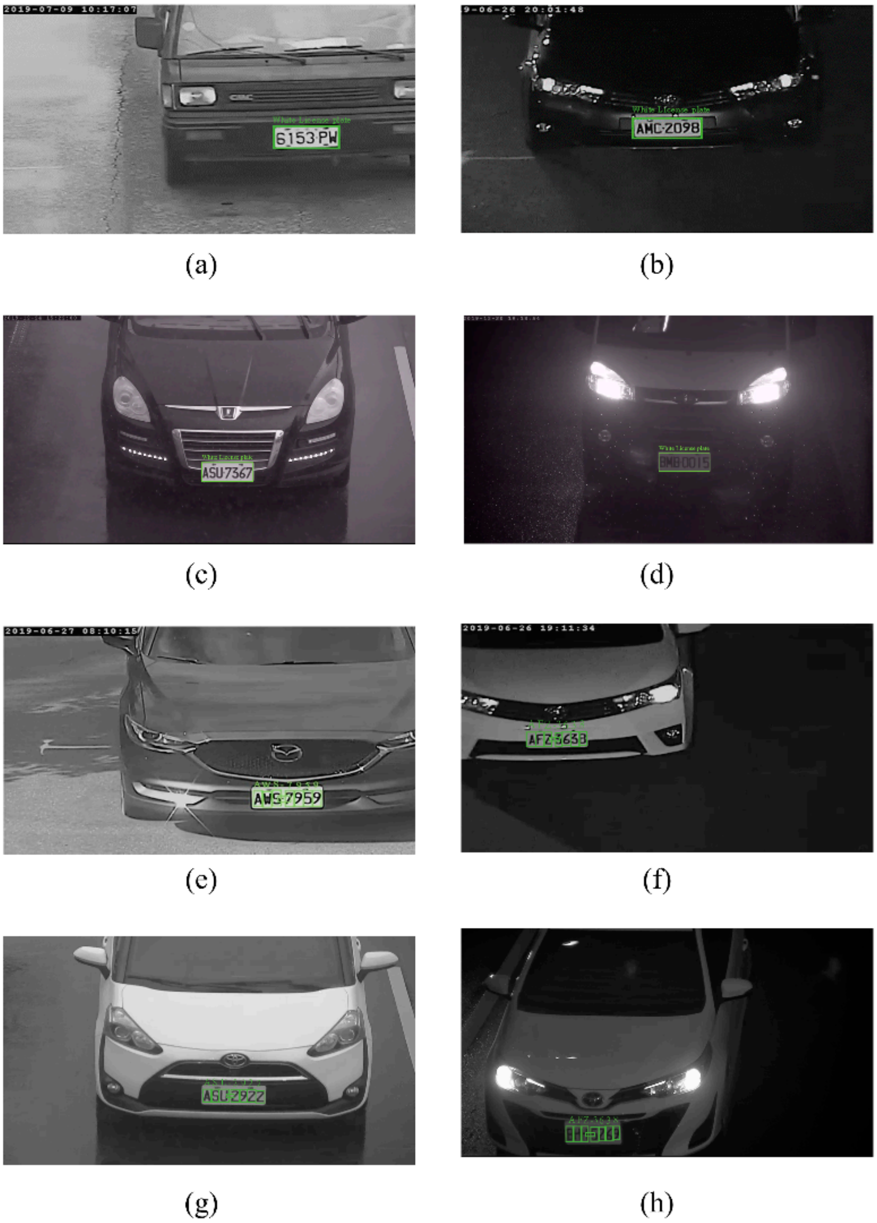

Figure 12. Recognition results obtained using real-time images of actual roads. (**a**) Daytime conditions. (**b**) Nighttime conditions. (**c**) Rainy daytime conditions. (**d**) Rainy nighttime conditions. (**e**) Daytime conditions. (**f**) Nighttime conditions. (**g**) Rainy daytime conditions. (**h**) Rainy nighttime conditions.

4. Conclusions

In the present study, an ER-ALPR system is proposed, in which an AGX XAVIER embedded system is embedded on the edge of a camera to achieve real-time image input to the AGX edge device. YOLOv4-Tiny is used for license plate frame detection. The virtual judgment line is used for determining whether a license plate frame has passed. Subsequently, a proposed M-YOLOv4 is used for license plate character recognition. In addition, a logic auxiliary judgment system is used for improving the accuracy of license plate recognition. The experimental results reveal that the proposed ER-ALPR system can achieve license plate character recognition rates of 97% and 95% under daytime and nighttime conditions, respectively.

In future works, the proposed ER-ALPR system can be improved to address the problem of reduced license plate recognition rates caused by dense fog or heavy rainfall. In addition, a new algorithm can be added to the proposed ER-ALPR system to enable behavior tracking with respect to vehicle movements. These advances aid in the automated detection of traffic violations.

Author Contributions: Conceptualization, C.-J.L. and C.-C.C.; methodology, C.-J.L.; software, C.-J.L., C.-C.C. and H.-Y.L.; data curation, C.-J.L. and H.-Y.L.; writing—original draft preparation, C.-J.L. and C.-C.C.; writing—review and editing, C.-J.L.; supervision, C.-J.L. and H.-Y.L.; funding acquisition, C.-J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology of the Republic of China, grant number MOST 109-2634-F-009-031.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are not publicly available due to their containing information that could compromise the privacy of data providers.

Acknowledgments: The authors would like to give thanks for the support of the Ministry of Science and Technology of the Republic of China in Taiwan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shashirangana, J.; Padmasiri, H.; Meedeniya, D.; Perera, C. Automated license plate recognition: A survey on methods and techniques. *IEEE Access* **2021**, *9*, 11203–11225. [CrossRef]
2. Chang, S.L.; Chen, L.S.; Chung, Y.C.; Chen, S.W. Automatic license plate recognition. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 42–53. [CrossRef]
3. Sirithinaphong, T.; Chamnongthai, K. The recognition of car license plate for automatic parking system. In Proceedings of the Fifth International Symposium on Signal Processing and Its Applications (ISSPA '99), Brisbane, QLD, Australia, 22–25 August 1999; pp. 455–457.
4. Han, B.G.; Lee, J.T.; Lim, K.T.; Choi, D.H. License plate image generation using generative adversarial networks for end-to-end license plate character recognition from a small set of real images. *Appl. Sci.* **2020**, *10*, 2780. [CrossRef]
5. Wen, Y.; Lu, Y.; Yan, J.; Zhou, Z.; von Deneen, K.M.; Shi, P. An algorithm for license plate recognition applied to intelligent transportation system. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 830–845. [CrossRef]
6. Anagnostopoulos, C.E.; Anagnostopoulos, I.E.; Psoroulas, I.D.; Loumos, V.; Kayafas, E. License plate recognition from still images and video sequences: A survey. *IEEE Trans. Intell. Transp. Syst.* **2008**, *9*, 377–391. [CrossRef]
7. Du, S.; Ibrahim, M.; Shehata, M.; Badawy, W. Automatic license plate recognition (ALPR): A state-of-the-art review. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 311–325. [CrossRef]
8. Lin, C.-H.; Li, Y. A License plate recognition system for severe tilt angles using Mask R-CNN. In Proceedings of the 2019 International Conference on Advanced Mechatronic Systems (ICAMechS), Kusatsu, Japan, 26–28 August 2019.
9. Huang, G.; Liu, Z.; Maaten, L.V.D.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
11. Selmi, Z.; Halima, M.B.; Alimi, A.M. Deep learning system for automatic license plate detection and recognition. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017.
12. Laroca, R.; Severo, E.; Zanlorensi, L.A.; Oliveira, L.S.; Gonçalves, G.R.; Schwartz, W.R.; Menotti, D. A robust real-time automatic license plate recognition based on the YOLO detector. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018.
13. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
14. Vijayan, G.; Reshma, S.R.; Dhanya, F.E.; Anju, S.; Nair, G.R.; Aneesh, R.P. A novel shadow removal algorithm using Niblack segmentation in satellite images. In Proceedings of the 2016 International Conference on Communication Systems and Networks (ComNet), Dayton, OH, USA, 21–23 July 2016.

15. Satyanarayanan, M. The emergence of edge computing. *Computer* **2017**, *50*, 30–39. [CrossRef]
16. Rausch, T.; Dustdar, S. Edge intelligence: The convergence of humans, things, and AI. In Proceedings of the 2019 IEEE International Conference on Cloud Engineering (IC2E), Prague, Czech Republic, 24–27 June 2019.
17. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017.
18. Hu, H.; Jiang, C. Edge intelligence: Challenges and opportunities. In Proceedings of the 2020 International Conference on Computer, Information and Telecommunication Systems (CITS), Hangzhou, China, 5–7 October 2020.
19. Wu, B.; Huang, H.; Chen, Y.; Lin, H.; Lee, T.; Chen, C. Degraded license plate recognition system for town buses on highway. In Proceedings of the 2013 10th IEEE International Confercnce on Networking, Sensing and Control (ICNSC), Paris, France, 10–12 April 2013.
20. Sarvate, D.; Bhati, A.; Srivastava, R.; Choudhary, V.; Raghavan, R. Color fading: Variation of colorimetric parameters with spectral reflectance. In Proceedings of the 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Solan, India, 6–8 November 2020.
21. Bochkovskiy, A.; Wang, C.Y.; Liao, H.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
22. Hsu, G.; Chen, J.; Chung, Y. Application-oriented license plate recognition. *IEEE Trans. Veh. Technol.* **2013**, *62*, 552–561. [CrossRef]
23. Davies, P.; Emmott, N.; Ayland, N. License plate recognition technology for toll violation enforcement. In Proceedings of the IEE Colloquium on Image Analysis for Transport Applications, London, UK, 16 February 1990.