Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 3723715, 13 pages
https://doi.org/10.1155/2021/3723715

*Research Article*

# A Deep Learning Model of Dual-Stage License Plate Recognition Applicable to the Data Processing Industry

**Chun-Liang Tung,[1] Ching-Hsin Wang,[2] and Bo-Syuan Peng[1]**

[1]*Department of Information Management, National Chin-Yi University of Technology, Taichung 411030, Taiwan*
[2]*Department of Leisure Industry, National Chin-Yi University of Technology, Taichung 411030, Taiwan*

Correspondence should be addressed to Ching-Hsin Wang; thomas_6701@yahoo.com.tw

Received 3 August 2021; Accepted 4 October 2021; Published 11 November 2021

Academic Editor: Kuei-Hu Chang



Automatic License Plate Recognition (ALPR) is a widely used technology. However, due to the influence of complex environmental factors, recognition accuracy and speed of license plate recognition have been challenged and expected. Aiming to construct a sufficiently robust license plate recognition model, this study adopted multitask learning in the license plate detection stage, used the convolutional neural networks of single-stage detection, RetinaFace, and MobileNet, as approaches to license plate location, and completed the license plate sampling through the calculation of license plate skew correction. In the license plate character recognition stage, the Convolutional Recurrent Neural Network (CRNN) integrated with the loss function of the CTC model was employed as a segmentation-free and highly robust method of license plate character recognition. In this study, after the license plate recognition model, DLPR, trained the PVLP dataset of vehicle images provided by company A in Taiwan's data processing industry, it performed tests on the PVLP dataset, indicating that its precision was 98.60%, recognition accuracy was 97.56%, and recognition speed was FPS > 21. In addition, according to the tests on the public AOLP dataset of Taiwan's vehicles, its recognition accuracy was 97.70% and recognition speed was FPS > 62. Therefore, not only can the DLPR model be applied to the license plate recognition of real-time image streams in the future, but also it can assist the data processing industry in enhancing the accuracy of license plate recognition in photos of traffic violations and the performance of traffic service operations.

## 1. Introduction

The research on Automatic License Plate Recognition (ALPR) has been done for more than 20 years [1]. ALPR technology has also made considerable progress and has been widely used in different application fields and industries, such as automatic traffic violation detection [2, 3]. Take Taiwan's traffic management, for example. After confirming the fact that the driving violation is committed, the traffic violation adjudication unit needs to issue a traffic ticket (containing a photo as proof of the violation) and send it to the driver to pay for the fine. Therefore, the data processing services industry can provide the police in charge of penalties with related services, such as assisting in printing and mailing traffic tickets. The processing procedure is as follows: first, obtain the data of traffic violation cases from various reporting units; next, submit the data to the filing center to import traffic violation photos and citations into the filing system and then to complete registration and monitoring after confirming the car registration information with the Motor Vehicles Office; last, pass the data on to the printing center to complete printing, postal delivery, and transferred mailing data packaging. While importing the traffic violation cases into the filing system, the filer can learn the car registration information, date, location, speed limit, driving speed, and so on from the photo and citation. Meanwhile, the filer must carefully confirm the data one by one and then import them. This process is all manually performed, which is not only cumbersome but also likely to cause business losses to the company when the imported data are incorrect

and even lead to an increase in the police's workload. If the violation photos and violation citations can be analyzed by ALPR first, not only can it reduce the rate of filing error, but it also can speed up the operation process. Therefore, this study adopts the Dual-stage License Plate Recognition Model (DLPR) based on MobileNets, RetinaFace, CRNN, and CTC to enhance the license plate recognition rate and speed for the traffic violation photos and thereby assist the data processing industry in the license plate recognition accuracy of the traffic violation photos as well as the efficiency of the service operation.

With the successful development and application of deep learning in the field of computer vision, such as face recognition and detection, relevant advanced face recognition algorithms are also widely applied to the research of Automatic License Plate Recognition [4–8]. However, the license plate recognition technology still has many challenges in reality, such as the pixel level of the camera, the effects of light and shadow during the day and at night, different weather conditions, different shooting angles, and even possible reflections or stains on the license plates, all of which are complex variables so that the license plate recognition system is prone to recognition errors or failures. In particular, the recognition procedures employed by the license plate recognition technology in the past mostly adopted the three-stage recognition method: license plate detection, character segmentation, and character recognition; especially in the stage of character segmentation, license plate images in a more complex environment were often segmented imperfectly, resulting in incorrect character recognition. Wang [9] established the Chinese City Parking Dataset (CCPD), which contained 250,000 images of different vehicles in different environments, classified the images according to different environmental factors, and proposed a vehicle license plate recognition methodology called Multitask Convolutional Neural Network for license plate detection and recognition (MTLPR). MTLPR is based on the Multitask Convolutional Neural Network (MTCNN) [10] originally used in the face detection model and then applied to the license plate detection. Also, MTLPR collocates the Convolutional Recurrent Neural Network (CRNN) with the Connectionist Temporal Classification (CTC) to conduct training to carry out an optical character recognition without character segmentation [11, 12]. The research result shows that the license plate recognition in CCPD using MTLPR can reach the recognition accuracy of 98%.

Although the application of license plate recognition technology is relatively common, the improvement of accuracy is still limited. Taking the application of smart parking lot management as an example, its license plate recognition accuracy of more than 90% indicates its application value. Besides, the interface of the parking payment system allows users to confirm whether the license plate numbers and vehicle photos belong to their own vehicles or the license plate recognition system can perform a fuzzy comparison with the dataset for the currently recognized license plate number, so there is a certain error tolerance space, and the photo-taking environment of the parking lot can be controlled to a certain degree, in order to avoid the environmental factor that affects the recognition rate. However, when facing the issues of traffic law enforcement, any character of the license plate number recognized by ALPR cannot be wrong or missing, especially character "-" in Taiwan's license plate, whose position cannot be wrongly recognized. If the license plates of all vehicles cannot be correctly recognized, the workload of manual processing will increase. On the other hand, there is a variety of photo sources for traffic law enforcement based on science and technology. For example, photos of traffic violations are taken around the clock by the fixed cameras on the roadside outdoors. Consequently, they may be influenced by various complex environmental factors, such as weather change, different day and night light, light and shadow reflections, and distances between cameras and offending vehicles. Another example is the photo for proof taken by a roadside parking toll collector. This photo was taken by a toll collector with a mobile phone. Therefore, the position of the license plate in the photo was affected by environmental factors. Not only was it affected by light and shadow, but also it was affected by different shooting angles due to the field situation or personal operation, usually including high-angle shot, skew angle of the horizontal axis, horizontal offset, and bias angle of the vertical axis. Under the influence of the above-mentioned various environmental factors, if the robustness of the license plate recognition method cannot be adopted, the recognition rate will often be unsatisfactory.

Every country has its own license plate encoding format or appearance arrangement, and the development of LPR needs to be adjusted based on different regions. Take the Taiwanese license plate regulated by the Directorate General of Highways of Taiwan, for example. The current vehicles in Taiwan have new-style license plates with 7 characters and old-style license plates with 6 characters, whereas special vehicles have 5- or 4-character license plates. In addition, even the font styles used on the new-style and old-style license plates are different as well. Moreover, based on application requirements, the "-" character on the license plate also needs to be correctly recognized. However, the "-" character is arranged in different positions of the new style, old style, and special vehicles, making it difficult to correctly identify the position of the "-" character on all license plates. Therefore, these special conditions have become one of the challenges of LPR.

The research purpose of this study is to propose a methodology of license plate recognition that can adapt to the influence of a variety of realistic environmental factors, reach a fairly high rate of precision and accuracy under the influence of various environmental factors after completing the test results of training the license plate recognition model, and enable the calculation speed to meet the requirements of real-time detection and recognition, leaving more room of development and application for the license plate recognition technology. In the experiment, this study adopts the Taiwanese public license plate dataset, Application-Oriented License Plate (AOLP) [13], and the Taiwanese license plate dataset of a company (the company has signed a confidentiality contract with the government), Private Vehicle License Plate (PVLP), in the Taiwan data service

industry to perform training and testing of the DLPR license plate detection and recognition model proposed by this study. However, due to legal issues, the PVLP dataset is a nonpublic database, so it is only applied to this study, whereas the AOLP dataset is a public database, so it can be provided for external use and academic research.

## 2. Related Works

If the early artificial neural network (ANN) needs to extract features from images to perform object detection and recognition, it must rely on the artificially defined feature descriptor or the so-called feature extractor. For example, Haar-like features [14] are used for object detection; features of histograms of oriented gradients (HOG) [15] are used for pedestrian detection; local binary patterns (LBP) [16] are used to calculate the texture characteristics of objects. However, these artificially defined feature descriptors are usually combined with machine learning algorithms, such as support vector machines (SVM) [17] and adaptive boosting (AdaBoost) [18], to effectively classify or predict eigenvalues. Lecun et al. [19] were the first scholars who proposed the concept of convolutional neural network (CNN). Also, they came up with gradient-based neural network learning and applied it to document recognition [20, 21]. The computation of CNN usually goes through multiple convolutional layers and pooling layers, respectively, and finally enters the fully connected layers to analyze the classification results of eigenvalues.

Nevertheless, the development of CNN was not quite smooth at the beginning. The reasons were the limitations of hardware technology in the 1990s and the CNN training's heavy reliance on a large number of sample datasets, but the concept of the establishment of large sample datasets was not very popular at that time. Later, GPU (Graphics Processing Unit) developed by Nvidia was born, which can help accelerate massive matrix computations of CNN. Krizhevsky et al. improved CNN and proposed AlexNet [22], combining GPU to perform high-speed training and inference computation [23–25]. AlexNet beat other methodologies in the object recognition contest of ImageNet 2012 and won the championship, which made many scholars' eyes widen at the sight. This is also a milestone in the entire field of machine learning. Since then, CNN and deep learning have gradually become today's significant studies. Nowadays, CNN is widely used in image recognition [26] and natural language processing [27]; for example, Tao et al. [8, 9] adopted a lightweight CNN as a network structure for license plate character recognition. Its main task is to perform feature extraction on data and carry out parameter learning through a large amount of data. It can adaptively adjust the weight parameters of the feature extractor and then extract more meaningful information from the data. Compared with the traditional artificially defined feature descriptors, CNN has better robustness, and the recognition rate in many studies using CNN is better than that in the traditional recognition methodologies. The reason is as mentioned by Zhao et al. [28]. If the features of the object extracted from the image adopt the artificially defined feature descriptor, it will be difficult to cover the factors that need to be resolved, such as light, shadow, and complex background. Therefore, as long as a large and diverse number of samples can be provided in the CNN training phase, a better feature extractor can be obtained after the training is completed.

The loss functions of the object detection models that adopt deep learning will directly affect the identification capability of the models. Among them, the loss functions more commonly used in the regression model include MSE (Mean Square Error) and MAE (Mean Absolute Error), the loss function more commonly used in the classification model includes cross entropy, and the loss functions more commonly used in the CNN model include Huber Loss, Focal Loss, and Center Loss. The advantage of Huber Loss is that MSE is less sensitive to outliers and able to accelerate the speed of convergence. In the well-known deep learning object detection models, RetinaNet [29] uses Focal Loss to deal with the foreground-background class imbalance or data imbalance in object detection; YOLOv4 [30] adopts BCE (Binary Cross Entropy) as the loss function of classification and MSE as the loss function of coordinate; SSD [31] employs cross entropy as the loss function of the predicted boundary box and class prediction.

Object detection algorithms using deep learning algorithms can be divided into three categories: sliding-window algorithm, two-stage detectors, and one-stage detectors [29]. The sliding-window algorithm is one of the early methodologies. LeCun et al. integrated the sliding-window algorithm with the CNN model to detect and recognize handwritten digits from the background [32]. Astawa et al. [33] adopted the sliding-window algorithm as a method of license plate location and performed license plate detection using photos taken by mobile phones with a detection rate of 94%. In their paper, the sliding-window algorithm was used to capture multiple candidate regions in the photos, HOG was employed by each candidate region to calculate an eigenvalue, and finally, SVM was applied to judge whether the eigenvalue was a license plate. Uijlings et al. [34] used two-stage detectors and a selective search algorithm to perform license plate detection. First, in the first stage called the proposal stage, a large number of candidate regions with more possibilities were proposed; although these regions still contained foregrounds and backgrounds, they also had excluded a large number of background regions at the same time. In the second stage, the detection model was employed to determine these candidate regions; if they were foregrounds, their categories would be identified; if they were backgrounds, they would be eliminated. The method of two-stage detectors is obviously more efficient than the sliding-window algorithm to detect all regions. R-CNN (Region Convolution Neural Network) [35] also applies the method of two-stage detectors to object detection. In the first stage, a selective search algorithm is adopted; in the second stage, the CNN model performs the classification task. As a result, it significantly improves the recognition accuracy. Faster R-CNN [36] merges the first-stage Region Proposal Networks (RPN) with the second-stage CNN, which not only greatly reduces a large number of meaningless candidate regions (alternative regions) but also enhances accuracy and

facilitates training so that there is no need to train these two network models separately. Compared with the two-stage detectors, one-stage detectors lack the proposal stage, so they require more object candidate regions for processing, and they also need to deal with the problem of the imbalance of foreground and background data like SSD [31, 37], YOLOv4 [30], RetinaNet [29], and RetinaFace [38] do. Based on the above description, this study adopts a one-stage detector as the algorithm for license plate location in order to achieve the purpose of real-time license plate location.

The development of the convolutional neural network (CNN) has become more and more mature and has been widely used [39, 40]. Therefore, the Dual-stage License Plate Recognition Model (DLPR) proposed by this study also adopts the CNN-based methodology in the license plate location stage and the license plate character recognition stage. The license plate location stage refers to the methodology of RetinaFace applied to human face detection, which is a one-stage object detector of multitasking learning. Since its architecture has an excellent effect on the application to face detection, this study applies the architecture of the RetinaFace framework and MobileNet to the license plate location stage. In the license plate character recognition stage, to help the character recognition technology adapt to the complex shooting environment, this study adopts an end-to-end segmentation-free method of license plate character recognition; that is, the CRNN model is integrated with the CTC loss function for optical character recognition.

## 3. Methods

The DLPR license plate recognition method proposed by this study aims to maintain high and robust recognition ability under the influence of a variety of real environments. The overall architecture of the license plate recognition system is displayed in Figure 1.

In the architecture of the DLPR license plate recognition system, the license plate location module regards RetinaFace as the basis for the license plate location. RetinaFace is a multitask neural network infrastructure. This model is used to perform tasks such as license plate detection, bounding box regression, and landmark regression. Therefore, after the location of the license plate and the coordinates of its four key points (upper left, lower left, upper right, and lower right) are predicted, these four key points can correct the license plate placed at a skew angle via perspective transformation, and then a screenshot of the corrected license plate can be received. Subsequently, the screenshot of the license plate is input into the license plate character recognition module, which is a segmentation-free character recognition method implemented by CRNN combining CTC, and finally, the license plate string can be recognized.

RetinaFace can accurately predict the location of the license plate and the coordinates of four key points on the license plate in the image frame. Accordingly, this network infrastructure includes the outputs of three tasks-a classification task, a bounding box regression task, and a landmark regression task. This study adopted the faster MobileNet [41] as the backbone network of the model and combined the Feature Pyramid Network (FPN) [42] to keep scale-invariant as much as possible. In FPN, the feature maps of each layer $P = \{p_1, p_2, p_3\}$ are input to the context module, and three different sizes of receptive fields are applied to the feature maps of each layer for the convolution operation to obtain the feature maps at different scales in order to further maintain the scale-invariant ability. Finally, the context module calculates the feature maps of each layer and then performs three multitask calculations again, including classification head, bounding box head, and landmark head. Besides, the prior box mechanism proposed by SSD is used as an encoding and decoding mechanism during model training and inference; that is, this is a method utilizing bounding box head and landmark head to figure out the feature maps corresponding to the coordinates of the original image.

*3.1. MobileNet and Feature Pyramid Network.* MobileNet is a lightweight convolutional neural network that employs the calculation method of Depthwise Separable Convolution to achieve the effect of reducing the computation load without affecting the size of the output structure. This study used MobileNet as the backbone network infrastructure combining the Feature Pyramid Network (FPN) to maintain the model's scale-invariant ability, which means that the model targeted at any objects of any scales has a certain degree of robustness which can predict their positions as well as sizes. FPN divides the backbone network into several sequential stages. In this study, there were three layers in the FPN so that $C = \{c_1, c_2, c_3\}$ represented the feature maps of each stage in the backbone network. The topmost feature map $c_3$ had deeper dimensional features and more meaningful semantics, but it could not accurately correspond to the coordinate of the original image where the object was located; on the contrary, the bottom feature map $c_1$ had a lower dimensionality, which could not acquire better semantic information, but it was closer to the original image in dimension so that it was easier to correspond to the coordinate of the original image. The establishment of the feature pyramid started with the top-layer feature map $c_3 = p_3$, and $p_3$ performed upsampling to make itself the same size with $c_2$; $c_2$ processed by the $(1 \times 1)$ convolution was added up to obtain the second feature pyramid layer $p_2$. Based on the above mentioned, a three-layer feature pyramid, $P = \{p_1, p_2, p_3\}$, was established, so FPN retained the feature map information of the low and high layers to achieve the scale-invariant ability.

*3.2. Context Module.* The application of the context module originated from the face detection method proposed by SSH [43]. This method can detect faces with different scales for the same input image at one time. The context module uses receptive fields in different sizes to generate different feature maps and finally concatenates them into a new feature map, which also achieves a certain degree of scale invariance. This study input three feature maps of $P = \{p_1, p_2, p_3\}$ obtained from the FPN calculation into their corresponding context module for calculation. The context module was composed of

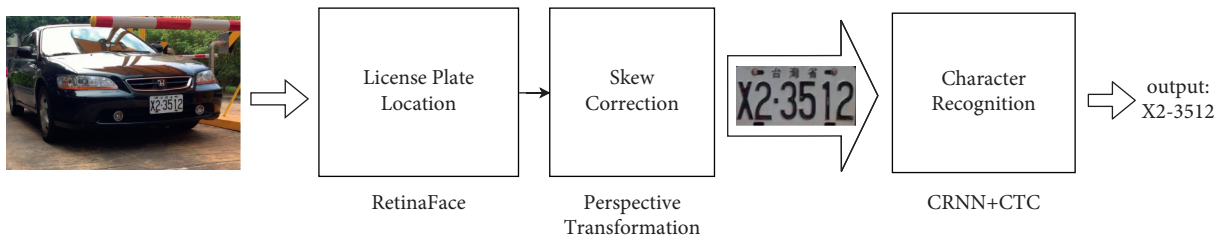



Figure 1: Architecture of DLPR license plate recognition model.

three groups of receptive fields in various sizes, whose convolution kernel sizes were, respectively, presented as $\{(3 \times 3), (5 \times 5), (7 \times 7)\}$. Lastly, the feature maps obtained by convolutions of these three groups of convolution kernels were merged into a new feature map, FM = $\{fm_1, fm_2, fm_3\}$.

*3.3. Prior Box.* The prior box mechanism used by SSD can be applied to the model training and inference of one-stage detectors. The purpose of this mechanism is to set up a prior box with a fixed number and fixed size. The prior box is set up according to the anchor on the final feature map, regarded as the basis for the feature map and its corresponding coordinate of the original image. Therefore, this study used the three feature maps of FM = $\{fm_1, fm_2, fm_3\}$ calculated by the context module to set an anchor for the center position of each cell in the two-dimensional tensor of each feature map and set up two prior boxes with each anchor viewed as the center. In other words, each anchor on the feature map was the center point of the prior box, and its coordinates on the original image were expressed as $(X_{p\text{box}}, Y_{p\text{box}})$, $X_{p\text{box}} \leftarrow (X_{fm_i} + 0.5) \times s_{fm_i}$, and $Y_{p\text{box}} \leftarrow (Y_{fm_i} + 0.5) \times s_{fm_i}$, in which $(X_{fm_i}, X_{fm_i})$ indicated all coordinates in the two-dimensional tensor of the feature map $fm_i$, 0.5 was the offset of the coordinates, and $s_{fm_i}$ was the stride of the layer. For each feature map of FM = $\{fm_1, fm_2, fm_3\}$, the two prior boxes set up on the anchors had different sizes. The prior boxes on $fm_1$ were matrix $fm_1^1 B_{p\text{box}} = [bij] \in R^{16\times16}$ of $(16 \times 16)$ and matrix $fm_1^2 B_{p\text{box}} = [bij] \in R^{32\times32}$ of $(32 \times 32)$; the prior boxes on $fm_2$ were matrix $fm_2^1 B_{p\text{box}} = [bij] \in R^{64\times64}$ of $(64 \times 64)$ and matrix $fm_2^2 B_{p\text{box}} = [bij] \in R^{128\times128}$ of $(128 \times 128)$; and the prior boxes on $fm_3$ were matrix $fm_3^1 B_{p\text{box}} = [bij] \in R^{256\times256}$ of $(256 \times 256)$ and matrix $fm_3^2 B_{p\text{box}} = [bij] \in R^{512\times512}$ of $(512 \times 512)$. Since the size of the original input image varies, the shape and size of the feature map FM calculated by the context module will be different as well. Therefore, the total number of the set prior boxes is $N_{p\text{box}}$, whose calculation is shown in equation (1), where $\text{num}_{p\text{box}} = 2$ represents that the number of prior boxes on each anchor is set to 2 in this study.

$$N_{p\text{box}} = \sum_i \text{num}_{p\text{box}} \times h_{fm_i} \times w_{fm_i}, \quad i = 1, 2, 3. \tag{1}$$

Targeted at numerous prior boxes, a matching mechanism combining the prior box and the ground-truth box is adopted in the training phase as a matching method for the positive and negative samples of the prior boxes. By calculating the Intersection over Union (IoU) between the ground truth $B_{gt}$ and the prior box $B_{p\text{box}}$, the IoU value is used as the basis of matching. Its calculation is displayed as follows:

$$J\left(B_{gt}, B_{p\text{box}}\right) = \frac{\left|B_{gt} \cap B_{p\text{box}}\right|}{\left|B_{gt} \cup B_{p\text{box}}\right|}, \tag{2}$$

where the value of output by function $J(B_{gt}, B_{p\text{box}})$ is IoU > 0.5, indicating that $B_{p\text{box}}$ will be matched as the positive sample of $B_{gt}$; otherwise, $B_{p\text{box}}$ will be matched as the negative sample of $B_{gt}$.

*3.4. Multitask Loss Function.* In the final multitask calculation stage, for the feature maps of FM = $\{fm_1, fm_2, fm_3\}$, three multitask network calculations are performed, respectively, which are classification head representing the classification task, bounding box head for the bounding box regression task, and landmark head for the landmark regression task [8, 9]. Each head analyzes all the prior boxes on FM according to the purposes of the tasks, and then the final outputs $\{A_{\text{cls}}, A_{b\,\text{box}}, A_{\text{landmark}}\}$ of the model are obtained. Among the outputs, the output of the classification head, $A_{\text{cls}} = [a_{ij}] \in R^{N_{p\text{box}}\times2}$, represents the confidence score of the $j^{\text{th}}$ category of the $i^{\text{th}}$ prior box in the classification task; the output of the bounding box head, $A_{b\,\text{box}} = [a_{ij}] \in R^{N_{p\text{box}}\times4}$, represents the offset of the $j^{\text{th}}$ coordinate value of the $i^{\text{th}}$ prior box in the bounding box regression task; the output of the landmark head, $A_{\text{landmark}} = [a_{ij}] \in R^{N_{p\text{box}}\times8}$, represents the offset of the $j^{\text{th}}$ coordinate value of the $i^{\text{th}}$ prior box in the landmark regression task. In order to train the model to accurately perform these three tasks, this study adopted a multitask learning method. In the case of weights shared by the model, the loss function of the classification task $L^{\text{cls}}(x_n, y_n)$, the loss function of the bounding box regression task $L^{b\,\text{box}}(x_n, y_n)$, and the loss function of the landmark regression $L^{\text{landmark}}(x_n, y_n)$ are defined, respectively. The target function of the overall model training is demonstrated in the following equation:

$$\theta^* = \arg\min_{\theta}\left[\frac{1}{N_{\text{samples}}}\sum_{n=1}^{N_{\text{samples}}}\left(\alpha_{\text{cls}}L_n^{\text{cls}} + \alpha_{b\,\text{box}}L_n^{b\,\text{box}} + \alpha_{\text{landmark}}L_n^{\text{landmark}}\right)\right], \tag{3}$$

where the neural network parameter $\theta$ is adjusted to minimize the overall loss value and $N_{\text{samples}}$ represents the total number of training samples. Also, this study set parameters $\{\alpha_{\text{cls}} = 1, \alpha_{b\,\text{box}} = 2, \alpha_{\text{landmark}} = 1\}$ as the weights of training all tasks. The purpose of training the classification task in this study is to accurately distinguish whether each prior box is a license plate, that is, to perform two types of binary classification. The output of the classification head, $A_{\text{cls}} = [a_{ij}] \in R^{N_{p\text{box}}\times 2}$, means that each prior box contains two confidence scores for judging whether it is a license plate, so they can also be called classification probabilities. The classification task uses cross entropy as the basis to design the loss function of its task, $L^{\text{cls}}(x_n, y_n)$, whose definition is exhibited in the following equation:

$$L^{\text{cls}}(x_n, y_n) = L_n^{\text{cls}} = \frac{1}{N_{p\text{box}}}\sum_{i=1}^{N_{p\text{box}}} -y_i^{\text{cls}}\log(p_i) + \left(1 - y_i^{\text{cls}}\right)(1 - \log(p_i)), \tag{4}$$

where there are $N_{p\text{box}}$ prior boxes in the $n^{\text{th}}$ image sample. After the prior boxes $N_{p\text{box}}$ match all the ground truths of $y_n$ on the image, each prior box will be matched to its corresponding ground truth. As to $y_i^{\text{cls}} \in \{0, 1\}$, 0 represents that the $i^{\text{th}}$ prior box is matched as a nonlicense plate, 1 represents a license plate, and $p_i$ represents the probability that the $i^{\text{th}}$ prior box is predicated as a license plate. Therefore, cross entropy calculates the difference between $y_i^{\text{cls}}$ and $p_i$ and retrieves their average to solve for the loss function of the $n^{\text{th}}$ sample.

In this study, the bounding box regression task was trained so that the model could accurately calculate the offset between the specific prior box and the ground-truth box. The output of the bounding box head, $A_{b\,\text{box}} = [a_{ij}] \in R^{N_{p\text{box}}\times 4}$, indicates that each prior box contains the offsets of four coordinate values, which are the offsets of the upper left corner coordinates, width, and height $(\Delta x_i, \Delta y_i, \Delta w_i, \Delta h_i)$. The bounding box regression task uses $L_2$ distance as the basis to design the loss function $L^{b\,\text{box}}(x_n, y_n)$ of the task, as shown in the following equation:

$$L^{b\,\text{box}}(x_n, y_n) = L_n^{b\,\text{box}} = \frac{1}{N_{p\text{box}}}\sum_{i\,\in\,\text{Positive}}^{N_{p\text{box}}}\left\|y_i^{b\,\text{box}} - o_i^{b\,\text{box}}\right\|_2^2. \tag{5}$$

Among all the prior boxes of the $n^{\text{th}}$ sample $x_n$, only the prior boxes that are matched as positive samples are considered to calculate the $L_2$ distance between the true offset $y_i^{b\,\text{box}}$ and the offset $o_i^{b\,\text{box}}$ predicted by the model, whose average is retrieved to solve for the loss function of the $n^{\text{th}}$ sample.

Since the landmark regression task was trained in this study, the model could accurately calculate the offsets between the upper left corner coordinates of the specific prior box and the 4 ground landmarks on the real object. The output of landmark head, $A_{\text{landmark}} = [a_{ij}] \in R^{N_{p\text{box}}\times 8}$, means that each prior box contains the offsets of the coordinates (upper left, lower left, upper right, and lower right) of 4 ground landmarks, that is, a total of 8 offset values $\{\Delta p_1 x_i, \Delta p_1 y_i, \Delta p_2 x_i, \Delta p_2 y_i, \Delta p_3 x_i, \Delta p_3 y_i, \Delta p_4 x_i, \Delta p_4 y_i\}$. Similarly, the landmark regression task also adopts the L2 distance as the basis to design the loss function of its task, $L^{\text{landmark}}(x_n, y_n)$, as displayed in the following equation:

$$L^{\text{landmark}}(x_n, y_n) = L_n^{\text{landmark}} = \sum_{i\,\in\,\text{Positive}}^{N_{p\text{box}}}\left\|y_i^{\text{landmark}} - o_i^{\text{landmark}}\right\|_2^2. \tag{6}$$

Among all the prior boxes of the $n^{\text{th}}$ sample $x_n$, only the prior boxes that are matched as positive samples are considered to calculate the $L_2$ distance between the true offset $y_i^{\text{landmark}}$ and the offset $o_i^{\text{landmark}}$ predicted by the model, whose average is retrieved to solve for the loss function of the $n^{\text{th}}$ sample.

*3.5. Perspective Transformation.* In the process of photographing license plates, the license plates may be skewed and difficult to be accurately recognized due to the different shooting angles. Therefore, this study adopted the method of MTLPR, performing perspective transformation with four corner points of the license plate. The license plate with a skewed angle on the image was projected to more positive new coordinates to correct its skew. Perspective transformation is adopted to help the two-dimensional plane $[u\ v\ 1]$ correspond to the three-dimensional space $[x\ y\ z]$ through the transformation matrix, $M = [a_{ij}] \in R^{3\times 3}$, as shown in the following equation:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{7}$$

where $(u, v)$ refers to the coordinates on the original image and $(x, y, z)$ represents the coordinates in the transmitted three-dimensional space. To make the transmitted three-dimensional space be seen as a two-dimensional plane, the three-dimensional space $[x\ y\ z]$ is divided by $z$ to form a new three-dimensional space $[x'\ y'\ 1]$. Consequently, $(x', y')$ can be regarded as new coordinates on the two-dimensional plane, as displayed in the following equation:

$$x' = \frac{x}{z} = \frac{a_{11}u + a_{12}v + a_{13}}{a_{31}u + a_{32}v + a_{33}} = \frac{k_{11}u + k_{12}v + k_{13}}{k_{31}u + k_{32}v + 1},$$
$$y' = \frac{y}{z} = \frac{a_{21}u + a_{22}v + a_{23}}{a_{31}u + a_{32}v + a_{33}} = \frac{k_{21}u + k_{22}v + k_{23}}{k_{31}u + k_{32}v + 1}, \quad (8)$$

where $k_{ij} = (a_{ij}/a_{33})$. Thus, $k_{ij}$ has 8 elements in total, and 4 sets of coordinates on the original image and the corresponding coordinates of each set on the new plane are needed to solve for $k_{ij}$ and further solve for the transmission matrix $M = [a_{ij}] \in R^{3\times3}$. Therefore, the coordinates of the four key points of the license plate on the image predicted in the license plate location stage are $P_1 = (p_1x, p_1y)$, $P_2 = (p_2x, p_2y)$, $P_3 = (p_3x, p_3y)$, and $P_4 = (p_4x, p_4y)$, representing the coordinates of the upper left corner, lower left corner, upper right corner, and lower right corner of the license plate. Also, the coordinates of these four key points and their corrected coordinates on the new plane, $P_1' = (0, 0)$, $P_2' = (0, \max(p_2y, p_4y))$, $P_3' = (\max(p_3x, p_4x), 0)$, and $P_4' = (\max(p_3x, p_4x), \max(p_2y, p_4y))$, are substituted into equation (8) to solve for the transmission matrix $M = [a_{ij}] \in R^{3\times3}$. Then, all the points in the quadrilateral surrounded by $P_1$, $P_2$, $P_3$, and $P_4$ on the original image can correspond to the new plane to complete the license plate correction.

*3.6. License Plate Character Recognition of CRNN and the CTC Loss Function.* The license plate character recognition in this study adopts the Convolutional Recurrent Neural Network (CRNN) as an infrastructure and CTC as the loss function to train the model, and finally, the result sequence predicted by the model is transcribed into the license plate number, as shown in Figure 2. First, the license plate image is standardized and compressed into size $(100 \times 32)$ and input into the CNN layer for feature extraction to obtain a feature map. Subsequently, the feature map is converted into a feature sequence, and then the feature sequence is input into the RNN layer for analysis and prediction to receive an output sequence. Finally, the output sequence is transcribed into a license plate string.

CRNN is composed of the Convolution Neural Network layer (CNN layer) and the Recurrent Neural Network layer (RNN layer) in order. In this study, the feature map $(16 \times 1 \times 512)$ calculated by the CNN layer is mapped into a feature sequence $X$ of $(512 \times 1 \times 16)$ with a sequence length of 16; the input of each time step is a 512-dimensional feature vector, as shown in Figure 3. As a result, the feature sequence can be input into the RNN layer for calculation.

The RNN layer performs the classification task of 36 characters (from $A$ to $Z$, and 0 to 9); that is, 36-dimensional classification result $y_t$ is received after feature vector $x_t$ at each time step is analyzed. When 512-dimensional $x_t$ enters the input layer, $W_1$ is the weight of the first hidden layer, which is a matrix of order $(N_1 \times 512)$; in addition, $b_1$ is the bias of the first hidden layer, and the output result of the first hidden layer, $H_1$, is calculated by equation (9) as follows:

$$H_1 = \mathrm{ACT}(W_1 \cdot x_t + b_1), \quad (9)$$

$$y_t = \mathrm{ACT}(W_{\text{output}} \cdot H_n + b_{\text{output}}), \quad (10)$$

where $H_1$ is obtained by means of $x_t$ calculated with weight $W_1$, bias $b_1$, and activation function ACT. $H_1$ is a vector of dimension $N_1$ and can be subsequently used as the input of the second hidden layer. After $H_1$ is calculated by the second hidden layer, $H_2$ as a vector of dimension $N_2$ is received and continuously regarded as the input of the next layer. The above process continues until the output layer is figured out. Similarly, the output of the last hidden layer $H_n$ is also used as the input of the output layer. Through the calculation of weight $W_{\text{output}}$, bias $b_{\text{output}}$, and activation function ACT, $y_t$ is obtained as a 36-dimensional classification result, as displayed in equation (10), where $W_{\text{output}} \in R^{36\times N_n}$ is a matrix of $(36 \times N_n)$. The single-layer architecture of LSTM can sequentially input the feature sequence $X = \{x_1, x_2, x_3, \ldots, x_{16}\}$ beginning from $x_1$, perform calculations to obtain the output result $y_1$, and transfer memory and output to the next time step. Based on this rule, circular calculations are performed until $x_{16}$. Finally, the result sequence $Y = \{y_1, y_2, y_3, \ldots, y_{16}\}$ is obtained.

This study adopted the Deep LSTM network with a hidden layer depth of 256 layers. Each layer of LSTM needs to train 4 sets of weights and biases, so there are $256 \times 4 \times 2$ sets of parameters. $h_t^d$ is expressed as the output of the $d^{\text{th}}$ hidden layer at time step $t$; $c_t^d$ is expressed as the memory of the $d^{\text{th}}$ hidden layer at time step $t$. In the Deep LSTM calculation, the output of each layer, $h_t^d$, will be used as the input of the next layer, and $h_t^d$ and $c_t^d$ will also be passed to the next time step. Finally, the output of the last layer, $h_t^{255}$, is equal to the result $y_t$ output at time step $t$. In the training and inference process of LSTM, this study used the CTC alignment to train the best parameter $\theta^*$, as shown in equation (11), where $S$ is the license plate characters used for training and $P(S|Y) = \sum P(h|Y)$:

$$\theta^* = \arg\max_{\theta} P_\theta(S|Y). \quad (11)$$

After the inference calculation of Deep LSTM, the result sequence $Y = \{y_1, y_2, y_3, \ldots, y_{16}\}$ is obtained. The purpose of the task is to analyze the probabilities of various character classes at each time step from the feature sequence. Therefore, $y_t$ is a 36-dimensional vector, representing the classification result of 36 classes. This study transmitted the result at each time step into a probability distribution of 36 classes via the softmax function so that the result sequence $Y$ could be viewed as the probability distribution of 36-character classes at each time step.

Finally, in CRNN's transcription layer, this study utilized the result sequence $Y$ to find the best path $L^*$ and output license plate characters by means of greedy decode. The greedy decode only considers the node with the highest probability as the path; therefore, although $h^*$ can only approximate the result of the best path $L^*$, it can save more computing resources and achieve the effect of rapid recognition.

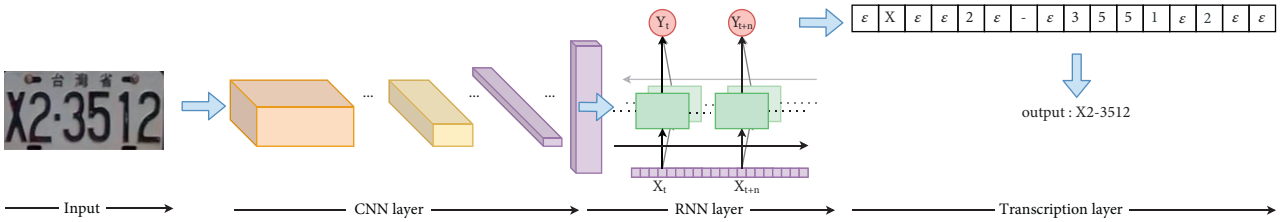



Figure 2: CRNN + CTC license plate character recognition procedure.

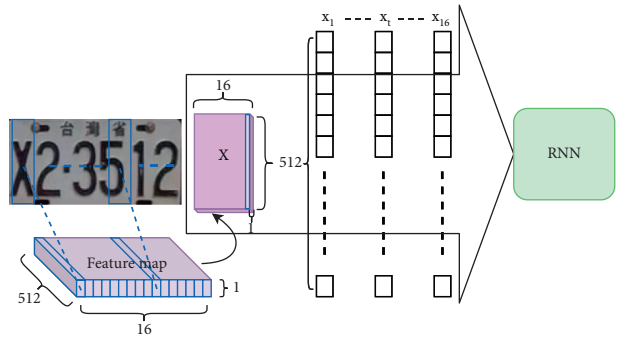
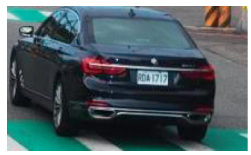



Figure 3: The feature map corresponding to the feature sequence.

## 4. Experiments

*4.1. Dataset.* This study used Taiwan's Application-Oriented License Plate (AOLP) dataset released by the National Taiwan University of Science and Technology with a total of 2049 sample photos and the Private Vehicle License Plate (PVLP) dataset of company A with a total of 25,707 sample photos as the training samples and test samples of the DLPR model proposed by this study. The AOLP dataset collected Taiwanese license plate images in a variety of real environments according to some specific criteria, while the PVLP dataset directly collected Taiwanese license plate images applied to different scenarios in the real world, such as highways, expressways, and roadside parking spaces. In the AOLP dataset, according to the criteria of image shooting, the dataset was divided into three different subdatasets, including Access Control (AC), Traffic Law Enforcement (LE), and Road Patrol (RP). AC had a total of 681 photos, consisting of 374 photos with image resolution 320 × 240 and 307 photos with image resolution 352 × 240, in which all the license plates had 6 characters. LE had a total of 757 photos, consisting of 582 photos with image resolution 640 × 480 and 175 photos with image resolution 320 × 240, in which there were 756 photos of six-character license plates and only one photo of seven-character license plates. RP had a total of 611 photos with image resolution 320 × 240, in which all the license plates had 6 characters. In the AC subdataset, the image was taken from the entrance or gate control, so the vehicle was basically driving at a low speed or completely stopped, and the camera was mounted within 5 meters from the vehicle; in the LE subdataset, the image source was from the offending vehicle taken by the fixed speed camera on the roadside, and the environment of the road vehicle image was relatively complicated; in the RP subdataset, the image was taken by a handheld camera or a camera mounted on a moving vehicle so that the vehicle image was shot at a random angle and distance. The AOLP license plate photo samples are shown in Table 1. In the PVLP dataset, there were many types of license plate photos, such as images taken by fixed speed cameras, images of roadside vehicles taken manually, and black-and-white photos taken by speed cameras on freeways. Therefore, these license plate photos contained a variety of real environmental factors, such as weather changes, day and night lights, light and shadow reflection, distances of photographing vehicles, image pixels, image quality, and different shooting angles, making PVLP license plate samples quite abundant and diverse. Besides, images taken by fixed speed cameras also included multiple types of vehicles, such as automobiles, locomotives, heavy locomotives, and large vehicles. The PVLP dataset included the subdataset of Fixed Speed Camera Image (FC) with image resolution 1337 × 977 and a total of 11,576 photos, comprising 27 images of four-character license plates, 71 images of five-character license plates, 4047 images of six-character license plates, and 7431 images of seven-character license plates, the subdataset of Taiwan Highway Shot Image (TS) with image resolution 1392 × 1040 and a total of 11,060 photos, comprising 300 images of five-character license plates, 4208 images of six-character license plates, and 6552 images of seven-character license plates, and the subdataset of Roadside Shot Image (RS) with image resolution 614 × 460 and a total of 3071 photos, comprising 4 images of five-character license plates, 1209 images of six-character license plates, and 1858 images of seven-character license plates. The PVLP photo samples are shown in Table 1.

Table 1: Samples of AOLP and PVLP in different types of datasets.

| Dataset | AC | LE | RP |
|---|---|---|---|
| AOLP |  |  |  |
| | FC | TS | RS |
| PVLP |  |  |  |

*4.2. Training for License Plate Location and Recognition.* The training of the DLPR license plate recognition model was to utilize PVLP data to perform a series of model training, such as license plate location and license plate character recognition, using a total of 25,707 image samples containing vehicles and license plates. In the training of the license plate location model, this study input these 25,707 image samples into the license plate location model for training with a total of 250 epochs, set 32 image samples each time to perform training iterations 804 times in each epoch of training, and set the initial learning rate as 0.3. During the training process, the value of classification loss is 4.858 in the initial epoch and converges to 0.469 in the final epoch; the value of bounding box regression loss is 4.342 in the initial epoch and converges to 0.234 in the final epoch; the value of landmark regression loss is 22.24 in the initial epoch and converges to 0.795 in the final epoch, as shown in Figure 4(a). In the training of the license plate character recognition model, this study utilized the trained license plate location model to locate and correct the license plates for the 25,707 samples of PVLP. After the model located the license plates and calculated their correct locations, those samples with IoU values greater than 0.5 would be selected as the samples of license plate character recognition. Therefore, a total of 25,218 license plate snapshot images were collected from PVLP and viewed as training and validation samples for the license plate character recognition model. This study divided the 25,218 license plate snapshot images into two parts: train set and validation set, in a ratio of 7 : 3. In the data analysis of machine learning, 70% of the samples are used for training, and 30% of the samples are used for verification and testing. The accuracy is approximately between 4-fold and 5-fold cross validation. Also, according to Nguyen et al. [44], the research results have revealed that the ratio of 70/30 used for training and testing datasets can provide the proposed models with the best training and verification effect. Although 10-fold cross validation can provide better accuracy, the data ratio of 70/30 is still used for training, verification, and testing because of the complexity of the model in this study. The train set could be regarded as the parameter training of the model. The validation set was viewed as a phased verification for the training result of each epoch, examining the model training result of the current epoch for the fitting ability of the validation set. Furthermore, the fitting ability was used as the judgment of the convergence condition. The training of the license plate character model had a total of 1851 epochs. In each epoch of training, 128 image samples were set each time to perform training iterations 198 times. The initial learning rate was set to 0.1. In the training process, the CTC loss value is 0.274 in the initial epoch, and the model ability is significantly improved when it approaches 200 epochs, in which the loss value is 0.0283; the CTC loss value is 0.0275 when the model finally converges. In the validation process, the CTC loss value is 0.116 in the initial epoch, the loss value is $6.46e-3$ when approaching 200 epochs, and the CTC loss value is $5.92e-3$ when the model finally converges, as shown in Figure 4(b). At this time, this model can achieve a recognition accuracy of 99.00% for the license plate character recognition of the validation set after calculation.

*4.3. Model Performance Evaluation with AOLP and PVLP Datasets.* In order to evaluate the DLPR's capability of license plate location, this study conducted an evaluation of the detection model by calculating precision, recall, and receiver operator characteristic curve (ROC curve). Precision can be used to evaluate how correctly the license plate location model determines the location of the license plate. The higher the precision is, the more accurate the detection model is for determining the location of the license plate. Recall can be used to evaluate how correctly the license plate location model finds the location of the license plate. The higher the recall is, the better the ability of the detection model is for finding the location of the license plate. Receiver operator characteristic curve (ROC) and area under the curve (AUC) can be used to assess whether the license plate location model has sufficient capability to discriminate license plates. Generally speaking, it is presumed that the model has a certain ability of judgment as the values of AUC are greater than 0.5. The experimental results show that the license plate location model in this study uses 25,707 license

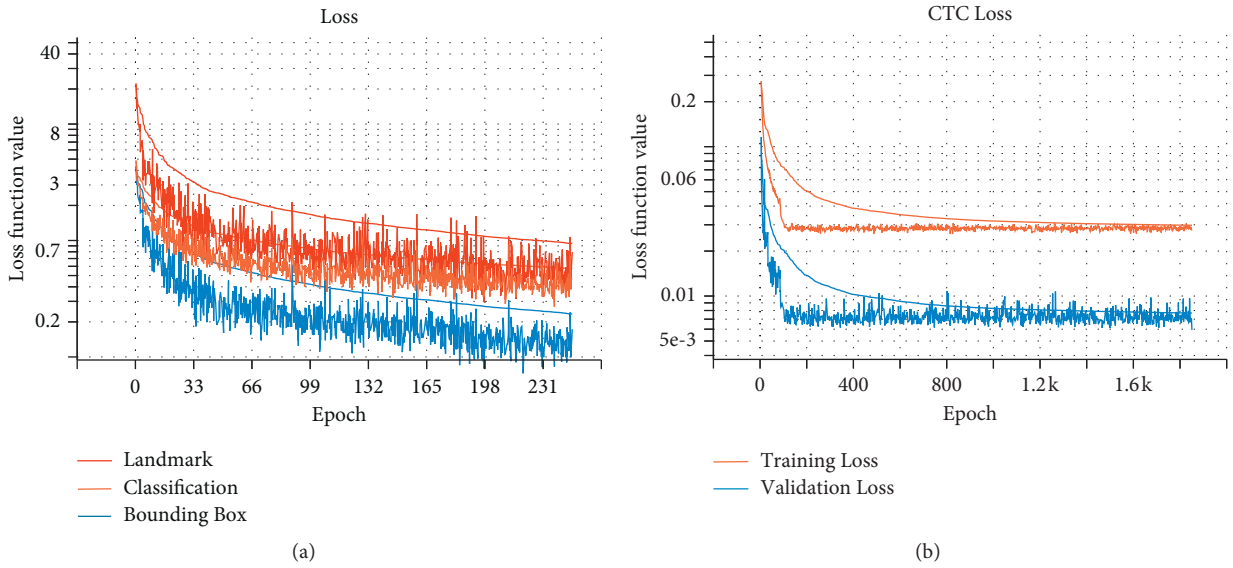



Figure 4: Loss function values of DLPR during the training of license plate location.

plate image samples from the PVLP dataset for the license plate location tests; under the conditions of confidence score threshold = 0.9 and IoU threshold > 0.5, the precision of the license plate location tests is 0.9860 and the recall is 0.9810. Also, 2049 license plate image samples from the AOLP dataset are used for the license plate location tests; under the conditions of confidence score threshold = 0.9 and IoU threshold > 0.5, the precision of the license plate location test is 0.8402 and the recall is 0.8549. Compared with the precision and recall of the PVLP dataset for testing, the precision and recall of AOLP drop significantly. The reason is that PVLP and AOLP have a slight difference in labeled locations of the bounding box of the license plate. Therefore, after the IoU threshold > 0.4 is revised, the experiment is conducted. Then, the precision of the license plate location tests increases to 0.9592, and the recall also rises to 0.9760, as shown in Table 2. In the experiment of measuring DLPR's ability to differentiate between license plates, the ROC curve is displayed in Figure 5, where AUC = 0.98 as the PVLP dataset is used for testing and AUC = 0.94 as the AOLP dataset is used for testing, indicating that they both have good capabilities of discriminating license plates.

In the part of verifying the accuracy of the DLPR license plate character recognition model, the measurement standard is that as long as all the license plate number characters and word sequences on the license plate snapshot images are recognized correctly without redundant characters, the recognition is considered correct; otherwise, it is seen as an error. The accuracy of the license plate character recognition also depends on the license plate location ability of the recognition model as well as the stability of the license plate skew correction. Therefore, when testing the license plate character recognition, this study only selected the license plate snapshot images of IoU > 0.5 between the bounding box predicted by the model and the ground-truth bounding box as samples of verifying the accuracy of the license plate character recognition model. The experimental results are demonstrated in Table 3. DLPR has a very good recognition level for the captured images of the license plates as long as the license plate location is accurate enough, that is, IoU > 0.5, and the "-" character among the license plate characters can also be correctly recognized. However, in the AOLP dataset, the labeled content of the license plate does not contain the "-" character. Therefore, this study removed the "-" character from the predicted license plate characters and then compared it with the labeled content of the license plate provided by AOLP. The results show that all the recognitions can be completely correct under the condition of IoU > 0.5. Conversely, in the tests of the PVLP dataset, the "-" character is taken into consideration. After the labeled content of the license plate provided by PVLP is compared, the average of accuracy can reach 99.88%.

Considering the accuracy of the complete license plate recognition, DLPR needs to go through the complete computational process, including license plate location, license plate skew correction, and license plate character recognition, to figure out the license plate number. After the number is compared with the labeled content of the license plate, the accuracy of license plate recognition can be calculated. In addition, the frame per second (FPS) is used to measure the speed of license plate recognition. The higher the FPS value is, the faster the calculation speed of recognition is. According to the experimental results, as shown in Table 4, when the PVLP dataset is applied to the license plate location and license plate recognition, the average of accuracy is 97.56%, and the speed of recognition is FPS > 21. However, as to the license plate sample of RS, the recognition rate is only 94.06% since the environmental conditions of shooting on mobile phones are more complicated. Overall, the DLPR license plate recognition model proposed by this study has been equipped with the ability of real-time license plate recognition in real environments. In consequence, it has a variety of values of practical applications, such as its capability of helping the data processing industry

Table 2: Precision and recall of license plate location tests with different threshold (TH).

| Dataset | Confidence score TH | IoU TH | Precision | Recall |
|---|---|---|---|---|
| PVLP | 0.9 | >0.5 | 0.9860 | 0.9810 |
| AOLP | 0.9 | >0.5 | 0.8402 | 0.8549 |
| AOLP | 0.9 | >0.4 | 0.9592 | 0.9760 |

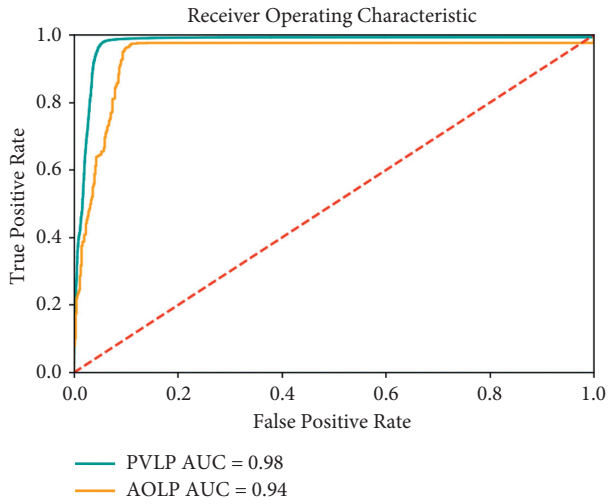



Figure 5: ROC curve of DLPR license plate location model.

Table 3: Accuracy (ACC) measurement of license plate character recognition to the proposed model with AOLP and PVLP dataset.

| IoU threshold | AOLP | | | | | | PVLP | |
|---|---|---|---|---|---|---|---|---|
| | AC | | LP | | RP | | | |
| | Number | ACC (%) | Number | ACC (%) | Number | ACC (%) | Number | ACC (%) |
| 0.5 < IoU < 0.75 | 476 | 100 | 455 | 100 | 291 | 100 | 2637 | 99.92 |
| 0.75 < IoU < 0.9 | 38 | 100 | 158 | 100 | 247 | 100 | 12024 | 99.83 |
| 0.9 < IoU | 0 | *X* | 30 | 100 | 14 | 100 | 10558 | 99.91 |

Table 4: Average accuracy (Avg. ACC) and speed (FPS) of DLPR using the PVLP dataset for license plate recognition.

| Method | PVLP | | | Avg. ACC (%) | FPS |
|---|---|---|---|---|---|
| | FC (%) | TS (%) | RS (%) | | |
| DLPR | 99.19 | 99.43 | 94.06 | 97.56 | >21 |

Table 5: Average accuracy (Avg. ACC) and speed (FPS) of DLPR using the AOLP dataset for license plate recognition.

| Method | AOLP | | | Avg. ACC (%) | FPS |
|---|---|---|---|---|---|
| | AC (%) | LE (%) | RP (%) | | |
| Li et al. [40] | 94.85 | 94.19 | 88.38 | 92.47 | 0.66 |
| Li et al. [41] | 95.59 | 96.43 | 83.80 | 91.94 | 2.5 |
| Björklund et al. [42] | 94.60 | 97.80 | 96.90 | 96.43 | 39.21 |
| DLPR (proposed) | 96.91 | 98.01 | 98.20 | 97.70 | 62.93 |

enhance the accuracy of license plate recognition in photos of traffic violations as well as the performance of traffic service operations.

DLPR uses the AOLP dataset to conduct a thorough experiment on license plate location and recognition and compares it with its related research, as shown in Table 5. The experimental results show that the accuracy of DLPR recognition performs the best in the categories of AC, LE, and RP. The license plate recognition method proposed by scholars Li et al. [45, 46] merged license plate location and character recognition into the same major neural network. Its license plate character recognition phase is similar to the architecture applied in this study, which is based on the end-to-end segmentation-free character recognition method. Accordingly, it has good performances in datasets of AC and LE, whereas it has a poor performance in the RP dataset since the license plate skew correction is not adopted. The framework of license plate recognition proposed by Björklund et al. [47] included the processing of license plate skew correction. In the character recognition stage, the character recognition based on the segmentation method was adopted. Therefore, in the tests of the AOLP dataset, its performance of accuracy was good. As to the DLPR framework proposed by this study, it has a stable and accurate prediction performance for its capabilities of license plate location and license plate skew correction. Also, it is

integrated with CRNN's segmentation-free character recognition method to achieve the best recognition accuracy in the tests of the AOLP dataset. Additionally, under the condition of image resolution 640 × 640, the speed of license plate recognition can reach 62.93 FPS.

## 5. Conclusions

In terms of research contribution, after the Dual-stage License Plate Recognition Model (DLPR) proposed by this study used the PVLP dataset provided by company A in Taiwan's data processing industry to train license plate location and license plate character recognition, the final accuracy of license plate recognition could reach 97.56% in the PVLP dataset, and FPS > 21. Besides, this study also performed tests on the AOLP dataset and compared it with its related research; the results indicated that the license plate accuracy for all of the AOLP subdatasets gathered in the DLPR model was the best, with an average of 97.70%, and FPS could reach 62.93. Consequently, the DLPR model can be applied to the license plate recognition of the real-time image stream in the future and assist Taiwan's data processing industry in improving not only the accuracy of license plate recognition in photos of traffic violations but also the performance of traffic service operations. In this study, DLPR adopted three major processes: license plate location, skew correction, and character recognition. In particular, the character recognition applied an end-to-end segmentation-free character recognition method based on CRNN, having good recognition accuracy for the precisely located and skew-corrected license plate snapshot images. Therefore, when it was practically applied to the license plate recognition using cellphone cameras, it could correctly recognize license plate characters from the license plate images with different angles.

Still, there is a research limitation in this study. This study used the undisclosed PVLP dataset to train the license plate recognition model as well as the AOLP dataset to test the license plate recognition model. In the AOLP dataset, a team was established to record and classify license plates in various real environments according to specific criteria, and these detailed parameters were sufficient enough to allow the license plate recognition model to improve its ability of license plate recognition in some specific environments. Nonetheless, the current PVLP dataset cannot provide detailed parameters related to the license plate samples, and it can only provide three subdatasets for the preliminary classification. Hence, the establishment of a license plate dataset with detailed classification parameters will be of great help to the study of license plate recognition. Future research on this field will help the PVLP dataset construct a complete record of license plate parameters and focus on the study of license plate recognition in more complex environments.

## Data Availability

Data are available upon request to the authors. The data source is obtained from the questionnaire analysis of the author's research.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

[1] J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, "Automated license plate recognition: a survey on methods and techniques," *IEEE Access*, vol. 9, pp. 11203–11225, 2021.

[2] H. Chandra, Michael, K. R. Hadisaputra, H. Santoso, and E. Anggadjaja, "Smart parking management system: an integration of RFID, ALPR, and WSN," in *Proceedings of the 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS)*, pp. 1–6, Bangkok, Thailand, August 2017.

[3] P. Agarwal, K. Chopra, M. Kashif, and V. Kumari, "Implementing ALPR for detection of traffic violations: a step towards sustainability," *Procedia Computer Science*, vol. 132, pp. 738–743, 2018.

[4] J. Liu, Y. Feng, and H. Wang, "Facial expression recognition using pose-guided face alignment and discriminative features based on deep learning," *IEEE Access*, vol. 9, pp. 69267–69277, 2021.

[5] D. Gonzalez Dondo, J. A. Redolfi, R. G. Araguas, and D. Garcia, "Application of deep-learning methods to real time face mask detection," *IEEE Latin America Transactions*, vol. 19, no. 6, pp. 994–1001, 2021.

[6] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, "Segmentation- and annotation-free license plate recognition with deep localization and failure identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2351–2363, 2017.

[7] X. Tao, F. Gu, and S. Xie, "The network design of license plate recognition based on the convolutional neural network," in *Proceedings of the 11th International Symposium on Artificial Intelligence Algorithms and Applications*, pp. 749–758, Guangzhou, China, November 2019.

[8] X. Tao, L. Li, and L. Lu, "A lightweight convolutional neural network for license plate character recognition," *Artificial Intelligence Algorithms and Applications*, vol. 1205, pp. 379–387, 2019.

[9] W. Wang, J. Yang, M. Chen, and P. Wang, "A light CNN for end-to-end car license plates detection and recognition," *IEEE Access*, vol. 7, pp. 173875–173883, 2019.

[10] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[11] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 369–376, Pittsburgh, PA, USA, July 2006.

[12] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[13] G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung, "Application-oriented license plate recognition," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 552–561, 2013.

[14] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proceedings of the 6th*

*International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pp. 555–562, Bombay, India, 1998.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, Honolulu, HI, USA, July 2005.

[16] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[18] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[19] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation*, California Univ. San Diego La Jolla Inst for Cognitive Science, San Diego, CA, USA, 1985.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[23] C. H. Wang, "An intuitionistic fuzzy set-based hybrid approach to the innovative design evaluation mode for green products," *Advances in Mechanical Engineering*, vol. 8, pp. 1–16, 2016.

[24] K.-P. Lin, H.-F. Chang, T.-L. Chen, Y.-M. Lu, and C.-H. Wang, "Intuitionistic fuzzy C-regression by using least squares support vector regression," *Expert Systems with Applications*, vol. 64, pp. 296–304, 2016.

[25] L.-L. Li, J. Sun, C.-H. Wang, Y.-T. Zhou, and K.-P. Lin, "Enhanced Gaussian process mixture model for short-term electric load forecasting," *Information Sciences*, vol. 477, pp. 386–398, 2019.

[26] K. Wang and M. Z. Liu, "Object recognition at night scene based on DCGAN and faster R-CNN," *IEEE Access*, vol. 8, pp. 193168–193182, 2020.

[27] X. Zhou, Y. Li, and W. Liang, "CNN-RNN based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2021.

[28] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[29] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Venice, Italy, October 2017.

[30] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "Scaled-YOLOv4: scaling cross stage partial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13029–13038, Nashville, TN, USA, 2021.

[31] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot MultiBox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Amsterdam, The Netherlands, October 2016.

[32] R. Vaillant, C. Monrocq, and Y. Le Cun, "Original approach for the localisation of objects in images," *IEE Proceedings-Vision, Image, and Signal Processing*, vol. 141, no. 4, pp. 245–250, 1994.

[33] I. Astawa, I. Caturbawa, I. Sajayasa, and I. M. Atmaja, "Detection of license plate using sliding window, histogram of oriented gradient, and support vector machines method," *Journal of Physics: Conference Series*, vol. 953, Article ID 12062, 2018.

[34] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, Columbus, OH, USA, June 2014.

[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[37] C. Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, *DSSD: Deconvolutional Single Shot Detector*, https://arxiv.org/abs/1701.06659, 2017.

[38] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, *RetinaFace: Single-Stage Dense Face Localisation in the Wild*, https://arxiv.org/abs/1905.00641, 2019.

[39] L. Zhao, Q. Li, C. H. Wang, and Y. C. Liao, "3D brain tumor image segmentation integrating cascaded anisotropic fully convolutional neural network and hybrid level set method," *Journal of Imaging Science and Technology*, vol. 64, Article ID 040411, 2020.

[40] K. Bai, Q. Li, and C. H. Wang, "Integrating improved U-Net continuous maximum flow algorithm for 3D brain tumor image segmentation," *Journal of Imaging Science and Technology*, vol. 64, Article ID 040412, 2020.

[41] A. G. Howard, *Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, https://arxiv.org/abs/1704.04861, 2017.

[42] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, Honolulu, HI, USA, July 2017.

[43] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "SSH: Single stage headless face detector," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4885–4894, Venice, Italy, October 2017.

[44] Q. H. Nguyen, H. B. Ly, L. S. Ho et al., "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil," *Mathematical Problems in Engineering*, vol. 2021, Article ID 4832864, 2021.

[45] H. Li, P. Wang, M. You, and C. Shen, "Reading car license plates using deep neural networks," *Image and Vision Computing*, vol. 72, pp. 14–23, 2018.

[46] H. Li, P. Wang, and C. Shen, "Toward end-to-end car license plate detection and recognition with deep neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1126–1136, 2019.

[47] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, "Robust license plate recognition using neural networks trained on synthetic images," *Pattern Recognition*, vol. 93, no. 1, pp. 134–146, 2019.