# Improving Warped Planar Object Detection Network For Automatic License Plate Recognition

**Final Year Project Final Report**

**Nguyen Dinh Tra – HE140061**

**Nguyen Cong Tri – HE141519**

**Vu Thanh Trung – HE130284**

A thesis submitted in part fulfilment of the degree of BSc. (Hons.) in Computer Science with the supervision of Assoc.Prof.. Phan Duy Hung.

Bachelor of Computer Science

Hoa Lac campus - FPT University

28 December 2022

## Acknowledgments

*This thesis is dedicated to Assoc. Prof. Hung, whose invaluable guidance, ongoing support, and financing enabled this study to attain its full potential as of today.*

# DECLARATION

This thesis is the result of own work and includes nothing, which is the outcome of work done in collaboration except where specifically indicated in the tex. It has not been previously submitted, in the part or whole, to any university of institution for any degree, diploma, or other qualification.

Signed: _____ Trà _________ Trí _______ Trung ________

Date: ___12/07/2022_____________________________________________

Nguyen Dinh Tra, Nguyen Cong Tri, Vu Thanh Trung and qualifications

FPT University

# ABSTRACT

This thesis aims to improve the Warping Planer Object Detection Network (WPOD-Net) using feature engineering to increase accuracy. What problem are solved using the Warping Object Detection Network using feature engineering? More specifically, we think that it makes sense to add knowledge about edges in the image to enhance the information for determining the license plate contour of the original WPOD-Net model. The Sobel filter has been selected experimentally and acts as a Convolutional Neural Network layer; the edge information is combined with the old information of the original network to create the final embedding vector. The improvement was compared with the original model on a set of data that we collected for evaluation. The results are evaluated through the Quadrilateral Intersection over Union value and demonstrate that the model has a significant improvement in performance.

**Keywords:** Convolutional Neural Network, WPOD-Net, License plate, Edge Detection, Sobel filter

# Table of Contents

# List of Table

## List of Figures

## List of abbreviations and acronyms

| Abbreviations | Meaning |
|---|---|
| AI | Artifical Intelligence |
| YOLO | You Only Look Once |
| CNN | Convolution Neural Network |
| qIOU | Quadrilateral Intersection over Union |
| WPOD-net | Wraped Planar Object Detection |
| AOLP | Automatic License Plate Recognition |
| SSD | Single Shot Multibox Detector |

# 1 Introduction

## 1.1 Background

Convolutional Neural Network (CNN) is technically deep learning used effectively in feature extraction of images, widely use in Computer Vision to solve problems: Object Detection [1], Image segmentation [2], Recognition [3], Tracking [4] and Alignment [5] and has significant performance compared to using traditional machine learning. This thesis focuses on real-life license plate detection and recognition. One primary approach when it comes to domain transfer problems is using Warping Planer Object Detection Network [6].

The problem of license plate (LP) detection is not a new problem with many different methods, but when applying the CNN model, specifically YOLO [7] in the Object Detection problem, the accuracy increases compared to using machine learning. However, there is a limitation of this method that Object Detection only results in the area containing the license plate. If the license plate is at a non-front angle, the results can affect the extraction of license plate information.


**Figure 1**: **License plate detection results using YOLO on image**

In the Figure 1, we can see that in the case of the front view (left image), the detected number plate area has almost no details of the body of the car. In the case of a front view and slightly tilted (right image), the detected license plate area includes quite a lot of excess images of the body.

In order to effectively detect objects such as license plates above when the input image is tilted too much, using WPOD-net is the first choice to detect the license plate area. But there are some limitations when using WPOD-Net as some license plates are missing the label if the license plate is small. This study gives an idea to improve that machine learning model architecture by adding information to the embedding vector. The edge information will be added as a CNN layer and the results will be compared with the original model on the benchmark data.

## 1.2 Related works

### 1.2.1 Complete ALPR Methods

Usually, the processing flow of the methods is described as shown below:

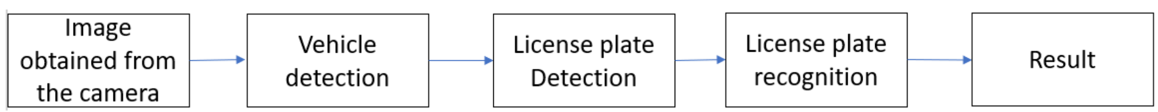



**Figure 2: Pipeline of Automatic License Plate Recognition**

In this pipeline, the image will be acquired from the camera and then using vehicle detection, we use the Yolov2 model for processing. Then detect this license plate from the vehicles transmitted from the previous step using WPOD-net to process for the task the number plates are tilted to convert to the front angle and this is our improvement step in the thesis. this. Then recognize the license plate by using YOLO as an OCR to recognize the character and store it in a database on the system. Looking at the picture above, we can see that to solve the problem of recognizing license plates, we have to solve three subproblems:

#### 1.2.1.1 Vehicle detection

There are now two types of vision-based media item recognition systems: standard machine vision methods and complicated deep learning approaches. Traditional machine vision approaches isolate the media from the fixed backdrop picture by allowing it to move. This approach may be classified into three types: background subtraction, continuous video frame difference, and optical flow. Convolutional neural networks (CNN) have shown to be extremely effective in the identification of vehicle objects. CNN can learn image characteristics and perform several related tasks, such as classification and bounding box regression. Methods of detection are often classified into two types. The two-stage technique builds an item's candidate box using multiple methods and then classifies the object using a convolutional neural network. The single-stage technique does not generate a candidate box, but instead translates the bounding box positioning problem straight into a regression problem for processing. The most important single-stage approaches Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO). In which we use Yolov2 model to apply in this pipeline to detect vehicles.

#### 1.2.1.2 License plate detection

The detection of license plates is the initial stage in the challenge of license plate recognition. It is to identify the location of the license plate from a photo that includes the backdrop, vehicle, and license plate, and then pick the proper bounding box to locate the license plate. There are two types of license plate detecting systems: classic approaches and neural network model-based methods. Traditional approaches are based on the license plate's size, shape, texture, color, and other features, which are commonly classified into four categories: edge feature [13], text feature [14], color feature [15], character feature [16], and combination of two or more [17]. Color and edge are two of the most commonly utilized qualities. With the fast growth of deep learning in image target recognition in recent years, the neural network target detection model has been widely employed in license plate detection. The deep learning-based license plate identification approach involves learning and training the pre-designed deep neural network on a huge amount of license plate data to create the model capable of license plate detection such as use of YOLO which returns the license plate position from the image.

The license plate image will be distorted in the process of obtaining the license plate image due to the influence of the shooting angle, lens, and other objective factors, such as horizontal tilt, vertical tilt, or trapezoid distortion, which causes difficulties in the subsequent recognition processing. If the license plate is rectified after detection, the noise such as the license plate border is removed, making it more favourable to character identification and improving the accuracy of following license plate recognition. Currently popular approaches for license plate rectification include Hough transformation [18], Principal Component Analysis (PCA) [19], straight line fitting (SLF) [20], and least variance [21] or can use Deep Learning methods.

Sergio et al. designed WPOD-Net based on the idea of YOLO, SSD (Single Shot MultiBox Detector) [8], STN (Spatial Transformer Network) [9]. As mentioned above YOLO and SSD only return 1 bounding box in the license plate regardless of the surrounding space. STN can detect non-rectangular regions, but it cannot handle multiple transformations at the same time, instead performing a single spatial transformation across the entire input. WPOD-Net return bounding box area surrounds the license plate and brings the number plate to the front view.

In the thesis published in 2021: “A Flexible Approach for Automatic License Plate Recognition in Unconstrained Scenarios” [10], Sergio et al. present a method to improve WPOD model by add more convolutional layer in the end of model.

#### 1.2.1.3 License plate recognition

Character recognition is the final module of a license plate identification system, and it is a standard image recognition issue. Currently, the most common car license plate identification algorithms are classifier-based character recognition algorithms such as artificial neural networks, SVM, fuzzy classifiers, and so on. Character segmentation is frequently required prior to license plate identification, and then optical character recognition technology is utilized to recognize each segmented character. Character segmentation's goal is to properly segment the characters in the text area one by one before sending them. The essential starting point in the field of license plate character segmentation is that when the license plate is rectified, the characters included in the license plate are largely distributed on a horizontal line. These characters are vertically projected, and individual characters are split based on the projected value's crest and trough. However, due to shooting angle, light, and other issues, license plate correction and segmentation become more difficult when the license plate is tilted at a large angle, and accuracy also decreases, making the traditional method difficult to accurately segment characters, affecting the accuracy of character recognition. In which we use Yolov2 model to apply in this pipeline to detect vehicles.

### 1.2.2 Edge Detection

Vincent et al. in the thesis "On Edge Detection" [11] present a commonly used digital image processing technique to find the contours of an image object. The author explained edge point: a pixel is considered an edge point if there is a rapid or sudden change in gray level (or color). Boundary: a set of consecutive boundary points. Edge detection is an image processing technique used to find the edges of objects in an image or can understand finding areas with a continuous loss in brightness (areas where there is a sharp difference in brightness).

It is the basic stage in our experiment system for identifying locations in a document picture when image brightness varies formally or sharply while keeping the image's important structural qualities. The steps we must take to identify the edge are as follows: the first stage is smoothing, which includes filtering the picture to decrease noise. The filter will be used to increase the quality of the picture edges in the second stage. The third phase is detection, in which all edge points are retrieved and then the noise-causing edge pixels are dropped. And last, localisation, which certifies the position of an edge. We have edge detection methods for extracting edges from images, such as vertical, horizontal, and so on. The quality of edges recognized by these approaches is heavily influenced by noise, lighting conditions, objects of similar intensities, and edge density. Because our DL is built on learnable filters to identify features, we will utilize Sobel and Canny edge detection techniques as a kernel filter to detect edge features from license detection.

#### 1.2.2.1 Sobel Edge Detection

In another thesis published in 2009: “A Descriptive Algorithm for Sobel Image Detection” [12], Vincent et al. use a Gaussian filter to remove noise, smooth the image first to make the edge detection algorithm work better. The Sobel edge detector employs two 3x3-sized masks, one of which estimates the slope in the x direction and the other the slope in the y direction (see figure 3). One-pixel square at a time is adjusted by sliding the mask over the picture. From sunrise to night, an algorithm determines the gradient of picture intensity at each location and then indicates the direction of image enhancement at each point. Strong contrast darker or brighter patches can be seen around edges. The Sobel convolution particle is made to react to edges that are both vertically and horizontally present. Each of these masks is put together with an image. It determines the absolute magnitude of the slope at each position by first calculating the horizontal and vertical slopes (Gx and Gy), then combining them.

| | | |
|---|---|---|
| -1 | 0 | +1 |
| -2 | 0 | +2 |
| -1 | 0 | +1 |

Gx

| | | |
|---|---|---|
| +1 | +2 | +1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

Gy

**Figure 3: 3x3 Sobel edge detection masks**

Figure 4 depicts the convolution method used to generate a feature map from input data using a convolution filter.

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Input

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

Filter/Kernel

| | | | | |
|---|---|---|---|---|
| 1x1 | 1x0 | 1x1 | 0 | 0 |
| 0x0 | 1x1 | 1x0 | 1 | 0 |
| 0x1 | 0x0 | 1x1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Convolution

| | | |
|---|---|---|
| 4 | | |
| | | |
| | | |

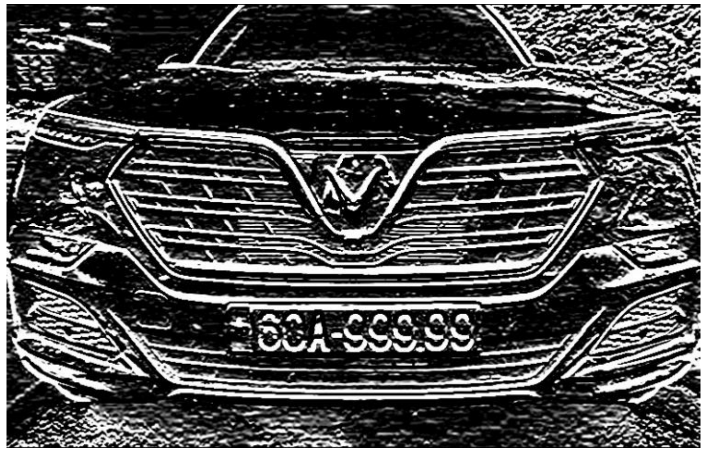
Result

**Figure 4**: **Visualization convolution kernel**

We can observe the results when using Sobel Edge Detection in Table 1:

**Table 1**: **Result use Sobel Edge Filter**

| | |
|---|---|
| Origin |  |
| SobelX |  |

| | |
|---|---|
| SobelY |   |
| Combining SobelX and SobelY |   |

#### 1.2.2.2 Canny Edge Detection

Ding et al. published in 2001: “On the Canny edge detector” [13]. In computer vision, the Canny edge detector is commonly used to identify abrupt intensity changes and object boundaries in images. The Canny edge detector considers a pixel to be an edge if its gradient magnitude is greater than that of pixels on both of its sides in the direction of highest intensity change.

The Canny edge detector stages conduct picture pre-processing and noise reduction. First, we must choose the output image, which can be in color or grayscale. If the image is in color, convert it to grayscale. After the image is captured, it must be smoothed by eliminating noise. A median filter is used to reduce noise in the picture input. The Euclidean distance is employed here to improve the image's edge. To do this, a filter is created that takes into account both the vertical and horizontal axis directions. The filter initially decides the size of the filter template based on the template's square distance from the center pixel, from tiny to big. The second stage is to detect the gradient in an image once the noise has been removed. The picture gradient is detected because there is a substantial shift in grayscale of an image's edge. The final step is to get maxima following derivation. The canny detector's final phase is hysteresis thresholding.

We can observe the results when using Canny Edge Detector in Table 2

**Table 2: Result use Canny Edge Detector**

| | |
|---|---|
| Origin |  |
| Canny Edge |  |

# 2 Methodology

In this thesis, we focus on improving the accuracy of license plate detection task in Complete ALPR Methods

This section presents the improved architecture, loss function and evaluation metric. The model is then used for license plate detection. Using image processing algorithms to find edges helps us to extract more information from the image. We can consider the Sobel filter as a special layer in the CNN model, more specifically the WPOD-Net that increases the accuracy of the License Plate detector.

## 2.1 The Architecture

The WPOD-net architecture is improvement using insights from [7-9]. The Figure 5 shows how the WPOD-Net works. From input image through the forward process, we get output features maps include: 8 channels in which the first 2 channels are probability with/without license plate and remaining 6 channels are parameters to calculate the transformation matrix. To extract the license plate area, the authors consider a fixed size square part with white border around the cell in the output features map. If the probability of the cell's object is greater than threshold then the values of the cell's remaining 6 channels will calculate the transform matrix from square part to license plate area. The matrix can be used to bring the license plate to the front view. From this idea, the authors give the WPOD network architecture.

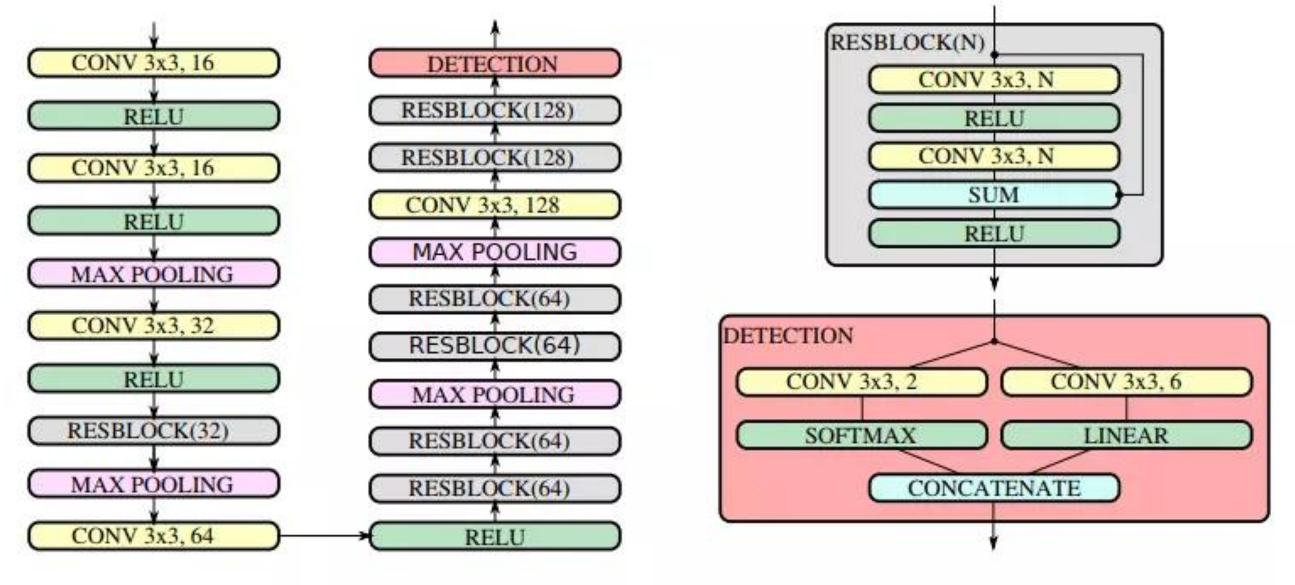



**Figure 5: WPOD Network Architecture**

From the above model, with the application of an edge detection algorithm in image processing, it is possible to extract more image information, increasing the performance of the model shown through Figure 6.

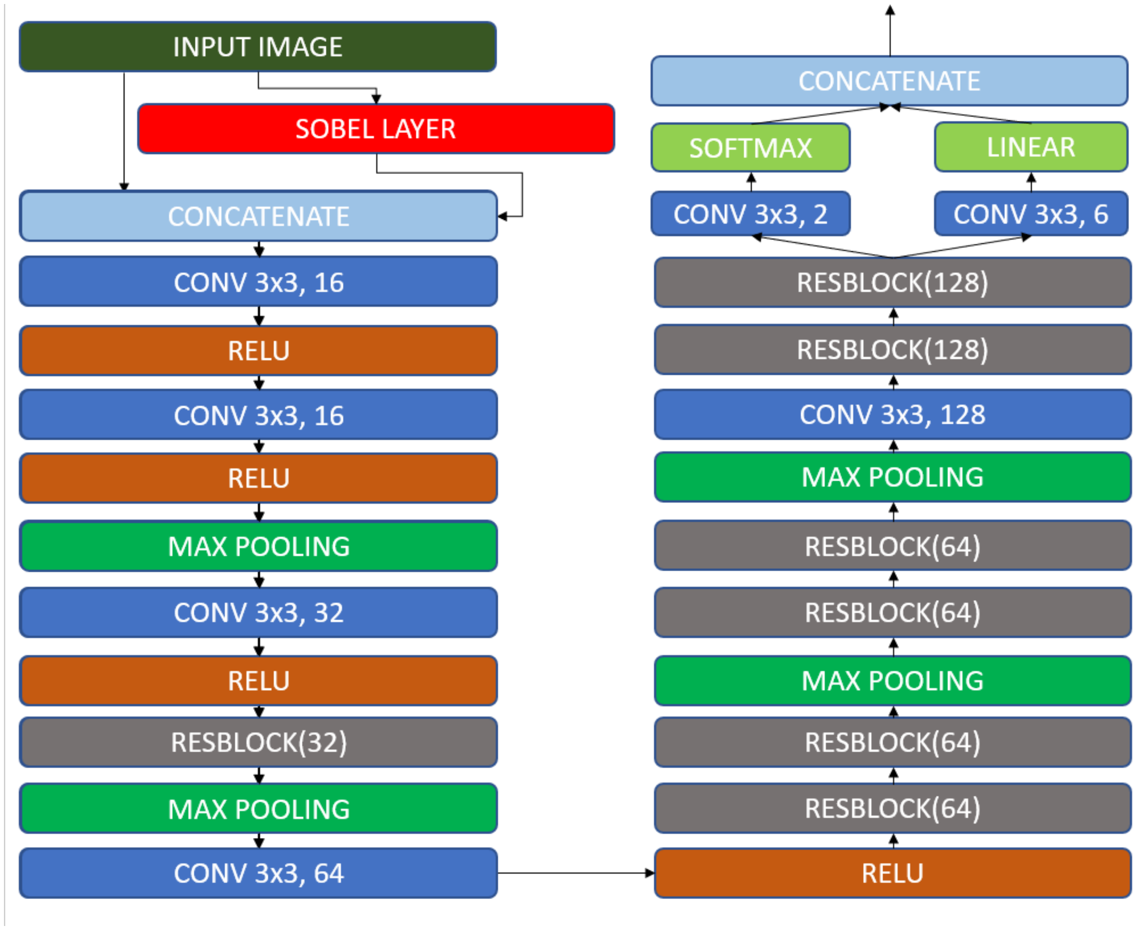



**Figure 6: The modification model**

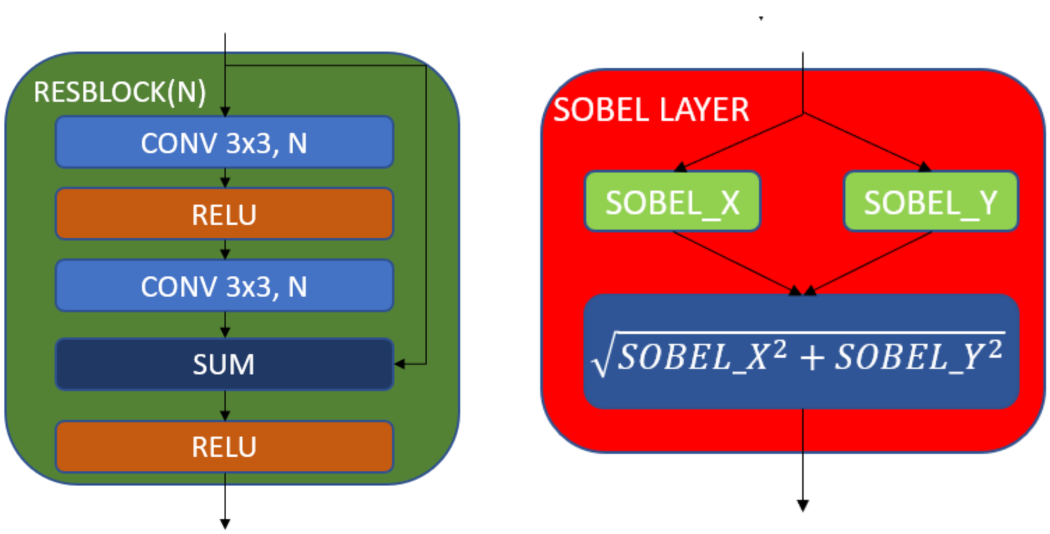



**Figure 7: Resblock(N) and Sobel Layer in the modification model**

The operator computes approximate derivatives using two $3 \times 3$ kernels, one for horizontal changes and the other for vertical changes, which are convolved with the original picture. If we consider $\mathbf{G}_x$ and $\mathbf{G}_y$ to be two pictures that, at each location, include the horizontal and vertical derivative approximations, respectively, and designate $\mathbf{A}$ as the source image, the calculations are as follows:

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \text{ and } \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Where the 2-dimensional signal processing convolution operation is shown by the symbol * here. The Sobel kernels compute the gradient with smoothing because they can be broken down into the sum of an averaging kernel and a differentiation kernel. For example, $\mathbf{G}_x$ and $\mathbf{G}_y$ can be written as:

$$\mathbf{G}_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \left( \begin{bmatrix} +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \right) \text{ and } \mathbf{G}_y = \begin{bmatrix} +1 \\ 0 \\ -1 \end{bmatrix} * \left( \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \mathbf{A} \right)$$

Here, rising in the "right" direction is defined as the x-coordinate, while increasing in the "down" direction is described as the y-coordinate. The gradient magnitude may be calculated by combining the gradient approximations at each place in the picture using:

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \text{ (SOBEL\_XY in Fig. 7)}$$

## 2.2 Loss function

In this work, the mean-squared-error (MSE) loss is used to estimate the error between a warped version of the canonical square and the normalized annotated points of the LP. The binary-cross-entropy (BCS) loss is used to handle the probability of having/not having an object at each pixel in final feature map.

For an input image with height $H$ and width $W$, and network stride given by $N_s = 2^4$ (four max pooling layers), the network output feature map consists of an $M \times N \times 8$ volume, where $M = H/N_s$ and $N = W/N_s$. For each point cell $(m, n)$ in the feature map, there are eight values to be estimated: the first two values ($v_1$ and $v_2$) are the object/non-object probabilities, and the last six values ($v_3$ to $v_8$ ) are used to build the local affine transformation $T_{mn}$ given by:

$$T_{mn}(\boldsymbol{q}) = \begin{bmatrix} max(v_3, 0) & v_4 \\ v_5 & max(v_6, 0) \end{bmatrix} \boldsymbol{q} + \begin{bmatrix} v_7 \\ v_8 \end{bmatrix},$$

where the max function used for $v_3$ and $v_6$ was adopted to ensure that the diagonal is positive (avoiding undesired mirroring or excessive rotations).

To match the network output resolution, the points $\boldsymbol{p}_i$ are re-scaled by the inverse of the network stride, and re-centered according to each point $(m, n)$ in the feature map. This is accomplished by applying a normalization function

$$A_{mn}(\boldsymbol{p}) = \frac{1}{\alpha}\left(\frac{1}{N_s}\boldsymbol{p} - \begin{bmatrix} n \\ m \end{bmatrix}\right),$$

where $\alpha$ is a scaling constant that represents the side of the fictional square? We set $\alpha = 7.75$, which is the mean point between the maximum and minimum LP dimensions in the augmented training data divided by the network stride.

With $T_{mn}(\boldsymbol{q}_i)$ and $A_{mn}(\boldsymbol{p}_i)$ are defined above, location loss as follow:

$$f_{location}(m, n) = \sum_{i=1}^{4} (T_{mn}(\boldsymbol{q}_i) - A_{mn}(\boldsymbol{p}_i))^2$$

Confidence loss as follow:

$$f_{\text{confidence}}(m, n) = \log\text{loss}(\mathbb{I}_{obj}, v_1) + \log\text{loss}(1 - \mathbb{I}_{obj}, v_2)$$

Total loss function is given by a combination of location loss and confidence loss:

$$\text{Total\_loss} = \sum_{m=1}^{M} \sum_{n=1}^{N} \left[\mathbb{I}_{obj} f_{location}(m,n) + f_{\text{confidence}}(m,n)\right]$$

where $\mathbb{I}_{obj}$ is the object indicator function that returns 1 if there is an object at point (m, n) or 0 otherwise?

## 2.3 Evaluation metric

The Quadrilateral Intersection over Union (qIoU) is used to evaluate the accuracy of the detection model.

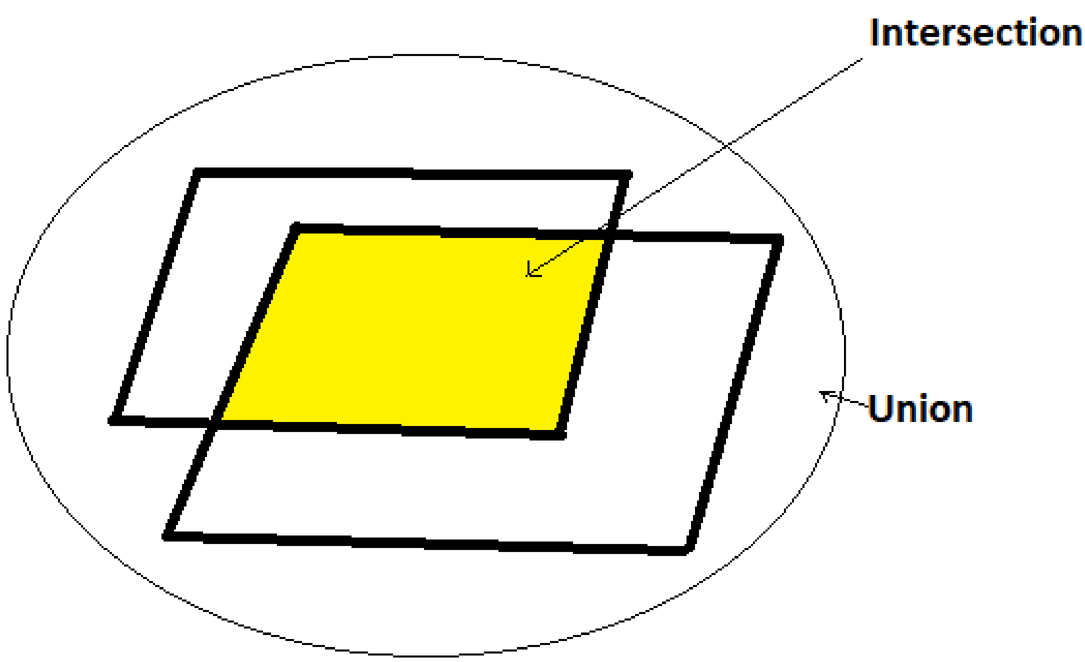



**Figure 8: Quadrilateral Intersection over Union(qIoU)**

qIOU metric as follow:

$$\text{qIOU} = \frac{Intersection}{Union}$$

In the Figure 8, we can see how to calculate qIoU, it is the ratio between the intersection and union of the predict box and the ground truth box. A high qIOU indicates that the detection result has a high degree of agreement with ground truth. We use this metric to compare WPOD-net and our improvement method.

We also want to answer the question” Does a better license plate detection model increase the accuracy of complete ALPR Methods?”.

To answer the above question, we use accuracy to evaluate complete ALPR Methods after replacing the WPOD-net with our improvement model.

Here is confusion matrix:

| | Predicted 0 | Predicted 1 |
|---|---|---|
| Actual 0 | TN | FP |
| Actual 1 | FN | TP |

**Figure 9: Confusion matrix**

Accuracy metric as follow:

$$accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}}$$

# 3 Implementation & Evaluation

## 3.1 Implement

This CNN model is implemented in TensorFlow [22]. The trained models in experiments are available for evaluation of methods. Whole experiments were performed on an NVIDIA GeForce RTX 3090 GPU.

The model is trained by using a per-batch training strategy and use Adam’s algorithm with learning rate 10-3, batch size 64 for 300000 iterations.

In each batch, we randomly use augment data. The following augmentation transforms are used:

- Rectification
- Aspect-ratio
- Centering
- Scaling
- Rotation
- Mirroring
- Translation
- Cropping
- Colorspace
- Annotation

# 3.2 Dataset

## 3.2.1 Data for training and testing our improvement model

We create a training dataset of 363 images, including some images taken from the AOLP dataset [23], some data of Vietnamese car and motorbike license plates we collected from Internet. Specifically, we have: 50 images from the AOLP dataset, 105 images of cars, and 208 images of motorbikes. An independently generated test data, 175 images include: 25 images from the AOLP dataset, 50 images of cars and 100 images of motorbikes.

For each image, we manually annotated the 4 corners of the LP in the picture. A few samples are shown in the Figure 10.


**Figure 10: Examples of the LPs in the training dataset**

## 3.2.2 Data for testing complete ALPR method with our improvement model

### 3.2.2.1 AOLP dataset

There are 2049 images of Taiwan license plates in the application-oriented license plate (AOLP) benchmark database. Access control (AC), which contains 681 samples, traffic law enforcement (LE), which contains 757 samples, and road patrol (RP), which contains 611 samples, are the three subsets that make up this database. We only use RP subsets for testing complete ALPR method. RP refers to situations where a camera is mounted on a patrol vehicle and pictures are taken from arbitrary angles and distances. A few samples are shown in the Figure 11.


**Figure 11: Examples of the AOLP-RP dataset**

### 3.2.2.2 OpenALPR dataset

OpenALPR benchmark database has 223 images of license plates. This database is categorized into two subsets: EU license plates with 108 samples, and Brazil license plates with 115 samples. We try to evaluate the data in many different locations and environments to demonstrate the high applicability of our improvement model. A few samples are shown in the Figure 12.


**a) EU licence plates**


**b) Brazil licence plates**

**Figure 12: Examples of the OpenALPR dataset**

## 3.3 Comparison to Other Methods

Model performance is compared to WPOD-net. Two metrics used for evaluation are qIoU in our test-set and accuracy in OpenALPR and AOLP RP dataset.

**Table 3: Quantitative evaluation**

| **Metric** | | qIoU | Accuracy | | |
|---|---|---|---|---|---|
| **Dataset** | | our test-set | OpenALPR EU | OpenALPR BR | AOLP RP |
| **Model** | WPOD-net | 84.81% | *92.59%* | 92.10% | 92.30% |
| | Our | **85.81%** | 92.59% | 92.10% | **92,96%** |

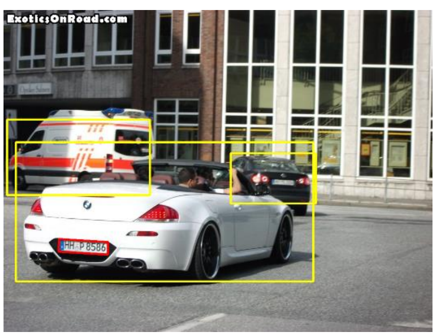
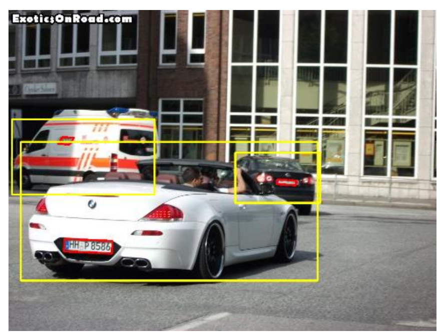
WPOD-net and our improvement model are trained by the same dataset, loss function, optimization algorithm, hyper parameters. The results in Table 3 show that the improved model gives significantly better results in qIoU (from 1 to 1.5%) than the original model when evaluated on our dataset.

The modification also helps improve performance of complete ALPR methods in challenging scenarios (AOPL RP dataset). In controlled scenarios (OpenALPR dataset), the improvement is as effective as WPOD-net.

## 3.4 Qualitative Comparison

To review the quality of our improvement, we compare our results with WPOD-net. The results are shown in Table 4

**Table 4: Compare WPOD-Net and Our improvement model result**

| WPOD-Net | Our |
| --- | --- |
|  |  |
|  |  |
|  |  |


The results show that information of the edge in the original image brings a lot of value, our model recognizes better than the original model in blurred or partially obscured license plate images. In some photos, license plate is really hard to detect but our model can detect it good.

## 3.5 Conclusion & Perspective

In this research, we propose a method to improve the performance of WPOD network by adding edge knowledge to the embedding vector of the network. The results of the thesis show that when applied to the license plate detection problem, the quality of license plate detection is significantly improved. Accurate object detection, especially in cases where the subject image is tilted, will help better in the subsequent information extraction. The thesis can be a good reference for many machines learning and data mining problems.

In the future, we would like to propose a robust network by changing the backbone of the network or using another feature extraction methods. We also want to introduce more applications of this model to other practical problems such as menu digitization or document digitization.

# 4 References


1. Zou, Zhengxia, et al.: Object detection in 20 years: A survey. arXiv:1905.05055 (2019).

2. Haralick, R.M., Linda G.S.: Image segmentation techniques. Computer vision, graphics, and image processing 29.1, 100-132 (1985).

3. Logothetis, N.K., David L.S.: Visual object recognition. In: Annual review of neuroscience 19.1, 577-621 (1996).

4. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: Proceedings of the IEEE international conference on image processing (ICIP). pp. 3464–3468 (2016).

5. Richard, S.: Image alignment and stitching: A tutorial. In: Foundations and Trends® in Computer Graphics and Vision 2.1, 1-104, (2007).

6. Silva, S.M., Jung, C.R.: License plate detection and recognition in unconstrained scenarios. In: Proceedings of the European conference on computer vision (ECCV), vol. 11216, pp. 580-596 (2018).

7. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 779-788 (2016).

8. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. arXiv:1512.02325 (2016).

9. Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems 28 (2015).

10. Silva, S.M., Jung, C.R.: A Flexible Approach for Automatic License Plate Recognition in Unconstrained Scenarios. In: IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 6, pp. 5693-5703 (2022).

11. Torre, V., Poggio, T.A.: On Edge Detection. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 2, pp. 147-163 (1986).

12. L. Ding and A. Goshtasby, "On the Canny edge detector," *Pattern Recognit.*, vol. 34, no. 3, pp. 721–725, doi: 10.1016/S0031-3203(00)00023-6 (2001).

13. Feng-ying Cui, Li-jun Zou and Bei Song, "Edge feature extraction based on digital image processing techniques," 2008 IEEE International Conference on Automation and Logistics, pp. 2320-2324, doi: 10.1109/ICAL.2008.4636554 (2008).

14. S. Yu, "A novel license plate location method based on wavelet transform and EMD analysis," *Pattern Recognition*, vol. 48, no. 1, pp. 114–125 (2015).

15. A. H. Ashtari, "An Iranian license plate recognition system based on color features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1690–1705 (2014).

16. M. Rasooli, 'Farsi License Plate Detection based on Element Analysis and Characters Recognition' (2011).

17. J. Huang, M. Ruzhanshy, H. Feng, L. Zheng, X. Huang, and H. Wang, "Feature extraction for license plate location based on $L_0$ -norm smoothing," *Open Comput. Sci.*, vol. 9, no. 1, pp. 128–135, doi: 10.1515/comp-2019-0007 (2019).

18. M. H Dashtban, Z. Dashtban, and H. Bevrani, "A novel approach for vehicle license plate localization and recognition," *International Journal of Computer Applications*, vol. 26, no. 11, pp. 22–30 (2011).

19. Y. Cheng, J. Lu and T. Yahagi, "Car license plate recognition based on the combination of principal components analysis and radial basis function networks," Proceedings 7th International Conference on Signal Processing. Proceedings. ICSP '04. 2004., 2004, pp. 1455-1458 vol.2, doi: 10.1109/ICOSP.2004.1441601 (2004).

20. M.-S. Pan, Q. Xiong, and J.-B. Yan, “A new method for correcting vehicle license plate tilt,” *International Journal of Automation and Computing*, vol. 6, no. 2, pp. 210–216 (2009).

21. K. Deb, A. Vavilin, J.-W. Kim, and K.-H. Jo, “Vehicle license plate tilt correction based on the straight-line fitting method and minimizing variance of coordinates of projection points,” *International Journal of Control, Automation and Systems*, vol. 8, no. 5, pp. 975–984 (2010).

22. M. Abadi *et al.*, “TensorFlow: A system for large-scale machine learning.” [Online]. Available: http://arxiv.org/abs/1605.08695 (2022).

23. G. -S. Hsu, J. -C. Chen and Y. -Z. Chung, "Application-Oriented License Plate Recognition," in IEEE Transactions on Vehicular Technology, vol. 62, no. 2, pp. 552-561, doi: 10.1109/TVT.2012.2226218 (2013).