Improving the performance of license plate detection using deep neural networks on diverse datasets

Phong Hai Bui[1], Vu Tran Anh Khoi[2], Pham Hoang Lam[2], Nguyen Thi Dieu Ha[2], Nguyen Tra My[2] and Tran Hai Linh[2]

[1] *Faculty of Information Technology, Hanoi Architectural University, Hanoi, Vietnam*
[2] *Vitech Institute, Hoang Mai, Hanoi, Vietnam*

---***---

Abstract – *The detection of license plates (LPs) is a crucial step to develop the intelligent traffic management systems. Several challenges exist for the detection of LPs such as the high variation of the geometry of LPs or the frequent variation in the conditions of LP image acquisition. The paper proposed improvements for the detection of LPs. Firstly, advanced deep neural networks are employed to detect LPs accurately. For training the deep neural networks efficiently, the data augmentation techniques are applied. Then, the strategies of the deployment and testing of the deep neural networks on various hardware plat- forms are proposed to improve the inference time of LP detection. We have performed the evaluation on two public datasets (Vietnamese license plate detection and Kaggle datasets). The performance comparison (the detection accuracy and execution time) with existed methods on various hardware platforms shows the effectiveness of the proposed method.*

Key Words: *Deep neural network, hardware platform, data augmentation, license plate detection.*

1.INTRODUCTION

The license plate detection is one of the most important step to develop the intelligent traffic management systems. The LP detection has been well researched in recent years [1]–[3], however, the accuracy of the detection remains still low and needs to be improved. License plate detection aims to obtain position information of LPs in images. Fig- 1shows examples of captured license plate images. As shown in Fig- 1, several challenges of the license plate detection can be described as follows:

- Due to conditions of image acquisition, license plate images may be blurred, curved and skew. The characteristics cause many difficulties in the detection of LPs.
- The high variation of the types and sizes of LPs cause many errors in the detection of LPs.
- The complex backgrounds of captured LPs may cause the ambiguities in the detection of LPs.

In recent years, embedded systems (e.g. Jetson Nano, Rasp- berry Pi3, NPU VIM3 toolkit) have significantly advanced

[4] so that the Deep Neural Network (DNN) models can be implemented on the systems. The implementation and deployment of DNNs on the systems allow to obtain high accuracy with low computational time for object detection task. Taking the advantages of the computational resources of the embedded systems, the paper presents the detection of LPs applying the YOLOv4 [5] on the NPU VIM3 toolkit to achieve the trade-off between the detection accuracy and the computational time. The contributions of the paper are twofold:

(1) Various deep neural networks [5] are employed and optimized to detect LPs accurately. Compared to traditional methods, the LP detection using the neural network model gains higher accuracy. Moreover, to improve the detection accuracy, the augmentation technique is applied in the training of the network.

(2) The trained deep neural networks are deployed on different hardware platforms [6]. The deployment allows to obtain more efficiency in term of inference time and financial cost of the LP detection.

The rest of the paper is organized as follows. Section 2 reviews related studies of the work. In Section 3, the proposed method is described. In Section 4, we describe and analyze experimental results. Finally, Section 5 provides the conclusion of and the future work.

2. RELATED WORKS

This section reviews important related work of the license plate detection. Both traditional and deep learning approaches for the LP detection are analyzed in the section. The overview of traditional and deep learning approaches for the LP detection are shown in Fig- 2.

(a) Examples of square LPs

(b) Examples of rectangle LPs

(c) Examples of captured LP images in Vietnam

(d) Examples of captured image of cars with LPs

Fig-1: Examples of various types and sizes of LPs in images.

2.1 Traditional methods for LP detection

For decades, several methods have been proposed to detect LPs. Early researches of the field applied the traditional image processing and handcrafted feature extraction techniques to detect LPs. The method in [7] applied the traditional image processing to detect LPs. Firstly, the wavelet transform is applied for LP images. Then, the LP position information is roughly estimated using the horizontal projection of vertical edge in the obtained images after the wavelet transform. Finally, the precise location of LPs is obtained using several threshold values of the column search method. The experiments were performed on small dataset (315 images) and the detection accuracy is 92.4%. The method is not robust for images with complex backgrounds and the determination of thresholds are difficult for large datasets. The work in [1] applied the Histogram of Oriented Gradient (HOG) and a linear Support Vector Machine (SVM) to detect Brazilian LPs. The detection accuracy of the method is low (78%) on a small dataset (377 images). The work in [2] proposed a multi stages method to detect LPs. Firstly, the connected components of LP images were extracted. Then the Haar-like feature and the Adaboost algorithm are applied to detect LPs. The method was evaluated on the Korean LP datasets containing 1800 images. The obtained detection accuracy of the method is 98.4%. The work in [8] proposed an approach based on the linear SVM and color saliency features to detect LPs. The approaches can detect LPs with high accuracy in some circumstances, however, these methods consist of muti-steps that are complicated for real time application.

2.2 LP detection using Deep Neural Networks (DNNs)

In recent years, the DNNs [3], [9] have shown the high performance in the LP detection and recognition tasks. The work [10] proposed a method to solve the detection and recognition of LPs in natural scene images. In the work, a unified deep neural network model is proposed to detect and recognize LPs simultaneously. A large datasets containing 46000 Chinese car LPs are collected to train and test the network. The work in [3] applied the lightweight convolutional neural networks (CNNs) to detect and recognize LPs. The work in [4] implemented the Tiny YOLOv3 [11] architecture to detect LPs. Then the detected LPs were recognized applying a small CNN network. The detection and recognition are performed on the Raspberry Pi3 with the PiNoIRV2 camera. The system was evaluated on small datasets containing about 610 PL images. The accuracies of the detection and recognition of LPs are 99.37% and 97.00%, respectively. The SSD network [12] was applied to detect LPs in pictures and videos on Jetson Nano platform [13] with the 39 Frame per second (FPS). As analyzed above, there are relatively few methods have attempted to detect LPs using DNNs deployed on various hardware platforms NPU hardware platforms (e.g., Jetson Nano, NPU VIM3 and server with Graphic Processing Unit (GPU)).

3. PROPOSED SYSTEM

The principle steps of our proposed system are described in Fig- 3. The framework consists of the following steps:

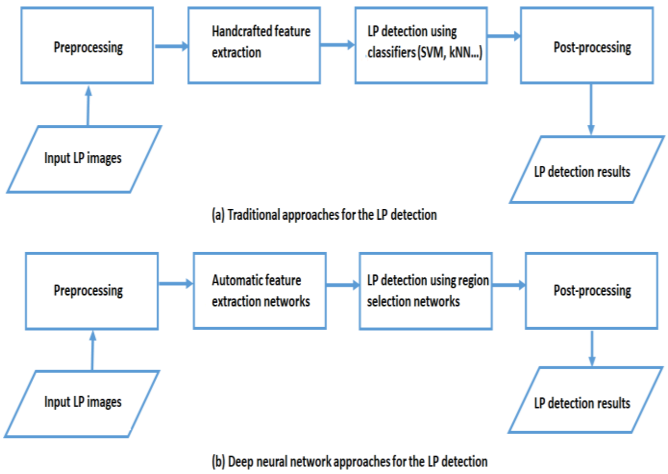


Fig- 2: Traditional and deep learning approaches for the LP detection.

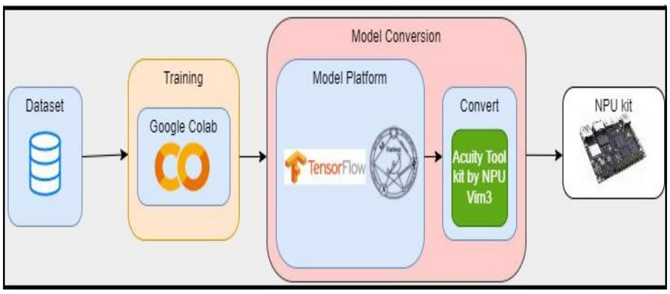


Fig- 3: Flowchart of the proposed system.

(1) **Step 1:** Input LP images are normalized to the size of 416x416 for the training and testing DNNs (YOLOv4 network).

(2) **Step 2:** The YOLOv4 [5] is trained using training datasets of images containing LPs. The network consists of the following sub networks:

- Cross Stage Partial Network (CSPDarknet-53) network

[14] plays an important role for extracting rich features of input images.

- Spatial Pyramid Pooling networks (SPPnet) consist of different max-pooling layers to increase the receptive field of the network.

- Path Aggregation Network (PANet) consists of Down- Sampling and UpSampling layers to extract features repeatedly.

- Three YOLO heads are employed to detect LPs of different sizes.

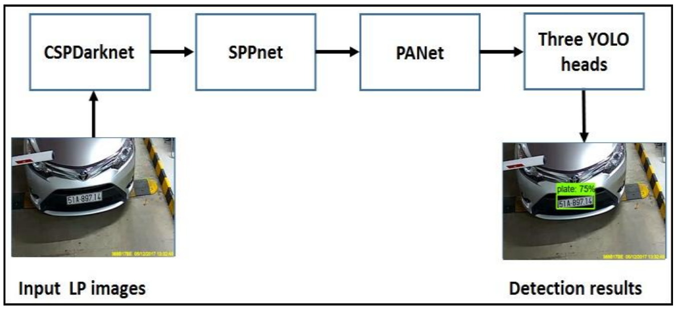


Fig- 4: YOLOv4 architecture for the detection of LPs using YOLOv4.

Fig- 4 illustrates the YOLOv4 architecture for the detection of LPs. Fig- 5 describes detail structure of Darknet-53

	Type	Filters	Size	Output
	Convolutional	32	3 x 3	256 x 256
	Convolutional	64	3 x 3/2	128 x 128
1 x	Convolutional	32	1 x 1	
	Convolutional	64	3 x 3	
	Residual			128 x 128
	Convolutional	128	3 x 3 / 2	64 x 64
2x	Convolutional	64	1 x 1	
	Convolutional	128	3 x 3	
	Residual			64 x 64
	Convolutional	256	3 x 3 / 2	32 x 32
8x	Convolutional	128	1 x 1	
	Convolutional	256	3 x 3	
	Residual			32x32
	Convolutional	512	3 x 3 / 2	16 x 16
8x	Convolutional	256	1 x 1	
	Convolutional	512	3 x 3	
	Residual			16x16
	Convolutional	1024	3 x 3 / 2	8 x 8
4x	Convolutional	512	1 x 1	
	Convolutional	1024	3 x 3	
	Residual			8 x 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

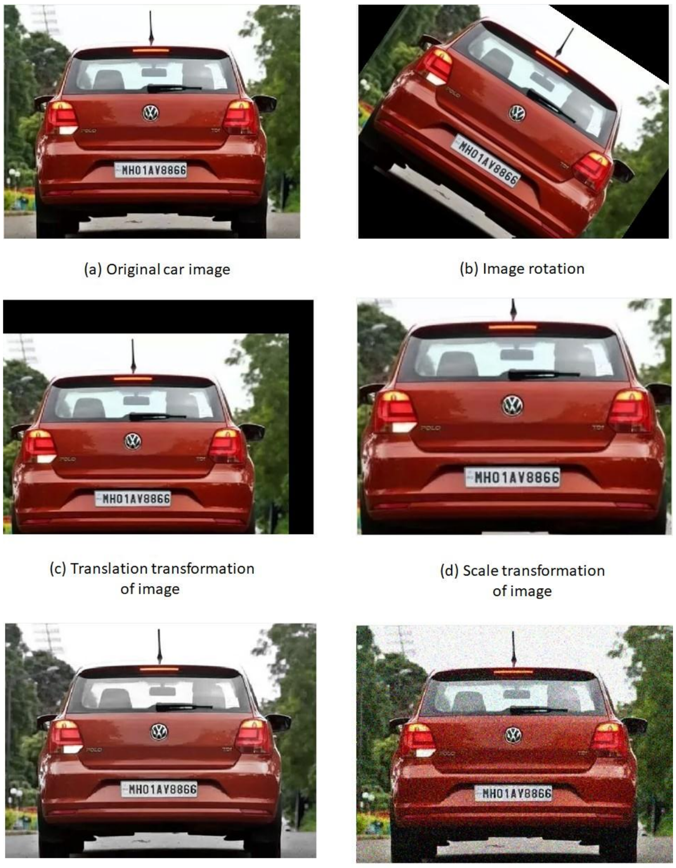
Fig- 5: Detail structure of the Darknet-53 network.

Network that is used as the backbone of the YOLOv4. The network is trained on the server equipped with Nvidia Tesla V100 (16GB VRAM), CPU 4 cores, and 8GB RAM memory. The initial learning rate is set as 0.001. The Adam solve algorithm [15] was utilized for training the network with 60 epochs, in which the momentum is set at 0.93.

To improve the detection accuracy, the data augmentation technique is applied. Data augmentation encompasses a suite of techniques that enhance the size and quality of training datasets such that better DNN models can be built using them [16]. Then, the augmented datasets are used to train DNNs. The following techniques are applied to augment the original image datasets:

- Create a translation transformation of images.
- Create a randomized rotation transformation of images.
- Random scaling images by 10 percent.
- Create Hue Jitter of images.
- Add synthetic noises to images.

Fig- 6 illustrates the data augmentation using image processing techniques.



(a) Original car image (b) Image rotation (c) Translation transformation of image (d) Scale transformation of image (e) Hue Jitter of image (f) Gaussian noise of image

Fig-6: Illustration of the data augmentation using image processing.

Training datasets consist of LP images and bounding boxes of the LPs. The loss function of the training the network is defined as follows:

$$\text{Loss} = L_{confidence} + L_{classification} + L_{CiOU} \quad (1)$$

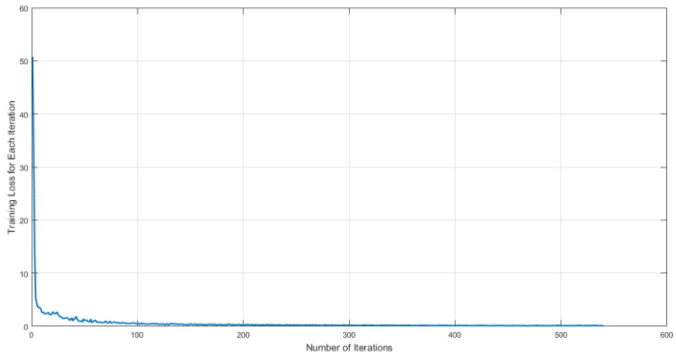


Fig- 7: Value of the loss function of the YOLOv4 during the training process.

The loss function of the YOLOv4 [17] network is the sum of the localization, confidence and the Complete Intersection over Union (CIoU) loss functions. The localization loss function measures the differences between the predicted and the ground-truth bounding boxes. The confidence loss function is calculated based on the classification results of the network. CIOU loss function is composed of the overlap area, center point distance, and aspect ratio, and is used to fit positioning information. Value of the loss function during the training the YOLOv4 network is illustrated in Fig- 7. The loss function value sharply decreases from 0 to 200 iterations and the value almost converges from 500 to 600 iterations.

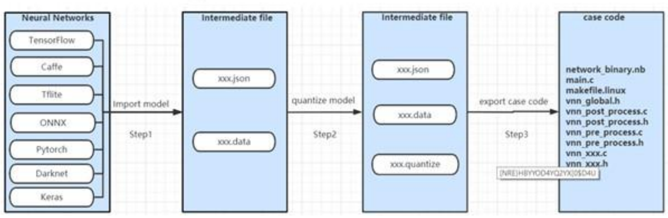


Fig-8: Flowchart of the quantization process in NPU VIM3 hardware platform.

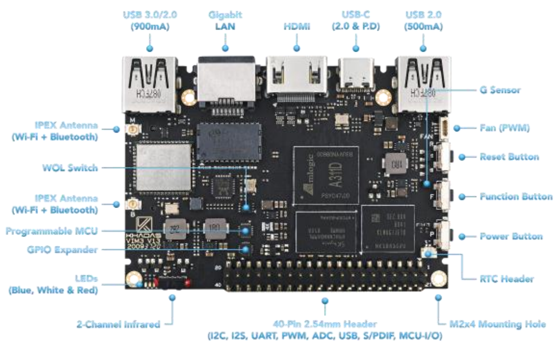


Fig-9: Hardware information of NPU VIM3 toolkit [6].

(3) **Step 3:** To deploy the trained DNNs on low computational hardware platforms, the quantization process is applied. The quantization process aims to convert the trained DNN models (described in step 2) to intermediate files and export the files to case codes so that

the models can be executed on the NPU tool kit. The process is illustrated in Fig- 8. Fig- 9 illustrates architecture of the NPU VIM3 tool kit. Fig- 10 demonstrates an example of the quantization from *floating point* to *integer* numbers.

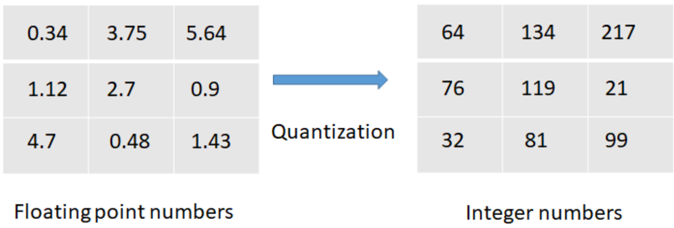


Fig-10: Illustration of the quantization process in NPU VIM3 hardware platform.

(4) **Step 4:** The LP detection is performed on the various platforms: server with graphical processing unit (GPU), Jetson Nano and NPU VIM3 tool kit. In the phase, the trained YOLOv4 is employed to detect LP images that are normalized to the size of 416x416. Detail information of the platforms are described as follows:

The NPU VIM3 tool kit [6] consists of the following components:

- Processors of Cortex A73 performance-cores (2.2GHz), and x2 Cortex A53 efficiency-cores (1.8GHz).
- Memory of 2GB Ram 16GB eMMC.
- The tool kit supports an onboard 5.0 Trillion Operations per Second (TOPS) computational resource [6] for testing the detection of LPs using trained DNNs.
- The operating system of tool kit is Ubuntu 20.04.

The server used for training DNNs consists of the following information:

- GPUs Geforce GTX 1050.
- 8 GB of RAM.
- Core i5-9300H processor.
- Windows 10 operating system.

Hardware information of the Nvidia Jetson Nano is described as follows:

- CPU Quad-core ARM Cortex-A57 1.43 GHz.
- GPU 128-core NVIDIA Maxwell.
- 2 GB Ram 64-bit LPDDR4.

4. EXPERIMENTAL RESULTS

4.1 Datasets and evaluation metric

4.1.1 Datasets

We performed the framework on two public datasets: the Vietnamese LP [18] and Kaggle car LP [19] datasets. The Kaggle car LP dataset contains 400 images with bounding box annotations of the car license plates within the image. Each image consists of minimum 1 and maximum of 4 LPs. Annotations are provided in the PASCAL VOC format. The Vietnam LP dataset consists of 8000 Vietnamese images (e.g. car, motor plates). LPs in each image are annotated with bounding boxes information and classes of object. The training and testing datasets are used for training and test DNNs. To improve the training accuracy, we applied the augmentation techniques for training dataset. For fair testing the DNNs, the testing dataset is not augmented. Statistic information of LP datasets before and after using the augmentation techniques is shown in Table-1.

4.1.1 Evaluation metric

The Intersection over Union (IoU) metric [20] was applied in the work for the performance evaluation of LP detection. The metric is defined as follows:

$$IoU = \frac{A \cap B}{A \cup B} \quad (2)$$

where $A \cap B$ is the intersection of the predicted and ground- truth bounding boxes of LPs. $A \cup B$ denotes the union of the predicted and ground- truth bounding boxes of LPs. A detection is correct if $IoU \geq 0.5$. To obtain the trust performance evaluation, the mean average precision (mAP) is also applied in the work. The LP detection is performed and evaluated ten times and the *mAP* value of the detection is obtained.

Table -1: Statistic information of LP datasets before and after applying the augmentation technique

Datasets	Training	Testing
Number of LP images (Vietnamese dataset) before applying augmentation	7000	1000
Number of LP images (Vietnamese dataset) after applying augmentation	35000	1000
Number of LP images (Kaggle car dataset) before applying augmentation	200	200
Number of LP images (Kaggle car dataset) after applying augmentation	1000	200

4.2 Performance evaluation

To evaluate the performance of the proposed method, we compared the detection accuracy with other DNNs (SSD, YOLOv3 networks) on various hardware platforms (Jetson Nano and GPUs). The comparison of the LP detection accuracy is shown in Table-2 , 3, 4 and 5. As shown in the tables, the YOLOv4 achieves the highest detection accuracy compared to the SSD and YOLOv3 networks. Table -6 indicates that the detection accuracy of LPs on the NPU VIM3 tool kit is competitive with that on the Jetson Nano. The detection of LPs on the GPUs allows to obtain the highest score. The results come from the fact that the hardware resources of the GPUs are better than those of NPU VIM3 tool kit. The Jetson Nano systems support the 16 floating point format precision, whereas the NPU VIM3 tool kit supports the 8 integer one. The differences of the hardware resources and the number precision affect the detection accuracy on GPUs, Jetson Nano and NPU VIM3 tool kit.

Table -2: LP detection results using the YOLOv4 with and without using the data augmentation techniques on Vietnamese LP datasets ($IoU \geq 0.5$).

Methods	**mAP**
YOLOv4 without applying the data augmentation technique	93.75%
YOLOv4 with applying data augmentation technique	**94.80%**

Table -3: LP detection results using the YOLOv4 with and without using the data augmentation techniques on Kaggle LP datasets ($IoU \geq 0.5$).

Method	**mAP**
YOLOv4 without applying the data augmentation technique	90%
YOLOv4 with applying data augmentation technique	**92%**

Table -4: LP detection comparison using various methods with Vietnamese LP datasets on server with GPU

Method	**mAP**
SSD MobilenetV2 [12]	89.30%
YOLOv3	92.50%
Proposed method based on YOLOV4	**94.80%**

Table -5: LP detection comparison using various methods with Vietnamese LP datasets on NPU VIM 3 tool kit

Method	**mAP**
SSD MobilenetV2 [12]	89.30%
YOLOv3	92.50%
Proposed method based on YOLOV4	**94.80%**

Table -6: LP detection comparison using various methods with Vietnamese LP datasets on various hardware platforms

Platform	**mAP**
GPU Geforce GTX 1050	96.11%
Jetson Nano [18]	94.39%
NPU Khadas VIM3	**94.80%**

Examples of the LP detection using the YOLOV4 are shown in Fig- 11 and 12. The results illustrate that the proposed system can detect LPs accurately in different cases (i.e., skew and tiny LPs). An example of the detection using the SSD network is shown in Fig-13. The detection using the SSD network is faster than that of the YOLOv4 network. However, the detection accuracy of the SSD is lower than that of the YOLOv4 network.

To evaluate the execution time of the proposed system and other hardware platforms, we compared the average inference time (in millisecond) of the YOLOv4 on various hardware platforms. Fig- 14 shows the inference time of the LP detection on using the YOLOv4 testing datasets on

Fig-11: Examples of LP detection using YOLOv4

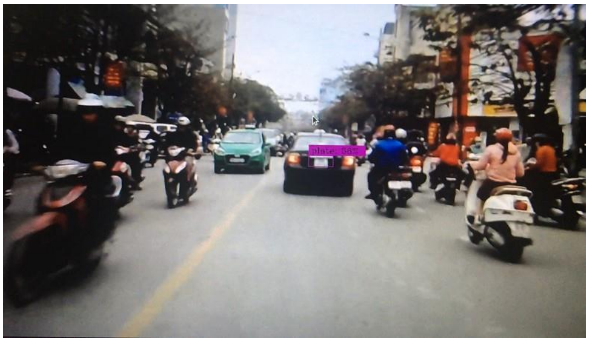
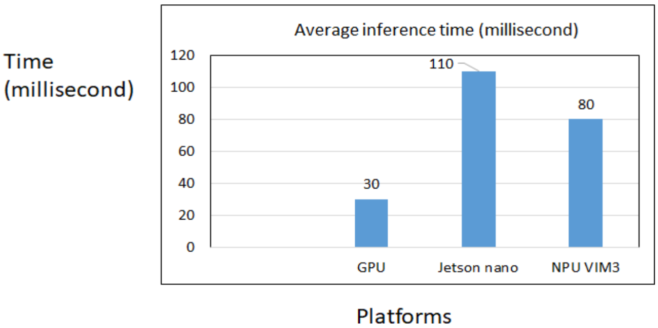


Fig-12: Examples of the detection of tiny and blur LPs using YOLOv4

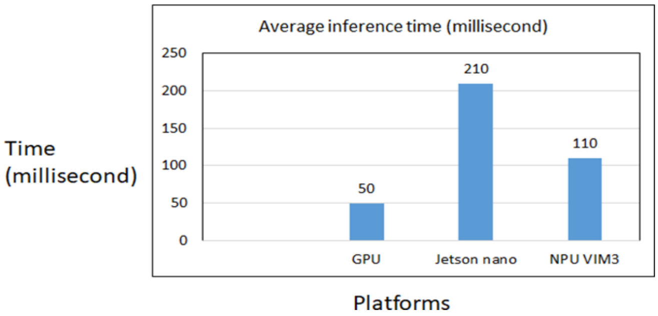
Fig-13: Examples of the detection of tiny and skew LPs using the SSD network

various hardware platforms. The Figure shows that the inference time on the NPU VIM3 tool kit is significantly improved in comparison with that on the Jetson Nano. The better results come from the deployment process of DNNs and the support of the 8 integer point format precision of the NPU VIM3 tool kit. Compared to the YOLOv4, the inference time of the SSD Mobilenet is much reduced because the architecture of the SSD Mobilenet is simpler.

4.3 Discussion and error analysis



(a) Execution time of the LP detection using the SSD Mobilenet



(b) Execution time of the LP detection using the YOLOv4

Fig-15: Comparison of average execution time (in millisecond) of the LP detection on the Vietnamese testing datasets on various hardware platforms.

As our results, the detection of LPs using the YOLOv4 on GPUs achieves the highest accuracy thanks to the powerful hardware resources. The detection accuracy on the Vietnamese dataset is higher than that on the Kaggle dataset because Vietnamese dataset is larger than the Kaggle. The data aug mentation allows to improve the detection accuracy by 1.05% and 2% on Vietnamese and Kaggle datasets, respectively. The detection accuracy of LPs on the NPU VIM3 achieves competitive results compared to that on the Jetson Nano. However, the execution time of the detection on the NPU VIM3 tool kit is much reduced than that on the Jetson Nano.

Some remained detection errors can be caused by the following reasons:

1. The quality of some LP images is too low. Some captured images can be blurred. Some images contain many noises and shadow.

2. Several objects that are similar to LPs in the images with complex backgrounds caused the ambiguities of the detection.

5. CONCLUSIONS AND FUTURE WORKS

The paper has presented the improvement of LP detection using deep neural networks trained on diverse datasets. The employment of the advanced neural network allows to obtain high accuracy of the LP detection. The data augmentation techniques in the training of deep neural networks show the improvements of the detection accuracy. Moreover, the strategy of deployment of the trained network on various hardware platforms achieves the significant improvement of the execution time. The performance comparison with conventional methods on the two public datasets (Vietnamese and Kaggle LP detection datasets) and on various hardware platforms illustrates the effectiveness of the proposed method. In the future, the detection results can be integrated with the recognition of LPs in several real applications such as intelligent transportation systems. Moreover, other advanced CNNs can be combined to improve the detection accuracy of the LP detection.

REFERENCES

[1] R. Prates et al., “Brazilian license plate detection using histogram of oriented gradients and sliding windows,” in International Journal of Computer Science Information Technology (IJCSIT), vol. 5, no. 6, 2013.

[2] M.Sarker et al., “A fast and robust license plate detection algorithm based on two-stage cascade adaboost,” in KSII Transactions on Internet

and Information Systems, vol. 8, no. 10, 2014.

[3] P. Anh, “Effective deep neural networks for license plate detection and recognition,” in Vis Comput. https://doi.org/10.1007/s00371-021-02375- 0, 2022.

[4] D. M. F. Izidio et al., “An embedded automatic license plate recognition system using deep learning,” in 2018 VIII Brazil- ian Symposium on Computing Systems Engineering (SBESC), DOI: 10.1109/SBESC.2018.00015, 2018.

[5] A. Bochkovskiy et al., “YOLOv4: Optimal speed and accuracy of object detection,” in arXiv:2004.10934, 2020.

[6] “Npu kit vim3,” in https://docs.khadas.com/linux/vim3/Hardware.html, 2022.

[7] C.-T. Hsieh et al., “Multiple license plate detection for complex back- ground,” in 19th International Conference on Advanced Information Networking and Applications, 2005.

[8] Y. Yuan et al., “A robust and efficient approach to license plate detection,” in IEEE Transactions on Image Processing, vol. 26, no. 3, 2017.

[9] S. Z. Masood et al., “License plate detection and recogni- tion using deeply learned convolutional neural networks,” in arxiv.org/abs/1703.07330, 2017.

[10] H. Li et al., “Toward end-to-end car license plate detection and recog- nition with deep neural networks,” in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 3, 2019.

[11] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” in arXiv:1804.02767, 2018.

[12] W. Liu et al., “Ssd: Single shot multibox detector,” in arXiv:1512.02325, 2016.

[13] “Jetson nano,” in https://github.com/dusty-nv/jetson-inference, 2022.

[14] C.-Y. Wang et al., “Cspnet: A new backbone that can enhance learning capability of cnn,” in https://arxiv.org/abs/1911.11929, 2019.

[15] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in arXiv preprint arXiv:1412.6980, 2014.

[16] C. Shorten and T. Khoshgoftaar, “A survey on image data augmentation for deep learning,” in Journal of Big Data 6(1), 2019.

[17] C. Guo et al., “Improved YOLOv4-tiny network for real-time electronic component detection, doi: 10.1038/s41598-021-02225-y,” in Scientific Reports, 2021.

[18] “License plate dataset,” in https://github.com/winter2897/Real-time- Auto-License-Plate-Recognition-with-Jetson-Nano, 2022.

[19] “Car license plate dataset,” in https://www.kaggle.com/datasets/andrewmvd/car- plate-detection, 2022.

[20] M. Everingham et al., “The pascal visual object classes (voc) challenge,” in Int J Comput Vis, vol. 88, 2010.