Real-Time Detection and Recognition of License Plates for Traffic Monitoring

Nam Van Nguyen[1,2(✉)] and Quan Minh Vu[3]

[1] Data Governance Department, Viettel Group, Alley 7, TonThatThuyet Street, CauGiay District, Hanoi, Vietnam
namnv78@viettel.com.vn, nvnam@tlu.edu.vn
[2] Thuyloi University, 175 Tayson, Dongda, Hanoi, Vietnam
[3] Viettel CyberSpace Center, Viettel Group, 7 TonThatThuyet Street, CauGiay District, Hanoi, Vietnam
quanvm4@viettel.com.vn

Abstract. We address the task of real-time detection and recognition for heterogeneous license plate images of diverse vehicles with characters arranged in multiple lines and captured in all day and night conditions. This paper presents MixLPR (Mixed License Plate Recognition), a framework to develop a real-time system deployable in dense urban traffic to fill that gap. MixLPR consists of two components, a license plate detector, and an OCR. The plate detector includes new Mish-enhanced residual nets equipped with geometric transformations to deal with view-induced distortions. The OCR component is a new segmentation-free method based on the transformers, which work directly on the 2D character block. We trained and validated MixLPR on two large public and private datasets. Our results exhibit both improvements in accuracy and inference speed compared to state-of-the-art approaches.

Keywords: License Plate Detection and Recognition · Convolutional neural networks · Attention mechanism and transformer

1 Introduction

Real-time, scalable Automatic License Plate Recognition (ALPR) is essential for traffic monitoring in modern smart cities. A typical ALPR system first detects the license plates which occupy only a very small proportion of the wide camera view. Then each plates are run through an OCR subsystem which recognizes the characters. The problems are highly challenging in real traffic monitoring practice due to multiple vehicle types and variations in license plate formats of moving vehicles captured by fixed monitoring cameras. The challenges are also amplified by the low quality of images under different lighting conditions, blurring due to fast motion, distortion due to camera angle, and occlusion in typical dense urban traffic (Fig. 1).

S.-H. Wang and Y.-D. Zhang (Eds.): ICMTEL 2022, LNICST 446, pp. 420–434, 2022.
https://doi.org/10.1007/978-3-031-18123-8_32

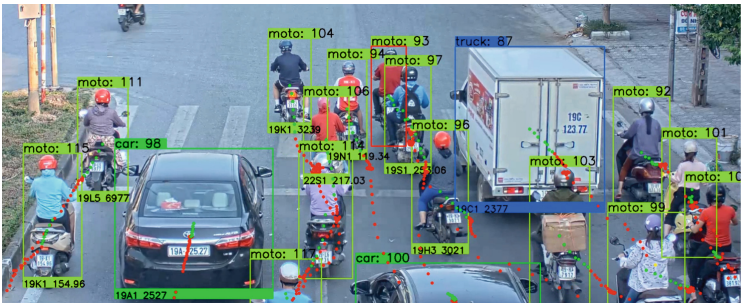

Fig. 1. Recognition of heterogeneous license plates in real-time under various lighting conditions. Vehicle bounding boxes are computed by a module not reported in this paper.

Many methods have proposed for recognising characters in single-line license plates, using numerous segmentation-based [2,3,6,7] and segmentation-free with CTC and attention mechanism [13,17,18,20]. However, there is no work on the recognition of license plate characters in multiple lines for small vehicles like motorbikes from live camera streams in crowded roads. Our main contributions are two-fold. First, we propose a robust neural network using our Mish-based residual blocks for accurate detection and rectification of license plates of cars, trucks, buses and bikes in both daytime and nighttime. Second, we propose an effective solution for the task of optical character recognition on multi-line license plates using Transformers.

We trained and validated MixLPR on two public datasets (CCPD-2020 and the Stanford-Cars) for cars only, and three large private datasets consisting of more than 200,000 images for a mix of bikes, cars, buses and trucks in total. Our results exhibit both accuracy and speed improvements over state-of-the-art approaches. MixLPR serves as the basis for a real-time traffic monitoring system for urban traffic in Vietnam.

The rest of paper is organised as follows. Section 2 reviews related works. Section 3 presents our main contributions – the MixLPR framework that can effectively work on mixed-type license plates. The effectiveness and efficiency are then evaluated in Sect. 4 on public and private datasets. Finally, Sect. 5 concludes the paper.

2 Related Work

Our work draws on recent works in object detection and optical character recognition (OCR) especially for license plate of vehicles.

2.1 License Plate Detection

Many ALPR are based on the state of the art object detection models [8,12,15,21] that output rectangular bounding boxes for license plates. Our detection

approach, on the other hand, works on skewed non-rectangular shapes. We are inspired from recent works [5,13] which first detect four vertices of the quadrilateral bounding boxes of license plates and then rectify them to the original rectangular ones. These CNNs are also more light-weight than the state-of-the-art object detectors in license plate detection. However, these still produce high ratio of false positive predictions especially at night where car lights are detected as licences plates due to insufficient feature extraction of small objects. We improve from these baselines by introducing deep residual blocks [4] and Mish activation function [19].

2.2 Optical Character Recognition (OCR)

OCR has a long history. Early methods required character segmentation followed by single character classification [2,3,6,7,10]. The recent combination of CTC-based RNNs and CNNs have resulted in many efficient methods [21] for recognizing characters from the entire single-line license plate images. Our approach, on the other hand, tackles the problem of recognition of multi-line license plates. We combined a CNN [14] with a transformer as this model uses the multi-head attention mechanism over an unordered set, and thus in theory can handle arbitrary spatial character arrangement.

3 Methods

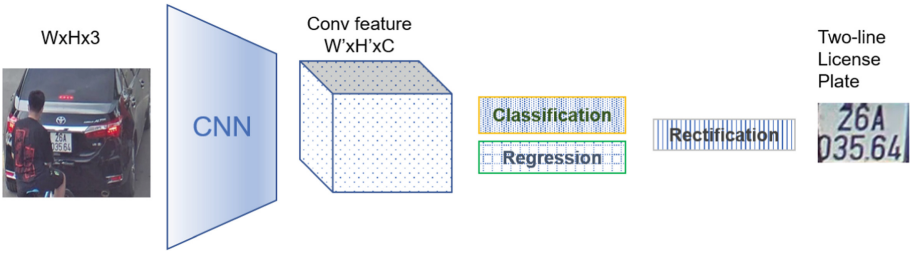

Fig. 2. Detection of license plate.

Our goal is to detect and rectify heterogeneous license plates of vehicles and then recognize their characters in urban traffic using fixed monitoring cameras. We assume that appropriate image pre-processing has been done to detect vehicles. Inspired by Vietnamese license plates, we develop generic methods for a mixture of single and multiple-line plate formats. Here the physical plates are rectangular metal panel containing a one-line or two-line character string. However, under the wide view angle from the cameras, plates appear non-rectangular. Those plates of moto-bikes are two-line attached at the back in Vietnam[1]. Cars,

[1] Plates can also be attached at the front of moto-bike in other countries.

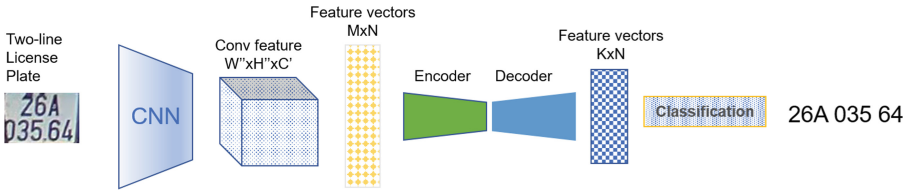

Fig. 3. Recognition of multi-line license plates.

buses and trucks carry both kinds of plates at both the front and back of the vehicles. The plates have between 7 to 11 characters.

Our solution to the problem is a framework dubbed MixLPR (which stands for Mixed License Plate Recognition), which is graphically illustrated in Figs. 2 and 3. MixLPR consists of two connected modules: License plate detection, and multi-line character recognition.

The plate detection takes as input the vehicle bounding box and produces the rectified licence plate image as output. The modules processes the input image using a CNN to produce a feature map. The map then goes through a regression network and a classification network to predict the bounding box of the license plate and its geometrical transformation parameters. The parameters are then used to rectify the license plate.

This is then followed by the recognition module which generates the string of characters in the plate. A CNN is first applied to produce a feature map, which is sequenced into a set of position-encoded feature vectors. This serves as input for a transformer-based encoder-decoder architecture to generate the characters.

In what follow we present the two modules in more detail.

3.1 License Plate Detection

License plate detection can be considered as an object detection task but with unique characteristics that demand special treatments. This is because most of the license plate captured from camera at different angles are polygonal planar which can not be precisely bounded by rectangular boxes used in current object recognition models. Inspired from WPOD-NET [13], the quadrilateral shape of captured license plates can be seen as affine transformation of their original rectangular form. This is because the affine transformation can be built from the basic transformations such as translation, scaling, rotation and shearing.

Affine Transformation of License Plates and Its Inverse. We consider a point-to-point transformation resulting from a rotation and a translation. Given a two-dimension point $p(x_p, y_p)$, its affine transformed $q(x_q, y_q)$ is as follows:

$$q = Rp + T \tag{1}$$

where $T(x_T, y_T)$ is the translation vector and $R = \begin{pmatrix} r_1 & r_2 \\ r_3 & r_4 \end{pmatrix}$ is the rotation matrix. The reverse affine transformation from q to p is as follows:

$$p = R^{-1}q - (R^{-1}T) \tag{2}$$

where $R^{-1} = \frac{1}{r_1 r_4 - r_2 r_3} \begin{pmatrix} r_4 & -r_2 \\ -r_3 & r_1 \end{pmatrix}$ is the inverse of R.

In our models for license plate detection, the regression sub-network outputs six parameters $(r_1, r_2, r_3, r_4, x_T, y_T)$. The affine transformation is used for training and the inverse is deployed for inference. In the training phase, an affine transformation with the six parameters is applied on every rectangular cell of the input image to propose quadrilateral license plates. The six affine parameters are estimated by minimizing a square loss induced by the gap between the proposed license plates and the ground truths. By contrast, in the inference phase, an inverse affine transformation with the six output parameters is applied on the predicted quadrilateral license plate to produce the rectangular one. Thus, using this regression sub-network, we can both detect and align the distorted license plates.

Mish-Based Residual Blocks. The backbone neural network plays the critical role in license plate detection. A key parameter of the network is its depth – the number of feature transformation layers, usually the more layers the more powerful feature extraction. However, more layers can cause gradient vanishing, and thus lower layers are not updated during training, and thus reducing the performance of the networks. An effective solution is through skip-connections, which have been found to work well in recent residual neural networks (ResNets) [4]. The network is built from multiple residual blocks using a connection that skips several layers.

However, we found the rectified linear unit (ReLU) used in the original ResNets to be less effective in noisy images often seen in traffic cameras, causing overfitting. We propose to use a recent alternative activation function known as Mish [9] to combat the noise. Mish is a non-linear monotonic function which transforms the input as follows:

$$f(x) = x.\tanh\left(\ln\left(1 + e^x\right)\right) \tag{3}$$

We design our Mish-enabled residual backbone network as shown in Fig. 4. The first three initial, intermediate and alignment blocks include 2D convolutional filters, batch normalizations and Mish activation functions to extract the features of the input but do not reduce its size. The intermediate block is formed by adding a Mish function at the beginning. A convolutional filter and a batch normalization layer are also inserted to the intermediate block to create an alignment one. The name of the block indicates where it is integrated to the whole backbone network. Meanwhile, the feature of the input can be extracted and its size is also twice reduced by the down-sampling block thanks to its 2D convolutional filter with stride of 2. The last block also contains a maximum pooling at

the shortcut connection so that it can be aggregated to the main one with the same feature size.

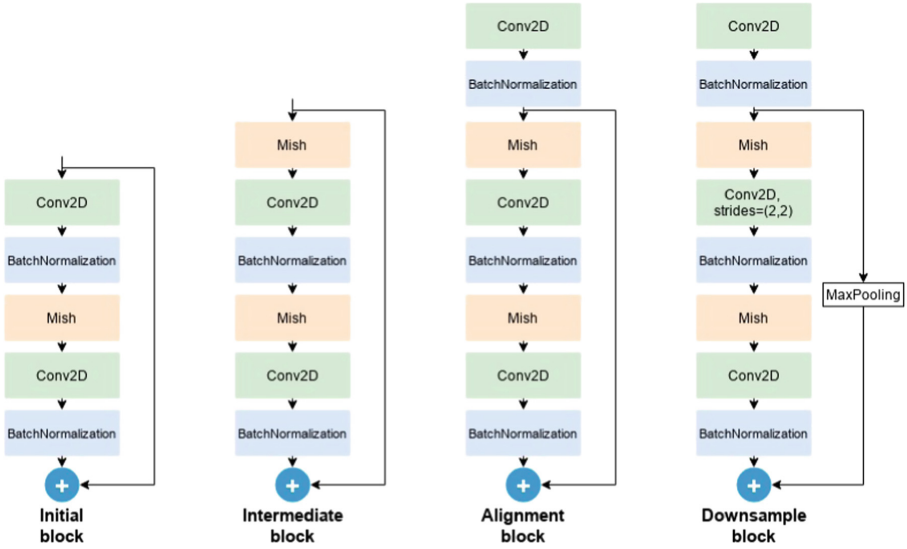

Fig. 4. Mish-based residual blocks.

Simple Network for Heterogeneous License Plate Detection (SHELP). Based on the proposed Mish-based residual blocks, we designed our **S**imple network for **HE**terogeneous **L**icense **P**late detection (called SHELP) as shown in Fig. 5. Our SHELP takes as input an image containing a vehicle with a size of $208 \times 208 \times 3$ for training and of $416 \times 416 \times 3$ for inference. The backbone network contains two down-sampling residual blocks so that the size of input will be four times reduced. It also consists of one initial, one alignment and four intermediate blocks for efficient featuring. With these residual blocks, contextual information can be aggregated via shortcut connections with the local receptive field of license plate making the backbone network of SHELP more robust even with less number of layers. For detection of license plates, SHELP ends with regression and classification sub-networks and a concatenation layer with eight output parameters including six affine transformation parameters and two for {object, non-object} indicators.

Loss Function In the training phase, the input size is $208 \times 208 \times 3$ and the last output shape is $13 \times 13 \times 8$. This means that the input's size is 16 times reduced to form a grid of 13×13 cells. Given four corners $p_k(x_k, y_k), k = \overline{1,4}$ of a ground truth license plate, they then corresponds to $p'_k(\frac{x_k}{16}, \frac{y_k}{16}), k = \overline{1,4}$ in the grid. Suppose that the center of a cell $(i, j), i, j = \overline{1,13}$ can be seen as an origin

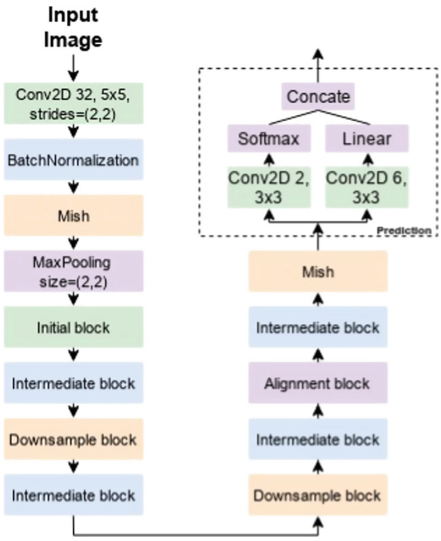

Fig. 5. The architecture of SHELP.

of the coordinate axes, the four corners of the ground truth license plate will be $q_k^{(i,j)}(\frac{x_k}{16} - i, \frac{y_k}{16} - j), k = \overline{1,4}$.

Given four corners of an unit box $u_k, k = \overline{1,4}$ originated at the center a cell $(i, j, i, j = \overline{1,13})$, at each epoch of training, the regression sub-network proposes six affine parameters which transform the unit box to a polygonal licence plate $v_k^{(i,j)} = Ru_k + T, k = \overline{1,4}$.

Inspired from SSD and WPOD-NET, the difference between the two boxes is $l_{loc}^{(i,j)} = \sum_{k=1}^{4} smooth_{L_1}(v_k^{(i,j)} - q_k^{(i,j)})$,where $smooth_{L_1}(x) = \begin{cases} 0.5x^2, \text{if } |x| < 1 \\ |x| - 0.5, \text{otherwise} \end{cases}$.

We denote $x^{(i,j)} = 1$, if the IoU (Intersection over Union) between the cell (i, j) of size 16×16 in the input image and the ground truth license plate is greater than or equal to 0.3. Otherwise, $x^{(i,j)} = 0$. The classification sub-network outputs two parameters (c_1, c_2). Using softmax function, we have (o_1, o_2) where $o_k = \frac{e^{-c_k}}{e^{-c_1}+e^{-c_2}}, k = 1, 2$. In this case, $o_1 + o_2 = 1$. Then, the confidence loss for this sub-network is the binary cross-entropy function: $l_{conf}^{(i,j)} = -x^{(i,j)}.o_1 - (1 - x^{(i,j)}).o_2$

Finally, the loss function between the predicted and the ground truth license plate to be minimized is as follows:

$$L = \sum_{i=1}^{13}\sum_{j=1}^{13}\left(x^{(i,j)}.l_{loc}^{(i,j)} + l_{conf}^{(i,j)}\right) \tag{4}$$

This loss function is used for both SHELP and since they have the same end-block. Both networks output $13 \times 13 \times 8$ parameters – there are 169 cells, each of which has six affine parameters and two probabilities of confidence.

In the inference phase, the size of the input images is $416 \times 416 \times 3$ which is double of the training images. This is because the ground truth license plates in the training images have been zoomed, centered and resized. After predicting six affine parameters and two object/non-object probabilities for every cell, the non-maximal suppression algorithm will be used to select the most accurate predictions. Using the reverse transformation with the predicted affine parameters as in Eq. (2), the detected polygonal bounding boxes of license plate are re-transformed to the rectangular ones and their inside characters are re-aligned which are then fed to the optical character recognition model.

3.2 Recognition of Multi-line License Plates

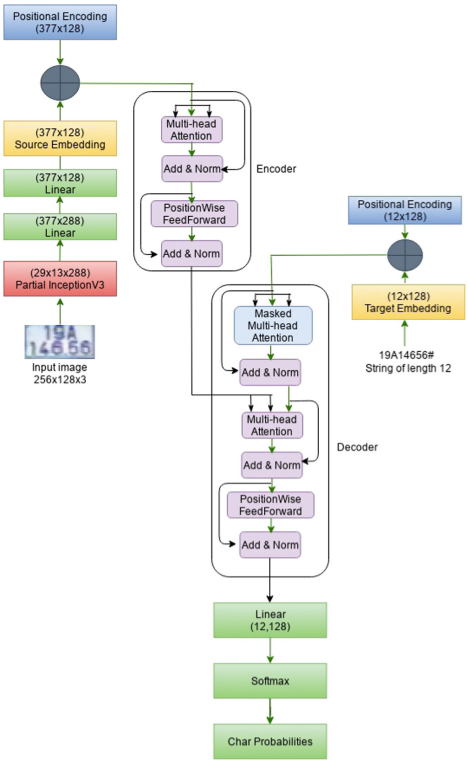

Fig. 6. MCR model architecture.

We present the second module of our MixLPR called MCR (Multi-line License Plate OCR), which translates an image of multi-line license plate into a string of characters. MCR employs a CNN for feature extraction and a transformer-based encoder-decoder for generation of characters.

In what follows we present the components in more detail: The CNN in Sect. 3.2, and the encoder-decoder and the loss function in Sect. 3.2.

CNN-Based Feature Extraction. As described in Fig. 6, all heterogeneous license plate images are resized to $256 \times 128 \times 3$. First, their heights are resized to 64 and 128 for one-line and two-line ones, respectively. Second, their left and right sides are all padded up to a width of 256. Third, the bottom of one-line images is filled up with a box of size $256 \times 64 \times 3$. The license plate images with longer width than 256 are only resized to 256 without padding. In other words, the CNN takes as input image of $256 \times 64 \times 3$ and outputs a multi-channel feature map of $29 \times 13 \times 288$.

For CNNs we use the Inception-v3 [14] thanks to its state-of-the-art performance on the ImageNet classification challenge [1,11]. The mixed-5d layer of the network is chosen for feature extraction due to its fast processing and high accuracy [16].

Encoder-Decoder and Loss Function. The core of our recognition network is the transformer-based encoder-decoder architecture to map the 377 location embedding of the source into characters in the target sequence of length 12. The encoder computes the similarity between source locations and refines the location embedding, making the features ready to be used in the decoder. This similarity is computed through a multi-head self-attention mechanism. The decoder produces one character at a time in the target sequence. At each decoding position $t = \{1, 2, ..., 12\}$, the decoder pools relevant source features and the previous decoded character embedding, also using the multi-head self-attention. The pooled information is then used to compute the probabilities of 37 possible characters (26 alphabet letters, 10 digits and a null):

$$y'_{t,i} = \frac{e^{o_{t,i}}}{\sum_{j=1}^{37} e^{o_{t,j}}} \tag{5}$$

where $o_{t,i}$ is the output of the linear layer corresponding to the ith character in the possible set at position t. This is depicted as the softmax layer in Fig. 6.

For training, the network uses the cross-entropy loss function for classification:

$$L_{OCR} = -\sum_{t=1}^{12}\sum_{i=1}^{37} y_{t,i} \log(y'_{t,i}) \tag{6}$$

where $y_{t,i}$ is the one-hot coding of the groundtruth character at position t.

4 Experiments

In this section we evaluate our proposed MixLPR on both public and private datasets, and compare its performance against strong rivals reported in the literature.

4.1 Experimental Settings

Datasets and Pre-processing. We use two public datasets for license plate detection (CCPD-2020 and Stanford-Cars) and three private datasets for both license plate detection and OCR. **CCPD-2020** (Chinese City Parking Dataset 2020) containing over 10,000 car images, with license plate location annotations. Similarly, **Stanford-Cars** is a dataset of cars with only 3,400 license plate annotations. Since these datasets contain only cars, they do not represent the real traffic in many Asian cities, which has many vehicle types, including bikes, cars, buses and trucks.

For that reason, we collected three private datasets from monitoring cameras in a densely populated Vietnamese city. The first dataset is for license plate detection (**uTVM-LP**), the second is for license plate OCR (**uTVM-OCR**), and the third is for end-to-end testing (**uTVM-N2N**). The **uTVM-LP** dataset has 100,000 images of vehicles (bikes, cars, buses and trucks) annotated with polygons of license plates. Similarly the **uTVM-OCR** dataset contains 100,000 images of license plates annotated with character strings. Lastly the **uTVM-N2N** dataset consists of 6,000 vehicles images which are annotated with their corresponding license plate characters, serving as a testbed for end-to-end system evaluation. Unless otherwise specified, datasets are divided into 70/30% for training and testing, respectively.

Evaluation Metrics. We measured the license plate detection and recognition models using the average precision (AP) and the accuracy of sequence (AOS).

Average Precision is a popular metric for evaluating the object detection models. This is the mean of precision values of the detector over multiple recall rates from 0 to 1. The precision and recall are formulated as follows:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}$$

where TP, FP, FN is the true positive, false positive and false negative value, respectively. These values are determined according to an IoU threshold (intersection over union) between the detected polygonal bounding box and the ground truth license plate.

Accuracy of Sequence (AOS) for license plate recognition is defined as follows:

$$AOS = \frac{n - \#error}{n}$$

where n, $\#error$ are the total number of tested license plates and of unrecognized ones, respectively. If any of the characters in license plate string is not correctly recognized then the license plate recognition is erroneous.

4.2 Results on License Plate Detection

We trained, evaluated and compared SHELP and WPOD-NET on three datasets Stanford-Cars, CCPD and uTVM-LP.

Table 1. Average Precision of license plate detection models.

Dataset	IoU	WPOD-NET	SHELP
Stanford-Cars	0.5	81.2	82.3
Stanford-Cars	0.65	72.6	74.9
uTVM-LP	0.6	96.9	98.2
CCPD-2020	0.6	92.9	93.1

Accuracy of SHELP. The overall results on **all datasets** are presented in Table 1. On all datasets and IoU thresholds, our proposed method (SHELP) are better than WPOD-NET.

The results on **uTVM-LP** dataset partitioned into bikes and big vehicles in daytime and nighttime are shown in Table 2. With IoU of 0.5 all methods perform exceptionally well on bikes, with more than 99.2% AP, where our proposed method SHELP still show better accuracy. The high performance is because the license plates account for a bigger ratio of area in unique bike images than in unique four-wheel ones. Moreover, there are normally lots of noises in four-wheel images such as the poster of advertisements, the reflection in the car mirror or even the license plate images of the neighboring vehicles.

On the **uTVM-LP** dataset, the performance gap for big vehicles is wider among methods. Again, our method is better than WPOD-NET with a margin of 2.29 points in daytime and 6.74 points in nighttime. Interestingly, our methods work better in nighttime than daytime, when SHELP can reach 98.09%. This contrasts with the expected behaviour of WPOD-NET, which drops by 3 points when going from daytime to nighttime.

Table 2. Average Precision of license plate detection models on **uTVM-LP** dataset (IoU = 0.5).

Vehicle	Time	WPOD-NET	SHELP
Big vehicles	Daytime	94.32	94.20
	Nighttime	91.35	**98.09**
Bikes	Daytime	99.24	99.48
	Nighttime	99.30	99.31

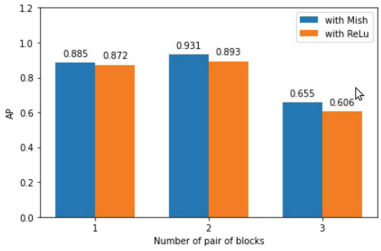

Fig. 7. Mish versus ReLU activation functions in SHELP license plate detection with various number of pair of blocks, evaluated on the **CCPD-2020** dataset.

Contribution of Mish Activation Functions. Figure 7 shows the comparison in Average Precision between the Mish and ReLU activation functions on the test set of CCPD-2020. The x-axis represents the number of pair of residual blocks (intermediate and down-sample) used in SHELP. It can be seen that Mish achieves a higher accuracy that the ReLU.

4.3 Results on License Plate OCR

The multi-line OCR results on the **uTVM-OCR** dataset are reported in Table 3. MCR achieved a very high AOS of 95.17%, 93.02%, 91.28% and 86.73% for recognition of license plate of four-wheel vehicle images and of bikes captured in day-time and in night-time, respectively. This accuracy is very significant given the fact that 83.7% of these license plates are two-lines.

Table 3. Accuracy of MCR license plate recognition on **uTVM-OCR** dataset.

Vehicle	Time	No. of images	AOS
Cars, trucks, buses	Daytime	8,419	95.17%
	Nighttime	1,579	93.92%
Bikes	Daytime	8,538	91.28%
	Nighttime	1,462	86.73%

4.4 End-to-End Results

So far we have evaluated MixLPR modules (license plate detection and OCR) separately. Since MixLPR is evaluated by the end user, it is important to know how license plate detection and recognition methods work in the joint system. We combine WPOD-NET and MCR as a baseline for comparison. Table 4 reports the results. As expected WPOD-NET + MCR is not as good as our MixLPR option (SHELP + MCR).

Table 4. Accuracy of sequence for end-to-end license plate detection and recognition methods on **uTVM-N2N** dataset.

Method	AOS
WPOD-NET + MCR	90.6
SHELP + MCR	**93.5**

4.5 Performance

For real-time systems, MixLPR uses SHELP (Sect. 3.1) for license plate detection due to the smaller model size. The combination of SHELP and MCR also exhibits higher accuracy for end-to-end recognition of license plate as shown in Table 4. These models were developed in Pytorch and Keras and were then converted into NVIDIA TensorRT for faster inference without loss of accuracy. In a workstation of 24-core CPU, 64G RAM, 2 RTX 2080Ti GPUs, the inference time of the models with different batch sizes is reported in Table 5. SHELP takes only 1.26, 15, 36 and 75 ms for inference with batch size of 1, 16, 32 and 64, respectively. On these batch sizes, the inference time of MCR is 6.2, 22.8, 64.8 and 86.8 ms, respectively. Together, these add up to the total running time of MixLPR of 7.46, 37.8, 100.8, and 161.8 ms, respectively. The speed of MixLPR is therefore faster than the end-to-end RPNet which is about 16.4 ms on Quadro P400 GPU.

Table 5. Inference time (in ms) of models with different batch sizes.

Model	Batch size			
	1	16	32	64
SHELP	1.26	15	36	75
MCR	6.2	22.8	64.8	86.8
SHELP+MCR	7.46	37.8	100.8	161.8

5 Conclusion

We have addressed the challenges of real-time detection and recognition of license plates in intelligent transportation systems. The system has to operate reliably in continuously changing contexts with high accuracy. Existing deep learning methods based on the convolutional and recurrent neural networks can only be efficient with grid and sequential data. Therefore, they are less effective or even inapplicable in heterogeneous environments such as those traffic in many of dense Asian cities, where license plates can be text block rather than a single line. We have presented a new framework dubbed MixLPR, which has been developed and validated in real scenarios. MixLPR is composed of a license plate

detector SHELP and a multi-line character recognizer (MCR). A comprehensive suite of experiments on two public and three private datasets demonstrated that MixLPR is fast and reliable, and it achieved competitive accuracy for both plate detection task and OCR task compared to state-of-the-art rivals. The end-to-end integration can also run fast at rate of 2.5 ms per image when run on a batch size of 64.

We plan several future works. One is to study the accuracy and scalability of MixLPR in the context of truly end-to-end traffic monitoring, starting from raw images from cameras to licence plate recognition of every vehicle in the visual field. It is also interesting to train the two modules of MixLPR in an end-to-end manner, rather than separately as in the current work.

References

1. Canziani, A., Paszke, A., Culurciello, E.: An analysis of deep neural network models for practical applications. CoRR arXiv:abs/1605.07678 (2016)
2. Goel, S., Dabas, S.: Vehicle registration plate recognition system using template matching. In: 2013 International Conference on Signal Processing and Communication (ICSC), pp. 315–318 (2013). https://doi.org/10.1109/ICSPCom.2013.6719804
3. Gou, C., Wang, K., Yao, Y., Li, Z.: Vehicle license plate recognition based on extremal regions and restricted Boltzmann machines. IEEE Trans. Intell. Transp. Syst. **17**(4), 1096–1107 (2016). https://doi.org/10.1109/TITS.2015.2496545
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
5. He, M.X., Hao, P.: Robust automatic recognition of Chinese license plates in natural scenes. IEEE Access **8**, 173804–173814 (2020). https://doi.org/10.1109/ACCESS.2020.3026181
6. Hendry, Chen, R.: Automatic license plate recognition via sliding-window darknet-YOLO deep learning. Image Vis. Comput. **87**, 47–56 (2019). https://doi.org/10.1016/j.imavis.2019.04.007
7. Wu, H.-H.P., Chen, H.-H., Wu, R.-J., Shen, D.-F.: License plate extraction in low resolution video. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 1, pp. 824–827 (2006). https://doi.org/10.1109/ICPR.2006.761
8. Laroca, R., et al.: A robust real-time automatic license plate recognition based on the YOLO detector. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–10, July 2018. https://doi.org/10.1109/IJCNN.2018.8489629
9. Misra, D.: Mish: a self regularized non-monotonic neural activation function. ArXiv arXiv:abs/1908.08681 (2019)
10. Montazzolli, S., Jung, C.: Real-time Brazilian license plate detection and recognition using deep convolutional neural networks. In: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 55–62 (2017). https://doi.org/10.1109/SIBGRAPI.2017.14
11. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
12. Selmi, Z., Ben Halima, M., Pal, U., Alimi, M.: DELP-DAR system for license plate detection and recognition. Pattern Recognit. Lett. **129**, 213–223 (2019). https://doi.org/10.1016/j.patrec.2019.11.007

13. Silva, S.M., Jung, C.R.: License plate detection and recognition in unconstrained scenarios. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11216, pp. 593–609. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01258-8_36
14. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826 (2016). https://doi.org/10.1109/CVPR.2016.308
15. Wang, Q.: License plate recognition via convolutional neural networks. In: 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 926–929 (2017). https://doi.org/10.1109/ICSESS.2017.8343061
16. Wojna, Z., et al.: Attention-based extraction of structured information from street view imagery. In: 14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, 9–15 November 2017, pp. 844–850. IEEE (2017). https://doi.org/10.1109/ICDAR.2017.143
17. Xu, Z., et al.: Towards end-to-end license plate detection and recognition: a large dataset and baseline. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11217, pp. 261–277. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_16
18. Yang, Y., Li, D., Duan, Z.: Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features. IET Intell. Transp. Syst. **12** (2017). https://doi.org/10.1049/iet-its.2017.0136
19. Zhang, Z., Yang, Z., Sun, Y., Wu, Y., Xing, Y.: Lenet-5 convolution neural network with mish activation function and fixed memory step gradient descent method. In: 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, pp. 196–199 (2019). https://doi.org/10.1109/ICCWAMTIP47768.2019.9067661
20. Zherzdev, S., Gruzdev, A.: LPRNet: license plate recognition via deep neural networks. CoRR arXiv:abs/1806.10447 (2018)
21. Zou, Y., et al.: A robust license plate recognition model based on Bi-LSTM. IEEE Access **8**, 211630–211641 (2020). https://doi.org/10.1109/ACCESS.2020.3040238