License Plate Localization for Low Computation Resources Systems Using Raw Image Input and Artificial Neural Network

Tjong Wan Sen[#1], Sinung Suakanto[*2], Amril Mutoi Siregar[^3]

[#]*Faculty of Computing, President University*
Jl. Ki Hajar Dewantara, Jababeka, Bekasi, Indonesia
[1]wansen@president.ac.id

[*]*Program Studi Teknik Elektro, Institut Teknologi Harapan Bangsa*
Jl. Dipati Ukur no. 80-84, Bandung, Indonesia
[2]sinung@ithb.ac.id

[^]*Faculty of Engineering and Computer Science, Buana Perjuangan University*
Jl. H. S. Ronggowaluyo, Teluk Jambe Timur, Karawang, Indonesia
[3]amrilmutoi@ubpkarawang.ac.id

Abstract— License Plate localization using Computer Vision needs a lot of computation resources. Thus, it is hard to deploy it on small systems. This paper presents an efficient license plate localization method using raw image input and artificial neural network. This is achieved by eliminating feature extraction stage and try to use as minimum as possible neural network architecture. Raw image input in dataset is cropped and labelled manually from random car images and video frames. The minimum architecture of the model has only three layers and 32,770 neurons. This is feasible to be deployed in today most single chip systems. The results, from various experiments, yield more than 90% of localization accuracy.

Keywords— computer and information processing, image analysis, image processing, object detection, license plate localization

Abstrak— Nomor plat kendaraan bermotor yang diperoleh dengan menggunakan Computer Vision membutuhkan banyak daya komputasi. Hal ini menyebabkan implementasinya ke dalam sistem minimum yang sederhana menjadi tidak mudah. Dalam penelitian ini, dikembangkan sebuah metoda untuk mendapatkan plat nomor kendaraan bermotor yang effisien menggunakan masukan langsung tanpa ektraksi ciri dan jaringan saraf tiruan. Penghematan daya komputasi dicapai dengan cara menghilangkan tahap ekstraksi ciri dan penggunaan arsitektur jaringan saraf tiruan yang seminimum mungkin. Citra masukan diperoleh dengan cara memotong dan memberi label gambar mobil dan frame video yang diperoleh secara acak. Arsitektur minimum yang dihasilkan berupa model yang hanya terdiri dari tiga lapisan dan 32,770 neuron. Model ini cukup fisibel untuk diterapkan pada kebanyakan system on a chip yang ada pada saat ini. Tingkat akurasi model dalam menemukan lokasi nomor kendaraan dari berbagai eksperimen berhasil mencapai lebih dari 90%.

Kata Kunci— pemrosesan informasi dan komputer, analisis citra, pemrosesan citra, deteksi objek, ekstraksi plat nomor kendaraan bermotor

I. INTRODUCTION

The Automatic License Plate Recognition (ALPR) is the main technology in almost any Intelligent Transportation Systems (ITS). ITS is needed by every city to provide good traffic management, enhance security, make better documentation, and mainly saving resources. Current ALPR systems consume a lot of computation resources such as processing cycles (CPU), memory (RAM and storage), and the most important one, energy to operate 24/7. To reduce resources requirements and consumptions in ALPR, systems and algorithms simplification should be done. One way to achieve it is to develop more efficient module in ALPR, which is license plate localization.

This reduction is achieved in two stages. First is eliminating complex feature extraction stages, which is common in most ALPR. Raw image from video frame is used as input directly without any processing. This would save a lot of resources and make faster system response. The second step is closely related to the first one, which is artificial neural network (ANN) model. ANN model trained by deep learning techniques is capable to find out features by itself from almost any training dataset. With or without feature extraction stage. ANN model shows a lot of good pattern recognition results on many different dataset such as speech, text, audio or image dataset today. Even though ANN model needs a lot of computational resources in training stage, it only needs few in testing stage. With simpler activation functions available today, it only depends on the size of the model. That is why to save more computation resources; ANN model trained by deep learning techniques architecture size must be as minimum or small as possible. Fewer neurons mean fewer computation resources. This minimum neural network architecture is produced via experiments. The target is to have a model, which is feasible to be deployed in today single chip systems to preserve energy but fast enough to be used to detect most vehicles with quite high speed.

The license plate is country-specific and has many different variation of the information and format. For example in Fig. 1,

there are some images of license plate from three different countries. This makes ALPR system for one country usually is not good to be used in another country. In this research, we try to overcome those variation using dataset which contains many different license plate images. The difference because of colour is handled by converting all images into grayscale format. This will reduce variation significantly.

This pape presents a new and simple license plate localization using raw image input and deep learning. The rest of the paper is organized as follows. In section II, system model and methods of the proposed license plate localization module and methods are discussed. Raw image input from car images and fixed camera video frame as dataset preparation, deep learning model architecture development and training are discussed in this section. Experimental results are discussed in section III. Finally, this is followed by the section IV for conclusion and some possibility of future work.

II. Methodology

A. License Plate Localization

License plate localization module plays an important role in ALPR. It could reduce computation resources significantly if handled properly. Since in most ALPR systems image does not have license plate, so there is a big opportunity to save computation resources. If an image has license plate in it, then it would be processed further by recognizer or search algorithm. Otherwise, if there is no license plate detected, then the image could be simply dropped and the process continue with the next available image. But there is a trade-off, simpler license plate localization algorithm (uses fewer resources) usually less accurate, for example, gets more false positive results. That is why some researchers tend to use more complex algorithm to increase accuracy. For example [1] uses up to 16 statistical features using Vertical Projection technique, Discrete Fourier Transform, and K-means clustering, while still preserving real-time processing for the system. Classifier in [1] is multilayer perceptron neural network technique to identify the location of a license plate and could achieve 99.1% accuracy. Heavier classifier/recognizer could be used to improve accuracy. For example in [2] a single convolutional neural network from YOLO (You Only Look Once) real-time object detection system [3] is used and could achieve 98.6% of overall mean average precision. Simpler system in [4] uses Harris corner algorithm and connected component analysis method to achieve lower results which is 93.84% overall accuracy.

Fig. 1 Example of two plate licenses from Indonesia (top row) and another two images from other countries (bottom row). All images belong to plate class of our dataset and before converted to grayscale yet.

B. Raw Image Input

Raw input is proven now since deep learning has brought a lot of breakthroughs in discovering hidden patterns from almost any datasets [5]. This could be used to eliminate one of the hardest part in pattern recognition which is feature engineering. Related to this research, this raw input is used to save computation resources. Raw dataset has been used a lot in many images or video [6], [7], [8], speech or audio [9], [10], [11], [12], [13], text, and other datasets [14]. In most pattern recognition systems, feature extraction stage is the second big computation resources consumer after searching phase. Even though for simple feature extractor such as shallow depth Haar or Daubechies wavelet transform, with efficient implementation, only needs few computation resources, the one with more depth does not [15]. In [16], more complex wavelet mother computation resources consumption indeed increase significantly. In addition, results from training phase would even give researcher insights for new feature engineering. This could be investigated further for example from complexity and accuracy trade-off point of view.

C. Deep Learning

Successful Deep Learning based methods for license plate localization have been published in many papers. Most of them are using Convolutional Neural Network (CNN) [17]. This is because of CNN has good or most suitable for image recognition or computer vision. Although there are a lot of good results from CNN, this type of network is usually quite complex and needs more computation resources. The reason is CNN has to scan an input image (two dimensional array of pixels) using certain filter to extract simple feature from a small region of the image one at a time. All results of this process form another two dimensional array via pooling mechanism. This steps are repeated until good result is achieved [18]. And at the end, fully connected layer would perform classification or recognition. In this research, fully connected layer is used directly without convolutional and pooling layer.

In [19], Region-based Convolutional Network (R-CNN) is proposed to improve mean average precision. R-CNN uses up to 2,000 bottom up region proposals per image and only apply CNN to those regions. Thus in a way also save computational resources. Fast R-CNN [20] improves speed and storage requirement by eliminating multi-stage training and Support Vector Machine as classifier from R-CNN and introducing VGG16 [21] instead. In Faster R-CNN [22] Region Proposal Network is used to share convolutional features with Fast R-CNN in training phase. Faster R-CNN focuses on reducing region proposal computation bottleneck.

In comparison to fully connected neural network (dense) used in our license plate localization methods, from architecture point of view, CNN based methods are more complex or need more computational resources. Both in training and testing phase. Our proposed method, based on experiments results, also contain fewer number of layers (also type of layers) and number of neurons.

The ability of ANN with deep learning techniques to learn from a lot of data (Big Data) is based on Back Propagation (BP) algorithm. First, deep learning uses this training algorithm to compute the forward pass of the neural network from the beginning to the end. From the input layer all the way to the output layer. For every neuron in every layer, the total value is calculated. The total value is simply a multiplication result of the weight and the input value of each neuron. After that the algorithm would get the total using summation operation. Then to get the output from each neuron, an activation function would be used. This output becomes input for the next layer. The end results of one forward pass process, is considered as the best prediction of the data or feature given to input layer at that iteration. This best prediction is compared to the desired result (from the label). If not the same, then error is produced. Depends on how big the error is, the algorithm would update all neuron weights using certain rule. If the error is positive, the weight value would be reduced. If the error is negative, the weight values would be increased. This updating process is done backwardly (backward pass). Start from output layer all the way to input layer. The process of forward pass and backward pass to update weights is repeated until the error is zero or minimum (below certain value) [5]. To save computation resources, several methods are proposed. Method such as Stochastic Gradient Descent (SGD) algorithm is one of the successful method. Improvement of this method could be seen in asynchronous stochastic gradient descent (ASGD) algorithm. This algorithm could be used to speed-up training time on parallelize computing with multi-GPU [22]. This is suitable mostly for big deep learning architecture with a lot number of layers and neurons.

In relation with saving computation resources, in this research, training phase is not important. Even though updating deep learning weights in training phase is very exhausted, even with improved BP algorithm such as SGD or ASGD, this could be done at a big machine or server or even cloud. The computation resources needed in testing phase, which is done at small or minimum systems, is only a fraction of it (training phase). Once the model is downloaded into these minimum systems, the computation resources needed for testing phase in this proposed method is as simple as only multiplication, summation, and activation function. The number of multiplication, summation, and activation function needed is determined by the number of neuron in the deep learning architecture. If the number of neuron is fewer, then less computation resources is needed. The number of neuron in the architecture is determine mostly by the input size (number of pixel).

Figure 2 shows steps that are used in the proposed method. First we are breaking image input from still image or video frame into several region of interest. Second, each region is then labelled accordingly. Third, each region is converted into raw pixel values and together with label becomes dataset. Fourth, artificial neural network architecture is prepared and deep learning techniques is used for training. Last step, fifth, model accuracy in determining whether a region has license

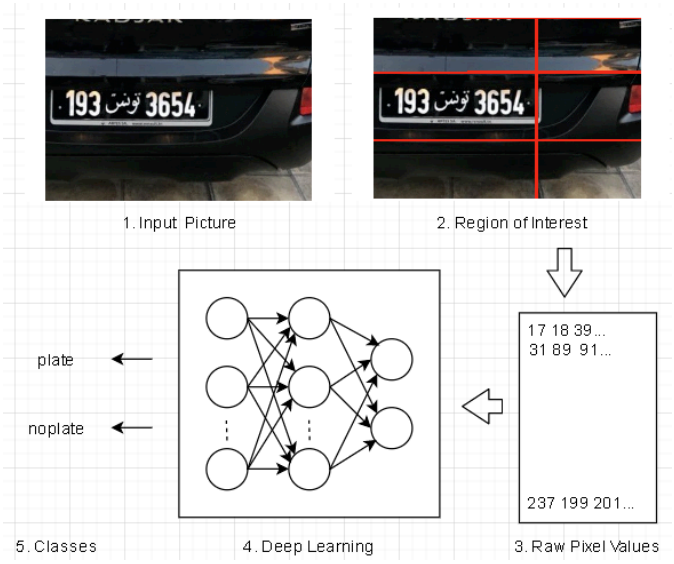

Fig. 2 License Plate Localization Using Raw Image Input and ANN with Deep Learning techniques: Input picture (1) is broken down into many region of interest (2). Each region is converted to a set of integer values (3) after converted to grayscale. Feed it to ANN model and use deep learning training tools (4) and get the plate or no plate class decision (5).

plate number or not is measured. Detail description of steps in methodology is as followed:

1) Dataset

Images in dataset are taken from random Google images and YouTube videos. Most license plates are from Indonesia. Some are from other countries such as United States and European countries. Images from Google images, first, are cropped manually by human operator team (operator) into 256x64 pixels size. With license plate position is at the centre of each image. The numbers are determined using experiments on different pixel size dimension by operator. Smaller number would give smaller deep learning architecture but not easy for operator to judge. This represents our region of interest (RoI). Second, operator would label that image with '0 1' to indicate license plate is present. All labels are in One Hot Encoding (OHE) format. Third, all images would be converted to grayscale. This would reduce computation resources significantly since, in ALPR, the colour information is not very relevant. Every image converted to grayscale still preserves the same important information needed by recognition phase. These all images will represent 'plate' class. For 'no plate' class, up to eight images would be taken from around the license plate position. Operator would verify manually that there is no license plate present in every image. For these images label '1 0' would be used to indicate no license plate is present. This would simplify the problem into two classes' object recognition problem. Since there are only two classes, which are, object exists or object does not exist.

Images from YouTube videos would be acquired using similar steps after frame extraction process. Sequence of images are extracted each video using software tool. Input size is determined by experiments. Based on license plate general

size, we used 4 to 1 ratio. We considered general license plate dimension and two to the power of n (2n) height and width pixel to simplified calculation. Smaller height and width value or number of pixel could be used such as 128x32 but introduced problem for human operator. Image with 32 pixel of height is not easy to see without zoom. Even though this would reduce computation resource up to 50%, we decided not to choose it in this research. Input size of 128x64 pixel is not used too because of most of the license plate images become truncated (not full) or too small.

To enhance dataset quality, for every image, another or different human operator team will verify manually and match the label accordingly. Mechanism for operator to judge license plate existence are: 1) license plate exists - license plate could relatively easy to see with bare eyes without zoom or other image processes, 2) license plate does not exist – it is easy to see that the license plate is not there visually or logically, and 3) if license plate exists but not clear for example because it is too small (operator is not sure) or there is another object that is look like a license plate, then current RoI will be excluded from dataset. Dataset consist of 120 images for plate class and 1100 images for no plate class. Four images from plate class is shown by Fig. 1. All RoI images are then converted into integer and sent to normalization process. The results are fed directly into deep learning architecture as a raw input without any feature extraction processes.

2) Deep Learning Architecture

Deep learning architecture is determined by experiments. First, starting with minimum architecture using as few as possible neurons. The minimum architecture would consist of one input layer, one hidden layer, and one output layer. According to raw image input and number of classes from dataset, there would be 256x64 neurons for input layer and only two neurons for output layer. Hidden layer would have the same number of neurons with input layer, which are 256x64 neurons. After that biases would be added if needed. Second, hyper-parameter set is determined using grid search to find the best or highest accuracy and avoid over fitting. The hyper-parameters are activation function, learning rate, optimization algorithm, dropout rate, and batch size (if needed). Third, for hidden layer, there are another two choices to test, which are expanded (using 50% more than input neurons) and compressed (using 50% less than input layer neurons). Expanded version could contrast the details of differences between license plates while compressed version could determine dominant pattern from all license plates in the dataset. After that, adding biases and hyper-parameters tuning are repeated.

Experiments on dataset are done with several approaches. First, the basic by dividing dataset into training and testing set. Training set 90% with testing set 10% and training set 80% with testing set 20% schemes are used. To further increase model accuracy, the same training data is used for model training and testing phase using 5-fold cross-validation or 80-20 ratio and 10-fold cross-validation or 90-10 ratio. Since the number of the images from both classes are not the same (unbalanced), plate class has fewer images than no plate class, we also used undersampling and oversampling methods in experiments.

III. Result and Discussion

Fig. 3 shows one of the best training results. Both loss and accuracy for training and validation are shown. Loss values continue to drop both for training (0.3648) and validation (0.4301) but not significant (below 0.4%) after epoch 120. Training accuracy achieves highest value (0.9375) start from epoch 46 and could not increase. The same for validation accuracy at 0.9124 start from epoch 69 and stop to improve. Deep learning architecture that achieves these results consists of three layers and 32,770 neurons. Activation function for input and hidden layer is ‘relu’ and for output layer is ‘softmax’. Loss function 'categorical_crossentropy' and SGD optimizer with 0.00001 learning rate are used. This result is quite promising (more than 90% of accuracy for training and validation) considering the dataset size is small (120 of license plate images and 1100 images without license plate).

IV. Conclusion

Simple License Plate Localization using Raw Image Input and Deep Learning to reduce computation resources requirement has been successfully developed. This method achieved license plate localization accuracy higher than 90% using minimum architecture with only three layers and 32,770 neurons. For the future this research would try to 1) simplified

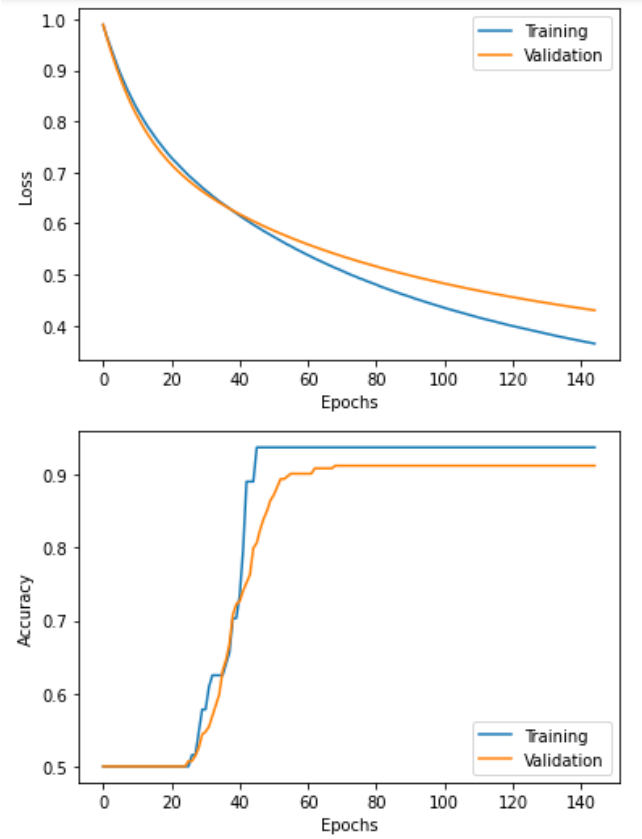

Fig. 3 Loss (top) and Accuracy (bottom) from one of the best training result. Both training and validation are shown.

the deep learning architecture to multilayer perceptron, using fewer than three layers, to further reduce number of layers and number of neurons, 2) use more classes to improve accuracy for license plate that is not at the center of image, 3) use frame sequences information (license plate location movement) to determine plate/no plate final decision, and 4) implement this method into a minimum device to measure its real time performance.

ACKNOWLEDGEMENT

This research was enabled in part by support from Research and Community Development Center of President University.

REFERENCES

[1] M. Rezaei and M. Isehaghi, "An efficient method for license plate localization using multiple statistical features in a multilayer perceptron neural network," in *9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium*, 2018.

[2] Y. Jamtsho, P. Riyamongkol, R. Waranusast, "Real-time Bhutanese license plate localization using YOLO," *ICT Express 6*, 2020 pp. 121-124.

[3] J. Redmon, A. Farhadi, "YOLO9000: better, faster, stronger," *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263-7271

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, vol. 0, pp. 580-587.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, 2015, vol. 521, pp. 436-444.

[6] J. Liu, C.-H. Wu, Y. Wang, Q. Xu, Y. Zhou, H. Huang,C. Wang, S. Cai, Y. Ding, H. Fan, and J. Wang, "Learning raw image denoising with bayer pattern normalization and bayer preserving augmentation," *CVPR*, 2019.

[7] X. Xu, Y. Ma, and W. Sun, "Towards real scene super-resolution with raw images," *CVPR*, 2019.

[8] A. Schwartzman, M. Kagan, L. Mackey, B. Nachman, L. De Oliveira, "Image processing, computer vision, and deep learning: new approaches to the analysis and physics interpretation of LHC events," *IOP Publishing Journal of Physics: Conference Series*, 762, 2016.

[9] W. S. Tjong, "Voice activity detector for device with small processor and memory," *International Conference on Sustainable Engineering and Creative Computing (ICSECC)*, 2019, pp. 212-217.

[10] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur, "Acoustic modelling from the signal domain using CNNs," *Interspeech*, 2016.

[11] Z. Tuske, P. Golik, R. Schluter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," *Interspeech*, pp. 890-894, 2014.

[12] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Interspeech*, pp. 1-5, 2015.

[13] R. Z. Candil, T. N. Sainath, G. Simko, and C. Parada, "Feature learning with raw-waveform CLDNNs for voice activity detection," in *Interspeech 2016*, pp. 3668-3672, San Fransisco, USA, 2016.

[14] S. Kohn, E. Racah, C. Tull, D. Dwyer, Prabhat, and W. Bhimji, "Deep learning with raw data from Daya Bay," *IOP Conf. Series: Journal of Physics*, 898, 2017.

[15] W. S. Tjong, B. R. Trilaksono, A. A. Arman, R. Mandala, "Robust automatic speech recognition features using complex wavelet packet transform coefficients," *Journal of ICT Research and Applications*, 2009, vol. 3, no. 2, pp. 123-134.

[16] W. S. Tjong, B. R. Trilaksono, A. A. Arman, "Evaluation of wavelet transform coefficients for robust speech recognition feature vectors," *Proceedings International Conference on Electrical Engineering and Informatics*, 2007.

[17] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, Contour and Grouping in Computer Vision*, 1999, pp. 319–344.

[18] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *TPAMI*, 2015.

[20] R. Girshick, "Fast R-CNN," in *ICCV*, 2015, pp. 1440-1448.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[22] F. Seide, H. Fu, J. Droppo, G. Li and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in *Proc. of Interspeech*, Singapore, 2014.

Tjong Wan Sen, born in Kediri, Indonesia. He received Ph.D. degree from Institut Teknologi Bandung in 2009. Since 2010, he has been with the Faculty of Computing, President University, Jababeka, Indonesia, where he is currently a lecturer and researcher. His research interests include automatic speech/speaker recognition, computer vision, artificial intelligence, and embedded systems.

Sinung Suakanto, born in Klaten, Indonesia. He received Ph.D. degree from Institut Teknologi Bandung. Since 2006, he has been with Institut Teknologi Harapan Bangsa, Bandung, Indonesia, as a lecturer and researcher. His research interests include application development, computer and telecommunication network, artificial intelligence, IoT, and big data.

Amril Mutoi Siregar, born in Padang, Indonesia. He is currently a Ph.D. candidate from Institut Pertanian Bogor University. Since 2018, he has been with the Faculty of Engineering and Computer Science, Buana Perjuangan University, Karawang, Indonesia, where he is currently a lecturer and researcher. His research interests include computational intelligence and optimization, computer vision, data mining, text mining, machine learning, and deep learning.

Halaman kosong