Recognition of license Plate by using Improved Neural Network

Yashmeet Kaur Reel[1], Abhinav Shukla[2]

[1]M.Tech Scholar, Department of ECE
[2]Asst. Prof., Department of ECE
Vedica Institute of Technology, RKDF University, Bhopal, India
yashukaurreel@gmail.com, abhinav.shukla@hotmail.com

* Corresponding Author: Yashmeet Kaur Reel

Abstract

In recent years, recognition of license plate has become a crucial role in the development of smart cities for vehicle management, investigation of stolen vehicles, and traffic monitoring and control. License plate recognition system has different three stages, including localizing of license plate, segmentation of character, and recognition of character. Although the recognition system of license plate has been successfully applied to the environment-controlled smart parking system, as it is very useful in smart cities but it still faces many challenging in the surveillance system such as congested traffic with multiple plates, ambiguous signs and advertisements, tilting plates, as well as obscure images taken in bad weather and night times.

In this paper, we propose an efficient license plate recognition system by using improved neural network as old neural network has some disadvantages and one of them is that the network is reduced to a certain value of the error on the sample means that the training has been completed. This value does not give us optimum results this is the reason why we are using improved neural network that first detects vehicles and then retrieves license plates from vehicles to reduce false positives on plate detection. Then, we apply convolution neural networks to improve the character recognition of blurred and obscure images. The experimental results show the superiority of the performance in both accuracy and performance in comparison with traditional license plate recognition systems.

Keywords: License plate recognition system, convolution neural networks, smart city, ANN, OCR, BPNN.

I. Introduction

Vehicles plate detection and recognition appear in vast variety of applications, including travel time estimation, car counting on highways, traffic violations detection, and surveillance applications [1]. Traffic monitoring cameras are mounted four to seven meters above the street level. Plate recognition range, where the cameras are able to capture the vehicles plates with sufficient resolution, starts from 20 to more than 50 meters away from the camera location. This range depends on the camera resolution and the lens mounted on the camera. At these heights and distances, vehicles plates are not as clearly visible as in other applications such as toll and parking fee payment systems. High camera installation point causes some difficulties against the correct detection of vehicles plates. Vehicles with dirty plates make the situation even more complicated. On the other hand, number plate is the only trustworthy identity of a vehicle in Intelligent Transportation Systems (ITS) and correct vehicle identification depends highly on the accuracy of automatic number plate recognition (ANPR) systems. An ANPR system consists of three different modules:

a) Monochrome/Color cameras.

b) IR projector, and

c) The processing board.

In addition to compatibility of interfaces, each section must be chosen properly for a specific application. In this dissertation, a detailed exploration on the important parameters of an ANPR module has been done. Basically, the License Plate Recognition (LPR) process is divided into three main parts: Plate Detection, Character Segmentation, and Character Recognition. Each of these parts plays an important role in the final accuracy. Many problems such as size variations, viewing angle, low contrast plates, vehicles high speed and time consuming algorithms have prevented researchers from introducing a single class of algorithms to solve the problem. There have been, however, many algorithms proposed for each part.

License Plate Detection

With increasing number of vehicles on roads, it is getting difficult to manually enforce laws and traffic rules for smooth traffic flow. Traffic Management systems are installed on traffic signals to check for vehicles breaking the traffic rules. In order to automate these processes and make them more effective, a system is required to easily identify a vehicle. The important question here is how to identify a particular vehicle? The obvious answer to this question is by using the vehicle's number plate as every vehicle has a unique number through which it is easily differentiated from other vehicles. Vehicles in each country have a unique license number, which is written on its license plate. This number distinguishes one vehicle from the other, which is useful especially when both are of same make and model. An automated system can be implemented to identify the license plate of a vehicle and extract the characters and numbers from the region containing a license plate. The license plate number can be used to retrieve more information about the vehicle and its owner, which can be used for further processing. Such an automated system should be small in size, portable and be able to process data at sufficient rate. Various license plate detection algorithms have been developed in past few years. Each of these algorithms has their own advantages and disadvantages. The main objective of the proposed design is to detect a license plate number from an image which is captured from camera. An efficient algorithm is proposed to detect a license plate under various conditions. This algorithm extracts the license plate data from an image and provides it as an input to the stage of License number Plate Recognition.

Figure1: ANPR Block diagram

This paper focuses on design, deployment and evaluation of a system for two industrial applications: detecting traffic violations at urban intersections and vehicle counting on highways. These two applications are among the most important ones in the ITS industry.

An ANPR system consists of three different modules:

a) Monochrome/Color cameras,

b) IR projector, and

c) The processing board.

In addition to compatibility of interfaces, each section must be chosen properly for a specific application.

For plate detection, several algorithms have been proposed. Some of these algorithms are based on finding image edges, such as horizontal and vertical edges [2]. For example, in [3], plates are localized using the Canny edge detector. Sobel operator is used by some other methods that work based on detecting image edges. These methods have two main advantages: smoothing the image noise because of the included averaging, and generating thick and bright edges because of the involved differentiation on two rows and columns [4]. The advantage of edge detector methods is their low computational complexity and memory requirements. In some other algorithms, plate detection is performed by finding the borders of a plate using the Hough transform [8], which is a memory and time consuming process. Such method fails in detecting plates without clear borders. Fig. 2 compares the results of applying the Hough transform on both clean and dirty plates. Wavelet analysis has also been utilized for detecting plates. In Wavelet based methods, high-frequency coefficients are used to detect plate candidates. Since these coefficients correspond to the edges, these algorithms suffer from the same disadvantages of edge detection algorithms. In [5], and [6], color is incorporated as an important feature in detecting plates. These algorithms fail on gray-scale images or images with low color disparity. Some detection algorithms are based on a combination of Mathematical Morphology and Connected Component analysis [7]. There are algorithms which first enhance the plate contrast and then apply the detection algorithm. Most of these algorithms are successful in identifying clean plates, but fail when it comes to detecting dirty and low contrast plates. This is due to the fact that these algorithms need a medium to high contrast images for plate detection. Moreover, for dirty plates color is not a reliable feature for plate detection. For character segmentation, there are many algorithms based on morphological operations and connected component analysis (CCA) [8]. In such methods, it is necessary to apply a proper thresholding method to obtain a binary image of the plate before any further processing. For example, CCA, which has been used in many research works, depends highly on the previously applied thresholding method. Applying such algorithms on plate candidates relies on appropriate setting of the involved parameters. Unfortunately, at the detection step there is no information about the input plate quality and the parameters cannot be tuned appropriately. Therefore, the recognition accuracy of such algorithms decreases when different plate qualities are involved. For character recognition, many different classification tools and techniques have been utilized so far, such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), Bayes classifier, and K-nearest neighbor and so on. Classifiers are applied to the features extracted from the image segmentation.

II. LITERATURE REVIEW

2.1 Background:

Vehicle License plate detection and recognition appear in vast variety of applications, including travel time estimation, car counting on highways, traffic violations detection, and surveillance applications. Traffic monitoring cameras are mounted four to seven meters above the street level. Plate recognition range, where the cameras are able to capture the vehicles plates with sufficient resolution, starts from 20 to more than 50 meters away from the camera location. This range depends on the camera resolution and the lens mounted on the camera. At these heights and distances, vehicles plates are not as clearly visible as in other applications such as toll and parking fee payment systems. High camera installation point causes some difficulties against the correct detection of vehicles plates. Vehicles with dirty plates make the situation even more complicated.

On the other hand, number plate is the only trustworthy identity of a vehicle in Intelligent Transportation Systems (ITS) and correct vehicle identification depends highly on the accuracy of automatic number plate recognition (ANPR) systems. Basically, the License Plate Recognition (LPR) process is divided into three main parts: Plate Detection, Character Segmentation, and Character Recognition. Each of these parts plays an important role in the final accuracy. Many problems such as size variations, viewing angle, low contrast plates, vehicles high speed and time consuming algorithms have prevented researchers from introducing a single class of algorithms to solve the problem. There have been, however, many algorithms proposed for each part. For plate detection, several algorithms have been proposed. Some of these algorithms are based on finding image edges, such as horizontal and vertical edges [9]. For example, in [10], plates are localized using the Canny edge detector. Sobel operator is used by some other methods that work based on detecting image edges. These methods have two main advantages: smoothing the image noise because of the included averaging, and generating thick and bright edges because of the involved differentiation on two rows and columns [11]. An example of Sobel edge detection is shown in Fig. 2.1. (a) and (b).

(a) (b)

Figure 2: Results of applying the Sobel edge detector on (a) a clean plate and (b) a dirty plate

The advantage of edge detector methods is their low computational complexity and memory requirements. In some other algorithms, plate detection is performed by finding the borders of a plate using the Hough transform, which is a memory and time consuming process. Such method fails in detecting plates without clear borders. Wavelet analysis has also been utilized for detecting plates [12]. In Wavelet based methods, high-frequency coefficients are used to detect plate candidates. Since these coefficients correspond to the edges, these algorithms suffer from the same disadvantages of edge detection algorithms. In [13], and [14], color is incorporated as an important feature in detecting plates. These algorithms fail on gray-scale images or images with low color disparity. Some detection algorithms are based on a combination of Mathematical Morphology and Connected Component analysis. There are algorithms which first enhance the plate contrast and then apply the detection algorithm. Most of these algorithms are successful in identifying clean plates, but fail when it comes to detecting dirty and low contrast plates. This is due to the fact that these algorithms need a medium to high contrast images for plate detection. Moreover, for dirty plates color is not a reliable feature for plate detection.

2.2 License Plate Detection Using Neural Network

Conventional license plate recognition system has three stages, including license plate localization, character segmentation, and character recognition. The first stage of license plate localization belongs to the object detection approach, including object localization, feature extraction, and image classification in three stages. Since images may have many different sizes of objects, conventional object detection approaches such as Deformable. There is a need to identify Parts Models (DPM) [15] use different sized sliding windows to scan the entire image to obtain candidate localizations. And then, object features are extracted from candidate localizations using scale-invariant feature transform (SIFT). Finally, the object features obtained by SIFT are submitted to the support vector machines (SVM) for classification. The approach of sized sliding windows causes a lot of unnecessary computation. To reduce unnecessary computations, the regional CNN (RCNN) [16] first predicts about 2,000 to 3,000 regional proposals through selective search, then adopts CNN models to extract features from regional proposals and finally completes classification by SVM. After the classification is completed, RCNN optimizes the detection results through the bounding-box regression. The RCNN has two major drawbacks. The first is that the RCNN requires each region proposal to pass the CNN forward, resulting in a large amount of repetitive computations for each single image. The second disadvantage is that it has to train three different models separately. A CNN that generates image features a classifier that predicts classes, and a regression model that refines the bounding boxes. This makes RCNN extremely difficult to train.

2.3 Research Gap Found

Han Xiang, et al. proposed [17] “Lightweight Fully Convolutional Network for License Plate Detection”. Dense connections and dilated convolutions are adopted for combing multilevel and multiscale vision features. A fusion loss structure is appended during training to further improve prediction accuracy. it significantly reduces inference time and storage space without sacrificing detection ratios. In a word, our proposed method is not suitable for many tasks that require slow detection.

Vijeta Khare, et al. proposed [18] “A Novel Character Segmentation-Reconstruction Approach for License Plate Recognition”. The authors introduce partial character reconstruction to segment characters.

- Angular information is explored for finding spaces between characters.

- Stroke width properties in different domains are used for shape restoration.

Yousri Kessentini,et al. proposed [19] “A two-stage deep neural network for multi-norm license plate detection and recognition”. The authors proposed two recognition engines are compared in this work: a segmentation-free approach based on a convolutional recurrent neural network where the recognition is carried out over the entire LP image without any prior segmentation and a joint detection/ recognition approach that performs the recognition on the plate component level.

Muhammad Rizwan,et al. proposed [20] ”License plate detection for multi-national vehicles: An illumination invariant approach in multi-lane environment”. The authors propose a novel illumination invariant method to handle multi-national vehicle license plates of different colors and styles. Red corona is initially used to detect the tail-lights of vehicles to establish region of interest as the license plates are in a vicinity of its tail-lights. The vertical edges within each region-of-interest are obtained using a unique approach that preserves license plate edges for improved performance. Heuristic energy map is then used to distinguish the license plate area. To validate the detected regions, high-level features extracted from AlexNet Convolutional Neural Network are used.

Tomas Bjorklund, et al. proposed [21] “Robust License Plate Recognition using Neural Networks Trained on Synthetic Images”. A complete vehicle license plate reading system is proposed

• CNNs trained for LPR on only synthetic samples generalizes well to real images

• A re-classification of fully convolutional classifications improves performance.

III. PROPOSED METHOD

3.1 Proposed System

License plate recognition systems have received a lot of attention from the research community. With the rapid growth in the number of vehicles, there is a need to improve the existing systems for identification of vehicles. A fully automated system is in demand in order to reduce the dependency on labour. However, it gained much interest during the last decade along with the improvement of digital camera and the increase in computational capacity. It is simply the ability to automatically extract and recognition a vehicle number plate's characters from an image. In essence it consists of a camera or frame grabber that has the capability to grab an image, find the location of the number in the image and then extract the characters for character recognition tool to translate the pixels into numerically readable character.

Phases of proposed system are shown below:

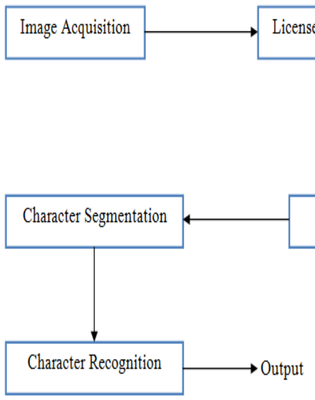

Figure 3: Framework of the system

These are explained as follows:

Image Acquisition:

This is the first phase in the system. This phase deals with acquiring an image by an acquisition method. In our proposed system, we used an image of a license plate.

License Plate Extraction:

License Plate Extraction is a key step in the system, which influences the accuracy of the system significantly. This phase extracts the region of interest i.e. the license plate from the acquired image.

Character Segmentation:

License Plate Segmentation, which is sometimes referred to as Character Isolation takes the region of interest and attempts to divide it into individual characters. In the proposed system segmentation is done by using bounding region method.

Character Recognition:

The last phase in the system is to recognize the isolated characters. After splitting the extracted license plate into individual character images, the character in each image can be identified. In the proposed system, we are using normalized back propagation neural network for character recognition.

3.2 Proposed Method for License Plate Recognition

In early days, template matching techniques were used for license plate recognition but these were sensitive to noise, so neural networks are used for recognition. Neural network has capability of learning, i.e. how to do tasks based on data given for training. Back propagation is supervised form of learning. In License plate recognition system, features extracted from license plate are used as input to neural networks and these are allowed to propagate forward to generate output. Learning of neural networks is done by loading targets and features extracted from license plate characters. Testing of neural network is done and parameters used for performance evaluation i.e. recognition rate and training time is computed.

Back Propagation Neural Network (BPNN), is a Multilayer Neural Network which is based upon back propagation algorithm for training. This neural network is based upon extended gradient descent based Delta learning rule, commonly known as Back Propagation rule. In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights. In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back propagation algorithm is the most widely used method for determining the derivative of weights. The back propagation algorithm is easiest to understand if all the units in the network are linear. In this network, error signal between desired output and actual output is being propagated in backward direction from output to hidden layer and then to input layer in order to train the network.

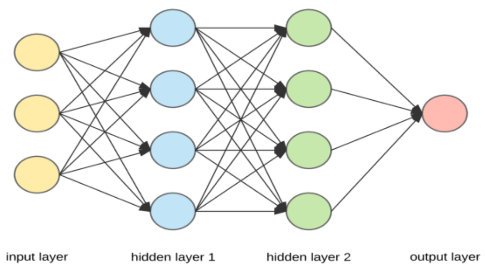

Figure 4: Back propagation neural network.

The advantage of this process is that the hidden layers between the input and output layer adjusts themselves in such a way that the different neurons are able to recognize different patterns about the input vectors. When the training is finished and a new unseen input vector containing noise is given to the network, the neurons in the hidden layers will fire if the new input vector contains a pattern the network learned to recognize during training. This pattern may be part of a larger feature characteristic for one of the classes the network tries to classify. The activation function is an important part of a neural network as it defines the output of a node given a set of inputs. It can be seen as a switch that may turn on or off a neuron depending on the given input.

Parametric ReLU (Rectified Linear Unit) is given by

$$f(x) = max(ax, x)$$

where $a <= 1$ is the leakage coefficient. It allows a small non-zero gradient when the neuron is not active. It performs nearly identical to normal ReLU, but is shown to converge slightly faster.

Batch normalization is a regularization step executed to solve a problem called *internal covariate shift*. Internal covariate shift is the phenomena that the distribution of network activations changes due to the change in network parameters. When weights in a specific layer are updated, the distribution of output vectors from that layer is also changed. These output vectors are input vectors for the next layer forcing it to adapt to the drift in input distribution slowing down the learning. A small perturbation such as an outlier in the initial layers may lead to a large change in the later layers resulting in the network requiring more training epochs to compensate for the outlier.

Batch normalization solves this problem by making it look like all layer inputs are normalized. The normalization step uses a linear transformation of the vectors called a *whitening transformation*. All variables are changed such that their covariance matrix is the identity matrix resulting in all underlying correlated data to be eliminated. Instead of performing full whitening of every layers input which is very costly, the mean is set to 0 and the variance of all the vectors is set to 1 after each mini-batch meaning only the scalar features need normalization. Batch normalization massively speeds up the training process as the learning rate can be increased without destabilizing the training process.

The neural network contains 3 layers, an input layer (input image pixels serve as the input to it), a hidden layer and an output layer. Each neuron in the output layer categorizes the input set to a classified output. Meaning, for a character 'a' to be trained, only the neuron which classifies character 'a' fires a value close to 1, the rest fires a value close to 0. The accuracy of the neural network is defined as how close the actual is to the target output, which is improved by decreasing the error defined per this formula:

$$\text{Total Error} = \tfrac{1}{2}\sum (\text{out}_z - \text{target}_z)^2$$

Where z is the output neuron. This approach of decreasing the mean squared error is called "The Gradient Descent", used in conjunction with a training algorithm called "Back Propagation", adjusting the network's weights in such a way that the total error converges to a global minimum. The weights and biases are initialized to random numbers between -0.5 and 0.5. If the weights were initialized such that they lie between 0 and 1, the neural network will never converge. This is because the input to all hidden neurons will be higher than the active input range of the sigmoid function, resulting in an output always 1.

Pseudo code for training method is given below:

```
Initialize weights and biases ()

while Total Error > Target Error

        for each element E in set X

                target output = {1 for neuron Z classifying E; 0 otherwise} feed forward through network ()

                calculate total error ()

                back propagate through net and adjust weights ()

loop until criteria is met.
```

After the network is trained on all elements in "X" it can be used for recognition. The number of output neurons depends on the number of characters to be classified. Since this neural network can classify inputs only if their quantity is fixed, the image must pass a pre-processing phase to allow such input uniformity.

Steps for back propagation algorithm are as follows:

Step-1: Initialization of weights. All weights were set into a few of small randomly selected values.

Step-2: Calculation of activation function for finding f(net). Determination of the Activation function of the input was by the samples offered to the network.

Determination of the Activation function Oj of Hidden and Output by following equation

$$O_j = f[\sum w_{ij} * O_i - \theta_j]$$

Where w_{ij}= input O_i to unit j weights, θ_j unit j node threshold, f The Activation Function.

Step-3: Training of weights.

Step-4: Updating weights. In this step, it starts at output then working in reverse to the hidden layer, and weight adjusting by following equation:

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}$$

Step-5: Computation of weight change:

Computing the weight change by following equation,

$$\Delta w_{ji} = \eta \delta_j O_j$$

Where η is the learning rate, δj is the error gradient/.

The error gradient can be computed at the Output Unit as follows

$$\delta_j = O_j(1 - O_j)(T_j - O_j)$$

And for the Hidden Unit

$$\delta_j = O_j(1 - O_j) \sum \delta_k w_{kj}$$

Where T is the Target Value, Oj is the Actual Output Value, δk is the Error Gradient k at connection point j.

Step-6: Repeating the iteration until converges.

IV. Results and Evaluation

Performance of the proposed system will be evaluated on the basis of training time and testing time on different samples. Training time obtained from implementation is as follows:

Table 1: Training Time Evaluation

Samples	Training Time(in ms) in Existing Method	Training Time(in ms) in Proposed Method
Sample1	18659	18522
Sample2	18112	17959
Sample3	17273	17118

Chart is shown below:

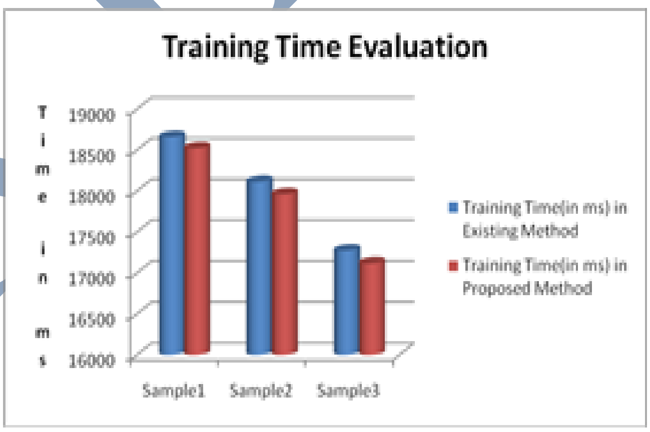

Figure 5: Chart Comparing existing and Proposed Methods.

Testing time obtained from implementation is as follows:

Table 2: Testing Time Evaluation

Sample	Testing Time (in ms) in Existing Method	Testing Time (in ms) in Proposed Method
Sample1	2653	1044
Sample2	1427	835
Sample3	666	615

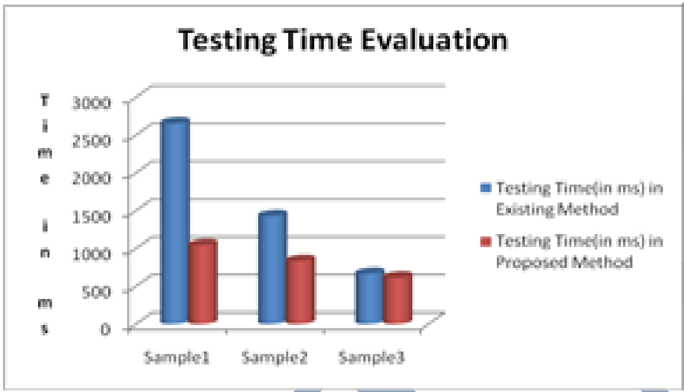

Figure 6: Chart Comparing both Methods.

Above tables and charts represents that our proposed method performs better in various samples in respect of training and testing time.

V. Conclusion and Future Work

5.1 Conclusion

License plate recognition plays an important role in intelligent transportation system and it has huge number of practical applications such as automatic toll collections, parking fee payment, detection of vehicle crossing speed limits and thereby reducing road accidents etc. Literatures have been extensively reviewed. Neural network has capability of learning, i.e. how to do tasks based on data given for training. Back propagation is supervised form of learning. In License plate recognition system, features extracted from license plate are used as input to neural networks and these are allowed to propagate forward to generate output. Learning of neural networks is done by loading targets and features extracted from license plate characters. Testing of neural network is done and parameters used for performance evaluation i.e. recognition rate and training time is computed. System simulated found that our proposed method performs better over existing one.

5.2 Future Work

Currently the Car Registration License Plate Detection and Recognition System support the basic functionality to detect license plates, recognize the characters and save the output in a file. This application can be extended in many ways. Some of them are listed below:

- Support to recognize the state: It will be nice if the application could recognize the state to which the vehicle belongs by seeing its license plate.

- There is lots of variety in the style of license plates. Their style varies from country to country. For example, at some places, dark characters are written on light plate; at other places, white characters on black background. Some motor departments use one-row plate and others use two-row plate. There is lots of variety in character sizes and fonts as well. So the project should be trained enough to work with various styles.

 The method should be able to automatically decide the threshold values for character extraction according to the quality of the picture.

Vehicle registration number recognition is beneficial in many areas such as:

- Traffic control in restricted area.
- Car parking management.
- Automatic toll barriers.
- Red light violation
- Access control
- Traffic monitoring
- Border crossing
- Security in military areas

REFERENCES

[1] S. D. Palmer and O. N. Aharoni, "System for collision prediction and traffic violation detection," U.S. Patent 20 130 093 895, Apr. 18, 2013.

[2] B. Hongliang and L. Changping, "A hybrid license plate extraction method based on edge statistics and morphology," in Proc. IEEE 17th ICPR, 2004, vol. 2, pp. 831–834.

[3] A. Mousa, "Canny edge-detection based vehicle plate recognition," Int. J. Signal Process., Image Process. Pattern Recognition, vol. 5, no. 3, pp. 1–8, 2012.

[4] W. Gao, X. Zhang, L. Yang, and H. Liu, "An improved Sobel edge detection," in Proc. IEEE 3rd ICCSIT, 2010, vol. 5, pp. 67–71.

[5] K. Deb, V. V. Gubarev, and K.-H. Jo, "Vehicle license plate detection algorithm based on color space and geometrical properties," in Emerging Intelligent Computing Technology and Applications. Berlin, Germany: Springer, 2009, pp. 555–564.

[6] K. Deb, H. Lim, S.-J. Kang and K.-H. Jo, "An efficient method of vehicle license plate detection based on HSI color model and histogram," in Next-Generation Applied Intelligence. Berlin, Germany: Springer,2009, pp. 66–75.

[7] Liorens, Marza, Palazon, and Vilar, Car License Plates Extraction and Recognition Based on Connected Components Analysis andHMMDecoding, vol. 3522. Berlin, Germany: Springer, 2005, pp. 571–578.

[8] Y.Wen,Y. Lu, J.Yan, Z. Zhou, K.M.von Deneen, andP. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," IEEE Trans. Intell. Transp. Syst., vol. 12, no. 3, pp. 830–845, Sep. 2011.

[9] V. Abolghasemi and A. Ahmadyfard, "An edge-based color-aided method for license plate detection," Image Vis. Comput., vol. 27, no. 8,pp. 1134–1142, Jul. 2009.

[10] A. Mousa, "Canny edge-detection based vehicle plate recognition," Int. J. Signal Process., Image Process. Pattern Recognit. vol. 5, no. 3, pp. 1–8, 2012.

[11] W. Gao, X. Zhang, L. Yang, and H. Liu, "An improved Sobel edge detection," in Proc. IEEE 3rd ICCSIT, 2010, vol. 5, pp. 67–71.

[12] P. Kanani, A. Gupta, D. Yadav, R. Bodade, and R. B. Pachori, "Vehicle license plate localization using wavelets," in Proc. IEEE ICT, 2013, pp. 1160–1164.

[13] K. Deb, V. V. Gubarev, and K.-H. Jo, "Vehicle license plate detection algorithm based on color space and geometrical properties," in Emerging Intelligent Computing Technology and Applications. Berlin, Germany: Springer, 2009, pp. 555–564.

[14] K. Deb, H. Lim, S.-J. Kang and K.-H. Jo, "An efficient method of vehicle license plate detection based on HSI color model and histogram," in Next-Generation Applied Intelligence. Berlin, Germany: Springer, 2009, pp. 66–75.

[15] R. Girshick, F. Iandola, T. Darrell, J. Malik, Deformable Part Models are Convolutional Neural Networks. arXiv preprint arXiv: 1409.5403, 2014. in CVPR, 2015.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in Proc. of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14), pp.580-587, 2014.

[17] Xiang H, Zhao Y, Yuan Y, Zhang G, Hu X,Lightweight Fully Convolutional Network for License Plate Detection, Optik (2018), https://doi.org/10.1016/j.ijleo. 2018.10.098.

[18] Vijeta Khare , Palaiahnakote Shivakumara , Chee Seng Chan , Tong Lu , Liang Kim Meng , Hon Hock Woon , Michael Blumenstein , A Novel Character Segmentation- Reconstruction Approach for License Plate Recognition, Expert Systems With Applications (2019), doi: https://doi.org/10.1016/j.eswa.2019.04.030.

[19] Yousri Kessentini, Mohamed Dhia Besbes, Sourour Ammar, Achraf Chabbouh," A two-stage deep neural network for multi-norm license plate detection and recognition", https://doi.org/10.1016/j.eswa.2019.06.036 0957-4174/© 2019 Elsevier.

[20] Muhammad Rizwan Asif a , b , ∗, Chun Qi a , ∗, Tiexiang Wang a , Muhammad Sadiq Fareed a , Syed Ali Raza , "License plate detection for multi-national vehicles: An illumination invariant approach in multi-lane environment",https://doi.org/10.1016/j.compeleceng.2019.07.0120045-7906/© 2019 Elsevier.

[21] Tomas Bjorklund, Attilio Fiandrotti, Mauro Annarumma, Gianluca Francini, Enrico Magli, Robust License Plate Recognition using Neural Networks Trained on Synthetic Images, Pattern Recognition (2019), doi: https://doi.org/10.1016/j.patcog.2019.04.007.