

A Design Flow for Robust License Plate Localization and Recognition in Complex Scenes

Dhawal Wazalwar, Erdal Oruklu, Jafar Saniie

Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, USA

Email: erdal@ece.iit.edu

Received September 22, 2011; revised October 24, 2011; accepted November 8, 2011

ABSTRACT

In this paper, we present a new design flow for robust license plate localization and recognition. The algorithm consists of three stages: 1) license plate localization; 2) character segmentation; and 3) feature extraction and character recognition. The algorithm uses Mexican hat operator for edge detection and Euler number of a binary image for identifying the license plate region. A pre-processing step using median filter and contrast enhancement is employed to improve the character segmentation performance in case of low resolution and blur images. A unique feature vector comprised of region properties, projection data and reflection symmetry coefficient has been proposed. Back propagation artificial neural network classifier has been used to train and test the neural network based on the extracted feature. A thorough testing of algorithm is performed on a database with varying test cases in terms of illumination and different plate conditions. Practical considerations like existence of another text block in an image, presence of dirt or shadow on or near license plate region, license plate with rows of characters and sensitivity to license plate dimensions have been addressed. The results are encouraging with success rate of 98.10% for license plate localization and 97.05% for character recognition.

Keywords: License Plate Localization; Character Recognition; Reflection Symmetry Coefficient; Artificial Neural Network

1. Introduction

License plate recognition is considered to be one of the fastest growing technologies in the field of surveillance and control. Apart from its increasing use in areas like road traffic management and speed checks, it is also becoming an important tool for police officials in solving traditional criminal investigations. With such an extensive use of this technology, efforts are constantly being made to make it more efficient. The technology research can be classified into two categories, namely fixed camera and mobile camera license plate recognition. Each of these categories has its own specific requirements and technical challenges. Our main focus in this paper is on fixed camera applications.

License plate localization is very crucial step in that the overall system accuracy depends on how accurately we can detect the exact license plate location. The input can be in the form of still images or video frames from surveillance cameras. The processing can be done at either color or grayscale level. One such method based on using unique color combination of license plate region was proposed in [1]. However, the large data that needs to be processed in case of color image processing makes it unsuitable for real time applications. On the other

hand, the amount of data that needs to be handled in case of gray scale technique is comparatively less and thus facilitates faster and more efficient implementations. A method based on using texture information of license plate region as a feature [2] was proposed with an accuracy of 98.5%. However failures for cases with existence of another text block, dirty license plate region in the input frame were documented [2]. In order to make the texture information of license plate clearer, a technique based on obtaining horizontal image difference was presented in [3]. This method however is sensitive to the license plate dimensions and is not robust enough to handle all practical conditions. A real time solution with intelligent frame selection from input video was presented in [4]. It then enhances the obtained frame using Gaussian filter and histogram equalization. Morphological operations are then done along with connected component analysis to extract the license plate region. Morphological operations, specifically dilation and erosion, were used in [5] to detect license plate region. This technique in case of complex images may require an initial system to get the part of image having vehicle separated from the rest of the image. This can be particularly difficult in cases where complete vehicle is not visible or where license plate exists at corner parts in an image.

The main objective in character segmentation is to obtain clean and isolated characters. Vertical projection information can be used to separate the characters [6]. However, this technique requires prior knowledge of license plate size as well as the number of characters in it. This sensitivity to license plate dimensions was reduced to an extent by employing region growing segmentation method along with pixel connectivity information [7]. In order to tackle tilted and distorted license plates, image tilt correction and gray level enhancement was proposed [8]. Binarization of a gray level image is a key aspect in character segmentation and researchers have been using both local as well as global thresholding [7] as a means of achieving it. Hybrid binarization technique using histogram information [9] improved the overall performance especially in cases with presence of dirt or improper shadows on the license plate.

The accuracy of a character recognition algorithm depends on uniqueness of feature vector information. Horizontal and vertical projection data, zoning density and contour features were combined to form a feature vector [10] with an accuracy of 95.7%. Recently, a method based on the concept of stroke direction and elastic mesh [8] was proposed with an accuracy in the range of 97.8-99.5%. Another important aspect of character recognition

step is the type of classifier employed. A two stage classifier based on Euclidean distance calculation is defined in [10]. Apart from various types of probability models, classifiers based on artificial neural network (ANN) [11] and support vector machine (SVM) [4,8] are widely used in the field of optical character recognition.

In this paper, we present a new design flow for a robust license plate localization and recognition system, shown in **Figure 1**. The system can be categorized into three key stages 1) license plate localization; 2) character segmentation; and 3) character recognition. In Section 2, the main objective was to come up with an optimal license plate localization algorithm which will address to some of the shortcomings in previous implementations like 1) presence of another text block in an image [2]; 2) existence of partial contrast between the license plate region and surrounding area; 3) license plate region located anywhere in the frame. In Section 3, the emphasis is on obtaining clean and isolated characters. Practical issues like presence of shadow or dirt on license plate, blur images have been highlighted. In Section 4, main focus is on obtaining unique feature vector as well as on training the back propagation artificial neural network classifier. Finally, in order to have common benchmarking process [12,13] and to determine accuracy of license

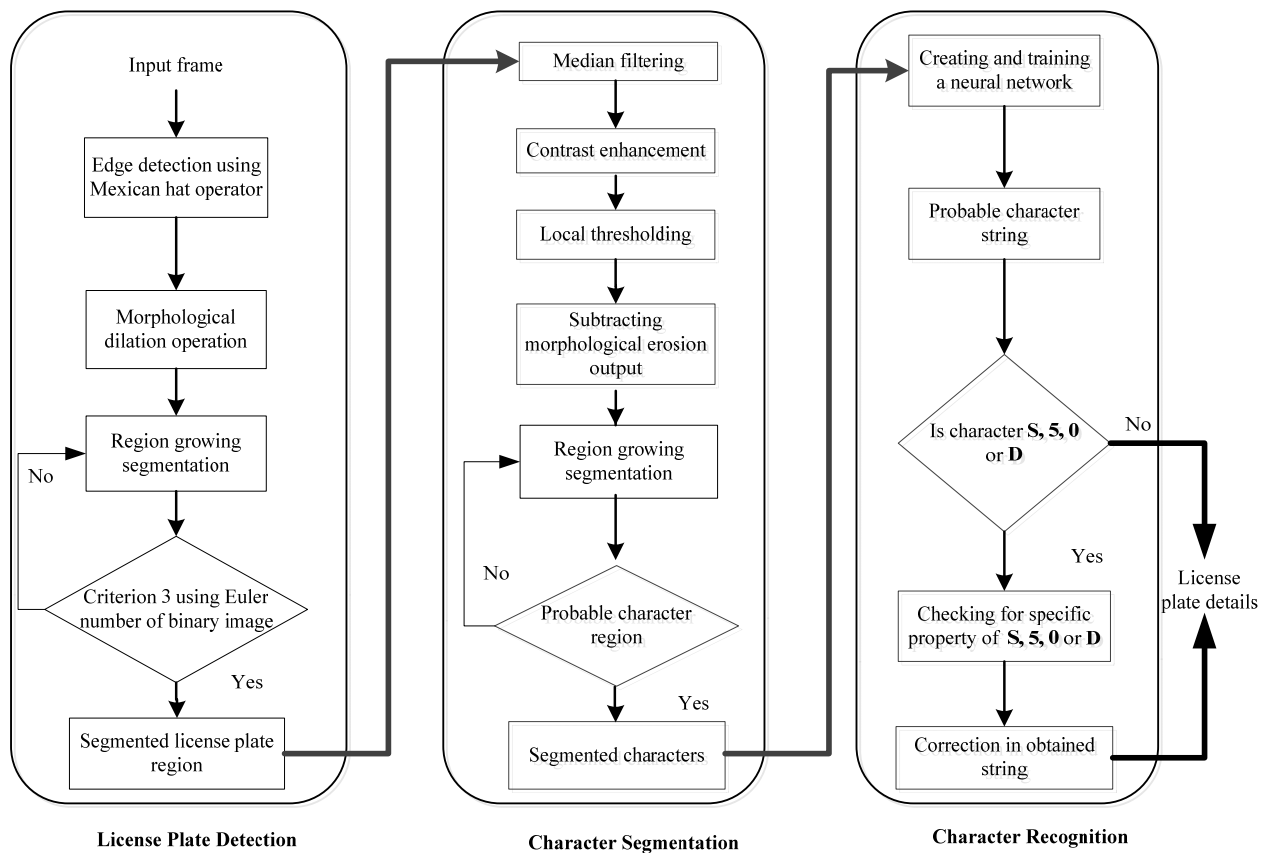


Figure 1. Design flow for robust license plate localization and recognition system.

plate detection step, we have tested our algorithm on different types of practical images. A detailed summary of observed results has been presented in Section 5. The images are taken from a commercial City Sync's automatic number plate recognition camera video [14].

2. License Plate Localization

A detailed flow diagram highlighting the key steps involved in the license plate recognition system is shown in **Figure 1**. The main steps involved in license plate localization are edge detection, morphological dilation operation and region growing segmentation.

2.1. Selection of Edge Detection Technique

In order to avoid any possibility of false license plate detection, it is important to obtain clean and continuous edges of license plate region. Ideally, these edges should not be connected to any other surrounding parts. General gradient operators like Prewitt, Sobel operators are useful in cases when there is a clear and distinct contrast between the license plate and region surrounding it, as seen in **Figure 2**.

However, in cases when there is a partial contrast, these general gradient operators fail to give the desired output. For example, in **Figure 3(b)** it can be seen that the license plate region is connected to the surrounding region. In order to avoid such cases, more advanced operators like Mexican hat operator have to be used, see **Figure 3(c)**. Mexican hat operator first performs smoothing operation and then extracts edges. This function is also called as Laplacian of Gaussian (LoG) and mathematically can be expressed [15] as

$$\nabla^2 h(r) = - \left[\frac{r^2 - \sigma^2}{\sigma^4} \right] e^{-\frac{r^2}{2\sigma^2}} \quad (1)$$

where $r^2 = x^2 + y^2$ and σ is the standard deviation.

Masks of different sizes 3×3 , 5×5 and 9×9 were tested and the best results were seen for 9×9 operator. Following is the 9×9 operator used in our implementation.

$$\begin{array}{cccccccc} 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & -3 & -3 & -3 & -1 & -1 & 0 \\ 0 & -1 & -3 & -3 & -1 & -3 & -3 & -1 & 0 \\ -1 & -3 & -3 & 6 & 13 & 6 & -3 & -3 & -1 \\ -1 & -3 & -1 & 13 & 24 & 13 & -1 & -3 & -1 \\ -1 & -3 & -3 & 6 & 13 & 6 & -3 & -3 & -1 \\ 0 & -1 & -3 & -3 & -1 & -3 & -3 & -1 & 0 \\ 0 & -1 & -1 & -3 & -3 & -3 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \end{array}$$

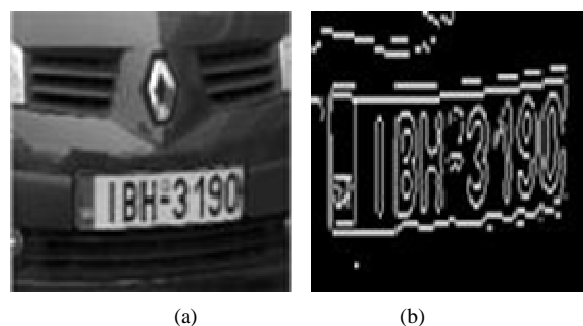


Figure 2. (a) Sample image 1; (b) Edge detection using Prewitt operator with license plate edges clearly separated from surrounding region.

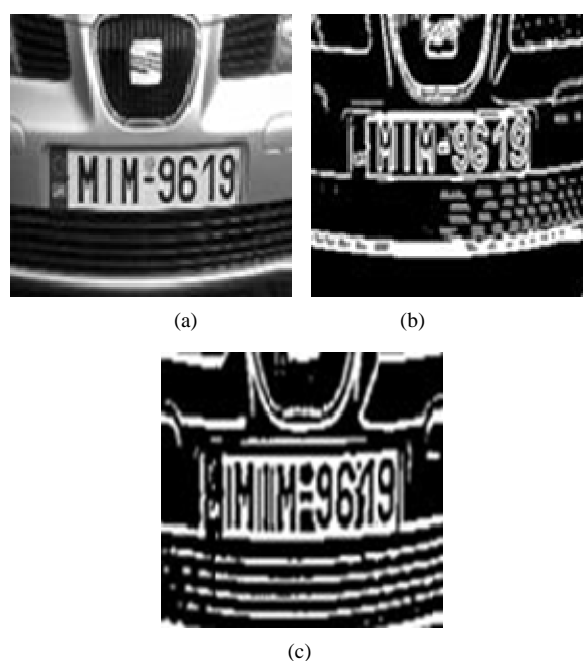


Figure 3. (a) Sample image 2; (b) Failure case of Prewitt operator with license plate edges connected to surrounding region; (c) Edge detection output using Mexican hat operator with clear and distinct edges.

2.2. Morphological Dilation Operation

If the license plate image is blurred, edge detection output can have discontinuity as seen in **Figure 4(b)**. This can result in failure cases or incorrect detection output, (see **Figure 4(c)**). In order to prevent such cases and make these edges thick and continuous similar to **Figure 4(d)**, a dilation operation is performed. The improved detection performance can be seen **Figure 4(e)**.

Dilation operation can be mathematically expressed [15] as

$$X \oplus B \triangleq \{x : B_x \cap X \neq \emptyset\} \quad (2)$$

where X is the object and B is the structuring element.

The size and nature of structuring element is important since smaller size can negate the desired effect and larger

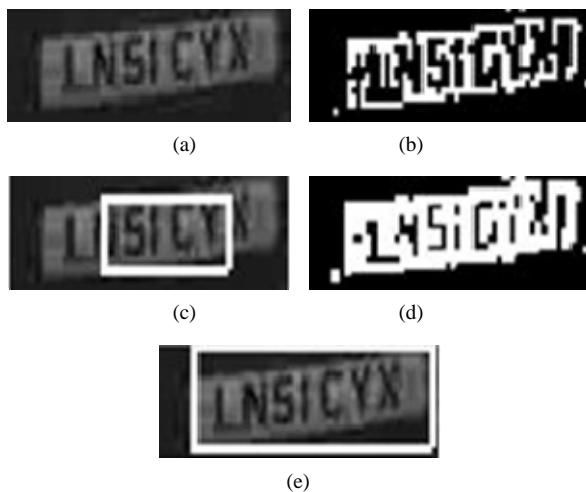


Figure 4. (a) Blur image (b) Edge detection output (c) Plate detection output without dilation step (d) Dilation output (e) Modified plate detection output with dilation step.

size can cause the license plate regions to be connected to the surrounding area. For our algorithm, best results were achieved with 3×3 ones matrix as a structuring element.

2.3. Region Growing Segmentation

Region growing is a procedure in which we group pixels based on some predefined pixel connectivity information to form sub regions. The performance of region growing algorithm depends on the starting points and the stopping rule selected for the implementation [15]. In our case, region growing process was performed on whole image so as to segment the entire image into different sub parts. For every bright pixel, its neighboring four pixel connectivity information was checked and region was grown based on this information. In order to increase the algorithm speed, sub-sampling was used, where algorithm operation was performed for every two pixels, instead of performing for each and every pixel in the image. The final output will be several sub regions labeled based on pixel connectivity information, see **Figure 5**.

2.4. Detecting the License Plate Region

After segmenting the entire image in smaller regions, following criteria can be used to identify license plate region from it.

Criterion 1: License plate dimensions

Generally license plate dimensions are fixed for a particular state/country and so it can be used to determine the license plate region. However, by doing this we make algorithm sensitive to license plate dimensions, which depend on the distance between the vehicle and camera.

Criterion 2: Euler number of a binary image

Characters used in license plate region are alphanu-

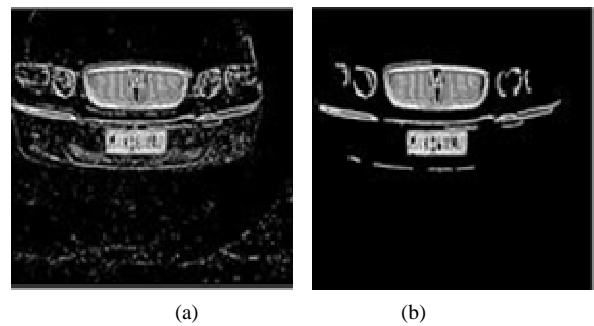


Figure 5. (a) Output after dilation step; (b) Region growing segmentation output.

meric (A-Z and “0-9”). If we carefully observe these characters in an image, this can be seen as closed curves like in 0, 9, 8, P etc. Thus clearly license plate region will have more closed curves than in any other part of an image. Euler number of an image can be used to distinguish the different regions obtained after the region growing output [16]. Euler number of a binary image gives the number of objects minus the closed curves (holes). Experimentally, it was found that in alphanumeric regions, value of Euler number is zero or negative, while for other regions it has a positive value. Using Euler number criterion will overcome the sensitivity issue encountered in using Criterion 1. However, if the vehicle image has alphanumeric characters in more than one region algorithm may fail; see **Figure 6(a)**.

Criterion 3: Combination of Criterion 1 and Criterion 2

In this implementation, we have used a combined approach of Euler number criterion as well as license plate dimensions to overcome problem raised in case of Criterion 2. Instead of predefining exact license plate dimensions, we specify a range of acceptable minimum and maximum license plate sizes. This addresses the sensitivity issue raised in Criterion 1. The modified result can be seen in **Figure 6(b)**.

3. Character Segmentation

After locating the license plate, the next important step is to segment each character individually, avoiding any possibility of joint characters. **Figure 1** shows all the necessary steps that need to be performed in order to ensure accurate character segmentation.

3.1. Image Preprocessing

3.1.1. Median Filtering

Median filter is a non-linear spatial filter and is used widely for image denoising [17,18]. In median filtering, we replace the original pixel data by median of pixels contained in a prescribed window. In order to ensure that the edges of the characters are preserved, analysis was done to determine the optimal median filter size. In this case, best results were obtained for 3×3 filter window.



Figure 6. (a) Failure case for Criterion 2; (b) Criterion 3 output for part (a) case.

3.1.2. Contrast Enhancement

After observing the pixel data in the license plate region, it was found that the character information for most of the images in database is usually in the intensity range of 0 - 0.4 on a scale of 1. In order to obtain further contrast enhancement, we increased the dynamic range for pixels in this (0 - 0.4) intensity range.

Figure 7 shows application of all the above steps on one such sample case. In Figure 7(b), we can see the enhanced version of original license plate image.

3.2. Threshold Operation

For further processing, we need to convert grayscale image into a binary image by using a threshold operation. There are basically two types of threshold operation:

3.2.1. Global Thresholding

In this case, we select one single threshold value for entire image. Mathematically it can be represented as

$$g(x, y) = \begin{cases} 1, & f(x, y) \geq T \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $f(x, y)$ is the original grayscale image, $g(x, y)$ is the threshold output image and T is the threshold value obtained.

This operation is easy to implement and also less time consuming. However, it is sensitive to illumination conditions and fails in cases if there is a brighter or darker region other than object. One such case is shown in Figure 8(b), where we can see character "O" and "2" are connected because of unwanted dark region in original grayscale image, see Figure 8(a).

3.2.2. Local Thresholding

In the local threshold operation, we divide the grayscale image into several parts by using a windowing operation and then perform the threshold operation separately for each case. Since, we perform threshold operation for each region separately; a single brighter or darker region cannot corrupt the threshold value. Local thresholding

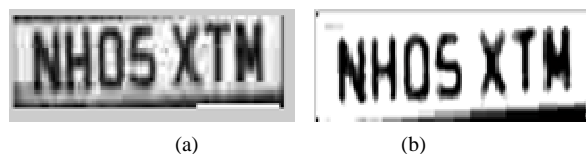


Figure 7. (a) Input license plate image before preprocessing; (b) Enhanced image after preprocessing.

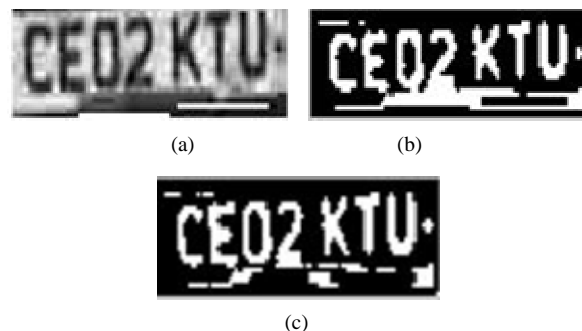


Figure 8. (a) License plate image with unwanted dark region; (b) Global thresholding output, with joint characters; (c) Local thresholding output with separated characters.

ensures that in most cases, we get clear and separated characters. In Figure 8(c), we can see all the characters are separated compared to Figure 8(b), where global thresholding was performed. The performance of local threshold operation will vary depending on the selected window size. Experimentally, best results were obtained when the window size was chosen approximately equal to the general individual character size in the license plate image. For our algorithm, the threshold value for local region is obtained using Otsu's method for gray level histogram [19].

3.3. Morphological Erosion Operation

In certain situations if some dirt or shadow is present on the license plate region, the output of local threshold operation may not be sufficient. For example in Figure 9(b) it can be seen even after the local threshold operation some amount of unwanted noise is still present near character "4" and "X". This noise can be removed by performing morphological erosion operation (see Figure 9(c)). Mathematically, erosion operation can be represented [15] as

$$X \ominus B \triangleq \{x \mid (B)_x \subseteq X\} \quad (4)$$

where X is the object and B is the structuring element. The size and nature of structuring element should be selected with an aim of keeping the character information intact and only removing unwanted parts in the binary image.

3.4. Region Growing Segmentation

The pixel connectivity information can be used to separate

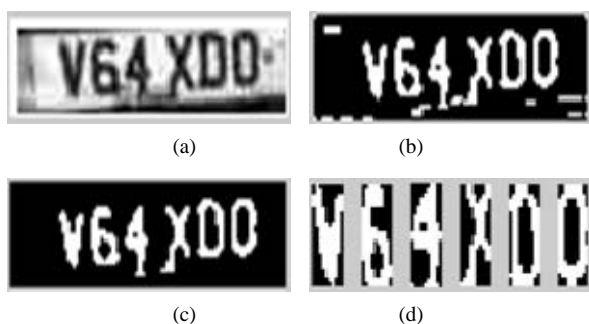


Figure 9. (a) License plate with shadow on characters 4 and X; (b) Output after local threshold operation; (c) Output after morphological erosion operation; (d) Separated characters after region growing segmentation.

the characters. Region growing segmentation along with four pixel connectivity approach is used in this case. **Figure 9(d)** shows the separated characters after performing region growing segmentation.

4. Character Recognition

After segmenting the characters in license plate, the next important step is to extract feature information from the character and recognize it.

4.1. Feature Extraction

The robustness of character recognition algorithm depends on how well it can recognize different versions of a single character. In **Figure 10**, all the images are for same character G of size 24×10 , however it varies in terms of spatial information, noise content and thickness of edges. The feature vector should be selected in such a way that algorithm can still accurately recognize all these varied representations of a single character.

In addition, sometimes it is difficult to distinguish between characters like D and 0 or 5 and S because of the similarity in their appearance as well as spatial information. In order to deal with these challenges, our algorithm uses a combination of spatial information as well region properties. Following is the description of some of these key features.

Feature 1: This feature comprises of region properties like area, perimeter, minor-major axis length, orientation and Euler number of binary image. Orientation is the angle between the x-axis and the major axis of the ellipse that has same secondary moments as that of region. An analysis of variation of all region properties with respect to the characters was performed and these six properties were selected from them.

Feature 2: The second part of feature vector includes projection information about X and Y axis. This forms major part of feature vector and is unique for each character.



Figure 10. Different versions of character “G”.

Feature 3: Some of the characters like 0, M, N are symmetric about X-Y axes. This information can be used to distinguish different characters especially 0 and D. This symmetry information can be quantified by a coefficient called as reflection symmetry coefficient. For 2D images, a way of calculating this coefficient is proposed in [20]. In this implementation for a character of size 24×10 , we calculate measure of symmetry about X and Y axis in following way

1) Consider C_{in} as the original character matrix where information is stored in binary format. The center of matrix is considered as the origin. C_x and C_y are the transformed matrix obtained after flipping about the X and Y axes respectively.

2) Calculate the area of original matrix C_{in} .

3) Determine the union of transformed matrixes C_x and C_y with original matrix C_{in} .

4) Calculate the areas of regions $(C_{in}C_x)$ and $(C_{in}C_y)$.

5) The reflection coefficient can be determined as

$$X_{\text{coefficient}} = \frac{\text{Area}(C_{in})}{\text{Area}(C_{in} \cup C_x)} \quad (5)$$

$$Y_{\text{coefficient}} = \frac{\text{Area}(C_{in})}{\text{Area}(C_{in} \cup C_y)}$$

Feature 3 has significantly improved algorithm's performance in dealing with complex cases like D and 0. Experimentally it was found that for 0, the value of X coefficient and Y coefficient are approximately similar. On the other hand, since D has symmetry only along X axis, value of X coefficient is larger compared to Y coefficient.

4.2. Artificial Neural Network Design

Artificial neural networks (ANN) are considered a powerful tool for solving complex engineering problems like pattern classification, clustering/categorization, prediction/forecasting etc. They are in general classified into two categories namely feed-forward networks and recurrent (or feedback) networks. In this implementation, we are using a feed-forward based neural network design.

A simplest form of feed-forward neuron model is called as “Perceptron”. In this model, (x_1, x_2, \dots, x_n) are the input vectors, (w_1, w_2, \dots, w_n) are the weights associated with the input vectors, h is the summation of all inputs with its weights and y is the output. The selection of ac-

tivation function is decided by the complexity of proposed problem. For our design, we have used a log sigmoid activation function. The basic algorithm for this model can be explained [21] as follows.

1) Initialize the associated weights and the activation function for the model.

2) Evaluate the output response for given input pattern $(x_1, x_2, \dots, x_n)^t$.

3) Update the weights based on following rule

$$w_j(t+1) = w_j(t) + \eta(d - y)x_j \quad (6)$$

In case of back-propagation model, the error at the output is propagated backwards and accordingly the weights for inputs are adjusted. This concept can be extended to multiple layer model based upon the requirements.

Following is the description of some of these key design parameters in reference to character recognition problem.

4.2.1. Input Layers

Input layers to the neural network are the feature vectors extracted for the all the characters. The overall size is decided by the character set and the feature vector size.

4.2.2. Output Layers

Output nodes are the set of all possible characters that are present in license plate information. The number of output nodes depends upon the numbers of characters that need to be classified. Typically, it consists of 26 alphabets "A - Z" and 10 numbers "0 - 9". However, to reduce algorithm complexity, certain countries avoid using both 0 and O together. In our test database, we have all the characters except O, 1 and Q. Therefore, the number of output layers is 33 in our implementation.

4.2.3. Hidden Layers

Hidden layers are the intermediate layers between the input and output layers. There is no generic rule as such for deciding the number of hidden layers. In our design, the number of hidden layers was experimentally found to be 25.

For our implementation, we have used a supervised learning paradigm. We obtained a set of 50 images for each character from the database images and trained the neural network using them. For better results, it is important to include images of all types of character representation (ideal, noisy, see **Figure 10**) in the training dataset. We have used MATLAB for training the neural network.

It was observed that for certain characters like 0-D and 5-S, error rate was higher compared to other characters, due to similar spatial information content. In order to improve the overall accuracy, a two stage detection process for these characters is used. If the characters are

possibly between 5, 0, S and D, then region properties like orientation, reflection coefficient were again used in the 2nd stage identification process and the final possible character was identified. This two stage process has improved the overall algorithm accuracy by 1% - 2%.

5. Results

Performance evaluation for the license plate recognition algorithm is a challenge in itself, since there is no common reference point for benchmarking [12]. We tested our algorithm on 1500 different frames obtained from a sample video, taken from commercial license plate recognition camera [14]. The resolution of all frames is 480×640 . All these frames were then classified into following types:

Type 1: In these samples, license plate is fully visible and all its characters are clear and distinct, see **Figure 11(a)**.

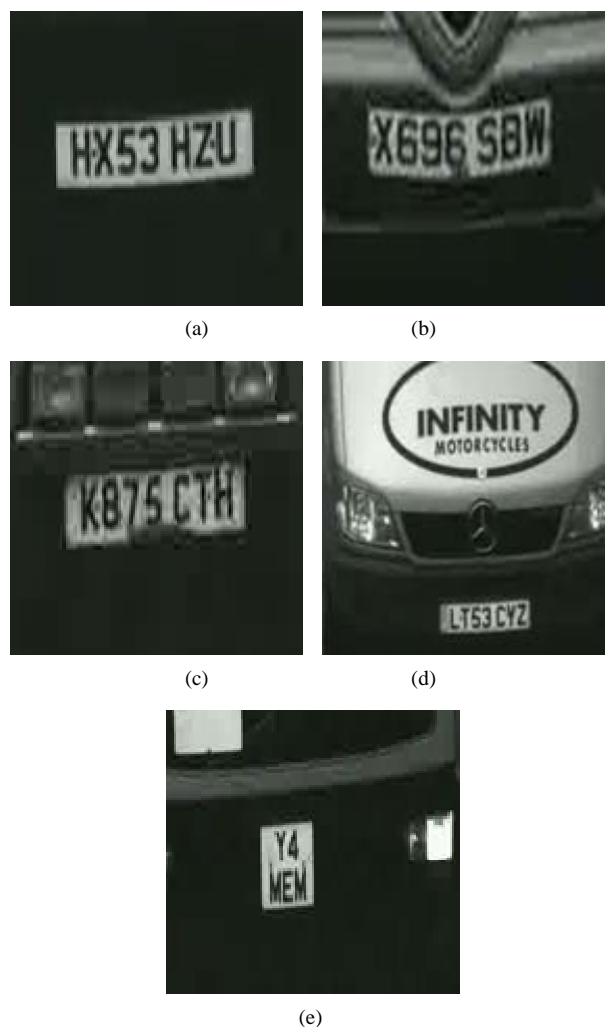


Figure 11. Different types of input frames (a) Type1 clear and distinct; (b) Type 2 image with blurred license plate details; (c) Type 3 license plate with dirt on it; (d) Type 4 presence of another text block in image; (e) Type 5 license plate details in two rows.

Type 2: Images in these samples are little blurred due to variations in illumination conditions; see **Figure 11(b)**.

Type 3: These samples have little dirt or shadows on or near license plate region, **Figure 11(c)**.

Type 4: In these samples, another text block apart from license plate is present in the frame, **Figure 11(d)**.

Type 5: License plate details are present in two rows, **Figure 11(e)**.

Figure 11 shows sample images for all the above image types. **Table 1** summarizes the license plate recognition results for all these image types.

A rare failure case is seen if the license plate is black in color. In this case, the edge detection step fails since it cannot locate license plate edges. One such case can be seen in **Figure 12**.

6. Conclusions

The algorithm presented in this paper is extremely useful in dealing with complex cases, while building a real-time license plate recognition system. The use of Mexican hat operator helps to improve the performance of edge detection step, when there is only partial contrast between license plate region and region surrounding it. Euler number criterion for a binary image helps to reduce the sensitivity of algorithm to license plate dimensions. Pre-processing step using median filter and contrast enha-

ancement ensures performance in case of low resolution and blurred images. Local threshold operation prevents a single brighter or darker region from corrupting the Threshold value and thereby improves binarization process. Reflection symmetry coefficient along with two stage identification process improves the character recognition performance significantly especially in dealing with complex cases like recognizing 0 and D.

The algorithm can be extended to mobile license plate recognition systems because of its ability to provide exact license plate location in complex cases. Also, to increase the intelligence of license plate detection algorithm while working with video input, motion analysis can be applied to the sequential frames and selection of proper frame can be done. The performance of algorithm will also improve if higher resolution input frames are used.

REFERENCES

- [1] X. F. Shi, W. Z. Zhao and Y. H. Shen, "Automatic License Plate Recognition System Based on Color Image Processing," *International Conference on Computational Science and Its Applications (ICCSA)*, Vol. 3483, 2005, pp. 1159-1168.
- [2] Z. Chen, C. Liu, F. Chang and J. Xu, "A Novel Algorithm of License Plates Automatic Location Based on Texture Feature," *IEEE International Conference on Automation and Logistics*, Shenyang, 5-7 August 2009, pp. 1360-1363. [doi:10.1109/ICAL.2009.5262747](https://doi.org/10.1109/ICAL.2009.5262747)
- [3] W. Pan and N. Yang, "A New Method of Vehicle License Plate Location under Complex Scenes," *2nd International Conference on Advanced Computer Control (ICACC)*, Shenyang, 27-29 March 2010, pp. 134-138. [doi:10.1109/ICACC.2010.5486754](https://doi.org/10.1109/ICACC.2010.5486754)
- [4] A. Delforouzi and M. Pooyan, "Efficient Farsi License Plate Recognition," *7th International Conference on Information, Communication and Signal Processing*, Macau, 8-10 December 2010, pp. 1-5.
- [5] C. N. K. Babu and K. Nallaperumal, "A License Plate Localization Using Morphology and Recognition," *IEEE Indian Conference (INDCON)*, Kanpur, 11-13 December 2008, pp. 34-39. [doi:10.1109/INDCON.2008.4768797](https://doi.org/10.1109/INDCON.2008.4768797)
- [6] B. M. Shan, "License Plate Character Segmentation and Recognition Based on RBF Neural Network," *Second International Workshop on Education Technology and Computer Science (ETCS)*, Wuhan, 6-7 March 2010, pp. 86-89. [doi:10.1109/ETCS.2010.464](https://doi.org/10.1109/ETCS.2010.464)
- [7] H. Lan and G. Shrestha, "Real Time License Plate Recognition at the Time of Red Light Violation," *International Conference on Computer Application and System Modeling (ICCASM)*, Taiyuan, 22-24 October 2010, pp. 243-247.
- [8] Y. Wen, Y. Lu, J. Q. Yan, Z. Y. Zhou, K. M. von Deneen, and P. F. Shi, "An Algorithm for License Plate Recognition Applied to Intelligent Transportation System," *IEEE Transactions on Intelligent Systems*, Vol. 12, No. 3, 2011, pp. 1-16.

Table 1. Results based on types of input images.

Image Type	Total images tested	Plate detection accuracy	Character recognition accuracy
Type 1	879	99.4%	97.81%
Type 2	442	96.3%	96.63%
Type 3	153	94.77%	95.69%
Type 4	44	100%	99.25%
Type 5	11	100%	97%
Overall Results	1529	98.1%	97.05%*

*In the database of 335 vehicle images, each vehicle on an average has 5 frames. License of 330 vehicles were recognized with an accuracy of 98.5%.

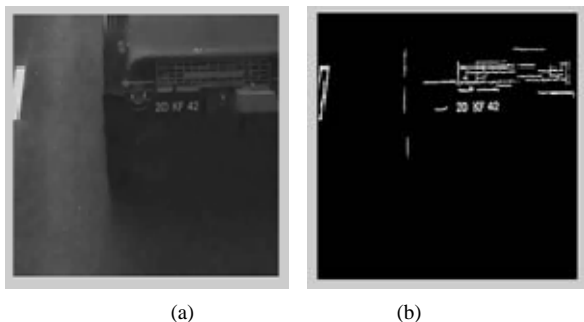


Figure 12. (a) Sample image with license plate black in color; (b) Failure of edge detection step in locating license plate edges.

- [9] J.-M. Guo and Y.-F. Liu, "License Plate Localization and Character Segmentation with Feedback Self-Learning and Hybrid Binarization Techniques," *IEEE Transactions on Vehicular Technology*, Vol. 57, No.3, 2008, pp. 1417-1424. [doi:10.1109/TVT.2007.909284](https://doi.org/10.1109/TVT.2007.909284)
- [10] Z.-X. Chen, C.-Y. Liu, F.-L. Chang and G.-Y. Wang, "Automatic License-Plate Location and Recognition Based on Feature Saliency," *IEEE Transactions on Vehicular Technology*, Vol. 58, No.7, 2009, pp. 3781-3785. [doi:10.1109/TVT.2009.2013139](https://doi.org/10.1109/TVT.2009.2013139)
- [11] S.-L. Chang, L.-S. Chen, Y.-C. Chung and S.-W. Chen, "Automatic License Plate Recognition," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 5, No. 1, 2004, pp. 42-53. [doi:10.1109/TITS.2004.825086](https://doi.org/10.1109/TITS.2004.825086)
- [12] C. Nikolaos, I. Anagnostopoulos, D. Psoroulas, V. Loumos and E. Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 3, 2006, pp. 377-392. [doi:10.1109/TITS.2006.880641](https://doi.org/10.1109/TITS.2006.880641)
- [13] L. Keilthy, "ANPR System Performance," *Parking Trend International*, June 2008.
- [14] "CitySync Single Lane PAL Demo Video," 2011. <http://www.citysync.co.uk/>
- [15] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," 3rd Edition, Prentice Hall, Saddle River, 2008.
- [16] X. Lin, J. Ji and Y. Gu, "The Euler Number Study of Image and Its Applications," *2nd IEEE Conference on Industrial Electronics and Its Applications*, Harbin, 23-25 May 2007, pp. 910-912.
- [17] X. Wang, "Multiscale Median Filter for Image Denoising," *IEEE 10th International Conference on Signal Processing*, Beijing, 24-28 December 2010, pp. 2617-2620.
- [18] C.-Y. Ning, S.-F. Liu and M. Qu "Research on Removing Noise in Medical Image Based on Median Filter Method," *IEEE International Symposium on IT in Medicine and Education*, Jinan, 14-16 August 2009, pp. 384-388.
- [19] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66. [doi:10.1109/TSMC.1979.4310076](https://doi.org/10.1109/TSMC.1979.4310076)
- [20] Z. M. Li, K. P. Hou and H. Li, "Continuous Symmetry Measure of Image," *IEEE International Conference on Information Acquisition*, Shandong, 20-23 August 2006, pp. 824-828. [doi:10.1109/ICIA.2006.305838](https://doi.org/10.1109/ICIA.2006.305838)
- [21] A. K. Jain, J. C. Mao and K. M. Mohiuddin, "Artificial Neural Networks: A Tutorial," *IEEE Computer Society*, Vol. 29, No. 3, 1996, pp. 31-44. [doi:10.1109/2.485891](https://doi.org/10.1109/2.485891)