**License Number Plate Recognition**

**Automatic Using OpenCV**

**Anil Dubey**

*(Sr. Asst. Professor , Dept of CSE , ABES Engineering College Ghaziabad , U.P , India)*

[*anil.dubey@abes.ac.in*](mailto:anil.dubey@abes.ac.in) *, +91-9783934206*

**Akshay Kr. Singh**

*(B.Tech Final Year , Dept of CSE , ABES Engineering College Ghaziabad , U.P , India)*

[*akshay.17bcs1075@abes.ac.in*](mailto:akshay.17bcs1075@abes.ac.in) *, +91-8299668246*

**Aakash Kumar**

*(B.Tech Final Year , Dept of CSE , ABES Engineering College Ghaziabad , U.P , India)*

[*aakash.17bcs1104@abes.ac.in*](mailto:aakash.17bcs1104@abes.ac.in) *, +91-7017723505*

**Abhishek Mishra**

*(B.Tech Final Year , Dept of CSE , ABES Engineering College Ghaziabad , U.P , India)*

[*abhishek.17bcs1202@abes.ac.in*](mailto:abhishek.17bcs1202@abes.ac.in) *, +91-7080657208*

**Alamgeer**

*(B.Tech Final Year , Dept of CSE , ABES Engineering College Ghaziabad , U.P , India)*

[*alamgeer.18bcs3009@abes.ac.in*](mailto:alamgeer.18bcs3009@abes.ac.in) *, +91-7310739163*

**ABSTRACT**

This paper suggests the LPRNet method - the end-to-end version of Automatic plate recognition without startup character classification. Our approach is inspired by recent Deep Neural Networks, and works on Real time accurate with up to **95%** Chinese license plates: **3 ms / plate** on nVIDIA R GeForceTMGTX 1080 and **1.3 ms / plate** on Intel R CoreTMi7-6700K CPU. LPRNet contains the Conweight Light Convolutional Neural Network, so it can be trained at the end and end. Ku As far as we know, LPRNet is the first real-time license Plate Recognition system that does not use RNNs. As a result, the LPRNet algorithm can be used to create embedding LPR solutions that demonstrate high-level accuracy or challenge Chinese license plates.

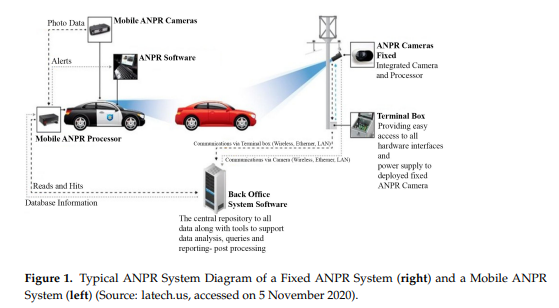
**Keywords**

Automatic Number Recognition (ANPR), Automatic Neural Network (ANN), image separation of characters Separation, Plate Number, Appearance Character Recognition

**1. INRODUCTION**

**1.1 Automatic Number Plate Recognition(ANPR)**

A few years ago, ANPR or license plate (LPR) recognition it has been one of the most useful ways of looking at cars. It can be used in many public places for fulfillment some of the objectives such as road safety, automatic collection of documents [1], car parking system [2] and Automatic parking system [3]. ANPR algorithms is usually divided into four steps: (1) Photographing a car (2) Acquisition of plate number (3) character separation and (4) Character recognition. As shown in Figure 1, the first step i.e. taking a picture of a car looks very easy but real exigent task as it is very difficult to take a photo of the move car in real time in such a way that none of the part of a car especially a plate number plate you should miss. Currently the discovery of the number plate once recognition processing time is less than 50 ms [4] in most cases programs. The success of the fourth step depends on the success of the second and in the third step they are able to find the number of vehicles as well separate each letter. These programs follow differently ways to find the number plate of a car in a car and extract the car number from that image.

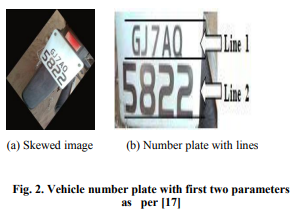


**1.2 Scope of this Paper**

Since it is impossible to judge which way is best, separate papers, based on the steps mentioned in Figure 1, are tested and classified based on each way. In each case where it is found parameters such as speed, accuracy, performance, image size and the platform is reported. Commercial product research is more than that the size of the paper as sometimes these products range more accurate than realistic for promotional purposes. The remainder of this paper is divided as follows: Section 2 contains a survey of various ways to find a number plate. Separation methods are reviewed in sections 3 and section 4 contains discussions about character recognition methods. This paper concludes with a discussion of what it is not done and what kind of research is possible in ANPR.

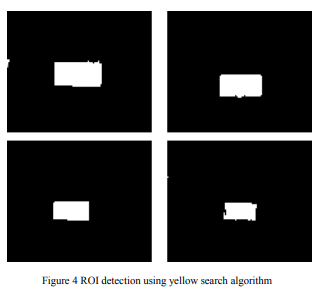
**2. NUMBER PLATE DETECTION**

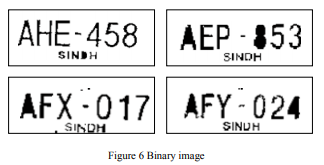
The system starts working when the sensor detects the file the availability of a car at the door. Small control post signal to PC for photography using a USB camera attached to PC. PC starts with ANPR algorithm and indicates vehicle authorization. ANPR the algorithm is tested for a large number of images with 800 x 600 pixel adjustment. The results show that developed ANPR algorithm successfully detects Sindh plates of normal car numbers in various situations of the day and shows a high level of acquisition and recognition. It can find and monitor car plates from various locations. The distance affects the size of the plate number in picture. Once the vehicle number plate has been obtained, individual letters are detected using OCR algorithm. OCR uses the integration method for character recognition and visual opportunities can also be counted. The program is computerized inexpensive and can be used in real time vehicle identification system.

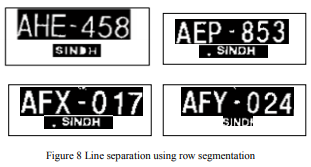
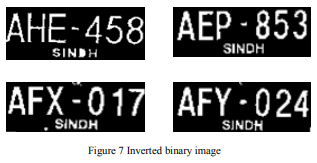


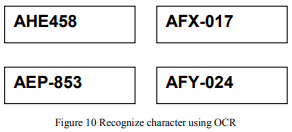
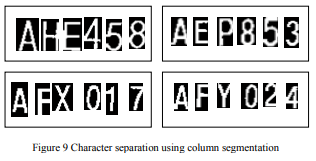
**2.1 Discussion**

This section presents the simulation results for developed the ANPR system. First, the camera is connected using MatLab via PC. The camera is attached using a USB port. The opposite pictures of cars of different colors and types of buildings are the same taken and stored on a PC. Different daylight effects they are also considered during consideration. Photos included RGB format and resolution at 800 x 600 pixels as shown in Figure 3. After taking the picture the next step was yellow search algorithm. Figure 4 shows the pictures after uses a yellow search algorithm. White region represents yellow or a near yellow color. Can be note that the yellow search algorithm has been found successfully a ROI that contains only a plate number of the car. The to lubricate the algorithm used next to extract the vehicle number plate as shown in Figure 5. If the vehicle number plate is extracted, converted to binary format. Figure 6 no Figure 7 shows the binary and contradictory format respectively. Row and column separation methods are used next to remove each character from the car number plate. The results of the row and column divisions are shown in Figure 8 and Figure 9 respectively. In the end the OCR is used for character recognition and individually the alphanumeric character is seen as shown in Figure 10.









**3. CHARCTER SEGMENTATION**

After finding the number, look for the letters of an ongoing process. As the plate separation exists various ways available to make a character separation. As many paths fall into more than one category it is impossible to make a category of intelligent conversations. In this section is a common related activity in this area followed by discussed. Other methods such as image binarization and CCA have been discussed in section 2 can will also be used for character classification.

**3.1 Related Work**

In [5], the constituency is determined by **78 X 228** Pixels using bi-cubic translation and then linked to SCW by segment. Authors have used a limit of **0.7** to optimize results. After the character sorting process, each character changed to a pixel size of **9 X 12.** Prathamesh Kulkarni et al. [24] concludes with blob Color schemes and altitudes are not appropriate Indian plate number. Authors have suggested the release of the image algorithm in which the number plate is directly scanned again scissored in a line where there is no white pixel with this the details are stored in the matrix. In the event of more than one matrix, the false matrix is ​​discarded according to the given formula in this paper. The same process is repeated with the horizontal direction by taking the width as a limit. CCA is a very useful way of processing binary picture. In [19] direct and indirect adjustment to the image development is done as pre-processing steps for punctuation. The CCA is used horizontally as well vertical adjustment. After performing these steps the plate is approx converted to black letters / white background and regenerated to **100 X 200**. Then all the characters are divided into unique size **32 X 32**. In [44] image separation also the means of connecting the connected objects are used In [16], three matrics are used to keep the plate location and binarization, number of columns in BW and line value in BW respectively.

After that accurately upper and lower boundary area found, namely followed by vertical guessing and Thresholding in the section characters. Erdinc Kocer [6] used the comparative extension, Medium filtering and coloring techniques for the character separation. Comparison extension is used to create an image it is sharp. According to H. Edincinc Kocer histogram equalization is a popular process to improve the appearance of the poor opposed image. In the middle filter which is not required for noisy regions are deleted. The Blob calculation method is used in the binary image to find closures and contacts with sub-regions. In this way, L The shaped template is used to scan the image from left to right as well top to bottom. This scanning process is used to locate independent regions by access to four contacts directions from a blank domain. Four directions The blob coloring algorithm is used in the license of binary coding bulletproof plate. At the end of this process numbers separated by size **28 X 35** no characters separated by size **30 X 40**. One an algorithm based on finding the blob proposed in [14]. The The process of separating characters consists of uppercase letters balance, character measurement and blob output. Character height estimates consist of three parts: color retreat, vertical edge detection and horizontal histogram.

Background color is used to color the licensed plate characters as black by using mathematical analysis at the edges. The straight edge detection is used to find a plate of a completed number. Sobel Mask and the image binarization image algorithms used to perform it. The histogram of the Horizontal projection is used to find the top and bottom character limit. Distance between lower limits are considered the height of the character. Character encoding consists of: image binarization and give projection histogram. Image binarization was used make the color black and white. Vertical guesses are used find spaces between characters. Process is similar to horizontal guess. Blob extraction is a two-step process including Blob detection and blob test algorithms. The blob detection algorithm extension CCA. Blob lookup is used to remove non-blob characters from separated characters. In a bullet clipper is used to separate character in a rectangular box. After that you use the extraction feature, classifier, post processor and training classes for each character separated. In [24], an improved guessing method (IPM) is proposed. The authors describe a three-step process of punctuation. In the first step horizontal, vertical again attached tilts are fixed. Then with the next two helpful steps lines are drawn between the first and last character to be found connected boundaries. In the final stage the characters are divided into sections after removing the sound. Tests were performed to use MATLAB 6.5 using VC ++ 6.0. According to Thomas, Nicolas et al. connected part token is accurate but may result in failure due to one pixel error written incorrectly. The writers also concluded that histogram projector is very powerful but not very accurate. Use the connected method for connected objects in the first set of lines. Specific and horizontal histogram methods are used in [11]. Determining the parameters of the characters, total of the column using vectors. According to two nearby algorithms the letters are divided into two parts.

**3.2 Discussion**

Separation of characters is very important in order to be done character recognition with the correct amount of accuracy. Sometimes letter recognition is not possible due to error by character separation. In other ANPR publications, The classification of characters is not mentioned in detail. Others methods such as image binarization, CCA, vertical and horizontal guessing can produce better character results separation.

**4. CHARACTER RECOGNITION**

As discussed in Section 2, character recognition helps in identify and convert text to an image. As most number plate recognition algorithms use one character recognition method. At this stage, each approach explained.

**4.1 Artificial Neural Network(ANN)**

The Artificial Neural Network (ANN) is sometimes known as the neural network is a mathematical term, which contains artificial neurons are integrated. Several algorithms are similar [5], [6], [7], [8], [18], [11], [10] are based on ANN. In [5] a layer of two possible layers of anatomy of 180-180-36. The character recognition process was performed in 128ms. In [6] multiple concentrated perceptron (MCP) The ANN model is used for character classification. In contains an input layer for decision-making, a hidden layer in calculate the most complex organizations and the output layer of the decision has already been made. Backward Feed (BP) an algorithm was used to train ANN. BP neural network removed systems are proposed in [8], [18], [10] for processing a period of 0.06s. In HNN is used for reduction ambiguity between similar characters e.g. 8 and B, 2 and Z etc. The authors claim to have more than 99% recognition measure.

**4.2 Template Matching**

Template matching helps to determine the default size characters. It can be used for acquisition usually in face detection and image processing. It further divided into two parts: feature-based comparisons and template-based simulation. A feature-based approach is helpful where the template image has solid elements outside the template an established approach can help. In the mathematical aspect [34].Exit method used to get **85%** character visual measurement. In [15], several features were also removed salient is calculated based on training letters. Line a standard algorithm is used to align all characters with same size. A **95.7%** recognition level is available among 1176 photos. SVM-based method used for the release of the Chinese, Canaanite and English element, Numeric characters. The authors achieved **99.5%** success, **98.6%,** and **97.8%** of values, Kana, and address sequential recognition. The template-based method is proposed in [16]. The authors used a low-resolution template the same method of working with a low resolution image as **4 X 8**. Authors used similar work to measure similarities between patterns.

**4.3 Other Methods**

In some algorithms the character recognition is done by Optical Character Recognition (OCR) tool. There they have software available for OCR use. One of OCR open source tools have multiple supported languages , also maintained by Google is found in [76]. Used in [14] character recognition. The author converted it for the benefit of **98.7%** of the characters visual measurement. Of the authors of the release model characters like Markov Random Fields (MRF) is used to match uncertainty in assignment of pixels. Character release is done as a usage problem based on previous information to enlarge the posteriori opportunities. Then a greedy person is used to make changes calculation costs. The proposed method [4 contains three steps: character classification, subject sorting an self-organizing (SO) recognition. The character set is approx used to distinguish a letter such as an alphabet or number. The second Characterization features are also calculated compared to saved character templates. The corresponding templates will form a test set, in which a character very similar to the input character determined. Template testing is performed by the SO character how to see The neural network that you have set up is based on Kohonen feature maps are designed to manage audio, broken, or incomplete characters. The same distinction letters from the pairs of letters such as **(8, B)** and **(O, D)** the authors describe a mysterious set containing characters 0, 8, B and D. In each character set, vague parts of the character are specified as shown on the fig tree. 3. After an unknown character is classified as one of characters in a set of mysterious, small comparisons between an unknown character and a character set by done. Then there is the only non-comparative process the complex parts of the characters are focused. Authors **95.6%** recognition rate has been obtained for a standing license plate pictures. In research on the default character detection method is discussed. There is major challenge in imitation recognition to handle anonymous text formats, a different font sizes, different lighting conditions, reflection, dimming and aliasing.

**5. CONCLUSION**

ANPR could continue to be exploited by the car owner vehicle identification, control model, vehicle speed control and vehicle location tracking. It can also be extended as a multilingual ANPR to identify the file language of training-based characters data It can provide various benefits such as traffic safety enforcement, safety- in case of suspicious activity by car, easy to use, quick access to information- like compare car owner registration details manually and costly in any country At a minimum graphics to solve other super algorithm-like development algorithms the correction of the images should be focused. Most of ANPR focuses on the continuation of a single vehicle number but in real time can have more than one car number plates while photos are being captured. With a lot of car [5] number plate images are considered ANPR while most of other programs offline car photos, taken online a database such as [78] is provided as an input to ANPR and therefore directly results may deviate from results.

**Summary**

In this work, we have demonstrated that through the License Plate In recognition a person can use small convolutional neural networks. The LPRNet model was introduced, which could be used for challenging data, up to **95%** visibility accuracy. Details of architecture, its mobilization and ablation research were conducted. We have shown that LPRNet can generate real-time indication of various types of Hardware including CPU, GPU and FPGA. We have no doubt that LPRNet can get real-time performance on even the most embedded low-power devices. LPRNet may be compressed using modern technology pruning and measurement methods, which can help reduce computer complexity even more. As a future direction of research, LPRNet activity could be expanded by integrating the CNN-based acquisition component into our algorithm, so that the acquisition and detection functions will perform tested as a single network to achieve the performance of LBP detector quality based on LBP.

**6. ACKNOWLEDGMENTS**

The authors thank Mr. Anil Dubey for donating the guidance needed to conduct this study. Authors and thanks to the ABES Engineering College to provide the necessary resources for to do this research.

**7. REFERENCES**

[1] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, “License Plate Recognition From Still Images and Video Sequences: A Survey,” IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 3, pp. 377–391, Sep. 2008. 1

[2] H. Li and C. Shen, “Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs,” arXiv:1601.05610 [cs], Jan. 2016, arXiv: 1601.05610. 2, 4

[3] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, 2012th ed. Heidelberg ; New York: Springer, Feb. 2012. 2, 3

[4] A. Graves, S. Fernndez, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 369–376. 2, 3

[5] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. 2

[6] T. K. Cheang, Y. S. Chong, and Y. H. Tay, “Segmentationfree Vehicle License Plate Recognition using ConvNetRNN,” arXiv:1701.06439 [cs], Jan. 2017, arXiv: 1701.06439. 2

[7] V. Jain, Z. Sasindran, A. Rajagopal, S. Biswas, H. S. Bharadwaj, and K. R. Ramakrishnan, “Deep Automatic License Plate Recognition System,” in Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing, ser. ICVGIP ’16. New York, NY, USA: ACM, 2016, pp. 6:1–6:8. 2

[8] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial Transformer Networks,” arXiv:1506.02025 [cs], Jun. 2015, arXiv: 1506.02025. 2, 3

[9] H. Li, P. Wang, and C. Shen, “Towards End-to-End Car License Plates Detection and Recognition with Deep Neural Networks,” ArXiv e-prints, Sep. 2020. 2

[10] X. Wang, Z. Man, M. You, and C. Shen, “Adversarial Generation of Training Examples: Applications to Moving Vehicle License Plate Recognition,” ArXiv e-prints, Jul. 2020. 2

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Networks,” ArXiv e-prints, Jun. 2014. 2

[12] W. Liu, A. Rabinovich, and A. C. Berg, “ParseNet: Looking Wider to See Better,” arXiv:1506.04579 [cs], Jun. 2015, arXiv: 1506.04579. 2, 3

[13] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters. 2,3

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” arXiv:1602.07261 [cs], Feb. 2020, arXiv: 1602.07261. 2, 3

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper with Convolutions,” arXiv:1409.4842 [cs], Sep. 2014, arXiv: 1409.4842. 2, 3

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” arXiv:1512.00567 [cs], Dec. 2020, arXiv: 1512.00567. 2, 3

[17] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” arXiv:1502.03167 [cs], Feb. 2015, arXiv: 1502.03167. 2, 3

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, Jan. 2014. 2

[19] S. Agarwal, D. Tran, L. Torresani, and H. Farid, “Deciphering Severely Degraded License Plates,” San Francisco, CA, 2020. 2

[20] A. Hannun, “Sequence modeling with ctc,” Distill, 2017, [https://distill.pub/2017/ctc. 3](https://distill.pub/2017/ctc.%203)

[21] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” arXiv:1603.04467 [cs], Mar. 2016, arXiv: 1603.04467. 3

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” arXiv preprint arXiv:1408.5093, 2014. 4

[23] U. Aydonat, S. O’Connell, D. Capalija, A. C. Ling, and G. R. Chiu, “An OpenCL(TM) Deep Learning Accelerator on Arria 10,” arXiv:1701.03534 [cs], Jan. 2017, arXiv: 1701.03534. 4

[24] “Intel OpenVINO Toolkit | Intel Software.” [Online]. Available: https://software.intel.com/en-us/articles/ OpenVINO-InferEngine 4