e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:01/January-2023 Impact Factor- 6.752 www.irjmets.com

AUTONOMOUS NUMBER PLATE IDENTIFICATION SYSTEM

Vasu Goel[*1]

[*1]Department Of Information Technology, Maharaja Agrasen Institute Of Technology, Delhi, India.

DOI : https://www.doi.org/10.56726/IRJMETS32994

ABSTRACT

A promising study subject in smart urban and the Internet of Things is number plate recognition, also known as licence plate realization or recognition utilizing image processing techniques. Many of the autonomous number plate recognition systems in use today only function in a controlled setting with photographs taken at a straight angle with acceptable lighting, clarity, and industry-standard typefaces. The fact that existing works are based on UK licence plates and may not function for Indian licence plates is another disadvantage. This study provides a novel image processing method for the detection and identification of Indian licence plates that can handle noisy, dimly lit, cross-angled, and non-standard font licence plates. In the pre-processing of this work, morphological transformation, Gaussian smoothing, and Gaussian thresholding are only a few of the image processing techniques used. Next, contours are applied by boundary following and filtered based on symbol proportions and spatial localization for number plate segmentation. After the target region filtering and de-skewing, character identification is done using the K-nearest neighbour technique. The suggested techniques showed encouraging outcomes.

I. INTRODUCTION

The proposed effort will utilise cutting-edge picture segmentation and other processing methods to the identification and realisation of licence plate numbers. The majority of recent works focus on common Number plate formats, as those used in the UK, Argentina, or Russia. Intelligent Template Matching, a revolutionary technique introduced by Nicolas et al., is focused on Argentinian licence plates [1]. The majority of plates in a huge nation like India are not standard, hence a solution that takes into account various formats and fonts is required. Additionally, since there is no uniform data set for Indian licence plates, pictures and videos of vehicles with licence plates were manually collected, taking into account issues like dim, uneven lighting, distant plates, fuzzy plates, different fonts, etc. The hypothesis assumes that the number plates follow the prescribed sequence because all the data were gathered in India and the majority of them include four characters and six numbers: The very first two characters of AP 31 BA 1432 represent the state in which the vehicle is registered. First, the Area of Concern needs to be found and extracted from the wider vehicle taken image. The next step is to identify the characters and digits that are present in the area of interest. Because the number plate region only makes up 10% of the whole imagery of the vehicle under consideration, 90% of the entire image can be ignored by localizing the search to the area of concern. The reduction of time and spatial complications is another benefit of number plate recognition.

Lotufo et al. performed one of the earliest studies on automated number plate recognition (ANPR) [2]. For the example of the Dutch number plates, Nijhuis et al. [3] utilised neural networks with fuzzy logic to recognise vehicle licence plates. Neural networks were utilised for classification and feature extraction while fuzzy clustering was applied to segmentation. Kim et al. [4] address automatic licence plate recognition for photos that have distortion. They attained a 92.8% accuracy in image segmentation using genetic algorithms. Lee et al. [5] conducted yet another study on the identification of Korean licence plates. In t heir research, the area of concern is first identified using neural networks for colour classification. The classification process makes use of the pixels' Hue, Lightness, and Saturation (HLS) values. The ratio of the plate's length to breadth is utilised to determine the plate region. Character extraction uses a histogram of the character's colour, and identification uses a template matching technique.

For an instance with a distorted background, Hsieh et al. [6] offer wavelet transformation-based character extraction. With a 92.4% accuracy, solely plate detection is taken into account in this work. Quadri and Asif [7] perform Sindh standard licence plate recognition by using yellow colour identification to corner the number plate region, a smearing method to segment the plate, and optical character recognition (OCR) to identify the

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:01/January-2023 Impact Factor- 6.752 www.irjmets.com

characters. However, neither the types of OCR algorithm utilised nor the accuracy rate are specified. Number plate detection was carried out by Surekha et al, [8] however no character recognition was reported. In their research, they often use morphological processing, cunning edge detection, and neural networks. A 97% accuracy rate was recorded. They did not take into account the photographs' complicated contextual factors when developing their strategy.

In her work on Indian number plate detection and recognition, Kaur [9] used morphological procedures to extract the number plate region, boundary box analysis and connected component analysis to extract characters, and template loading and matching to recognize characters.

Character recognition was claimed to have a 96.67% accuracy rate. This method has the advantage of being able to identify blurry, noisy, and low contrast images. However, this method does not address slanted licence plates. Chinese licence plate recognition was carried out by Shi et al [10] for varied picture illuminations. The recognition of numbers was 90% accurate. The test cases do not, however, take into account photos that are noisy, blurry, or skewed. An effective licence plate detection and recognition method was put out by Chang et al. [11] Number plate extraction is done via colour edge projection and Hue Saturation Intensity (HIS) dependent on colour modification techniques, such as fuzzy aggregation. Character recognition uses OCR techniques. An efficiency of performance in noisy settings, various lighting conditions, and even damaged licence plates was reported with an accuracy of 93.7%. The Indian licence plates, however, are not examined. Tejas et al. [12] suggested utilising neural networks for recognition, Sobel edge detection, bounded area segmentation, and Indian number plate detection. However, their methodology did not test it in adverse environmental circumstances [13]. For licence plate region detection, Salau et al. [14] achieved a remarkable accuracy of 99.8%; character recognition was not taken into account.

The pre-processing part of this work includes the use of a number of images processing techniques, including Gaussian smoothing, Gaussian thresholding and morphological conversion. Next, contours are applied by boundary following and filtered based on character proportions and spatial localization for number plate segmentation. After the target region filtering and de-skewing, character identification is done using the K-nearest neighbour technique.

II. METHODOLOGY

Pre-processing, detection, and recognition are the three main stages of the suggested methodology.

2.1. Pre-processing:

Both images and videos are acceptable as input. Video is viewed as a collection of images or frames, so the image source needs to be prepared for additional processing before number plate recognition can begin. The sequence in which image processing methods are used is as follows:

2.1.1 Image Under-Sampling: The algorithm for number plate identification and recognition is designed to operate at a constant frame rate. It should come as no surprise that image processing algorithms take longer to process high-resolution photos. In actuality, considering photos that have a high definition is superfluous. If the resolution exceeds a set threshold, this stage lowers it. Videos are under-sampling every 20th frame and transmitted for additional processing.

2.1.2. RGB to HSV Conversion: The following OpenCV function transforms an input RGB image to an HSV image and returns the transformed image. The image in this case is saved internally as a numpy array and cv2. The RGB to HSV conversion flag is called COLOR BGR2HSV. This instructional discusses extensive documentation for changing colour spaces [15]. HSV channels provide the added benefit of separating brightness from colour specification. This guarantees that the method will function for photos with a variety of light levels.

2.1.3. Grayscale extraction: To create the grayscale image, the luminance or value channel of the transformed HSV image is extracted. Without a doubt, the resultant grayscale image differs significantly from it's own RGB grayscale equivalent.

2.1.4. Morphological transformations: Morphological transformations, which are certain procedures that can be carried out on binary images, include top-hat and black-hat filters. The black-hat operation, also known as bottom-hat, is used to accomplish the opposite, i.e., boost dark items of interest in a slightly brighter

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:01/January-2023 Impact Factor- 6.752 www.irjmets.com

foreground. The top-hat operation is used to amplify brightness of items of interest in a relatively dark background. According to mathematics, the Top-Hat is the ratio between the picture and the image's beginning, and the Black-Hat is the difference between both the image and the image's ending. In this study, black-hat results are deleted from the original image and top-hat results are added [16].

2.1.5. Gaussian Smoothing: A linear Gaussian function is used in Gaussian filtering or smoothing. Reducing noise and specifics is the goal of Gaussian filtering. This will be useful for later stages of picture processing. The additional benefit of avoiding aliasing artefacts is another benefit of applying a Gaussian filter to an image. Every point in the given array is fused by using Gaussian equation in the process of gaussian filtering [17]. The total of all of these points yields the output array. To create a two-dimensional matrix, a series of one-dimensional Gaussian matrices is first applied horizontally, and the procedure is then repeated vertically. Comparing that to its two-dimensional equivalent leads to a reduction in computational complexity [18].

2.1.6. Inverted Adaptive Gaussian Thresholding: Thresholding is used in image segmentation to convert a grayscale image into a binary image. Famously referred to as image binarization, this procedure. The image is binarized using a global threshold in the simplest thresholding technique. This approach, however, might not be appropriate in non-uniform lighting situations. To carry out the adaptive thresholding, a window of predetermined size is taken, and a weighted sum of nearby pixels is calculated. A mathematical negation called a "inversion" is carried out to meet our practical requirements [19].

2.2 Number plate detection:

Inverted Adaptive Gaussian Thresholding, the last step in the image pre-processing process, outputs a binaryized a picture with the values 0 or 255. The binarized picture is used as a starting point for later detection and recognition processes.

2.2.1. Applying contours: The algorithm used to create contours is known as contour tracing, sometimes known as border following. A line of equivalent intensity points running along a boundary is known as a contour. Detecting contours in OpenCV is similar to finding a white object against a black backdrop, hence inversion operation had to be implemented during the adaptive gaussian thresholding stage.

2.2.2. Filtering and Grouping contours: The use of contours is made for tiny regions, particularly for jagged corners and noise outliers. Such outlines are superfluous, as may be seen by the human eye, but a software must take this into account. Each contour was first given a set of bounding boxes. Then, the following parameters were considered for each contour: the necessary contour area, necessary contour length and width, and the minimum and maximum permitted aspect ratios. Consequently, the majority of the pointless outlines were filtered out, bringing us closer to achieving our goal—detection of a licence plate. During the second level of filtering, each contour is compared with all other contour based on variables such the separation among outlines, the delta angle among them, the delta variation in their space, the delta width, and the delta height. When a set of contours meets each of these requirements, they are combined into one. It is possible to get two or more of these groupings.

2.2.3. Plate angle correction: Each number plate is given a boundary box at this point. An affine transformation is carried out if any of the plates have angle distortion. To conserve points and lines, an affine transformation—a mapping between two spaces—is utilised. The parallel lines maintain their parallelism after the change. The distances between both the points that form a straight line are still proportional. The lengths of the points and the angles inside the lines, however, are not maintained [20]. The OpenCV function getRotationMatrix2D may be used to rectify the plate angle [21].

2.2.4. Remove overlapping contours: As with the number "zero," it is conceivable for several contours to entirely encircle one another. The inner contour may entirely enclose the outer contour if it is picked up during the contouring process. This behaviour may cause both contours to be identified during the recognition task as different characters. This step is thus included to assure the elimination of those overlapping characters.

2.3 Character recognition:

2.3.1. Characters transformation and Prediction: Each contour, which stands for a character on the licence plate, is then enlarged into a 20 by 30 picture after any overlapping characters have been eliminated. This procedure is carried out to guarantee compliance with the learning model's input format, in which each font

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:01/January-2023 Impact Factor- 6.752 www.irjmets.com

character supplied during training is a resized 20 × 30 picture. The symbol is predicted once the scaled image is supplied to the model. In order to produce a string of characters, this is repeated for every contour. The prediction step is unsuccessful for the other contour groups created, which do not resemble the licence plate. Number plates are identified by the group that has the greatest number of expected characters.

2.3.2. Training the model: The concept of K Nearest Neighbors algorithm was utilised for training. Several other models including Decision Tree Gradient Boosting were studied, however KNeighbours fared better. Randomized search was employed to find the model's ideal hyperparameters [22]. A rigorous search is conducted using a manually created space of a group of hyper-parameters that are associated with the learning algorithm in randomised search, which is an optimised version of variable sweep or grid search. Grid search is guided by performance indicators like cross-validation of the validation set assessment. Both the grid search as well as the randomised search examine the same parameter space. Although the parameter settings are somewhat comparable, randomised search takes much less time to perform. Within K Nearest Neighbors classification, a query point is assigned a class based on the simple majority vote of the nearest neighbour, which has the most neighbourhood support. Scikit-learn was employed for the model's implementation. KNeighbors Classifier [23], wherein k is indeed an integer, is offered by Scikit-learn. A larger k number has the benefit of suppressing noise, but the distinction between class borders will deteriorate. The optimum k-value relies on the information. Weight is a parameter that is also accepted by KNeighbors Classifier. When predicting, weight is a factor. Both uniform weight and distance are possible. In the case of "uniform" weight, all the points in a neighbourhood are given the same weights. When applying a "distance" weight, the points that are closer together receive greater weights while the points that are farther apart receive smaller weights. The model and a set of hyperparameters that were attempted during training are inputs for the randomised search. K and weight were given varied values for this research, and it was discovered that when K = 4 and weight = distance, the algorithm recognised objects more effectively. Several typefaces that approximate the typical Licence Plate fonts were utilised to train the model. The character is scaled to an uniform size of 20 by 30 pixels before being saved. This will guarantee that the model's input is consistent.

III. RESULTS AND DISCUSSION

The tests are performed on a computer running Windows 10 with 8GB of RAM and an i5 CPU clocked at 2.4 GHz. The image processing techniques are implemented using the Python OpenCV package. Images have been used to test the system. All of the aforementioned scenarios, including plates with erratic illumination, plates with stylized lettering, plates that were near up or far away, and plates that were angularly skewed, were taken into consideration as System test cases. For testing, pictures under various environmental circumstances are obtained.

The system has effectively recognized more than 96.2% of the symbols from licence plates. and from the dataset of 101 licence plates, containing Indian and international plates, 97% of the specified licence plates were successfully detected. The outcomes are also contrasted with comparable studies that focused on the detection and identification of Indian licence plates**.**

3.1 Number Plate Detection: Licence Plate Detection based on Image Processing is easier and quicker than machine learning-based as well as other complicated approaches since it just needs to perform basic mathematical operations. It will require a lot of work, though, to include a variety of assumptions and real-world examples. Machine learning-based models typically perform better in these situations. Any algorithm based on machine learning needs data to function properly. For this project, there wasn't a lot of information accessible. However, it has been demonstrated that image processing approaches outperform their machine learning-based equivalents significantly when there is a lack of data. As stated before, having a trustworthy input source is essential for accurate detection and recognition, particularly one that has a still camera with adequate shutter speed and lighting. Although the system occurs to work well for hazy instances, it isn't always a generic expression.

3.2 Number Plate Recognition: Up to a point, different typefaces' characters might be recognised. However, it's crucial to remember that while weighing different typefaces, a suitable balance needs to be struck. A model may become overfit and produce poor generalisation and biassed prediction if it has too many fonts. Less typefaces will cause the model to underfit, which will lead to inaccurate prediction. Due to the limited amount

of data, advanced models like gradient boosting also have a tendency to overfit the data. Poor forecasts were the outcome of this. Therefore, KNeighbours and other simpler models seemed to function well.

IV. CONCLUSION

Regarding Indian car number plates or licence plates, this study discusses licence plate detection and recognition. The main contributions of this study include taking into account difficult scenarios including changing lighting, blurry, skewed, noisy pictures, and non-standard and substantially worn-out licence plates. The pre-processing part of this study includes the use of a number of images processing techniques, including morphological transformation, Gaussian smoothing, and Gaussian thresholding. Next, for number plate segmentation, outlines are generated by boundaries followed and filtered based on character proportions and spatial localization. After the area of interest filtering and de-skewing, character identification is done using the K-nearest neighbour technique.

V. REFERENCES

[1] N. F. Gazcón, C. I. Chesñevar and S. M. Castroa. (2012) "Automatic vehicle identification for Argentinean license plates using intelligent template matching." Pattern Recognition Letters, 33 (9): 1066-1074.

[2] R.A.Lotufo, A.D.Morgan and A.S.Johnson. (1990) "Automatic number-plate recognition," IEE Colloquium on Image Analysis for Transport Applications, London.

[3] J. A. G. Nijhuis, M. H. Ter Brugge, K. A. Helmholt, J. P. W. Pluim, L. Spaanenburg, R. S. Venema and M. A. Westenberg. (1995) "Car license plate recognition with neural networks and fuzzy logic," Proceedings of ICNN'95 - International Conference on Neural Networks, Perth.

[4] Sang Kyoon Kim, D. W. Kim and Hang Joon Kim. (1996) "A recognition of vehicle license plate using a genetic algorithm-based segmentation," Proceedings of 3rd IEEE International Conference on Image Processing, Lausanne.

[5] Eun Ryung Lee, Pyeoung Kee Kim and Hang Joon Kim. (1994) "Automatic recognition of a car license plate using color image processing," Proceedings of 1st International Conference on Image Processing, Austin.

[6] Ching-Tang Hsieh, Yu-Shan Juan and Kuo-Ming Hung. (2005) "Multiple License Plate Detection for Complex Background," 19th International Conference on Advanced Information Networking and Applications (AINA'05), Taipei.

[7] Muhammad Tahir Qadri and Muhammad Asif. (2009) "Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition," 2009 International Conference on Education Technology and Computer, Singapore.

[8] P Surekha, Pavan Gurudath, R Prithvi and VG Ritesh Ananth. (2018) "Automatic license plate recognition using image processing and neural networks," ICTACT Journal on Image and Video Processing (IJIVP), 8 (4): 1786-1792.

[9] S. Kaur, (2016) "An Automatic Number Plate Recognition System under Image Processing," International Journal of Intelligent Systems and Applications, 8 (3):14-25.

[10] Xifan Shi, Weizhong Zhao and Yonghang Shen, (2005) "Automatic License Plate Recognition System Based on Color Image Processing," Gervasi O. et al. (eds) Computational Science and Its Applications – ICCSA 2005.Lecture Notes in Computer Science, vol 3483, Singapore.

[11] Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung and Sei-Wan Chen, (2004) "Automatic License Plate Recognition," IEEE Transactions on Intelligent Transportation Systems, 5 (1): 42-53.

[12] K Tejas, K Ashok Reddy, D Pradeep Reddy, K P Bharath, R Karthik and M. R. Kumar, (2018) "Efficient License Plate Recognition System with Smarter Interpretation Through IoT," Bansal J., Das K., Nagar A., Deep K., Ojha A. (eds) Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing, 817: 207-220.

[13] Md Yeasir Arafat, Anis Salwa Mohd Khairuddin, Uswah Khairuddin and Raveendran Paramesran, (2019) "Systematic review on vehicular license plate recognition framework in intelligent transport systems," IET Intelligent Transport Systems.

[14] Ayodeji Olalekan Salaua, Thomas Kokumo Yesufua and Babatunde Sunday Ogundare, (2019) "Vehicle plate number localization using a modified GrabCut algorithm," Journal of King Saud University - Computer and Information Sciences, In Press.

[15] "Changing Colour Spaces," [Online]. Available: https://docs.opencv.org/3.2.0/df/d9d/tutorial_py_colorspaces.html.

[16] Yu Liu, Beibei Dong, Benzhen Guo, Jingjing Yang and Wei Peng, (2015) "Combination of Cloud Computing and Internet of Things (IoT) in Medical Monitoring Systems," International Journal of Hybrid Information Technology, 10 (2):366-376.

[17] "Gaussian Blur -Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Gaussian_blur.

[18] "Image Filtering -OpenCV," [Online]. Available: https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html#gaussianblur.

[19] "Image Thresholding -OpenCV," [Online]. Available: https://docs.opencv.org/3.3.0/d7/d4d/tutorial_py_thresholding.html.

[20] "getRotationMatrix2D - Geometric Image Transformations - OpenCV," [Online]. Available: https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html#getrotationmatrix2d.

[21] "warpAffine - Geometric Image Transformations - OpenCV," [Online]. Available: https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html#warpaffine.

[22] "sklearn.neighbors.KNeighborsClassifier - scikit-learn docs," [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html.

[23] "Nearest Neighbours - scikit-learn docs," [Online]. Available: http://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification.

[24] M M Shidore, and S P Narote. (2011) "Number Plate Recognition for Indian Vehicles" International Journal of Computer Science and Network Security11 (2): 143-146