March - April 2020
ISSN: 0193-4120 Page No. 10163 - 10170

Monitoring the In-Traffic of a Community by Vehicle Number Plate Detection using Digital Image Processing

Menta Anudeep 1, Woonna Sreenitya 2, and Dr. M. Kanchana 3
1 2 Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India
3 Associate Professor, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai,India

Article Info
Volume 83
Page Number: 10163 - 10170
Publication Issue:
March - April 2020

Article History
Article Received: *24 July 2019*
Revised: *12 September 2019*
Accepted: *15 February 2020*
Publication: *12 April 2020*

Abstract:
Vehicle license plate recognition (VLPR) is very useful in permitting the vehicles inside the parking area. This project presents the details of a novel VLPR scheme in which the license plate is detected using YOLO (You only look once) algorithm. Character segmentation is done with contour detection algorithm. Also, the proposed scheme uses the minimum edit distance algorithm to identify missing or misclassified characters on the license plates. The project is implemented using Raspberry Pi and license plate detection is implemented using Python programming. It is found through experimentation; the design is 95.31% accurate.

Keywords: *License plate detection, segmentation, minimum edit distance*.

I. Introduction

In the present day's scenario, license plate recognition plays a vital role The VLPR has a lot of applications in real life. Consider a gated community, where hundreds of people live and a lot of vehicles enter and exit. Security here is a major concern. The authorities usually do not allow cars that do not belong in the community. The authorities usually look for the registration number in their database. The design we propose here will automate this process and do not let any outside vehicle in.

There are problems when it comes to using image processing techniques. Sometimes the license plate is detected improperly; sometimes the OCR fails to detect all the characters perfectly. Any such problems will lead to not letting the actual user inside the community. To solve this problem, a concept called minimum edit distance which helps us to find the matching license plate using a database.

The approach of this mechanism is divided into 6 steps such as license plate detection and isolation, character segmentation, character recognition, verification, hardware response and manual confirmation. A digital image is provided where the design first finds the license plate. Then the license plate part of the image is isolated. The characters on it are segmented. The segmented characters are given to an OCR where they are classified.

For detecting the license plate, we used YOLO. YOLO is an object detection model. It is trained to detect license plate in an image. After applying few preprocessing techniques like thresholding and filtering the contour detection algorithm is used to detect the characters. The segmented characters are classified using an OCR. The results from the OCR are not always accurate. The minimum edit distance algorithm will help in finding the best match using a database of license plate.

II. LITERATURE SURVEY

There are multiple steps in the process of extracting characters from the license plate. But each step can be implemented in different ways.

March - April 2020
ISSN: 0193-4120 Page No. 10163 - 10170

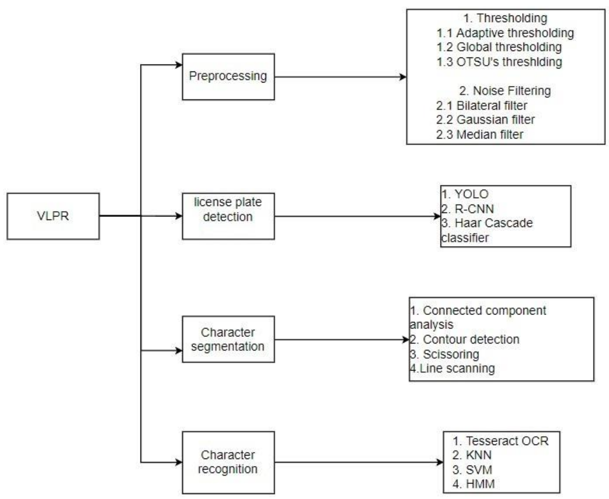


Fig 1: Taxonomy of VLPR

From the above Fig 1; it represents the classification of our work at various modules and all the sub-modules involved in VLPR, hence it was expressed as taxonomy of VLPR.

2.1 Thresholding

Global thresholding: The conversion of an RGB into a binary image using a pixel value threshold.

Local thresholding: The pixel value is decided based on the pixel values of its neighbors.

The Global thresholding has its drawbacks. If the illumination is quite low; the whole binary image gets dominated by black pixels. Similarly, if the illumination is high, the binary image gets dominated by white pixels. This will lead to loosing of details in an image.

2.2 Noise Filtering

Median filter: It's a technique used for removing noise. The pixel value is substituted with the median of the neighboring pixel values by the median filtering

Bilateral filtering: The intensity value of every pixel is replaced with an average weight of intensity values from nearby pixels.

2.3 License plate detection

License plates in images have high contrast due to their light background. License plates are usually horizontal rectangles. Using these properties, the license plate is detected.

Haar cascade classifier: The Haar cascade classifier is an object detection algorithm. Viola-Jones is its background algorithm. Many negative and positive images are used for training the cascade function. Initially, it takes a lot of images, then extracts feature from it. It is just like the convolutional kernel. Each feature is nothing but the difference between the sums of white pixels and black pixels.

R-CNN (Region-based convolution neural network): R-CNN like the Haar cascade classifier is an object detection model. In [5], the R-CNN first performs selective search on the image based on the color and texture regions. The selective search will result in small segmented area. All the smaller ones combine to become few large segmented areas. Then the segmented areas are classified. Detection of license plate is using properties like contrast and size is not accurate. It doesn't work when the image is captured in low light. Haar cascade classifiers are better than the first method, but these are not accurate. The R-CNN and YOLO are neural networks and work better than Haar cascade classifiers. YOLO is chosen as YOLO is faster than R-CNN.

2.4 Character segmentation

Line Scanning: First a binary image of the plate is generated using left to right scan [7]. Using 'lines' function the text on the license plate is found.

Connected component labeling: In [8], the method called connects component analysis produces connected regions by grouping pixels of similar pixel values. The pixels on the characters are grouped together and are identified as characters.

Image scissoring: In [9], the license plate is categorized into many images and each image has an isolated character. The image is first thresholded, and then the connected component labeling is performed. All the characters are resized using width and height.

The connected component analysis treats a group of pixels with similar pixel value as one body. It cannot differentiate between noise and details. Image scissoring doesn't work when the aspect ratio of a license plate is 2:3.

2.5 Character recognition

KNN (K Nearest Neighbors): KNN is a classification algorithm. It uses the distance between the points as a metric and classifies. A test point is classified by finding the distance from all the training points, then the distances are sorted. The first "k" points from the sorted list are takes. The maximum number point in the set that belongs to a particular class will be the class of the test point. Images are usually of high dimensionality. Calculating the distance from all training points is computationally expensive. KNN is not suited for Image Classification.

In the proposed design, to detect the license plate, the design proposed uses YOLO (You only look once). When the OCR misclassifies a character or when the character segmentation fails to detect all the characters, the minimum edit distance concept is used to match the result with the actual license plate using an existing database.

III. PROPOSED WORK

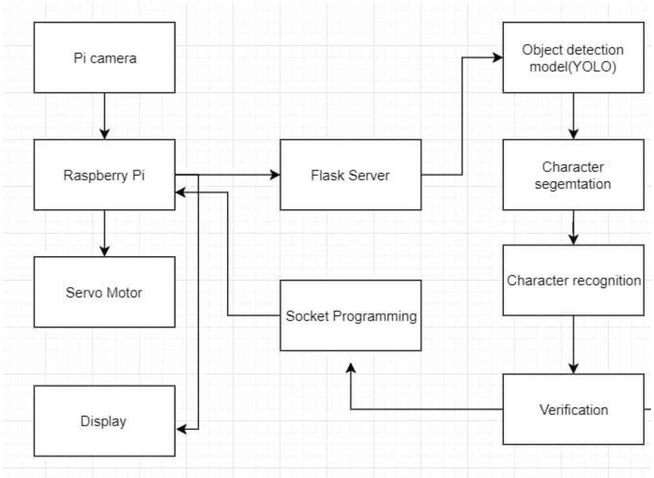


Fig 2: Architecture

The IoT device used here is Raspberry Pi. The raspberry pi sends images and waits for the local machine's response to open the gate. When the button is clicked the camera captures an image. The Image analysis part is hosted in Local machine and is divided into the following steps,

a. License plate detection and isolation
b. Character segmentation
c. Character recognition
d. Verification
e. Hardware response

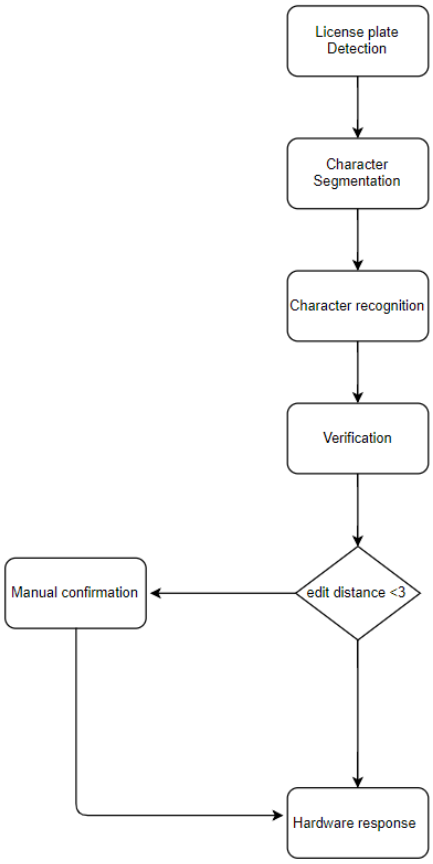


Fig 3: Flow of design

3.1 License plate detection and isolation

The model used in this project to detect a license plate is YOLO [4] (You only look once). The YOLO is a cutting edge object detection model, uses a CNN underneath. The YOLO is trained with various annotated car images. The model divides the image into an SXS grid. It iterates through each grid and calculates the prediction vector.

The prediction vector contains the probability of object's existence (p_c), x coordinate of top left coordinate (b_x), y coordinate of the top left coordinate (b_y), height and width of the license plate b_h and b_w respectively and the class c. The prediction vector is defined for every grid of the license plate. Pc is set to "1" of the grid has the object (i.e) the license plate.

The YOLO has a CNN working underneath. The loss function of the YOLO is the sum of classification loss, localization loss and confidence loss. The loss function used here is the mean squared function.

YOLO uses a single CNN for both classification and localization. Initially, the YOLO splits the input image into a SXS grid. The YOLO iterates to each grid and calculated a value Y. Y is called the prediction vector. The prediction vector is defined in the following way

Y = [p_c, b_x, b_y, b_h, b_w, c]

- pc defines whether an object is present in the grid or not .
- b_x, b_y, b_h, b_w specify the bounding box if there is an object
- c represents the class.

When training, the YOLO takes annotated images. Annotated images are the ones with the detection object and dimensions of the object are stored in an annotation file. The VOC type is an XML file with the details of the location of the object and its coordinates of a particular images

The coordinates are of this type (x, y, w, h). (x, y) denote the top left corner of the object and (x + w, y + h) denote the bottom right corner of the object.

When CNN trains, there are different types of losses involved, namely:

i. Classification loss
ii. Localization loss
iii. Confidence loss

Classification loss:

Usually, any loss function is the mean squared error between the predicted and the actual class. Here, when the YOLO iterates through every grid of training if there is an object the probability (Pc) is 1. The Probability determined by the YOLO is the predicted probability. The Mean squared difference between them is the Classification loss.

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

(1)

Localization loss:

This loss is the measure of the loss between the predicted bounding box and the actual box of the trained images. This loss is measured only when the grid has an object, that is if the P_c is 1.

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

(2)

Confidence loss:

The mean squared difference between the predicted confidence and the actual confidence. The Actual stays at 100% if there is an object in the grid.

$$\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

(3)

For a Grid, if there is an object, there can be any number of bounding boxes the set for that particular object. There can be a lot of bounding boxes that are suitable for an object. To fix this, YOLO uses non-maximal suppression.

IoU (Intersection Over Union): It is the measure of the similarity between two bounding boxes for an object. This method helps us in omitting the duplicates and chooses the best one.

$$IoU = \frac{size\ of\ the\ intersection\ area}{size\ of\ the\ union\ area}$$

(4)

The YOLO predicts multiple prediction boxes around the object. This multiple set includes duplicates of the same object. The YOLO uses IoU (Intersection over union) and confidence scores of each bounding box to eliminate duplicates.

The YOLO result will be the top-left and bottom-right coordinates of the object. Using these coordinates, the image of the object is isolated.

3.2 Character Segmentation

3.2.1 Gaussian Filtering:

The isolated image which is the license plate is resized into 1000X300 pixels. The image is converted into a binary image. There are various techniques like Global thresholding and OTSU's thresholding. The Global

March - April 2020
ISSN: 0193-4120 Page No. 10163 - 10170

thresholding takes a pixel value as the threshold and converts the whole image into a binary image. This method is not accurate all the time. The Local thresholding or the Adaptive thresholding decides the pixel value of a pixel based on the pixel value of its neighbors. The binary image is then filtered using the Gaussian filter [6]. The preprocessing of the image is done.

3.2.2 Contour Detection:

The Contour detection algorithm is applied to the resultant image [1]. Edges are computed based on extreme of the image gradient in its direction. Edge pixels signify local notion. Contours are obtained from the edges. The contours represent the spatial stretch of an object.

Table 1: Condition 1

Condition 1	j -1	j
	0	1

Table 2: Condition 2

Condition 2	j	j+1
	0	1

Contour detection algorithm:

First the image has to be scanned from top to bottom and left to right.

while (current pixel != bottom and right most pixel):

if(current pixel satisfies any of the conditions in the above tab) then Increment the NBD value of the pixel.Remember (i,j) and (i_2,j_2). If (condition 1 is satisfied) then the current pixel belongs to an Outer border or else the current pixel belongs to a hole border. Then End while. From the previous step decide the relationship between the current border and the previous border using tables 1 and 2. If the Previous border is an outer border and current is an outer border, then the previous border is the parent border of the current border. With Reference to pixel (i_2,j_2) around the current pixel (i,j) in clock wise direction find a non-zero pixel. Let the non-zero pixel be (i_1,j_1), (i_2,j_2) implies (i_1,j_1) and (i_3,j_3) implies(i,j).While ((i_4, j4) not equal to (i, j)) then look for another pixel counter clock from (i_2,j_2) with reference to (i_3,j_3), let the found pixel be (i_4,j_4).If ((i_3, j_3 +1) equals to 0-pixel) then leave the pixel (i_3,j_3) as it is. Else if((i_3, j_3 +1) equals to 1-pixel) then make the pixel value negative or else leave the pixel as it is and end while.

On the image, all the contours are detected. Bounding box around each contour is drawn to filter out unnecessary contours. The Bounding box of each contour gives us properties like height, width and area of the contours.

The contours are eliminated based on the set the conditions which involve the height, width and area. These conditions are set static as the isolated license plate image is resized to 1000X300.

The filters set to filter out contours are:

i. Width < 150 pixels
ii. Height > 75 pixels
iii. Area > 2500 Pixels

The condition for filtering will vary for other image sizes. The result after filtering the contours will be the coordinates of the characters on the resized image.

3.3 Character recognition

The segmented characters are classified using the Tesseract OCR [2],[3]. The Tesseract OCR uses KNN. The KNN used in Tesseract is quite optimized. It uses LSH (Locality Sensitive Hashing). It draws multiple hyper planes to cluster the training points. The test point is plotted and the KNN is applied to the cluster of training the test point falls under.

3.4 Verification

In a practical scenario, the OCR results are sometimes inaccurate. Least inaccurate results from the OCR can be matched with the reference database of the license plate. An inaccurate result is matched used minimum edit distance algorithm. The minimum edit distance algorithm provides us with a score between two strings. The score signifies the match between both the strings. Less the score more the match. The algorithm takes two strings and tries to change one of them to the other by inserting or deleting or substituting. The total number of operations will the score between the two strings. The license plate match from the database shall only be considered if the score is less than or equal 3.

3.5 Hardware response

After finding the perfect match from the verification process, the servo motor which was connected to raspberry pi will open the gate and allows the vehicle inside the community. Using UDP Socket programming we send the signal to open the gate. While using UPD Socket programming we connect our pi to pc which

TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 10163 - 10170

interfaces and helps to communicate with each other. UDP server side program will run continuously until a keyboard interrupt occurs. The gate will be open for 25sec after UDP's approval and give a green signal to leave otherwise it stays in red color until new entry to be found.

IV. RESULTS

Fig 4: Test Image

Fig 5: License plate detection result

Fig 6: Result of Gaussian blurring

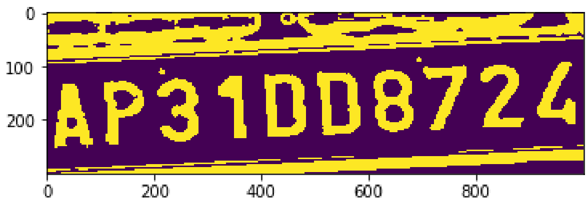


Fig 7: Result of adaptive thresholding

Fig 8: Result of Contour Detection

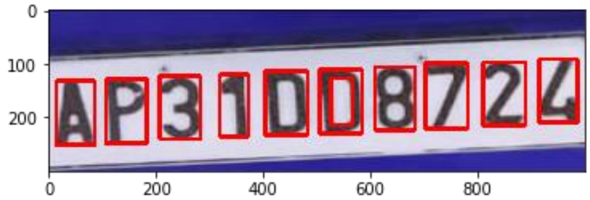
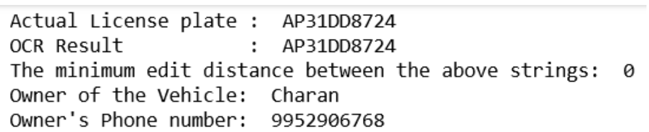


Fig 9: Result of Character segmentation by bounding boxes

Figure 5 describes the license plate isolated from the master image Figure 4. The resultant image is undergone through character segmentation which contains thresholding and the outcome is referred at Figure 6 &7. Figure 8 represents the result of contour detection. Figure 9 represents the Result of Character segmentation by bounding boxes. Figure 10represents the actual license plate string, the OCR result and the minimum edit distance between them. The license plate "AP31DD8724" already exists in my database. The minimum edit distance algorithm will find the actual string using the OCR result.



```
Actual License plate :  AP31DD8724
OCR Result           :  AP31DD8724
The minimum edit distance between the above strings:  0
Owner of the Vehicle:  Charan
Owner's Phone number:  9952906768
```

Fig 10: Final Result

Table 3: Performance Measures

Parameters	Input Images	Output Images	Accuracy
Number plate Extraction	64	63	98.43
Character Segmentation	64	62	96.87
Character Recognition	64	61	95.31

Table 4: Confusion matrix

	Positive	Negative
True	61	3

March - April 2020
ISSN: 0193-4120 Page No. 10163 - 10170

Table 5: Comparisonof Performance

References	Accuracy of license plate detection	Accuracy of Character Segmentation
[10]	96.22	94.04
[11]	96.4	98.82
[12]	97.1	96.4
[13]	90	90
[14]	93.2	95
Proposed work	98.43	96.87

Table 3 indicates the performance measures. It shows the number of images that are taken as inputs and it shows the outputs which are detected correctly by the system. It also shows the efficiency of the proposed work. Table 4 indicates the confusion matrix. The True Positive indicates the license plate of the vehicle belongs to the community and the system has detected it correctly. The True negative block indicates the license plate of the car belongs to the community and the system has not detected it from all these results. It is found through experimentation; the design is 95% accurate. Table 5 represents the performance results of the proposed method and how it out performs some other algorithms.

The time complexity of the algorithm varies from 1-3 minutes. The complexity can be decreased by using powerful local machine or a hosting it on an IoT device with high computational capabilities.

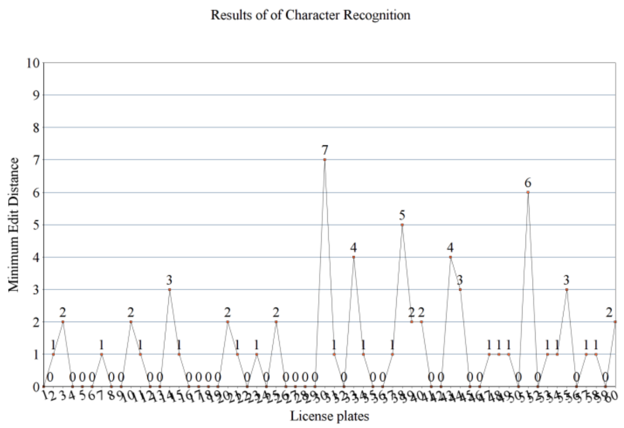


Fig 11: Observations

From the above Fig 11; we can observe the minimum edit distance of all the license plates from the values 0-7. In this project our device can extract 63 images out of 64 and the image which cannot is located at 30 and its value is 7, can segment 62 out of 64 images and the image that cannot be segmented is 51 and its value is 6, can recognize 61 out of 64 images and the image that cannot be recognized is 38 and its value is 5. Hence from the table 3 and Fig 11 explains the performance measures of our project.

V. CONCLUSION

In this proposed system, we designed a system which allows in the vehicles that belong to a particular community. This design can also be implemented in places like offices. First, the image is captured. The license plate from the image isolated, then it is preprocessed to find the characters on it. The OCR classifies the characters. The minimum edit distance works to support the results of OCR. . It is found through experimentation; the design is 95.31% accurate.

VI. REFERENCES

1. Efficient Scale-Adaptive License Plate Detection System
2. Miguel Molina-Moreno ,Iván González-Díaz , Member, IEEE, and Fernando Díaz-de-María, Member, IEEE-2018.
3. Lakshmi, G. Vara, and P. Kamala. "Improving Degraded Document Images Using Binarization Techniques." Int. J. Electron. Commun.Eng 5.2 (2016): 1-8.
4. R. Smith, An Overview of the Tesseract OCR Engine- 2017.
5. S. Banu Chitra1, Dr R.S.Vetrivel2 “Text Recognition using digital image processing techniques” International Journal of Computer Engineering and Applications, Volume XII, Issue I, Jan. 2018, www.ijcea.com ISSN 2321-3469
6. You Only Look Once: Unified, Real-Time Object Detection byJoseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi -2015.
7. Paliwal, Sonal, Rajesh Singh, and Mandoria HL."A survey on various text detection and extraction techniques from videos and images." International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) ISSN (P) (2016): 2249-6831.
8. Fast R-CNN, Ross Girshick, Microsoft Research- 2015.
9. Jain, Abhishek & Gupta, Richa. (2015). Gaussian filter threshold modulation for filtering

TEST Engineering & Management
March - April 2020
ISSN: 0193-4120 Page No. 10163 - 10170

flat and texture area of an image. 760-763. 10.1109/ICACEA.2015.7164804.
10. Chowdary, UndavalliVivek, et al. "Home automation system using IR sensors." International Journal of Electrical and Electronics Engineering 4.6 (2015): 11-1.
11. Anuja P. Nagare, “License Plate Character Recognition System using Neural Network”, International Journal of Computer Applications, (0975 – 8887) Volume 25– No.10, July 2011.
12. Napolitano, Salvatore, and Antonio Ascione. "Performance Posture Correlation: A Study in the Women's Water Polo." International Journal of Humanities and Social Sciences (IJHSS) 6.6 (2017): 53-62.
13. Khalid Maglad, Dzulkifli Mohamad, Nureddin A. Abulgasem, “Saudian Car License Plate Number Detection and Recognition Using Morphological Operation and RBF Neural Network”, 2011
14. C. Nelson KennadyBabu, Siva Subramanian T and Kumar Parasuraman Member, IEEE, “A Feature Based Approach for License Plate-Recognition of Indian Number Plates” IEEE, 978-1-4244-5967-4/10/$26.00, 2010.
15. M. Sarfraz, M. J. Ahmed, and S. A. Ghazi, “Saudi Arabian license plate recognition system,” IEEE Proceedings of the 2003 International Conference on Geometric Modeling and Graphics (GMAG’03), 2003.
16. Xiangjian He#1, Lihong Zheng “Segmentation of Characters on Car License Plates ” Conference Paper • October 2008.
17. Praseeja, P. S., and K. Gangadharan."Thyroid Segmentation and Volume Estimation Using CT Images." Int J Innovative Res Sci, EngTechnol 3.05 (2014): 334-9.
18. J. M. Guo et al., License Plate Localization and Character Segmentation with Feedback Self-learning and Hybrid-binarization Techniques, IEEE Region 10 Conference, 2007, pp.1-4.
19. Kulkarni Suraj Deeliprao&Prof. Pallavi S. Deshpande, “Vehicle Number Plate Recognition System for Theft Detection ”, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181Vol. 4 Issue 02, February-2015.
20. Balakrishnan, V. I. D. Y. A., K. P. Asifa, and K. C. Chitra."Genotoxic potential of nonylphenol in freshwater fish, Oreochromismossambicus." International Journal of Applied and Natural Sciences 3.2 (2014): 81-88.
21. C. Wu, L. C. On, C. H. Weng, T. S. Kuan, and K. Ng, “A macao license plate recognition system,” IEEE Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 2005, pp.4506-4510.