Turkish Online Journal of Qualitative Inquiry (TOJQI)

Volume 12, Issue 3, July 2021:2135- 2145

Research Article

Character Segmentation with Bounding Box Filtering on Indian License Plates

R. Gautham[1], D. Shailaja[2], S. Nagadevi[3]

Abstract

This paper presents a simple but effective method to segment characters through contour detection by filtering out contours of imperfections, but retaining that of the characters. This is done by filtering bounding boxes based on their height. The approximate height at which most bounding boxes lie is selected and those bounding boxes are retained, while the rest are filtered out. This method has significant potential in the application of Automatic License Plate Recognition (ALPR) systems. Compared to vertical project, which is the most common way to segment characters, this method is more versatile and provides better accuracy.

Keywords: *Character segmentation, bounding box filtering, indian license plate, contour detection, dimensional filtering.*

Introduction

Automatic License Plate Recognition (ALPR) system is becoming an increasingly popular technology, especially due to its application in automation of parking systems. This system consists of a sequence of important steps, one of which is character segmentation. Good character segmentation is the key to good performance of this system, because improper segmentation can severely affect the accuracy of character recognition which is the final step of an ALPR system.

The most common method used for implementing character segmentation is the vertical projection method [3]. Although this method works well for character segmentation in plain backgrounds, it fails to perform with license plates in practical conditions [2]. This is due to the fact that license plates can have various imperfections and may not have a perfectly uniform background for the text. Imperfections include bolts used to fasten the plate to the car, tiny specks of dirt, black borders around the plate, etc.

[1] UG Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Kancheepuram District, Tamilnadu, India.

[2] UG Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Kancheepuram District, Tamilnadu, India.

[3] Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Kancheepuram District, Tamilnadu, India, nagadevs@srmist.edu.in

Character Segmentation using contour detection method can perform well in this scenario, but a problem arises [4]. Contour detection will also detect the above-mentioned imperfections, and can totally hinder segmentation. In this paper we present an algorithm that filters out imperfections and significantly improve character segmentation on license plates.

Related Works

A. Vertical Project Method

This method segments the characters of license plates by calculating the vertical histogram projection value for pixel columns in an image, and segmenting it at the columns where the value dips below certain threshold. These dips occur in the spacings between characters, and hence can be used for segmentation. The vertical histogram projection value is calculated as the sum of the values of a column of pixel.

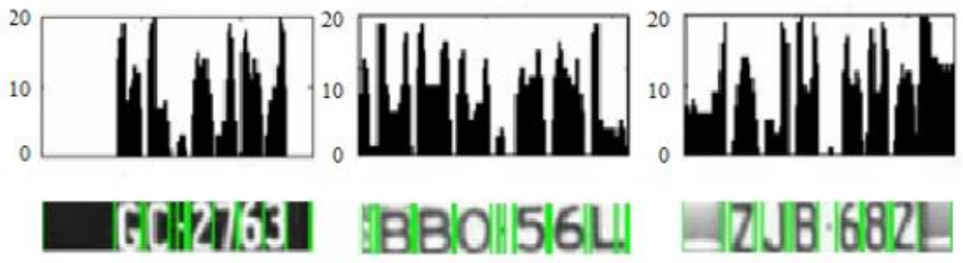


Fig. 1. Graphs for Vertical Projection

We can see from Fig.1 that the characters are segmented at the zero points in the projection graph [2]. The problem with this method is that it is required to have a clean background to the text after binarization.

Although this method works well for documents and other cases where a clean background is available practically, it fails in the case of license plates in day-to-day conditions. This is because the zero-points in the projection graph maybe altered due to imperfections such as peeled paint, bolts used to fasten the plate to the vehicle, or even pieces of dirt.

Since vertical projection methods don’t account for the position of these imperfections in the y-axis direction, it is impossible to separate them out, and will only work in ideal conditions and is not very robust.

B. Template Matching

Template matching is the most direct method of segmentation available [2]. This works by finding the border of the license plate, and matching the license plate template over it, and segmenting the characters based on it.

The template positions are found by training it with an annotated database of license plates, where the annotation contains the positions of the bounding boxes of the characters. The training can be done based on the distance between the correct positions of the top-left and the bottom-right points of each bounding box and trying to minimize it.

This method works well for countries where the license plate system follows a strict template. But this is not the case with Indian License Plates, and hence fails in this case. We can see that there are multiple templates that are accepted with Indian license plates from the Fig.2, Fig.3 and Fig 4.

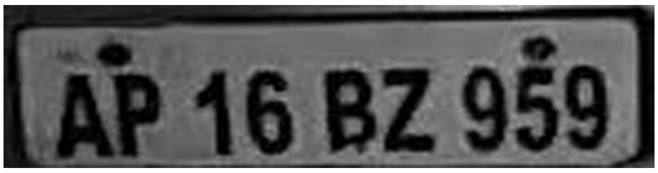


Fig. 2. License Plate with 9 Characters and Ending with 3 Digits

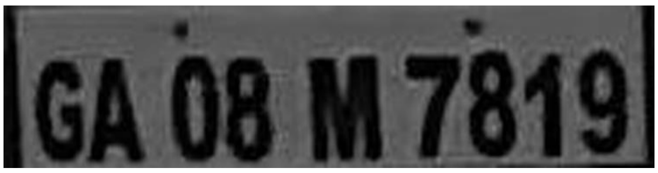


Fig. 3. License Plate with 9 Characters and Ending with 4 Digits

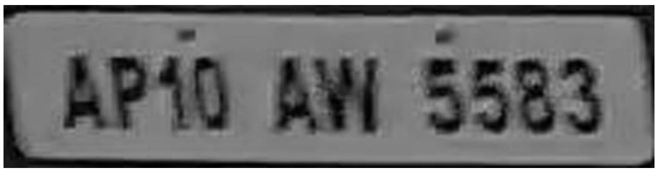


Fig. 4. License Plate with 10 Characters and Ending with 4 Digits

C. Segmentation using Sliding Window Method

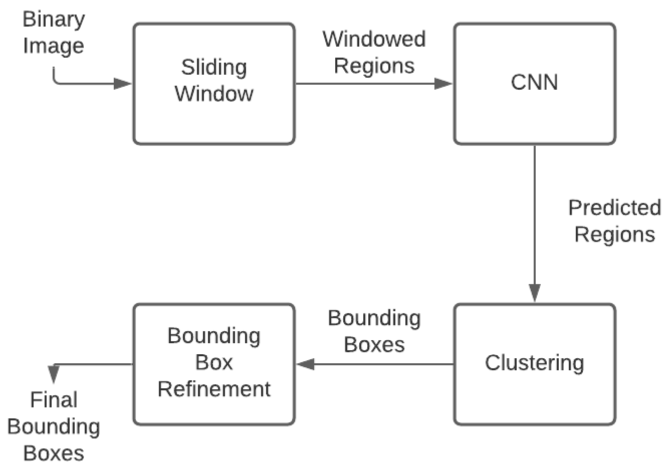


Fig. 5. Steps in Sliding Window Character Segmentation

In this method, the Sliding Window technique using Convolutional Neural Network is used to segment characters [5]. A window of a fixed size is initially trained to detect whether or not a character is present inside it. This window is then traversed, or in other terms, slid over the whole image with a constant step size. This method involves various steps are seen in Fig. 5.

This window can be a CNN network of any architecture, but should be architected in such a way that the output layer outputs a binary value, i.e., the network should detect whether of not a character is present inside it. An output layer with two nodes with SoftMax activation is generally used with inner layers using the ReLU activation.

This step detects the positions of the window in the license plate image where a character was recognized. The same character might get recognized in various position of the window if it is smaller than the window itself. To remove the duplicate positions and refine the bounding boxes, the positions are clustered by using any of the various clustering techniques available. Hierarchical Agglomerative Clustering finds a cluster of bounding boxes for each character and it performs well in this case [6]. It then outputs a median bounding box for each cluster as seen in Fig.6. This median bounding box is considered for the final output, and a refined set of bounding boxes for the license plate characters are obtained.

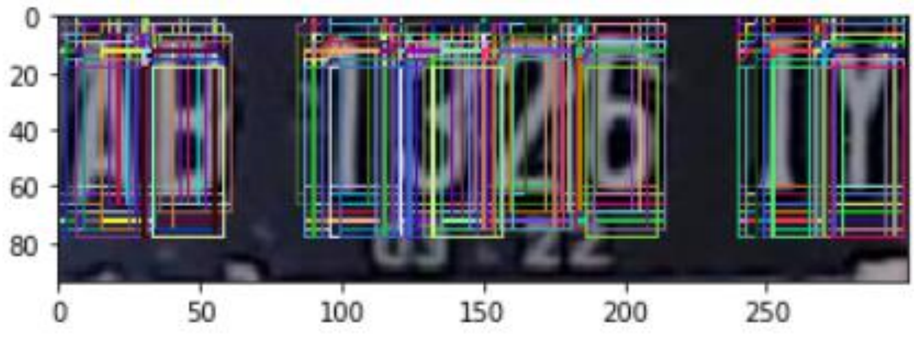


Fig. 6. Bounding Boxes Found by Sliding Window, which will be Clustered

D. Countour Detection Method

This is the method that is the most promising. In this method, we detect the contours of the character, and find the bounding box using these contours [4]. Contours are closed shapes that are formed by the edges of an object, in this case, the characters in the license plate. Contour Detection can be performed on Binary Images to segment out the characters, and it performs well, without needing any other steps if the background is uniform.

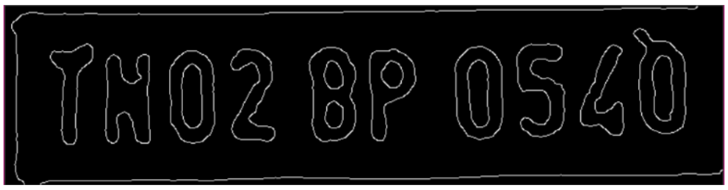


Fig. 7. Output of Contour Detection on a License Plate

We can see from Fig. 7 that each character is segmented using the edges. A drawback of this method is that it also detects the contour of the borders of the license plate, and also detects the contours of any imperfections as seen in Fig. 8.

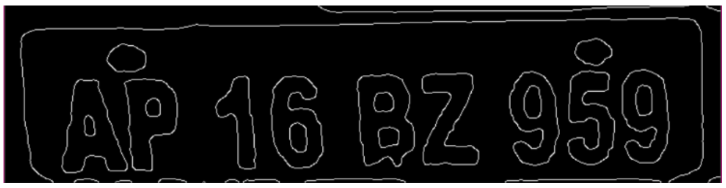


Fig. 8. Contour Detection with the Contours of various İmperfections

In this paper we improve upon this by proposing a filtering algorithm that can filter out these imperfections. Unlike Vertical Projection, contour detection differentiates between the contents of the license plate (including the imperfections) regardless of the axis. This allows us to separate the imperfections from the characters as we will go on to demonstrate later on in this paper.

Proposed System

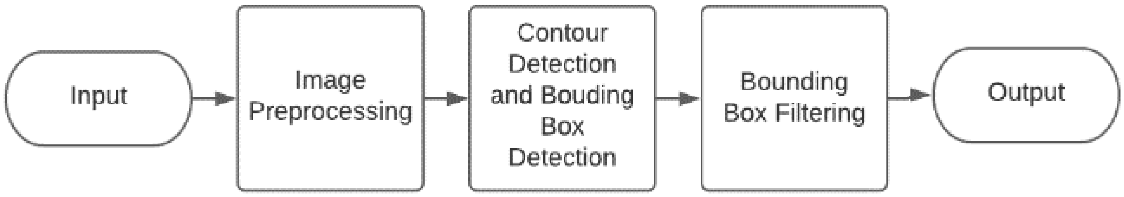


Fig. 9. Architecture of Character Segmentation

This system primarily uses Contour Detection method to detect bounding boxes [4]. It consists of three primary stages – Preprocessing, Contour Detection and Bounding Box Detection, and Bounding Box Filtering.

E. Image Preprocessing

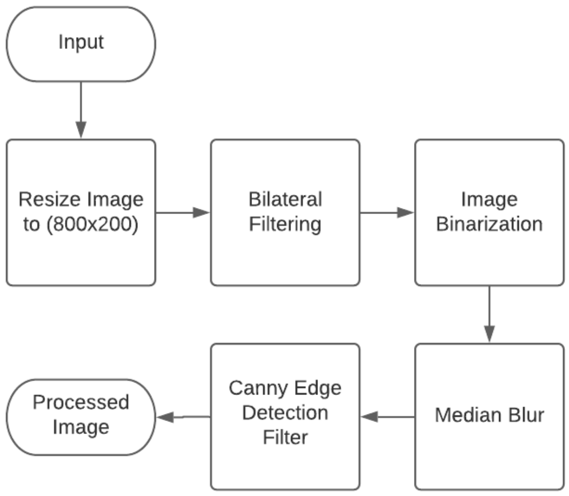


Fig. 10. Steps of Image Processing

This is an important step because it converts the raw image into a useful form. Contour Detection requires the image to be in binary color. To avoid excessive amounts of imperfection in the image, we can denoise the image using various methods. We have used two particularly:

i. Bilateral Filtering

Bilateral Filtering can filter out noises in the image while still retaining sharpness of the edges [7]. This preserves the edges of the character that we want to segment. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels.

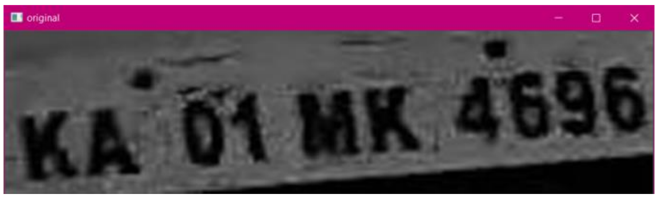


Fig. 11. Original Image

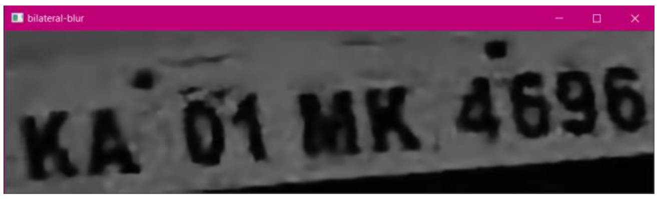


Fig. 12. Image after Bilateral Filtering

ii. Median Blur

Median Blur seems to work best after binarization of the image, since it retains the binarization, and effectively blacks out miniscule noises in the image [8]. Median blur works by running through each pixel, and replacing it by the median value of its neighboring pixels. When used on binary image, small imperfection can be easily erased out by neighboring pixels since they have the same value.

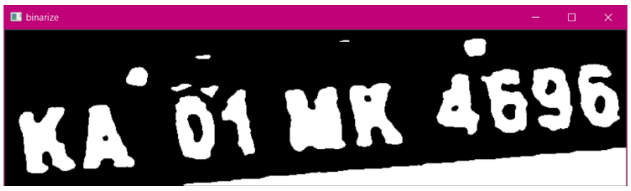


Fig. 13. Image after Binarization

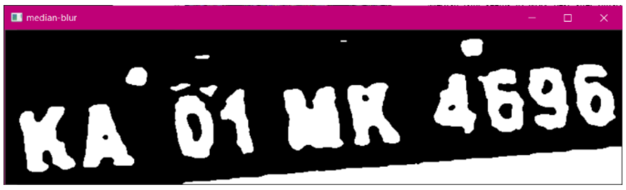


Fig. 14. Image after Median Filtering

As we can see from Fig. 14, median blurring reduces the size of small imperfection, and can sometimes erase them out.

i. Canny Edge Detection

Another important part of the processing step is Canny Edge Detection. Canny Edge Detector highlights edges present in the image using a multi-step algorithm and makes Contour Detection much more effective [9]. Canny Edge Detector works by finding the intensity gradient of an image and apply a Hysteresis Threshold to detect the edges.

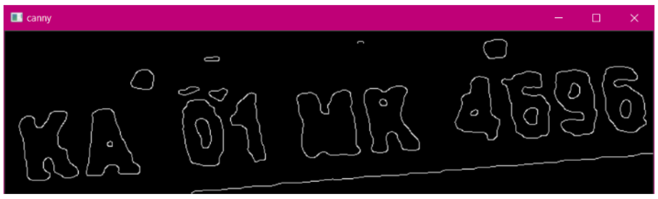


Fig. 15. Image after Canny Edge Detection

F. Contour Detection

Contours are curves that are formed by continuous points of the same intensity in an image. Contour detection not only detects the edges present in an image, but also places them in a hierarchy of closed shapes. The coordinates of the points of these contours can be used to detect various shapes and objects, including character of the license plate [4]. But contour detection also detects other shapes that are present in the license plate, including all the imperfections in the image. This paper addresses this issue by proposing an algorithm that can effectively filter out the bounding boxes imperfections from the bounding boxes of the characters.

G. Bounding Box Filtering

This step is the key to make this process work reliably. Bounding boxes detected through contours detect all the imperfections in the image that cannot be totally filtered out during the image processing steps. Imperfections such as screws, dirt, peeled paint etc., will also get their own bounding boxes. It is important to filter these out before we can start recognizing the characters.

The filtering process is done in two steps:

i. Dimensional Range Filtering

This filtering relies on a property of English alphabets and numerals - The capital letters and numbers are all of approximately the same height. This gives us the interesting opportunity of filtering bounding boxes that are of approximately the same height. Moreover, since it is highly unlikely to have a lot of imperfections of the same height, the smallest height range which contains the highest number of bounding boxes is most likely to be the range which contains the characters.

Algorithm:

Let,

H_MIN *= Minimum Height*

H_MAX *= Maximum Height*

BASE *= Base height within the range of (H_MIN, H_MAX)*

TRESH *= Threshold from base height*

BBOXES *= Unfiltered Bounding Boxes*

F_BBOXES *= Final Filtered Bounding Boxes*

T_BBOXES *= Temporary Filtered Bounding Boxes*

```
F_BBOXES = []
For BASE in range (H_MIN, H_MAX):
T_BBOXES = Filtered BBOXES with height in range (BASE, BASE+THRESHOLD)
If length of T_BBOXES is greater than length of F_BBOXES:
F_BBOXES = T_BBOXES
End If
End For
```

ii. Intersection Over Union Filtering

This step is important to remove doubled contours, caused by parameters served to the cv2.findCountour() function, from the result. We know that doubled contours highly overlap, so we can remove one of the bounding boxes when two have a high overlap, or in this context, high intersection over union [10].

Intersection over Union (IOU) is calculated as:

$$\text{IoU} = \frac{\text{Area of intersection of the two bounding boxes}}{\text{Area of union of the two bounding boxes}}$$

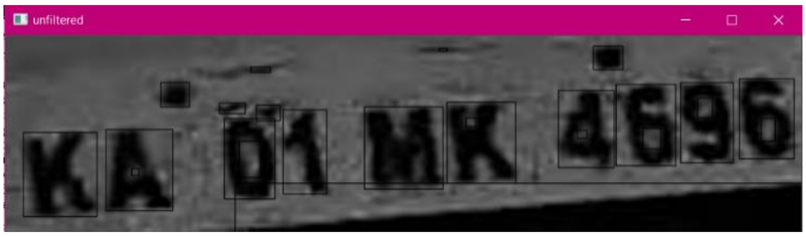


Fig. 16. Image before Bounding Box Filtering

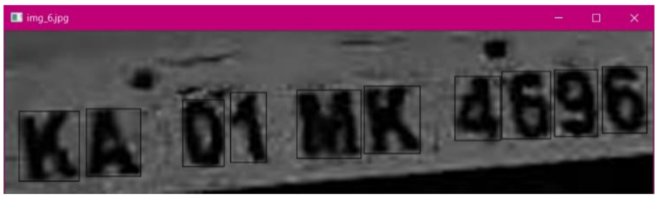


Fig. 17. Image after Bounding Box Filtering

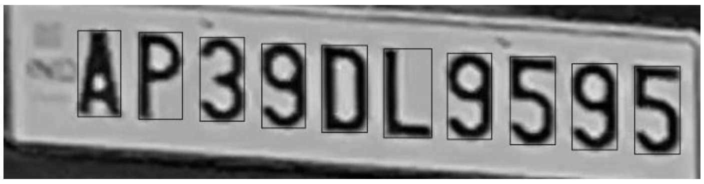
Analysis and Result

The following table shows the accuracy calculated by different metrics, and compares it with other segmentation methods. A dataset of 350 Indian License Plates was used.

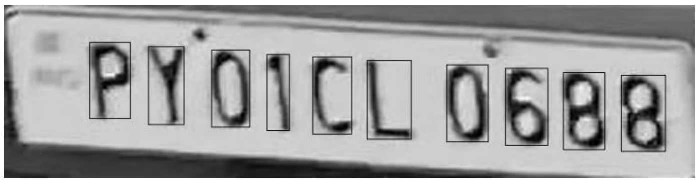
Table. 1.1

Accuracy Metric	**Accuracy in Percentage**		
	With Filtering	**Without Filtering**	**Vertical Projection**
Per Plate	96.45%	66.38%	75.4%
Total	85.71%	54.21%	63.67%

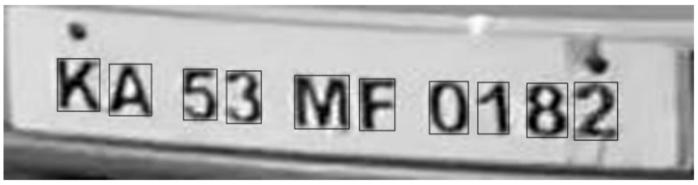
Per Plate – Average percentage of character that were segmented correctly in a single license plate
Total – Percentage of license plates that were segmented correctly.



(a)



(b)



(c)

We can see from images (a), (b) and (c) that this filtering algorithm works very well and filters out imperfections in the license plate background that may be detected by the contour detection step. We can see that the bolts holding the license plate to the car, and other small marking are being filtered out.

This algorithm is especially useful because OCR and histogram projection methods perform poorly in anything other than documents because of uneven backgrounds. Template matching technique also fails in Indian License Plates because the Indian License Plate template is not strictly enforced and hence will not work in a practical scenario.

Conclusion

In this paper, we proposed a method to filter bounding boxes of imperfections and make character segmentation through Contour detection more effective. This method makes character segmentation for license plates more versatile, and is not restricted by any strict templates. It also makes character segmentation more robust as imperfections cannot reduce the effectiveness of segmentation as much. The biggest issue with this approach is a very unlikely event that there are more bounding boxes of about the same height that are imperfections. In this case this algorithm will fail but in practical scenarios, it is highly unlikely.

References

1. dos Santos, R.P., Clemente, G.S., Ren, T.I., & Cavalcanti, G. D. (2009). Text line segmentation based on morphology and histogram projection. *In 2009 10th International Conference on Document Analysis and Recognition*, 651-655. https://doi.org/10.1109/ICDAR.2009.183

2. Wang, J.X., Zhou, W.Z., Xue, J.F., & Liu, X.X. (2010). The research and realization of vehicle license plate character segmentation and recognition technology. *In 2010 International Conference on Wavelet Analysis and Pattern Recognition*, 101-104. https://doi.org/10.1109/ICWAPR.2010.5576426

3. Zin, T.T., Thant, S., Htet, Y., & Tin, P. (2020). Handwritten Characters Segmentation using Projection Approach. *In 2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech),* 107-108.
    a. https://doi.org/10.1109/LifeTech48969.2020.1570625477

4. Praseetha, M., & Deepa, S.S. (2014). Segmentation in Malayalam OCR—Handling broken characters using active contour model. *In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT),* 958-962. https://doi.org/10.1109/ICCICCT.2014.6993097

5. Musaddid, A.T., Bejo, A., & Hidayat, R. (2019). Improvement of Character Segmentation for Indonesian License Plate Recognition Algorithm using CNN. *In 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI),* 279-283. htpps://doi.org/10.1109/ISRITI48646.2019.9034614

6. Lee, J., Bang, J., & Yang, S.I. (2017). Object detection with sliding window in images including multiple similar objects. *In 2017 international conference on information and communication technology convergence (ICTC),* 803-806.
    a. https://doi.org/10.1109/ICTC.2017.8190786

7. Xiong, C., Chen, L., & Pang, Y. (2010). An adaptive bilateral filtering algorithm and its application in edge detection. *In 2010 International Conference on Measuring Technology and Mechatronics Automation, 1,* 440-443.
    a. https://doi.org/10.1109/ICMTMA.2010.41

8. Deivalakshmi, S., Sarath, S., & Palanisamy, P. (2011). Detection and removal of salt and pepper noise in images by improved median filter. *In 2011 IEEE Recent Advances in Intelligent Computational Systems,* 363-368.
    a. https://doi.org/10.1109/RAICS.2011.6069335

9. Kim, Y.W., Oh, A.R., & Krishna, A.V. (2019). Analyzing the performance of canny edge detection on interpolated Images. *In 2019 International Conference on Information and Communication Technology Convergence (ICTC),* 726-730. https://doi.org/10.1109/ICTC46691.2019.8939595

10. Zhang, G.L., Ge, L.L., Yang, Y.N., Liu, Y.Q., & Sun, K.X. (2019). Fused Confidence for Scene Text Detection via Intersection-over-Union. *In 2019 IEEE 19th International Conference on Communication Technology (ICCT),* 1540-1543. https://doi.org/10.1109/ICCT46805.2019.8947307

11. Hongyao, D., & Xiuli, S. (2009). License plate characters segmentation using projection and template matching. *In 2009 International Conference on Information Technology and Computer Science, 1,* 534-537. https://doi.org/10.1109/ITCS.2009.116

12. Maheswari, K.U., & Govindarajan, S. (2019). Hybridisation of oppositional centre-based genetic algorithms for resource allocation in cloud. *International Journal of Networking and Virtual Organisations, 21*(3), 307-325.

13. Umamaheswari, K.M., Chanana, T., & Fernandes, K. (2020). Meme Chat: An Innovative Social Media Platform for Content Monetization & Corporate Outreach Using OCR & NLP. *Journal of Critical Reviews, 7*(4), 206-208.

14. Maheswari, K.U., Roy, S., & Govindarajan, S. (2017). Hybrid Green Scheduling Algorithm using Genetic Algorithm and Particle Swarm Optimization Algorithm in IAAS Cloud. *ARPN Journal of Engineering and Applied Sciences, 12*(12), 3762-3766.