Various License Plate Detection and Recognition Methods using Computer Vision and Machine Learning

Spandan Joshi[1*] and *Mehul* Parikh[2]

[1]Research Scholar, Information Technology Department, L. D. College of Engineering, Ahmedabad, Gujarat, India
[2]Associate Professor, Information Technology Department, L. D. College of Engineering, Ahmedabad, Gujarat, India

Abstract - With the increasing advancements in the technology, our lives have become significantly more convenient. We now have automated many things. One example of such things is the automated number plate recognition system. There are many ways to perform the ANPR (Automatic Number Plate Recognition). Performing ANPR in wild still remains a big challenge. This review focuses on some techniques that have tried to overcome this challenge.

1. Introduction

Number plate recognition is a simple application of computer vision and optical character recognition, yet it is a challenge even after such advancements. The reason behind number plate recognition being so challenging even today is that it is really hard to perform ANPR in the wild. This is due to reasons such as imperfect lighting, noise, damaged number plates etc. This review presents various number plate recognition methods developed using image processing and character recognition algorithms that have lesser error than the traditional methods.

2. Overview

Y. Lee et al. [1] focused on detection of license plates in wild (uncontrolled scenarios). They used deep learning techniques to perform ANPR. This approach uses a network created by combining the ResNet50 and the FPN (Feature Pyramid Network). Application Oriented License Plate (AOLP) dataset has been used by them. This approach has achieved many good and challenging detections and achieved more than 99% of accuracy, outperforming its predecessors.

*Corresponding author: joshispandan912k@gmail.com

I. El-Shal et al. [2] present a method for recognition of license plate using General Adversarial Network (GAN) and YOLO v5. This technique uses GAN to enhance the resolution of the low-resolution input images. After using the super resolution enhancement, yolo v5 is used to recognize the characters of the license plate. This technique has been tried on the AOLP dataset and the car plate dataset as well. This approach has also outperformed its predecessors and achieved an accuracy of more than 95%.

V. Khare et al. [3] present a method for license plate recognition using stroke pair width detection, Laplacian edge detection, neural network, principle component analysis and character reconstruction techniques. This technique also deals with problems such as over segmentation and under segmentation during character recognition. This proposed approach was tried on many datasets and it achieved better accuracy than its traditional predecessors in most of them.

J. Zhang et al. [4] present a method for detection of license plates by using Recurrent Convolutional Neural Networks (R-CNN) and Bidirectional Long Short Term Memory (BLSTM). They used this approach on license plate datasets from Caltech University and Sun Yat-Sen University, along with another dataset collected by them. All these datasets contain high resolution images of licence plates in natural environments. This method is proven better than its predecessors in terms of accuracy. Also, even obstructed, defaced license plates have been detected successfully by this method. They have achieved more than 98% accuracy in Taiwanese and US license plates.

Q. Huang et al. [5] present a method for detection and recognition of mixed types of license plates. In this model, two fully convolutional one-stage object detectors are employed to detect bounding boxes and classify LPs and characters. They have used the HZM multi style dataset, PKU dataset and the AOLP dataset. This method has achieved more than 99% accuracy.

3. Methodology

3.1 License Plate Detection via information maximization

3.1.1 Tools used:

ResNet50:

ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

FPN:

FPN stands for Feature Pyramid Network. It combines low-resolution, semantically strong features with high-resolution, semantically weak features via a top-down pathway and lateral connections. This feature pyramid that has rich semantics at all levels and is built quickly from a single input image scale, thereby without sacrificing representational power, speed, or memory.

AOLP Dataset:

The application-oriented license plate (AOLP) benchmark database has 2049 images of Taiwan license plates. This database is categorized into three subsets: access control (AC) with 681 samples, traffic law enforcement (LE) with 757 samples, and road patrol (RP) with 611 samples. AC refers to the cases that a vehicle passes a fixed passage with a lower speed or full stop. This is the easiest situation. The images are captured under different illuminations and different weather conditions. LE refers to the cases that a vehicle violates traffic laws and is captured by roadside camera. The backgrounds are really cluttered, with road sign and multiple plates in one image. RP refers to the cases that the camera is held on a patrolling vehicle, and the images are taken with arbitrary viewpoints and distances.

3.1.2 Method:

Table 1: symbols used in the diagram [1]

Symbol	Description
f and θ_f	The backbone network and its network parameter.
g and θ_g	The LP detection sub-network and its network parameter.
h and θ_h	The non-LP detection sub-network and its network parameter.
S and l_S	The sharing block and its output.

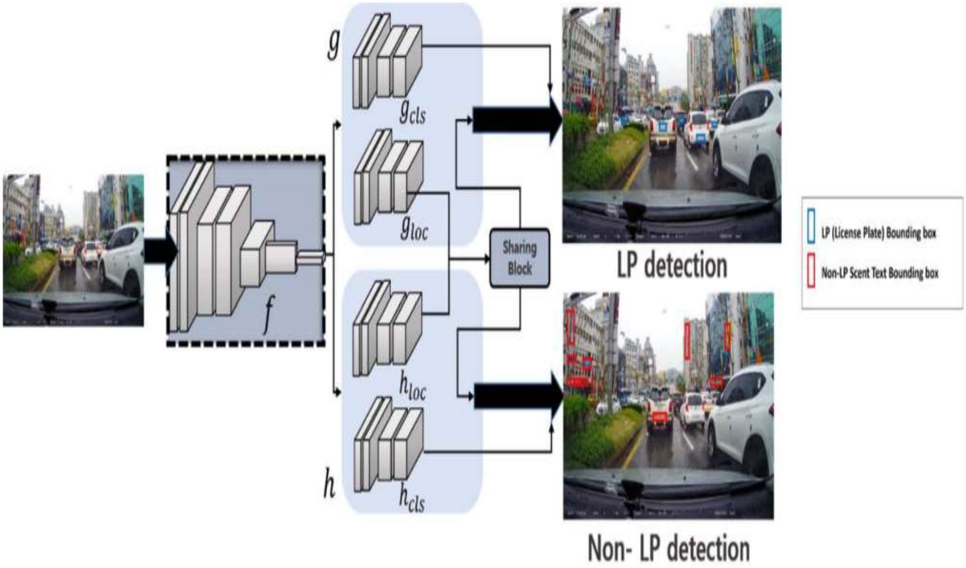
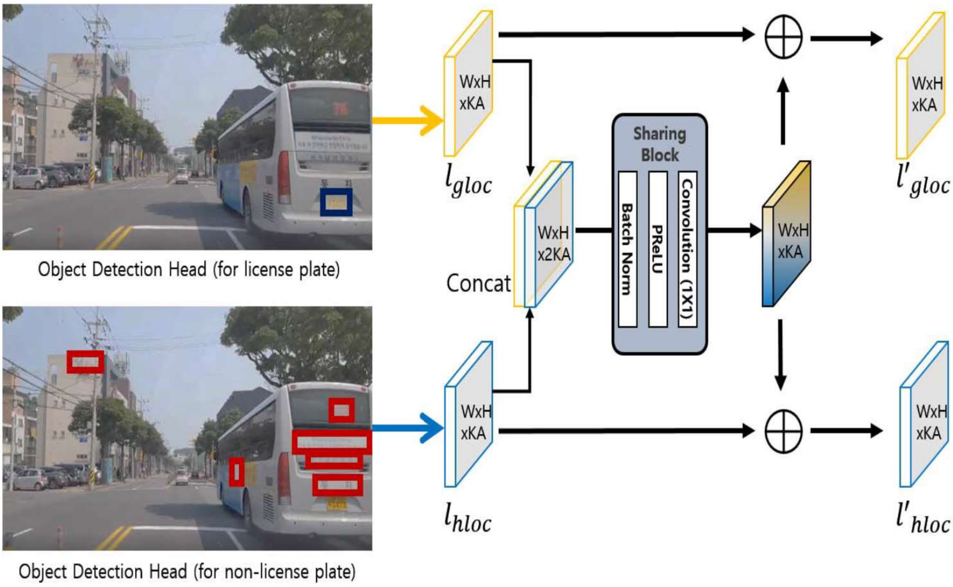


Fig. 1: Workflow of Proposed Model [1]

As we can see in the figure 1, the structure of the proposed model contains a network created by combining the ResNet50 and Feature Pyramid Network, which further detects objects (License plates and non license plate objects). This detection process is explained below:



Object Detection Head (for license plate)

Object Detection Head (for non-license plate)

Fig. 2: Workflow of Detection Process [1]

This model has achieved some very challenging detections as well. Some of which are shown below:

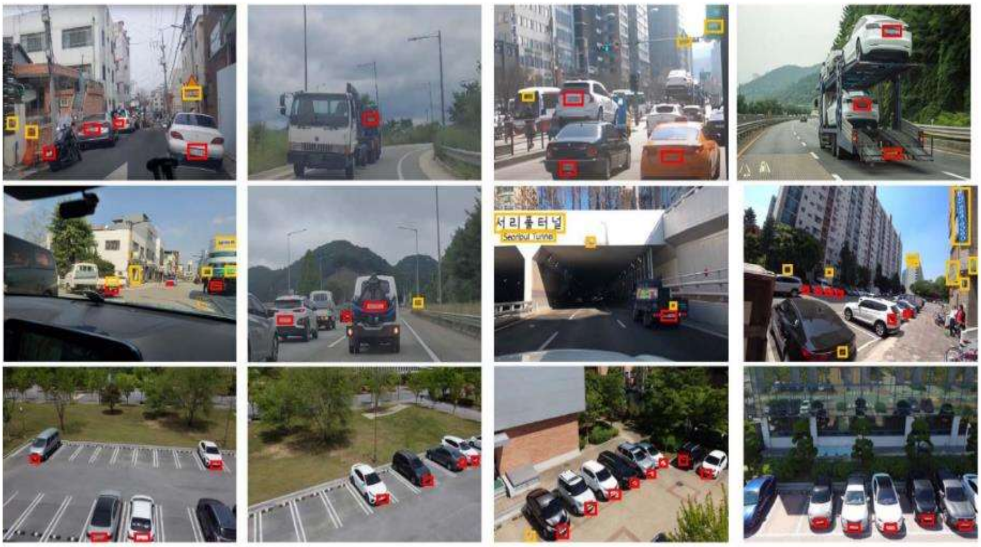


Fig. 3: Some Challenging Detections, Red bounding boxes present number plates and Yellow bounding boxes present non number plate objects) [1]

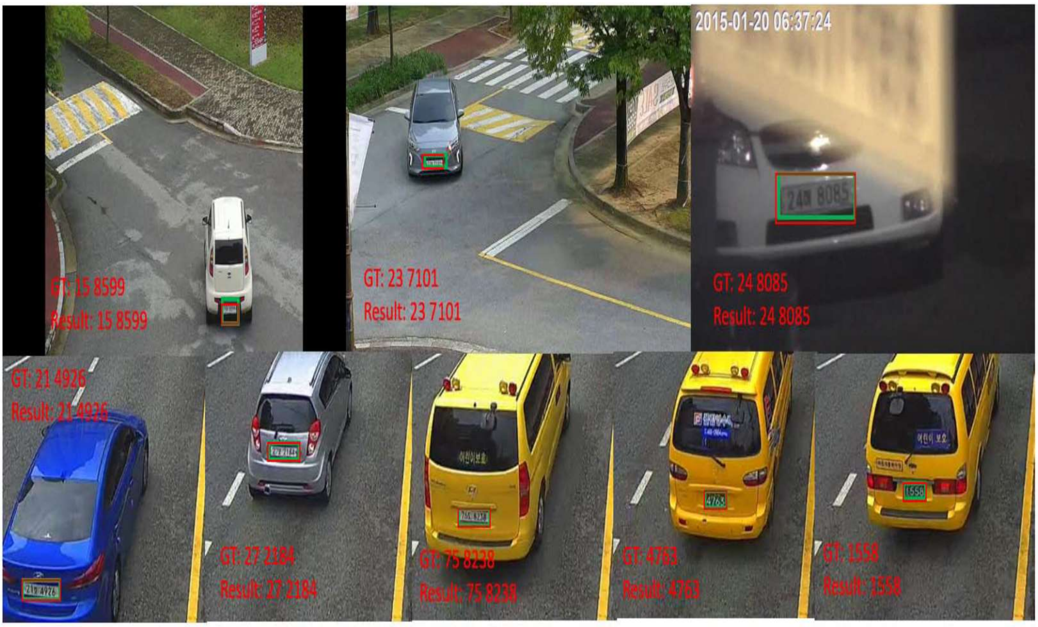


Fig. 4: Some License Plate Recognition Results) [1]

This method has outperformed its predecessors, some results are shown below:

Table 2: Results Comparison of Different Methods on AOLP Dataset [1]

Method/Subset	**AC**		**LE**		**RP**	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)	Precision (%)	Recall (%)
Hsu et al., [43]	91	96	91	95	91	94
Li. et al., [39]	98.53	98.38	97.75	97.62	95.28	95.58
Selmi et al., [57]	92.6	96.8	93.5	93.5	92.9	96.2
Rafique et al., [20]	-	98.09	-	-	-	89.03
Xie et al., [19]	99.51	99.51	99.43	99.43	99.46	99.46
Li eta l., [58]	-	99.12	-	-	-	98.2
Bjorklund et al., [59]	100	99.3	99.8	99.8	99.8	99.0
Selmi et al., [60]	99.3	99.4	99.2	99.2	98.9	98.8
Proposed	99.71	99.8	99.9	99.47	99.83	99.46

As we can see, the method is clearly very effective and has surpassed its predecessors greatly. However, the method still faces some problems sometimes during cases like the challenging discussions mentioned above. Overall the method is very good.

3.2 License Plate Image Analysis Empowered by General Adversarial Neural Networks [2]

3.2.1 Tools used:

General Adversarial Network:

GAN is a framework represented by Ian Goodfellow. It addresses the issue of unsupervised learning by training two deep neural networks, known as Generator (G) and Discriminator (D), that compete and collaborate to estimate generative models using adversarial methods. The game of zero-sum is the fundamental concept of the GAN model. The GAN trains the network to achieve Nash equilibrium (In game theory, the Nash equilibrium, named after the mathematician John Nash, is the most common way to define the solution of a non-cooperative game involving two or more players. In a Nash equilibrium, each player is assumed to know the equilibrium strategies of the other players, and no one has anything to gain by changing only one's own strategy.)

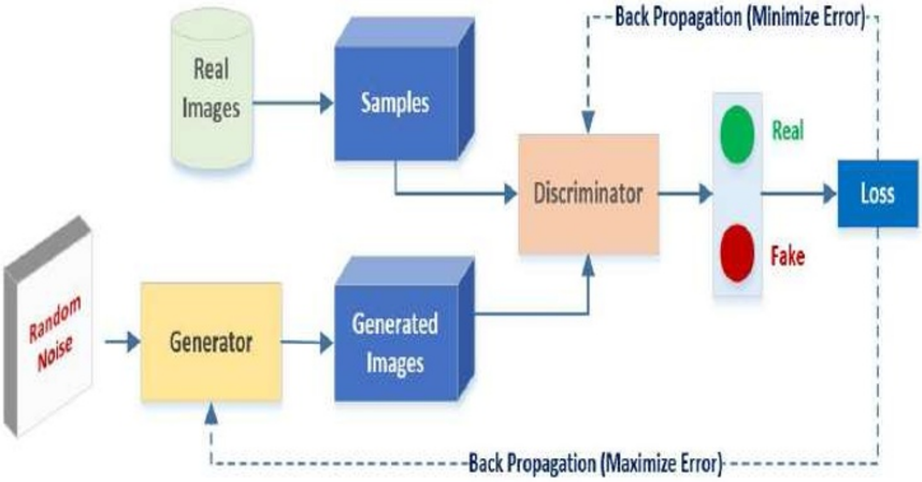


Fig. 5: Primary Architecture of General Adversarial Networks [2]

AOLP Dataset:

Car Plate Dataset:

The training dataset includes 1162 HR images, divided into training and validation sets. The Car Plate Dataset 856 samples for a single object (LP), 172 samples for multiple LP in a single image, and the remaining samples have a wider range of variation conditions, like partially blocked or spatial cases.

3.2.2 Method:

As we can see in the figure 6, the system works in this manner, firstly, the low resolution input images are fed to the GAN. Then a super resolution image is obtained.

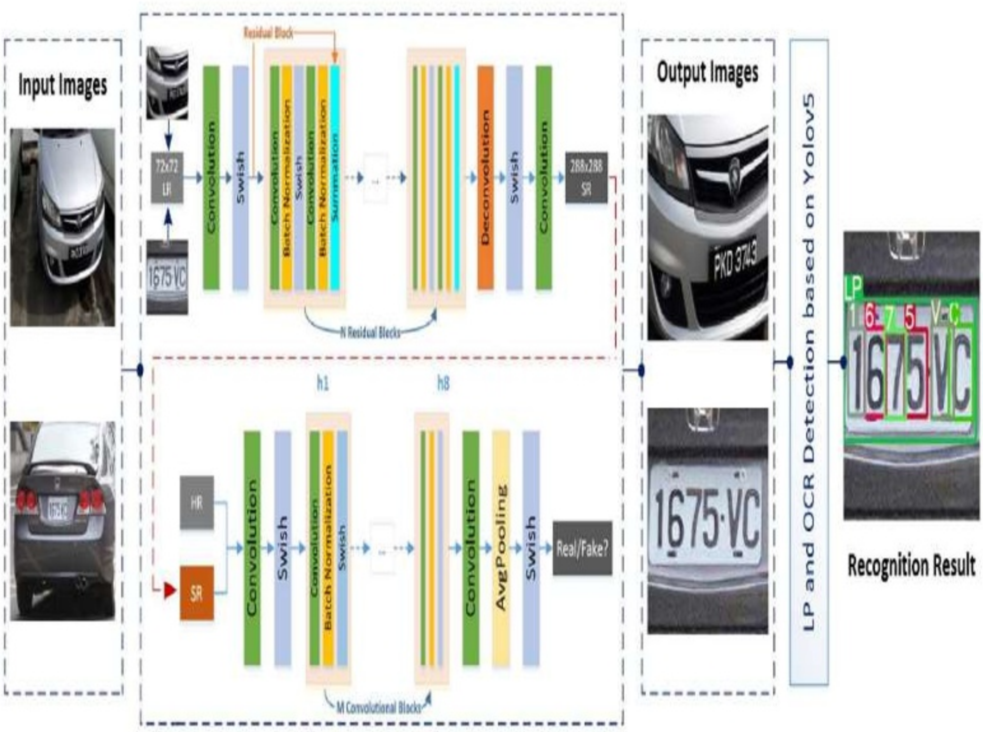


Fig. 6: Overview of the Proposed System [2]

A high resolution image of license plate is received as the output of this network and it is then fed to the License Plate detector and Optical Character Recognition models based on Yolov5. We then get our final detection image as an output that has recognized license plate and characters. This method has performed better than its predecessors and competitors. Some comparison results are given.

Table 3: Results Comparison of Different Methods [2]

Model	**LR Images**	**DIP**	**SRCNN**	**SRGAN**	**SRGAN-TV**
Accuracy	18.8%	43%	93.5%	95.4%	93.2%

As we can see in Table 3, the proposed Super Resolution General Adversarial Network (SRGAN) performs better than its competitors. Overall this is a very nice method but it will face problems when faced with ‘deformed’ number plates (unclear, damaged number plates). The model can be improvised to deal with them also.

3.3 A Novel Character Segmentation and Reconstruction Approach for License Plate Recognition [3]

3.3.1 Method:

This paper presents a method for detection and recognition of license plates using the character reconstruction approach. Firstly, partial reconstruction is performed using Laplacian edge detection and stroke width. After this, PCA and Major Axis are used to determine the angular information of characters. Then the full character reconstruction and recognition is performed.

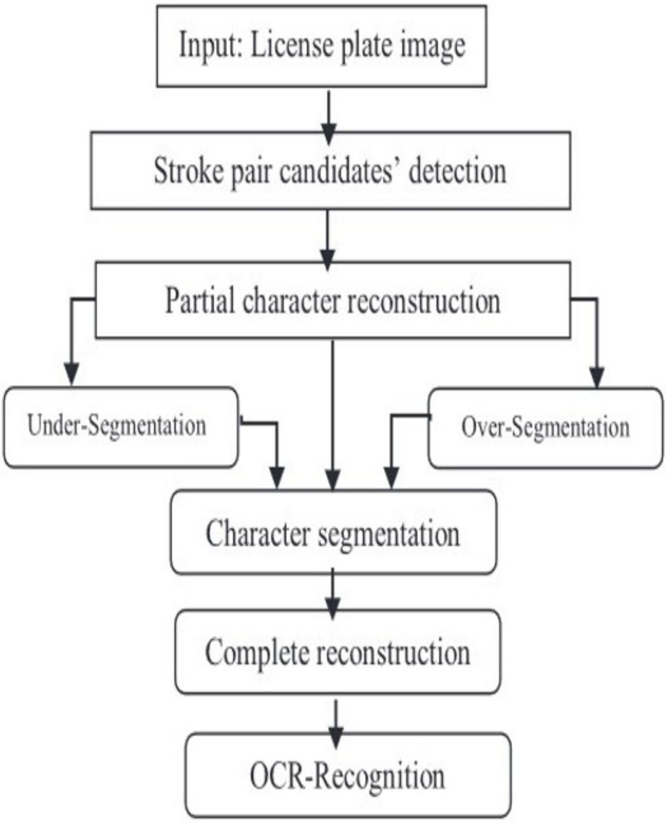
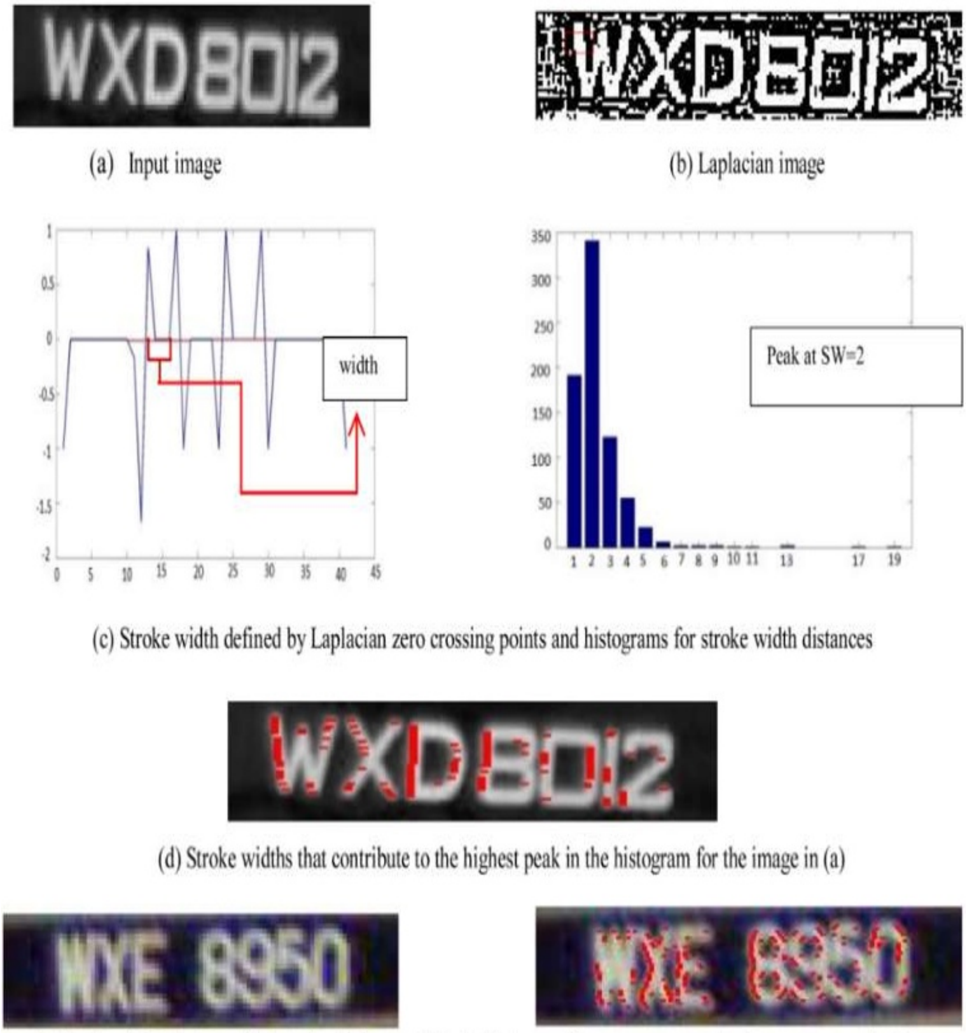


Fig. 7: Workflow of the Proposed Method [3]

As we can see in the figure 7, the license plate image is given as input, then the stroke pair width detection is performed. After that, partial character reconstruction is done, any problems such as over or under segmentation are dealt with and a proper character segmentation is performed. After that a complete reconstruction is done. And finally, Optical Character Recognition is performed.



(a) Input image

(b) Laplacian image

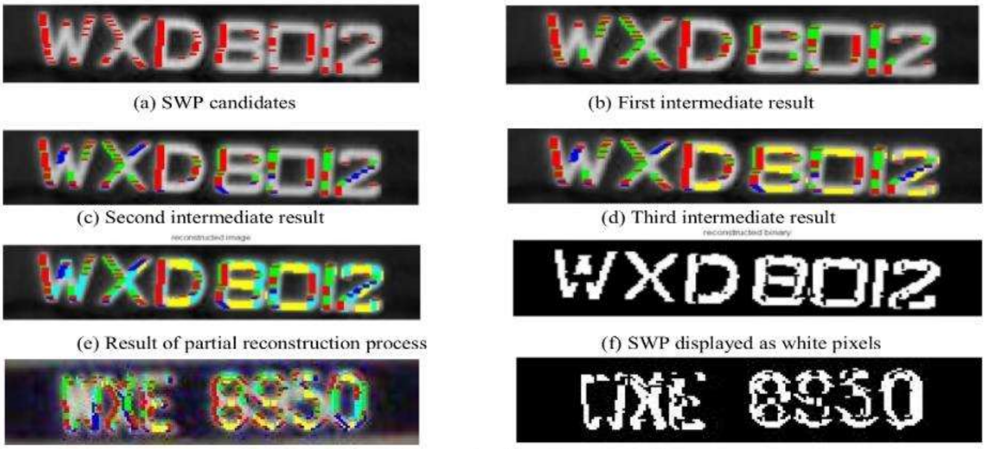
(c) Stroke width defined by Laplacian zero crossing points and histograms for stroke width distances

(d) Stroke widths that contribute to the highest peak in the histogram for the image in (a)

(e) Stroke widths that contribute to the highest peak in the histogram for one more complex image compared to the image in (a).

Fig. 8: Stroke Width Pair Candidate Detection [3]

As we can see in Figure 8, an example of the stroke width pair candidate detection is shown. Laplacian filter is an edge detector used to compute the second derivatives of an image, measuring the rate at which the first derivatives change. This determines if a change in adjacent pixel values is from an edge or continuous progression. The stroke-width property is used to set the width of a border in a SVG (Scalable Vector Graphics) shape. This property can only be applied to elements that have a shape or are text content elements. After this, the partial character reconstruction process is performed. After that, with the use of Principal Component Analysis and Major Axis, any over or under segmentation problems are dealt with and a full character reconstruction is performed.



(a) SWP candidates

(b) First intermediate result

(c) Second intermediate result

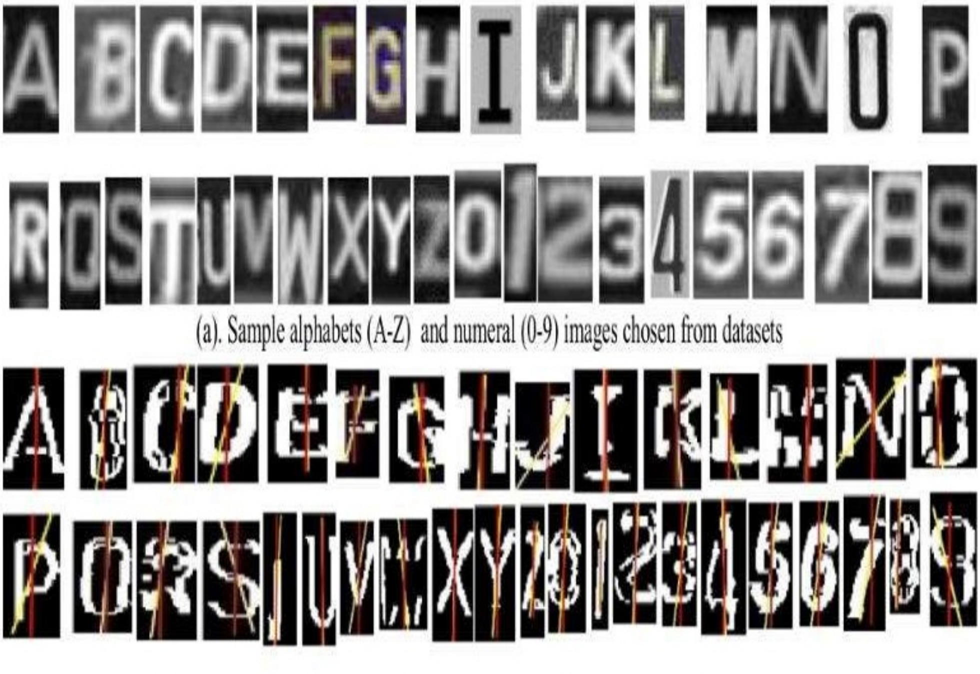
(d) Third intermediate result

(e) Result of partial reconstruction process

(f) SWP displayed as white pixels

(g). Result of partial reconstruction of a complex image

Fig. 9: Partial Character Reconstruction [3]

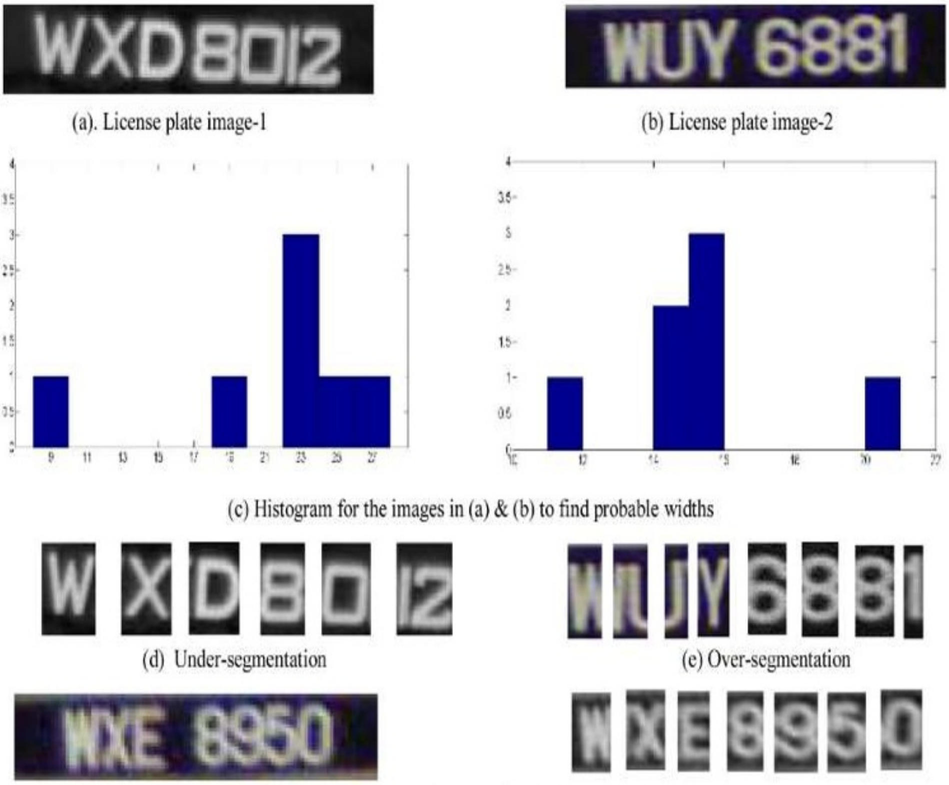


(a). Sample alphabets (A-Z) and numeral (0-9) images chosen from datasets

(b). PCA and MA axes for partial reconstruction results of alphabets and numerals in (a).

Fig. 10: Alphanumeric Dataset and the angle information provided by PCA and MA [3]

As we can see in Figure 10, The Major Axis is represented by a red colour and the Principal Component Analysis axis is represented by a yellow colour.



(a). License plate image-1

(b) License plate image-2

(c) Histogram for the images in (a) & (b) to find probable widths

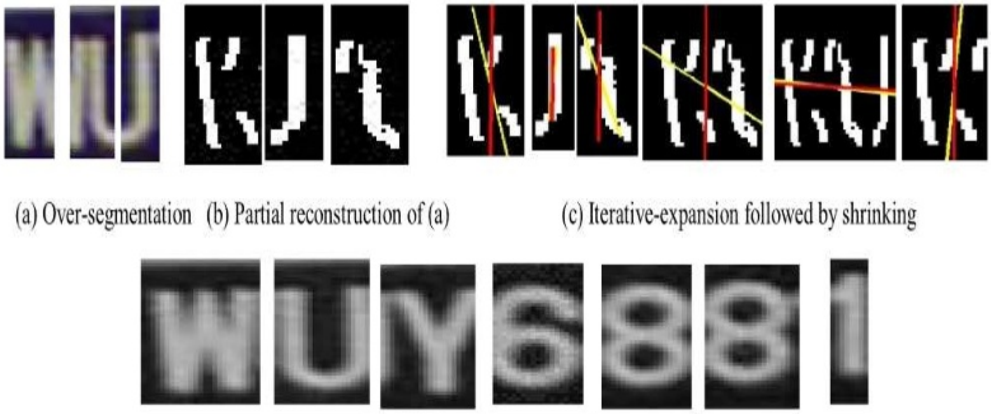
(d) Under-segmentation

(e) Over-segmentation

(f) Character segmentation result for the image in Fig. 3(e)

Fig. 11: Character Segmentation [3]

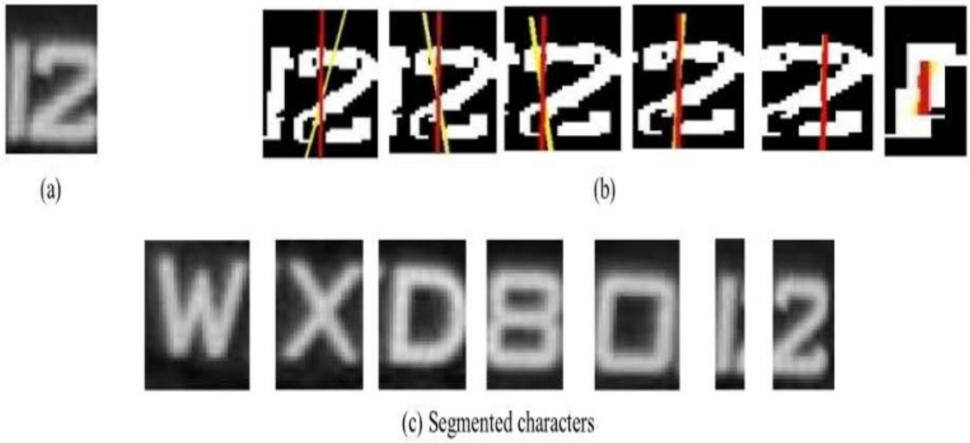
In Figure 11, character segmentation is shown, we can also see the over and under segmentation problems. These are dealt with Iterative Expansion and Iterative Shrinking Approaches respectively.



(a) Over-segmentation (b) Partial reconstruction of (a) (c) Iterative-expansion followed by shrinking

(d) Characters segmented

Fig. 12: Iterative Expansion Process in case of Over Segmentation [3]



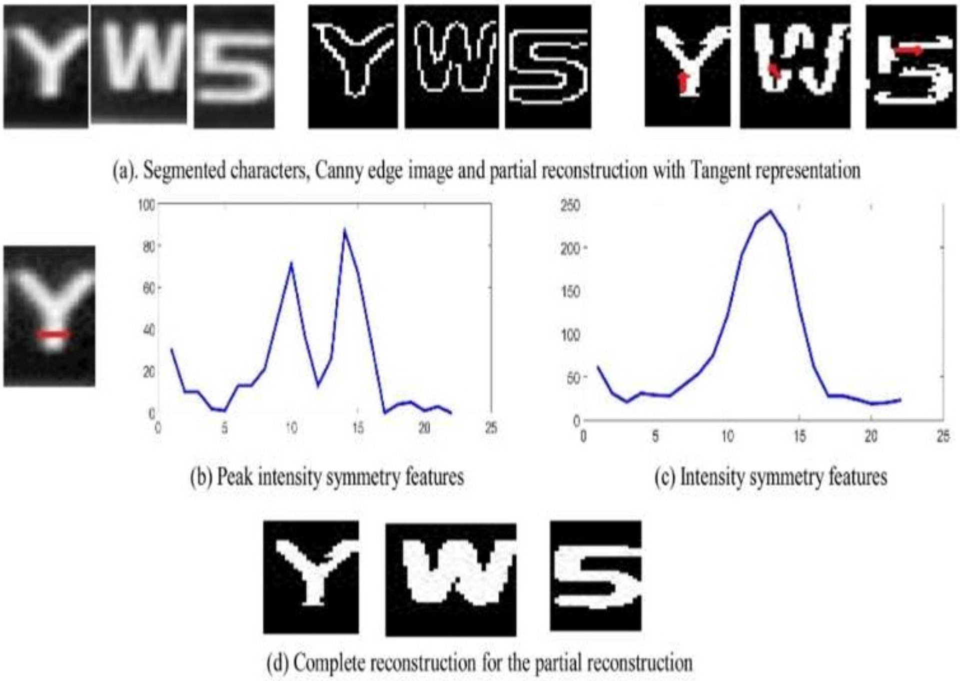
(a)

(b)

(c) Segmented characters

Fig. 13: Iterative Shrinking Process in case of Under Segmentation [3]

As we can see in Figures 12 and 13, the Iterative Expansion and Iterative Shrinking Approaches are used to deal with over and under segmentation cases. After this, a complete reconstruction is performed in the gray domain.



(a). Segmented characters, Canny edge image and partial reconstruction with Tangent representation

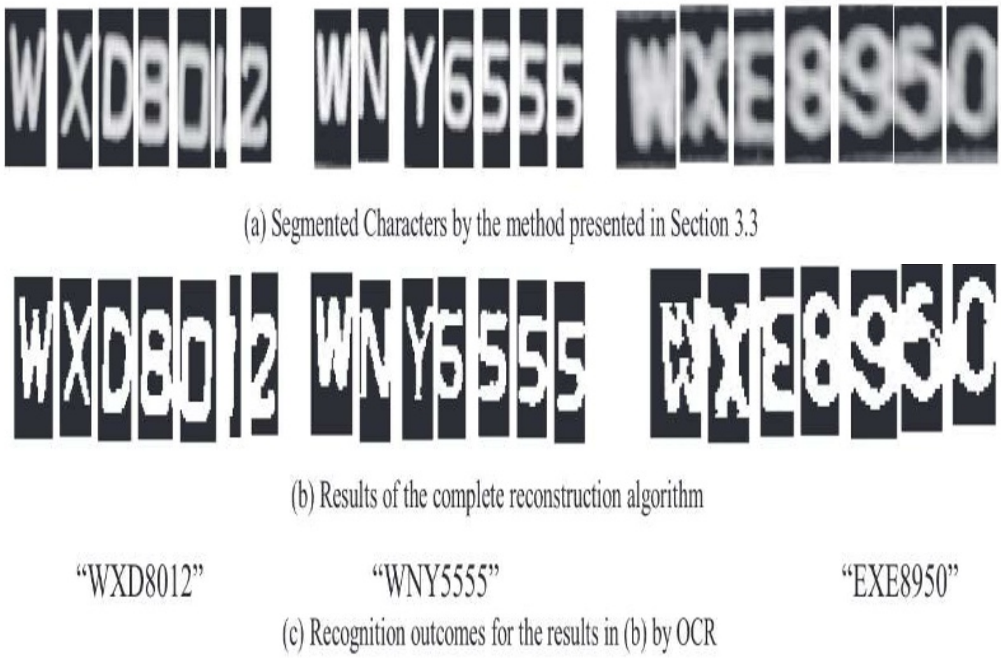
(b) Peak intensity symmetry features

(c) Intensity symmetry features

(d) Complete reconstruction for the partial reconstruction

Fig. 14: Complete Reconstruction in Gray Domain [3]

As we can see in Figure 14, the complete reconstruction of partial reconstruction is performed in the gray domain. After this, the reconstructed image is given as input to the Optical Character Recognizer System.



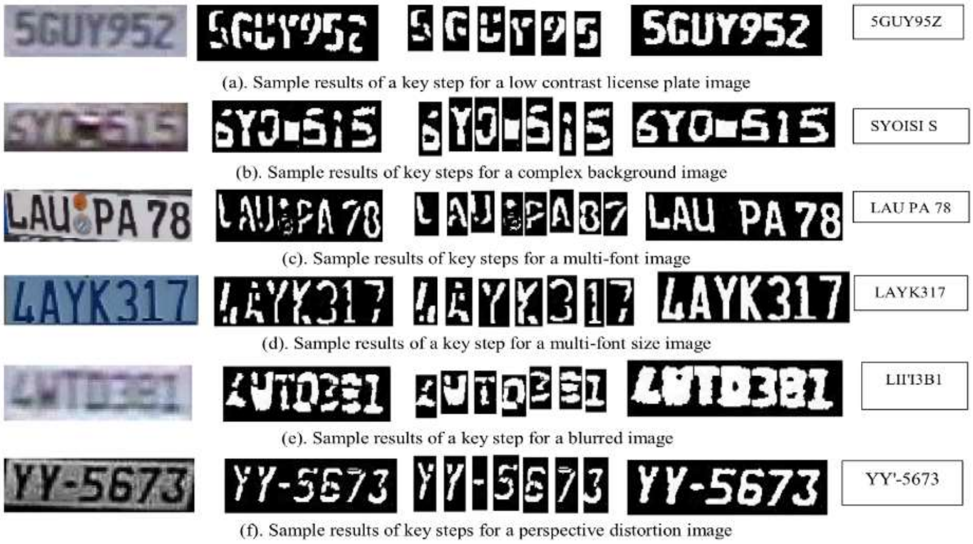
(a) Segmented Characters by the method presented in Section 3.3

(b) Results of the complete reconstruction algorithm

(c) Recognition outcomes for the results in (b) by OCR

Fig. 15: Recognition Results

As we can see in the Figure 15, OCR is performed as the final step. Overall this approach is good but as shown in Figure 16, the OCR is problematic and doesn't give correct results in some cases, especially in the cases of deformed number plates. The proposed method however gave a good competition and performed well in most cases against its competitors on many datasets as shown in Table 4.



(a). Sample results of a key step for a low contrast license plate image

(b). Sample results of key steps for a complex background image

(c). Sample results of key steps for a multi-font image

(d). Sample results of a key step for a multi-font size image

(e). Sample results of a key step for a blurred image

(f). Sample results of key steps for a perspective distortion image

Fig. 16: Some Problematic Cases of OCR

Table 4: Results of Comparison of Different Methods on Different Datasets [3]

Datasets	Measures	Canny	Anagnostopoutos er al., 2006	ZJ1ou er al., 2013	Tian er al., 2015a	Bulan ec al., 2017	Silva and Jung (2018)	Lin ec al., 2018	Proposed
MIMOS	RR	58.7	63.2	47.4	57.6	86.3	89.3	78.3	88.4
	RP	54.J	64.7	52.J	59.7	82.6	83.2	74.9	84.3
	RF	56.4	63.8	50.3	58.6	84.5	86.2	76.6	86.3
Medialab	RR	59.3	64.7	52.4	61.2	83.7	86.4	75.6	82.3
	Rf>	52.4	66.9	56.8	62.7	75.3	82.3	71.9	79.3
	RF	55.3	65.7	54.6	61.6	79.5	84.3	73.7	81.3
UCSD	RR	29.2	42.3	47.2	44.9	52.4	58.3	51.7	65.7
	RP	32.7	44.7	48.1	46.2	47.4	55.3	49.5	62.1
	RF	31.3	43.6	47.6	45.5	49.9	56.8	50.6	63.9
Uninsubria	RR	62.4	65.3	68.3	64.3	76.4	78.4	77.1	78.7
	RP	66.7	68.7	69.4	69.4	72.4	75.3	77.4	80.3
	RF	*64.8*	66.9	68.8	*67:1*	74.4	76.8	77.2	79.5
ICDAR 2015 Video	RR	*662*	68.9	71.8	72.6	83.4	86.4	84.3	78.6
	RP	61.3	75.7	72.3	72.7	81.3	80.3	78.9	73.4
	RF	63.7	72.7	72.1	72.6	82.3	83.3	81.6	76.2
YVT Video	RR	72.4	72.9	66.9	71.4	83.4	85.9	84.9	78.3
	RP	65.3	77.8	70.3	74.8	79.2	81.4	78.8	82.6
	RF	68.4	75.8	68.7	72.9	81.3	83.G	81.8	80.5
ICDAR 2013 Video	RR	*682*	78.7	71.3	74.9	81.6	83.2	79.5	83.7
	Rf>	61.3	79.3	68.9	71.3	80.4	81.5	78.4	84.2
	RF	65.7	78.5	69.8	72.8	81	82.3	78.9	83.5
ICDAR 2015 Scene	RR	66.8	77.3	72.1	65.3	82.3	85.7	80.2	80.3
	RP	67.3	72.1	74.3	62.1	81.4	84.4	80.3	82.1
	RF	66.9	75.2	73.6	64.4	81.8	85	80.2	81.5
ICDAR 2013 Scene	RR	59.3	72.3	71.3	65.6	83.1	86.1	81.4	78.3
	RP	56.3	72.4	68.7	*643*	81.5	84.3	80.4	*732*
	RF	58.6	72.3	70.1	64.9	82.3	85.2	80.9	75.8
SVT Scene	RR	58.3	76.4	71.4	66.3	78.2	79.3	76.4	80.4
	RP	59.7	78.3	74.7	67.2	77.3	76.3	74.8	81.G
	RF	58.6	77.9	73.1	66.8	77.7	77.8	75.6	81.0
MSRA-TD-500	RR	64.3	73.9	75.9	72.4	78.4	81.3	74.9	82.4
	RP	65.8	76.4	74.3	77.3	74.8	80.4	73.8	81.6
	RF	64.9	75.9	75.1	75.4	76.6	80.8	74.3	81.9
Only Challenged Images	RR	58.7	51.9	47.4	57.6	54.8	57.6	55.9	62.9
	RP	54.3	52.3	52.3	59.7	51.7	56.6	51.4	65.7
	RF	56.5	52.1	49.8	58.6	53.2	57.1	53.6	64.3

3.4 License Plate Localization in Unconstrained Scenes Using a Two-Stage CNN-RNN [4]

3.4.1 Datasets:

Table 5: Test Set of License Plates Images [4]

Country	**Canada**	**Malaysia**	**Taiwan**	**Thailand**	**US**
Number	135	150	112	75	127
Country	**Hong Kong**	**Europe**	**Croatia**	**Chinese**	
Number	141	718	200	1680	

3.4.2 Method:

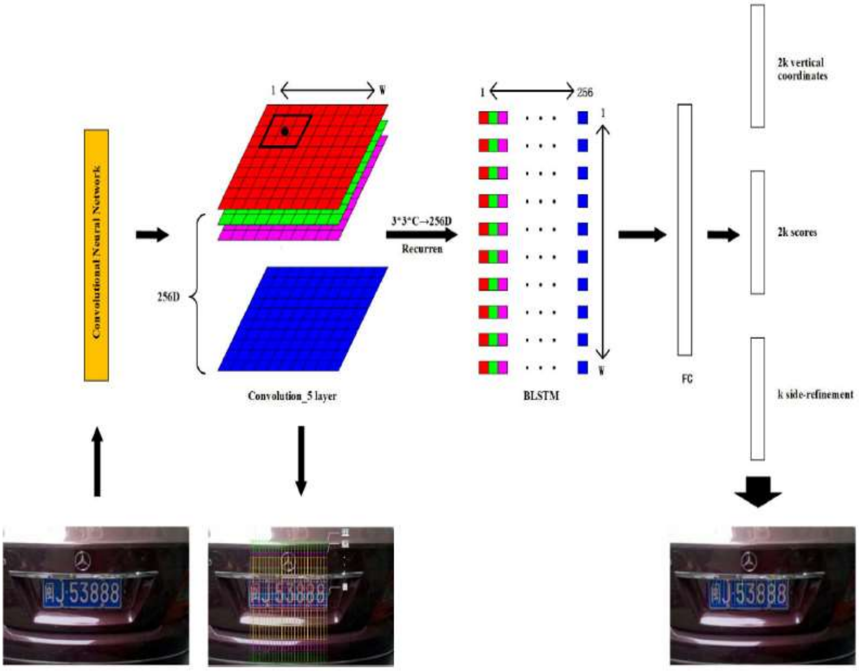


Fig. 17: The Workflow of the Proposed Method [4]

As we can see in Figure 17, they densely slide a 3 × 3 spatial window on the convolution_5 layer. The sequential windows in each row are recurrently connected by a Bi-directional LSTM (BLSTM), where the convolutional feature (3 × 3 × C) of each window is used as input of the 256D (D represents dimension) BLSTM. The BLSTM layer is connected to a 512D fully-connected layer, followed by the output layer, which jointly predicts confidence scores, y-axis coordinates and boundary-refinement offsets of k anchors. Finally, the segmented portions we see as the final output in Figure 17 are connected together. This method is very effective in detection of license plates. It manages to detect even obscured, blurred, obstructed, damaged number plates.

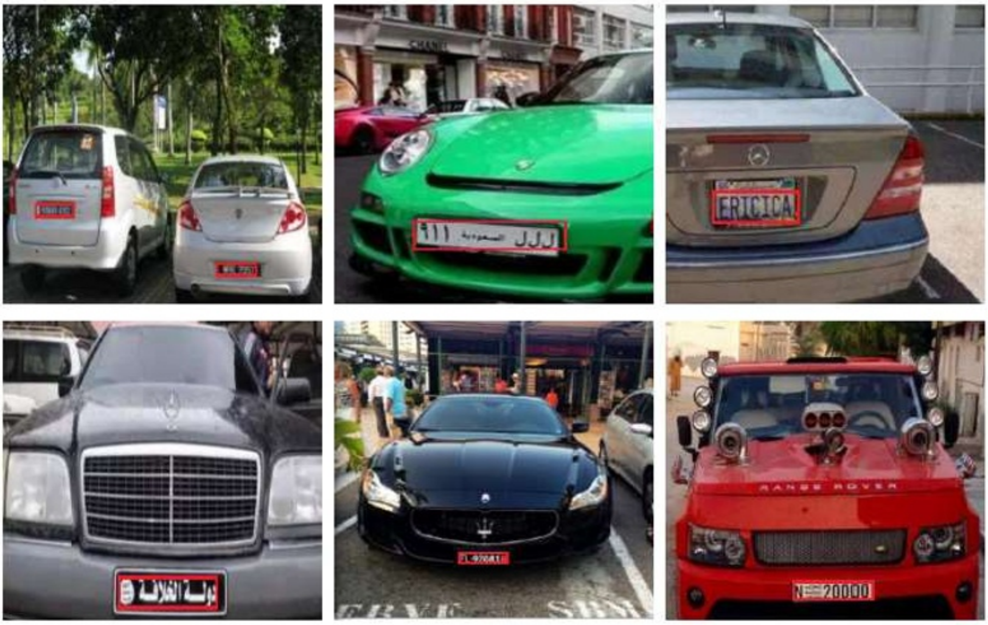


Fig. 18: Some Detection Results [4]

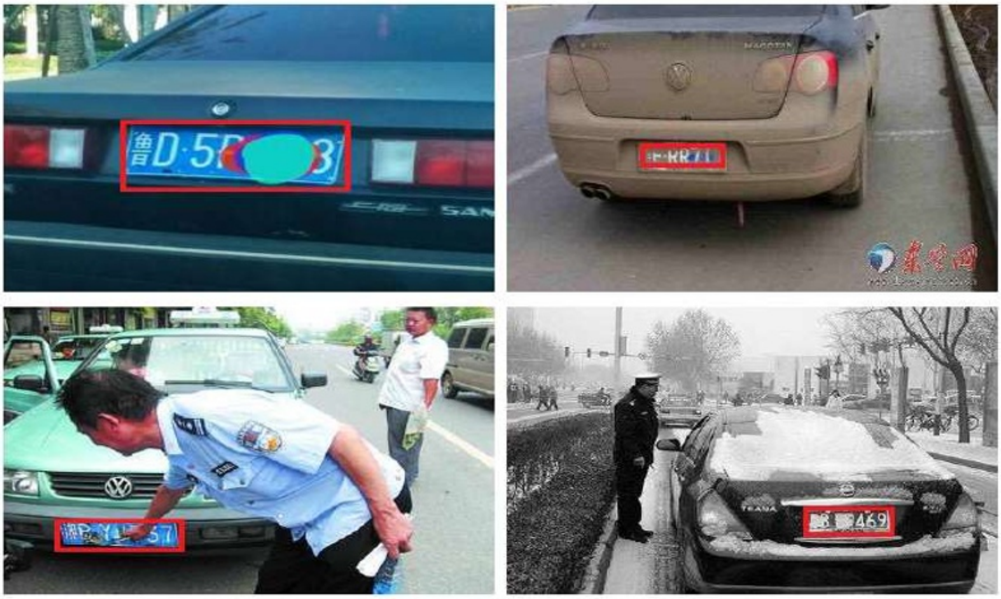


Fig. 19: Detection Results on Obscured/Damaged Number Plates [4]

As we can see in Figures 18 and 19, excellent detection results have been achieved by the proposed method. The method performs very well against its competitors as well. Some results are given in the Table 6.

Table 6: Precision Results of Comparison of Different Algorithms for License Plates of Different Countries [4]

Method	Canada	Malaysia	Taiwan	Thailand	Hong Kong	Europe	Croatia	us	Average	Time (ms)
Faster R-CNN+ZF	48.09%	67.69%	63.53%	72.60%	85.00%	68.74%	70.37%	61.38%	66.89%	61
Faster R-CNN+_CNN_M_1024	57.66%	72.89%	79.39%	87.15%	91.30%	74.42%	75.69%	69.80%	75.93%	125
Faster R-CNN+VGG 16	51.26%	69.85%	69.86%	83.12%	90.39%	71.36%	74.63%	73.71	72.29%	176
Our method	**94.49%**	**96.45%**	**97.96%**	**96.31%**	**98.71%**	**97.47 %**	**97.83%**	**98.41 %**	**97.19%**	113

3.5 A Single Neural Network for Mixed Style License Plate Detection and Recognition [5]

3.5.1 Datasets:

HZM Dataset:

The HZM multi-style dataset includes three styles of LPs: Mainland China LP, Hong Kong LP with white background, and Macao LP with black background, which is a private dataset including 1376 images collecting from the real-world system running on the Hong Kong-Zhuhai-Macao Bridge. The resolution of these images is 1190x500 pixels.

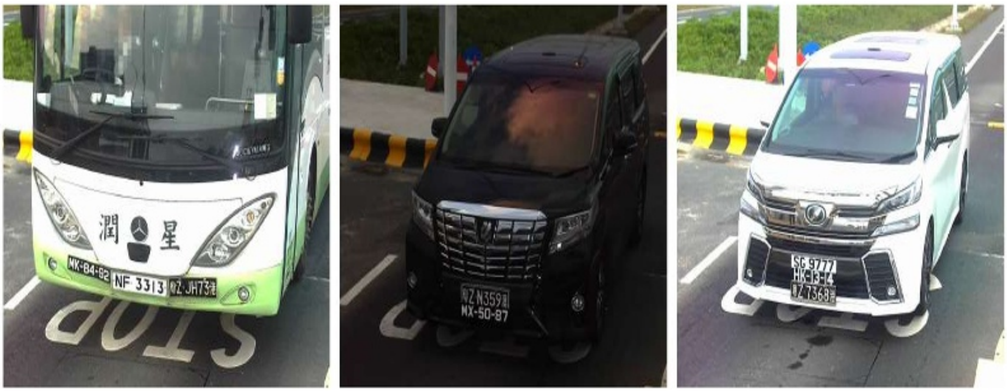


Fig. 20: Examples Images from HZM Dataset [5]

3.5.2 Methods:

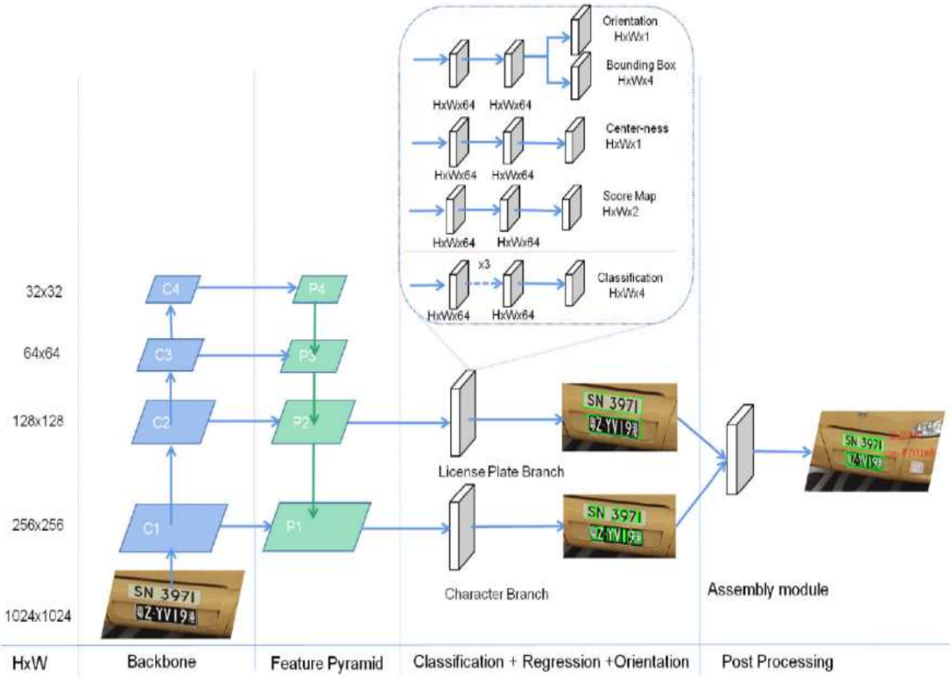


Fig. 21: Workflow of the Proposed Method [5]

As we can see in the Figure 21, firstly, the multiple license plate is given as an input to the ALPRNet. Then it is forwarded to the Feature Pyramid. After that, the license plate and character branching are performed. And in the final post processing step, the output of breaching phase is taken as input of the assembly module and the Optical Character Recognition is performed. The model has achieved excellent results. Some of which are shown in Figure 22.

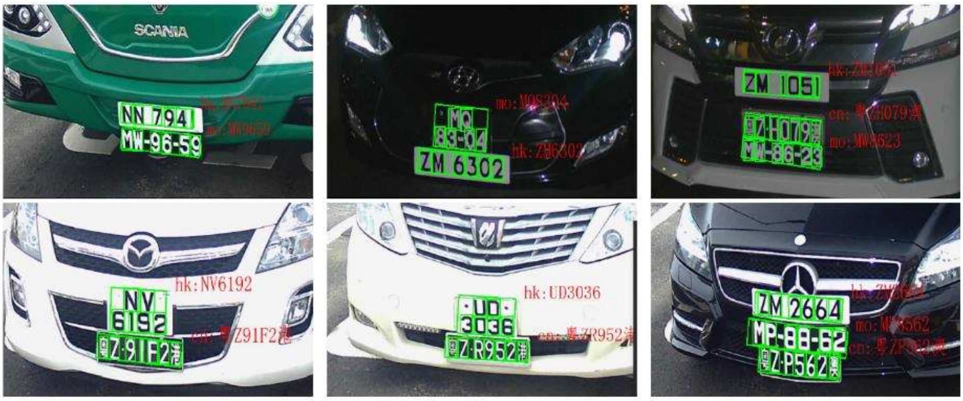


Fig. 22: Detection and Recognition Results [5]

As we can see in Figure 22, the proposed method shows excellent results. The proposed method has also left its predecessors behind in terms of accuracy.

Table 7: Results of Comparison with Different Algorithms [5]

Method	AC (%)		LE (%)		RP (%)	
	LPDA	E2E	LPDA	E2E	LPDA	E2E
Hsu *et al.* [18)	96	-	95	-	94	-
Li *el al.* [25]	98.38	94.85	97.62	94.19	95.58	88.38
Li *et al.* [26]	99.56	95.29	99.34	96.57	98.85	83.63
ALPRNet	**99.82**	**95.78**	**99.66**	**96.62**	**99.50**	**91.58**

4. Conclusion

We studied several methods for the detection and recognition of License Plates. All these methods presented some very good approaches and ways to make ALPR (Automatic License Plate Recognition) more convenient. However, even these methods will face a lot of difficulties when it's a case of recognition of deformed number plates. We still need to improvise our methods further in order to deal with such cases.

References:

1. Y. Lee, J. Jeon, Y. Ko, M. J, IEEE Trans. Intell. Transp. Syst., 23, 14908 - 14921 (2022)
2. I. El-Shal, O. Fahmy, M. Elattar, IEEE Access, 10, 30846 – 30857 (2022)
3. V. Khare, P. Shivkumara, C. Chan, T. Lu, L. Meng, H. Woon, M. Blumenstein, Elsevier Expert Syst. Appl., 131, 219-239 (2019)
4. J. Zhang, Y. Li, t. Li, L. Xun, C. Shan, IEEE Senss. J., 19, 5256-5265 (2019)
5. Q. Huang, Z. Cai, T. Lan, IEEE Access, 9, 21777-21785 (2021)
6. P. Ezhilarasi, S. Rajeshkannan, Kovendan. A K P, T. Sasilatha, K. R. Kayalvizh, Int. J. Eng. Adv. Technol., 9, 22-25 (2020)
7. W. Weihong, T. Jiaoyang, IEEE Access, 8, 91661-91675 (2020)
8. S. Silva, C. Jung, IEEE Trans. Intell. Transp. Syst., 23, 5693-5703 (2021)
9. X. Kong, K. Wang, M. Hou, X. Hao, G. Shen, X. Chen, F. Xia IEEE Trans. Ind. Inform, 17, 8523-8530 (2020)
10. A. Sahoo, Elsevier Mater. Today: Proc., 49, 2982-2988 (2022)
11. R. Chauhan, K. Chauhan, Elsevier ISWA, 15 (2022)
12. C. Henry, S. Ahn, S. Lee, IEEE Access, 8, 35185-35199 (2020)
13. C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, IEEE Trans. Intell. Transp. Syst., 7, 377–392 (2006)
14. J. Jiao, Q. Ye, and Q. Huang, Pattern Recognit., 42, 358–369 (2009)
15. G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung, IEEE Trans. Veh. Technol., 62, 552–561 (2013)
16. H. Li, P. Wang, and C. Shen, IEEE Trans. Intell. Transp. Syst., 20, 1126–1136 (2019)
17. T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, Pattern Recognit., 93, 134–146 (2019)

18. Z. Selmi, M. B. Halima, U. Pal, and M. A. Alimi, Pattern Recognit. Lett., 129, 213–223 (2020)
19. H. Li, P. Wang, and C. Shen, IEEE Trans. Intell. Transp. Syst., 20, 1126–1136, (2019)
20. O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, IEEE Trans. Intell. Transp. Syst., 18, 2351–2363 (2017)
21. M. Zareapoor, H. Zhou, and J. Yang, Neural Comput. Appl., 32, 14521–14531 (2020)
22. Z. Cheng, M. Gadelha, S. Maji, and D. Sheldon, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 5443–5451 (2019)
23. P. Shamsolmoali, M. Zareapoor, E. Granger, H. Zhou, R. Wang, M. E. Celebi, and J. Yang, Inf. Fusion, 72, 126–146 (2021)
24. C. Dong, C. C. Loy, K. He, and X. Tang, IEEE Trans. Pattern Anal. Mach. Intell., 38, 295–307 (2015)
25. T. K. Lai, A. F. Abbas, A. M. Abdu, U. U. Sheikh, M. Mokji, and K. Khalil, in Proc. IEEE 15th Int. Colloq. Signal Process. Appl. (CSPA), 80–85 (2019)
26. S. Tenzin, P. Dorji, B. Subba, and T. Tobgay, in Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), 1–6 (2020)
27. V. Nayak, Int. J. Adv. Trends Comput. Sci. Eng., 9, 3783–3787 (2020)