IJARIIT
INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 2)
Available online at: www.ijariit.com

License plate detection and string conversion using Haar-like cascade classifier

Ashwin B.
ashwinmeena98@gmail.com
Sri Krishna College of Technology, Coimbatore, Tamil Nadu

T. Goutham
gtsgowtham@gmail.com
Sri Krishna College of Technology, Coimbatore, Tamil Nadu

P. Kalpana
p.kalpana@gmail.com
Sri Krishna College of Technology, Coimbatore, Tamil Nadu

ABSTRACT

This paper presents a core module for intelligent transportation based on the method Haar-like cascade classifier for high accuracy license plate detection. Many real-time car license plate detection is reasonable and effective only under certain conditions and assumptions. Therefore a real-time method Haar-like cascade and Tesseract search for Optical Character Recognition (OCR) has been proposed. Using accurate prediction and fast analysis strategy our proposed system can constructively out pass the problems in real-time scenarios. After binge analyzing the system with various inputs to establish that the proposed system is superior to the existing systems in terms of accuracy and time consumption.

Keywords— *Haar-like feature, Optical character recognition, Convolutional neural network, Tesseract, Edge features, Adaboost, Cascading classifiers, Recurrent neural network, Long short term memory*

1. INTRODUCTION

In this modern technologically growing world, the number of automobiles are increasing widely each and every year. This has resulted in violations of many traffic rules, careless accidents and extreme problems. Traditional or physical management of this wide range of automobiles is too difficult which has stimulated the rise of automatic systems to manage automobiles. Specifically, license plate detection can be used to monitor the cars in real-time. This approach has attracted many types of research to develop systems. In future, this automatic car license plate detection can be implemented in various day-to-day scenarios, e.g., unattended toll plaza, parking lot ticket collection in malls, traffic rules violations.

Many problems have to be addressed to develop a state of the art license plate detector system, which includes:

- Plate variations: The plate size, colour, character style may differ from vehicle to vehicle.
- Environment variations: Weather conditions, quality of the input image can be a factor for false detection.
- Image variations: Clarity, resolution, properties may affect the detection process.

The traditional methods involve hand-crafted features such as colour, edge and morphology, which are primarily confined by stringent conditions. Many systems need high in resolution image as input to process the license plate detection, which is the biggest disadvantage when implemented in the real-world scenarios. Real-world scenarios can be quite challenging in the detection process. Thus making traditional methods inefficient to process such a robust method in such complex scenarios.

To avoid such problems people have widely started adopting methods based on CNN, which automatically learn features from the acquired data. But the disadvantage of using a Convolutional Neural Network (CNN) is that the time consumption is higher.[1] To avoid the problem of time consumption and to increase the accuracy in detecting the license plate the method we propose is deep learning based “Haar-like cascade classifier”. Haar-like cascade is superior to the existing CNN based model in factors like time and scenario based on the detecting.

2. RELATED WORK

Many interesting works have been carried out by the community of computer vision for the past few years to overcome the problems faced in license plate detection. The process of license plate detection can roughly be classified into two major methods:

- Image processing methods
- Machine learning methods

2.1 Image Processing Methods

- Region-Based Method: Region-based method the input image is segmented into small regions which are identified using pre-specified attributes of the license plate. [2]
- Colour Based Method: Color-based method deals with the RGB colour images to HSV-color image and the converted image is segmented into the smaller block for processing. [3]

2.2 Machine Learning Methods

Colour-Based Method: CNN Based methods have been used, which can automatically learn features from the acquired dataset. This type of detection methods have yielded good results but their time consumption is high. [1]

Many researchers have taken up Convolutional Neural Networks (CNN) to solve the problems in license plate detection. Recently, researches of computer vision tasks in transportation system have made significant progress. Under high demand for robustness, some new methods, tend to employ the features extracted by CNN instead of hand-crafted features. Still, CNN fails to solve the time taken for the detection. Even though the convolutional kernel has freedom in determining it couldn't manage with the time taken to process the data. To overcome this we have adopted a classifier in which we train the weightings of each feature.

A Haar-Feature is just like a kernel in Convolutional Neural Network (CNN), except that in Haar-Feature the values are manually determined and not by training. Haar-Feature is mostly observed or used to detect faces whereas it can be more efficient in terms of detecting multiple license plate of vehicles.[5] A detailed explanation of the feature adopted is given in section 3.

3. METHODOLOGY

In this section, we see in detail about the algorithm and features.

3.1 Haar-Cascade Classifier

Haar Cascade is a machine learning-based object detection algorithm used to identify objects, faces in an image or video. Based on the concept of features. [4] This was the base to all the object detection done using Haar-like features algorithm. Haar cascade is a machine learning based approach where the cascade function is trained using a lot of positive and negative images.

First, let us take a look at the main differences that we have between a Convolutional Neural Network (CNN) and a Harr-like feature. A CNN method uses training to determine the values, while a Haar-feature manually determines the values.

Below given image shows features that are used in the "Haar-like feature" algorithm.

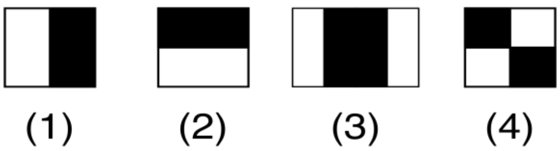

Fig. 1: Representation of edge features

In figure 1, the first two in the given image are "Edge Features", which are used to detect the edges in the input images. The third represents the "Line Feature" used in the detection process. The fourth shows the "Rectangle Feature" which are most likely to be used to detect a slanted line.

The below-shown image is the numerical representation of a Haar-feature model.

-1	-1	5	5	5	5	-1	5	-1	5	-1	-1
-1	-1	5	-1	-1	-1	-1	5	-1	-1	5	-1
-1	-1	5	-1	-1	-1	-1	5	-1	-1	-1	5
	(1)			(2)			(3)			(4)	

Fig. 2: Numerical representation of edge features

Each of the 3×3 kernels moves across the image and does matrix multiplication with every other 3×3 part of the input image, emphasizing some features.

Haar-Features are good at detecting edges and lines. Haar features have to be determined manually, there is a certain limit to the types of thing that can be determined. However, if you give the classifier, that is the algorithm edge and line features, then it will be able to detect objects with clear edges and lines with good accuracy. On the plus side, Haar-Features don't need to training, can create a classifier with a relatively small dataset. The main thing and important thing that has to be done is to train the weightings for each feature and which Haar-Feature should be used more. In that way it allows us to train the classifier well without a lot of training images. Haar-based classifiers are typically higher in terms of execution speed and time. Also, the computations involved are less.

The algorithm basically has four stages:
1. Haar Feature Selection
2. Creating Internal Images
3. Adaboost Training
4. Cascading Classifiers

The first step is to collect the Haar Features. As said a Haar-feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums.

The second step is to make use of the internal images. An internal image can help speed up the process. But among all the feature we calculated, most of them are irrelevant. To rectify the problem with the irrelevant feature, the concept Adaboost, which both selects the best feature and trains the classifiers that have to use them to process. This the third step in the algorithm. [6]

During the detection phase, a window of target size is moved over the input image and for each subsection of the image and Haar features calculated. The difference is then compared to a learned threshold. As each Haar-like feature is only a "weak classifier" a large number of Haar features are required to describe an object with proper accuracy and therefore organized into a "cascade classifier" to create a strong classifier.

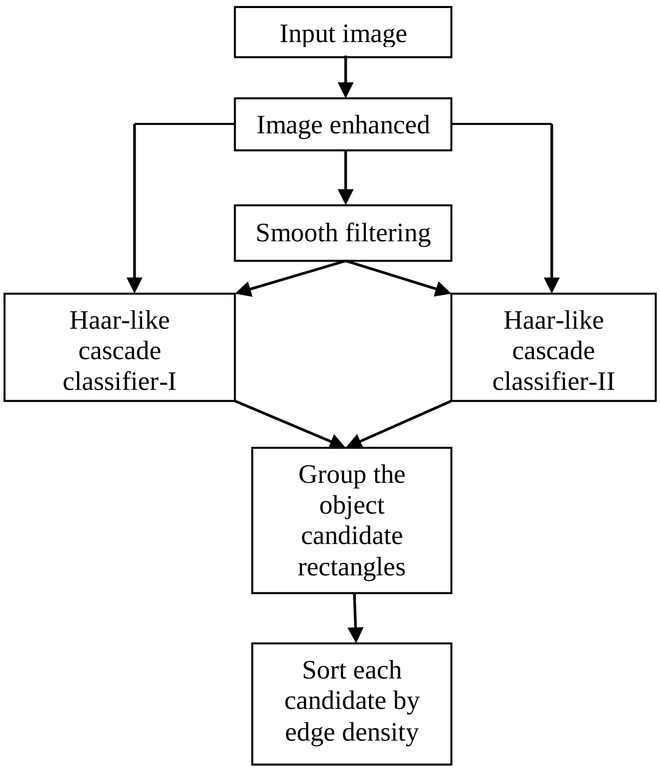

Fig. 3: Haar-Cascade classifier flow

3.1.1 Image Processing: The original colour of the image is converted to a grey scale as we don't need the information about the colour. The image size is resized to1/4 so that the algorithm can be speeded up. The contrast of the image is improved by equalizing the histogram of the image. Small detailed images are removed using the Gaussian filter to avoid false detection.

3.1.2 Haar-like cascade classifier with Adaboost: Some number of Haar-like features in default are extracted from the window. Adaboost is used to train a great number of features as it is so effective. Adaboost selects certain good and efficient features to develop a weak classifier, stump classifier or classification and regression tree classifier. The process of building a stump weak classifier is described in the upcoming steps.

- Put positive and negative samples together.
- Randomly choose a Haar-like classifier to calculate the feature value for each of the samples.
- The samples are sorted based on their values.
- The best threshold value is selected to divide the positive and negative samples.
- In such a way a stump weak classifier is developed.
- If the detection rate is more than 0.5, it is used to develop a strong classifier.

Strong classifiers are also formed by combining weak classifiers together. The structure of a strong classifier is shown below figure.

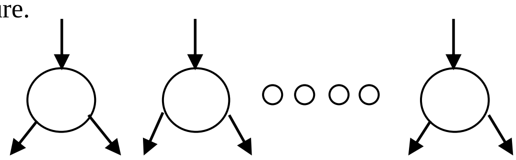

Fig. 4: Combination of weak classifiers

To reduce computation time in the detection process a cascade structure is adopted. The total time is reduced by processing large areas of the input image with the help of simple classifiers and processing the rest of the input image with complex classifiers.

Two Haar-like cascade classifiers are used to form an ensemble classifier. This can help us to detect more than one license plate detection in the given input image. Both the classifiers are trained with the same positive and negative sets. By doing in this way we can improve accuracy and speed.

3.1.3 Image post-processing: The classifier may output more than one candidate region. In this case, we use some image processing methods which has some prior knowledge about how to choose the best candidate region.

- The area of license plates is moderate. Here the defined width is larger than 50 pixels and smaller than 1/3 of the image's width.
- The aspect ratio of the region that is, width and height should be within a given range 2 to 3.5.
- Different parts of the region of a license plate may be chosen by classifiers many times.
- The region of a license plate has rich vertical edge information.

The algorithm to sort the candidate regions detected is as follows:

- The vertical edge of the candidate region is computed using a Sobel operator.
- Calculate the area of each candidate region S.
- Calculate the density of the vertical edge in each region by:

$$D = \frac{\sum RD_y}{S} \qquad (1)$$

Where D_v is the value in a candidate region of vertical edge image. D is the vertical edge density in the candidate region.

- Normalize each candidate region's area and density to [0,1]
- The final principle is:

$$P=D+S \qquad (2)$$

- The larger the P is likely the region is a license plate.

First, the candidate regions of the detected license plate which do not satisfy the first and second principles are removed. Then the candidate regions whose distance to each other are lower than the given threshold are combined, the distance is defined as,

$$D = (min\{w1,w2\}+min\{h1,h2\})*0.5. \qquad (3)$$

Where D is the distance of two candidate regions, w1 and w2 are the widths of the two regions, and h1 and h2 are their height, respectively. The threshold value to merge regions is fixed as 0.2D. If the distance of corresponding points between two rectangles is lesser than 0.2D, they would be considered as the belonging to the same rectangle. Then the 4 vertices coordinates are averaged to derive an average region.

Finally using the fourth principle the remaining regions can be sorted. This is done because even the false candidate regions may have rich vertical edge information so, the candidate region is also considered.

3.2 Optical character recognition

OCR is the abbreviated form of Optical Character Recognition, which also known as Optical Character Reader used to convert the electronic or handwritten or printed-text into a machine-encoded text. The input to this can be in the form of a scanned document, a photo with words, signboards, handwritten documents etc. OCR is widely used in the fields of pattern recognition, Artificial Intelligence and computer vision.

Optical Character Recognition is trained with images of each character in different fonts and styles. Classical fonts and most common fonts are trained so well that it produces a high degree of accuracy in predicting the characters in the given input. There is a lot of open source OCR available which can be taken up or adopted to evaluate the images to machine-encoded text.[8] Tesseract has been adopted to evaluate the images that have to be converted into text.

Tesseract is one of the popular OCR available in the open CV has been adopted to process the license plate image which has been cropped out from the whole image of a car. The algorithm used to crop the license plate was Haar-like cascade classifier. The license plate is sent as the input image for evaluation.

Tesseract was developed as an open source software in 2005.

Tesseract supports many languages and also many image formats. It is based on the Long Short Term Memory (LSTM) recognition engine. Long Short Term Memory (LSTM) is a form of Recurrent Neural Network (RNN) for text conversion. To recognize an image containing a single character, typically a Convolutional Neural Network (CNN) is used. To recognize or detect a sequence of characters LSTM is used, which is a popular form of Recurrent Neural Network (RNN).

The Tesseract can be used by integrated into the coding language like C++ or python. This tool can be used to perform OCR on the image and the output is stored in a text file. All the library package has to install to access the Tesseract API. [7]

The working of Tesseract is explained in the following lines. Tesseract has two OCR engines Legacy Tesseract Engine and Long Short Term Memory (LSTM) Engine. During the OCR engine mode, there are four operations given to process the input image.

- Legacy Engine only
- Neural nets LSTM engine only
- Legacy + LSTM engines
- Default, based on what is available

These four operations help to carry out the required operation for processing the input image. Tesseract best works when we have a solid white background n a common font style. The performance of Tesseract can improve when the text is approximately horizontal and the height is 20 pixels. Many methods have been adopted to overcome this problem and process every type of inputs properly.

4. IMPLEMENTATION AND WORKING

Each and every module are implemented to obtain the desired output. The implementation and the working part of the system are briefed in the upcoming section.

The whole system has been implemented in the local host web page, where the input image of a car whose license plate should detect cropped and analyzed should be uploaded. This action receives the input image and sends it to the processing stages. First, the image is processed by the Haar-like feature cascade classifier, where the image is evaluated frame by frame. The whole image is processed by a 3x3 matrix. This process is done to locate the license plate in the whole image. When the given threshold value for a license plate matches with the input image it is considered as the license plate and that region is marked with a rectangular box. After locating the license plate, it is then cropped and processed before sending it into the next module of the system.

The next module is about converting the detected and cropped out the image's digital characters into machine-encoded text. This operation is carried out by an OCR called Tesseract. Tesseract uses image segmentation and processes the image given. The characters in the image are compared with the existing trained knowledge of the OCR to find each and every character. RNN is used in Tesseract which is efficient in recognizing a string of characters. By this way, the license plate number can be obtained as a machine-encoded text.

The next phase is where the converted machine-encoded text is matched with the manually created database with limited data. This is done to compare which license plate of the car matches the details in the database. If the data matches with the database, the details of the license plate are printed on the web page with the license plate number, name of the owner, phone number etc. This is the working and implementation of the whole system.

5. IMPLEMENTATION RESULTS

This section consists of the implementation results of the whole system. Implementation of the processing of Figures is attached.

- The local host web page where the admin has to log in and upload the image for processing. This web page module consists of username and password fields for the admin to login.
- After the authentication of a valid Username and Password. The user is shown a welcome sign and ready to upload an input image for processing.
- The page where the image has to be uploaded for processing image to crop out license plate from the whole input image.
- The window that opens after the option chooses a file is clicked on the web page. The image to be processed can be browsed in the system folders and selected for uploading.
- The license plate is marked with a rectangular box, after choosing the image it enters the algorithm, where the license plate is located.
- The license plate is cropped out of the whole input image given and shown separately. This cropped out the image of the license plate is used as the input image for the next phase, where this gets converted into a machine-encoded text.
- The final output of the system where the cropped out license plate image has been converted into a machine-encoded text. The converted string is then matched with the manually created database with a small amount of data to find out of the license plate number matches any of the car license plate numbers in the database. If the data matches, the corresponding details of the car owner is printed. In case the detected license plate doesn't match the database an error message stating "License plate mismatch" is shown in the result page of the web page.

Fig. 5: License plate detection by Haar-cascade

Figure 5 shows the license plate detection and marking of the borders to process the image for cropping out.

Fig. 6: Cropped out license plate after detecting

Figure 6 shows the image of the license plate that has been cropped out from the whole image after being detected by the Haar-cascade classifier.

5.1 Datasets

Datasets for a Haar-cascade to get trained and to identify the object that has to be detected can be grouped into two categories. Positive and negative datasets. The positive dataset consists of positive images which are correct images of the license plate. This can be used to develop the classifier to learn about the positives of how a license plate should be. An average of 2000 positive images is given to the classifier to train the model.

Negative images are added to the negative dataset which has all the improper and wrong images which can be miss interpreted as a license plate. These datasets help the classifier not to identify the parts in the image which can be wrong assumptions. To improve the detection process of the classifier the cascade needs to be trained with lots and lots of images of the license plate and negative images of which may look like a license plate.

The more dataset images were given to the classifier the more its accuracy in detecting the object (here license plate) increases. An average of 2000 positive and 2000 negative datasets are required to create an efficient classifier. The datasets of a Russian license plate models have been adopted here. This is some already trained datasets which are available online at opencv. These datasets can be used to perform the process of detecting a license plate and extracting t separately. These datasets can be improved by adding even more images of the license plate for both positive and negative datasets. Haar-cascade can even predict the object with a lesser number of datasets used, like 600-700 positive and negative images trained properly can develop a good classifier.

5.2Analysis

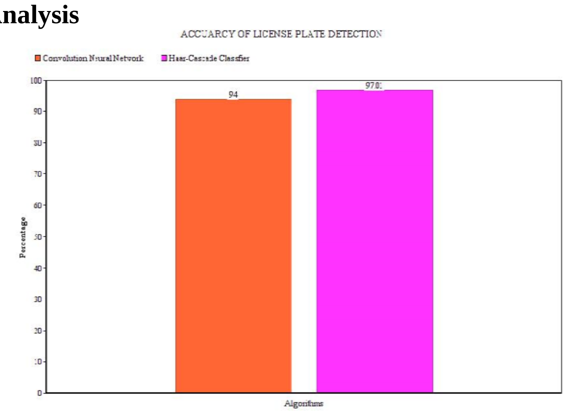

Fig. 7: Accuracy comparison with existing method (CNN)

Figure 7 shows the graph analysis between Convolutional Neural Network (CNN) the previous method which has been used and the Haar cascade classifier which has been adopted now. This graph is generated in terms of accuracy in detecting the object which is here the license plate for both the methods. CNN has a slightly lower rate in terms of accuracy which is 94%, whereas in Haar-cascade classifier the accuracy for object detection is a bit higher than CNN i.e 97.01%. This difference gives a bit advantage in using Haar-cascade for detecting license plate. The accuracy rate of the Haar-cascade classifier can further be improved by training the classifier model with more and more positive and negative datasets. This can help improve the percentage rate even more.

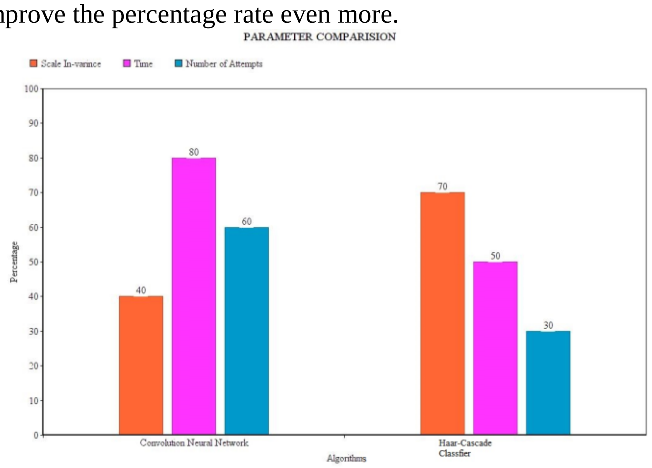

Fig. 8: Parameter comparison between CNN and Haar cascade

Figure 8 shows the graphical analysis comparison between the two methods Haar cascade classifier and CNN. The parameters that have been taken for comparison are Scale invariance, time and number of attempts or training period. A good object detection algorithm should have strong scale invariance which Haar-cascade classifier has more than the Convolutional Neural Network. The time taken to process an image and detect the specified object (here license plate) should be less so that it makes the algorithm robust. Haar-cascade has a relatively lesser time consumption than CNN. The number attempts or the period to train the datasets has to lesser but the results should be good. This is achieved well in Haar-cascade where the classifier can be trained with 600-700 image datasets to perform a good detecting model, whereas a Convolutional Neural Network needs a lot of time to train the model which is a disadvantage in CNN.

6. CONCLUSION

In this paper, the proposed system is that the license plate of the car is located in a whole image and it is detected. The detected license plate is then cropped out to separate image. This image is then processed for character recognition and the characters in the image are converted into a machine-encoded text. This system can be implemented in various real-time scenarios to avoid traffic violation, to identify an unknown car, and more using the license plate. The system can also be proposed in unattended car parking, an unmanned toll booth. This proposed system has improved the efficiency in time consumption for image processing. Also, the system is dependent on several factors which can be said as the disadvantage for this proposed system. This proposed system is able to detect all kinds of number plates irrespective of the colour, but it can only detect a plate in a particular angle and can accept only slight deviations. The conversion of the digital image characters into machine coded text is effective only on a white background board. To avoid this problem alternative methods were adopted which lead to high time consumption. The future work of this system can be proposed to develop a robust technique in the character conversion module.

7. REFERENCES

[1] A New CNN Based Method for Multi-Directional Car License Plate Detection. Lele Xie, Tasweer Ahmed, Lianwen Jin, Member, IEEE, Yuliang Liu, and Sheng Zhang November 2017.

[2] A Hybrid License Plate Extraction Method Based On Edge Statistics and Morphology. Bai Hongliang, Liu Changping 2004.

[3] An Iranian License Plate Recognition System Based On Color Features. Amir Hossein Asshtari, Md Jan Nordin, Mahmood Fathy August 2014.

[4] License Plate Location Based on Haar-like Cascade Classifiers and Edges. Yuanxing Zhao1,2,3, Jing Gu1,2, Chui Liu1,2, Shumin Han1,2, Yong Gao1,2, Qingmao Hu1,2 1. Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China 2. The Chinese University of Hong Kong, Hong Kong 3. The Hebei University of Technology in 2010.

[5] Multiple License Plate Detection for Chinese Vehicles in Dense Traffic Scenarios. Muhammad Rizwan Asif, Qi Chun, Sajid Hussain, Muhammad Sadiq Fareed 2016.

[6] License Plate Detection Using Adaboost. Louka Dlangnekov, Computer Science and Engineering Department, San Diego, 2004.

[7] Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study. Chirag Patel, Atul Patel, Dharmendra Patel 2012.

[8] Optical Character Recognition. Ravina Mithe, Supriya Indalkar, Nilam Divekar 2013.

