2012 10th IAPR International Workshop on Document Analysis Systems

On the Enhancement and Binarization of Mobile Captured Vehicle Identification Number for an Embedded Solution

T. Chattopadhyay
Innovation Lab, Tata Consultancy Services,Kolkata, India
Email: t.chattopadhyay@tcs.com

Ujjwal Bhattacharya, Bidyut B. Chaudhuri
CVPR Unit, Indian Statistical Institute, Kolkata, India
Email: {ujjwal,bbc}@isical.ac.in

***Abstract*—An embedded solution for automatic detection of Vehicle Identification Numbers (VIN) captured by a mobile camera has a number of real world applications. But the performance of available open source Optical Character Recognition (OCR) systems on VIN images captured by mobile phones is extremely poor because of the image quality affected by various noises. In a recent study of such images, we have observed that the performance of existing open source OCR systems can be improved by applying several image enhancement techniques on these images before sending them to the OCR engine. In this article, we have presented such a method that improves the recognition accuracy from 5.89% up to 82.3%.**

***Keywords*-OCR; Median filtering; Retinex; VIN images;**

I. Introduction

According to the definition provided in Wikipedia [1], VIN is "a unique serial number used by the automotive industry to identify individual motor vehicles". The VINs are imposed to be used by the National Highway and Traffic Safety Administration of the USA. So, in case of any damage or accident, the VIN of the concerned vehicle is needed to be sent to the insurance company to process any insurance claim. On the other hand, with the advancement of the consumer electronics technology most of the mobile handsets are now equipped with a digital camera. Smart phones are also capable to do some processing [2] on the embedded hardware platform of the mobile handset like Android. So the insurance companies are thinking of providing some application for the smart phone users that can recognize the characters from the VIN images captured by the common users. These 17 alpha-numeric characters cannot be sent by simply typing them because of the authentication issues. The problem we have addressed in this paper is mainly motivated by the need of one such major insurance company of the world.

The OCR for texts in mobile camera captured images is still an unsolved problem. Some research in this area in can be found in [3], [4], [5], [6], [7]. A few benchmark databases [8] of camera captured outdoor scene images containing texts are now available to facilitate fruitful research in related areas. However, the present problem of recognizing characters in VIN images captured by mobile camera is bit more difficult because of the reasons described below.

- It is required to extract individual characters on embedded mobile platform which has constraints with respect to both memory and processor. Moreover, it requires real-time recognition. On the other hand, the client (insurance company) does not favor a cloud-based option.
- The VIN images are captured by common people at difficult situations. Thus, such images are naturally affected by different sorts of noises and artifacts. A few samples of such images are shown in Fig. 1.
- Some suitable low-complexity binarization method is required before invoking the OCR module.

Sample VIN images shown in 1 poses different research issues to the research community. However, in this paper we mainly concentrate on alpha-numeric characters against complex background.

In Section II of this paper, we discussed proposed preprocessing strategies suitable for the present images. Experimental results are presented in Section III. Section IV concludes the present article.

II. Proposed preprocessing operations

In order to recognize the VIN from the images, we need to follow the sequence of methods used in an OCR problem namely (i) preprocessing, (ii) Binarization, (iii) OCR, and (iv) Post processing. We have used Tesseract as the OCR engine. The main motivation behind selecting Tesseract as our OCR engine is that it is light weight and it is now-a-days available in mobile platform like Android. Our value proposition lies in image enhancement and binarization of such images so that the recognition accuracy can be improved. The recognition accuracy can be further improved by incorporating the post processing modules like language model which we keep as a future research scope.

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.62

(a)

(b)

(c)

(d)

(e)

Figure 1. A few sample VIN images: (a) VIN on complex background, (b) VIN with perspective distortion, (c) VIN affected due to light reflected from other objects, (d) parts of a few characters are missing from its metallic plate, (e) embossed VIN.

A. Image enhancement

We have observed that the images of VIN we are considering here consist of two sources of noises. One is multiplicative in nature that comes because of the background text and the reflection from the glass. On the other hand the source of additive noises are also there. We have seen that the performance of binarization module can be improved if we apply some image enhancements in prior to binarization. Image enhancement by retinex strategy [9] followed by median filtering has provided the best result. In Fig. 2 the effect of this enhancement strategy are shown.

B. Binarization

We have used the existing well known binarization techniques [10] like Otsu, Sauvola, Niblack, Wolf [11], and Howe [12] for binarization of the present sample images. But the Tesseract can not recognize the characters when the binarized output by these methods are submitted as input for the OCR. Output of these methods are shown in Fig. $3(a)$–Fig. $3(g)$.

So we have formulated a novel low complexity binarization module that can be implemented on an Android platform as well as can increase the recognition accuracy. Our proposed method is based on two observations of these

(a)

(b)

(c)

(d)

(e)

Figure 2. Effects of image enhancement: (a) input color image, (b) grayscale image, (c) only retinex filtered image, (d) only median filtered image (e) retinex followed by median filtered image

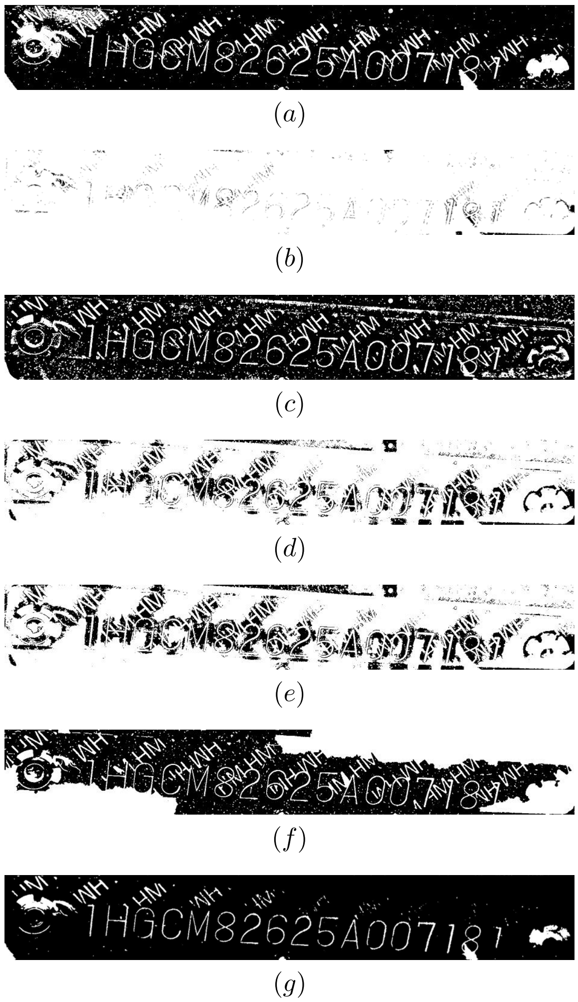
VIN images like (i) There is slight gray scale variation between background text (BGT) and the text of interest (TOI) and (ii) strictly 17 alphanumeric characters are there in a VIN image. The proposed method has the following steps:

- Compute the histogram of the gray scale image.
- Find the two major peaks. One located near 0 and the other located near 255.
- These two peaks represents the background and foreground part of the image.
- Use 16 equally spaced points (pixel values) between these two peaks as threshold for binarization.
- Thus obtain 16 binarized images from the single input.
- Label the foreground pixels of each such images using connected component labelling methods.
- Remove the components that are too small or too big. We define a component to be too small if the number of pixels with that particular level is less than 100 or it has a height (h) or width (w) less than 3 pixels. Similarly the too big components are defined to be those having a $h > ht\_image/3$ or $w > wd\_image/4$ where ht_image and wd_image represents the height and width of the image.
- If the number of component is less than 8, it means that actual 17 . characters are either very under segmented or the binarized image doesn't include all valid characters as foreground. So this binarized image is not considered as a candidate image.
- Similarly if the number of components is greater than 51 then on the average one valid character is over segmented to more than 3 segments. We disregard over-segmented and under segmented binarized images. The remaining binarized images are considered as the candidate image.
- Thus we have only a few valid images left out of those 16 binarized images and on average we found that the number of such candidate images for each input image is less than or equal to 3.
- We combine the candidate images by marking the pixels as BGT only if it is decided as a BGT in more than 50% of these candidate images and thus construct the final candidate image. One such candidate image is shown in Fig. $3(g)$.

C. Segmentation and morphological closing

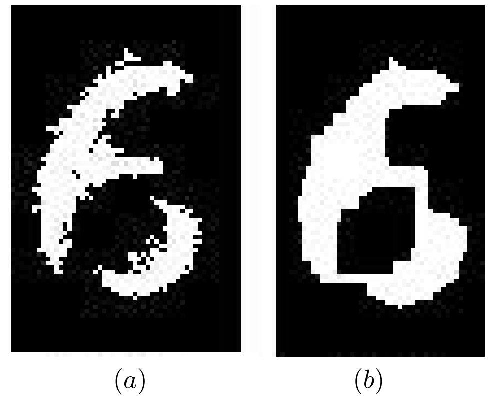
Once we get such candidate binarized image we try to split it into 17 segments so that each segment contains only one valid character (may be in over segmented form). We have used conventional method of skew correction prior to segmentation. The character/numeral segmentation and recognition method is described below.

- Find the columns without any foreground pixel. If we get consecutive such rows, we take the middle of the these columns as the candidate cut column (CCC). Let the number of CCC we obtain be n.
- Find the distance (δ) between the consecutive CCCs. Let the distance between the i^{th} and the $(i+1)^{th}$ CCC be defined as $\delta_i = |CCC_{i+1} - CCC_i|$.
- Find the median (med_δ) of $\delta_i \forall i \in (1, 2, \cdots, n)$ where n is the number of CCCs in the image. We have also used a heuristically obtained tolerance factor τ to define the threshold (Th_δ) which is defined as $Th_\delta = (med_\delta - \tau)$.
- If we get $n = 16$ $\delta_i \geq Th_\delta \forall i \in (1, 2, \cdots, n)$ i.e. we get 16 components which are nearly equally spaced columns we use each segment as a candidate segments.
- If $n > 16$ then we conclude that some valid character is horizontally over segmented. Then we initially mark all i's for which $\delta_i < Th_\delta$. Then we merge such CCCs and reduce n by one iteratively. The terminating condition is obtained when $n = 16$ and $\delta_i \geq Th_\delta \forall i \in (1, 2, \cdots, n)$.
- If we get $n < 16$, we conclude that there is definitely some valid characters touching each other. We have used the method described in [13] to segment them.
- Thus we obtain 17 segments each having a valid character may be in over segmented form. If any such segment includes multiple component labels, we merge them by applying morphological closing. One such example is shown in Fig. 4.
- These 17 segments are given as input for the tesseract OCR engine.



(a) (b) (c) (d) (e) (f) (g)

Figure 3. Effect of various binarization methods (a) Otsu, (b) Savoula, (c) Niblack, (d) Wolf, 2001, (e) Wolf, 2007, (f) Howe, (g) proposed strategy



(a) (b)

Figure 4. (a) over segmented character, (b) character after morphological closing

III. Experimental results

The effect of binarization method proposed here is qualitatively compared in the images above. In this section we are showing quantitative performance by comparing the proposed method against different other binarization techniques. The results are shown in Table 1.

It was observed that our segmentation and the morphological closing module helped a lot to increase the recognition accuracy as in some cases the recognition accuracy of the Tesseract suffers because of over segmentation. For example "M" is recognized as a sequence four characters namely "$|\backslash/|''$ or "6" (as shown in Fig. 4) is recognized as two characters in spilled over two lines namely "ϵ" and ",". On the other hand if we apply the morphological closing prior to segmentation a sequence of characters "1" and "3" is recognized as "B". Thus the recognition accuracy of the system increases to 82.35% from 53.97% while the segmented and morphologically closed characters are used instead of the entire binarized image as input for OCR. Also the noise cleaning module that removes the too small components helps to improve the recognition accuracy. At the same time use of domain knowledge that which character would be alphabetic or numeric helps to reduce the scope of false recognition because of over segmentation. For example the scope of confusion between "0" and "o" is reduced in this manner. But in case of Niblack's method the recognition accuracy marginally increases as the TOI in the binarized output itself includes the BGT. So the segmentation module could not detect the 17 required segments. Moreover, we have manually segmented the binarized file, but the recognition accuracy didn't increase still because of the complex touching characters (touch between BGT and TOI). On the other hand, in case of Sauvola, and the two methods by Wolf et.al, the binarized output includes a lots of background components detected as foreground and thus the recognition accuracy didn't improve. But in case of Otsu's method the recognition accuracy increases after segmentation and morphological closing for some of the characters where the binarization technique can successfully segregate the background text from the foreground text. The effect of noise cleaning also reduces the possibility of the merging with wrong neighboring components to reduce the effect of over segmentation. The improvement by introduction of the post processing (noise cleaning, segmenting into 17 segments and morphological closing) and use of domain knowledge while using the OCR is depicted in Fig. 5.

The recognition accuracy of the system can be improved by adding some post processing on top of it like among these 17 characters some characters are strictly character and some are strictly numerals. We also know that for a particular car manufacturer (encoded in first 3 characters) only some characters are valid in next 5 character set. We can define some lexicon using domain knowledge to improve the performance of recognition by applying some language

Table I
COMPARISON OF RECOGNITION ACCURACY OF DIFFERENT BINARIZATION TECHNIQUES

Binarization Method	Recognition Accuracy before segmentation	Recognition Accuracy after post segmentation
Otsu	0%	23.52%
Niblack	5.89%	11.76%
Sauvola	0%	0%
Wolf 2001	0%	0%
Wolf 2007	0%	0%
Howe	0%	0%
Proposed	53.97%	82.35%

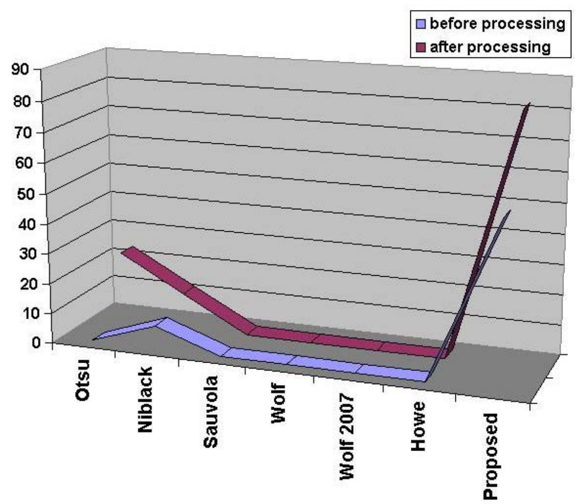


Figure 5. Improvement by introducing post processing

model like [14] on top of it.

Since the first three characters represent manufacturer id and since there are only a few manufacturers, we can use a Minimum edit distance based correction scheme if the first three recognized characters do not represent any valid manufacturer. It is not computationally expensive. Also, the recognized set of rest characters may represent a valid or invalid car id. In the second case the same technique may be employed. Even if the system makes say 6% error, ie one in 16 characters can be wrong, the receiver can find the subset of valid car numbers out of this and use other info to zero in the correct number.

IV. CONCLUSION AND FUTURE SCOPE

A low-cost approach for VINs captured by mobile phone cameras is studied in this paper. While more sophisticated algorithms could improve the recognition, we have to restrict their uses because of the resource constraints of the selected hardware platform like low computing power of the system. We have used Tesseract as the recognition engine, but we plan to design a dedicated recognition engine by taking care of the font characteristics of the VINs that may be more accurate and less computation intensive. Moreover, we plan to develop a detailed system of retrieving the VIN from partial and erroneous results, as stated above. Thus the optimization, porting related issues of the system is also left as a future work.

ACKNOWLEDGMENT

The authors of this paper would like to thank Mr. Arpan Pal and Mr. Aniruddha Sinha of Innovation Lab, Tata Consultancy Services, Kolkata for their help to execute the project.

REFERENCES

[1] http://en.wikipedia.org/wiki/Vehicle_Identification_Number, *last accessed on Oct, 2011.*

[2] http://www.abbyy.com/mobileocr/, *last accessed on Oct, 2011.*

[3] A. R. Chowdhury, U. Bhattacharya and S. K. Parui, "Text detection of two major Indian scripts in natural scene images", *Proc. of 4th CBDAR*, pp.73-78, 2011.

[4] U. Bhattacharya, S. K. Parui and S. Mondal, "Devanagari and Bangla text extraction from natural scene images", *Proc. of 10th ICDAR*, pp. 171 - 175, 2009.

[5] J. Liang, D. Doermann and H. Li, "Camera-based analysis of text and documents: a survey", *Int. Journ. on Docu. Anal. and Recog.*, vol. 7(2-3), pp. 84-104, 2005.

[6] D. Doermann, J. Liang, H. Li, "Progress in camera-based document image analysis", *Proc. of the 7th ICDAR*, vol. 1, pp. 606-616, 2003.

[7] P. Shah, S. Karamchandani, T. Nadkar, N. Gulechha, K. Koli, K. Lad, "OCR-based chassis-number recognition using artificial neural networks," *Proc. IEEE Int. Conf. on Vehicular Electronics and Safety*, pp. 31-34, 2009.

[8] http://algoval.essex.ac.uk/icdar/Datasets.html, *last accessed on Oct, 2011.*

[9] R. Kimmel, M. Elad, D. Shaked, R. Keshet, I. Sobel, "A variational framework for retinex", *Int. J. Comput. Vision*, vol. 52, no. 1, pp. 7-23, 2003.

[10] O. D. Trier and T. Taxt, "Evaluation of Binarization Methods for Document Images", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, pp. 312-315, 1995.

[11] C. Wolf and D. Doermann, "Binarization of low quality text using a Markov Random Field model, *Proc. of ICPR*, vol. 2, pp. 160-163, 2002.

[12] N. Howe, A Laplacian energy for document binarization, *Proc. of 11th ICDAR*, pp. 6-10, 2011.

[13] H. Ghosh, S. K. Kopparapu, T. Chattopadhyay, A. Khare, S. Wattamwar, A. Gorai and M. Pandharipande, "Multimodal indexing of multi-lingual news video", *Int. Journ. of Digital Multimedia Broadcasting*, vol. 2010, Article ID 486487, 18 pages, 2010.

[14] S. Chowdhury, U. Garain, and T. Chattopadhyay, "A weighted finite transducer (WFST)-based language model for online Indic script handwriting recognition", *Proc. of 11th ICDAR*, pp. 599-602, 2011.