ISSN 2047-3338

Settlement Quote Generator for Auto Business Using OCR Technology

A. Satish Kumar[1] and CH. Satyananda Reddy[2]

[1,2]Andhra University College of Engineering, Visakhapatnam, Andhra Pradesh, India

***Abstract*– Optical Character Recognition (OCR) is a widely used and emerging method in various applications.OCR deals with the concepts in pattern recognition, artificial intelligence and computervision.OCR is the technique of converting images of handwritten or printed text into user understandable format. Accuracy of OCR is based on constraints namely, text pre-processing and segmentation algorithms. Sometimes it becomes hard to recognize characters from the image due to presence of light and dark text or due to varying size, style, orientation, complex foundation of picture and so on. In this paper, OCR techniques are been used for the recognition of vehicle number from vehicle number plate. The retrieved OCR data is then used as a web service hence reducing human work and computerizes the system. This paper explains the Tesseract integration procedure, post-processing techniques and focuses on the training data preparation process. A large number of images are taken as input files and experimental consequence of OCR performed by Tesseract on these pictures are discussed. The results obtained are compared with other commercial OCR tool, Transym OCR and analysed using different parameters.**

***Index Terms*– Optical Character Recognition (OCR), Open Source, DLL, Tesseract, Trained Language, Transym, JAVA, JVM and Android**

I. INTRODUCTION

OPTICAL Character Recognition (OCR) is a transformation of scanned or printed content pictures, handwritten content into editable content for future processing. In the early days, OCR was used as a hand-held scanner which scans across a printed page and produces tones for specific letters and characters. With the increase in number of Smartphone and smart glasses, OCR is been used to extract text from image captured from the device's camera or image retrieved from device's storage memory. The OCR API displays the output in user editable format for processing.

It is just like the working of eye and mind in a human body. An eye has the ability to view the content from images but the content read by the eye is actually processed and translated by the brain. The OCR analyses the image and recognises the patterns on it by comparing each and every character with the nodes in the Artificial Neural Networks (ANN's).Several problems may occur during the construction of a computerized OCR framework. Firstly, there is a very minute difference between some of the characters and digits. Secondly, the output gets affected due to the presence of dark and dull background or words imprinted on other words or illustrations. In order to rectify this problem various pre-processing techniques and segmentation algorithm is been used. Pre-processing is essential to remove unnecessary background colours, spots, lines from the picture. Segmentation is needed in order to separate each and every character from words and recognizing individually. In this paper, OCR recognizes text images using various Artificial Neural Networks (ANN's) techniques. This system is able to achieve average 83% recognition of English letters and digits that represent training data sets. If Support Vector Machines (SVM) method is used for Scanned images to text recognition, then the system is able to achieve 99.18% accuracy for English characters. This suggested local binarization method for document images by using a thresholding method and dynamic window. This system is able to achieve F-mean values of 89.1% for handwritten text and 93.93% for printed text.

The procedure used in OCR to scan and extract images is shown in Fig. 1. Various steps regarding markov model for handwritten character are explained.The image is obtained and converted into gray scale image. The resultant gray scale image is then processed and transformed into binary image. This process is called Digitization of image (Binarization). Practically not every scanner is perfect; the scanned image may have some noise. The reasons for noise can be due to the presence of some unwanted and unnecessary features in the image. By applying suitable methods, the denoised image is produced. The denoised image thus obtained is saved for further processing.

Character Extraction: The pre-processed image serves as the input to this and each single character in the image is found out.

Recognition: The resultant image from the previous stage is correlated with all the templates which are preloaded into the system. After the correlation is complete, the template with the maximum correlated value is declared as the character.

Post Processing: After the recognition stage, if there are still some unrecognised characters found, then those characters are given their meaning in the post-processing stage. Additional templates can be added to the system for providing a wide range of compatibility checking in the systems database.

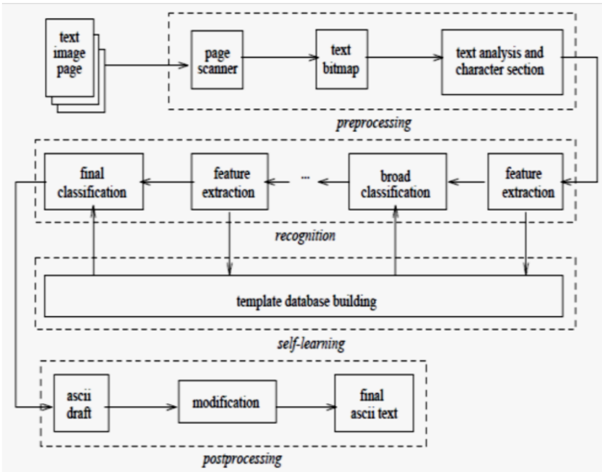

Fig. 1. Pre-processing and Post-processing

In this work, a huge number of colour and gray scale images are taken and it is converted into text format which can be readable to the user. The results are being computed in tabular form and comparing the analysis on Tesseract and Transym. The algorithm used here is Segmentation algorithm where each and every character is being segmented and recognised individually using data path where individual characters are being compared with the characters in data path. This algorithm involves ANN (Artificial Neural Networks) techniques. Processing of the image data is also performed concurrently [4]. The output is then displayed in text format. The readable text is then send to the web service and the text is then searched in the database of the bank. The response is being provided to the user of the API interface. The paper takes a huge number of images as input and processes it into editable text format using Tesseract library. The results are compared with that of Transym OCR. The accuracy of each image in coloured or gray scale format is analysed and shown in tabular form.

II. RELATED WORKS

Optical Character Recognition (OCR) is a transformation of scanned or printed content pictures, handwritten content into editable content for future processing [15]. It is just like the working of eye and mind in a human body. An eye has the ability to view the content from images but the content read by the eye is actually processed and translated by the brain [18]. During the construction of a computerized OCR framework [4], several problems may occur. First, there is a minute difference between some of the letters and digits which are difficult for the computer to understand. For instance, it might be difficult for the APP to distinguish between digit "0" and letter "o". Second, the words in dark and dull background or words imprinted on other words or illustrations are hard to recognize by the APP. There are many applications used for OCR, which includes License plate recognition extracting text from scanned documents [1] etc. The system is to rectify the text retrieved form camera captured images [4]. This used to recognize text images by using Artificial Neural Network (ANN). This system is able to achieve average 83% recognition of English letters and digits. Is used Support Vector Machines (SVM) method for Scanned images to text recognition [17]. This system is able to achieve 99.18% accuracy for English characters. This suggested local binarization method for document images by using a Thresholding method and dynamic window [3].

The current procedure followed by companies and banking application for billing process has maximum part of manual process and only for calculating bill an automated process is used. The OCR is used to extract to image from text updating that reading into server. The above described procedure increases the work load on the employees, also it will more time for completion when compared to computerized methods using the now available technology.

III. PROBLEM SPECIFICATION

This is a mobile application which makes possible obtaining text and working with it from pictures taken from a camera hardware. It has a simple and direct multilingual interface, which lets us access its features in a fast and effective way. The customized interface allows us to adapt the characteristics of the camera for an optimal image pre-processing that will improve the results obtained by the OCR. Also, the advanced text post processing techniques developed by the Smart Mobile Software increase the effectiveness until the achievement of almost perfect results. It is not an open source application. It is compatible with IOS and Android devices. The application supports English languages. It works for digital as well as printed text. The application is lighter than other competitors. One of the cons for this application with respect to the proposed system is that the app needs to download app data for each language. The processing of the overall system is slow. Working is as simple as taking a picture, from the camera, of a document that you would like to OCR, and e-mail the image to yourself or share it. You will receive the image as well as the text file that contains the editable text that is extracted from the image. This currently supports English documents only.

The reader clicks image and submits all images to the administrator were after pre-processing operation of text extraction form prepare training data for English script is described in which is what we followed to prepare the training data. Initially training data make them able to deal with English language and UTF-8 character. Tesseract can handle any Unicode characters (coded with UTF-8), but there are limits as to the range of languages that it will be successful. Tesseract needs to know about different shapes of the same character by having different fonts separated explicitly. The number of fonts is limited to 64 fonts images on desktop computer bill has been generated. The drawback of this solution was that it was a time-consuming process and required high configured so this solution was not practically applicable.

Fig. 2. Training Data sets

However, to resolve the problem related to recent human work, a practical relevant solution was required. Hence it is efficient to utilize Android devices for fast processing. The proposed technology incorporates android application and web application. It helps reduce most of the human work and make process fast and computerized. Android application computerizes the procedure of meter reading and updating server with meter reading. Web application enhances the coordination of Dealer with Banks.

IV. PROPOSED SYSTEM

In this work, a huge number of colour and gray scale images are taken and it is converted into text format which can be readable to the user. The results are being computed in tabular form and comparing the analysis on Tesseract and Transym. The algorithm used here is Segmentation algorithm where each and every character is being segmented and recognised individually using data path where individual characters are being compared with the characters in data path. This algorithm involves ANN (Artificial Neural Networks) techniques. Processing of the image data is also performed concurrently. The output is then displayed in text format. The readable text is then sending to the web service and the text is then searched in the database of the bank. The response is being provided to the user of the API interface. The paper takes a huge number of images as input and processes it into editable text format using Tesseract library. The results are compared with that of Transym OCR. The accuracy of each image in coloured or gray scale format is analysed in tabular form.

V. ARCHITECTURE OF TESSERACT

Tesseract OCR works in step by step manner per the block diagram shown in Fig. 2. First step is Adaptive Thresholding, which convert the image into binary images. Next step is connecting component analysis, which is used to extract character outlines. This method is very because it does the OCR of image with white text and black background.

Tesseract was most likely first to impart this kind of processing. After that the outlines are converted into Blobs. Blobs are organized into text lines, and the and regions are analysed for some fixed area or equivalent text size.

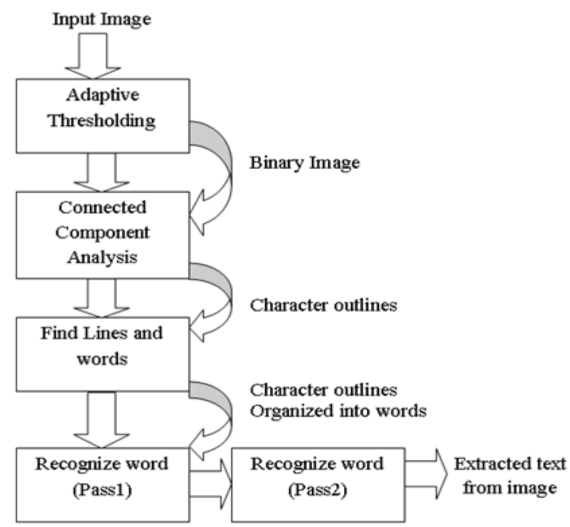

Fig. 3. Architecture of OCR

Text is fragmented into words using definite spaces and fuzzy spaces. Recognition of text is then started as two-pass process as shown in Fig. 3. In the first pass, an attempt is made to recognize each word from the text. Each word passed satisfactory is passed to an adaptive classifier has received some training data it has learn something new so final phase is used to resolve various issues and to extract text from images.

Algorithm to convert colour image to gray scale image

The image is converted to the gray scale by using the algorithm that is discussed in the following section. After processing the above image output is shown in the Fig. 5. It is clearly visible that Tesseract has successfully retrieved the text “Business”, which is expected as output. This means that Tesseract has extracted the text with 100% accuracy.

The following section contains algorithm to convert any colour image to the gray scale image. Here in this algorithm, a digital image with dimensions M width (row) and N height (column) is represented as discrete function f(x, y) as:

$$(x, y) = (x_i, y_j), \text{ where } i = 0, i < N, j = 0, j < M \quad (1)$$

Here the pair (x_i, y_j) is known as pixel. The pair (0, 0) is the first pixel and pair (M-1, N-1) is the last pixel in the image. Every pixel has its own RGB colour value. If the pixel has the same RGB value then it falls into gray colour family (black to white). So, based on this observation the algorithm to convert colour image to gray scale is developed, which is shown under.

$$\mu(x,y) = \frac{\Sigma((x,y)r, (x,y)g, (x,y)b)}{3}\ \forall(x,y)\ \text{where} -1 < r,g,b < 256 \quad (2)$$

Here r, g and b are red colour, green colour values of pixel (x, y) respectively. Range of r, g and b are mentioned and $\boldsymbol{\mu}$ is the mean value of these pixels that is always less than 256. So, value of $\boldsymbol{\mu}$ is in between o to 255, which is assigned to red, green and blue pixels of pixel (x, y). This process is depicted in

$$(x,y)r = \mu(x,y)\forall(x,y)w\square ere\ x \in N,\ y \in M \quad (3)$$

$$(x,y)g = \mu(x,y)\forall(x,y)w\square ere\ x \in N, y \in M \quad (4)$$

$$(x,y)b = \mu(x,y)\forall(x,y)w\square ere\ x \in N, y \in M \quad (5)$$

So after applying above algorithm on colour image it becomes gray scale image.

VI. METHODOLOGY

The procedure involved here is obtaining input images and performing scan and crop of the image respectively. By applying different OCR techniques for the image, the corresponding text is being retrieved by the system.

Working of Tesseract OCR using auto Business

An image with the text is given as input to the Tesseract engine that is android based. The image form camera, then it is processed by Tesseract android crop image as shown in Fig. 4.

Tesseract takes two arguments: First argument is image file path or camera that contains text and second argument is output text file where the extracted text is displayed. The extension for the output file is given as text by Tesseract, so while specifying the output file name as a second argument in Tesseract in android, there is no reason to indicate file extension.

As Tesseract supports different languages, the language training data file must be kept in the tessdata folder. In this research, the purpose is to extract English text from the images so we have kept only English language file in the tessdata folder. After the processing of image is finished, the content of the output file is shown in Fig. 5. In simple images with or without colour (gray scale), Tesseract provides results with 100% accuracy. But in the case of some complex images Tesseract provides better accuracy results if the images are in the gray scale mode as compared to colour images and gray scale images and gray scale images is performed and in both cases different result are achieved.

Fig. 4. Image camera

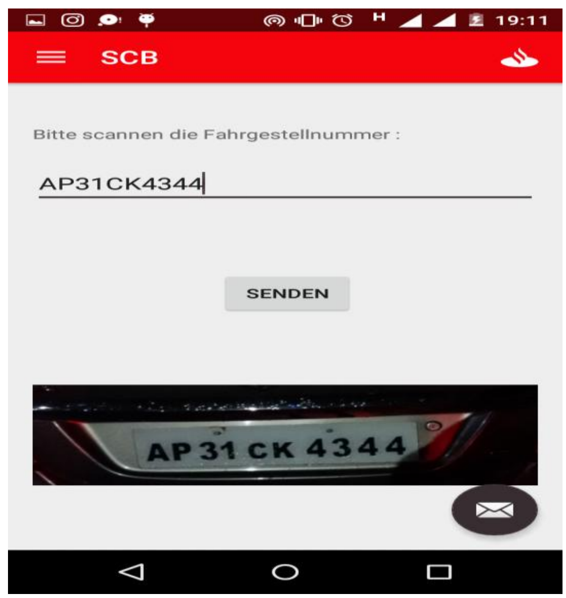

Fig. 5. Output of processing done by Tesseract

A) OCR of colour image by Tesseract

OCR of a complex colour image shown in the Fig. 6. As performed by Tesseract and after OCR processing of image the text extracted in the image is not as accurate as it is expected. As it visible in the Fig. 7, the extracted text is not exactly same as it is visible in the image of Fig. 6. That means Tesseract is not able to extract the text with 100% accuracy. So, to get more precise outcome, the same image is converted into gray scale image and OCR is performed on this image. The experiment result is discussed in the following section.

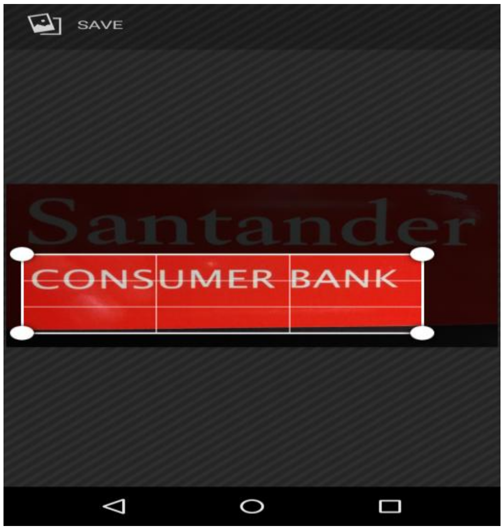

Fig. 6. Colour Image

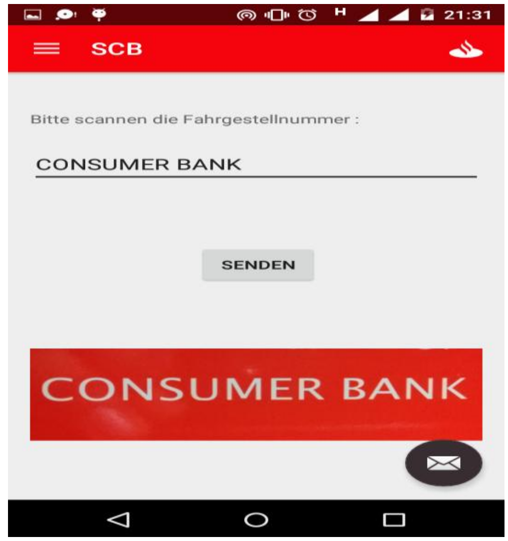

Fig. 7. Colour Image Recognition

Table I: Tesseract OCR Result analysis

Image NO	Image Type	Number of character in image	No of characters extracted	Accuracy of OCR of colour images (in Percentages)	Time taken for OCR (in Seconds)	Image Type	No of characters extracted after converting colour to gray scale image	Accuracy of OCR of gray scale image (in Percentages)	Time taken for OCR (in Seconds)	Change in Accuracy (In Percentage)
1	color	**12**	5	42	0.4	gray scale	5	42	0.397	
2	color	**12**	12	100	0.202	color	12	100	0.202	
3	color	**12**	8	67	0.301	gray scale	8	67	0.601	
4	color	**9**	9	100	0.5	color	9	100	0.5	
5	color	**8**	8	100	0.505	Color	8	100	0.505	
6	color	**9**	7	78	0.909	gray scale	7	78	0.909	
7	color	**8**	8	100	0.805	color	8	100	0.805	
8	Color	**9**	7	78	0.909	gray scale	7	78	1.01	
9	Color	**10**	7	70	0.85	gray scale	7	70	0.798	
10	**Color**	9	**4**	**44**	**0.907**	**gray scale**	**5**	**56**	**0.402**	**20**
11	**Color**	10	**1**	**10**	**1.007**	**gray scale**	**4**	**40**	**0.548**	**75**
12	**Color**	10	**4**	**40**	**0.699**	**gray scale**	**7**	**70**	**0.402**	**42.86**
13	**Color**	10	**3**	**30**	**1.51**	**gray scale**	**4**	**40**	**0.701**	**25**
14	**Color**	9	**0**	**0**	**1.008**	**gray scale**	**4**	**44**	**0.705**	**100**
15	**Color**	9	**0**	**0**	**1.815**	**gray scale**	**2**	**22**	**0.7**	**100**
16	**Color**	11	**6**	**55**	**1.619**	**gray scale**	**8**	**73**	**1.717**	**25**
17	**Color**	9	**5**	**56**	**0.99**	**gray scale**	**6**	**67**	**0.806**	**16.67**
18	**Color**	11	**9**	**45**	**0.907**	**gray scale**	**6**	**55**	**0.596**	**16.67**
19	Color	**9**	9	100	3.048	Color	9	100	3.048	
20	Color	**9**	9	100	1.007	Color	9	100	1.007	
			Average Accuracy	61			Average Accuracy	70		

The complex colour image is being taken and the required text is cropped along the x-y axis from the image. The word "Business" is then extracted from the image using Segmentation algorithm for performing pre-processing of the data path. Here, the data path is mainly denoted using CSV file. The extracted text is then segmented internally such that every single alphabets and letters is been pre-processed and is been detected so that the text can be recognised by the user. The data path is then used for pre-processing and the extracted text is then send as an output file as shown in the Fig. 7. The segmented letters are then compared with the letters on the data path and the recognized letters are displayed as an output. A number of images is being taken for analysis out of which the results of 20 images is shown in the Table I.

The Table II shows the result analysis of the same images using Transym OCR, Table III shows a brief comparison between the Tesseract and Transym OCR result obtained.

Table II: Transym OCR Result analysis

Image NO	**Image Type**	**Number of character in image**	**No of characters extracted**	**Accuracy of OCR of colour images (in Percentages)**	**Time taken for OCR (in Seconds)**
1	color	**12**	5	42	0.4
2	color	**12**	12	100	0.202
3	color	**12**	8	67	0.301
4	color	**9**	9	100	0.5
5	color	**8**	8	100	0.505
6	color	**9**	7	78	0.909
7	color	**8**	8	100	0.805
8	color	**9**	4	44	0.907
9	color	**10**	1	10	1.007
10	color	**9**	3	30	1.51
			Average Accuracy	**47**	

Table III: Tesseract and Transym OCR Comparison

Feature	**Tesseract OCR**	**Transym OCR**
Free	Yes	No (Trial Version is available)
Open Source	Yes	No
License	Apache	Proprietary
Online	No	No
Operating System	Window, Mac, Linux	Windows
Latest Stable version	3.01	3
Release Year	2010	2008
DLL Available	Yes	No
Accuracy (For extracting character from vehicle number plate)	61% (color images) 70% (gray scale images)	47%
Average Time	1 second (color images) 0.82 Seconds gray scale images)	6.75 seconds
□□ □C	34.21(For color Images) 24.64(For Gray Scale Images)	40
□T	0.61(For gray scale images scale images	12

σ_{AC} = Standard deviation of Accuracy

σ_{T} = Standard deviation of Time taken to perform OCR

B) OCR DATA in WCF Webservice

The WCF is Windows Communication Foundation web service which act like an interface between android application and server. With the help of WCF the data can be load into android app and extracted text in android app can be send to the server. The working of web services is shown in Fig. 8.

Fig. 8. Web Services

It is designed using service-oriented architecture standards to support distributed computing where service have remote consumers.

C) Experimental Results

We have captured more number of various types of number plates' images from different types of vehicles and performed OCR of these number plates to determine the vehicle number. Table I shows that Tesseract provides 61% accuracy with the colour images and 70% accuracy with gray scale images. So, it indicates that Tesseract provides better accuracy in gray scale images as compared to colour images. This experiment was carried out on computer with Intel core i5 4.20GHz CPU and 8 GB RAM. The pictures of number plates are captured by 5-mega-pixel camera. We can observe that if colour images are converted to gray scale and given as input to Tesseract than accuracy of text extraction is increased. In some colour images where text extraction accuracy result is 100% or near 100%, and if it is converted to gray scale then it produces same amount of extraction accuracy. In some colour images Tesseract is not able to provide more than 40% of accuracy, we have converted these images into gray scale images by using the algorithm discussed in the previous section and then given these images are taken as an input to Tesseract. So, after doing this process there is an increase in the average accuracy to extract the characters form the vehicle number plate. Notice that there is change UN accuracy from 16% to 100% as shown in Table 1 with the image numbers 10 to 18. Additionally, it is observed that the processing time of extracting characters from gray scale images is decreased. It is reduced by 10% to 50%. So, we can agree with the fact that Tesseract works fast and provides better text extraction accuracy in the processing of gray scale number plate images.

VII. CONCLUSION

In spite of the fact that Tesseract is command-based tool however as it is open source and it is available in the form of Dynamic Link Library, it can be easily made available in graphics mode. The results obtained in above sections are obtained by image extracting vehicle number from vehicle plate. The input images are specific, which are vehicle number plate, so in these specific images Tesseract provides better accuracy image to extract text in input to send server. The servers retrieve customer information

REFERENCES

[1]. ARCHANA A. SHINDE, D. 2012.Text Pre-processing and Text Segmentation for OCR. International Journal of Computer Science Engineering and Technology, pp. 810-812.

[2]. Aostopoulos,C.,Anagnostopoulos, I., Loumos, V, & Kayafas, E. 2006. A License Plate Recognition Algorithm for Intelligent Transportation System Applications.., IEEE Transactions on Intelligent Transportation Systems, pp. 377-399.

[3]. Y. WEN, Y. L. 2011. An Algorithm for License Plate Recognition Applied to Intelligent Transportation System, IEEE Transactions on Intelligent Systems, pp. 1-16.

[4]. XIN FAN, G. L. 2009. Graphical Models for Joint Segmentation and Recognition of License Plate Characters. IEEE Signal Processing Letters, pp. 10-13.

[5]. HUI WU, B. L. 2011. License Plate Recognition system. International Conference on Multimedia Technology (ICMT). pp. 5425-5427.

[6]. PAN, Y.-F., HOU, X., & LIU, C.-L. 2008. A Robust System to Detect and Localize Texts in Natural Scene Images. The Eighth IAPR International Workshop on Document Analysis Systems.

[7]. SMITH, R. 2007. An Overview of the Tesseract OCR Engine. In proceedings of Document analysis and Recognition.. ICDAR 2007. IEEE Ninth International Conference.

[8]. GOOGLE. Google Code. goggle code. [Online] 2012. http://code.google.com/p/tesseract-ocr/.

[9]. F. SHAFAIT, D. K. San Jose, CA: s.n., 2008.Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images.. In Document Recognition and Retrieval XV, S&T/SPIE Annual Symposium on Electronic Imaging.

[10]. 1stwebdesigner. 1stwebdesigner. [Online] 2012. http://www.1stwebdesigner.com/wp-content/uploads/2009/11/typography-tutorial/text1-how-to-create-typographic-wallpaper.jpg.

[11]. dsigninspire. Desing Inspire. [Online] 2012. http://dsigninspire.com/wpcontent/uploads/2011/ 09/moon-shine.jpg

[12]. Geometric Rectification of Camera-Captured Document Images. Jian Liang; DeMenthon, D.; Doermann, D.; April 2008. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.30, no.4, pp.591-605.

[13]. Y. WEN, Y. L. 2011, An Algorithm for License Plate Recognition Applied to Intelligent Transportation System. IEEE Transactions on Intelligent Systems, pp. 1-16.

[14]. Lihong Zheng, Xiangjian He, Bijan Samali, Laurence T. Yang. An algorithm for accuracy enhancement of license plate recognition. Journal of Computer and System Sciences, Available online 9 May 2012.

[15]. Deselaers, T.; Gass, T.; Heigold, G.; Ney, H.; Latent Log-Linear Models for Handwritten Digit Classification. June 2012. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, no.6, pp.1105-1117, doi: 10.1109/TPAMI.2011.218.

[16]. Jianbin Jiao, Qixiang Ye, Qingming Huang, A configurable method for multi-style license plate recognition. 2009. Pattern Recognition, Volume 42, Issue 3, , Pages 358-369.

[17]. H. Erdinc Kocer, K. Kursat Cevik. 2011.Artificial neural networks based vehicle license plate recognition. Procedia Computer Science, Volume 3, Pages 1033-1037.

[18]. Apurva A. Desai. 2010. Gujarati handwritten numeral optical character reorganization through neural network, Pattern Recognition, Vol. 43, Issue 7, Pages 2582-2589, ISSN 0031-3203, 10.1016/j.patcog.2010.01.008.

[19]. Roy, A.; Ghoshal, D.P. 2011. Number Plate Recognition for use in different countries using an improved segmentation, 2nd National Conference on Emerging Trends and Applications in Computer Science (NCETACS), pp.1-5, doi: .1109/NCETACS.2011.5751407.

[20]. Umapada Pal, Partha Pratim Roy, Nilamadhaba Tripathy, Josep Lladós. December 2010. Multi-oriented Bangla and Devnagari text recognition, Pattern Recognition, Volume 43, Issue 12,Pages4124-4136, 10.1016/j.patcog.2010.06.017.

[21]. Bilal Bataineh, Siti Norul Huda Sheikh Abdullah, Khairuddin Omar. 2011.An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows, Pattern Recognition Letters, Vol. 32, Issue 14, pp. 1805-1813, ISSN: 0167-8655, 10.1016/j.patrec.2011.08.001.

[22]. Fink, Gernot.2009. Markov models for offline handwriting recognition: a survey. International Journal on Document Analysis and Recognition. Springer Berlin/Heidelberg pp. 269-298, Vol. 12, Doi: 10.1007/s10032-009-0098-4