Research and implementation of license plate recognition based on android platform

Yueyue Sun[1,*], and *Xuechen* Zhao[2]

[1]Jinan Engineering Vocational and Technical College, Ji'nan, Shandong, China
[2]Shangdong, Woman,University, Ji'nan, Shandong, China

Keywords: OpenCV, Android, Tesseract-OCR, Laplace, HSV, jTessBoxEditor.

Abstract. This paper studies and optimizes license plate location and recognition in license plate recognition. A license plate recognition system based on Android platform is designed and implemented. Opencv and Tesseract OCR are integrated in Android studio environment. The license plate number is located by combining Laplace algorithm and HSV model. On the basis of fully understanding the principle of Tesseract OCR recognition, a large number of training pictures are generated by license plate number simulation generator, and license plate character library is generated by using jtessboxeditor tool, which realizes offline recognition of license plate number.

1 Introduction

In recent years, with the continuous improvement of modern economic level, cars have entered ordinary people's homes from the altar of luxury, and the explosion of the number of cars has also brought some difficulties to automobile management. Nowadays, license plate recognition has become one of the key tasks of vehicle management in every city. Effective, accurate and timely license plate recognition can facilitate traffic enforcement and vehicle management in parking lots. With the rapid popularization and development of smart terminals and 4G technology, the constant improvement of mobile phone configuration, the research on information collection, image processing and data transmission based on mobile platform has also become a hot spot, which makes license plate recognition based on mobile platform possible. Traditional license plate recognition system is generally based on a fixed desktop platform, and image acquisition is not flexible. Especially for traffic management departments, automatic registration of vehicle license plates in violation of regulations is very inconvenient, so license plate recognition based on the platform of Andriod has emerged. Mobile phone license plate recognition system, support Android, ios and other mainstream mobile operating systems. The camera of mobile phone and tablet computer was used to capture the image of vehicle license plate. OpenCV was first used to process the image of vehicle, and then training samples were collected and made to form a character library, which was imported into the tesseract-ocr recognition library for vehicle license plate identification.

[*] Corresponding author: sunyueyue_good@163.com

2 License plate character

The primary task of license plate recognition is to determine the position of the license plate from the input image and segment the license plate from the original image. Only after accurate license plate positioning can the accuracy of subsequent license plate recognition be guaranteed. China's motor vehicle license plate number has certain characteristics, according to these characteristics can better choose the way and method of positioning. According to the latest document ga36-2014 of the ministry of public security, the main size of motor vehicles in China is 440×140 and 440×220, and the color is mainly white characters on blue background, white frame line, black characters on yellow background and black frame line.

This paper, only for small car plate, which can identify small car plate, white color is blue background, the content is composed of seven characters and an interval, including the Chinese characters, upper case letters and Numbers, license plate from left to right in turn is the shortened form of the provinces, autonomous regions and municipalities directly under the central government, the licensing authority code, interval operator and the composition of five serial number coding. Therefore, the following characteristics of the license plate of small cars in China are briefly summarized: the color is blue with white characters, the license plate size is 440×140, and the aspect ratio is about 3:1.

3 Graphic preprocessing

From a picture of a car taken by a mobile phone, the license plate can be easily picked up by the human eye. This is because the color of the license plate is so contrasting with other backgrounds that it is easy to locate the license plate. The main idea of this paper is to find the pixel points with strong changes through the edge detection by looking for the differences between the pixel points, then locate the boundary, extract each contour in the picture, and screen through the characteristics of the license plate number. This paper uses OpenCV library, which needs to be built in the development environment of Android studio.

3.1 Filtering noise reduction

Generally, images obtained will have different degrees of noise, so it is necessary to conduct noise reduction, and smooth operation can be used to reduce the image noise, because in the image, the noise energy is mostly concentrated in the low and medium frequency part of the amplitude spectrum, while in the higher frequency band, some important details are often drowned by the noise. In an image, the high frequency part refers to the image pixel values divide a large part of the low frequency is next to the pixel value and pixel values were the same, even some of the detail of the part of the picture, and often show by high frequency information, doping in the image noise is often at high frequencies, this creates some details information submerged by noise, can according to different types of noise in different filter processing [6-8]. In order to smooth the image, a filter is usually added to the image. The most common types are linear gaussian filtering and nonlinear bilateral filtering. In the following comparison, it is obvious that the bilateral filter can filter out the noise of the image and save the edge information. According to the characteristics of histogram, the noisy histogram is gentle, continuous and sharp with high definition. bilateral filter is used for denoising.

3.2 Edge detection

After noise reduction, the image is turned to gray level for edge detection. Edge extraction algorithms commonly used in OpenCV library include Sobel, Scharr, Laplace and Canny. The first three of them have directional edge detection. Scharr operator and Canny detection added more details, but for license plate recognition, only the part of the license plate should be highlighted, while the other part should be weakened. It is convenient to locate the license plate number. In [2-5], the vertical edges of a car image using image enhancement and a Sobel operator were employed, followed by removing most of the background and noise edges and searching for the plate region using a rectangular window. Through contrast, it is found that since the edge of the image is the region where the gray level changes, Laplace sharpening template has a good effect in edge detection. In general, it is difficult to determine the position of edge lines for steep edges and slowly changing edges. However, the operator can be determined by the crossing of the zero between the positive and negative peaks, which is more sensitive to outliers or endpoints. Therefore, it is especially suitable for the purpose of highlighting outliers, outliers or line endpoints in the image.The Laplace operator is used in this paper.

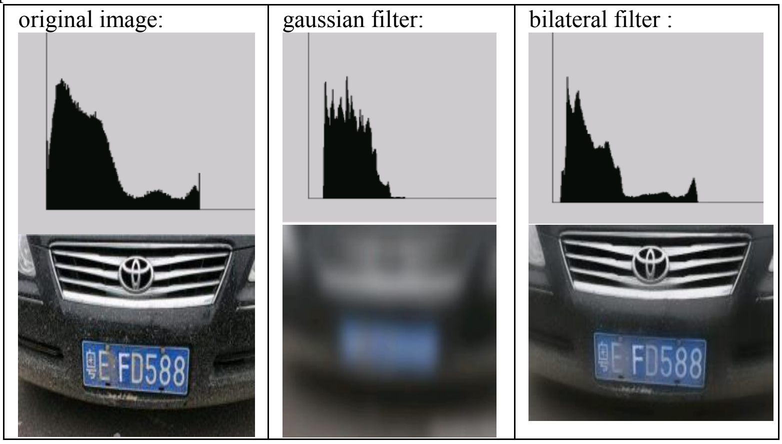

Fig. 1. Graphic preprocessing.

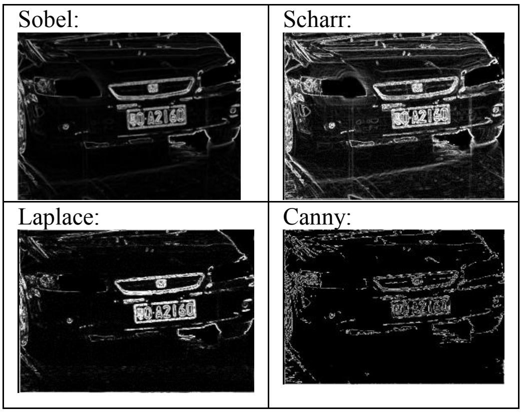

Fig. 2. Edge detection.

3.3 Morphological processing

In OpenCV, the contour extracted by the findcontour function in OpenCV is generally used for contour extraction. The algorithm principle is based on satoshi suzuki, published a paper [1], also known as Suzuki85 contour tracking algorithm, which mainly extracts the connected domain boundary of binarization image when extracting the contour. Therefore, the whole license plate is expected to form a connected domain. The connected domain boundary extracted in this way is also the boundary of the license plate, so it needs to be expanded and corroded morphologically. Techniques based upon combinations of edge statistics and mathematical morphology [6-9] featured very good results. Corrosion and swelling are for the white part (highlighted part), not the black part. Expansion refers to the expansion of the highlighted part of the image, "field expansion". The effect diagram has a larger highlighted area than the original image. Corrosion is the corrosion of the highlighted parts of the original image, "nibbled area", the effect of the image has a smaller highlighted area than the original image. The operation of first eroding and then expanding is called opening operation. It has the effect of eliminating small objects, separating objects in slender places and smoothing the boundaries of larger objects. The operation of expanding first and then corroding is called a closed operation. It has the function of filling tiny holes in objects, connecting neighboring objects and smooth borders. This paper deals with closed operations.

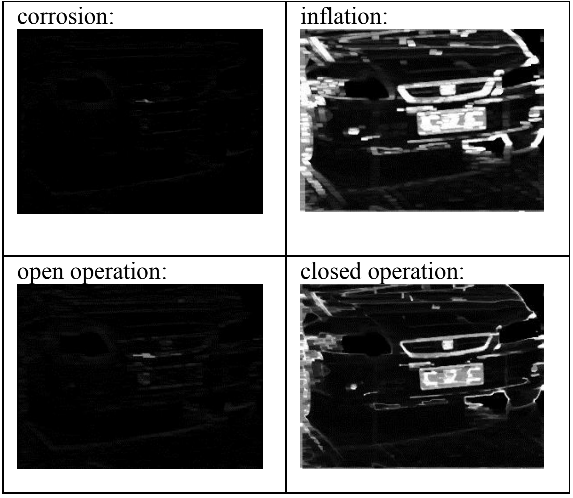

Fig. 3. Morphological processing.

3.4 Binaryzation

The binarization of the image is to set the gray value of the pixels on the image to 0 or 255, which will make the whole image show a clear black and white effect. Binary image plays a very important role in digital image processing. The binarization of the image greatly reduces the amount of data in the image, which can highlight the outline of the target. OpenCV provides functions of global fixed threshold and local adaptive threshold to realize binarization image. When performing global binarization, you can set a threshold yourself, or you can determine an optimal threshold through OTSU or triangle algorithm. Similar methods appear in [10-13], where the vehicle image is scanned with N-row distance, counting the existent edges. The OTSU method is more commonly used.

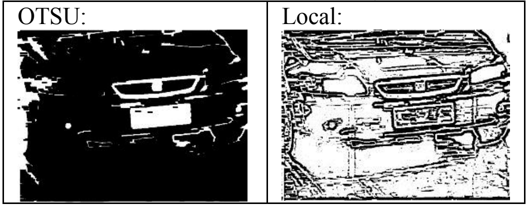

Fig. 4. Binaryzation.

4 Contour extraction

After completing the binarization, you can call the findContours function in OpenCV to extract the contours, and use the minAreaRect function to extract the smallest circumscribed rectangle. The obtained contours will be preliminarily filtered. In the previous description, we know that the width height ratio of the license plate is about 3:1, but sometimes because of the angle deviation between the license plate and the shooting position, the minimum circumscribed rectangle ratio cannot be accurate Fixed. Therefore, if the area of the external rectangle is larger than one thousandth of the input picture, it is initially considered as the license plate. If it is smaller than one thousandth, then even the license plate will not be recognized clearly due to its low resolution.

At this point, after screening, we found that there are still some invalid contours to be selected, and then the specific license plate number is located by color, and HSV model is used for color location. HSV model is a cone model created according to the intuitive properties of color. Different from RGB color model, where each component represents a color, HSV does not represent a color. Instead, each component represents: hue (H), saturation (S), and brightness (V). Therefore, it is necessary to combine the two methods to improve the accuracy of positioning. In this case, the overlap area of color positioning and Laplace positioning was identified as the license plate, which has a good effect in most cases. The overall process is as follows: Fig. 5.

5 Identification and verification of license plate number

Tesseract is an open source OCR (Optical Character Recognition) engine developed by HP LABS and maintained by Google. Compared with Microsoft Office Document Imaging (MODI), Tesseract is a library that can be continuously trained to enhance the ability of image to transform text. Tesseract is not only available on Windows, it's also widely available on android. Now Tesseract supports the recognition of more than one hundred languages, and its latest training data has been completed based on the neural network learning LSTM model. The accuracy and speed of recognition have been improved to varying degrees. In [14-16], by training your own data set, the recognition rate can reach more than 95%. Not only supports integration with OpenCV, Python and other language frameworks, but also supports the introduction and use of the Android platform, which greatly facilitates mobile Android developers to develop applications. Currently Tesseract open source is maintained on GIHUB source code. The address supported by the Android version is https://github.com/rmtheis/tess-two, where the pre-trained models for various language versions are located at https://github.com/ tesseract-ocr / tessdata. To integrate Tesseract into Android, you only need to add dependency support in the build.grad file corresponding to the app module.

5.1 Character library trainning

Tesseract's official website provides a rich character library, including sim.traineddata, eng.traineddata and nums.traineddata. However, the font of license plate number includes the combination of Chinese and English numbers. The Chinese characters are not all Chinese characters, but only Beijing, Tianjin, Shanghai, Chongqing, Hebei, Henan, Yunnan, Liaoning, Hei, Xiang, Anhui, Shandong, Xin, Su, Zhejiang, Gan, Hubei, Gui, Gan, Jin, Mongolia, Shaanxi, Ji, Fujian, Gui, Guangdong, Qing, Tibet, Sichuan, Ning, Qiong. Tesseract engine allows users to generate their own character library, so that the character library has a high recognition rate in a certain category of recognition. The license plate simulator is used to generate 200 license plates with a fixed blue background, and then the jTessBoxEditor tool is used to combine the 200 valid license plate images into a TIFF image file. The process of training character library with jtessboxeditor is as follows: Fig. 6.

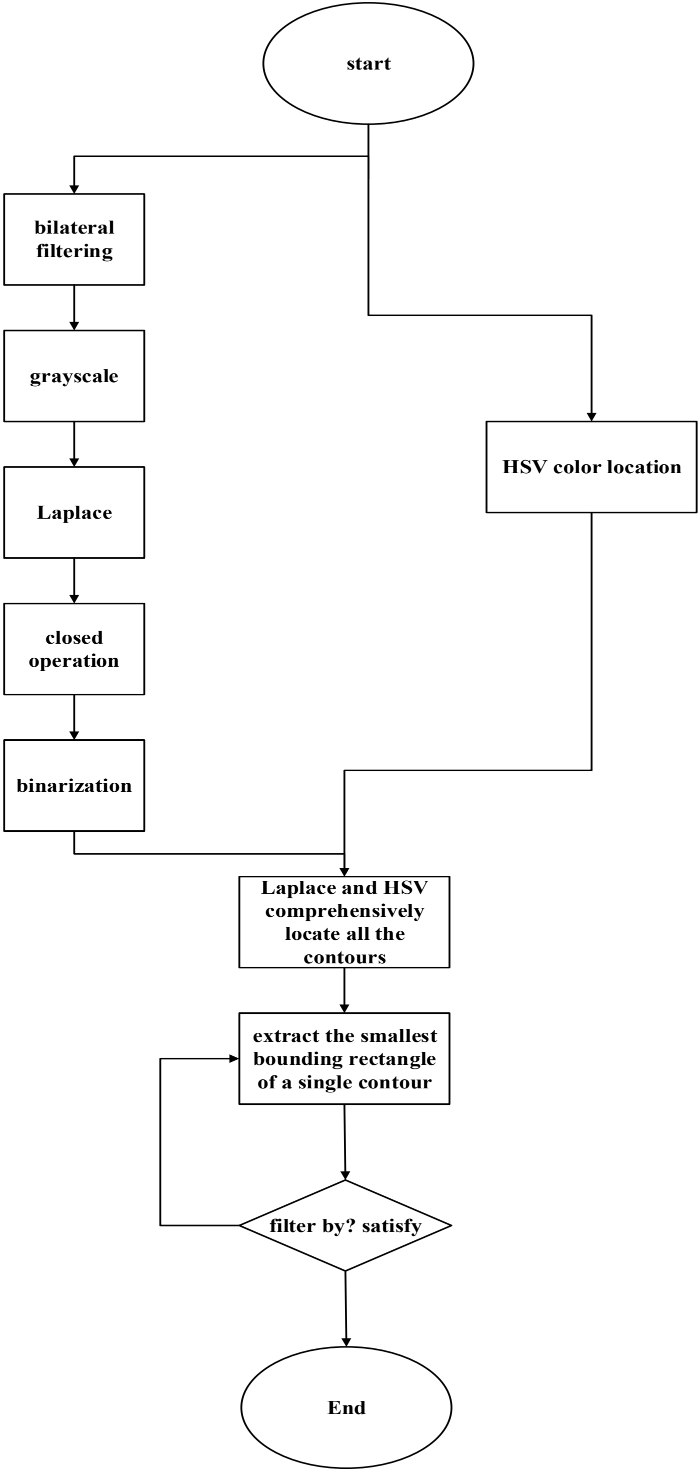
start
bilateral filtering
grayscale
Laplace
closed operation
binarization
HSV color location
Laplace and HSV comprehensively locate all the contours
extract the smallest bounding rectangle of a single contour
filter by? satisfy
End

Fig. 5. The overall process.

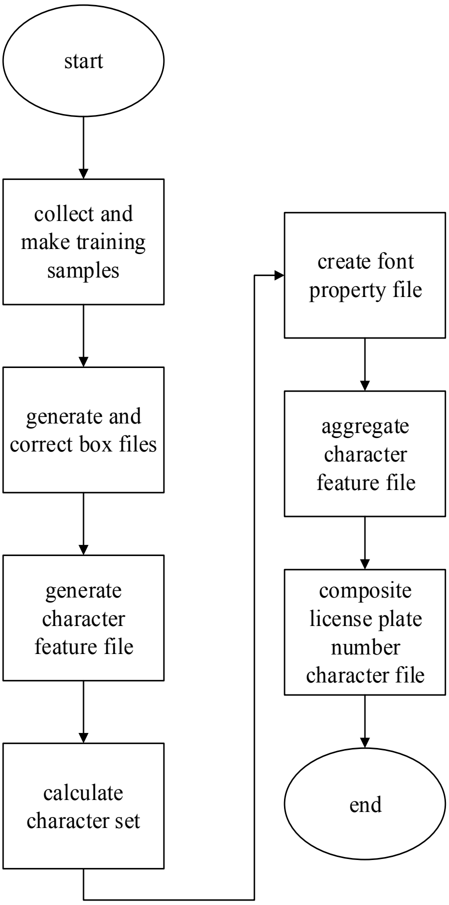
start
collect and make training samples
generate and correct box files
generate character feature file
calculate character set
create font property file
aggregate character feature file
composite license plate number character file
end

Fig. 6. Character library trainning.

(1) Collect and make training samples: The license plate simulation generator is used to generate 150 images, and the resulting license plate is set to be all fixed on a blue background, including the license plates of all provinces. Use jTessBoxEditor tool to combine these 150 pictures into a TIFF format picture file. The source picture set supports multiple formats such as jpg, tif, png, bitmap, etc. Tif file naming format [lang]. [fontname] .exp [num], where lang is the language, fontname is the font, and num is a custom number. So the format of the composite picture set is [car num]. [test]. [exp0]. tif.

(2) Generated and corrected box file: open the command line and input the command: D: \ jtesboxeditorfx \ testeract OCR \ testeract.exe car_num.test.exp0.tif car_num.test.exp0 - L chi_sim + eng batch.nochop makebox, where the - L chi_sim + eng parameter is to use the existing Chinese and English training word library, which is in the testsdata directory and can be copied in by yourself. Open the jTessBoxEditor tool and open the car_num.test.exp0.tif file (the .box and .tif sample files generated in the previous step must be placed in the same directory). You can see that some characters are not correctly segmented and recognized. You can manually use this tool. Correct the incorrectly recognized characters in each picture and save it after the correction is complete.

(3) Generate character feature file: After adjusting the box file, you need to generate the tr file D:\jTessBoxEditorFX\tesseract-ocr\tesseract.exe car_num. Test. Exp0. Tif car_num. Test. Exp0 nobatch box.train, where car_num. Test. Exp0.tr is the character feature file.

(4) Calculate character set: Generate a unicharset file and execute the command D: \ jTessBoxEditorFX \ tesseract-ocr \ unicharset_extractor.exe car_num.test.exp0.box, where the unicharset file is a character set file that contains all characters in the training sample. Open the unicharset file, which contains 31 Chinese characters and 10 digits and 23 uppercase English characters.

(5) Create font property file: Use Notepad to create a new plain text font_properties.txt with the content format of < fontname > < Italic > < bold > < fixed > < serif > < Fraktur >, where fontname represents the file name, italic represents italics, bold represents bold, fixed represents fixed, serif represents serif, Fraktur represents decorative arc, and 1 or 0 respectively indicates whether it has this property. The format is defined as: Test 0 000 0, test is the content in the middle of the new TIF (car_num.test.exp0. tif).

(6) Aggregate character feature file: Run three commands to operate the tr feature set: generate shape file D: \ jTessBoxEditorFX \ tesseract-ocr \ shapeclustering.exe -F font_properties.txt -U unicharset car_num.test.exp0.tr; generate the aggregate character feature file D: \ jTessBoxEditorFX \ tesseract-ocr \ Mftraining.exe -F font_properties.txt -U unicharset -O unicharset car_num.test.exp0.tr; Generate character normalization feature file D: \ jTessBoxEditorFX \ tesseract-ocr \ cntraining.exe car_num.test .exp0.tr.

(7) Composite license plate number character file: Rename the unicharset, inttemp, pffmtable, shapetable and normroto files in the directory, and add test (the middle name of TIF) before the name. Execute the following command D: \ jtessboxeditorfx \ testeract OCR \ combine \ testsdata test to generate the test.traineddata training library file.

6 Summary

This article integrates the OpenCV library in the environment of Andriod Studio, implements the license plate screening, and uses Tesseract-OCR to identify the license plate number screened out. This method first uses the Laplacian algorithm to extract contours from the original image, and then selects contours that meet certain features based on the characteristics of the license plate number, and then uses the HSV model to eliminate fake license plate numbers. In order to improve the recognition rate of tesseract-ocr, because the characterset provided by the government is relatively single, training samples are made based on the jTessBoxEditor tool provided by tesseract-ocr, and finally the recognition rate of character set in license plate number is improved. In the whole research process, the following issues need to be further discussed in the future.

Because the HSV model is greatly affected by ambient light, when setting the range of the HSV, it cannot be done all at once and needs to be constantly adjusted to adapt to the light, or the range is set to be larger so that it contains blue with different light. However, if the body color is blue, the HSV range needs to be fine-tuned so that it can filter out fake license plate numbers.

References

1. Satoshi Suzuki, KeiichiA be .Topological structural analysis of digitized binary images by border following.1985, 10. 1016/0734-189X (85)90016-7.
2. F. Yang and Z. Ma, “Vehicle license plate location based on histogramming and mathematical morphology,” in Proc. 4th IEEE Workshop Autom. Identification Advanced Technol., 2005, pp. 89–94.
3. P. V. Suryanarayana, S. K. Mitra, A. Baneree, and A. K. Roy, “A mor-phology based approach for car license plate extraction,” in Proc. IEEEIndicon, Chennai, India, Dec. 2005, pp. 24–27.

4. H. Mahini, S. Kasaei, F. Dorri, and F. Dorri, “An efficient features–based license plate localization method,” in Proc. 18th ICPR, Hong Kong, 2006, vol. 2, pp. 841–844.
5. S. Kim, D. Kim, Y. Ryu, and G. Kim, “A robust license-plate extraction method under complex image conditions,” in Proc. 16th Int. Conf.Pattern Recog., Quebec, QC, Canada, 2002, vol. 3, pp. 216–219.
6. F. Martin, M. Garcia, and J. L. Alba, “New methods for automatic reading of VLP’s (Vehicle License Plates),” in Proc. IASTED Int. Conf.SPPRA, 2002.
7. B. Hongliang and L. Changping, “A hybrid license plate extraction method based on edge statistics and morphology,” in Proc. ICPR, 2004, pp. 831–834.
8. D. Zheng, Y. Zhao, and J. Wang, “An efficient method of license plate location,” Pattern Recognit. Lett., vol. 26, no. 15, pp. 2431–2438,Nov. 2005.
9. S. Wang and H. Lee, “Detection and recognition of license plate characters with different appearences,” in Proc. Conf. Intell. Transp. Syst., 2003, vol. 2, pp. 979–984.
10. S. Draghici, “A neural network based artificial vision system for license plate recognition,” Int. J. Neural Syst., vol. 8, no. 1, pp. 113–126,Feb. 1997.
11. A. Broumandnia and M. Fathy, “Application of pattern recognition for Farsi license plate recognition,” in Proc. Int. Conf. GVIP, Cairo, Egypt,2005.
12. T.-H. Wang, F.-C. Ni, K.-T. Li, and Y.-P. Chen, “Robust license plate recognition based on dynamic projection warping,” in Proc. IEEE Int. Conf. Netw., Sens. Control, 2004, pp. 784–788.
13. J. Cano and J. C. Perez-Cortes,Vehicle License Plate Segmentation in Natural Images, vol. 2652, F. J. Perales et al., Eds. New York:Springer-Verlag, 2003, pp. 142–149.
14. Sagar, B.M. , Shobha, G. , Kumar, P.R. , “Complete Kannada Optical Character Recognition with syntactical analysis of the script”, IEEE 1 – 4 , Dec. 2008.
15. Gita Sinha Rajneesh Rani Renu Dhir, “Handwritten Gurmukhi Numeral Recognition using Zone-based Hybrid Feature Extraction Techniques”, International Journal of Computer. Applications (0975-8887), Volume 47- No. 21 June 2012.
16. Jalal Uddin Mahtnud, Mohammed Feroz Raihan and ChowdhuryMofizur Rahman, “A Complete OCR System for Continuous Bengali Characters”,IEEE 1372 - 1376 Vol. Oct. 2003.