ISSN: 2321-7782 (Online)
Volume 2, Issue 11, November 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Survey on Text Extraction in Android Mobile Application using Character Descriptor

Prajakta D. Sawle[1]
M.E Second Year CSE
P. R. Patil COET, Amravati
Maharashtra – India

Prof. Chetan J. Shelke[2]
Department of Information and Technology
P. R. Patil COET, Amravati
Maharashtra – India

Abstract: Images contain various types of useful information that should be extracted whenever required. A large number of algorithms and methods are proposed to extract text from the given image, and because of that a user will be able to access the text from any image. Extraction of this information involves text region detection, text localization, tracking, character extraction, enhancement, and recognition of the text from a given image. Variations in text may occur because of differences in size, style, orientation, alignment of text, and low image contrast, composite backgrounds make the problem during extraction of text. In text detection, our previously proposed algorithms are applied to obtain text regions from scene image. First, we design a discriminative character descriptor by combining several state-of-the-art feature detectors and descriptors. Second, we model character structure at each character class by designing stroke configuration maps. Our algorithm design is compatible with the application of scene text extraction in smart mobile devices. In this paper, we used cloud based API to increase the speed and accuracy of the recognition.

Keywords: Text extraction, Text detection, Text localization, Text retrieval, OCR.

I. INTRODUCTION

Text in the image contains useful information which helps to acquire the overall idea behind the image. Character extraction from image is important in many applications. It is a difficult task due to variations in character fonts, sizes, styles and text directions, and presence of complex backgrounds and variable light conditions. Several methods for text (or character) extraction from natural scenes have been proposed. If we develop a method that extracts and recognizes those texts accurately in real time, then it can be applied to many important applications like document analysis, vehicle license plate extraction, text-based image indexing, etc and many applications have become realities in recent years [1].

To extract text information by mobile devices from natural scene, automatic and efficient scene text detection and recognition algorithms are essential. However, extracting scene text is a challenging task due to two main factors: 1) cluttered backgrounds with noise and non-text outliers, and 2) diverse text patterns such as character types, fonts, and sizes.

The frequency of occurrence of text in natural scene is very low, and a limited number of text characters are embedded into complex non-text background outliers. Background textures, such as grid, window, and brick, even resemble text characters and strings. Although these challenging factors exist in face and car, many state-of-the-art algorithms [2], [3] have demonstrated effectiveness on those applications, because face and car, have relatively stable features.

Text data is particularly interesting, because text can be used to easily and clearly describe the contents of an image [4]. A variety of applications are found in recent studies that uses extracted text. One such recently developed application is the mobile banking application provided by the banking institutions that facilitates the customers to carry out the transactions even on passing the image of the cheque to the server [5]. A translation camera is another application that captures the images, detect the text from it and then translate it in required language from the local language. This can also be useful for visually impaired people if text to speech program (TTS) is used within Text Detection and Localization Recognition (OCR) [6][7].

II. Related Work

Ezaki et al. [8] propose four character extractions methods based on connected components. The performance of the different methods depends on character size. The most effective extraction method proves to be the sequence: Sobel edge detection, Otsu binarization, connected component extraction and rule-based connected component filtering. Yamaguchi et al. [9] propose a digits classification system to recognize telephone numbers written on signboards. Candidate regions of digits are extracted from an image through edge extraction, enhancement and labeling. Since the digits in the images often have skew and slant, the digits are recognized after the skew and slant correction. To correct the skew, Hough transform is used, and the slant is corrected using the method of circumscribing digits with tilted rectangles.

In the work of Matsuo et al. [10] a method is proposed that extracts text from scene images after an identification stage of a local target area and adaptive thresholding. Yamaguchi and Maruyama [11] propose a method to extract character regions in natural scene images by hierarchical classifiers. The hierarchy consists of two types of classifiers: a histogram-based classifier and SVM. Finally, Yang et al. [12] have proposed a framework for automatic detection of signs from natural scenes. The framework considers critical challenges in sign extraction and can extract signs robustly under different conditions (image resolution, camera view angle, and lighting).

We observe that text characters from different categories are distinguished by boundary shape and skeleton structure, which plays an important role in designing character recognition algorithm. Current optical character recognition (OCR) systems [13], [14] can achieve almost perfect recognition rate on printed text in scanned documents, but cannot accurately recognize text information directly from camera-captured scene images and videos, and are usually sensitive to font scale changes and background interference which widely exists in scene text. Fig.1 shows optical Character Recognition process. Although some OCR systems have started to support scene character recognition, the recognition performance is still much lower than the recognition for scanned documents. Many algorithms were proposed to improve scene-image-based text character recognition. Although some OCR systems have started to support scene character recognition, the recognition performance is still much lower than the recognition for scanned documents. Many algorithms were proposed to improve scene-image-based text character recognition. Weinman *et al.* [15] combined the Gabor-based appearance model, a language model related to simultaneity frequency and letter case, similarity model, and lexicon model to perform scene character recognition.

Neumann *et al.* [16] proposed a real time scene text localization and recognition method based on extremal regions Smith *et al.* [17] built a similarity model of scene text characters based on SIFT, and maximized posterior probability of similarity constraints by integer programming. Mishra *et al.* [18] adopted conditional random field to combine bottom-up character recognition and top-down word-level recognition. Lu *et al.* [19] modeled the inner character structure by defining a dictionary of basic shape codes to perform character and word retrieval without OCR on scanned documents. Coates *et al.* [20] extracted local features of character patches from an unsupervised learning method associates with a variant of K-means clustering, and pooled them by cascading sub-patch features.

In [21], a part-based tree structure model was designed to detect text characters under Latent-SVM [22], and recognize text words from text regions under conditional random field. In [23], Scale Invariant Feature Transform (SIFT) feature matching was adopted to recognize text characters in different languages, and a voting and geometric verification algorithm was presented to filter out false positive matches. In [24], generic object recognition method was imported to extract scene text information. A dictionary of words to be spot is built to improve the accuracy of detection and recognition. Character structure was modeled by HOG features and cross correlation analysis of character similarity for text recognition and detection. In [25], Random Ferns algorithm was used to perform character detection and constructed a system for query-based word detection in scene images.

Kim [26] has proposed an approach in which LCQ (Local Color Quantization) is performed for each color separately. Each color is assumed as a text color without knowing whether it is real text color or not. Color quantization takes place before processing to reduce blight, an input image is converted to a 256-color image when they show features the text field text lines to find candidates, connected components that are extracted for each color merged. LCQ for each color is executed since this drawback of the method is processing time.

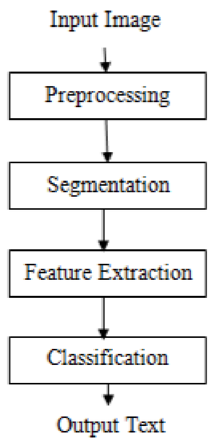

Fig.1 Optical Character Recognition

III. Challenges of Scene Text Detection

First of all, in order to understand challenges of this field, new imaging conditions and newly considered scenes need to be detailed:

A. Raw sensor image and sensor noise: in low-priced HIDs, pixels of a raw sensor are interpolated to produce real colors, which can induce degradations. Demos icing techniques, viewed more as complex interpolation techniques, are sometimes required. Moreover, sensor noise of an HID is usually higher than that of a scanner.

B. Angle: scene text and HIDs are not necessarily parallel creating perspective to correct.

C. Blur: during acquisition, some motion blur can appear or be created by a moving object. All other kinds of blur, such as wrong focus, may also degrade even more image quality.

D. Lighting: in real images, real (uneven) lighting, shadowing, reflections onto objects, interreflections between objects may make colors vary drastically and decrease analysis performance.

E. Resolution and Aliasing: from webcam to professional cameras, resolution range is large and images with low resolution must also be taken into account. Resolution may be below 50 dpi which causes commercial OCR to fail. It may lead to aliasing creating fringed artifacts in the image.

F. *Outdoor/non-paper objects:* different materials cause different surface reflections leading to various degradations and creating inter-reflections between objects.

G. Scene text: backgrounds are not necessarily clean and white, and more complex ones make text extraction from background difficult. Moreover scene text such as that seen in advertisements may include artistic fonts.

H. *Non-planar objects:* text embedded in the bottles or cans suffer from deformation.

I. Unknown layout: priori information not available on structure of text to detect it efficiently.

J. Objects in distance: space between text & HIDs can vary, & character sizes may vary in a wide range, leading to a wide range of character sizes in a same scene [27].

IV. Different Text Detection Approaches

A. Connected Component Based-It groups neighbouring pixels of similar colours into connected components and grouping small components into successively larger components until all regions are identified in the image.

Drawback--Not robust because they are based on geometrical properties of components

B. Edge-based methods -The edges of the text boundary are identified and merged, and then several heuristics are used to filter out the non-text regions.

Drawback- -It give more false positives when the complex background present.

C. Texture-based methods –It use the observation that text in images have distinct textural properties that distinguish them from the background. Can be used to detect of a text region in an image.

Drawback- -May be unsuitable for small fonts and poor contrasting text [28].

V. Proposed Work

In this paper we used cloud based API to increase speed and accuracy of recognition.

A. Cloud API

Cloud APIs are segmented into infrastructure, service and application clusters. Applications typically combine these APIs as needed.

i) Infrastructure

Infrastructure APIs modify the resources available to operate the application. Functions include provisioning (creating, re-creating, moving, or deleting components - such as virtual machines) and configuration (assigning or changing attributes of the architecture such as security and network settings). These components and their common use is referred to as infrastructure as a service (IaaS).

ii) Service

Service APIs provide an interface into a specific capability provided by a service explicitly created to enable that capability. Database, messaging, web portals, mapping, e-commerce and storage are all examples of service APIs. These services are commonly referred to as platform as a service (PaaS).

iii) Application

Application APIs provide methods to interface and extend applications on the web. Application APIs connect to applications such as CRM, ERP, and social media and help desk. These applications are delivered as software as a service (SaaS).

B. Cloud provider cloud APIs

Cloud provider cloud APIs provide abstractions over a specific provider cloud and usually have custom or unique provider calls that are designed to enhance the amount of control of that cloud by using the provider's API implementation. Cloud provider APIs are implemented to support HTTP and HTTPS based communications protocols. Cloud provider cloud APIs have authentication mechanisms put in place to ensure that only authorized API calls are made to their systems. Most cloud provider based APIs have an ID or Authentication Key which provides an authorization/authentication and is usually passed over HTTPS to ensure security. Cloud provider APIs also may use the ID or another Key to create a hash-based token or a password to authenticate provide additional security (similar to public key infrastructure).

i) Cross-platform cloud APIs

Cross-platform cloud APIs provides a higher level of abstraction than cloud provider based cloud APIs. This is accomplished by taking cloud provider specific cloud API calls and making them generic. The benefits of using a cross-platform based cloud API is the ability to use a single API call, to access or leverage cloud resources on more than one provider's cloud computing platform. This saves a considerable amount of time, reduces complexity of the code rather than implementing multiple cloud provider based cloud APIs.

A. Character Descriptor:

We propose a novel character descriptor to model character structure for effective character recognition. Fig.2 depicts the flowchart of our proposed character descriptor. It employs four types of keypoint detectors, Harris detector (HD) to extract keypoints from corners and junctions, MSER detector (MD) to extract keypoints from stroke components, Dense detector (DD) to uniformly extract keypoints, and Random detector (RD) to extract the preset number of keypoints in a random pattern. As shown in Fig. 6, four feature detectors are able to cover almost all representative keypoints related to character structure. At each of the extracted keypoints, the HOG feature is calculated as an observed feature vector *x* in feature space.

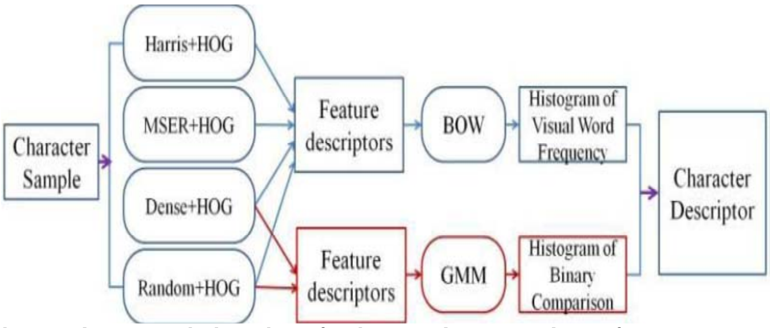

Fig.2 Flowchart of our proposed character descriptor, which combines four keypoint detectors, and HOG features are extracted at keypoints. Then BOW and GMM are employed to respectively obtain visual word histogram and binary comparison histogram.

HOG is selected as local features descriptor because of its compatibility with all above keypoint detectors. Some other feature descriptors like SIFT and SURF are not employed in our method, because our experimental results show that their performance on character recognition is lower than HOG. It might be because SIFT (or SURF) descriptor relies on the keypoints obtained from SIFT (or SURF) detector of its own. Each character patch is normalized into size 128 × 128, containing a complete character. In the process of feature quantization, the Bag-of-Words (BOW) Model and Gaussian Mixture Model (GMM) are employed to aggregate the extracted features. BOW is applied to keypoints from all the four detectors. GMM is applied to those only from DD and RD, because GMM-based feature representation requires fixed number and locations of the keypoint all character patch samples, while the numbers and locations of keypoints from HD and MD depend on character structure in the character patches. In both models, character patch is mapped into characteristic histogram as feature representation. By the cascade of BOW-based and GMM-based feature representations, we derive the character descriptor with significant discriminative power for recognition.

i) Bag-of-Words Model (BOW): The BOW model represents a character patch from the training set as a frequency histogram of visual words. The BOW representation is computationally efficient and resistant to intra-class variations. At first, *k*-means clustering is performed on HOG features extracted from training patches to build a vocabulary of visual words. Then feature coding and pooling are performed to map all HOG features from a character patch into a histogram of visual words. We adopt soft-assignment coding and average pooling schemes in the experiments. More other coding/pooling schemes will be tested in our future work. For each of the four feature detectors HD, MD, DD, and RD, we build a vocabulary of 256 visual words. This number of visual words is experimentally chosen to balance the performance of character recognition and the computation cost. At a character patch, the four detectors are applied to extract their respective keypoints, and then their corresponding HOG features are mapped into the respective vocabularies, obtaining four frequency histograms of visual words.

ii) Gaussian Mixture Model (GMM): In DD and RD, keypoints are extracted from each character patch according to predefined parameters rather than character structure. In our experiments, DD generates a uniform 8 × 8 keypoint array and RD generates 64 keypoints randomly, but all character patches share the same random pattern. Therefore, the keypoints extracted by RD and DD are always located at the same positions in all character patches, as shown in Fig. 3. To describe the local feature distributions, we build a GMM over all character patches in training set. In our experiments, each GMM contains 8 Gaussian distributions. This parameter is selected from the best results of scene character recognition.

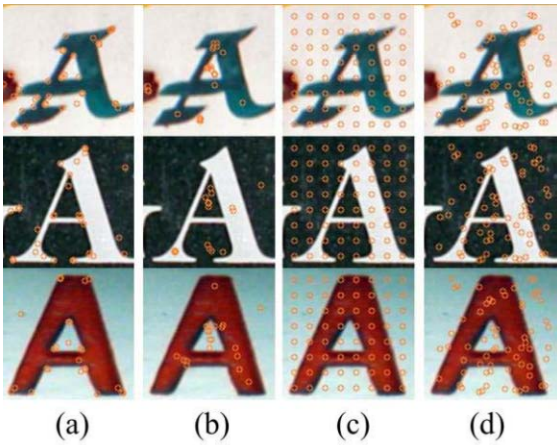

Fig.3 Keypoints extracted respectively by the four detectors from three character patches. (a) Keypoints detected by HD. (b) Keypoints detected by MD. (c) Keypoints detected by DD. (d) Keypoints detected by RD.

VI. CONCLUSION

We have presented a method of scene text recognition from detected text regions, which is compatible with mobile applications. It detects text regions from natural scene image/video, and recognizes text information from the detected text regions. Our proposed character descriptor is effective to extract representative and discriminative text features for both recognition schemes. In this paper, we used cloud based API to improve the performance of speed and accuracy of recognition.

References

1. F. Chassaing , C. Wolf, J. M. Jolion, and, "Text localization, enhancement and binarization in multimedia documents," in Pattern Recognition, Aug. 2002, vol. 2 of Proceedings. 16th International Conference on, pp. 1037–1040.
2. N. Dalal and B. Triggs, "Histograms of oriented gradients for humanmdetection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,m Jun. 2005, pp. 886 – 893.
3. P. Viola and M. J. Jones, "Robust real-time face detection," Int. J. Comput. Vis., vol. 57, no. 2, pp. 137–154, 2004.
4. Giri, P. S.[2013], "Text Information Extraction And analysis from Image Using Digital Image Processing Techniques
5. M. Swamy Das, B. HimaBindhu, A. Govardhan [2012], "Evaluation of Text Detection and Localization Methods in Natural Images."
6. Angadi, S.A. and Kodabagi, M.M, Text region extraction from low resolution natural scene images using texture features, 2ndInternational Advance Computing Conference, IEEE,2010.
7. Xiaoqing Liu and Jagath Samarabandu, An Edge-based text region extraction algorithm for Indoor mobile robot navigation, Proceedings of the IEEE, Jul 2005.
8. N. Ezaki, M. Bulacu and L. Schomaker, "Text detection from natural scene images:Towards a system for visually impaired persons", In Proceedings of the International Conference on Pattern Recognition (ICPR'04), 2004, pp. 683-686.
9. T .Yamaguchi, Y. Nakano, M. Maruyama, H. Miyao, and T. Hananoi, "Digit classification on signboards for telephone number recognition", In Proc. of 7th Int. Conf. on Document Analysis and Recognition (ICDAR 2003), volume I, Edinburgh, Scotland, 3-6 August 2003, pp. 359–363
10. K. Matsuo, K. Ueda, and U. Michio, "Extraction of character string from scene image by binarizing local target area", Transaction of The Institute of Electrical Engineers of Japan, 122-C(2), February 2002, pp. 232–241.
11. T. Yamaguchi and M. Maruyama, "Character Extraction from Natural Scene Images by Hierarchical Classifiers", In Proceedings of the International Conference on Pattern Recognition (ICPR'04), 2004, pp.687-690.
12. J. Yang, J. Gao, Y. Zang, X. Chen, and A. Waibel, "An automatic sign recognition and translation system", In Proceedings of the Workshop on Perceptive User Interfaces (PUI'01), November 2001, pp, 1-8.
13. R. Beaufort and C. Mancas-Thillou, "A weighted finite-state framework for correcting errors in natural scene OCR," in Proc. 9th Int. Conf. Document Anal. Recognit., Sep. 2007, pp. 889– 893.

14. R. Smith, “An overview of the tesseract OCR engine,” in Proc. Int. Conf. Document Anal. Recognit., Sep. 2007, pp. 629–633.
15. J. J. Weinman, E. Learned-Miller, and A. R. Hanson, “Scene text recognition using similarity and a lexicon with sparse belief propagation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 10, pp. 1733–1746, Oct. 2009.
16. L.Neumann and J. Matas, “Real-time scene text localization and recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 3538–3545.
17. D. L. Smith, J. Feild, and E. Learned-Miller, “Enforcing similarity constraints with integer programming for better scene text recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2011, pp. 73–80.
18. A. Mishra, K. Alahari, and C. V. Jawahar, “Top-down and bottom-up cues for scene text recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 1063–6919.
19. S. Lu, L. Li, and C. L. Tan, “Document image retrieval through word shape coding,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 11, pp. 1913–1918, Nov. 2008.
20. A. Coates et al., “Text detection and character recognition in scene images with unsupervised feature learning,” in Proc. ICDAR, Sep. 2011, pp.440–445.
21. C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, “Scene text recognition using part-based tree-structured character detection,” in Proc. CVPR, Jun. 2013, pp. 2961–2968.
22. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE TraPattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627–1645,Sep. 2010
23. Q. Zheng, K. Chen, Y. Zhou, G. Cong, and H. Guan, “Text localization and recognition in complex scenes using local features,” in Proc. 10th ACCV, 2010 , pp. 121–132 .
24. K. Wang and S. Belongie, “Word spotting in the wild,” in Proc. Eur. Conf. Comput. Vis., 2010.
25. K. Wang, B. Bbenko, and S. Belongie, “End-to-end scene text recognition,” in Proc. Int. Conf. Comput. Vis., Nov. 2011, pp. 1457–1464.
26. P. Shivakumara, T. Q. Phan, C. L. Tan, “Video text detection based on filters and edge features”, Int. Conf. on Multimedia & Expo (ICME 2009),pp.514-517.
27. Ms. Saumya sucharita Sahoo,” Review of Methods of Scene Text Detection and its Challenges”, International Journal of Electronics and Communication Engineering & Technology (IJECET), pp. 74-81, ISSN 0976 – 6464(Print), ISSN 0976 – 6472(Online), Volume 5, Issue 1, January (2014), © IAEME.
28. Prajakta Bastawade,” Detection of Text String from Natural Scene Images By Image Partition and Grouping”, Journal of Global Research in Computer Science, pp.1-7, Volume 4, No. 11, November 2013.