Scene Text Recognition in Mobile Applications by Character Descriptor and Structure Configuration

S. Dhiveha[1]
St.Joseph's College of Engineering and Technology
Thanjavur, India

Mrs. M. Sujatha[2]
St.Joseph's College of Engineering and Technology
Thanjavur, India

***Abstract*—Text characters and strings in natural scene can provide valuable information for many applications. Extracting text directly from natural scene images or videos is a challenging task because of diverse text patterns and variant background interferences. This paper proposes a method of scene text recognition from detected text regions. 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. An Android-based demo system is developed to show the effectiveness of our proposed method on scene text information extraction from nearby objects. The demo system also provides us some insight into algorithm design and performance improvement of scene text extractionThe evaluation results on benchmark data sets demonstrate that our proposed scheme of text recognition is comparable with the best existing methods..**

***Keywords*— Scene text detection, scene text recognition, mobile application, character descriptor, stroke configuration, text understanding, text retrieval, mobile application.**

I INTRODUCTION

camera-based text information serves as effective tags or clues for many mobile applications associated with media analysis, content retrieval, scene understanding, and assistant navigation. In natural scene images and videos, text characters and strings usually appear in nearby sign boards and hand-held objects and provide significant knowledge of surrounding environment and objects. 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. For example, a frontal face normally contains a mouth, a nose, two eyes, and two brows as prior knowledge. However, it is difficult to model the structure of text characters in scene images due to the lack of discriminative pixel-level appearance and structure features from non-text background outliers. Further, text consists of different words where each word may contain different characters in various fonts, styles, and sizes, resulting in large intra-variations of text patterns. To solve these challenging problems, scene text extraction is divided into two processes: text detection and text recognition. Text detection is to localize image regions containing text characters and strings. Text recognition is to transform pixel-based text into readable code. It aims to accurately distinguish different text characters and properly compose text words. Pixel-based layout analysis is adopted to extract text regions and segment text characters in images, based on color uniformity and horizontal alignment of text characters. In text recognition process, we design two schemes of scene text recognition. The first one is training a character recognizer to predict the category of a character in an image patch. The second one is training a binary character classifier for each character class to predict the existence of this category in an image patch. The two schemes are compatible with two promising applications related to scene text, which are text understanding and text retrieval. Text understanding is to acquire text information from natural scene to understand surrounding environment and objects. Text retrieval is to verify whether a piece of text information exists in natural scene. These two applications can be widely used in smart mobile device. The main contributions of this paper are associated with the proposed two recognition schemes. Firstly, a character descriptor is proposed to extract representative and discriminative features from character patches. It combines several feature detectors (Harris-Corner, Maximal Stable Extremal Regions (MSER), and dense sampling) and Histogram of Oriented Gradients (HOG) descriptors .Secondly, to generate a binary classifier for each character class in text retrieval, we propose a novel stroke configuration from character boundary and skeleton to model character structure.

II RELATED WORK

In this section, we present a general review of previous work on scene text recognition respectively. While text detection aims to localize text regions in images by filtering out non text outliers from cluttered background text recognition is to transform image-based text information in the detected regions into readable text codes. Scene text recognition is still an open topic to be addressed. In the

Robust Reading Competition of International Conference on Document Analysis and Recognition (ICDAR) .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.

.

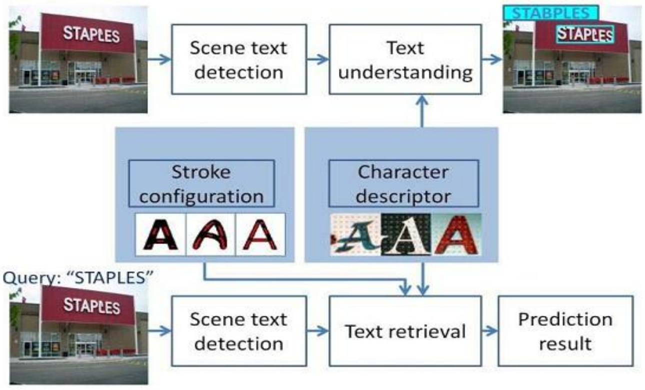

Fig. 1. The flow chart of our designed scene text extraction method

Current optical character recognition (OCR) systems 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. 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. Gabor-based appearance model, a language model related to simultaneity frequency and letter case, similarity model, and lexicon model to perform scene character recognition. proposed a real time scene text localization and recognition method based on extremal regions Smith built a similarity model of scene text characters based on SIFT, and maximized posterior probability of similarity constraints by integer programming. Mishra adopted conditional random field to combine bottom-up character recognition and top-down word-level recognition. modeled the inner character structure by defining a dictionary of basic shape codes to perform character and word retrieval without OCR on scanned documents.

In a part-based tree structure model was designed to detect text characters under Latent-SVM ,and recognize text words from text regions under conditional random field. In 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 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 Random Ferns algorithm was used to perform character detection and constructed a system for query-based word detection in scene images.

III . LAYOUT-BASED SCENE TEXT DETECTION

In natural scene, most text information is set for instruction or identifier. Text strings in print font are located at signage boards or object packages. They are normally composed of characters in uniform color and aligned arrangement, while non-text background outliers are in the form of disorganized layouts. The color uniformity and horizontal alignment were employed to localize text regions in scene images. In our current work, scene text detection process is improved to be compatible with mobile applications. According to our observations, the text on sign boards or print labels on nearby objects in general appear in uniform color. Thus we can group the pixels with similar color into the same layers, and separate text from background outliers in different colors.

To decompose a scene image into several color-based layers, we have designed a boundary clustering algorithm based on bigram color uniformity in our previous work. Text information is generally attached to a plane carrier as attachment surface with uniform colors respectively. We then model color difference by a vector of color pair, which is obtained by cascading the RGB colors of text and attachment surfaces. Each boundary can be described by a color-pair, and we cluster the boundaries with similar color pairs into the sample layer.

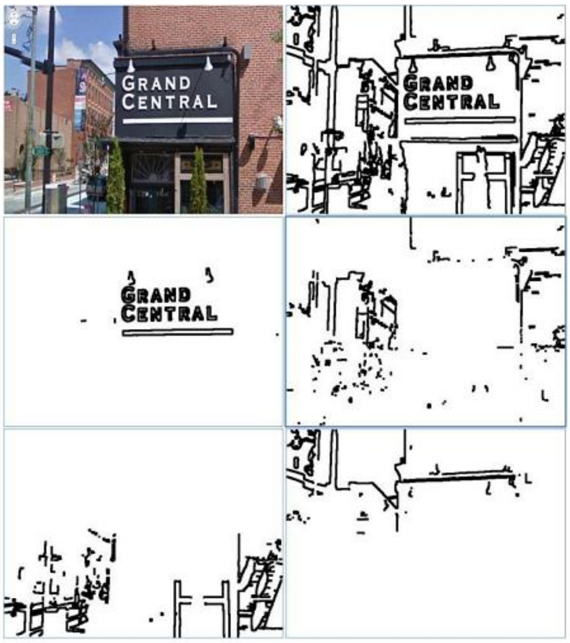

Fig. 2. Color decomposition of scene image by boundary clustering algorithm. The top row presents original scene image and the edge image obtained from canny edge detector. The other rows present color layers obtained from bigram color uniformity. It shows that the text information in signage board is extracted from complex background in a color layer.

The boundaries of text characters are separated from those of background outliers. In each color layer, we analyze geometrical properties of the boundaries to detect the existence of text characters. According to our observation, text information generally appears in text strings composed of several character members in similar sizes rather than single character, and text strings are normally in approximately horizontal alignment. Thus we design an adjacent character grouping algorithm to search for image regions containing text strings. To model the boundary size and location of a text string, a bounding box is assigned to each boundary in a color layer. For each bounding box, we search for its siblings in similar size and vertical locations (horizontal alignment).

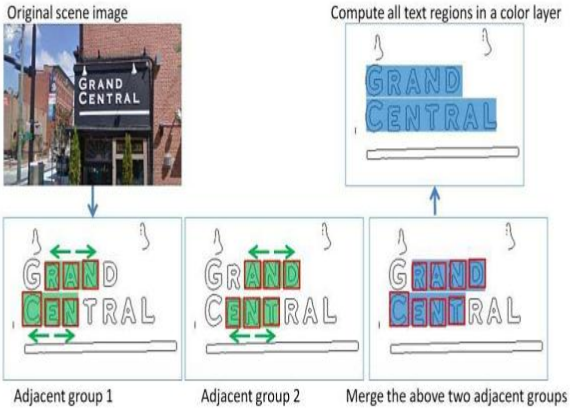
Original scene image
Compute all text regions in a color layer
GRAND
CENTRAL
Adjacent group 1
Adjacent group 2
Merge the above two adjacent groups

Fig. 3. The adjacent character grouping process. The red box denotes bounding box of a boundary in a color layer. The green regions in the bottom left two figures represent two adjacent groups of consecutive neighboring bounding boxes in similar size and horizontal alignment. The blue regions in the bottom-right figure represent the text string fragments, obtained by merging the overlapping adjacent groups.

If several sibling bounding boxes are obtained on its left and right, then we merge all these involved bounding boxes into a region. This region contains a fragment of text string. Then we repeat above method to calculate all text string fragments in this color layer, and merge the string fragments with intersections. depicts the process of adjacent character grouping.

In order to extract text strings in slightly non-horizontal orientations .we search for possible characters of a text string within a reasonable range of horizontal orientation. When estimating horizontal alignment, we do not require all the characters exactly align in horizontal orientation, but allow some differences between neighboring characters that are assigned into the same string. In our system we set this range as ±p/6 degrees relative to the horizontal line. This range could be set to be larger but it would bring in more false positive strings from background.

A. Layout Analysis of Color Decomposition

According to our observations, the text on sign boards or print labels on nearby objects in general appear in uniform color. Thus we can group the pixels with similar color into the same layers, and separate text from background outliers in different colors.

To decompose a scene image into several color-based layers, Text information is generally attached to a plane carrier as attachment surface with uniform colors respectively. We define the uniformity of their color difference as bigram color uniformity. Color difference is related to the character boundary, which serves as a border between text strokes and the attachment surfaces. We then model color difference by a vector of color pair, which is obtained by cascading the RGB colors of text and attachment surfaces. Each boundary can be described by a color-pair, and we cluster the boundaries with similar color pairs into the sample layer.

B. Layout Analysis of Horizontal Alignment

In each color layer, we analyze geometrical properties of the boundaries to detect the existence of text characters. According to our observation, text information generally appears in text strings composed of several character members in similar sizes rather than single character, and text strings are normally in approximately horizontal alignment. Thus we design an adjacent character grouping algorithm to search for image regions containing text strings.

To model the boundary size and location of a text string, a bounding box is assigned to each boundary in a color layer. For each bounding box, we search for its siblings in similar size and vertical locations (horizontal alignment). If several sibling bounding boxes are obtained on its left and right, then we merge all these involved bounding boxes into a region. This region contains a fragment of text string. Then we repeat above method to calculate all text string fragments in this color layer, and merge the string fragments with intersections. Fig. 3 depicts the process of adjacent character grouping.

C. Extract Text Strings

In order to extract text strings in slightly non-horizontal orientations, we search for possible characters of a text string within a reasonable range of horizontal orientation. When estimating horizontal alignment, we do not require all the characters exactly align in horizontal orientation, but allow some differences between neighboring characters that are assigned into the same string. In our system we set this range as ±p/6 degrees relative to the horizontal line. This range could be set to be larger but it would bring in more false positive strings from background. In addition, our scene text detection algorithm can handle challenging font variations

To be compatible with blind-assistant demo system, some technical details of our scene text detection algorithm are adjusted. At first the input image is down-sampled to improve the efficiency. Then in color decomposition, only the edge pixels from a boundary that satisfies specific geometrical constraints are adopted to build color layers. Also some parameters related to horizontal similarity and alignment are adjusted according to our evaluations in real environments.

IV. Structure-Based Scene Text Recognition

From the detected text regions, character recognition is performed to extract text information. In current work, scene text characters include 10 digits [0-9] and 26 English letters in upper case [A-Z] and lower case [a-z], 62 character classes in total. Three public datasets are employed for training character recognizer and evaluating its performance, and these datasets contain image patches of complete and regular text characters cropped from scene images. We will provide detailed descriptions in Section VI.

As mentioned in Section I, we design two character recognition schemes. In text understanding, character recognition is a multi-class classification problem. We train a character recognizer to classify the 62 classes of characters. In text retrieval, character recognition is a binary classification problem. For each of the 62 character classes, For example, we train a binary classifier for character class 'A', then this classifier will predict a patch containing 'A' as positive, and predict a patch containing other character classes or non-text outliers as negative. The specified character classes are defined as queried characters.

In both schemes, a robust character descriptor is required to extract structure features from character patches. In text retrieval, to better model character structure, we define stroke configuration for each character class based on specific partitions of character boundary and skeleton.

A. Character Descriptor

We propose a novel character descriptor to model character structure for effective character recognition. Fig. 5 depicts the flow chart of our proposed character descriptor.

It employs four types of key point detectors, Harris detector (HD) to extract key points from corners and junctions, MSER detector (MD) to extract key points from stroke components, Dense detector (DD) to uniformly extract key points, and Random detector (RD) to extract the preset number of key points in a random pattern.

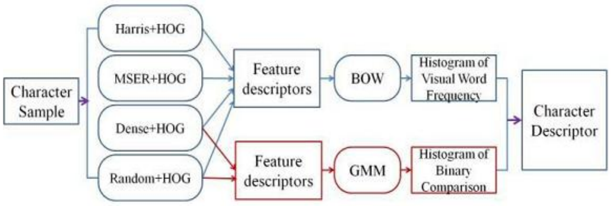

Fig. 4. Flowchart of our proposed character descriptor, which combines four key point detectors, and HOG features are extracted at key points. Then BOW and GMM are employed to respectively obtain visual word histogram and binary comparison histogram.

At each of the extracted key points, the HOG feature is calculated as an observed feature vector x in feature space. 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 key points 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 key point all character patch samples, while the numbers and locations of key points 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.

1) 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 key points, and then their corresponding HOG features are mapped into the respective vocabularies, obtaining four frequency histograms of visual words. Each histogram has 256 dimensions. Then we cascade the four histograms into BOW-based feature representation in 256 × 4 = 1024 dimensions.

2) Gaussian Mixture Model (GMM): In DD and RD, key points are extracted from each character patch according to pre defined parameters rather than character structure. In our experiments, DD generates a uniform 8 × 8 key point array and RD generates 64 key points randomly, but all character patches share the same random pattern. Therefore, the key points extracted by RD and DD are always located at the same positions in all character patches, as shown in Fig. 6. 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.

In the process of building GMM, K-means clustering ($K = 8$) is first applied to calculate K centers of the HOG descriptors, where the s-th ($1 = s = K$) center is used as initial means of the s-th Gaussian in GMM. Then the initial weights and co-variances are calculated from the means. Next, an EM algorithm is used to obtain maximum likelihood estimate of the three parameters, weights, means, and co-variances of all the Gaussian mixture distributions

B. Character Stroke Configuration

In text retrieval application, the query character class is considered as an object with fixed structure, and we generate its binary classifier according to structure modeling. Character structure consists of multiple oriented strokes, which serve as basic elements of a text character. From the pixel-level perspective, a stroke of printed text is defined as a region bounded by two parallel boundary segments. In order to locate stroke accurately, stroke is redefined in our algorithm as skeleton points within character sections with consistent width and orientation.

A character can be represented as a set of connected strokes with specific configuration which includes the number, locations, lengths and orientations of the strokes. Here, the structure map of strokes is defined as stroke configuration. In a character class, although the character instances appear in different fonts, styles, and sizes, the stroke configurations is always consistent. For example, character 'B' is always a vertical stroke with two arc strokes in any pattern. Therefore for each of the 62 character classes, we can estimate a stroke configuration from training patches to describe its basic structure.

We have developed demo systems of scene text extraction in Android-Mobile platforms. We integrate the functional modules of scene text detection and text recognition. It is able to detect regions of text strings from cluttered background, and recognize characters in the text regions.

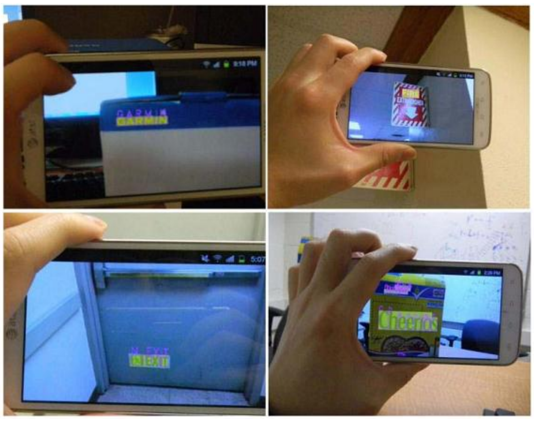

Fig. 5. Our demo system of scene text extraction in Android platform. The text strings "GARMIN", "FIRE", "EXIT", and "Cheerios" are extracted from complex background. Although some characters are incorrectly recognized in current demo, we will introduce lexicon analysis to improve the recognition performance.

In blind-assistant application, the demo system has been used to assist blind or visually impaired people to recognize hand-held object. Another demo system consists of camera, processing device, and Bluetooth earplug. In this system, our proposed method is implemented and deployed into the processing device. Camera is used as input device for capturing natural scene, and Bluetooth earplug is used as output device for broadcasting the recognized text information.

V. Q UANTITATIVE EXPERIMENTAL ANALYSIS

Scene text extraction consists of detection and recognition. However, the main technical contributions of this paper are the two scene character recognition schemes compatible with mobile applications. We perform experiments to evaluate the two schemes over benchmark datasets.

A. Datasets

To evaluate the proposed character descriptor and the character stroke configuration, we employ three public datasets of scene text characters, in which we conduct scene character recognition. The first one is Chars74K EnglishImg Dataset. It contains all the 62 character classes with the approximately balanced number of samples. The samples in this dataset are divided into two categories, GoodImg and BadImg, according to the recognition difficulty. The second one is Sign Dataset it captures 96 camera-based signs with 1209 scene characters. The third one is ICDAR-2003 Robust Reading Dataset. It contains about 11600 character samples which are cropped from text regions of natural scene images. In Sign Dataset and ICDAR-2003 Dataset, the number of character samples from different categories is unbalanced.

B. Scene Character Recognition for Text Understanding

In performance evaluations of text understanding, we use accuracy rate (AR) as evaluation measure, which is defined as the ratio between the number of correctly recognized text characters and the total number of text characters.

In addition, we further evaluate the two feature representations of our character descriptor independently. BOW-based feature representation obtains 0.53 and GMM-based feature representation 0.47. This may be due to the fact that BOW-based representations cover key points from all four detectors, while GMM-based representations rely only on DD and RD key points. In DD and RD, it is unavoidable that some key points are not located on characters.

C. Scene Character Recognition for Text Retrieval

The proposed character structure modeling is applied to extract structure features from stroke configuration of the characters to learn a binary classifier for each character class. We evaluate these binary classifiers by queried character classification in the above three datasets.

In each character class, two measurements, accuracy rate (AR) and false positive rate (FPR), are calculated to evaluate the performance of queried character classification.

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. In scene text detection, layout analysis of color decomposition and horizontal alignment is performed to search for image regions of text strings. In scene text recognition, two schemes, text understanding and text retrieval, are respectively proposed to extract text information from surrounding environment. Our proposed character descriptor is effective to extract representative and discriminative text features for both recognition schemes. To model text character structure for text retrieval scheme, we have designed a novel feature representation, stroke configuration map, based on boundary and skeleton. Quantitative experimental results demonstrate that our proposed method of scene text recognition outperforms most existing methods. We have also implemented the proposed method to a demo system of scene text extraction on mobile device. The demo system demonstrates the effectiveness of our proposed method in blind-assistant applications, and it also proves that the assumptions of color uniformity and aligned arrangement are suitable for the captured text information from natural scene.

REFERENCES

[1] X. Bai, L. J. Latecki, and W.-Y. Liu, “Skeleton pruning by contour partitioning with discrete curve evolution,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 449–462, Mar. 2007.

[2] 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.

[3] X. Chen, J. Yang, J. Zhang, and A. Waibel, “Automatic detection and recognition of signs from natural scenes,” *IEEE Trans. Image Process.*, vol. 13, no. 1, pp. 87–99, Jan. 2004.

[4] A. Coates *et al.*, “Text detection and character recognition in scene images with unsupervised feature learning,” in *Proc. ICDAR*, Sep. 2011, pp. 440–445.

[5] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[6] T. de Campos, B. Babu, and M. Varma, “Character recognition in natural images,” in *Proc. VISAPP*, 2009.

[7] B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in *Proc. CVPR*, Jun. 2010, pp. 2963–2970.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[9] T. Jiang, F. Jurie, and C. Schmid, “Learning shape prior models for object matching,” in *Proc. CVPR*, Jun. 2009, pp. 848–855.

[10] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, and S. D. Johsi, “Text extraction and document image segmentation using matched wavelets and MRF model,” *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2117–2128, Aug. 2007.

[11] L. J. Latecki and R. Lakamper, “Convexity rule for shape decomposition based on discrete contour evolution,” *Comput. Vis. Image Understand.*, vol. 73, no. 3, pp. 441–454, 1999.

[12] Y. Liu, J. Yang, and M. Liu, “Recognition of QR code with mobile phones,” in *Proc. CCDC*, Jul. 2008, pp. 203–206.

[13] 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.

[14] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, “ICDAR 2003 robust reading competitions,” in *Proc. Int. Conf. Document Anal. Recognit.*, Aug. 2003, pp. 682–687.

[15] 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.

[16] N. Nikolaou and N. Papamarkos, “Color reduction for complex document images,” *Int. J. Imag. Syst. Technol.*, vol. 19, no. 1, pp. 14–26, 2009.

Dhiveha.S received her B.E degree in computer science and engineering in P.R engineering college, Thanjavur in 2013, Tamilnadu, India. Now she is doing her master in engineering in St.Joseph’s college of engineering and technology, Thanjavur, Tamilnadu, India.