DRT: A Unified Framework for Text Detection, Recognition and Tracking in Video

Shamnasana V[1], Binoy D L[2]

[1]MEA Engineering College, State Highway 39, Nellikkunnu – Vengoor, Perinthalmanna, Malappuram, Kerala, India

[2]MEA Engineering College, State Highway 39, Nellikkunnu – Vengoor, Perinthalmanna, Malappuram, Kerala, India

Abstract: ***The successful analysis of video data is currently in great demand because a video is a major source of data in our lives. The text is a direct source information, while recent surveys on the detection of and recognition in imagery mainly focuses on extracting text scene pictures. Here, this work presents an implementation of text detection, video tracking and recognition with three Contributions. First, a unified generic framework is proposed for video text extraction that consistently implementing detection, recognition and tracking. Secondly, this framework is implemented recognize caption text and scene text..This work proposes a new generic framework that successively implements text detection, recognition and tracking in videos, the text can be of any type, both caption text and scene text.***

Keywords: OCR, SWT, MSER, Caption text

1. Introduction

The explosive growth of smart phones and online social media have led to the accumulation of large amounts of visual data, in particular, the massive and increasing collections of video on the Internet and social networks. For example, YouTube1 streamed approximately 100 hours of video per minute worldwide in 2014. These countless videos have triggered research activities in multimedia understanding and video retrieval.

In the literature, text has received increasing attention as a key and direct information sources in video. As examples, caption text usually annotates information concerning where and when and the events in video happened or who was involved, and signage text is widely used as a visual indicators for navigation and notification in scenes. Hence, text extraction and analysis in video has attracted considerable attention in multimedia understanding systems. Specifically, some researchers performed investigations of video retrieval by leveraging both textual video representations (extracted from text in frames and audio) and visual representations using high-level object and action concepts and found that the ability to understand video text can significantly improve the retrieval performance.

A wide variety of methods have been proposed to extract text from images and videos, and several studies have contributed reviews. Despite this, numerous advanced techniques for video text extraction have proliferated impressively over the past decade. Although video text extraction techniques have been addressed in previous surveys, they were treated in either an image based framework or separated in tracking or enhancement sections. Many video text extraction methods detect and recognize text in each sampled individual frame (i.e., frame by frame) without multiframe integration. It involves

- Text Detection
- Text Recognition
- Text Tracking

The concept of text detection and recognition is shown in figure below.

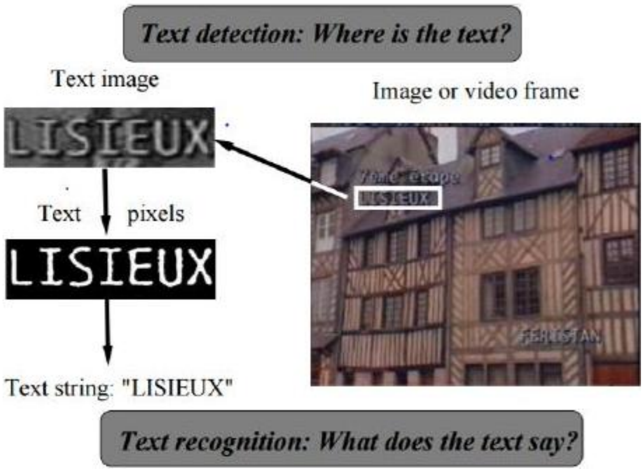

Figure 1: Concept of text detection and recognition

Text detection is the task of localizing the text in each video frame. Text recognition means what the localized text reads. The goal of text tracking is to continuously determine the location of text across multiple dynamic video frames. There are a variety of recent challenges for text extraction in scene by robots and users, like heterogeneous background, varied text, non-uniform illumination, arbitrary motion and low contrast. Most previous video text detection methods are reviewed with local information within individual frames, with limited performance. There are a limited number of approaches for scene text detection in video; most of them focus on extracting text with local information. At the same time, there are a few techniques with spatial and temporal information utilization for detecting text within multiple frames. Some researchers have presented specific frameworks for video text extraction. For example, Antani et al. divided video text extraction into four tasks: detection, localization, segmentation, and recognition. In their system, the tracking stage provides additional input to the spatial temporal decision fusion for improving localization. Jung et al. summarized the sub problems of a text information extraction system for both images and video into text detection, localization, tracking, extraction and enhancement, and recognition . This paper proposes a unified video text extraction framework, that detect, recognize and track any type of English texts. Text extraction and analysis in video has attracted considerable attention in multimedia understanding systems. Specifically, some researchers performed investigations of video retrieval by leveraging both textual video representations (extracted from text in frames and audio) and visual representations using high level object and action concepts and found that the ability to understand video text can significantly improve the retrieval performance.

Following the method of categorization text in video is categorized as either caption or scene text. Caption text is also called graphic text or artificial text. Caption text provides good directivity and a high-level overview of the semantic information in captions, subtitles and annotations of the video, while scene text is part of the camera images and is naturally embedded within objects (e.g., trademarks, signboards and buildings) in scenes. Moreover, we classify caption text into two subcategories: layered caption text and embedded caption text. Layered caption text is always printed on a specifically designed background layer while embedded caption text is overlaid and embedded on the frame.

2. Methodology

This section presents the main ideas and details of the proposed system. The proposed framework is designed to implement text detection, recognition and tracking in videos and natural scene images in a single pipeline. In this system, the proposed unified framework takes text detection, tracking and recognition as a whole and performs all tasks in a single unified pipeline.

A schematic overview of the proposed framework is illustrated in Figure 2.

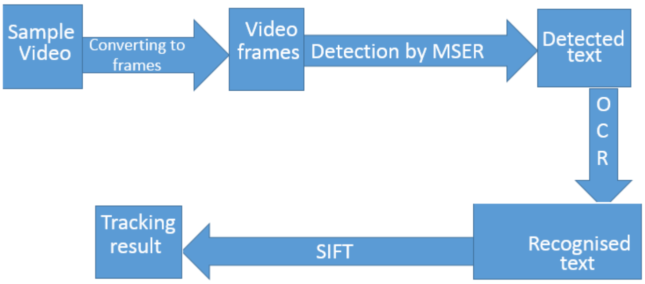

Figure 2: Overview of the proposed framework

Here the proposed framework deals with the following three tasks.

- Detection
- Recognition
- Tracking

Here each task follows different methods to complete the work .ie, Detection will be done by a specific method, recognition by another method and tracking by another method. And these three tasks are incorporated into a framework. Here the method used for text detection is Maximally Stable External Regions (MSERs) with Stroke Width Transform (SWT) method. And recognition is done by OCR(Optical Character Recognition) method. And the third task tracking will be done by tracking with SIFT (Scale Invarient Feature Transform)features.

a) Text Detection by Maximally Stable Extremal Regions (MSERS) with Stroke Width Transform (SWT) Method

Detection is the task of localizing the text. This method automatically detect and recognize text in natural images. This is a common task performed on unstructured scenes. Unstructured scenes are images that contain undetermined or random scenarios. For example, you can detect and recognize text automatically from captured video to alert a driver about a road sign. This is different than structured scenes, which contain known scenarios where the position of text is known beforehand. This method is explained here in five steps with figure.

(1) Detect Candidate Text Regions Using MSER.

The MSER feature detector works well for finding text regions. It works well for text because the consistent color and high contrast of text leads to stable intensity profiles. Use the detect MSER Features function to find all the regions within the image and plot these results. Notice that there are many non-text regions detected alongside the text. Detected MSER regions are shown in figure 3.

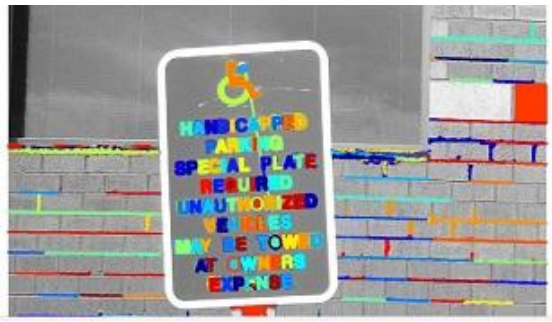

Figure 3: MSER regions

(2) Remove Non-Text Regions Based On Basic Geometric Properties

Although the MSER algorithm picks out most of the text, it also detects many other stable regions in the image that are not text. You can use a rule-based approach to remove non-text regions. For example, geometric properties of text can be used to filter out non-text regions using simple thresholds. Alternatively, you can use a machine learning approach to train a text vs. non-text classifier. Typically, a combination of the two approaches produces better results .This is shown in figure 4.There are several geometric properties that are good for discriminating between text and non-text regions including:

- Aspect ratio
- Eccentricity
- Euler number
- Extent
- Solidity

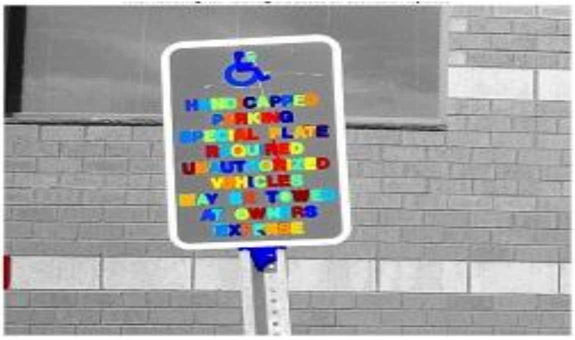

Figure 4: After removing non text regions based on geometric properties

(3) Remove Non-Text Regions Based On Stroke Width Variation

Another common metric used to discriminate between text and non-text is stroke width. Stroke width is a measure of the width of the curves and lines that make up a character. Text regions tend to have little stroke width variation, whereas non-text regions tend to have larger variations. To help understand how the stroke width can be used to remove non-text regions,estimate the stroke width of one of the detected MSER regions. You can do this by using a distance transform and binary thinning operation. Figure 5 shows the MSER regions after removing non-text regions.

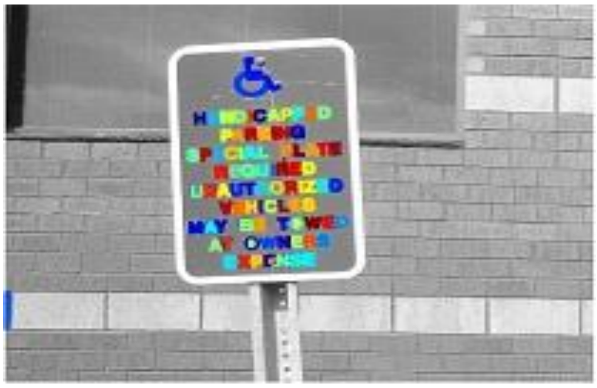

Figure 5: After removing non text regions based on stroke width variation.

(4) Merge Text Regions For Final Detection Result.

At this point, all the detection results are composed of individual text characters. To use these results for recognition tasks, such as OCR, the individual text characters must be merged into words or text lines. This enables recognition of the actual words in an image, which carry more meaningful information than just the individual characters. For example, recognizing the string ’EXIT’ vs. the set of individual characters ’X’,’E’,’T’,’I’, where the meaning of the word is lost without the correct ordering. One approach for merging individual text regions into words or text lines is to first find neighboring text regions and then form a bounding box around these regions. To find neighboring regions, expand the bounding boxes computed earlier with region props. This makes the bounding boxes of neighboring text regions overlap such that text regions that are part of the same word or text line form a chain of overlapping bounding boxes. The expanded bounding boxes text is shown in figure 6.

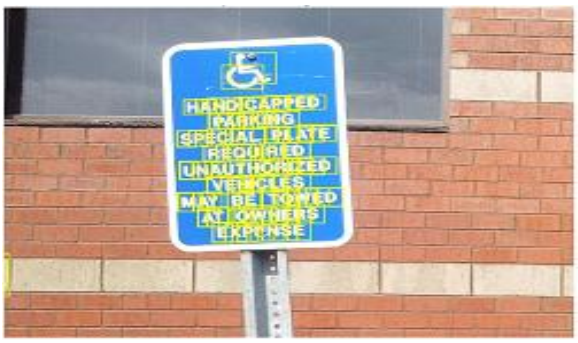

Figure 6: Expanded bounding boxes text.

(5) Recognize Detected Text Using OCR.

After detecting the text regions, use the ocr function to recognize the text within each bounding box. Note that without first finding the text regions, the output of the ocr function would be considerably more noisy.The detected text is shown in figure 7.

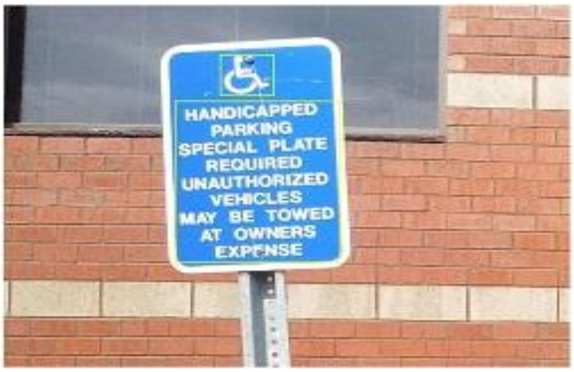

Figure 7: Detected text

b)Text Recognition By OCR Method

Video text recognition is conventionally performed using existing OCR (Optical Character Recognition) techniques; in other words, text regions are first segmented from video frames and then fed into a state-of-theart OCR engine . Optical character recognition, or OCR, is a method of converting a scanned image into text.

When a page is scanned, it is typically stored as a bit-mapped file in TIF format. When the image is displayed on the screen, we can read it. But to the computer, it is just a series of black and white dots.It is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example from a television broadcast). It is widely used as a form of information entry from printed paper data records, whether passport documents, invoices, bank statements, computerised receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitising printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision. Early versions needed to be trained with images of each character, and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs. Some systems are capable of reproducing formatted output that closely approximates the original page including images, columns, and other non-textual components. Optical character recognition software takes several steps to convert an image file into an editable document. Each step in this process uses a specific algorithm to alter, enhance, and interpret the images found within a file. Each and every step involved in this process is critical to the overall success of OCR. Even the smallest error will cause major issues, resulting in a poorly translated final document.

The flowchart of OCR is shown below.

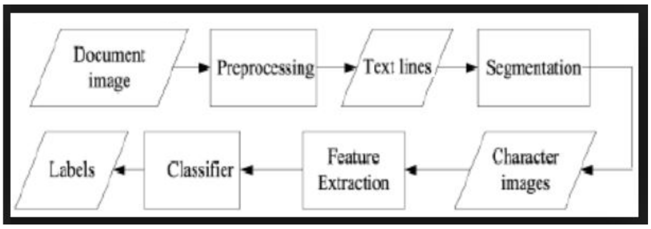

Figure 8: OCR Diagram

The OCR Process:

(1) Loading the image file.
In order for OCR to be effective, it must support a wide array of file formats, including PDF,BMP, TIFF, JPEG, and PNG files. Once the file is loaded, the software can begin to work. These files can be scanned documents, photographs, or even read-only files. Regardless of the original format, OCR software will transform these files into easily accessible editable data.

(2) Improving image quality and orientation.
In this stage of OCR, the software will work to de-skew, remove any noise, and improve the overall quality of the images. This is a critical step as blurry or skewed images are not interpreted properly.

(3) Removing lines
Lines can prove to be disastrous when interpreting characters. In order to remain as accurate as possible lines are detected and removed. This allows for better recognition quality when converting tables, underlinedwords, etc. Much like the importance of image quality, the removal of lines will ensure that characters are recognized accurately.

(4) Analyzing the page
During this stage of Optical Character Recognition, the layout of the original file is noted and processed. This includes the detection of text positions, white space, and the prioritization of important text areas or sections.

(5) Detecting words and lines of text
This is the beginning stage of actual character recognition. The software begins to identify individual words and entire lines of data. This is a critical pre-process for properly recognizing characters as it sets the stage for the analysis and correction of broken or merged characters.

(6) Analyzing and fixing of broken or merged characters.
Depending on the quality of the original file, there are often errors in which characters are broken or blurred together. The OCR software must now break down and resolve these errors in order to properly interpret the appropriate characters.

(7) Recognizing characters
This is the primary function of Optical Character Recognition. Now that the original file has been processed, cleaned, and fixed the OCR technology can begin to read and translate characters. Each image of every character is converted into a character code. If the algorithm is unsure of a character, the software will produce multiple character codes and choose the proper character later on.

(8) Saving the file
After the file has been fully interpreted, it can be saved to your desired file format. While there is much more to OCR software, these 8 steps make up the primary processes involved in Optical Character Recognition.

c) Video Text Tracking Based On SIFT Feature and Geometric Constraint

Video text provides important clues for semantic-based video analysis, indexing and retrieval. And text tracking is performed to locate specific text information across video frames and enhance text segmentation and recognition over time. Here presents a multilingual video text tracking algorithm based on the extraction and tracking of Scale Invariant Feature Transform (SIFT) features description through video frames. SIFT features are extracted from video frames to correspond the region of interests across frames. Meanwhile, a global matching method using geometric constraint is proposed to decrease false matches, which effectively improves the accuracy and stability of text tracking results. Based on the correct matches, the motion of text is estimated in adjacent frames and a match score of text is calculated to determine Text Change Boundary (TCB). Experimental results on a large number of video frames show that the proposed text tracking algorithm is robust to different text forms, including multilingual captions, credits, scene texts with shift, rotation and scale change, under complex backgrounds and light changing. The flowchart of the method is shown in figure 9.

Scale invariant feature transform (SIFT) is an algorithm in computer vision to detect and describe local features in images. SIFT key points of objects are first extracted from a set of reference images and stored in a database. An object is recognized in a new image by individually comparing each feature from the new image to this database and finding candidate matching features based on Euclidean distance of their feature vectors. Here the boundary of each letter is detected by blobe detection. Blob detection methods are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other.

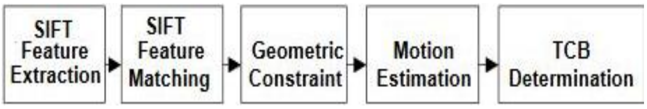

Figure 9: Flowchart of video text tracking method

3. Conclusion

The successful analysis of video data is currently in great demand because a video is a major source of data in our lives. The text is a direct source information, while recent surveys on the detection of and recognition in imagery mainly focuses on extracting text scene pictures. Here a unified framework is proposed and implemented to detect, recognize and track texts from videos. And the framework will be applicable to video texts like signboards, subtitles from movies and scene texts with different fonts. Firstly, the framework is implemented to detect, recognize and track texts from English language. The accuracy of standard OCR to recognize English fonts with various texts can be improved by creating template for each font in English language. It is a challenging job because there are numerous fonts in English.

References

[1] Barbu, T. (2012). Template matching based video tracking system using a novel n-step search algorithm and hog features. In *International Conference on Neural Information Processing*,pages 328–336. Springer.

[2] Gómez, L. and Karatzas, D. (2014). Mser-based real-time text detection and tracking. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 3110–3115. IEEE.

[3] Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2016). Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*,116(1):1–20.

[4] Jain, A., Peng, X., Zhuang, X., Natarajan, P., and Cao, H. (2014). Text detection and recognition in natural scenes and consumer videos. In *Acoustics, Speech and Signal Processing(ICASSP), 2014 IEEE International Conference on*, pages 1245–1249. IEEE.

[5] Messelodi, S. and Modena, C. M. (2013). Scene text recognition and tracking to identify athletes in sport videos. *Multimedia tools and applications*, 63(2):521–545.

[6] Nguyen, P. X., Wang, K., and Belongie, S. (2014). Video text detection and recognition:Dataset and benchmark. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 776–783. IEEE.

[7] Rong, X., Yi, C., Yang, X., and Tian, Y. (2014). Scene text recognition in multiple frames based on text tracking. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pages 1–6. IEEE.

[8] Roy, S., Shivakumara, P., Pal, U., Lu, T., and Tan, C. L. (2016). New tampered features for scene and caption text classification in video frame. In *Frontiers in Handwriting Recognition(ICFHR), 2016 15th International Conference on*, pages 36–41. IEEE.

[9] Shetty, S., Devadiga, A. S., Chakkaravarthy, S. S., and Kumar, K. V. (2014). Ote-ocr based text recognition and extraction from video frames. In *Intelligent Systems and Control(ISCO), 2014 IEEE 8th International Conference on*, pages 229–232. IEEE.

[10] Sun, L., Huo, Q., Jia, W., and Chen, K. (2014). Robust text detection in natural scene images by generalized color-enhanced contrasting extremal region and neural networks. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2715–2720. IEEE.

[11] Tian, S., Pei, W.-Y., Zuo, Z.-Y., and Yin, X.-C. (2016). Scene text detection in video by learning locally and globally. In *Proc. 25th Int. Joint Conf. Artif. Intell.(IJCAI)*.

[12] Wu, L., Shivakumara, P., Lu, T., and Tan, C. L. (2015). A new technique for multioriented scene text line detection and tracking in video. *IEEE Transactions on Multimedia*,17(8):1137–1152.

[13] Yang, H., Quehl, B., and Sack, H. (2014). A framework for improved video text detection and recognition. *Multimedia Tools and Applications*, 69(1):217–245.

[14] Yao, C., Bai, X., and Liu,W. (2014). A unified framework for multioriented text detection and recognition. *IEEE Transactions on Image Processing*, 23(11):4737–4749.

[15] Ye, Q. and Doermann, D. (2015). Text detection and recognition in imagery: A survey.*IEEE transactions on pattern analysis and machine intelligence*, 37(7):1480–1500.

[16] Yin, X.-C., Zuo, Z.-Y., Tian, S., and Liu, C.-L. (2016). Text detection, tracking and recognition in video: a comprehensive survey. *IEEE Transactions on Image Processing*,25(6):2752–2773.

[17] Yu, C., Song, Y., Meng, Q., Zhang, Y., and Liu, Y. (2015). Text detection and recognition in natural scene with edge analysis. *IET Computer Vision*, 9(4):603–613.

[18] Yusufu, T., Wang, Y., and Fang, X. (2013). A video text detection and tracking system.In *Multimedia (ISM), 2013 IEEE International Symposium on*, pages 522–529. IEEE.