Enhanced Stroke Width Transform to Detect Text Regions in Natural Scene Images

Mahitha G[1], Surabhi K R[2], Rahul Kumar[3]
Assistant Professors[1, 2], BE Student [3]
[1, 2, 3] *Department of Information Science and Engineering, DSATM, Bangalore, India*

ABSTRACT

Detection of text in natural images is a vastly sought field in machine learning, and there are frequent approaches to resolve this problem. The various appearance of text and intricacy of scene context make the text detection, a challenging task. A text detection system comprises of commanding and trustworthy tools to detect the regions of text in any natural images to accomplish the high-quality end result. A proper segmentation and feature extraction technique leads to the accurate detection of text region in which text is present. Improved Text detection system discovers the letters which were previously vanished by diminishing the noise of the image. The text regions may have several missing letters in them; however the bounding box of those regions contains them. The result of the text detection system is the textual region, containing only text which is recognized by OCR engine, which then can be edited.

Keywords: Machine Learning, Computer Vision, Segmentation, Feature Extraction, OCR

I. INTRODUCTION

The text detection scenario from the natural images has been broadly studied, and has ended with many accomplishments in the field of machine learning. Detecting text in natural images is a significant fragment of applications in Computer Vision, such as onscreen assist for visually weakened, robotic and automatic navigation in metropolitan environments. Retrieval of texts in both internal and external surroundings presents contextual intimations for multiplicity of vision tasks. Furthermore, the performance of algorithms on image retrieval depends strictly on their modules of text detection and the complex behavior of the image backgrounds. For e.g. two book covers having similar structure but with dissimilar text, demonstrate to be virtually impossible to differentiate without detecting and optical character recognizing the text as shown in Fig.1. In a natural scene, there appears to be a tradeoff between the amount of restrictions (i.e. specific languages, style, scale and direction of text) applied and the quality of result.

The approaches for text detection can be classified into two categories: Sliding Window-based method and Connected Component Analysis (CCA) method. Sliding window-based methods scan the entire image with the help of a sliding window and filters out the features to foretell whether the window has text or not. While CCA method commonly initiates with filtering potential textual regions by several CC extraction approaches and then extracting out non-textual regions. The CC extraction methods exhibit Niblack's binarization method [6], Stroke Width Transform [1] and Maximally Stable Extremal Region (MSER) [7].

In this paper, an influential tool to detect the regions of text has been proposed by using Stroke Width Transform (SWT) [1]. The idea of SWT is gathering pixels together in an intellectual way, rather than looking for straightening out pixel features. By using SWT, the assumptions described in section III satisfies, and maintains a good quality result. The motive here is to improve and implement the algorithm as defined in [1], so almost maximum of the text in a natural image will be detected, possibly with very less noise.

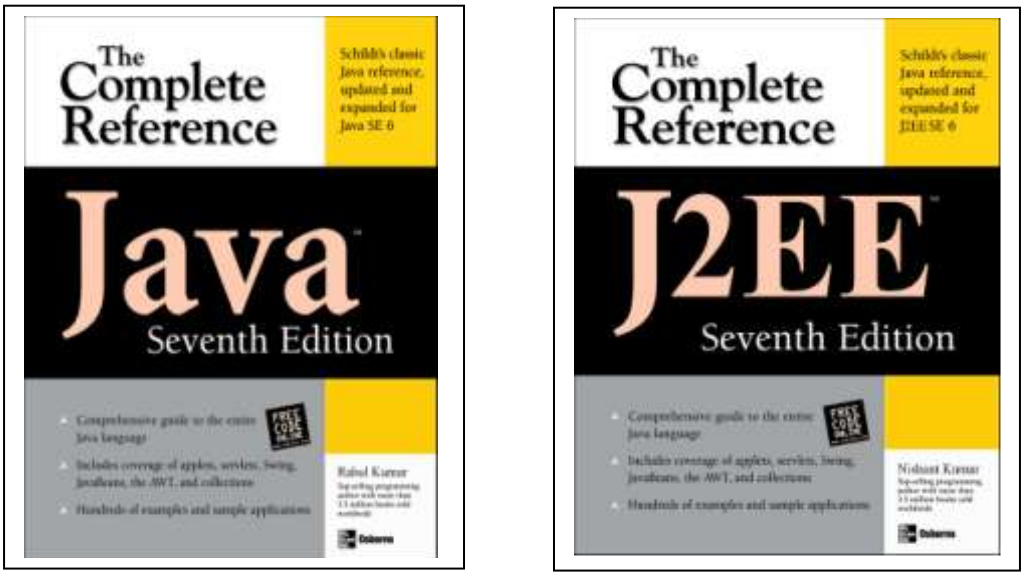


Fig 1: Two book covers of similar design but with different text.

One of the most vital operations in real-time processing like pattern analysis, computer vision and machine intelligence is Labeling Connected Components. Assigning unique label to each connected component in a binary image is used as the base information for high level image processing algorithms. A Connected component in the binary image entails of connected pixels. The purpose of connected component algorithm is to find the connected components which are individual in a binary image and uniquely label them as individual objects. Afterward, the objects which are labeled can be inspected to locate the object boundaries, shape, area, etc. The performance of OCR engine can be exceedingly enhanced by indentifying the proper regions of the available text.

II. LITERATURE SURVEY

A proficient algorithm to automatically detect, localize and extract horizontally aligned text in images with complex backgrounds is presented. The proposal is based on employing a color reduction technique, a method for edge detection and region segmentation, and selecting text regions based on their horizontal projection and geometrical properties. The anticipated approach is based on the application of a color reduction technique, a technique for edge detection, and the localization of text regions using projection profile analysis and geometrical properties. Experimental results on a set of images have demonstrated the performance of the approach, achieving an overall recall of 88.7w and a precision of 83.9w. The algorithm has an output in the form of textbox with a rectified background, which then can be given to an OCR engine for required character recognition. The proposed algorithm works roughly in case of various font sizes, font colors, languages and background complexities. The performance of the method is confirmed by presenting capable practical results for a group of images. The approach can be improved to also work on videos instead of static images. To execute a hybrid system where, CC-based methods are combined with texture-based methods, to achieve further improvements in the performance [2].

An unsupervised learning method for text detection and localization in images with complex backgrounds has been proposed. The main feature for subsequent classification process was to use the standard deviation of histograms of high-frequency wavelet coefficients. After that, the text areas in the image are classified using k-means algorithm. The detected text areas experience a projection analysis in order to refine their localization. Finally, a binary segmented text image is used as input to an OCR engine. The fruitful experimental results show that the chosen features are robust with respect to the distinction between textual regions and non-textual regions in an image. The improvisation of the proposed text detection approach to videos will be implemented. The MPEG motion information for text tracking purposes will be also examined. Furthermore, the integration of OCR system will be examined to support the whole processing chain from the input image to the ASCII text in the end [3].

The AdaBoost algorithm uses a strong classifier for detecting text in unconstrained city scenes. In addition, log-likelihood ratio is used to test on the joint probability distributions of pairs of features. The resulting strong classifier is very effective on the dataset taken by blind users and normally sighted subjects. From this dataset, the text regions can be labeled and extracted manually. The feature set selects a key element which has features with less entropy of the positive training examples. The rates of detection examined only in false positive rate. Adaboost algorithm takes weak classifiers as input to train a strong classifier. In observation, a flow with 4 strong classifiers containing 79 features is trained. OCR engine is there to detect the text or reject if non-textual region. The complete algorithm has a achievement rate of over 90% on the test set. Presented idea can be improved further for better performance and accuracy [4].

A new CC-based approach was introduced for examination of images for text-localization and extraction which detects character strokes. The approach places very few restrictions on the font size and color of text and is able to handle the both scene text and artificial text as well. A detected stroke is used to estimate a threshold and stroke-width which are used for character segmentation. A 3-stage algorithm is there to detect strokes along a horizontal check of the intensity image has been described. The reliability of the algorithm to 3 key factors is estimated for 4 criteria: stroke precision, character recall, word recall, and computing time.

Stages	**Features**
1.	Adequate difference between a character and its immediate neighbors.
2.	Acceptance for distinction in color (and intensity) within a character boosts directly to the character's color-contrast with its neighbors.
3.	The stroke of characters (of a word) has similar width and is moderately close to each other.

The algorithm was given to work well on several natural scene images. Even though the text detection application works fine on most fonts, and works badly on italic fonts or when there is overlapping of characters. So, the characters are too disposed and their stroke widths are difficult to detect. The system uses strengthen images to detect changes. If the work is directly done on the color space to detect character strokes then performance can be improved [5].

A mixed approach to vigorously detect and localize texts in natural images is presented by integrating region information into a robust CC-based method. A system is designed to divide candidate text components by local binarization. The binary contextual component is incorporated in a conditional random field (CRF) model, whose parameters are together optimized by supervised learning. At last, text components are grouped into text lines/words. The practical results observed that the implemented method is effective in unconstrained scene text localization in various respects:

- For text component segmentation and analysis, region-based information is very helpful;
- Incorporating contextual component relationships, the CRF model distinguishes text components from non-textual components better than local classifiers;
- Combined optimization of base classifier parameters with CRF is advantageous; and
- Text components can be grouped into text lines (words) by using learning-based energy minimization method.

The approach is evaluated on a multilingual image dataset with promising results. It fails on some hard-to-segment texts. Taking color information into account is one possible solution. Text detection can also help to lessen false positives of text extraction [6].

An accurate and robust method for detecting texts in natural scene images is proposed over traditional methods. A quick and successful pruning algorithm is implemented to extract Maximally Stable Extremal Regions (MSERs) as letter candidates using the tactic of reducing regularized variations which detect most the letters even when the image is in low quality. The single-link clustering algorithm helps in grouping character candidates into text candidates, where distance weights and threshold of the clustering algorithm comes under self-training distance metric learning algorithm. The later chances of text candidates with respect to non-textual regions are evaluated with a letter classifier; text candidates with high chances are then removed and finally texts are recognized with a text classifier [7].

A simple and efficient run-based connected component algorithm in a binary image to reduce the number of conditional labels and the number of access to memory for labeling has been proposed. Furthermore it does not use merging operation for determining label equivalences among conditional labels (sets), but it uses post processing stage to reduce complexity for resolving label equivalency. During the first scan, the post processing is prepared to determine label equivalency. The smallest conditional label among all conditional labels that are allocated to a connected component is assumed as a representative label and obtains the run data. During the second scan, the run data is used to access foreground pixels directly and assigns conditional labels to them. Practical results on variety of images show that the proposed algorithm is better than the conventional labeling algorithms and is more proficient than other algorithms [8].

The text detection system follows 4 connected steps: character candidate detection, false character candidate removal, text line extraction, and text line verification. On the other hand, errors come and gathered all the way through each of these connected steps which usually tend to decrease detection performance. To overcome these issues, a unified scene text detection system, named Text Flow, by operating the min-cost flow network model had been proposed. The text detection system comprises of 2 steps: character candidate detection managed by cascade boosting and text line extraction resolved by a min-cost flow network. The proposed system can capture the whole character instead of isolated character strokes and the processing time is comparable with the Connected Components based techniques. The projected technique can be used for detecting text in different languages, if the performance is improved on multilingual dataset. Besides, the implemented system is better in identifying texts in other non-Latin languages [9].

A text detection system featured on multi-scale Stroke Width transform (SWT) with 2 edge-based features, i.e. stroke pair ratio (SPR) and edge density of a connected component (EDC), to detect text in scene images. Edge components are edited using two reliable features. Then, the required edge components on each level are gathered into text lines. And these lines are reversed back onto a single image and merged. At last, text lines of the candidate are verified by integrating block level features and line level features. SWT detects the text defected by reflection, illumination, blurring, etc. In addition, the two features EDC and SPR both are reliable for filtering background edges. Furthermore, improvements can be done in terms of SWT algorithm implementation and the accuracy and the performance can be improved further [10].

A novel method for scene text detection and segmentation based on cascaded convolution neural networks (CNNs) consists of 3 steps including text-aware CTR extraction, CTR refinement, and CTR classification. All steps are accomplished by adopting convolution neural networks, which makes the proposed method more robust and effective than other approaches. A CNN-based CTR refinement model (named segmentation network, SNet) is then implemented to precisely divide the coarse CTRs into text to get the refined CTRs. The proposed text-aware CTR extraction model can extract more true text regions and much fewer false text regions than other approaches. The CTR refinement model can effectively remove the background from each CTR and the false text regions. The CNN-based CTR classification model can correctly classify the CTRs and get the true text regions. With all of these processes, the proposed method gets the best recall, F-measure and comparable precision on three benchmark datasets. The proposed method also has the ability to find some text which is not annotated in the ground truth due to the power of all of the CNN networks used [11].

III. PROBLEM STATEMENT AND ASSUMPTIONS

The text recognition system receives a natural image either by camera or scanning or fax or business cards. It can receive any image of JPG/PNG format. The regular size of the tested image is to be 800x600 (for larger images the system will take a bit longer time). The image need not have to be of best quality, but the text should not be blurry. The text in the image can be of variety of sizes. Also, the recognition is dependent on the text language and independent of the direction of it. Curved text will be more difficult to detect, yet it is still possible. Furthermore, the detected text in an implementation can either be dark text on a light background or light text on a dark background. This is owed to the features of the SWT, as described later on. The result exhibit the recognized characters grouped into a region. These regions possibly will have several letters missing in them; however, as long as the bounding box of the region contains them, the excluding is not as problematic.

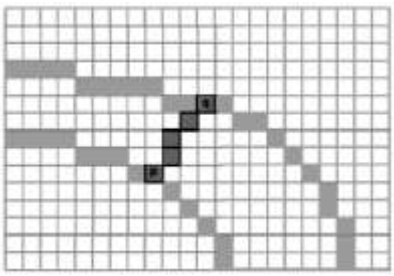
IV. STROKE WIDTH TRANSFORM

In this section, the Stroke Width Transform algorithm as it is presented in [1] has been described with several enhancements and improvements. The algorithm takes an RGB/PNG image and gives back an image of the similar size, where the supposed text regions are marked. This algorithm has 3 main steps: stroke width transform, finding the character candidates depending on their stroke width values, and grouping character candidates into text regions.

Stroke Width Transform

A stroke in an image is a continuous crew of an almost constant width. These are the dominant element in written and typed characters, but are relatively rare elsewhere, shown in Fig: 2.

(a) (b)



(c)

Fig 2: Implementation of SWT

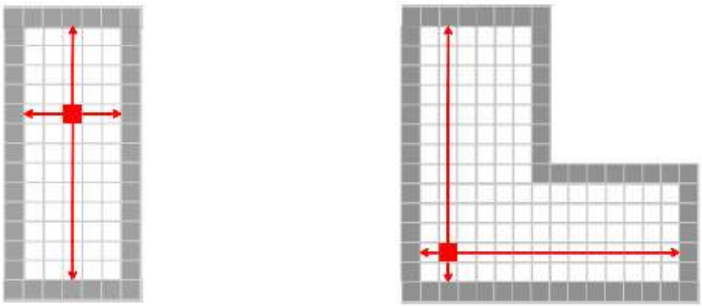
The stroke SWT is a local image operator which computes for each pixel the width of the most likely stroke containing the pixel. The original width of the stroke is not known but rather it could be recovered.

Initially, the stroke width of all the pixels is assigned to ∞. Then in order to recover the strokes, the edge of the image is calculated using Canny Edge Detector. The edges are considered as stroke boundaries, and the width of such stroke is going to be found. If **p** is an edge pixel and lies on stroke boundary, then the gradient direction is orthogonal to the direction of the stroke boundary. The direction of gradient $\mathbf{d_p}$ of the edge pixels is computed in the next step, and tracks the ray $\mathbf{r=p+n*d_p}$ (n>0) until the next edge pixel **q** is found. Now, the gradient direction $\mathbf{d_q}$ is considered at **q** and if $\mathbf{d_q}$ is orthogonal to $\mathbf{d_p}$ ($\mathbf{dq=-dp\pm\pi/6}$), then the ray in every pixel is set with the distance between **p** and **q** as their stroke width, unless it already has a less value, shown in Fig 3(a). Otherwise, if, an edge pixel **q** is not found, or if $\mathbf{d_q}$ is not opposite to $\mathbf{d_p}$, the ray is rejected. In order to

International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.07, April 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

accommodate both dark on light and light on dark condition, the algorithm will be applied two times: once with the ray direction $\mathbf{d_p}$ and then with $\mathbf{-d_p}$.

According to the above methodology, pixels in intricate situations, like corners, will not hold the true stroke widths value, Fig: 3(b). Therefore, through all each non-discarded ray will be passed, where the each and every ray pixels will get the minimum value between its current values and calculate the median SWT value **m** along that ray. (In the novel algorithm, the median value is set to the pixels, the improvised experiments tends to the better results when the minimal value has been taken).



(a) (b)

Fig 3: Filling pixels with SWT values

Discarding single lines from SW-Map

For the advancement of the results, addition of a step is required, whose goal is to get better character separation. Several times the characters in the image get attached with each other and identify a set of characters as a single word. As the features of a set of characters do not essentially comply with the characteristics of a single character, these words might be rejected in the next step. While analyzing the stages of the algorithm, it has been observed that the stroke width operator gives back many single lines that bond characters together. Once such lines are removed, these characters are no longer regard as the part of the same word. Thus, a stage has been added in which the algorithm scans all the pixels, and if its neighbor does not have pixels enough from its component, then from the component, the pixel is removed.

Finding Character Candidates

Now, there is a map of the most possible stroke-widths for every pixel in the original image. The next step is to cluster these pixels obsessed on letter candidate. This is done when by neighboring pixels are grouped first with similar stroke width, and then applying numerous rules to differentiate the letter candidates.

A simplified and efficient Connected Component algorithm is used for grouping the image by changing the association rule from a binary mask to a predicate that compares the SWT values of pixels. In order to permit smoothly varying stroke widths in a letter, assume two pixels to be grouped jointly if their SWT ratio is less than 3.0.

Now the connected components are detected that can be passed as letter candidates, by applying a set of moderately flexibly rules. These rules are as follows:

- Within a component, the variance of the stroke-width must not be too large. This helps with decline parts in natural images, which are commonly mistaken for text.
- The aspect ratio of a component must be within a small range of values, in order to decline long and narrow components.
- The ratio between the median stroke width and its diameter of the component to be less than an expected threshold. This also helps in discarding large and thin components.
- Components with size having value with too large or too small will also be rejected. This is done by restricting the length, width, and pixel count of the component.
- Further, another rule has been added which helped in eliminating extra noise. This rule state that the ratio between the pixel count of the component and the amount of pixels in the bounding box of the component should be within a bounded range. This rejects components that spread over a huge space, yet have a small pixel count, and components which cover most of their bounding box.

The thresholds used initially have been taken from the SWT description [1], and were slightly updated according to the results. The remaining connected components are considered as letter candidates, and they are now to be integrated into regions of text.

Grouping character candidates into text regions

Since single characters are not likely to appear in images, try to group strongly positioned letter candidates into regions of text. This strains out many pseudo-identified letter candidates, and improves the consistency of the algorithm results.

Again, letters are grouped together into regions of text by using set of small rules. These rules will believe pairs of letters, and are as follows:

- Two letter candidates should pose similar stroke width. Therefore, limit the ratio between the median stroke-widths to be less than some threshold.
- The ratio between the widths of the letters and between the heights of the letters should not be greater than 2.5. This is because of the appearance of upper case letters next to lower case letters.
- The distance between letters should be 3 times lesser than the width of the wider one.
- Same word characters should similar color; therefore, for pairing compare the average color of the candidates.
- Again, add another rule which restricts the pixels count of the pair of letters in the bounding box of the pair.

While deciding to pair two letters together, there are 2 options- either both letters were not allocated a region yet, or one of them has already grouped with additional letters. If both are not assigned, all require to be done is to declare a new region and allocate them to it. Otherwise, it's needed to check, if addition of one letter to the region of the other is reasonable. In improved implementation, a join is reasonable if the count of pixel in the letters is divided by the size of the bounding box containing the combination of all the letters is not less than some threshold. This will make sure the region of text will not have loose ends, and will form a bounding box. The approach in the original algorithm differs with this improved approach, which gathers letters together into chains. Finally, regions with less than three letters are discarded. The system architecture of the algorithm is shown in Fig 4. The implementation mentioned here is up to the determining the text region phase.

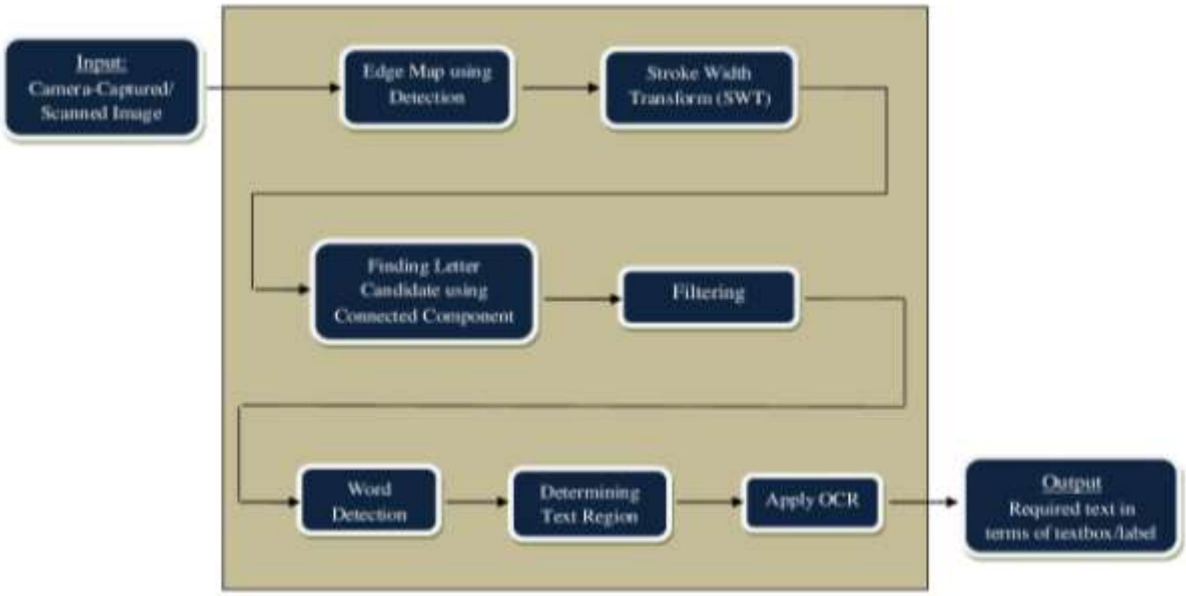


Fig 4: System Architecture of Text Detection and Recognition System

I. APPLICATION

The SWT application is implemented to detect a mark the regions of an image that are supposed to contain text. The application receives an RGB/PNG image, and checks whether the text to search is dark-on-light or light on dark. An image of the same size is returned and pixels of each text detected region are marked. The value of each pixel belongs to the labeled region in the image if 0 appears; it means that it doesn't correspond to any region. This application not just shows the bounding box of the region but also shows the better efficiency of the algorithm.

- **Indexing of Multimedia Archives**

The text data present inside multimedia contains the information about automatic annotation and indexing. Extracted information is used for recognition of the scene text from the video or image. Retrieval of videos and images can be done by the extracted text.

- **Recognizing Signs in Driver Assisted Systems**

Road conditions have been drastically improved in today's world as compared with past decade. Express highways outfitted with extended lane made up with cement concrete. Apparently speed of the vehicles has been increased. So according to driver's point of view there is a lot of chances of ignoring mandatory road sign while driving. The automatic recognition of road-signs is a significant part of Driver Assisting Systems which helps driver to increase safety and driving comfort.

The main motive of driving assistance systems is to gather important information for drivers in order to lessen their effort in safe driving. Drivers have to take care of various conditions, which include vehicle speed and orientation, passing cars, the distance between vehicles, and unusual events or potential dangerous ahead. If driver assistance systems are aforementioned with such information, it will surely lessen the burden of driving for drivers and make driving safer and easier.

- **Recognizing Vehicles by Reading Their License Plates**

Vehicles have a significant role in transportation. Population growth leads to increase in vehicles. Consequently, controlling of vehicles is becoming a big dilemma. So, vehicles should recognize properly. Vehicle registration plate can be used since every vehicle will have unique vehicle number plate. There is a necessity for automation of process of vehicle plate recognition because manual recording of vehicles is not efficient, costly and time consuming. Automatic License Vehicle Plate Recognition (ALVPR) is a pattern recognition and image processing problem. Automatic recognition of license plates in car has a significant role in traffic surveillance systems. The applications of such systems can be find in parking areas, highways, bridges and tunnels which help a human operator and improve overall quality of a service.

Automatic recognition of number plate is a mass surveillance method that uses OCR on images to read vehicle registration plates. The main purpose of ALVPR is security. This method is very supportive in parking management, toll collection, access control, radar based speed control, road patrolling and boarder control.

- **Text Extraction from Camera-captured Scene**

With the sudden increase and extensive dispersion of low-priced digital cameras and mobile phones capable with good quality cameras, text extraction from camera-captured scenes has achieved a renewed attention in computer vision research.

- **Automatic Post**

Automated post and courier processing requires detection and recognition of the address block on the cover. It is difficult because the syntax of address is not strictly uniform, it could be machine printed or handwritten, or it may be cluttered by other text and images such as label of the seller has an extensive review of different techniques used for automatic mail sorting. The following section discusses some common techniques used specifically for automatic postal mail processing. The key to sorting mail is to recognize the postal zip code. The preprocessing step accounts for extracting and enhancing the zip code. Segmentation involves separation of each digit in the zip code. Features are extracted which will distinguish one digit from the other, which can be used for recognition of the digit.

Improvement Approaches

The implementation differs from the original algorithm in several aspects:

First is the addition of another rule to the 'discovery of letter candidates' phase. Given rule, as mentioned in the previous section, extracts out components which spread over too much space, compared to the amount of pixels in them (their size). For example:

Fig 5: The component clearly spreads over too much space, even though other features might fit the features of a character

International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.07, April 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

Similar rule to the ‘aggregation of letters’ phase has been added, which restricts the pairing of letters. This is to ensure that two letters are combined into the same region only if the bounding box surrounding them is of reasonable size compared to the size of the letters. For example:

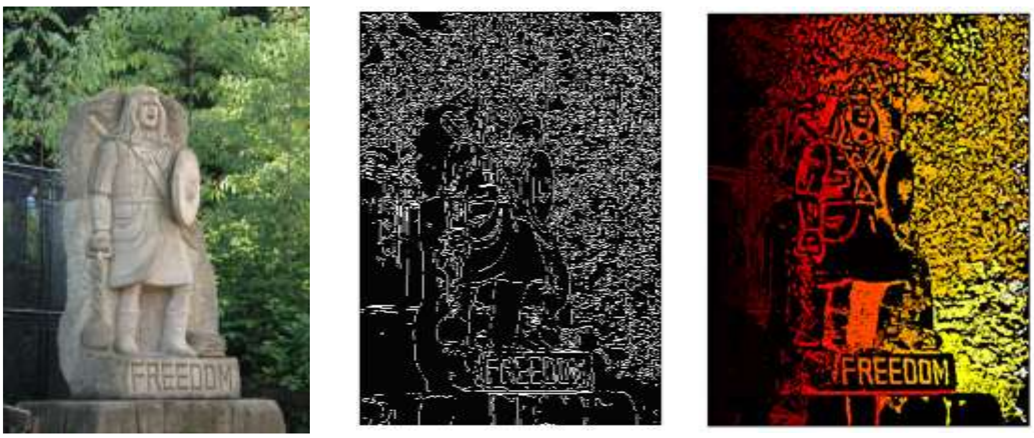
Fig 6: The left mage shows two components that will not be grouped, while the right image shows two components that can be grouped together

In attempt to improve the results, another step is added to the algorithm. It has been observed that many times the SW-map contains lines that connect different components of the image. Those lines may attach far away elements, or element close by. The most damage, as it has been observed, is done by connecting close elements. In this way, letters which are close to each other will be grouped into one component. The thresholds for character and region recognition are forced to be less strict, hence allowing more noise to appear in the result.

The above step iterates through each pixel and checks its neighbor pixels: if it contains 3 or less than 3 pixels with the same stroke width label, the label is removed from that pixel. In this way, stray pixels and single lines are discarded, and some components will be falsely grouped together. Next section comprises the experimental results. Another difference added to the SWT algorithm was reducing some restrictions to the separation algorithms. Separate the letter candidates into into regions of text. The purpose for this change is to detect and locate text in many directions, including curvy text.

V. EXPERIMENTS AND RESULTS

Here, the outcome of the text detection system is shown on some set of images, and compare the results to the different approaches as discussed previously.



(1)

(2)

(3)

Fig 7: Different steps of algorithm

Strengths of this algorithm:

As seen in the above images, the improved text detection system detects the letter of different fonts and sizes or letters. The text can be of varying size and different orientations (including curvy text), and can be detected by this improved system. Even handwritten text can be also detected (3).

Weakness of this algorithm:

In some cases, little noise can be detected in the result (1). This usually happens when there is some tampering in the image. The features of tamper resemble those of letters, and might give a pseudo detection of letter candidates. This can be seen in the letters of examples 1 and 2.

Sometimes, the application does not handle round and curved letters. The cursive letters are not recognized, as opposed to the print letters. Similarly, curved lines produce weak results. This varies with the level of strictness in the 'grouping letter candidates into text regions' phase. If thresholds are reliable then, more letters will be grouped together, yet more noise will appear as well. Further weakness has been discovered were small and close letters tend to be grouped together in the SW labeling phase. Since a group of letters behaves differently than a single letter, these groups may be dismissed in the 'finding letter candidates' phase. For example, in Fig 8, the word ROADS has not recognized since the letters 'ROA' were labeled as a single component and their features together differs from the features of a single letter. Even though D and S were recognized, it has diminished since a region is expected to contain at least 3 letters.

Fig: R,O and A are labeled as a single component

In attempt to avoid these occasions, the phase has been added where stray pixels are removed from the SW map. In above example, the pixel connecting R and O, and the pixel connecting O and A should be discarded, hence allowing to recognize the letters ROA as three separate letters.

VI. FUTURE WORK AND CONCLUSION

In this paper, it is been concluded that the improved SWT algorithm detects the text regions in natural images and recognize the text from the same detected textual area, even it detects the curvy texts and handwritten text regions also. The accuracy of text detection depends on the complexity of the background of the images and also the technique of segmentation and feature extraction of the input image. An improved labeling method of components can improve the detection of characters and will allow the use of harsher thresholds. This way, better results for circular text can be detected, which tends to be dismissed as noise due to the grouping of the letters. This also allows identifying curvy letters better. Various techniques and algorithms of image segmentation, edge detection, feature extraction and clustering/connecting components are reviewed and discussed.

REFERENCES

[1]. B. Epshtein, E. Ofek, and Y. Wexler, "*Detecting text in natural scenes with stroke width transform,"* in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, *pp. 2963-2970.*

[2] J. Gllavata, R. Ewerth, B. Freisleben, *"A Robust Algorithm for Text Detection in Images" in Image and Signal Processing and Analysis*, 2003. ISPA 2003. Proceedings of the 3rd International Symposium on 2003, IEEE Conference on, 2003.

[3]. J. Gllavata, R. Ewerth, B. Freisleben, "*Text Detection in Images Based on Unsupervised Classification of High-Frequency Wavelet Coefficients"*, 17th International Conference on Pattern Recognition (ICPR'04) – Vol.- 1, IEEE Conference on, 2004, pp. 425-428.

[4]. X. Chen, A. Yuille, *"Detecting and Reading Text in Natural Scenes"*, Computer Vision and Pattern Recognition (CVPR), 2004 IEEE Conference on, 2004, *pp. 366-373.*

[5]. K. Subramanian, P. Natarajan, M. Decerbo, D. Castañòn, *"Character-Stroke Detection for Text-Localization and Extraction"*, International Conference on Document Analysis and Recognition (ICDAR), 2005.

[6]. Yi-F. Pan, X. Hou, and C-Lin Liu, Senior Member, IEEE, *"A Hybrid Approach to Detect and Localize Texts in Natural Scene Images"*, IEEE Transactions On Image Processing, *Volume.-20, 2011* IEEE Conference on, 2011.

[7]. X.C. Yin, X. Yin, K. Huang, Hong-Wei Hao, *"Robust Text Detection in Natural Scene Images"*, IEEE Transactions on Pattern Analysis and Machine Intelligence *2014 – Vol.- 36, pp. 970-983.*

[8]. M.M. Gharasuie, A. Gaffari, *"An Efficient Run-Based Method for Connected Component Labeling"*, in 9th Iranian Conference on Machine Vision and Image Processing (MVIP), 2015 IEEE Conference on, 2015.

[9]. S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, C. L. Tan, *"Text Flow: A Unified Text Detection System in Natural Scene Images"*, in IEEE International Conference on Computer Vision 2015, on IEEE Conference 2015.

[10]. Y. Feng, Y. Song, Y. Zhang, *"Scene Text Detection Based On Multi-Scale SWT And Edge Filtering*", in International Conference on Pattern Recognition (ICPR), 2016 23rd IEEE Conference on, 2016, *pp. 645-650.*

[11]. Y.Tang, X. Wu *"Scene Text Detection and Segmentation Based on Cascaded Convolution Neural Networks"*, in IEEE Transactions on Image Processing, 2017,IEEE Conference on, 2017, *Vol.-26, NO.3.*

[12].E. Ahmed, A. Clark, G. Mohay, *"A Novel Sliding Window Based Change Detection Algorithm for Asymmetric Traffic"*, 2008 IFIP International Conference on Network and Parallel Computing, 2008 IEEE Conference on, 2008.

[13]. J. Canny, *"A Computational Approach to Edge Detection",* IEEE Trans. Pattern Analysis and Machine Intelligence, *8:679- 714, 1986.*

[14]. A. AbuBaker, R. Qahwaji, S. Ipson, Md. Saleh, *"One Scan Connected Component Labeling Technique"*, 2007 IEEE International Conference on Signal Processing and Communications (ICSPC 2007), *pp. 1283-1286.*