ISSN (Online): 2455-7838
SJIF Impact Factor (2017): 5.705

EPRA International Journal of

Research & Development

(IJRD)

Monthly Peer Reviewed & Indexed International Online Journal

Volume: 3, Issue:11,November 2018

Published By :
EPRA Journals

SJIF Impact Factor: 5.705 Volume: 3 | Issue: 11 | November | 2018 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)

SURVEY OF TEXT DETECTION METHODS IN SCENE IMAGES

Sachi D. Agrawal
Information & Technology Dept. Pimpri Chinchwad College of Engineering
Sector No. 26, Pradhikaran, Nigdi, Pune,Maharashtra, 411044

Mrs. V.C. Kulloli
PhD perusing, Information & Technology Dept. Pimpri Chinchwad College of Engineering, Sector No. 26, Pradhikaran, Nigdi, Pune, Maharashtra, 411044

ABSTRACT
Reading the text embedded in natural scene images is essential to many applications. In this paper, we compared the methods for detecting text in scene images based on multi-level connected component (CC) analysis and learning text component features via convolutional neural networks (CNN), followed by a graph-based grouping of overlapping text boxes, Bidirectional Information Aggregation, Morphological Techniques, Enhanced Multi-channels MSER, etc. The multi-level CC analysis allows the extraction of redundant text and non-text components at multiple binarization levels to minimize the loss of any potential text candidates. The features of the resulting raw text/non-text components of different granularity levels are learned via a CNN. Those two modules eliminate the need for complex ad-hoc preprocessing steps for finding initial candidates, and the need for hand-designed features to classify such candidates into text or non-text. The components classified as text at different granularity levels, are grouped in a graph based on the overlap of their extended bounding boxes, then, the connected graph components are retained. This eliminates redundant text components and forms words or text lines.

KEYWORDS: *Scene text detection; CNN; multi-level binarization; multi-level connected components; graph-based grouping; bidirectional infor*

I. INTRODUCTION

Optical Character Recognition:

Optical character recognition (also optical character reader, OCR) 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).

Widely used as a form of information entry from printed paper data records – whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation – it is a common method of digitizing 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.

Text Detection:

Recent deep learning models have demonstrated strong capabilities for classifying text and non-text components in natural images. They extract a high-level feature globally computed from a whole image component, where the cluttered background information may dominate true text features in the deep representation. This leads to less discriminative power and poorer robustness. Reading the text embedded in natural scene images is essential to many applications. Text appears everywhere in our natural surrounding environments such as in traffic signs, tags, license plates, advertisement, billboards, business cards, building signs, labels on posted parcels and on name plates.

Text detection and recognition in natural images have received increasing attention in computer vision and image understanding, due to its numerous potential applications in image retrieval, scene understanding, visual assistance, etc. Though tremendous efforts have recently been devoted to improving its performance, reading texts in unconstrained environments is still extremely challenging and remains an open problem. Most text detection systems are mainly composed of three stages. Firstly, finding character/word candidates or regions of interest. This could be done at pixel, interest point or zone levels. Usually, this stage is the most challenging and involves many complicated preprocessing steps. Secondly, the filtering stages where initial candidates are classified as text or non-text components.

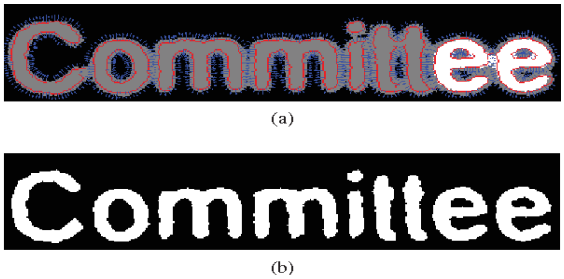

Fig.1. Example of Text detection in scene images

Text appears everywhere in our natural surrounding environments such as in traffic signs, license plates, advertisement billboards, business cards, building signs, labels on posted parcels and on name plates. The textual content in these images is a valuable source of information and useful for many applications such as interactive tourists' guidance, data mining and providing text accessibility for visually impaired people whether reading such text is a necessity for their everyday life or simply for navigating or enjoying the world around them.

Although it bears similarity to OCR problems in traditional document images, text detection in scene images is much more challenging due to, on one hand, complex layout with variable backgrounds and the high variations in text color, font, size and orientation, and on the other hand, lighting/shadow/occlusion problems introduced by acquisition conditions. New challenges also emerge in scene images of modern cities such as detecting multi-lingual text.

Most text detection systems are mainly composed of three stages. Firstly, finding character/word candidates or regions of interest. This could be done at pixel, interest point or zone levels. Usually, this stage is the most challenging and involves many complicated preprocessing steps. Secondly, the filtering stage(s) where initial candidates are classified as text or non-text components. Some methods use hand-designed features and multiple filtering steps within this stage. Finally, the grouping stage, in which text components are grouped into characters, words or text lines. Grouping methods are typically not adapted to multi-oriented or multi-lingual text.

II. DIFFERENT TECHNIQUES FOR TEXT DETECTION

1. Fully Convolutional Network Technique

In this paper[1], they propose a novel method for fast arbitrary-oriented text detection in scene images. Proposed method is simple and effective which can predict word-level bounding boxes via a single fully convolutional network. Method extracts features from the input images by residual network and apply multi-level fusion over the extracted features. It has two outputs, pixel-wise classification between text and non-text and word-level bounding boxes.

Feature Extraction : Feature extraction part is used to extract deep features of scene texts. they use classical ResNet50[2] to construct the convolutional feature ex-traction part. Several advantages make ResNet50 suitable for this task. Firstly, compared with other classic convolutional neural network like VGG net, ResNet50 has fewer parameters. Text feature is not as complicated as generic objects,

hence a large number of parameters is not needed. Also, less parameters can reduce the cost of computation and accelerate the detection speed which is quite important.

Multi-level Feature Fusion : Multi-level feature fusion part is used to fuse features from different feature layers extracted from ResNet50. High level features are required for detecting of large texts, while small texts require low level features. With this part, this model is capable of detecting texts of multiple scales. What's more, coarse and semantic information provided by high level features help the model localize the text region while local and appearance information from low level features help the model localize the edges of texts precisely.

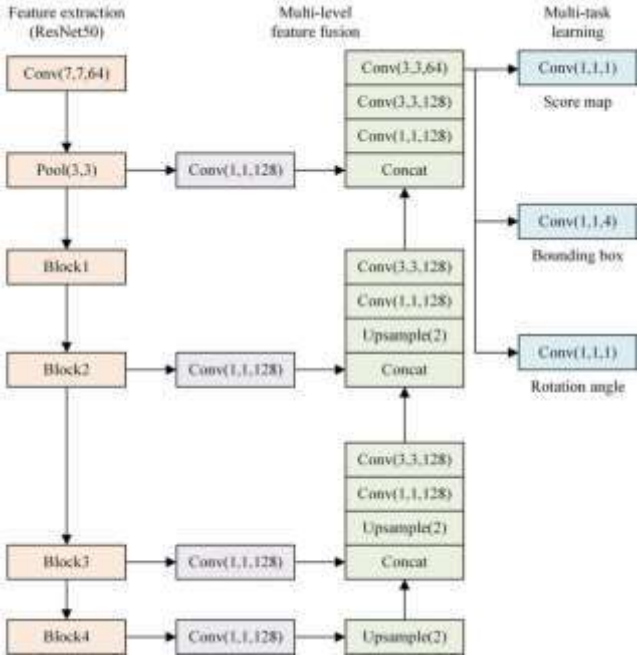

Fig.2. Network architecture

Multi-task Learning : Multi-task learning part is the output layer of the network, it has two tasks: classification task and localization task. Classification task is used to segment text and non-text. This task outputs a one channel score map between text and non-text. Pixels with higher score are more likely to be in the text region, this is the foundation of proposed method. Localization task has two sub-tasks, one subtask predicts four distances between current pixel location and the top, right, bottom, left boundaries of the text's bounding box. The other subtask outputs the rotation angle of bounding box predicted by current pix-el in order to detect arbitrary-oriented texts. Combine the classification task and localization task, they can get a dense prediction of rotated boxes. Since we only up-sample the fused features to one quarter size of the input image, every 4 $\times$ 4 pixels area in the original input image would predict a rotated box.

Post-processing : In post-processing part, bounding boxes predicted by pixels with low score in the classification task are deleted in order to filter the non-text areas. Then a standard NMS is used to delete the redundant bounding boxes.

2. Convolutional Neural Network

In this paper[3], they propose a method for detecting text in scene images based on multi-level connected component (CC) analysis and learning text component features via convolutional neural networks (CNN), followed by a graph-based grouping of overlapping text boxes. The multi-level CC analysis allows the extraction of redundant text and non-text components at multiple binarization levels to minimize the loss of any potential text candidates. The features of the resulting raw text/non-text components of different granularity levels are learned via a CNN. Those two modules eliminate the need for complex ad-hoc preprocessing steps for finding initial candidates, and the need for hand-designed features to classify such candidates into text or non-text. The components classified as text at different granularity levels, are grouped in a graph based on the overlap of their extended bounding boxes, then, the connected graph components are retained.

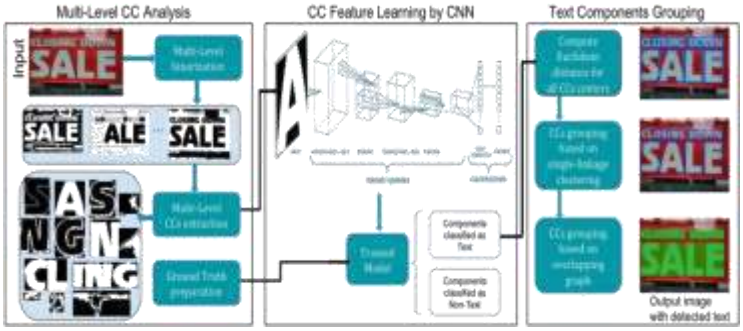

Fig. 3. Block Diagram

Fig.3 shows the architecture of proposed method. The first module handles multi-level connected component extraction where a scene image is fed to this module as input. Multiple binarizations are applied on the input image and its complement before extracting the connected components from each binary image. This ensures the extraction of low-contrast, light-on-dark and low resolution components. The resulting text and non-text components could be broken and/or extracted multiple times.

The second module is a classification module composed of a CNN that learns powerful features of the raw components in the training phase. In the test phase, the trained CNN model classifies the components extracted in module 1 into text or non-text. The third module aims at creating the final output as text words from the components classified as text in module 2. They propose a grouping method in this module that starts by linkage-based clustering to group broken and/or redundant components of the same character/group of characters into the same cluster. Then, text words are formed by finding the connected components of a graph whose edges represent the amount of overlap among the bounding boxes of text components.

i) Multi-Level Connected Components Analysis :

Working at the connected component (CC) level to find initial text candidates is preferred to the pixel-level or interest point level – which are noise-sensitive and slow –, and to the sliding window level which cannot be easily adapted to multiple scales and orientations among other problems. However, CC extraction relies on the lossy binarization step, and may result in broken text components. We propose a multi-level CC extraction that overcomes these challenges.

Fig.4. Multi-level binarization results of two test images

ii) Learning Text Component Features via a CNN Classifier :

The previous multi-level CC extraction module, results in thousands of raw text and non-text components with variable content and size characteristics. The enclosing boxes of these components are fed as separate input images to this classification module. To compute the likelihood of a

component being text or not, the deep features of the components are learned using a CNN classifier.

Fig.5. Samples of extracted connected components from different binarizations. Those are the most frequent type of extracted text components.

iii) Graph-based Grouping of Text Components :

For the third module of this method, They propose a general grouping method which takes as input all the components labeled as text in a test image by the classifier in the previous module. The grouping method consists of two main steps. First, linkage-based clustering which aggregates the redundant and broken text components of the same character(s) into the same clusters. In the second step, a graph is formed based on overlapping criteria of the components bounding boxes of each cluster. The connected graph components form text words.

In the first step, a Euclidean distance matrix is computed between the centers of each two text components. Then, a dendrogram is created by a single-linkage hierarchical clustering. The text component clusters are built in a bottom-up fashion, where at each step, a pair of text components are grouped into the same cluster if they are closest to each other according to the distance matrix. Clusters are formed by merging smaller clusters, and this pair-wise merging process is repeated until no pairs can be further merged. This grouping step forces broken components of the same character(s) to be grouped in one cluster, as well as the redundant versions of the same text component. A bounding box is created for the text components within each cluster. These boxes represent the input text candidates for the next grouping step.

The second step builds a graph where the bounding boxes (text candidates) are the nodes. To create the edges, each two boxes are processed at a time as follows. The overlap between the extended bounding boxes of each two text candidates is computed. Two candidates (nodes) are linked by an edge if their overlap is higher than a threshold that is adaptive to the scale of the boxes. The adjacent nodes in this graph represent parts of the same word in the cases of successful grouping. Finally, the connected components of this graph are extracted as the detected text words.

Fig.6. Grouping steps applied on an image

The advantages of grouping method could be shown through its ability to find very small text components which may be lost in the preceding modules. For example, the dots or the small letters (or parts of letters) would be included in the final word box in the grouping module. By extending the size of the bounding box of a connected component with respect to its original size, the small components will be recovered if they have neighboring text components.

3. Bidirectional Information Aggregation (BIA)

Text Boxes[4] is one of the most advanced text detection method in both aspects of accuracy and efficiency, but it is still not very sensitive to the small text in natural scenes and often can not localize text regions precisely. To tackle these problems, we first present a Bidirectional Information Aggregation (BIA) architecture by effectively aggregating multi-scale feature maps to enhance local details and strengthen context information, making the detector not only work reliably on multi-scale text, especially the small text, but also predict more precise boxes for texts. This architecture also results in a single classifier network, which allows this model to be trained much faster and easily with better generalization power.

i) Bidirectional Information Aggregation

A Bidirectional Information Aggregation (BIA) architecture that effectively aggregate both the upper and lower features, as illustrated in Fig.7(a). The convolutional and box prediction components of the pro-posed architecture mainly inherit from SSD detector[5] and Text Boxes detector respectively. What is different is that we add feature maps of conv3 3 to the feature pyramid and remove the last global pooling layer. The reason for adding feature maps of conv3 3 for detection is that we think feature maps from the lower layers can capture more fine details of the input texts. And removing the last

global pooling layer is because the global pooling layer makes multi-scale inputs unable in our BIA architecture. Besides, we try to add an additional convolutional layer conv11 and use the layer conv11 2 for detection, which can extract higher level abstraction.

In the BIA architecture, pooling and de-convolution are utilized simultaneously to aggregate multi-scale feature maps with an clear relationship among different layers. We employ one convolution layer and one de-convolution layer for downward features aggregation, while one convolution layer and one max pooling layer for upward features aggregation. For instance, Fig.7(b) shows the details of features aggregation in the layer conv7. The two right pointing arrows in the figure signify upward aggregation. The layer conv3 3 use a 1x1 convolution followed by ReLU to generate feature maps of size 16x16x256, and the layer conv4 3 generates feature maps of size 16x16x512 in the same way. Then the generated feature maps are

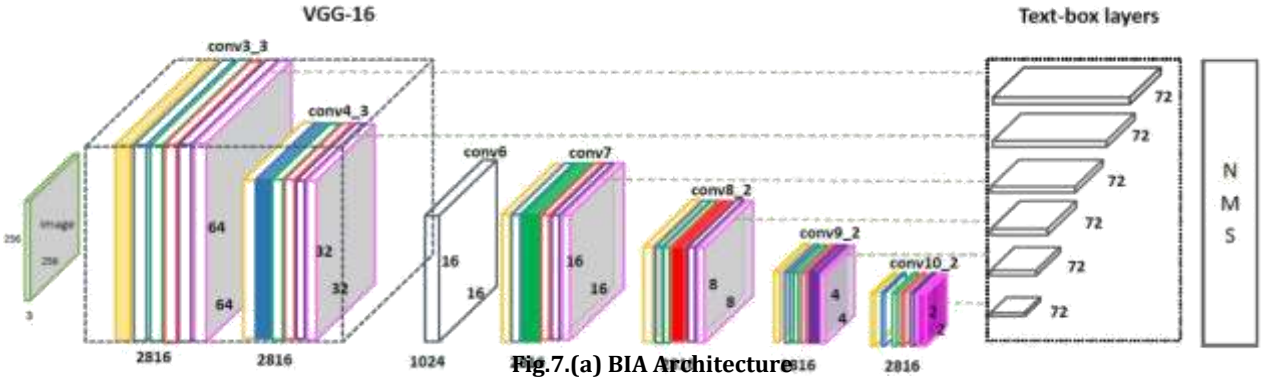

Fig.7.(a) BIA Architecture

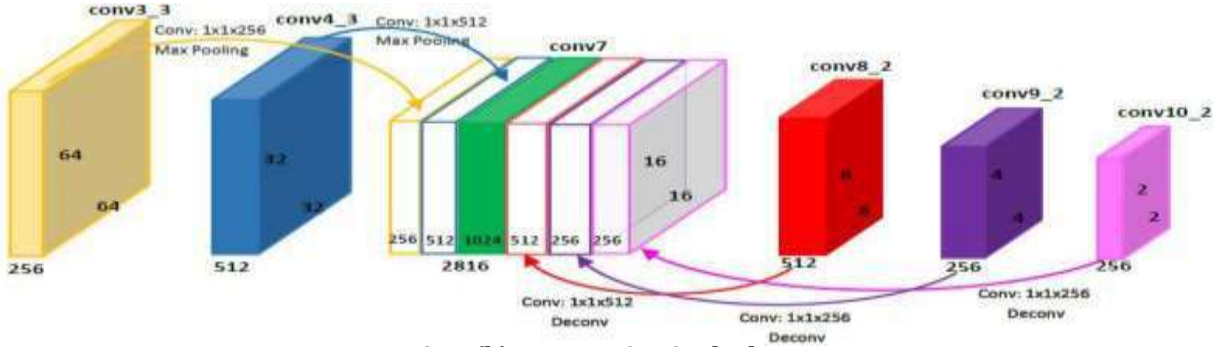

Fig.7.(b) Aggregation in the layer conv7

concatenated to conv7 after going through a max pooling layer respectively. While all the left pointing arrows signify downward aggregation. A 1x1 convolution followed by ReLU and a de-convolution layer are performed for each layer (conv8 2, conv9 2, conv10 2) respectively to magnify the size of feature maps to 16x16. This aggregation method make each layer in the feature pyramid contains the information from both the lower layers and the upper layers, so we call it "Bidirectional Information Aggregation". Once the whole aggregation is done, each layer in the feature pyramid contains 2560 feature maps(the sum of 512, 1024, 512, 256 and 256). Therefore, classifier networks in different layers can share weights, resulting in a single classifier network with faster training speed and progressive generalization power. What's more, we also unify 6 default boxes in different layers with weight sharing.

ii) Multiple Symmetrical Feature Maps

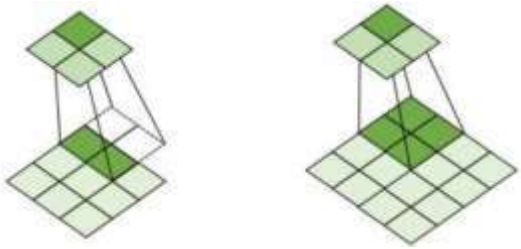

Fig.8.(a) Asymmetry (b) Symmetry

The front part of our model is Convolutional Neural Network (CNN) architecture and it performs down sampling by max pooling (kernel:2x2, stride:2). If the feature maps are asymmetrical, the

receptive field center of each point in output feature map does not coincide with the center of each cell, like Fig.8(a), resulting in some text, especially small-scale text, may be learned as background. On the contrary, if the CNN part of our model extracts features by performing max pooling in symmetrical feature maps as shown in Fig.8(b), our network can learn more local details, resulting in a more accurate text detector. Therefore, we train our network at 256x256 and use multiple rescaled versions of the input image for detection, including (width x height) 256x256, 512x512, 1024x256, 1024x512, and 1024x1024.

iii) Automatically Generated Aspect Ratios for Default Boxes

The aspect ratios for default boxes of the Text Boxes model are hand picked, it is not easy for the network to learn to adjust the boxes appropriately in a way, leading to imprecise predicting boxes. Instead of choosing the aspect ratios by hand, we propose a statistical grouping method that operates on the training set bounding boxes to automatically find aspect ratios for default boxes. Our method consists of two steps. First, we calculate the aspect ratios of all the training set bounding boxes. And then all the aspect ratios are rounded to the nearest positive integers. If the values of some aspect ratios are set to 0 after rounded, we reset them to 0.5. Next we sort all the rounded aspect ratios in an ascending order and count the frequency number of each aspect ratio.

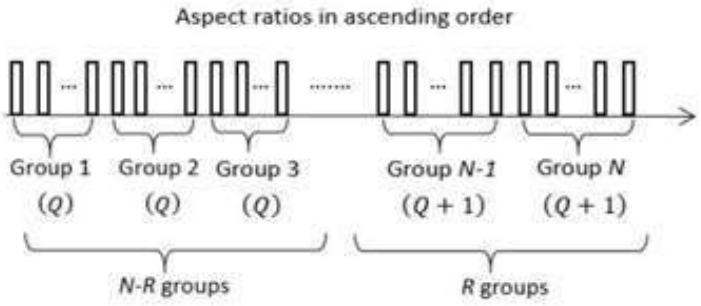

Fig.9. Illustration of dividing the sorted aspect ratios into N groups

iv) Unidirectional Information Aggregation

To demonstrate how efficient BIA model is, we present a comparison model inspired by [6], called unidirectional Information Aggregation Architecture (UIA). UIA simply uses convolution layer and max pooling layer to perform upward aggregation introduced in subsection A, aggregating feature maps from the layers in feature pyramid lower than current layer to current layer, and the other components of UIA are the same as BIA.

4. Enhanced Multi-channels MSER

The improvement of this paper has three parts: (a) The extraction of MSER is enhanced, so more challenging text regions can be detected. (b) Two new scene text features, local contrast and boundary key points are introduced, and the MSER regions are classified by trained SVM [7] with a RBF kernel [8]. (c) A fast grouping of the text regions is realized by the two-layer algorithm (get the initial text lines in the vertical direction and group the word region in the horizontal direction), and the time complexity is reduced from $O(n2)$ [9,10,11] to $O(n\log_2 n)$. The overall algorithm flowchart is shown in Fig. 10.

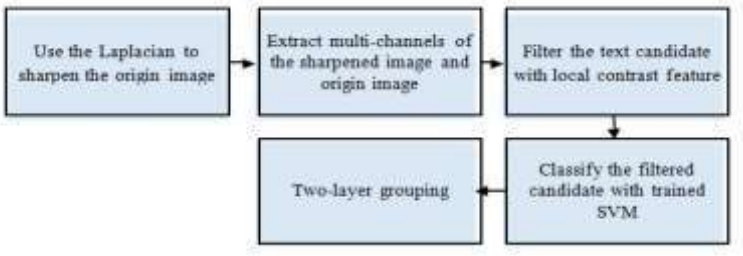

Fig.10. Flowchart

i) Enhanced Multi-channels MSER

The original MSER algorithm is only for grey images, and color information is not taken into account, but there is usually a significant difference between text and non-text regions in aspect of color. Neumann et al. [12] used the multi-channels processing to exploit the color information for better extracting of text MSER. Based on the multi-channels MSER, we first apply the Gaussian blur to the original image (Gaussian blur before sharpening can effectively reduce the influence of noise), then use the Laplacian template to preprocess the image and then the sharpened image is obtained according to Gaussian blur,

which enhances the contrast between text regions and their background, so the proposed method can better extract the text in a complex background.

ii) Scene Text Features

Local contrast *lc*. It is obviously that text which may be recognized by people must have some contrast against its background. In a local area, the non-text region extracted by MSER algorithm has a low contrast against the background.

There's a common feature among these non-text regions, that is the contrast between MSER region and its background is low, based on this feature, the local contrast feature is added to filter the non-text regions. For better utilization of color information, the R, G and B channels are extracted for every single text region and its neighbor or corresponding background.

iii) Two-Layer Text Grouping

Existing text grouping is usually done by calculating the degree of association between regions (like spatial position relations [9] and text line constraints [10]) and then iteratively clustering those regions according to experimental thresholds. However, these methods need to consider the relationship between each two text regions or among every triple, so the time complexity is $O(n2)$ (n is the number of extracted regions). Based on these methods, this paper improves the text region grouping algorithm and reduces the time complexity to $O(n \log_2 n)$.

In this paper, the text grouping is divided into two stages. The first stage detects the initial text line in the vertical direction, as shown in Fig. 11(a). In the second stage, the region of a word is detected in the horizontal direction (see Fig. 11(b)).

(a) Vertical grouping (b) Horizontal grouping

Fig.11. Two-layer text grouping

5. Morphological Techniques

Proposed system strives toward Morphological methodologies like dilation, erosion etc. that aids automatic detection, segmentation and recognition of visual text entities in complex several images and thus resulting in optimal performance [13].

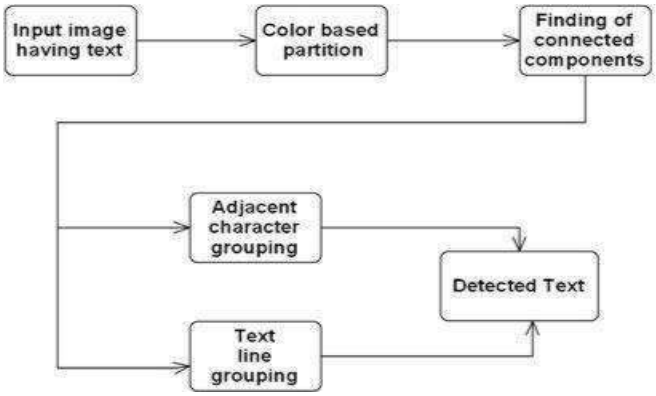

Fig.12. Proposed System

Above figure shows system architecture of Text string detection system from natural scenes. System works in four modules: Thresholding and basic morphological operations, finding of connected components, Adjacent character grouping and Text line grouping. First input image is converted into binary image. Then some morphological operations are applied on it. Then connected components are found out. After finding connected components height, width, centroid and area of each connected component is calculated. These parameters are required for the implementation of next modules. Then adjacent character grouping and text line grouping methods are implemented. Adjacent character grouping and Text line grouping are methods that can detect text string from natural scene images. Various modules of proposed system are described as follows:

i) Thresholding

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images. Following is an example of Thresholding image of a given color image.

Fig.13. Thresholding of given color image

ii) Dilation and Erosion

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. Following is an example of erosion and dilation when applied to above binary image.

Fig.14. An example of Dilation and Erosion

iii) Connected components labeling

Connected-component labeling is an algorithmic application of graph theory, where subsets of connected components are uniquely labeled based on a given heuristic. Connected-component labeling is used in computer vision to detect connected regions in binary digital images, although color images and data with higher dimensionality can also be processed.

iv)Adjacent Character Grouping

Text strings in natural scene images usually appear in alignment, namely, each text character in a text string must possess character siblings at adjacent positions. The structure features among sibling characters can be used to determine whether the connected components belong to text characters or unexpected noises.

Here, five constraints are defined to decide whether two connected components are siblings' of each other.

1. Considering the capital and lowercase characters, the height ratio falls between T1 and 1/T1.
2. Two adjacent characters should not be too far from each other despite the variations of width, so the distance between two connected components should not be greater than T2 times the width of the wider one.
3. For text strings aligned approximately horizontally, the difference between Y-coordinates of the connected component centroids should not be greater than T3 times the height of the higher one.
4. Two adjacent characters usually appear in the same font size, thus their area ratio should be greater than 1/T4 and less than T4.
5. If the connected components are obtained from gradient based partition, the color difference between them should be lower than a predefined threshold T5 because the characters in the same string have similar colors.

v) Text Line Grouping

In order to locate text strings with arbitrary orientations, we develop text line grouping method. To group together the connected components which correspond to text characters in the same string which is probably non horizontal, here we use centroid as the descriptor of each connected component.

We design an efficient algorithm to extract regions containing text strings. At first, we remove the centroids from the set M(M as a set of centroids) if areas of their corresponding connected components

are smaller than the predefined threshold T_s. Then, three points m_i, m_j, m_k are randomly selected from the set to form two line segments. Then we calculate the length difference, and incline angle difference between line segments m_i m_j and m_j m_k. Let Δd and $\Delta\theta$ be length difference and incline angle between those line segments, then if $0.5<=\Delta d<=2$ and $\Delta\theta<=\pi/12$, we construct a fitted line joining m_i, m_j, m_k. We continue same procedure till all connected components are covered. Finally in text line grouping detected string is represented by fitted line in red color.

III. COMPATIVE STUDY OF FIVE PAPERS

Here we studied 5 papers about their datasets, text detection methods, Feature extraction methods, text grouping methods used also which type of text will be detected. Following table shows comparative study of different five papers. So by comparing this papers we can judge accuracy of each paper and finds how each techniques will accurate for application.

TABLE 1 Comparison of Papers

Sr. No.	1	2	3	4	5
Text Detection Method	Fully Convolut-ional Neural Network	Convolutio-nal Neural Network	Bidirectional Information Aggregation	Enhanced multi-channels MSER	Morpho-Logical Techniques
Feature Extraction Method	Residual Network	Multilevel Connected Component	Convolutional Neural Network	Multi-channel MSER	Connected Component Analysis
Text Grouping Method	-	Graph-based, Euclidean distance - based, Clustering – based	Stastical grouping method	Two-layer text grouping	Adjacent character, Text-line grouping
Datasets	ICDAR 2015 COCO-Text	ICDAR 2013	Synth Text ICDAR 2011 ICDAR 2013	ICDAR 2011 ICDAR 2013	-
Accuracy	83.46% 56.39%	96.81%	81.1% 86.6%	71% 77%	70%
Type of Text detect	Horizont-al text	Multi-oriented, Multi-lingual text	Small- scale text	Horizont-al text, English words	Visual text

IV. APPLICATIONS

1. License/Container /name plate recognition
2. Tourist understanding native language
3. Easy to recognize road signs scripts
4. Automated text removal system
5. Data mining
6. Advertisement billboards
7. Providing text accessibility for visually impaired people

V. CONCLUSION

We studied various methods to detect the text from scene images like Convolutional neural network, Enhanced multi-channels MSER, Morphological techniques, Bidirectional information aggregation, etc. Learned the advantages and disadvantages of various methods for text detection. Text detection using Convolutional neural network overcomes all the limitations of earlier techniques used for detecting the text. Upon comparison of various feature extraction methods, it is concluded that Multi-level connected component is efficient one. Text detection via Convolutional neural network gives better accuracy than other methods.

REFERENCES

1. *Shaohui Ruan, Junguo Lu, Fengming Xie, Zhongxiao Jin, "A novel method for fast arbitrary-oriented scene text detection", The 30th Chinese Control and Decision Conference -2018.*
2. *K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition", Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.*
3. *Wafa Khlif, Nibal Nayef, Jean-Marc Ogier, Adel Alimi, "Learning text Component Features via convolutional neural networks for scene text detection",13th IAPR - 2018.*
4. *M. Liao, B. Shi, X. Bai, X. Wang, "Textboxes: A fast text detector with a single deep neural network."*

in AAAI, 2017.

5. *W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," inEuropean conference on computer vision. Springer, 2016.*
6. *X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: an efficient and accurate scene text detector," CoRR, vol. abs/1704.03155, 2017.*
7. *Nosov AV. "An introduction to support vector machines[M]". China Machine Press, 2005.*
8. *Muller KR, Mika S, Ratsch G, et al. "An introduction to kernel-based learning algorithms[J]". IEEE Transactions on Neural Networks, 2001.*
9. *Neumann L, Matas J. "Real-time scene text localization and recognition[C]"// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2012:3538-3545.*
10. *Liu J, Su H, Yi Y, et al. "Robust text detection via multi-degree of sharpening and blurring[J]". Signal Processing, 2016, 124(C):259-265.*
11. *Neumann L, Matas J. "Efficient Scene text localization and recognition with local character refinement[C]"// International Conference on Document Analysis and Recognition. IEEE, 2015:746-750.*
12. *Neumann L, Matas J. "A method for text localization and recognition in real-world images[C]"//Asian Conference on Computer Vision. Springer Berlin Heidelberg, 2010: 770-783.*
13. *V.V Rampurkar, Sahil K. Shah, G.J. Chhajed S. K. Biswash "An Approach towards Text Detection from Complex Images Using Morphological Techniques",Second International Conference on Inventive Systems and Control (ICISC 2018).*