ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
Your ulti-MATE Research Paper !!!

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 11, March 2017

NEURAL LANGUAGE TEXT DETECTION USING MAXIMALLY STABLE EXTREMAL REGIONS

S.Deepa[1], M.Arulprakash M.Tech.,[2]

[1]*Department of Computer Science, Sri Subramanya College of Engineering and Technology, Palani*

[2]*Department of Computer Science, Sri Subramanya College of Engineering and Technology, Palani*

[1]*deepaselvakumar17@gmail.com*

Abstract : *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 (patch), where the cluttered background information may be the deep representation. This leads to less discriminative power and poorer robustness. we present a new system for scene text detection by proposing a novel text-attentional convolutional neural network (Text-CNN) that particularly focuses on extracting text-related regions and features from the image components. We develop a new learning mechanism to train the Text-CNN with multi-level and rich supervised information, including text region mask, character label, and binary text/non-text information. The rich supervision information enables the Text-CNN with a strong capability for discriminating ambiguous texts, and also increases its robustness against complicated background components. The training process is formulated as a multi-task learning problem, where low-level supervised information greatly facilitates the main task of text/non-text* classification. *In addition, a powerful low-level detector called contrast-enhancement maximally stable extremal regions (MSERs) is developed, which extends the widely used MSERs by enhancing intensity contrast between text patterns and background. This allows it to detect highly challenging text patterns, resulting in a higher recall. Our approach achieved promising on ICDAR 2013 dataset improving the state-of-the-art result.*

Keywords :*Maximally Stable Extremal Regions, text de-tector,convolutional neural networks, multi-level supervised information, multi-task learning.*

1. Introduction

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, as substantiated by recent literature [3], [4], [1], [5], where the leading performance on the detection sub task is of 80% F-measure on the ICDAR 2011 [4], and current result of unconstrained end-to-end recognition is only 57% accuracy on the challenging

SVT dataset [3]. In this work, we focus on the detection task that aims to correctly localize exact. Many previous methods focus on developing hand-crafted features based on a number of heuristic image properties (e.g.intensity variance, sharp information or spatial location) todiscriminate text and non-text components [7], [6]. These low-level features inherently limit their generality on highly challenging text components. They also reduce the robustness against text-like components, which often have similar low-level properties as the true texts, such as bricks, windows or leaves. These facts pose main challenge of current scene text detection systems, and severely harm their performance in both precision and recall. Recently, a number of deep models have been developed for text component filtering/classification [1], [8], [2], orword/character recognition [9], [8], [10], [2], by leveraging advances of deep image representation. To realize this capability and incorporate additional supervised knowledge, we propose a novel Text-Attentional Convolutional Neural Network (Text-CNN) for text component filtering. The Text-CNN is incorporated with a newly-developed Contrast-Enhanced MSERs (CE-MSERs) to form our text detection system.

1. System Analysis

1.1 Existing System

Currently, a large number and variety of applications have been extensively developed for distributed systems linking computers and other computing devices together in a seamless and transparent way. In distributed systems, various nodes act autonomously and cooperate with each other, which can achieve the purposes of resource sharing, openness, concurrency, scalability, fault-tolerance, and transparency [16][17]. scene text detection can be roughly categorized into two groups, sliding-window and connected component based methods [11]. The sliding-window methods detect text information by moving a multi-scale sub-window through all possible locations in an image They separate text and non-text information at pixel-level by running a fast low-level detector. The retained pixels with similar properties are then grouped together to construct possible text components. The ERs/MSERs [31], [32] and Stroke Width Transform (SWT) [25] are two representative methods in this group. Extending from the original SWT, Huang et. al. [22] proposed a Stroke Feature Transform (SFT), by incorporating important color cues of text patterns for pixel tracking. In [1],[5], [23], [26], the MSERs detector has demonstrated strong capability for detecting challenging text patterns, yielding a good recall in component detection. In [24], Character rness was proposed by incorporating three novel cues The connected component based methods exhibit great advantage in speed by fast tracking text pixels in one pass computation, with complexity of O(N). However, low-level nature of these methods largely limits their capability, making them poorer robust and discriminative. Therefore, a sophisti-cated post-processing method is often required to deal with large amount of generated components, which causes main challenge of this group of methods. A powerful text/non-text classifier or component filter is critical to success of both the sliding-window and connected component based methods. Huge efforts have been devoted to developing an efficient hand-crafted feature that could correctly capture discriminative characteristics of text. Chen and Yuille [13] proposed a sliding-window method by using the Adaboost classifiers trained on a number of low-level statistical features. proposed a symmetry-based text line detector that computes both symmetry and appearance features based on heuristic image properties. For the connected component approaches, in [25], a number of heuristic rules were designed to filter out the non-text components generated by the SWT detector. Extending from this framework, a learning based approach building on Random Forest [36] was developed by using manually-designed histogram features computed from various low-level image properties [7]. To eliminate the heuristic procedures,Huang et. al. [22] proposed two powerful text/non-text classifiers, named Text Covariance Descriptors (TCDs), which compute both heuristic properties and statistical characteristics of text stokes by using covariance descriptor [37]. However, low-level nature of these manually-designed features largely limit their performance, making them hard to discriminate challenging components accurately, such as the ambiguous text patterns and complicated background outliers. Deep CNN models are powerful for image representation by computing meaningful high-level deep features [38],[39], [40], [41]. Traditional CNN network (such as the well-known LeNet) has been successfully applied to text/document community for digit and hand-written character recognition [42], [43]. Recently, the advances of deep CNN A CNN model was employed to filter out non-text components, which are generated by a MSERs detector in [1] or the Edgebox in [3],while Wang et al. [2] and Jaderberg et. al. [8] applied a deep CNN model in the sliding-window fashion for text detection.Gupta et al. developed a new Fully-Convolutional Regression Network (FCRN) that jointly performs text detection and bounding-box regression [44]. Though these deep models have greatly advanced previous manually-designed features, they mostly compute general image features globally from a whole image component/patch mixing with cluttered background, where background information may be computed dominantly in feature learning processing. They only use relatively simple text/non-text information for training, which is significantly insufficient to learning a discriminative representation.

1.2 Proposed System

The proposed system mainly includes two parts: AText-Attentional Convolutional Neural Network (Text-CNN)for text component filtering/classfication, and a Contrast-Enhanced MSERs (CE-MSERs) detector for generating component candidates. we propose the Text-CNN by training it with multi-level highly supervised text information including text region segmentation, character label and binary text/non-text information. These additional supervised information would ‘tell’ our model with more specific features of the text, from low-level region segmentation to high-level binary classification. This allows our model to sequentially understand *where*, *what* and *whether* is the character, which is of great importance for making a reliable decision .Our Text-CNN is also different from previous MTL model for facial landmark detection. An important property of our model is that the three tasks have strong hierarchical structure, and can be optimized sequentially from the low-level region regression to the high-level binary classification. Such hierarchical structure inherently follows basic procedure of our people to identify a text/non-text component. People should first be able to segment a text region arcuately from cluttered background.Then people could make a more confident decision if they recognize the label of a character. Therefore, a reliable high-level decision is strongly built on robust learning of the multilevel prior knowledge. the MSERs algorithm has strong ability to detect many challenging text components, by considering of them as a 'stable' extremal region the text components generated by the MSERs are easily distorted by various complicated background affects, leading to numerous incorrect detections, such as connecting the true texts to the background components, or separating a single character into multiple components. This makes it difficult to identify the true texts in the subsequent filtering step. Second, some text components are ambiguous with low-contrast or low-quality characters. They may not be defined as the 'stable' or extremal regions, and thus are discarded arbitrarily by the MSERs. multiple tasks, making it possible to learn more discriminative text features by using the additional low-level supervision. This greatly facilitates convergence of the main task for text/non-text classification is that the nodes face relatively high computational costs which may place heavy loads on large systems and make the task allocation processes difficult to control effectively.

3.System Design

3.1 Detect Candidate Text Regions Using MSER

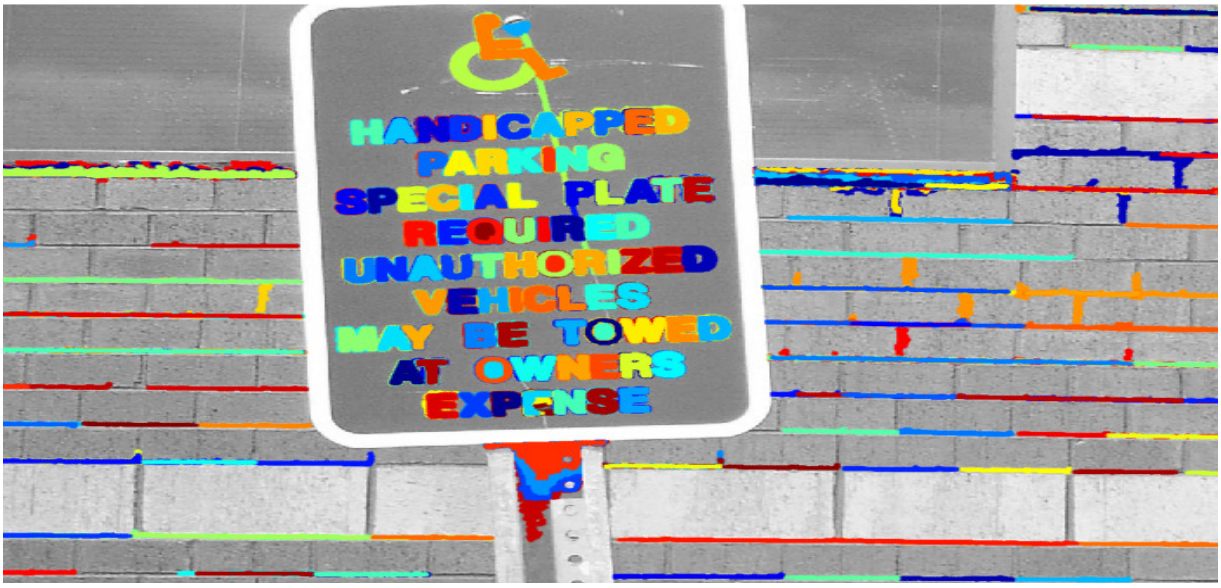
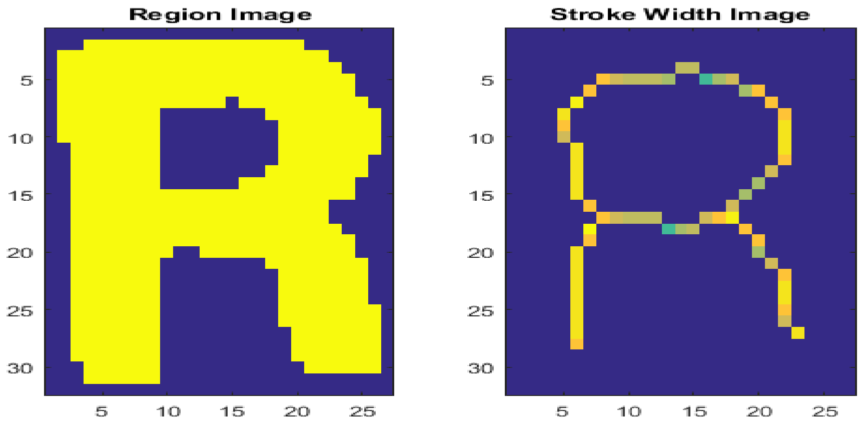


Fig 3.1 Detect Candidate Text Regions Using MSER

3.2. Remove Non-Text Regions Based On Stroke Width Variation



3.2 Remove Non-Text Regions Based On Stroke Width Variation

3.3 Merge Text Regions For Final Detection Result

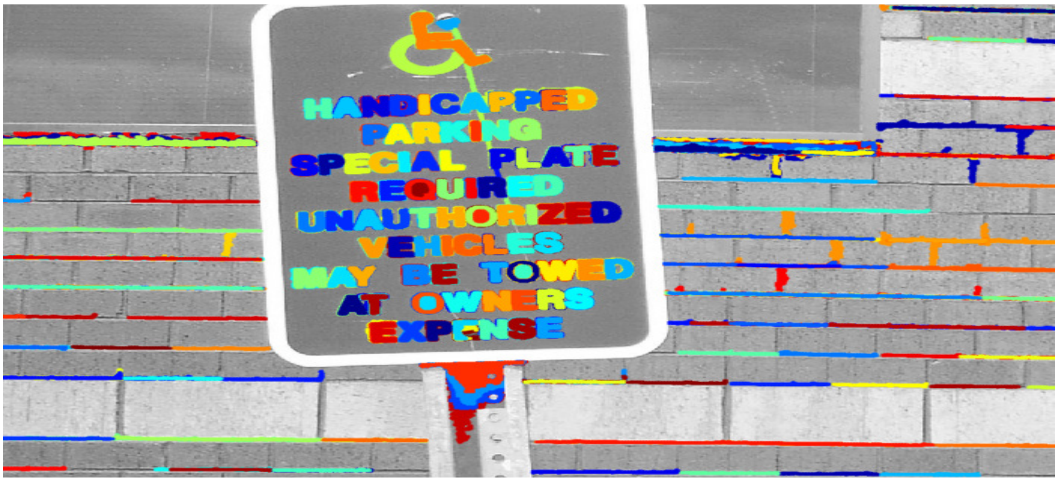
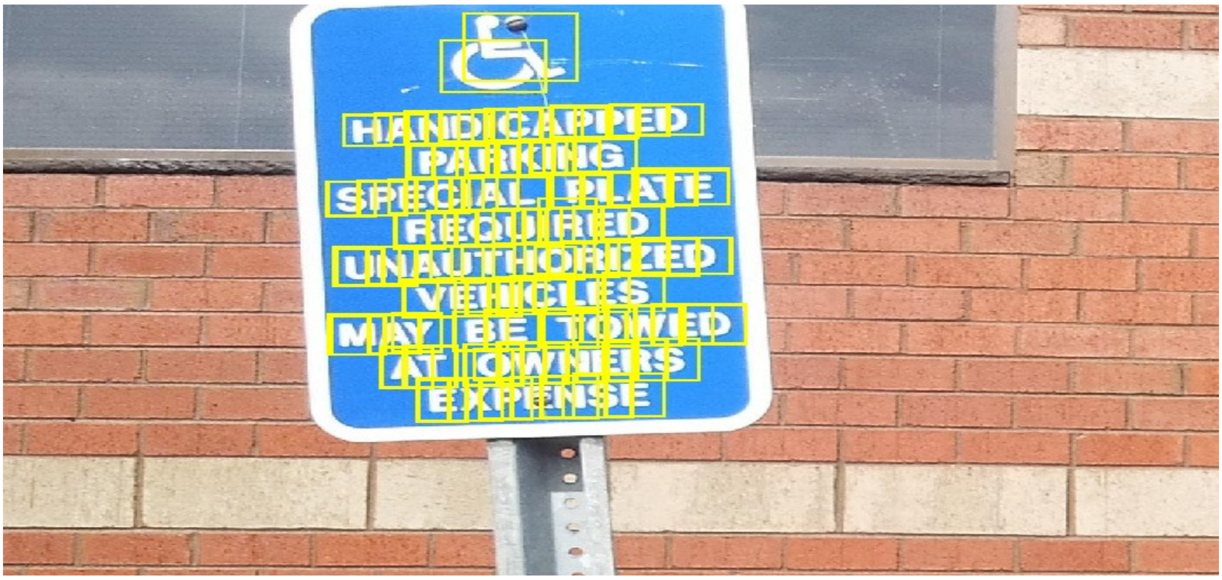


Fig 3.3 Merge Text Regions For Final Detection Result

3.4 Merge Text Regions For Final Detection Result



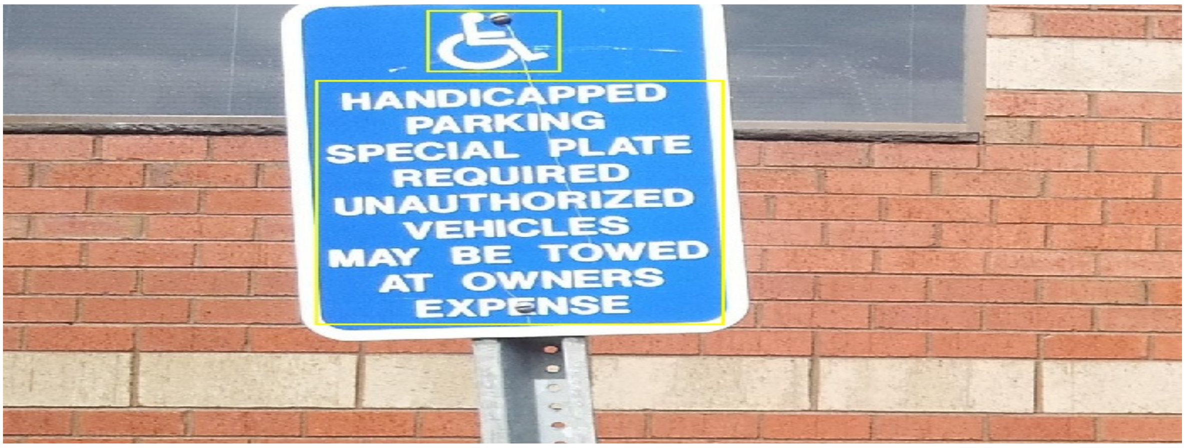


Fig 3.5 Merge Text Regions For Final Detection Result

4 SYSTEM REQUIREMENTS

4.1 Software Requirements

Operating System	: Windows 2008
Application	: MATLAB

4.2 Hardware Requirements

Processor	: Pentium IV 2.4 GHz
Hard Disk	: 320 GB
RAM	: 2GB

4.3 Introduction To Matlab

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using MATLAB, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

4.4 Working Formats In Matlab

However, in order to start working with an image, for example perform a wavelet transform on the image, we must convert it into a different format. This section explains four common formats.

4.5 Intensity Image (Gray Scale Image)

This is the equivalent to a "gray scale image" and this is the image we will mostly work with in this course. It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored. There are two ways to represent the number that represents the brightness of the pixel: The double class (or data type). This assigns a floating number ("a number with decimals") between 0 and 1 to each pixel. The value 0 corresponds to black and the value 1 corresponds to white. The other class is called uint8, which assigns an integer between 0 and 255 to represent the brightness of a pixel.

The value 0 corresponds to black and 255 to white. The class uint8 only requires roughly 1/8 of the storage compared to the class double. On the other hand, many mathematical functions can only be applied to the double class.

4.6 Indexed Image

This is a practical way of representing color images. (In this course we will mostly work with gray scale images but once you have learned how to work with a gray scale image you will also know the principle how to work with color images.) An indexed image stores an image as two matrices.

The first matrix has the same size as the image and one number for each pixel. The second matrix is called the color map and its size may be different from the image. The numbers in the first matrix is an instruction of what number to use in the color map matrix.

4.7 RGB Image

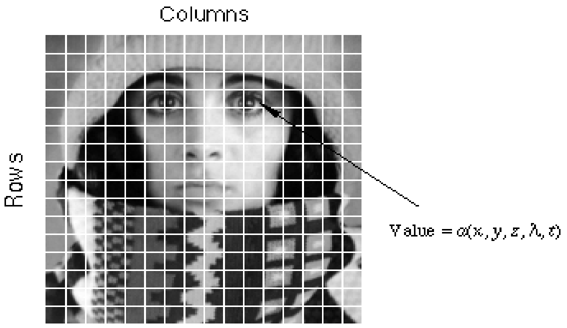
This is another format for color images. It represents an image with three matrices of sizes matching the image format. Each matrix corresponds to one of the colors red, green or blue and gives an instruction of how much of each of these colors a certain pixel should use.

4.8 Multiframe Image

In some applications we want to study a sequence of images. This is very common in biological and medical imaging where you might study a sequence of slices of a cell. For these cases, the Multiframe format is a convenient way of working with a sequence of images. In case you choose to work with biological imaging later on in this course, you may use this format.

Digital Image Definitions

A digital image $a[m, n]$ described in a 2D discrete space is derived from an analog image $a(x, y)$ in a 2D continuous space through a *sampling* process that is frequently referred to as digitization. The mathematics of that sampling process will be described in Section 5. For now we will look at some basic definitions associated with the digital image. The effect of digitization is shown in Figure 1.



The 2D continuous image $a(x, y)$ is divided into *Nrows* and *Mcolumns*. The intersection of a row and a column is termed a *pixel*. The value assigned to the integer coordinates $[m, n]$ with $\{m=0,1,2,...,M-1\}$ and $\{n=0,1,2,...,N-1\}$ is $a[m, n]$. In fact, in most cases $a(x, y)$--which we might consider to be the physical signal that impinges on the face of a 2D sensor--is actually a function of many variables including depth (z), color (), and time (t).

5. System Implementation

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

- The active user must be aware of the benefits of using the system
- Their confidence in the software built up
- Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

5.1 User Training

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important.

Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

5.2 Training on the Application Software

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

5.3 Operational Documentation

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

5.4 System Maintenance

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make adaptable to the changes in the system environment. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only thru proper system maintenance procedures, the system can be adapted to cope up with these changes. Software maintenance is of course, far more than “finding mistakes”.

5.4.1 Corrective Maintenance

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

5.4.2 Adaptive Maintenance

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore Adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

5.4.3 Perceptive Maintenance

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, Perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

5.4.4 Preventive Maintenance

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

6 Conclusion

We have presented a new system for scene text detection,by introducing a novel Text-CNN classifier and a newlydeveloped CE-MSERs detector. The Text-CNN is designed tocompute discriminative text features from an image component. It leverages highly-supervised text information, including text region mask, character class, and binary text/non-text information. We formulate the training of Text-CNN as a multi-task learning problem that effectively incorporates interactions of multi-level supervision. We show that the informative multi-level supervision are of particularly importance for learning a powerful Text-CNN which is able to robustly discriminate ambiguous text from complicated background. In addition, we improve current MSERs by developing a contrast enhancement mechanism that enhances region stability of text patterns. Extensive experimental results show that our system has achieved the state-of-the-art performance on a number of benchmarks.

7 References

[1] Cao.Z.Z, Kodialam.M and Lakshman.T.V. Joint Static and Dynamic Traffic Scheduling in Data Center Networks in Proceedings of IEEE INFOCOM 2014, pp.2445-2553.

[2] Erdman.A.G, Keefe.D.F, Schiestl.R, Grand Challenge: ApplyingRegulatory Science and Big Data to Improve Medical Device Innovation, IEEE Transactions on Biomedical Engineering, Vol.60,No.3, 2013, pp.700-706.

[3] Feilong Tang Member, IEEE, Laurence T. Yang.A Dynamical and Load-Balanced FlowScheduling Approach for Big Data Centers inClouds. IEEE Transaction on cloud computing 2016.

[4] Greene.K, TR10: Software-Defined Networking, MIT TechnologyReview, Retrieved Oct. 7, 2011.

[5] Gude.N, Koponen.T, Justin Pettit et al., NOX: Towards an Operating System for Networks, SIGCOMM Computer Communication. Rev.,Vol.38, July 2008, pp. 105-110.

[6] Han.X, Li.J, Yang et al.D., Efficient Skyline Computation on BigData, IEEE Transactions on Knowledge and Data Engineering,Vol.25, No.11, 2013, pp.2521-2535.

[7] Handigol.N, Seetharaman.S, Flajslik.M, McKeown.N, andJohari.R, Plug-n-Serve: Load-balancing web traffic using Open-Flow, Demo at ACM SIGCOMM, Aug. 2009.

[8] Lu.J, Li.D, Bias Correction in Small Sample from Big Data, IEEETransactions on Knowledge and Data Engineering, Vol.25, No.11,2013, pp.2658-2663.

[9] McKeown.N, Anderson.T, Balakrishnan.H, Parulkar.G, Peterson.L, Rexford.J, Shenker.S, and Turner.J, Open Flow Enabling Innovation in Campus Networks, SIGCOMM. Rev., 2008.

[10] Pathan.R.M and Jonsson.J. Load regulating algorithm for static-priority task scheduling on multiprocessors. Proc. of 2010 IEEE International Symposium on Parallel and Distributed Processing (IPDPS), 2010, pp.

[11] Sharma.S, Singh.S, and Sharma.J, Performance Analysis of LoadBalancing Algorithms for cluster of Video on Demand Servers, in Proceedings of IACC, 2011.

[12] Schlansker.M, Turner.Y, Tourrilhes.J, and Karp.A, Ensemble Routing for Datacenter Networks, In ACM ANCS, La Jolla, CA, 2010.

[13] Tsai.C.-W, W.-C. Huang, M.-Hsiuetal.A Hyper-HeuristicScheduling Algorithm for Cloud. IEEE Transactions on

CloudComputing, Vol.2, No.2, pp.236-249, 2014.

[14] Wang.R, Butnariu.D, and Rexford.J, Open Flow Based ServerLoad Balancing Gone Wild, in: Proceedings of Workshop on Hot-ICE, Mar. 2011.

[15] Zhang.F, Cao.J, Hwangetal.K. Adaptive Workflow Schedulingon Cloud Computing Platforms with Iterative Ordinal Optimization. IEEE Transactions on Cloud Computing, Vol.PP, No.99,2014.