Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

Identifying Text From An Image By Using Binarization and Normalization

Ninumol S[1], Riyamol Sidhic[2]

[1,2]*Department of Computer Science, IIET,Nellikuzhi*

Abstract— The growth in digital camera usage combined with a worldly abundance of text has translated to a rich new era for a classic problem of pattern recognition, reading. While traditional document processing often faces challenges such as unusual fonts, noise, and unconstrained lexicons, scene text reading amplifies these challenges and introduces new ones such as motion blur, curved layouts, perspective projection, and occlusion among others. Reading scene text is a complex problem involving many details that must be handled effectively for robust, accurate results. In this work, describe and evaluate a reading system that combines several pieces, using probabilistic methods for coarsely binarizing a given text region, identifying baselines, and jointly performing word and character segmentation during the recognition process. By using scene context to recognize several words together in a line of text, this system gives state-of-the-art performance on three difficult benchmark data sets. Making a robust, accurate scene text reader is a complex problem involving many details that must be handled effectively. In this work, describe and evaluate a reading system that integrates many of these pieces: a simple region-grouping algorithm and probabilistic models for coarsely binarizing a given text region, identifying baselines and jointly performing word and character segmentation during the recognition process.

Keywords—pattern recognition; binarization; segmentation; region grouping; baselines

I. INTRODUCTION

Recognizing scene text is a challenging problem, even more so than the recognition of scanned documents. Given the rapid growth of camera-based applications readily available on mobile phones, understanding scene text is more important than ever. While humans have been writing for nearly six millennia, machines have made great strides in reading over the last century. In traditional optical character recognition (OCR), a printed document is scanned to an image and translated into some machine readable text format. Although researchers have made significant progress, machines have yet to match human reading performance. Now the widespread availability of consumer cameras and the worldly abundance of text have created a new era for machine reading. Scene text recognition (STR) involves finding and reading ambient text in the environment captured by a camera. While traditional document processing often faces challenges such as imaging defects, novel or rare fonts, noise, and unconstrained lexicons, STR amplifies these challenges and introduces new ones such as motion blur, curved layouts, perspective projection, and occlusion among others.

Making a robust, accurate scene text reader is a complex problem involving many details that must be handled effectively. In this work, describe and evaluate a reading system that integrates many of these pieces: a simple region-grouping algorithm and probabilistic models for coarsely binarizing a given text region, identifying baselines, and jointly performing word and character segmentation during the recognition process. By recognizing several words together in a line of text[2], our system gives state-of-the-art performance on three difficult benchmark data sets.

Work makes several contributions to scene text reading. Unlike documents, scene text lines may have just a few words. Still, utilizing collinear text words facilitates improved appearance normalization, an important contribution that significantly improves accuracy. Another contribution is the use of the discriminative semi-Markov model, which integrates learning from several information sources such as character appearance, geometry, and language[7]. Finally, we explicitly incorporate word segmentation for STR, a task made challenging by highly irregular and unconstrained character spacing. Because nearly all modules of the system are probabilistic, we pass forward multiple hypotheses to subsequent modules, delaying final decisions until top-down information has been incorporated.

On one extreme Optical Character Recognition is considered as one of the most successful applications of computer vision, and on the other hand text images taken from street scenes, video sequences, text-captcha, and born-digital (the web and email) images are extremely challenging to recognize. The computer vision community has shown a huge interest in this problem of text understanding in recent years. It involves various sub-problems such as text detection, isolated character recognition, word recognition. Text detection accuracies have significantly improved, but they were less successful in recognizing words. However, the availability of lists is not always possible, or the word may not be part of the given list.

II. RELATED WORK

Noise, unusual fonts and typesetting, and low resolution are endemic in scene text captured by portable camera so that few assumptions can be made about the input. This frequently makes scene text recognition more difficult than many document recognition problems. Making a robust, accurate scene text reader is a complex problem involving many details that must be handled effectively. In this work, describe and evaluate a reading system that integrates many of these pieces: a simple region-grouping algorithm and probabilistic models for coarsely binarizing a given text region , identifying baselines and jointly performing word and character segmentation during the recognition process.

Many authors have studied the primary task of finding text in images. The 2003 ICDAR Robust reading competition did much to spur interest in this area. Although other STR work appeared and a follow-up 2005 contest was organized , 6 years passed before anyone would benchmark the open-vocabulary word recognition task of this difficult data set. In 2011, the data set was revised to reduce (though not eliminate) annotation errors, give tighter bounding boxes, and increase certain forms of variability; the word recognition contest received just three entries, the best performing with a 59 percent word error rate, which this work reduces to 42 percent.

Wang and Belongie introduced a new task for their street view text (SVT) data set[7] . Words in each SVT image are to be recognized from a small lexicon of about 50 words. Though the images are more intrinsically challenging due to resolution, mosaicking errors, and perspective, the word spotting task is much simpler than the open-vocabulary ICDAR benchmark. Our prior work assumed character and word boundaries for recognition . To find word boundaries, Neumann and Matas[4] , use heuristics of gaps on binarized character regions, but it is weak image gradients. Before distances between characters can be measured, characters must be binarized correctly, a significant challenge when noise and low resolution cause both broken and touching characters. Even if characters could be binarized and isolated, word boundaries are not easily predicted because scene text is often less constrained by character kerning and tracking conventions. For example, the intra-word gaps in are larger than the interword space in.

To resolve these ambiguities, we integrate word and character segmentation with character recognition , giving bottom-up and top-down information flows influence so that low-level segmentation commitments are not made too early and high-level recognition processes need not examine unsupported hypotheses. Several others have since applied similar weighted finite state transducer (FST) variants to the character segmentation and recognition problem. Convolutional neural network to predict character boundaries, building a recognition graph with segmentation and letter scores from a convolutional neural network as edge weights. With no language model included, only one letter hypothesis per edge is needed to find the maximal score. Elagouni[3] follow by incorporating a trigram language model. Using a weighted FST, with lexicon-constrained paths. Such lexicon constraints are suited to the word spotting task, where Wang and Belongie use a pictorial structures model, a type of weighted FST with quadratic edge scores. An intermediate approach by using positional bigram statistics, an approach whose power devolves as the lexicon grows and is only applicable when word segmentations are assumed.

III. PROPOSED WORK

While humans have been writing for nearly six millennia, machines have made great strides in reading over the last century. In traditional optical character recognition (OCR), a printed document is scanned to an image and translated into some machine readable text format. Although researchers have made significant progress, machines have yet to match human reading performance. Now the widespread availability of consumer cameras and the worldly abundance of text have created a new era for machine reading. Scene text recognition (STR) involves finding and reading ambient text in the environment captured by a camera[1]. While traditional document processing often faces challenges such as imaging defects, novel or rare fonts, noise, and unconstrained lexicons, STR amplifies these challenges and introduces new ones such as motion blur, curved layouts, perspective projection, and occlusion among others. This work makes several contributions to scene text reading. Unlike documents, scene text lines may have just a few words. Still, utilizing collinear text words facilitates improved appearance normalization, an important contribution that significantly improves accuracy[1].

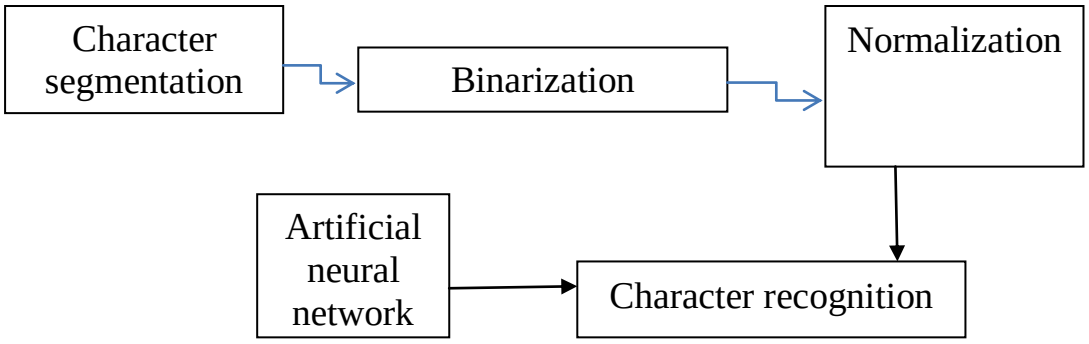

***Figure 1.* Architecture of proposed method**

A. Character Segmentation

Images are input to the commercial OCR software. The segmentation (also called binarization) results influence the final recognition results directly. Text segmentation is to label the text pixels and non-text pixels. Although traditional method is to find optimal global or local threshold to binarize the gray image, thresholding has a problem with handling complex background.

Clustering is an effective and popular method in image segmentation. Here, color image is used rather than gray image because color images have three channels and each pixel is described by three dimensions. Our method of clustering based on color depends on the assumption that text

pixels are in consistent color. As long as the difference is not very large, text pixels tend to be clustered together because the difference between text and non-text pixels is usually larger. Exceptions exist in practical but it is very rare that there is great color inconsistency in text.

K-means is adopted in our method because of its simplicity and efficiency. The k-means algorithm partitions a set of elements into k clusters so that the intra cluster similarity is high but the intercluser similarity is low. In text segmentation, RGB values of pixels are input to k-means clustering algorithm and the output is the cluster label for each pixel. Here we get two important issues:

1. How to decide K, the number of clusters?
2. Given that k clusters have been classified, how to decide which cluster or clusters to be text pixels or background pixels?

Researches about k-means clustering for gray scale text segmentation[10] have been done. They take k as 2, 3 and 4 and choose the best OCR result. Intuitively, 2 is a good choice for the first issue. But it maybe more complicated in practical considering the contours around the text and background complexity. Thus done several experiments to find out the most appropriate value for k. In this paper, we only consider situations that when k is less than 5. This is because that if more than 4 clusters have been generated, it is complicated to decide which clusters are text clusters and which are not.

Theoretically[9], text can be of any color, that is to say, can be any cluster in the k-means clustering result. But in practical, to highlight the texts, they are usually set to be the lightest or the darkest. Consequently, the cluster with the highest grayscale or the lowest gray scale is most probably the text cluster. Many works called color polarity classification have been done to differentiate these two cases. In this paper, only consider normal text (text is brighter and background is darker) for simplicity. So the cluster with high gray scale centre is considered text in my experiment and it is proved feasible. So when k is 2, the cluster with higher gray scale cluster-centre is labelled as text and the other one is non-text. When k is 3, the clusters with highest gray scale cluster-centre or the first two highest gray scale cluster-centres are considered text. It is deduced by analogy when k is 4.

B. Binarization

Several candidate segmentations given by the regression mixtures to yield a final binarization. Fit one mixture of K ¼ 3 components and another with K ¼ 4. In many cases, the components in the K ¼ 3 model correspond to text, background, and mixed pixels. The primary question is how to identify which component(s) correspond to the text. This as a probabilistic classification problem, where a logistic regression scores several features of all binary images induced by the mixture model[9]. The candidate binarization with the highest probability of being text is forwarded to the subsequent stage (guide line fitting). Next, describe how candidate binary images are generated[5], the features used to make the classification, and how the classifier is trained.

- Connected Component and Binary Image Features

For each mixture component, we create a binary image where a pixel is “on” if the given component is the most probable under the model . I also consider the union of
pairs of binary images, which allows us to handle strong shadows, dual-color characters and so on. Because the background is sometimes segmented more uniformly than the text, we include the

binary complement of each candidate image for consideration. In total, there are six unique images for K ¼ 3, plus 20 more candidate binarizations for K ¼ 4.

Use two classes of features to represent each binary image: statistics of connected component features and global statistics of the binary image. Measure 14 features for each connected component: normalized area, hole/ area ratio, compactness, solidity (three features used by Neumann and Matas), eccentricity, normalized major and minor axis lengths, aspect ratio, and Hu's seven invariant moments. Because the number of components varies, here represent the distribution of each component feature with a fixed set of 11 feature statistics: mean, variance, skew, kurtosis, quantiles at 2.5, 25, 50, 75, and 97.5 percent, and sums of the absolute deviation from the mean and median.

Compute five global features of the binary image: normalized total area, fraction of "on" border pixels, Euler number, and normalized horizontal and vertical range[2]. Finally, we compute the same 11 statistics described above on five functions of the pixels: stroke width (distance from every skeleton point to the nearest "off" pixel), normalized row and column coordinates, and normalized marginal (e.g., column and row sums divided by height and width). All features together form a 225D representation of a binary image.

- Text Image Classifier

For training data, we labeled several word image segmentations from the K ¼ 3 regression mixture model . Additional positive instances came from automatically binarized images of words from the ICDAR 2003 scenes training data provided, use a logistic regression classifier to score each of the 26 candidate images given by the K ¼ 3 and K ¼ 4 regression mixture models, identifying the image with the highest probability of being text. If none of these images yields a positive classification, we run another mixture with K ¼ 5. Considering all components and pairs of components along with the complements yields 30 candidate binarizations[10] for the K ¼ 5 model. The binarization with the highest classification score (now among all 56 candidates) is
finally chosen as the binarized text image. Trying the more restricted models first prevents a spurious but high-scoring K ¼ 5 over segmentation from being chosen when a simpler model suffices. On the SVT test data, the percentage of binarizations drawn from the K ¼ 3; 4; 5 models are 74, 20, and 6 percent, respectively. Of those using the K ¼ 5 model, 70 percent are still classified as non text.

C. Normalization

Text in the wild is captured from arbitrary views and exhibits rotation, perspective projection from nonplanar surfaces, and even intrinsically nonlinear baselines (the conventional term for the curve on which the letters rest, whether it is truly linear or not). While it is possible to train a character classifier with examples of these variations, the result is often inherently less discriminative unless invariant features are used or the variations are explicitly modeled rather than being treated as noise. To sharpen performance of our character recognition system by using the binarized image to normalize the text to a rectilinear pose, minimizing the effect of viewpoint variations.

In formulation, only the binary image and must infer which points correspond to the extreme that the guides intersect. Unfortunately, these extreme are defined in a coordinate system relative to the guide curves themselves. Consider the character "o." Under any rotation the image coordinate system gives an arbitrary point on the boundary as a lower extreme. Forcing a guide through this arbitrary point makes the character rest below the baseline. The converse happens for upper extreme. Prior work is prone to these biases for rotated text. Because we need the guides to infer the extreme and the extreme to infer the guides[8], we take an approximate, iterative refinement approach to discovering both. First find the least squares quadratic fit to all the points (pixels) in the binarized

image. This fit tends to give a good approximation of the text guide shape because most of the pixels are text.

However, a simple least-squares regression is not robust, and nontext outliers can significantly bias the results. Therefore discard any connected components this initial curve does not touch and perform another regression on what remains. This secondary fit is usually much closer to the correct guide shape (assuming top and bottom are similar to one another)[6], tends to traverse the characters, and gives us a reasonable, approximately correct coordinate system for finding extrema to which lower and upper guides may be fit.

To find candidate extrema, consider a series of local coordinate systems aligned with the curve's normal and tangent. At each column, we search along the normal of the secondary fit to find the farthest point from the curve on each connected component intersecting that normal. Then retain only those that are extreme points of the binary image in this local coordinate system. To assess which points are extrema, consider a neighborhood of 5 pixels to either side (in the tangent orientation) for comparison[6]. While this neighborhood is not scale invariant, results are stable over a wide range of scales.

D. Character recognition

In machine learning and cognitive science, artificial neural networks (ANNs) are a family of statistical learning models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning.

A neural network for text recognition from an image is defined by a set of input neurons which may be activated by the pixels of an input image. After being weighted and transformed by a function (determined by the network's designer), the activations of these neurons are then passed on to other neurons. This process is repeated until finally, an output neuron is activated. This determines which character was read.

Like other machine learning methods - systems that learn from data - neural networks have been used to solve a wide variety of tasks that are hard to solve using ordinary rule-based programming, including computer vision and speech recognition.

There is no single formal definition of what an artificial neural network is. However, a class of statistical models may commonly be called "Neural" if it possesses the following characteristics: contains sets of adaptive weights, i.e. numerical parameters that are tuned by a learning algorithm[6], and capability of approximating non-linear functions of their inputs. The adaptive weights can be thought of as connection strengths between neurons, which are activated during training and prediction.

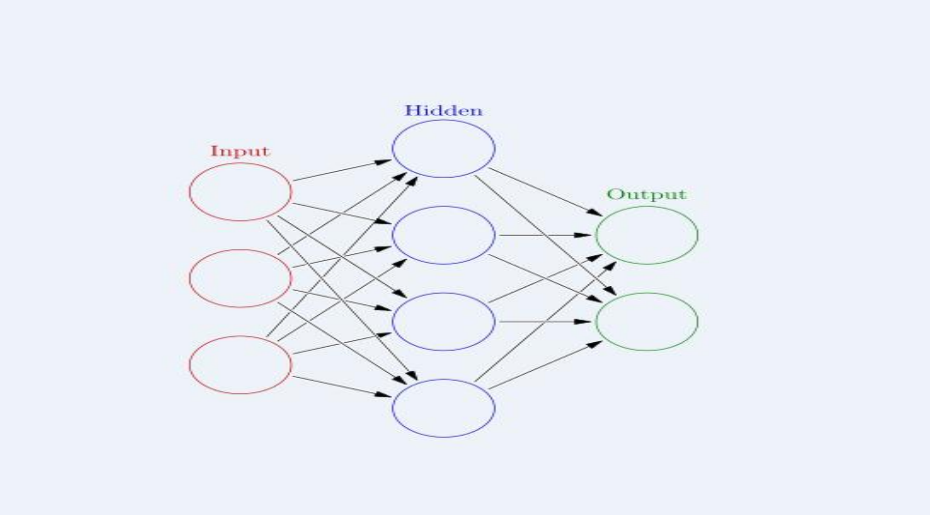

Figure 2. Database contexts

Neural networks are similar to biological neural networks in the performing of functions collectively and in parallel by the units, rather than there being a clear delineation of subtasks to which individual units are assigned. The term "neural network" usually refers to models employed in statistics, cognitive psychology and artificial intelligence. Neural network models which command the central nervous system and the rest of the brain are part of theoretical neuroscience and computational neuroscience.

Algorithm: Scene text reading
Input: Camera image
Output: Texts in the image

- Providing a scene
- Performing foreground and back ground separation using novel color transformation
- Finding the k- mean of coloring index
- Choosing the minimum value of k-mean.
- Select it as foreground.
- Converting the separated image into binarised image
- Finding the absolute deviation of attributes.
- Calculate classification score.
- Choosing the highest value .
- Performing normalization.
- Finding the intersecting point of normal with connected components.
- Consider the external point of binary image
- Providing it to an artificial neural network for identifying which character is.
- Output the text from an image

In modern software implementations of artificial neural networks, the approach inspired by biology has been largely abandoned for a more practical approach based on statistics and signal processing. In some of these systems, neural networks or parts of neural networks (like artificial neurons) form components in larger systems that combine both adaptive and non-adaptive elements. While the more general approach of such systems is more suitable for real-world problem solving, it has little to do with the traditional, artificial intelligence connectionist models[8]. What they do have in common, however, is the principle of non-linear, distributed, parallel and local processing and adaptation. Historically, the use of neural network models marked a directional shift in the late eighties from high-level (symbolic) AI, characterized by expert systems with knowledge embodied

in if-then rules, to low-level (sub-symbolic) machine learning, characterized by knowledge embodied in the parameters of a dynamical system.

IV. RESULT

To evaluate the effectiveness of this method, grab text blocks including characters from images. The characters in the dataset involve English, and digits. All the experiments are done on the computer with a CPU of intel core i3 of 2.53GHZ. The proposed approach is evaluated by the recognition results and the process time cost per image. In this paper, uses performance, gradient, regression and validation checks. In the open vocabulary VIDI data, demonstrate the utility of joint word segmentation and recognition. Note the large intraword gaps and small interword spaces correctly identified. Overall error rates are lower than for ICDAR and SVT because the guidelines are given; yet some fonts and complex backgrounds make the remaining errors challenging.

On the end-to-end task, top-down word segmentation resolves cases where the bottom-up detector cannot reliably segment words. Despite using lower quality text detections as input[1], achieve results at par with the state of the art. A better initial text detection score would likely yield even better overall results.

Firstly, to evaluate the algorithm of text segmentation more accurately, the text part is cut out manually from the frames to make the test set. There are english characters in images and 2082 English letters in images[2]. Then the binary images are input to the software. Segmentation performance is evaluated by the resulting OCR recognition rate (RR) and precision rate (PR).

K-means algorithm based on RGB color space is run when k is 2, 3 and 4. The experimental results of english texts are When k is 3, the cluster with the highest grayscale center is considered as text and the others are non-text[7]. When k is 4, two kinds of situations are considered. The cluster with the highest grayscale center and the clusters with the first two highest grayscale centers are picked. When k is 3 we get the highest RR and PR. It is noticed that when k is 3 English texts are segmented and recognized well.

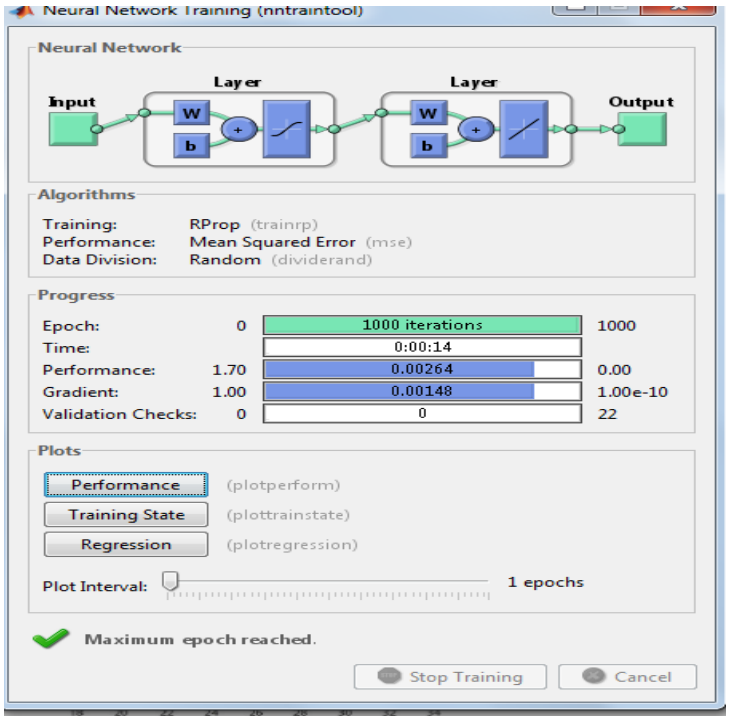

Figure 3. Artificial neural network

For a complete end-to-end test including detection, normalization, word segmentation, and recognition, use text detection result is submitted to the detection software. Following Figures shows the performance graph of the project. Many spurious detections give only one character words, so report results where these are dropped from the system output; ground truth is unaltered. Texts with height smaller than 8 pixels tend to be missed and they are usually found in multi-scale location algorithm.

V. CONCLUSION

Here presented a system that handles many stages of scene text reading in a probabilistic fashion, from binarization to appearance normalization to character segmentation and recognition. I do not rely on individual character detections, which are prone to false negatives. Only a coarse binarization is necessary for detecting guidelines that allow us to handle curved and rotated text. Model uses a hybrid open/closed vocabulary approach to balance bottom-up signals with top-down priors. Most importantly, it fully integrates segmentation and recognition. While these latter two stages are integrated, partially in training and fully in testing, we should like for the entire system to be even more integrated, in learning first. The binarization, guideline fitting, and character classification modules remain somewhat rudimentary. It is the integration of these stages and evaluating multiple hypotheses that makes the system perform well. However, further refinements to these key modules could bring even better performance. Scene text recognition is difficult—the world's vivid colors, uncontrolled lighting, and unpredictable perspectives conspire to make general machine reading a "grand challenge"-worthy task.

REFERENCES

[1] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic Detection and Recognition of Signs from Natural Scenes," IEEE Trans. Image Processing, vol. 13, no. 1, pp. 87-99, Jan. 2004.

[2] X. Chen and A.L. Yuille, "Detecting and Reading Text in Natural Scenes," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 366-373, 2004.

[3] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting Text in Natural Scenes with Stroke Width Transform," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2963-2970, 2010.

[4] L. Neumann and J. Matas, "Real-Time Scene Text Localization and Recognition," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 3538-3545, 2012.

[5] J. Ohya, A. Shio, and S. Akamatsu, "Recognizing Characters in Scene Images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 16, no. 2, pp. 214-220, Feb. 1994.

[6] Z. Saidane, C. Garcia, and J.L. Dugelay, "The Image Text Recognition Graph (iTRG)," Proc. Int'l Conf. Multimedia and Expo, pp. 266-269, 2009.

[7] L. Neumann and J. Matas, "A Method for Text Localization and Recognition in Real-World Images," Proc. Asian Conf. Computer Vision, pp. 770-783, 2010.

[8] A. Mishra, K. Alahari, and C.V. Jawahar, "Top-Down and Bottom- Up Cues for Scene Text Recognition," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2687-2694, 2012.

[9] J.J. Weinman, E. Learned-Miller, and A. Hanson, "Scene Text Recognition Using Similarity and a Lexicon with Sparse Belief Propagation," IEEE Trans. Pattern Analysis and Machine Intelligence vol. 31, no. 10, pp. 1733-1746, Oct. 2009.

[10] P. Shivakumara, S. Bhowmick, B. Su, C.L. Tan, and U. Pal, "A New Gradient Based Character Segmentation Method for Video Text Recognition," Proc. Int'l Conf. Document Analysis and Recognition, pp. 126-130, 2011.

