Scene Text Recognition with Bilateral Regression

Jacqueline Feild and Erik Learned-Miller

Technical Report UM-CS-2012-021
University of Massachusetts Amherst

Abstract

This paper focuses on improving the recognition of text in images of natural scenes, such as storefront signs or street signs. This is a difficult problem due to lighting conditions, variation in font shape and color, and complex backgrounds. We present a word recognition system that addresses these difficulties using an innovative technique to extract and recognize foreground text in an image. First, we develop a new method, called *bilateral regression*, for extracting and modeling one coherent (although not necessarily contiguous) region from an image. The method models smooth color changes across an image region without being corrupted by neighboring image regions. Second, rather than making a hard decision early in the pipeline about which region is foreground, we generate a set of possible foreground hypotheses, and choose among these using feedback from a recognition system. We show increased recognition performance using our segmentation method compared to the current state of the art. Overall, using our system we also show a substantial increase in word accuracy on the word spotting task over the current state of the art on the ICDAR 2003 word recognition data set.

1 Introduction

Scene text recognition is the problem of recognizing arbitrary text in the environment. Examples of scene text include street signs, business signs, grocery item labels, and license plates. With the increased use of smartphones, scene text recognition has the potential to contribute to a number of important applications, including improving navigation for people with low vision and recognizing and translating text into other languages.

This problem is similar to the well studied area of optical character recognition (OCR) for documents, but cannot be solved by plugging in existing solutions for several reasons. Images of natural scenes have many characteristics that make them difficult to analyze. They contain more extreme lighting variation, may include unusual or highly stylized fonts, often vary in color and texture and may be captured from a wide range of viewing angles. In addition, scene text images usually contain only a few words, so it is more difficult to benefit from linguistic constraints or from repeated patterns of appearance. Due to these differences, a separate solution to the scene text problem is warranted.

In this paper we present a solution to the problem of word recognition in natural images. Specifically, we focus on word spotting, which assumes recognized words come from a specific lexicon. This problem was introduced for scene text recognition by Wang et al. [1, 2].

In this paper, we make the following contributions:

- We present a new method to segment individual regions in a complex scene. Scene text images often have smooth color changes across the image, like those created by lighting. We develop a robust regression technique that we call *bilateral regression* that can model the color variations of a portion of the image (like the foreground) while ignoring other complex portions of the image. Bilateral regression does for regression in images what the bilateral filter [3] does for averaging in images. It gives us a clean way to do regression on one set of values even when they are spatially near another set of values. That is, *it lets us do local color modeling without the results being muddled by colors which are not intended to be part of what that regression is modeling.*
- After selecting from a number of foreground segmentation hypotheses using a weak recognition model, we demonstrate that combining this segmentation component with an algorithm to choose the best lexicon word label leads to a substantial increase in word accuracy over the current state of the art on the ICDAR 2003 word data set for the word spotting problem. We also obtain comparable word recognition accuracy on another public data set.

In the next section we describe how our proposed techniques relate to existing work. In section 3 we describe our segmentation technique, followed by our word recognition method. Section 4 includes experimental details and in section 5 we discuss results and future work.

2 Related Work

There is existing work on many different subproblems of scene text recognition. Many people have proposed techniques for the character recognition problem, where character bounding boxes are given. De Campos et al. present an evaluation of six types of local features with a bag of words approach [4]. Weinman et al. incorporate many different sources of information into their recognition process, such as character appearance, bigram frequencies, similarity and lexicons [5]. Smith et al. also incorporate character similarity information to improve recognition performance [6]. Other approaches include convolutional neural networks that require no preprocessing [7] and a technique that uses image binarization followed by GAT correlation [8]. Donoser et al. also show that character recognition results can be improved using information from a web search engine [9]. The current state of the art character recognition results on the ICDAR 2003 data set are presented by Coates et al. [10]. They take an unsupervised approach to learning features from unlabeled data.

Others describe solutions for the word recognition problem, assuming cropped word images. Weinman et al. integrate both character segmentation and recognition using a semi-Markov model and character width information [11]. Saidine et al. use a graph-based

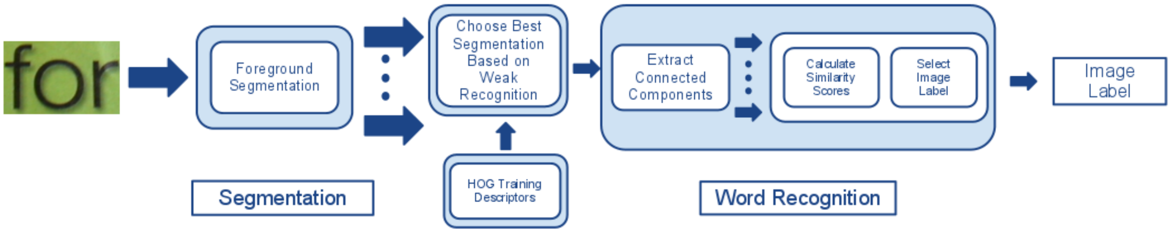


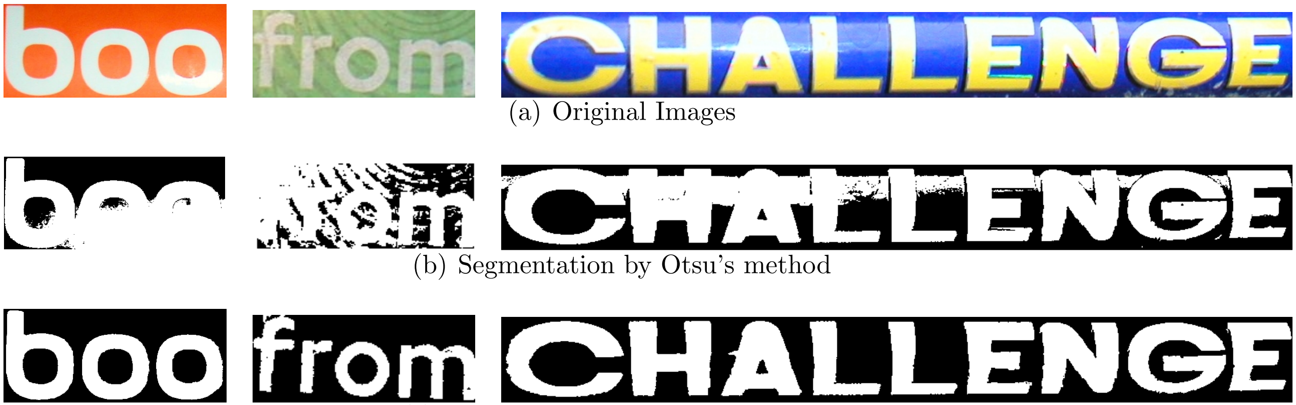
Figure 1: System Overview.

method to segment and recognize characters [12]. Wang et al. present a technique using histogram of oriented gradients (HOG) descriptors with a nearest neighbor classifier and pictorial structures [1].

There are also several examples of end-to-end systems that locate and label text in images. Chen et al. present a system to detect and recognize text in city scenes [13]. They use AdaBoost to classify text regions that are then binarized and processed by commercial optical character recognition (OCR) software. Chen et al. present a system to detect, recognize and translate text from Chinese signs [14]. Most recently, Wang et al. describe an end-to-end system using random ferns and pictorial structures [2]. Neumann et al. also present a text detection and recognition system based on maximally stable extremal regions (MSERs) that uses feedback information to improve the process [15].

One of the major differences between the existing approaches to word recognition and our work is whether images are initially segmented into foreground and background layers. Many recent techniques evaluate all possible character locations and sizes to find potential characters in an image [11, 2, 15, 7]. The benefit of these types of techniques is that they do not rely on an initial hard segmentation step. The downside is that comparisons must be done for every sub window in the image, and there is potential for confusion when non-text regions exhibit characteristics of a character. For example, characters are sometimes seen in the 'negative space' of a word. Instead, a technique that includes segmentation can take advantage of coherence across an image. The color characteristics of easier characters can be used to help recognize more difficult characters. In this work we demonstrate that a system designed with an initial hard segmentation can outperform a system that evaluates all character locations.

A complete survey of segmentation methods is beyond the scope of this paper. Here we describe the techniques for the segmentation that are the most relevant to our work. Many of these methods cluster colors in the image to produce several possible segmentations, then choose the one that is most likely to be correct [16, 17, 18]. Similarly, Wang et al. [19] extract color information from confident text regions and use it to create segmentations. Mishra et al. [20] also extract foreground and background colors, and use an MRF model in an iterative graph cut framework. The approach we present in this paper is similar to these methods, but we use color clustering as a starting point to fit a regression model for each image. This allows us to segment a larger class of images, since we can model smooth color



(a) Original Images

(b) Segmentation by Otsu's method

(c) Bilateral Regression Segmentation

Figure 2: Example images where the color changes across the image. We model these changes using a regression-based segmentation method. This figure is best viewed in color.

changes, which often occur in scene text images.

Since color may change smoothly across and image, pixels that belong to the foreground text are not well modeled by the unimodal localized distributions (like the Gaussians) usually used in clustering. We can easily model these changes with our regression based model to extend the range of images that can be successfully segmented. Tu et al. [21] also use the idea of modeling changing colors across an image to improve segmentation results. Their work focuses on segmenting natural scenes, and they model regions with smooth color changes using Bezier surfaces. Relative to their work, ours presents a simpler framework and is adapted specifically to the problem of scene text recognition.

3 System Design

3.1 Segmentation

During the segmentation stage, our goal is to separate pixels in an image into two groups. The foreground should contain pixels that represent text, and all other pixels should be assigned to the background. Some existing object segmentation techniques divide images into coherent regions. However, in a scene text image, disjoint letters may be segmented as different regions, with no way to associate them as all belonging to the foreground. Other segmentation techniques use color information to group regions that are similar. These work well when images have two distinct colors, but as colors change across images and image backgrounds become more complex, it becomes harder to find the correct distinction between background and foreground pixels.

We observe that in scene text images, the foreground pixels are very often a single constant or smoothly varying color and the background may be very complex. Figure 2 shows examples of images with colors that change across the image. Figure 3 shows examples

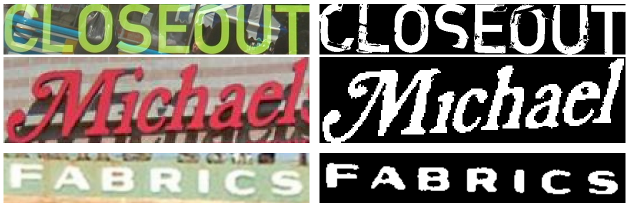


Figure 3: Examples of images with complex backgrounds and outlined text and their segmentations using bilateral regression.

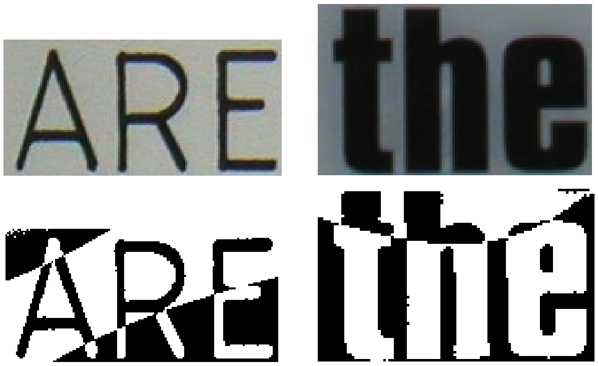


Figure 4: Sample segmentations that result from poor initialization using a mixture of two regressions.

where there are more than two prominent colors in an image and backgrounds are complex.

To address these characteristics of scene text images and problems with existing techniques, we present a regression-based segmentation technique. Regression allows us to model the smoothly varying color changes that often occur due to lighting. One possible approach for modeling an image with regression is to use a mixture of regressions.[1] To optimize such a model, an expectation-maximization procedure can be used to alternate between assigning pixels to different regressions and re-estimating the regressions based on the assignments. These can be hard or soft assignments. This type of method poses several difficulties. First, it can be difficult to initialize these models. Figure 4 shows examples of the type of segmentations that can result from poor initialization. Also, the complex backgrounds often found in scene text images are not well modeled by simple mixture models.

Instead of modeling every pixel in an image with a regression as the mixture of regressions framework does, we propose a technique that only models a subset of the pixels. We present a method to extract and model just the subset of pixels that belong to a coherent region that we are interested in modeling (like the foreground). This gives us a simple way to model pixel colors without the results being affected by nearby, unrelated pixels. We use this technique to model foreground hypotheses and present a selection procedure that chooses the best foreground segmentation from this set. Since this allows us to ignore background pixels, this technique is robust for images with complex backgrounds.

Next we describe our regression model in detail and then we explain its application to segmentation. We will also discuss how recognition guides the choice of the best candidate segmentation.

3.1.1 Bilateral Regression Segmentation

We now introduce a new regression based segmentation technique that models only the foreground of each image. We call this method bilateral regression, because it borrows ideas

[1] This is also known as a mixture of experts [22]

from bilateral filtering [23, 3].

Polynomial regression models can be used to model the relationship between two variables x and y as a polynomial curve. For example, a regression model of order two is the quadratic curve that best models y as a function of x,

$$y = ax^2 + bx + c$$

This can be easily extended to two dimensions, where the regression represents the quadratic surface that best models z as a function of x and y,

$$z = ax^2 + by^2 + cxy + dx + ey + f$$

In this form, we can use a regression model to model smooth brightness changes in an image as a function of pixel location. We work with color images in this paper, which we model using a separate quadratic surface for each color plane.

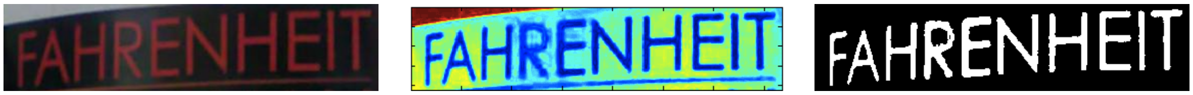
Our goal is to model only the foreground of an image, so our approach is to use a weighted regression, where each pixel is weighted according to how close it is to the foreground in feature space. This allows the regression to select out pixels we are interested in modeling (those that are part of the foreground text) and to ignore pixels that are a poor fit (those that are part of the background scene). Since we do not know the color of the foreground text a priori, we model the top n most prominent colors in each image separately and then automatically select the best segmentation.

For each foreground color, we calculate pixel weights as in bilateral filtering to select the subset of similar pixels automatically. Each pixel is weighted according to its spatial distance from a representative seed pixel, combined with its distance in color space. To calculate these distances, we use two Gaussian distributions generated from the seed pixel p from image I. We define $p = I(x, y)$ to have color $c_p = (r_p,g_p,b_p)$. The first distribution G_s is a two dimensional Gaussian distribution based on the spatial location of pixel p. It has $\mu = (x, y)$ and $\sigma = \sigma_s$ in both dimensions. The second distribution G_c is a three dimensional Gaussian distribution based on the color of pixel p with $\mu = (r_p, g_p, b_p)$ and $\sigma = \sigma_c$ in all dimensions. The weight of each pixel q with color c_q is then

$$w_q = G_s(||p - q||) * G_c(||c_p - c_q||).$$

These weights allow the regression model to ignore pixels that are a poor fit, so the regression represents a close fit to the foreground pixels. Additionally, the model can ignore an arbitrary amount of data that is too far away in feature space. This can be thought of as a type of image-adaptive robust regression, just the way the bilateral filter can be thought of as a image-adaptive, robust way of estimating the local mean of an image. This idea is similar in spirit to several extensions to the bilateral filter that include linear components [24, 25, 26]. However, our goal is not to smooth images, but to use the weights to select a subset of pixels to build a local model that fits the data well.

We can create a segmentation from this model by calculating the error between each pixel and the model. We threshold the error image using Otsu's method to obtain a segmentation. Figure 5 shows an example image, the regression error image and the resulting segmentation.



(a) Original Image (b) Error Image (c) Segmentation

Figure 5: An example image, the corresponding regression error image (blue represents low error and red represents high error) and the resulting segmentation image. This figure is best viewed in color.

Once we have segmentations for the n most prominent color regions, we want to automatically select the segmentation representing the true foreground. To do this, we choose the segmentation with the components that can best be recognized as characters. We represent each cropped connected-component image with a histogram of oriented gradients (HOG) descriptor and calculate the l_1 distance to each image in a reference set of synthetic character images from 200 different fonts for 62 character classes. These include twenty-six uppercase letters, twenty-six lowercase letters and ten digits. The images are provided by Weinman [5]. The score of a segmentation image is the average of these distances over all components.

Before scoring connected components, we filter out noisy components that are not likely to be text. We remove components that have a height of less than one third of the image height and those that are more than 2.5 times as wide as they are tall. We also remove components that span the entire width or height of the image, since we know that the input images have a at least a small border around each word. In addition, we filter out images that contain a large amount of overlapping connected components, since the characters in a good segmentation should not be overlapping.

We want to choose a segmentation with foreground components that cover the image area as much as possible, so we choose the segmentation with the best score from the those that are within 10 percent as covered as the most covered of the choices.

We implemented this technique in Matlab and it takes an average of 3 seconds per image to produce the final foreground/background segmentation from a color image. We ran this implementation on a system with a 2.6GHz AMD Opteron 4180 processor with 32 GB of RAM.

3.2 Word Recognition

During the word recognition phase, the goal is to label each image with a lexicon word. The following sections describe how we select a word label for each image, given a foreground segmentation image.

3.2.1 Nearest Neighbor Similarity Scores

For each connected component in a segmentation image, we compute similarity scores to each of the 62 possible character classes. As above, we represent each cropped connected-component image with a HOG descriptor and calculate the l_1 distance to each image in the reference set described in the previous section. We use a nearest-neighbor approach where the similarity score for each character class is the distance to the nearest neighbor in that class. So for each connected component, we compute a vector of 62 similarity scores.

3.2.2 Image Label Selection

Now we present our method for choosing the most likely lexicon word label for an image, given the similarity scores for each character in the word calculated in the previous step.

For each character, we form an equivalence class containing the three character classes with the highest similarity. Then we calculate the string edit distance to each lexicon word, where the substitution of a character for a member of its equivalence class has zero cost. The string edit distance returns the minimum number of insertions, deletions and non-equivalence class substitutions required to transform one string into the other.

We label the image with the lexicon word that has the smallest edit distance. If there is more than one lexicon word with the smallest edit distance, we repeat the process, only we form larger equivalence classes from the top ten choices for each character. We do this because we want to favor words that include characters that were found to be similar in appearance. We calculate the edit distance to the remaining tied words and choose the lexicon word with the smallest value. If tied words remain, we choose a random word from this final set of ties.

We implemented this word recognition method in Matlab and it takes an average of 1 second per image to produce a label, given a segmentation image. We ran this implementation on a system with a 2.6GHz AMD Opteron 4180 processor with 32 GB of RAM.

4 Experiments

In the following sections, we evaluate both our proposed segmentation method and our complete word recognition system.

4.1 Parameter Selection

For our experiments, we chose the value of n segmentation choices by looking at the performance of a range of values on a training set. We use the publicly available ICDAR 2003 Robust Reading Competition word training set [27], which contains 1156 cropped word images. For each number of segmentation choices from two to ten, we calculated the word recognition accuracy of our system. Figure 6 contains results of this experiment. This shows that the accuracy of our system is not very sensitive to the choice of this parameter. We use a value of $n = 6$ for all experiments described in this paper since it performs the best.

We set values for σ_s and σ_c for the regression weights experimentally. For all experiments on all data sets, σ_s is one third of the image width and $\sigma_c = 10$.

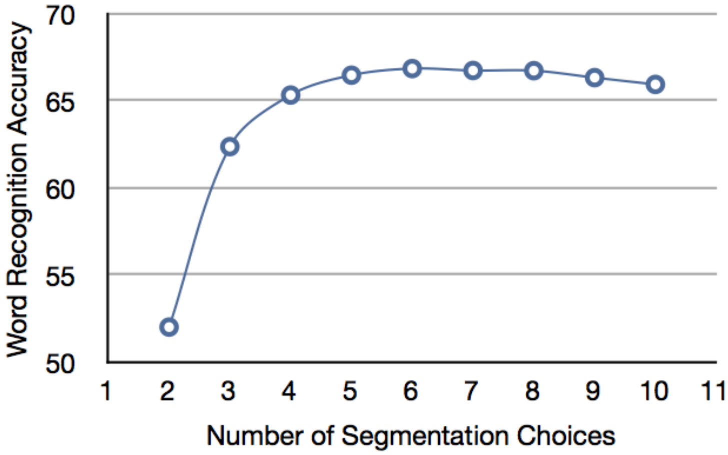


Figure 6: Word recognition accuracy results for different numbers of segmentation choices

4.2 Bilateral Regression Segmentation Evaluation

To evaluate the new segmentation technique we propose in this paper, we need to compare the segmentations we produce to those produced by existing segmentation methods for scene text. One way to do that is to compare the segmentations to foreground/background ground truth information for a data set. As far as we know, complete ground truth information does not exist for any scene text data set, including the widely used ICDAR scene text data sets. In addition, this analysis does not capture exactly what we are interested in evaluating. Since the segmentation of scene text is done as an initial step for a recognition process, we want to compare whether our segmentations allow us to recognize words better than another segmentation method. We do this by varying the segmentation method used by our complete recognition system.

4.2.1 Data set

We use the ICDAR 2003 data set for this evaluation. We use the scene test set, which contains 1107 cropped word images. We use the scene test set instead of the word test set because we were provided segmentations from the state-of-the-art segmentation method published by Mishra et al. [20] for direct comparison.

Word spotting also requires that a lexicon be provided. We follow the experiments of Wang et al. [2] and use two different approaches. The first is to use a lexicon made of the ground truth for all images in the data set. This is referred to as ICDAR03(FULL). The second is to use a lexicon that contains the ground truth for the image plus 50 random words from the data set. This is called ICDAR03(50).

Segmentation Method	ICDAR03(FULL)	ICDAR03(50)
Otsu	58.81	66.40
Mishra et al.	66.33	74.76
Bilateral Reg.	67.76	76.53

Table 1: Word accuracy for word spotting on the ICDAR 2003 scene data set of 1107 words.

4.2.2 Results

Table 1 shows the word recognition accuracy for both lexicon versions for the ICDAR03 data set. These results are evaluated in a case-insensitive way. This means that the label 'The' for an image with ground truth 'THE' is considered correct. Since our algorithm may contain a random choice during the labeling process, the accuracies we report are the average over 50 trials. We compare our technique to two existing segmentation methods. The first is Otsu's method [28] and the other is by Mishra et al. [20].

These results show that our segmentation method provides more accurate recognition than existing methods. Our method is more than an order of magnitude faster than the method by Mishra et al. Their method takes an average of 32 seconds per image while our method takes an average of 3 seconds per image.

4.2.3 Segmentation Selection Evaluation

Since our method produces n segmentations and chooses the best automatically, we want to analyze this selection process. We performed the following experiments using the ICDAR 2003 word test set with 1110 cropped word images. We compare our selection process to two baseline selection techniques and an oracle. The first baseline process is to always choose the segmentation created by the most prominent color in the image (assuming it is the foreground). The second baseline process is to always choose the segmentation created by the second most prominent color in the image (assuming that the most prominent is the background). The oracle chooses the segmentation that results in the best labeling of an image. That is, if a segmentation results in the correct labeling it is chosen. The word accuracies for the first and second baseline processes are 13.24% and 44.23% respectively and the word accuracy of the oracle is 71.80%. The word accuracy for our selection process is 66.94%, which is just a few percent less than the oracle. This shows that the high level recognition information we use in our selection process plays an important role in improving segmentation selection.

4.3 Complete Word Recognition System Evaluation

We evaluate our complete word recognition system by comparing it to the existing state-of-the-art system for the problem of word spotting.

	ICDAR03(FULL)	ICDAR03(50)	SVT
Wang et al.	62.00	76.00	57.00
Otsu + Word Rec.	67.21	72.13	43.16
Bilateral Regression + Word Rec.	73.43	79.47	54.20

Table 2: Word accuracy for word spotting on the ICDAR03 and SVT data sets. The ICDAR03 data set used is a subset of the original, to allow for a fair comparison to existing work.

	ICDAR03(FULL)	ICDAR03(50)	ICDAR11(FULL)	ICDAR11(50)
Bilateral Regression	66.78	76.03	62.28	72.69

Table 3: Word accuracy for word spotting on the complete ICDAR 2003 and ICDAR 2011 data sets.

4.3.1 Data sets

We evaluate our method on three publicly available data sets. The first is the ICDAR 2003 data set. The word recognition version of this test set contains 1110 cropped words. Following the experiments of Wang et al. [2], we present results on a subset, removing all words that contain non-alphanumeric characters and those with a length of two or less, for a total of 862 words. We also present results on the complete test set of 1110 words. We include these results to allow for future comparisons with our method. We also provide results on the new ICDAR 2011 word test set [29]. We do not know of any existing word spotting results for this data set, but provide ours for future comparison and completeness. The third data set is the Street View Text (SVT) data set [1], designed specifically for the word spotting problem. This test set consists of 647 words from 250 images.

For the ICDAR data sets, we include the same lexicons as described as above, and include results for ICDAR(FULL) and ICDAR(50). Since the Street View Text data set was designed for this problem, each image has an associated lexicon of around 50 words, including the ground truth.

4.3.2 Results

Table 2 shows the word recognition accuracy for both lexicon versions for the ICDAR data set and the SVT data set. As in the previous evaluation, these results are also evaluated in a case-insensitive way. Additionally, the accuracies we report are the average over 50 trials. We compare our method to the current state of the art system by Wang et al. [2]. We also compare to a version of our word recognition system that uses Otsu's method for segmentation.

Using our method, there is a large increase in word accuracy on ICDAR(FULL) from 62% to 73.43%. This is a 30% reduction in error over the current state of the art. There is

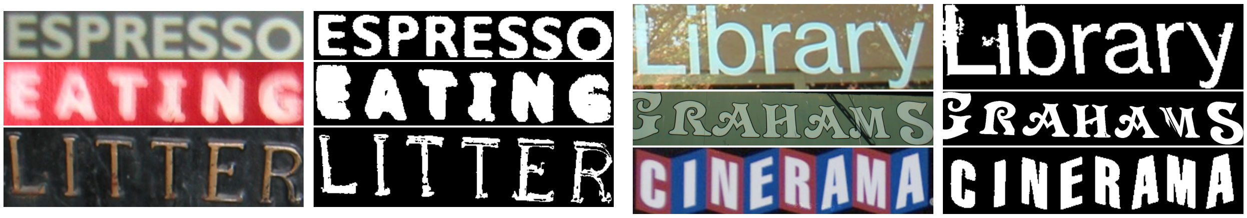


Figure 7: Examples of words that we identify correctly and their foreground segmentations.

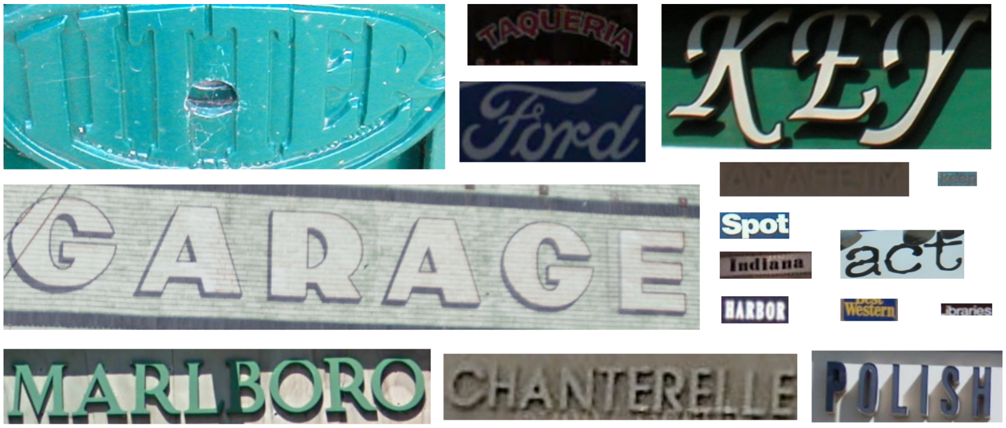


Figure 8: Examples of words that we identify incorrectly. Characteristics that make these images difficult are low resolution, abrupt lighting changes, connected text, and low contrast.

also a smaller increase from 76% to 78.8% on ICDAR(50) and a decrease from 57% to 53.4% for the SVT data set. Figure 7 shows examples of sign images that we label correctly and their segmentations. Figure 8 shows examples of sign images that we label incorrectly. The difficulties include low resolution, low contrast, abrupt lighting changes and connected text.

Table 3 shows the word recognition accuracy for word spotting on the complete ICDAR 2003 and ICDAR 2011 data sets. These are provided for future comparison and completeness.

5 Discussion

We chose a segmentation-based approach because we observed that by segmenting images into foreground and background components, we can eliminate many areas of the image that might exhibit features of text. We believe that this contributes to the method's success in many instances. The disadvantage to this approach is that if text is connected, or if an image is blurry or low resolution and the boundary between characters becomes less clear, it is difficult to find the correct segmentation. This is because our technique relies on recognizing distinct connected components to select the best segmentation. We hope to address this problem in future work.

In our experiments, we see a large increase in word accuracy on the ICDAR03 data set, but a modest decrease on the SVT data set. We believe this is because the images in the SVT data set are much more difficult to segment. Overall, they have a lower resolution than the images in the ICDAR03 data set and they exhibit more artifacts due to blur. This may be because they were collected from Google Street View and the images are taken from a moving vehicle. In this setting, approaches that do not rely on segmentation seem to perform better. However, when images have sufficient resolution and less blur, such as in the ICDAR03 data set, our approach based on segmentation performs better.

These experiments show that bilateral regression segmentation provides a good way to model foreground text in images. It allows us to model smooth color changes in just the subset of pixels that belong to the foreground text, while ignoring the background pixels altogether. This makes it suitable for segmenting images with color changes like those caused by lighting and complex backgrounds. Combined with a simple and fast nearest neighbor word recognition component, this technique improves performance on the word spotting task over the current state of the art.

Acknowledgements

The authors thank Evan Shelhamer for his help with this work. Jacqueline Feild was supported by an NSF Graduate Research Fellowship.

References

[1] Wang, K., Belongie, S.: Word spotting in the wild. In: European Conference on Computer vision. (2010)

[2] Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: International Conference on Computer vision. (2011)

[3] Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: International Conference on Computer Vision. (1998)

[4] De Campos, T., Babu, B., Varma, M.: Character recognition in natural images. In: International Conference on Computer Vision Theory and Applications. (2009)

[5] Weinman, J., Learned-Miller, E., Hanson, A.: Scene text recognition using similarity and a lexicon with sparse belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2009)

[6] Smith, D., Feild, J., Learned-Miller, E.: Enforcing similarity constraints with integer programming for better scene text recognition. In: International Conference on Computer Vision and Pattern Recognition. (2011)

[7] Saidane, Z., Garcia, C.: Automatic scene text recognition using a convolutional neural network. In: International Workshop on Camera-Based Document Analysis and Recognition. (2007)

[8] Yokobayashi, M., Wakahara, T.: Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation. In: International Conference on Document Analysis and Recognition. (2005)

[9] Donoser, M., Bischof, H., Wagner, S.: Using web search engines to improve text recognition. In: International Conference on Pattern Recognition. (2009)

[10] Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D., Ng, A.: Text detection and character recognition in scene images with unsupervised feature learning. In: International Conference on Document Analysis and Recognition. (2011)

[11] Weinman, J.: Typographical features for scene text recognition. In: International Conference on Pattern Recognition. (2010)

[12] Saidane, Z., Garcia, C., Dugelay, J.: The image text recognition graph (itrg). In: International Conference on Multimedia and Expo. (2009)

[13] Chen, X., Yuille, A.: Detecting and reading text in natural scenes. In: International Conference on Computer Vision and Pattern Recognition. (2004)

[14] Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic detection and recognition of signs from natural scenes. IEEE Transactions on Image Processing **13** (2004)

[15] Neumann, L., Matas, J.: A method for text localization and recognition in real-world images. In: Asian Conference on Computer Vision. (2010)

[16] Thillou, C., Gosselin, B.: Color binarization for complex camera-based images. In: Proc. Electronic Imaging Conference of the International Society for Optical Imaging. (2005)

[17] Wang, B., Li, X., Liu, F., Hu, F.: Color text image binarization based on binary texture analysis. Pattern Recognition Letters **26** (2005)

[18] Kita, K., Wakahara, T.: Binarization of color characters in scene images using k-means clustering and support vector machines. In: International Conference on Pattern Recognition. (2010)

[19] Wang, X., Huang, L., Liu, C.: A novel method for embedded text segmentation based on stroke and color. In: International Conference on Document Analysis and Recognition. (2011)

[20] Mishra, A., Alahari, K., Jawahar, C.: An mrf model for binarization of natural scene text. In: International Conference onDocument Analysis and Recognition. (2011)

[21] Tu, Z., Zhu, S.: Image segmentation by data-driven markov chain monte carlo. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002)

[22] Quandt, R., Ramsey, J.: Estimating mixtures of normal distributions and switching regressions. Journal of the American Statistical Association **73** (1978) 730–738

[23] Aurich, V., Weule, J.: Non-linear gaussian filters performing edge preserving diffusion. In: DAGM Symposium. (1995)

[24] Buades, A., Coll, B., Morel, J.: The staircasing effect in neighborhood filters and its solution. IEEE Transactions on Image Processing **15** (2006)

[25] Choudhury, P., Tumblin, J.: The trilateral filter for high contrast images and meshes. In: Eurographics Symposium on Rendering. (2003)

[26] Elad, M.: On the origin of the bilateral filter and ways to improve it. IEEE Transactions on Image Processing **11** (2002) 1141–1151

[27] Lucas, S., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: Icdar 2003 robust reading competitions. In: International Conference on Document Analysis and Recognition. (2003)

[28] Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics **9** (1979)

[29] Shahab, A., Shafait, F., Dengel, A.: Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In: International Conference on Document Analysis and Recognition. (2011)