Copyright ©2010 IEEE. Reprinted from K. Boyer, M. Çetin, and S.-W. Lee editors, *IAPR Proc. Intl. Conf. on Pattern Recognition*, Istanbul, Turkey, August 2010. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

Typographical Features for Scene Text Recognition

Jerod J. Weinman
Dept. of Computer Science
Grinnell College
weinman@grinnell.edu

Abstract

Scene text images feature an abundance of font style variety but a dearth of data in any given query. Recognition methods must be robust to this variety or adapt to the query data's characteristics. To achieve this, we augment a semi-Markov model—integrating character segmentation and recognition—with a bigram model of character widths. Softly promoting segmentations that exhibit font metrics consistent with those learned from examples, we use the limited information available while avoiding error-prone direct estimates and hard constraints. Incorporating character width bigrams in this fashion improves recognition on low-resolution images of signs containing text in many fonts.

1 Introduction

Scene text images captured by portable cameras feature an abundance of variety due to noise, unusual fonts/typesetting, low resolution, and other factors that make the recognition problem challenging. Simultaneously, there is typically very little text with which to reliably build a model of font or language properties, unlike many document recognition problems.

Techniques that adapt the recognition model to the query properties have been successful in document processing. Examples include direct character appearance [4, 5, 1] and font metrics (e.g., bold, italics) or font identity [6, 3, 9, 14]. Unfortunately, these methods require the relatively large amounts of data present in most documents and thus are not directly applicable to scene text recognition. Other work has leveraged the assumption of consistency in character appearance by using a soft, probabilistic model for scene text [12], but the model required that characters be segmented beforehand. While some font properties can be difficult to estimate with a small sample, the width of characters (given the font size) often exhibit predictable relationships.

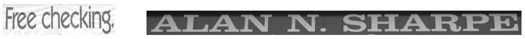

Figure 1. Signs may have unusual character aspect ratios. Normalized for x-height, the median widths of characters in these two signs differ by more than $5\times$.

In camera-based scene text recognition, it is not always reasonable to assume that characters and words may be easily binarized and segmented prior to recognition. Thus, these processes must be integrated for optimal results [13]. In this paper, we propose the use of additional font-based features in a robust probabilistic model that combines word and character segmentation with recognition. In addition to the usual character appearance and bigram features, we introduce an adaptive, contextual font property feature that does not require prior segmentations or large amounts of query data.

In the next section we describe the motivation for a simple font feature and then briefly describe the elements of our recognition model. The recognition process is detailed and, finally, we present experiments on scene text images demonstrating the potential for such typographical features to improve recognition.

2 Character Width

Figure 1 shows signs in two very different fonts. When a recognition algorithm must perform segmentation of the image into its constituent characters, the width of those character segments must be considered, even if only implicitly. If these character widths exhibit regular patterns, they should be taken advantage of during the segmentation/recognition process.

We may consider two potential types of information: local and contextual. For instance, a character unigram probability $p(y)$, where y is some character label, is a *local* information source that may help bias our recognition for the more common characters. However, a *contextual* bigram probability $p\left(y \mid y'\right)$ can greatly improve recognition by jointly considering the label y' assigned to the previous character in the bias for a current character y.

We propose that character width, a property of the font a character is rendered in, exhibits similar statistical regularities useful for recognition, especially when labels *and* segmentations are unknown. The "unigram" probability of a single character's label and corresponding width $p\left(y, w\right)$ contains some information; for example, an "m" is usually wide and an "i" is typically narrow. However, in any given font, the "m" may be very narrow, or the "i" may be very wide in a different font. Thus, a contextual "bigram" model of character labels and widths should be more informative. That is, the width w' and label y' of some character should inform us about the width w and label y of another character in the same font. A probability model of the form $p\left(y, w \mid y', w'\right)$ would capture just this dependency.

Figure 2 demonstrates such regularity using alphanumeric character images from 1866 fonts. Keeping the font size constant, we measure the correlation between the widths of each character pair, as given by $\rho = \frac{\sigma^2_{XY}}{\sigma_X \sigma_Y}$ where σ^2_{XY} is the covariance of X and Y and σ_X and σ_Y are the standard deviation of X and Y, respectively. These figures demonstrate that some character pair widths are nearly perfectly correlated, and most are highly correlated.

Thus, while many text recognition systems include a langage-based character bigram model along with the character recognition component, we propose to add a font-based character *width* bigram model to our system. This has the advantage of retaining the ability to use an efficient dynamic programming algorithm for inference, while incorporating font properties in the recognition process without the need for large amounts of observed query data.

3 Recognition Model

As in prior work [13], we use a discriminative semi-Markov field for jointly performing segmentation and recognition [8]. Because of its segmentation properties, the semi-Markov model is richer than the typical Markov model. Further, since it is trained to optimize the predictive posterior distribution, it uses richer contextual features of the input image than an HMM, which requires stricter independence assumptions. Here we review how to score a parse—a segmentation and labeling—of an image containing text, while the next section describes how to find the optimal parse.

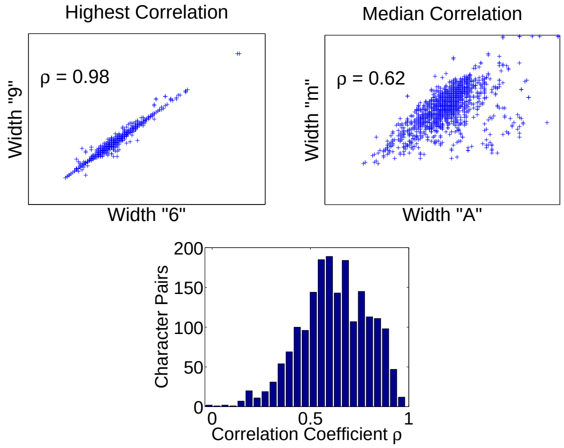

Figure 2. TOP: Scatter plots of character widths over several fonts for the most correlated (6 and 9, left) and more typically correlated (A and m, right) character pairs. A fixed font size is used. BOTTOM: Histogram of correlation coefficients for widths of all characters pairs.

A segmentation of an image containing a text string induces a set of unknowns $\mathbf{y} = y_1 y_2 \cdots y_N$ requiring labels and a corresponding set of discriminant functions $\{U_C\}_{C \in \mathcal{C}}$ for rating the labels assigned in a complete parse. Each unknown y_i takes a label from the set $[\mathtt{A} - \mathtt{Za} - \mathtt{z0} - \mathtt{9}\sqcup]$, where $\sqcup$ is an inter-word space. Modeling spaces explicitly allows word segmentation to be seamlessly integrated. Since different segmentations must compete with each other, two properties of the segmentation itself are also scored: overlaps of character segments (allowing for ligatures) and gaps between the segments requiring labels.

Given a segmentation, we define the conditional probability

$$p\left(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}\right) \propto \exp\left\{ \sum_{C \in \mathcal{C}} U_C\left(\mathbf{y}_C, \mathbf{x}; \boldsymbol{\theta}\right) \right\}, \qquad (1)$$

where $\mathbf{x}$ represents the observed image, and $\mathcal{C}$ is a set of discriminants applied to the segmentation. The exact set of discriminants depends on the segmentation. For example, each unknown character y_i will have one appearance-based recognition discriminant. The notation $\mathbf{y}_C$, where $C \subseteq \{1, \ldots, N\}$, indicates a discriminant-specific subset of unknowns from $\mathbf{y}$ that are arguments to the discriminant U_C.

We use five classes of discriminant functions to calculate the score of a parse. Because this is a one-dimensional, integral (pixel-based) segmentation problem, it is helpful to understand where each segment begins and ends. Let r and t represent the beginning and end indices (respectively) of some segment with label y and corresponding width w. Let n be the end index of the previous segment with label y' and corresponding width w'. The vectors $\boldsymbol{\theta}$ are learned weights associated with image feature vectors $F\left(\mathbf{x}\right)$. The discriminant functions used by our model are as follows.

Appearance Each character segment is scored by a linear discriminant, $U^A_{r,t}\left(y, w, \mathbf{x}; \boldsymbol{\theta}^A\right) = \boldsymbol{\theta}^A\left(y, w\right) \cdot F_{r,t}\left(\mathbf{x}\right)$, for a character y and the span width w from r to t, learning not only appearance, but that "m"s are wide and "i"s narrow.

Character Bigrams Each pair of neighboring character segments is given a bigram score, $U^B\left(y', y\right)$.

Width Bigrams Each pair of neighboring character segments is given a score, $U^W\left(w', w, y', y\right)$, based on the segment widths w' and w in correspondence with the labels.

Overlap If neighboring character segments overlap, the term $U^O\left(n - r\right)$ is added, which depends on the degree of overlap, as indicated by n and r.

Gap Conversely, a gap between character segments from n to r is scored by a learned, linear discriminant $U^G_{n,r}\left(\mathbf{x}; \boldsymbol{\theta}^G\right) = \boldsymbol{\theta}^G_{n,r} \cdot F_{n,r}\left(\mathbf{x}\right)$ using the image features.

The MAP estimate of model parameters—linear weights $\boldsymbol{\theta}^A$ and $\boldsymbol{\theta}^G$ along with entries for U^B, U^W, and U^O—is learned from data via decoupled piecewise training [11], a convex optimization problem.

4 Model Inference

Recognition involves finding the parse—a segmentation and corresponding labeling—that maximizes the probability (1). This is equivalent to maximizing the sum in the exponent, whose summands are described in the previous section. Here we describe how to find the best score.

To find the best parse, we construct a three-dimensional dynamic programming table. Let $S\left(t, w, y\right)$ be the optimal parse score for a segment ending at index t with character y of width w. The table is built by adding a new segment and labeling (for each possible character width) to the previous best parse via the recurrence

$$S\left(t, w, y\right) = \max_{n, y', w'} S\left(n, y', w'\right) + P\left(n, w', w, t, y', y\right). \tag{2}$$

$P\left(n, w', w, t, y', y\right)$ represents the parse score for adding a segment of width w that ends at index t and is labeled y, while the previous character y' of width w' ended at index n. The base cases are $S\left(0, w, y\right) = 0$, for all w and y, with $S\left(t, w, y\right) = -\infty$ for $t < 0$. The parse score is the sum of the new discriminants it induces,

$$\begin{aligned} P\left(n, w', w, t, y', y\right) = U^O_{n,r} + U^G_{n,r} + U^A_{r,t}\left(y\right) W^A_{r,t} \\ + \left[U^B\left(y', y\right) + U^W\left(y', y, w', w\right)\right] W^B_{m,t}, \end{aligned} \tag{3}$$

where m is the *start* index of the previous character segment. Thus, the total score S for the best parse is the sum of all the P terms, which is just the exponent in the Markov model (1). To normalize this sum for the number of segments in a parse, we add weights that assign the appearance and bigram scores (character or width) to every index of the segments they cover : $W^A_{r,t}$ is the width of the individual character span, and $W^B_{m,t}$ that of the bigram span. Gap indices are scored individually and thus do not require normalization.

5 Image Features and Pre-Processing

We use the same features and pre-processing described in earlier work [13], briefly summarized here. Steerable pyramid filters [10] at six orientations are used on the original grayscale image and then normalized for brightness to form the feature vectors $F\left(\mathbf{x}\right)$ for the gap and character appearance discriminants.

Although our data is manually annotated, classifiers can detect approximate scene text baselines [2]. The function U^A produces a score for the combined character identity and approximate width at each pixel in a region detected as text. Each such text region is coarsely binarized with Niblack's algorithm [7] to speed recognition by limiting the number of segmentations. We assume this is an over-segmentation, so that components may be combined.

Because the dynamic programming algorithm assumes a spatially one-dimensional array of scores, we collapse 2-D detected text regions into a line by retaining only the maximum score for each hypothesis (e.g. character label and width) over the rows of each column marked as text.

6 Experiments

Our evaluation is conducted on a set of 85 sign images, containing 183 words and 1144 characters in a variety of fonts. They are manually scaled to a 12.5px x-height and annotated with an approximate baseline.

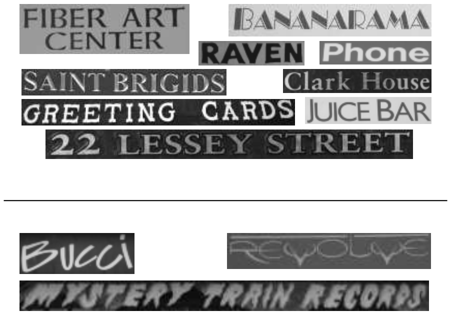

Figure 3. Examples of sign images correctly recognized by our model (top) and those remaining a challenge (bottom).

Table 1. Recognition results with increasing amounts of information.

Information	Char. Error (%)
Appearance	22.20
+ Char. Bigram	17.40
+ Char. and Width Bigrams	16.87

Model parameters are learned from data. We scaled 1600 base training fonts with horizontal aspect ratios scaled at $2^{-\frac{1}{2}}$, $2^{-\frac{1}{4}}$, 2^{0}, and $2^{\frac{1}{2}}$ to train U^A with 6400 fonts.[1] The bigram model U^B is trained on a corpus of 82 English books (49 million characters). U^O is a truncated quadratic, increasingly penalizing greater overlap of character bounding boxes. We quantized character widths to $w \in \{4, 8, 12, 16, 20, 24, 32\}$ pixels and trained the width bigram model U^W using the frequency of association of between pairs of character labels and quantized widths in the 1866 training fonts.

Table 1 shows the contribution of each information source in the model. Adding character bigrams to the baseline model using only character appearance reduces 21.7% of errors. Adding the character *width* bigrams on top of this reduces error another 3.0%, for a combined total of 24.0% error reduction.

7 Conclusion

We have demonstrated a significant degree of correlation between the widths of characters across fonts. Because scene text often features a variety of fonts, but only small samples for recognition, any attempt at adapting a recognition model must be somewhat speculative. We proposed a bigram model for incorporating statistical regularities between the widths of characters in a given font. Such a model has the advantage of being contextual—helping to promote internal consistency among the characters in a parse, while also retaining locality so that efficient dynamic programming algorithms may be used.

It is relatively straightforward to integrate such a width bigram into a flexible probabilistic model for recognition. We have shown that such a font model has power to resolve errors that a local language model—character bigrams—does not.

Acknowledgments

Thanks to Augustus Lidaka and Shitanshu Aggarwal, who implemented a parallelized learning algorithm used in the experiments, and Grinnell College, which supported them.

References

[1] T. M. Breuel. Character recognition by adaptive statistical similarity. In *Proc. ICDAR*, volume 1, pages 158–162, 2003.

[2] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *Proc. CVPR*, pages 366–373, 2004.

[3] R. Cooperman. Producing good font attribute determination using error-prone information. In *Proc. SPIE*, volume 3027, pages 50–57, 1997.

[4] J. D. Hobby and T. K. Ho. Enhancing degraded document images via bitmap clustering and averaging. In *Proc. ICDAR*, volume 1, pages 394–400, 1997.

[5] T. Hong and J. J. Hull. Improving OCR performance with word image equivalence. In *Symp. on Doc. Analysis and Info. Retr.*, pages 177–190, 1995.

[6] S. Khoubyari and J. J. Hull. Font and function word identification in document recognition. *Comp. Vis. and Im. Understanding*, 63(1):66–74, 1996.

[7] W. Niblack. *An Introduction to Digital Image Processing*. Prentice-Hall, 1986.

[8] S. Sarawagi and W. W. Cohen. Semi-Markov conditional random fields for information extraction. In *NIPS*, pages 1185–1192, 2005.

[9] H. Shi and T. Pavlidis. Font recognition and contextual processing for more accurate text recognition. In *Proc. ICDAR*, pages 39–44, 1997.

[10] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proc. ICIP*, volume 3, pages 444–447, 1995.

[11] C. Sutton and A. McCallum. Piecewise training of undirected models. In *Proc. UAI*, pages 568–575, 2005.

[12] J. J. Weinman and E. Learned-Miller. Improving recognition of novel input with similarity. In *Proc. CVPR*, pages 308–315, June 2006.

[13] J. J. Weinman, E. Learned-Miller, and A. Hanson. A discriminative semi-Markov model for robust scene text recognition. In *Proc. ICPR*, Dec 2008.

[14] A. Zramdini and R. Ingold. Optical font recognition using typographical features. *IEEE Trans. PAMI*, 20(8):877–882, 1998.

[1] Font and sign images available at `http://www.cs.grinnell.edu/~weinman/research`