Probabilistic model for segmentation based word recognition with lexicon.

Sergey Tulyakov and Venu Govindaraju
CEDAR, SUNY Buffalo
{tulyakov,govind}@cedar.buffalo.edu

Abstract

In this paper we describe the construction of a model for off-line word recognizers based on over-segmentation of input image and recognition of segment combinations as characters in a given lexicon word. One such recognizer, Word Model Recognizer (WMR), is used extensively. Based on the proposed model it was possible to improve the performance of WMR.

1. Introduction.

The problem of off-line reading of unconstrained handwritten words has been studied extensively due to its role in many important applications such as reading addresses on mail-pieces [3, 6, 11], reading amounts on bank checks [7, 10], extracting census data on forms [2, 9], and reading address blocks on tax forms [12]. The main challenges are wide variety of writing styles, poor image quality and missing or extraneous strokes caused by segmentation errors.

The intuitive solution to the problem is to segment the word image into probable character sub-images, then try to recognize separate characters and combine results [4, 8]. The function of optical character recognizer (OCR) used is to provide confidence scores for supposed character images. Although many different OCRs are available, they are all mainly focused on classifying isolated character images. In practice, when dealing with unconstrained handwritten word images there is no guarantee that segmented sub-images will be single isolated characters. So OCR used for word recognition should be able to provide low confidence scores for non-character images.

Besides choosing the right OCR for word recognition, it is also important to know how to incorporate OCR confidence scores for individual characters into an overall confidence score for the entire lexicon word - should we take arithmetic mean, geometric mean or some other normalizing formula? This question addresses what OCR score truly means. For example, given an image and a hypothesis character c_i should the OCR produce a score representing the posterior probability $p(c_i|image)$ or the prior probability $p(image|c_i)$?

In this paper we describe the construction of a possible mathematical model for word recognizers that are based on the segmentation paradigm and use of a lexicon. The construction of the model is motivated by the comparison of two word recognizers existing in CEDAR: CMR (Character Model Recognizer)[4] and WMR (Word Model Recognizer)[8]. These recognizers use similar preprocessing and segmentation techniques. Using seemingly inferior character recognizer, WMR is able to perform better than CMR on word images. CMR uses the GSC (Gradient, Structural, Concavity)[5] character recognizer which is widely accepted as being very accurate.

2. Description of WMR recognition scheme.

WMR is a lexicon based recognizer, that is given a list of words. Its task is to assign a score to each word in the lexicon. The score represents how well the particular lexicon word matches the image.

WMR is also a character based recognizer, meaning that characters are the units that get recognized and the score for the lexicon word is a function of scores for all characters in that word. Over-segmentation and subsequent combinations of segments into super-segments produce hypothetical splittings of image into character sub-images used for recognition.

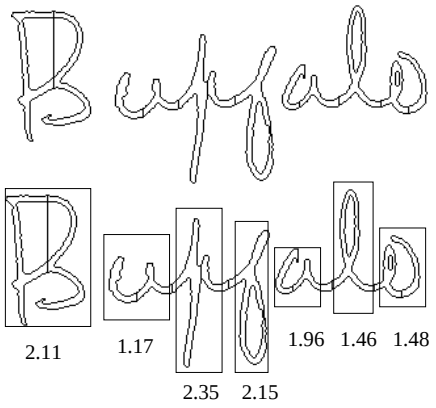
Character recognition uses 74-dimensional feature vector. Features are extracted from contour representation of the image, and they represent number of pixels with particular slope in 3 by 3 split of the image. Two features are global and represent width to height ratio and the number of pixels in horizontal direction to the number of pixels in vertical direction. A set of training images is used to produce a set of feature vectors. After that the k-means clustering algorithm is applied to produce a set of clusters (20-30 clusters). Denote $cl(c_i, j)$ as the center of j-th cluster for character c_i. The score of matching character c_i in the lexicon word to some sub-image is defined as the minimum of the Euclidean distances from feature vector $\overline{fv}$ corresponding

to the image to cluster centers $cl(c_i, j)$ for all j. A smaller score indicates better match of the sub-image to the character in the word.

The score for lexicon word matching particular splitting of image is the arithmetic mean of squares of character scores. Minimum of such scores gives the score of matching lexicon word to the image(figure 1).

(a)

(b)



(c) S("Buffalo")= $\frac{2.11^2 + 1.17^2 + 2.35^2 + 2.15^2 + 1.96^2 + 1.46^2 + 1.48^2}{7}$ = 3.44

Figure 1. Original image with segmentation points shown (a), best found combination of segments for lexicon word "Buffalo" (b) which minimizes score for this word computed in (c).

To summarize, the algorithm is as follows:

1. Split the image into segments so that characters in the image could be represented as a combinations of segments.

2. For each possible combination of segments (no more than 4 segments in combination) calculate the distance $\underline{d(i)} = min_j D(\overline{fv}, cl(c_i, j))$ where c_i is the character, $\overline{fv}$ is the feature vector of segment combination image, $cl(c_i, j)$ is the center of jth cluster for character i, and D(*,*) is the Euclidean distance in 74 dimensional feature space.

3. For each word $W = (i_1, i_2, .., i_k)$ in the lexicon calculate its score as $score(W) = \min \frac{d(i_1)^2 + d(i_2)^2 + .. + d(i_k)^2}{k}$ where min is taken over all possible groupings of segments to represent characters, $d(i_m)$ change accordingly. Finding the minimum is achieved by efficient dynamic programming algorithm.

3. Probabilistic model.

Here we try to develop a probabilistic model for a word recognizer. Assume a lexicon is given and recognition is based on segmentation of a word into character sub-images.

Let W be a lexicon word and $image$ be the image to be recognized. By Bayesian rule $P(W|image) = \frac{p(image|W)P(W)}{p(image)}$, and by implicit assumption $P(W)$ is the same for all lexicon words W in the lexicon. Hence finding the maximum of $P(W|image)$ is equivalent to finding maximum of probability density function $p(image|W)$ over all lexicon words W.

To approximate $p(image|W)$ we make rather broad assumptions:

1. Segmentation was successful, that is all characters in the image are separated and one of the segment combinations produces the correct split of the word image into character sub-images.

2. The probability density function $p(subimage_m|c_m)$ does not depend on other characters in the image or in the lexicon.

Based on these assumptions,

$$\begin{aligned} & p(image|W) \\ & = \max p(subimage_1, .., subimage_k|c_1, .., c_k) \\ & = \max \prod_{m=1,..,k} p(subimage_m|c_m) \end{aligned} \quad (1)$$

where max is taken over all possible combinations of segments into character sub-images.

Further, replacing the event of random image by the event of random feature, instead of looking for $\max \prod p(subimage_m|c_m)$ we are looking for

$$p(image|W) \cong \max \prod_{m=1,..,k} p((\overline{fv_m}|c_m). \quad (2)$$

One way to estimate $p((\overline{fv_m}|c_m)$ is to represent it as a mixture of Gaussian models.

$$p(\overline{fv_m}|c_m) = \sum_j P_{mj} \frac{1}{\sigma_{mj}^n (2\pi)^{n/2}} \exp\left(-\frac{1}{2} \frac{D(\overline{fv_m}, c(m,j))^2}{\sigma_{mj}^2}\right) \quad (3)$$

where P_{mj} is the probability of matching the jth Gaussian component for character c_m, $c(m,j)$ is the center of jth Gaussian component, σ_{mj} is its variance and n is the dimension of the feature vector. Parameters P_{mj}, $c(m,j)$ and σ_{mj} can be obtained using the EM algorithm[1].

Although formula (2) finds the best splitting of a word W into sub-images successfully, it does not work well for

finding the best word W among lexicon words. Running the word recognizer would show that the score of word W $p(image|W)$ is biased towards shorter words in the lexicon. This is generally due to the imperfection in feature choices and general inability of the character classifier to distinguish between character images and non-character images. To account for this bias we need to make additional normalization.

$$p(image|W) \cong \max \sqrt[k]{\prod_{m=1,..,k} p((\overline{fv_m}|c_m)} \quad (4)$$

To justify such normalization, assume that expectation of the score $p((\overline{fv_m}|c_m)$ is some constant c given that c_m is exactly a character represented by the image. $E(p((\overline{fv_m}|c_m))) = c$. Then given that word W is the truth word of $image$ and because of assumptions made above we obtain from formula (4):

$$E\left(\sqrt[k]{\prod_{m=1,..,k} p((\overline{fv_m}|c_m)}\right)$$
$$= \sqrt[k]{\prod_{m=1,..,k} E\left(p((\overline{fv_m}|c_m))\right)} = c$$

This means that expectation of truth word score from formula (4) does not depend on the word length, and it is reasonable to compare scores of different length words in the lexicon.

4. Adjusting WMR to probabilistic model.

It turns out that WMR recognizer fits well into the above described model. Given a cluster of center points we can model the probability of recognizing a character as

$$p(\overline{fv_m}|c_m) = \max_j \exp\left(-\frac{1}{2}\frac{D((\overline{fv_m}, cl(c_m, j))^2}{\sigma^2}\right) \quad (5)$$

with some arbitrary σ. Calculating $p(\overline{fv_m}|c_m)$ becomes equivalent to finding $d(i) = min_j D(\overline{fv}, cl(i, j))$. At the same time formula (4) leads to

$$p(image|W) = \max \sqrt[k]{\prod_{m=1,..,k} \exp\left(-\frac{1}{2}\frac{d(m)^2}{\sigma^2}\right)} = \max \exp\left(-\frac{1}{2\sigma^2}\frac{\sum_m d(m)^2}{k}\right) \quad (6)$$

which is equivalent to finding
$score(W) = \min \frac{d(i_1)^2 + d(i_2)^2 + .. + d(i_k)^2}{k}$.

Improvements to WMR could be made simply by modifying formula (5) to include the information about all clusters:

$$p(\overline{fv_m}|c_m) = \sum_j R_{mj} \exp\left(-\frac{1}{2}\frac{D((\overline{fv_m}, cl(c_m, j))^2}{\sigma^2}\right) \quad (7)$$

where R_{mj} is the ratio of the number of characters c_m templates used in jth cluster over the number of all character c_m templates. Another way is to use more general formula (3) and train parameters of the model directly, for example, by the EM algorithm.

5. Experiments.

Experiments were conducted on a 3000 set of word images extracted from postal address images. Lexicons of sizes 10 and 100 were used. Truth was always present in the lexicon. The percentages of cases with "truth" word getting highest score and cases with "truth" word being among two best entries was measured.

Results of the experiments are shown in table 1. We needed to derive the acceptable values of σ for formula (7). Different values between 2 and 0.1 were tried. The best results were achieved for $\sigma = 0.5$ though the improvement was present for a broad range of values between 0.8 and 0.3 . Similar improvements were obtained on other word sets.

The use of different σ_{mj} in formula (3) was hindered by over-fitting during training. Taking a uniform $\sigma = 0.5$ gives marginally better results compared to those obtained by using formula (7).

The reason for the smaller than expected increase in performance can be explained by the fact that WMR uses a very similar recognition methodology. Trying to adjust other recognizers to use the probabilistic model in formula (4) could lead to a bigger degree of improvement. Note that the above modifications require almost no additional processing time since number of templates (cluster centers or centers of Gaussian functions) remains the same.

Lexicon size	10		
Method	Original	Using (7)	Using (3)
% 1st correct	96.43%	96.60%	96.86%
% 1st OR 2nd	98.73%	98.83%	98.80%
Lexicon size	100		
Method	Original	Using (7)	Using (3)
% 1st correct	90.36%	91.39%	91.36%
% 1st OR 2nd	94.73%	95.33%	95.30%

Table 1. Results of applying probabilistic approach to WMR.

Acknowledgments

We would like to thank our colleagues at CEDAR for many fruitful discussions. In particular, the support of Evelyn Kleinberg, Jaehwa Park and Peter Slavik was very helpful.

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

[2] T. M. Breuel. Design and implementation of a system for the recognition of handwritten responses on us census forms. In *Proceedings of the Conference on Document Analysis System*, pages 109–134, Kaiserslautern, Germany, 1994.

[3] E. Cohen, J. Hull, and S. N. Srihari. Control structure for interpreting handwritten address. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1049–1055, 1994.

[4] J. T. Favata. Character model word recognition. In *Fifth International Workshop on Frontiers in Handwriting Recognition*, pages 437–440, Essex, England, 1996.

[5] J. T. Favata and G. Srikantan. A multiple feature/resolution approach to handprinted digit and character recognition. *International Journal of Imaging Systems and Technology*, 7(4):304 – 311, 1996.

[6] V. Govindaraju, A. Shekhawat, and S. N. Srihari. Interpretation of handwritten addresses in us mail stream. In *The third International Workshop on Frontiers in Handwriting Recognition*, pages 197–206, Buffalo, New York, 1993.

[7] D. Guillevic and C. Y. Suen. Cursive script recognition: A sentence level recognition scheme. In *The fourth International Workshop on Frontiers in Handwriting Recognition*, pages 216–223, Taipei, Taiwan, 1994.

[8] G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real-time applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):366–379, Apr. 1997.

[9] S. Madhvanath, V. Govindaraju, V. Ramanaprasad, D. S. Lee, and S. N. Srihari. Reading handwritten us census forms. In *Third International Conference on Document Analysis and Recognition (ICDAR-95)*, pages 82–85, Montreal, Canada, 1995.

[10] J. C. Simon, O. Baret, and N. Gorski. A system for the recognition of handwritten literal amounts of checks. In *Proceedings of the Conference on Document Analysis System*, pages 135–155, Kaiserslautern, Germany, 1994.

[11] S. Srihari. High-performance reading machines. In *Proceedings of the IEEE*, volume 80, pages 1121–1132, 1992.

[12] S. N. Srihari, Y. C. Shin, V. Ramanaprasad, and D. S. Lee. Name and address block reader system for tax form processing. In *Third International Conference on Document Analysis and Recognition (ICDAR-95)*, pages 5–10, Montreal, Canada, 1995.