

# Exploiting Fisher Kernels in Decoding Severely Noisy Document Images

Jindong (JD) Chen and Yizhou Wang

Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304-1314, USA

{jindong.chen,yizhou.wang}@parc.com

## Abstract

*Decoding noisy document images is commonly needed in applications such as enterprise content management. Available OCR solutions are still not satisfactory especially on noisy images, and re-trainable systems require difficult and tedious training example preparation. Motivated by this challenging real application, we propose a novel solution that organically combines generative template models with discriminative classifiers via RBF Fisher kernel derived from a generative model. We show that the new approach is highly accurate in decoding noisy document images, making the system more generalizable to variations in font and degradation, and hence significantly reduces the burden in training example preparation. We also show that as it weights the pixel features by their relevancies, RBF Fisher kernel is more robust, and leads to smaller, faster models by dimensionality reduction.*

## 1. Introduction

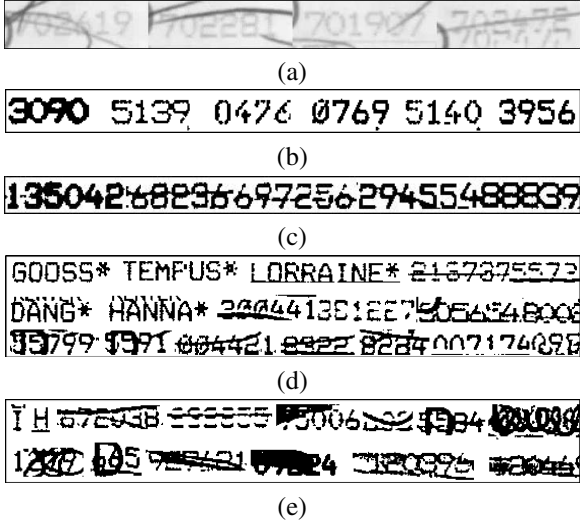
Decoding degraded document images is commonly needed in important applications, such as enterprise content management. Every day in health care, rental and other service sectors, a huge volume of NCR (no carbon required) forms are used in recording business information and need to be reliably processed. Printing on multi-sheet NCR forms requires impact printing (commonly in the form of dot-matrix printers) and often produces very noisy and low contrast document images (Fig.1). Therefore tools are in demand to handle severely low quality document images with high accuracy and efficiency. These tools also need to be able to flag unconfident items, robust enough to deal with unseen data, in some cases fast enough to be integrated into interactive solutions, and preferably able to do on-line training. As an example, we have studied a real commercial setting, in which over 120,000 forms per day are scanned and indexed for archiving, with under 1% errors. They are mostly sheets from multi-sheet NCR forms with customer information in dot-matrix printed texts, which are in low contrast and in various fonts, often crossed over by handwritten marks, or overlapped with features of the pre-

printed forms, as shown in cropped regions in Fig.1.

It is not a surprise that none of the popular commercial OCR solutions we have tested (including ABBYY FineReader, Prime OCR, and ScanSoft OmniPage) is effective enough for this application, as they focus on general character shapes, but not images with severe noise and clutter as shown in Fig.1(a). Re-trainable font-specific approaches, such as DID[7] provide the greatest accuracy when the font and degradation model is known. However, our study has shown that a bit-flip model [7, 6] is too sensitive to variations in font shape and degradation, and requires categorizing training examples into specific combinations with subtle differences, which is very difficult to do. Training example preparation in general has been recognized as “a high skilled, tedious, and thus often prohibitively expensive manual effort”[12], and has continued to be a topic for research attention in the field[11, 8, 3]. However, none of the existing techniques has effectively addressed the need of our application. We are hence motivated to search for a scheme that is less sensitive to variations in font and degradation, and therefore remove the burden of manual categorizing training examples with subtle variations. We show that our scheme is significantly better compared to pure generative solutions in accuracy, while significantly robust to unseen data and faster in performance compared to pure discriminative solutions, as the models are smaller (in feature dimensions and number of support vectors).

## 2. Fisher Kernel In Generative-Discriminative Learning

Two approaches to data classification have been extensively studied in the machine learning literature: generative methods and discriminative methods. Ng *et.al.*[9] and Holub *et.al.*[4] show that, in many applications, compared to generative models, discriminative models usually are easier to train and can achieve higher classification accuracy. However, generative models have their own irreplaceable advantages in flexibility and robustness, especially in handling complicated scenarios, as is demonstrated in [15]. [9] shows that when training data sets are small, generative approaches often have better performance than discrimi-



**Figure 1. Challenging examples of noisy document image segments in our data set, (a) in gray scale, (b) several different fonts in various degradation levels. Leftmost two groups are from the same font but different degradation levels, (c) decoded correctly by AB-BYY OCR and other methods discussed in Section 4, (d) decoded correctly by our approach, but incorrectly by other methods, (e) decoded incorrectly by both our approach and other methods.**

native approaches. The above suggests that it will be advantageous to combine the two complementary models into a hybrid framework, which is not only flexible in learning, but also has high performance in terms of prediction accuracy and computational efficiency. The literature includes several examples of work taking advantage of this idea[5, 14, 13, 10, 4]. Among them, [5, 13, 4] utilize Fisher scores obtained from a learned generative model and use it for classification and clustering purposes.

It is well known that in the context of pattern classification, likelihood value is inadequate to express data difference for classification. However, the Fisher score derived from generative models extracts richer information about the inner representation of each data item. More specifically, as the data information is summarized in the feature space, each feature  $\phi_i$  can be constructed as

$$\Phi(x, \Theta) = (\phi_i(x; \Theta))_{i=1}^n = \left( \frac{\partial \log p(x|\Theta)}{\partial \theta_i} \right)_{i=1}^n \quad (1)$$

where  $\Theta = (\theta_1, \dots, \theta_n)$  is the model parameter vector which can be obtained by maximum likelihood estimation.  $n$  is the number of parameters or, equivalently, the feature space dimension. When  $\Phi(x, \Theta)$  is evaluated at a given model  $\hat{\Theta}$ , the feature vector in Eqn.1, known as the *Fisher score*, is the gradient of the log-likelihood function at  $\hat{\Theta}$ .

Via the Fisher scores, two data items can be compared in kernel methods from the viewpoint of the generative model  $\hat{\Theta}$ .

Since the mid-1990s, kernel methods have become powerful tools in pattern analysis, and kernel-based learning approaches provide an efficient way for researchers to analyze nonlinear distributions in high dimensional feature spaces. Among them, RBF kernel is one of the most versatile ones. In this work, we are most interested in a composite RBF kernel with the Fisher score vectors, given as  $K(x_i, x_j) = \exp\{\gamma\|\phi(x_i) - \phi(x_j)\|^2\}$ . We refer to it as *RBF Fisher kernel*.

Inspired by the above work and observations, we explore a hybrid method that combines organically generative models with discriminative classifiers via Fisher kernel, derived from independent bit-flip template model. The new approach exploits the advantages of template-based generative models to achieve high accuracy, while making the system more generalizable to variations in font and degradation, and robust to unseen data. Relating it to dimensionality reduction, we discover that Fisher kernel can also leads to more compact models (in feature dimensions and number of support vectors) and faster decoding speed compared to pure discriminative methods. Recently in information retrieval, researchers also explore using Fisher kernels from topic models for dimensionality reduction[1].

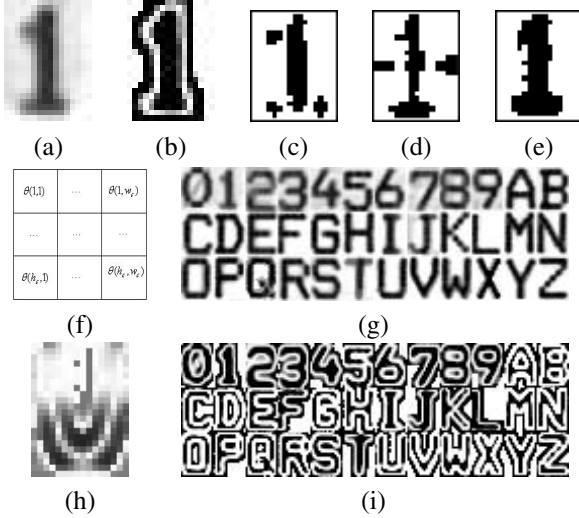
### 3. Our Hybrid Method

It has been shown that a HMM-based approach[7, 3], where character segmentation and recognition happen iteratively, is effective, if not necessary, in decoding noisy document images, and the scheme described here can be integrated as part of a HMM-based approach. However, due to page length limit, we restrict our discussion to the character level, assuming that character segmentation is given.

Our hybrid method includes a generative module and a discriminative module. Under the assumption that the document images are binary, a probabilistic template  $T_c$  is estimated for each glyph from  $m$  possible alphabets and the model parameter is denoted as  $\theta_c = \{h_c, w_c, \vartheta_c : |\{c\}| = m\}$ , where  $h_c$  and  $w_c$  are height and width of  $T_c$ , respectively, in pixels. Specifically,  $T_c$  is constructed as an independent bit-flip model as shown in Fig.2(f), in which  $\vartheta_c$  forms a two dimensional array, a component of which,  $\vartheta_c(i, j)(i \in [1, h_c], j \in [1, w_c])$ , gives the probability of the pixel at  $(i, j)$  being a black pixel in  $T_c$ , an observed image of glyph  $c$ . In other words,  $\vartheta_c(i, j)$  describes a random variable that follows a Bernoulli distribution,

$$p(I_c(i, j) = q) = \vartheta_c(i, j)^q (1 - \vartheta_c(i, j))^{1-q}, q \in \{0, 1\}.$$

The probability of a glyph image  $I_g$  given glyph  $c$  is then



**Figure 2. Probabilistic templates for glyphs, and weight maps of the RBF Fisher kernel, the darker the larger the weight. (a) template model for glyph “1”. (b) weight map for glyph “1”. (c)-(e) Samples of glyph “1”. (f) Parametric probabilistic template for glyph c ( $h$ -pixel high and  $w$ -pixel wide).  $\vartheta_c(i, j)$  gives the probability of a pixel at  $(i, j)$  being black in an observed glyph image. (g) Learned templates of all glyphs. (h) weight map of “U” vs “V” for 1-vs-1 classifiers. (i) weight maps of all glyphs for 1-vs-all classifiers.**

given as the product of the likelihood of each pixel,

$$p(I_g|c; \theta_c) = \prod_{i=1, j=1}^{h_c, w_c} \vartheta_c(i, j)^{I_g(i, j)} (1 - \vartheta_c(i, j))^{1 - I_g(i, j)}, \quad (2)$$

where  $I_g$  is white-padded if it is smaller than  $\vartheta_c$  in any dimension.

In principle, we should use a layered model that is maximum likelihood estimated[6], as discussed later. In this work, we build the template by simply counting how many pixels are on at each location among all training samples.

Eqn.2 describes a generative model, smoothly parameterized in  $\vartheta_c(i, j)$ . From Eqn.1, each component of the Fisher score of the model therefore is

$$\phi_{c, i, j}(I_g) = -\frac{I_g(i, j) - \vartheta_c(i, j)}{\vartheta_c(i, j)(1 - \vartheta_c(i, j))} \quad (3)$$

Eqn.3 intuitively shows that Fisher scores “emphasize” important information for discriminative classification. The magnitude of the mismatch is magnified greatly by either  $\frac{1}{\vartheta_c(i, j)}$  or  $\frac{1}{1 - \vartheta_c(i, j)}$ , when  $\vartheta_c(i, j)$  is close to 0 or close to 1, i.e. a pixel location that is most likely to be black or white. We concatenate the Fisher scores of all the glyph templates

into one single vector to form the feature mapping function,

$$\Phi(I_g) = (\phi_{c, i, j}(I_g))_{(c=1, i=1, j=1)}^{(m, h_c, w_c)} \quad (4)$$

Note that the size of each glyph ( $w_c, h_c$ ) can be different. We construct a RBF Fisher kernel for glyph images  $I_s$  and  $I_t$  over  $\Phi(I)$

$$K(I_s, I_t) = \exp(\gamma \|\Phi(I_s) - \Phi(I_t)\|^2) \\ = \exp\left(\gamma \sum_{c, i, j} \frac{(I_s(i, j) - I_t(i, j))^2}{\vartheta_c^2(i, j)(1 - \vartheta_c(i, j))^2}\right) \quad (5)$$

$$= \exp\left(\gamma \sum_{i, j} w^2(i, j) (I_s(i, j) - I_t(i, j))^2\right) \quad (6)$$

where

$$w(i, j) = \left(\sum_c \frac{1}{\vartheta_c^2(i, j)(1 - \vartheta_c(i, j))^2}\right)^{-\frac{1}{2}} \quad (7)$$

From Eqn.5 to 6, all the  $\vartheta_c(i, j)$  terms are collected into a *weight map*  $w(i, j)$  as in Eqn.7. Each template  $\vartheta_c$  is padded (with .5) to the size of the union of all templates (aligned at the upper left corner). The dimension of the feature space is the same as that of a template, regardless of the number of templates. Again, through the weight map, the kernel “emphasizes” the differences of two data items in locations where the pixel values are highly definitive.

Rewriting the RBF Fisher kernel as standard RBF kernel,  $K(I_s, I_t) = \exp\{\gamma \|\hat{I}_s - \hat{I}_t\|^2\}$ , where  $\hat{I}(i, j) = w(i, j)I(i, j)$ , we can use standard software to learn a classifier with little extension in the Fisher kernel induced feature space. In our prototype, we use LibSVM[2] to learn a support vector machine (SVM) model.  $\gamma$  is chosen to be .0019 by the parameter search tool provided in [2]. Depending on the application, we use either the “one vs one” or “one vs all” strategy, or a combination of them, for multi-class classification. We favor the “one vs all” strategy as it is allowed to classify a sample as “unknown class”, which is necessary for on-line training. In applications described in Section 1, training data often does not cover all possibilities.

To discriminate between two classes, we further regulate the weight map by the magnitudes of the differences between their templates, as follows.

$$w(i, j) = \left(\sum_{c=0,1} \frac{(\vartheta_0(i, j) - \vartheta_1(i, j))^2}{\vartheta_c^2(i, j)(1 - \vartheta_c(i, j))^2}\right)^{-\frac{1}{2}} \quad (8)$$

Note that  $w(i, j)$  may be very high if  $\vartheta(i, j)$  is very close to either 0 or 1. And a few entries with spiking values may dominate in kernel evaluation. We hence need to regulate the template, preferably with a layered model that is maximum likelihood estimated[6], which can be obtained by EM training. In this work, we use heuristic techniques to regulate the weight map (clipping high spikes). We then normalize the weights to [0,1].

We further exploit the robustness of the generative models by restricting the discriminating inference to a subset of classes that have high enough likelihood scores. Experiments have shown that this treatment increases computational efficiency dramatically. On average, we can infer on a subset of classes that is 60% of the full set, with no trade-off in accuracy. We have also tried using the same treatment to increase accuracy by excluding labels that are low in likelihood. But the benefit is negligible, which shows that the Fisher kernel exploits the desirable properties of the generative models.

**Dimensionality Reductions.** In supervised learning, for the purposes of performances in generalization (robust to new data), speed and space, dimensionality reduction is about selecting relevant feature dimensions, or making “hard” decisions to assign the weights of relevant feature dimensions to 1 and the rest to 0. In this view, RBF Fisher kernel can be regarded as “soft” dimensionality reduction, where each feature dimension is weighted by its relevancy  $w(i, j) \in [0.0, 1.0]$ , suggesting that it improves robustness as does “hard” or regular dimensionality reduction. To reduce of time and space, however, we perform “hard” dimensionality reduction by setting to zero the  $w(i, j)$ ’s that are small enough, as the contribution of the feature value at  $(i, j)$  is negligible.

In this work, we set to zero all the values below the medium or the mean (.5) but not larger than .75. Fig. 2 (b), (h) and (i) illustrate the weight maps. Each weight map in (b) (and (i)) consists of a dark “skeleton” glyph, surrounded by a light ring, which is in turn surrounded by a dark ring, and the rest are light. The dark “skeleton” and the dark ring capture the dimensions that are important to identify the glyph (black in the skeleton, white in the dark ring). The light ring indicates that those dimensions are not important as shape of the glyph varies. And the outer light area is not important as they are far away from the glyph. Fig. 2 (h) shows the bottom half of the weight map is more relevant as that is where “U” and “V” differs in shape.

Roughly speaking, the complexity of an SVM model can be measured by the number of nonzero entries of the all the support vectors, the fewer the non-zero entries, the faster the prediction speed, and the smaller the memory footprint. our experiments show that our models have significantly less complexity compared to the pure discriminative models that we benchmark with (Tables 1 and 2). We have not yet had an implementation that is suitable for speed comparison.

## 4. Experimental Results and Discussions

The data are collected from the real commercial application described in Section 1. The alphabet set includes 10 digits and 26 capital characters. The authors are able to tell that there are at least 5 fonts with roughly the same size

Method	Error	Dim	Complexity
GenDis (1-vs-all)	1.4%	135	369K
Pixel SVM (1-vs-all)	1.6%	352	693K
GenDis (1-vs-1)	3.2%	141	330K
Pixel SVM (1-vs-1)	3.7%	352	785K
SVM on Likelihood	9.2%	N/A	N/A
Likelihood	11.1%	N/A	N/A
ABBYY OCR 7.0	42.5%	N/A	N/A

**Table 1. Classification accuracy comparison among several methods. The two 1-vs-all methods only handle 87.7% of the data and classify the rest as new classes. The numbers under the “Dim” column are the average numbers of dimension in the models. The numbers under the “Complexity” column are the numbers of the non-zero entries of all support vectors.**

( $16 \times 22$  pixels), and within the same font, the glyphs are in a wide variety of degradation levels (Fig.1(b)). The training set is picked randomly from the documents. It contains 20,834 digits, 12,617 letters, and 2,155 symbols. The testing set has 18,000 glyphs independent of the training set, which consists of a larger percentage of challenging examples, chosen by visual inspection.

We have not been able to identify a publicly available data set that is noisy enough to demonstrate the strength of our approach. We have published part of the data set on our website: (<http://www.parc.com/pda>).

We then compare our approach (GenDis) with the following alternatives.

1. Pixel SVM (Discriminative): Perform classification on glyph images in pixels.
2. Likelihood SVM (Hybrid): Use the vector of the likelihoods of the glyph templates,  $\Phi_c^{[M]} = (\log p(I_g | c; \theta_c))_{c=1}^m$ , as the feature vector for SVM classification.
3. Likelihood(Generative): The predicted class is the one with the largest likelihood score from the learned generative glyph templates.
4. ABBYY OCR 7.0: Decoded by ABBYY OCR 7.0. From our evaluation, ABBYY is of one of the best commercial solutions in decoding noisy document images (see also Fig.1(c)).

We have conducted two tests. The first one measures accuracy in decoding all 36 classes, as reported in Table 1. The second one measures robustness against new classes as reported in Table 2. In the second test, “one vs all” models are trained for the 10 digits, and are used to decode the test

Method	Error	Dim	Complexity
GenDis	8.75%	164	161K
Pixel SVM	20.07%	352	230K

**Table 2. Robustness comparison among GenDis and Pixel SVM. Both decode upper case letters with models built for digits. The columns are organized the same way as those in Table 1.**

Truth	'A'	'D'	'G'	'I'	'O'	'S'	'T'
Predicted	'4'	'0'	'8'	'1'	'0'	'8'	'1'
GenDis	0	20	17	2	65	35	10
Pixel SVM	87	30	19	59	33	58	83

**Table 3. In columns, comparisons of the numbers of the dominative errors in the robustness test where new classes (upper case letters) are mistaken as digits (known classes).**

data of new classes of 26 upper case letters. So it is an error if a test sample is decoded as a digit. We measure the complexity of each SVM model in both tests.

From the tests we can draw the following conclusions:

1. GenDis achieves high decoding accuracy, similar to Pixel SVM, and significantly higher compared to the other methods, including the generative method.
2. Compared to Pixel SVM, GenDis is more robust to noise, clutter and font variations as shown in Fig.1, and also more robust to new classes.
3. The Fisher score is a better feature not only indicating the generative process, but also encoding structure/shape information of the data items.

Table 3 shows the dominative confusion pairs from the robustness test (Table 2). We find that as they are somewhat similar to each other (see their templates in Fig.2), the Pixel SVM trained for only the digits can not distinguish them very well. GenDis, as the discriminative training is guided by the Fisher kernel based weight map, captures the structures of the digits, and is less likely (except for "O" & "0") to mistake a sample of a new class as one it is trained for.

## 5. Summary

In summary, we have presented a hybrid method that exploits the advantages of generative models and discriminative models via Fisher kernel for decoding severely noisy document images. We have also shown that the dimension of the feature space can be dramatically reduced to improve robustness and efficiency. Experiments have shown that our approach is significantly better than a method that uses a

generative classifier or a discriminative classifier alone, or a naive combination of the two, such as in experiment 2. We have also shown that our approach is generalizable enough to variations in font and degradation, and training example preparation becomes much easier by employing our method. The approach is not only attractive to enterprise document management, but also attractive to mobile camera applications, where resources (memory and speed) are limited and image quality is usually low.

## Acknowledgments

We thank our colleagues Prateek Sarkar, Eric Saund and Jing Lin for numerous insightful discussions and generous helps on proof readings, and Darren Schroeder and Randy Gill of Xerox for providing us the data.

## References

- [1] G. Chandalia and M. Beal (2006), Using Fisher Kernels from Topic Models for Dimensionality Reduction, *NIPS 2006 Workshop: Novel Applications of Dimensionality Reduction*
- [2] C-C Chang and C-J Lin, LIBSVM: a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [3] J. Edwards and D. Forsyth (2005) Searching for Character Models, *Neural Info. Proc. Systems (NIPS)*.
- [4] A.D. Holub, M. Welling and P. Perona (2005) Combining generative models and Fisher kernels for object recognition, *Int'l Conf. on Computer Vision (ICCV)*, 1, 664-671.
- [5] T.S. Jaakkola and D. Haussler (1998) Exploiting generative models in discriminative classifiers, *NIPS*.
- [6] G.E. Kopec (1997) Multilevel character templates for document image decoding, *Document Recognition IV, SPIE 3027*.
- [7] G.E. Kopec and A. Chou (1994) Document image decoding using Markov source models, *IEEE Tran. on Pattern Analysis and Machine Intelligence (PAMI)* 16.6, 602-617.
- [8] H. Ma and D. Doermann (2005) Adaptive OCR with limited user feedback, *Int'l Conf. on Document Analysis and Recognition (ICDAR)*, 814-818.
- [9] A.Y. Ng and M. Jordan (2001) On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes, *NIPS*.
- [10] R. Raina, Y. Shen, A.Y. Ng and A. McCallum (2003) Classification with hybrid generative/discriminative models, *NIPS*.
- [11] P. Sarkar and H. S. Baird (2004) Decoder banks: versatility, automation, and high accuracy without supervised training, *Int'l Conf. on Pattern Recognition (ICPR)*, 2, 646-649.
- [12] P. Sarkar, H. S. Baird and X. Zhang (2003) Training on severely degraded text-line images. *ICDAR*, 38-43.
- [13] K. Tsuda, M. Kawanabe and K-R Muller (2002) Clustering with the Fisher score, *NIPS*.
- [14] S. Tong and D. Koller (2000) Restricted Bayes Optimal Classifiers, *National Conference on Artificial Intelligence (AAAI)*.
- [15] M. Weber, M. Welling and P. Perona (2000) Towards automatic discovery of object categories, *Computer Vision and Pattern Recognition (CVPR)*, 2101.