Enhancing Word Image Retrieval in Presence of Font Variations

Viresh Ranjan[1] Gaurav Harit[2] C. V. Jawahar[1]

[1]CVIT, IIIT Hyderabad, India [2]IIT Jodhpur, India

***Abstract*—This paper investigates the problem of cross document image retrieval, i.e. use of query images from one style (say font) to perform retrieval from a collection which is in a different style (say a different set of books). We present two approaches to tackle this problem. We propose an effective style independent retrieval scheme using a nonlinear style-content separation model. We also propose a semi-supervised style transfer strategy to *expand* the query into multiple styles. We validate both these approaches on a collection of word images which vary in fonts/styles.**

I. Introduction

Font and style variations make the problem of recognition and retrieval challenging while working with large and diverse document image databases. Commonly, a classifier is trained with a certain set of fonts available *apriori*, and generalization across fonts is hoped due to either the quality of the features or the power of the classifier. However, in practice, these solutions give degraded performance when used on *target* documents with a new font. If the entire target dataset is available at the time of training, then it is possible to learn a classifier [1] which could work on several fonts. If the details of the fonts in the database are known, one could render the textual queries in each of these fonts and retrieve from the database [1]. In some cases, a style clustering [2], [3] is done and then separate classifiers are learnt for each of the style clusters. In this work, we are interested in an effective retrieval solution, where the query is a word image, and the database has an unknown set of fonts. We formulate the retrieval problem in a nearest neighbor setting. In this setting, the distance for finding nearest neighbors can be Euclidean [4] or the cost of alignment of two feature vector sequences with a Dynamic Time Warping (DTW) [5].

If the query is a word image, then we need to *transfer* or *expand* the query into multiple fonts. Query expansion, which is a technique for reformulating a seed query, is a common practice in information retrieval. In query expansion, a seed query is reformulated by also taking into account semantically and morphologically related words. A natural extension of the query expansion in cross document word image retrieval could be to automatically reformulate the query word in multiple fonts. In this paper, we propose a query reformulation strategy which builds up on this very idea. To motivate the challenges in cross document retrieval, we conduct an experiment on words rendered in two different fonts. We argue that the distance between the two feature vector representation could become ineffective in presence of font variations. In Figure 1, we present the Euclidean distance between profile feature representations of different words in the same font, as well as the same word in different fonts. Smaller inter-class distance and larger intraclass distance lead to many false positives and poorer retrieval. This shows that font variation could be a crucial factor while performing cross document word image retrieval (see more in Sec. II).

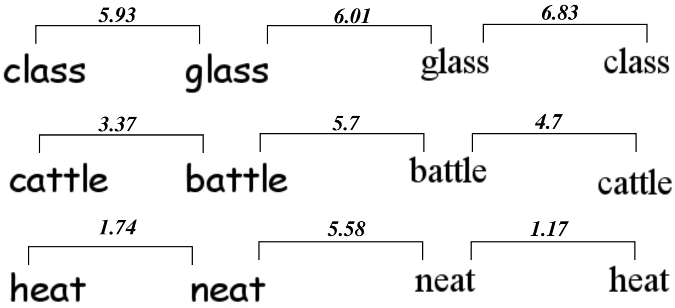

Fig. 1. Euclidean distance between profile feature representation of pairs of word images. Euclidean distance could be affected more by font variation than a difference in underlying word labels, for example, distance between “battle” in the two fonts is more than “battle” and “cattle” in the same font.

Many efficient approaches for word image retrieval has been proposed in the recent past. Rath and Manmatha [5], as well as Meshesha and Jawahar [6] use a profile based representation along with DTW based retrieval. In many of the recent works, either DTW or Euclidean distance is used. Euclidean distance is often preferred for scalability in retrieval [7]. These approaches primarily depend upon training data in order to handle font variations and may not generalize well in case of previously unseen fonts.

If the target style is not known *apriori* but certain samples (labeled or unlabeled) of the target dataset are known, then it is possible to transfer (adapt) the classifiers learned on the training data so that they are able to handle the new style of the target dataset. This technique is known as transfer learning [8], and it has been widely used in applications like handwriting recognition [2], [9], face pose classification [10] etc. Transfer learning may involve (i) Feature transformations, e.g. updating the regression matrix [11], updating the LDA transformation matrix [12] (ii) Classifier adaptation, e.g. Retraining strategy for neural network [13], SVM [14], etc. The adaptation process needs to be unsupervised if labeled data from the target dataset is not available. The classifier would then need to use some suitable self-learning strategy [15], [16] to learn the style context in a group of patterns.

The objective of this work is to perform word image retrieval from a collection of books/documents, where the query word image could be in a different style from those

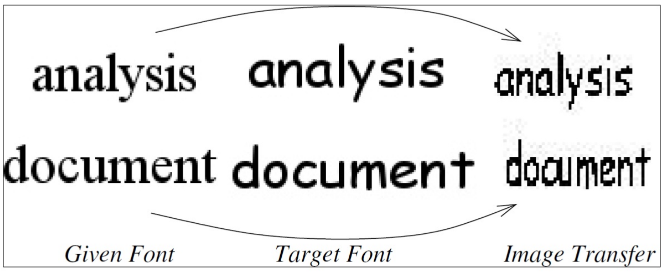

Fig. 2. Application of bilinear model for transferring image from one font to another is shown. Content vectors corresponding to word images in the first font are transferred to the second font using style vectors of the second font.

in the database. Our primary contributions are the following:

1) Effective retrieval from multi-font database is formulated as an automatic query expansion with no human intervention or labeled examples.
2) A nonlinear style-content factorization scheme is proposed. The method is compatible with the popular document retrieval schemes (e.g. those which use some appearance features with a distance based retrieval) and can improve their performance at minimal computational overhead.
3) We validate the method on real data sets with font variations and report qualitative and quantitative results. To analyze the solution better, we also build a dataset in a laboratory setting.

II. Direct approaches

A common approach to deal with font variations is to heuristically define and extract features. Then one empirically validates the insensitivity to feature variations on multiple fonts. For addressing font style variations in word image retrieval, a common strategy is to use some font independent feature representation. Profile based representation [5], [17] is one such popular feature. Profile features are considered to be reasonably robust to font variations (however see Figure 1). It works well in the presence of a single or a limited set of fonts. Use of a DTW based sequence alignment further improves the robustness of retrieval as DTW is able to take care of local variations in sequences. Manmatha and Rath [5] use a profile based representation and DTW based alignment for retrieval on a dataset with some amount of variation in writing styles. However, such an approach may not scale-up to large multi-font databases because of large font variations and high computational cost. Another possible approach for handling font variations is to reformulate the query word image in the target document font. This strategy is discussed in Sec. II-A.

A. Style Transfer

Style transfer strategy has been used in the past for handwriting recognition. Connell and Jain [2] do a general to specific adaptation of their model using few examples of handwritten words from each user. This results in a specific model for each user. Zhang and Liu [9] address writer adaptation by learning a style transfer matrix for each user which projects word samples of each user to a style free space where a style independent classifier is used for classification. A straightforward method to do style transfer of the query is to decompose it into style and content factors using a bilinear model [10]. The style factor can then be modulated separately to make it similar to that of the target document.

Our hypothesis is that a style-transformed query would be more closer to the correct matches and would lead to a better performance of the nearest neighbor classifier. Following the asymmetric bilinear model in [10], we represent the query observation y^{sc}, in style s and content c, as

$$y^{sc} = A^s b^c, \tag{1}$$

where A^s is the set of style dependent basis vectors, b^c is the content vector depicting the underlying word label. If the set of style vectors A^s and A^t pertaining to style s and t respectively are known, a word image y^{sc} can be transferred from style s to the new style t by first finding the content vector b^c corresponding to the word image and then using the style basis vectors A^t as $y^{tc} = A^t b^c$. We show such style transfer examples in Figure 2. The transfer does not look to be visually impressive due to the nature (binary) of the image. In addition, a serious limitation of using this style transfer approach in large multi-font databases is the need for some labeled examples of all the distinct words in the database for each of the fonts. In other words, this approach cannot effectively generalize to previously unseen fonts.

III. Query expansion using semi-supervised style transfer

In the retrieval setting, we have a single example (query) to transfer the style. We modify the reformulation strategy discussed in Sec. II-A so that minimal amount of labeled data is required for the style transfer. We propose a semisupervised style transfer strategy for reformulating the query word image into target fonts without using any target labels. This strategy uses labeled data only from a single font, learns a bilinear model over it and adapts the bilinear model to any target dataset in an unsupervised manner. This strategy saves us from the costly practice of obtaining labeled word images corresponding to every different font in the database. The reformulation strategy used here is akin to the query expansion strategy used in information retrieval. An initial seed image is reformulated into multiple versions and all versions have in common the underlying word label.

Given a set of word image observations for different word labels arranged as column vectors in matrix Y^s(each column corresponds to average of all the images of a particular word label), basis vectors A^s and content vectors B^c(each column is a content vector corresponding to a word label) can be obtained by solving the following optimization problem

$$\min_{A^s, B^c} ||Y^s - A^s B^c||_F^2 \,. \tag{2}$$

If the same number of word images are available for all the word labels, this problem can be solved with the help of SVD of the matrix Y^s.

Consider the task of rendering word images in a new font using the asymmetric bilinear model. We learn the model parameters (A^s, B^c) from the training dataset of word images. To transfer the content vectors in B^c to any desired style r, a few labeled examples Y^r from the target dataset in style r

can be used to adapt A^s to obtain A^r by solving the following optimization problem

$$\min_{A^r} ||Y^r - A^r B^r||_F^2 + \lambda \, ||A^r - A^s||_F^2 \, . \qquad (3)$$

Here, columns of B^r are a subset of the columns of B^c. Using the original pixel based representation of word images for performing style transfer has a few shortcomings. We believe that image transfer is a difficult task because of the high dimensionality of the image space. The bilinear model may overfit the training images, and may not generalize well to the word images and fonts which are not there in the training dataset. Also, there is a high computational cost associated with the SVD of a large matrix. Therefore we prefer a low dimensional feature space. In this work we use a profile feature [5] based representation of word images and perform transfer and retrieval in the feature space. Using a low dimensional profile feature representation reduces the computation required for model learning as well as retrieval.

Consider the same number of word images for each of the N word classes, where each class corresponds to the different underlying word label. We represent each word image by its profile feature representation (Section V) and stack the mean vector for each word label along the column of matrix Y^t. We obtain the font dependent basis vectors A^t and a matrix of content vectors B^t by doing SVD of Y^t. The i^{th} column of Y^t corresponding to the mean vector of i^{th} word label can be represented using asymmetric bilinear model as $y^{it} = A^t b^{it}$, where b^{it} is the i^{th} column of B^t and it is content vector for the i^{th} word label. Since a content vector b^{it} is independent of the style, it is possible to transfer b^{it} to the target dataset font if we have the style dependent basis vectors (A^r) for the target dataset font. Mean vector for i^{th} word label can be obtained in target dataset font using Equation 1.

Our method, outlined below, does not require labeled data from the target dataset.

1) Learn bilinear model A^t, B^t from labeled training dataset.
2) Propagate the labels corresponding to the word images in the training dataset to the word images in the target dataset by doing a nearest neighbor search over it. Say we propagate the labels for M word labels.
3) We assign labels to only the top few results of the nearest neighbor search. Therefore we get labeled examples corresponding to M word labels such that these M labels are a subset of the N training dataset labels.
4) We then form the content vector matrix B^r using the content vectors from B^t which correspond to the labels assigned in the previous step.
5) We use Equation 3 to obtain A^r.
6) Once we have obtained A^r, we use Equation 1 to obtain a feature vector representation of the word images in the target dataset font. These vectors can now be used to perform nearest neighbor based retrieval over the target dataset.

The asymmetric bilinear model, which we use here for style transfer, is a linear model and hence it cannot capture the nonlinearities in the data. Also, this strategy requires retraining for each new target font. In next section, we introduce our nonlinear style-content factorization model which takes care of these issues.

IV. Kernalized Style-content separation

To make linear models more robust, it is a common practice to first map the feature vectors in the original space to a high dimensional space and then learn the linear model over the high dimensional space. If a feature vector in this high dimensional space is some nonlinear function of the corresponding vector in original space, then a linear model in this space will correspond to a nonlinear model in original space.

Let ϕ be a mapping such that $\phi : R^n \rightarrow H$ where R^n is original observation space and H is a Reproducing Kernel Hilbert Space (RKHS) which could have a very high dimensionality in comparison to R^n. The feature map ϕ could be a nonlinear mapping. If any algorithm can be expressed solely in terms of dot products of feature points in H, then we do not need to know the exact mapping ϕ and a kernel function κ can be defined such that $\kappa(x, y) =< \phi(x), \phi(y) >$, where $x, y \in R^n$ and κ corresponds to some mapping ϕ [18]. This technique is known as the kernel trick and has been widely used for obtaining nonlinear versions of PCA [18], LDA [19] and many other algorithms.

We call our nonlinear version of bilinear model as asymmetric kernel bilinear model (AKBM). In order to obtain nonlinear version of the bilinear model, we first define the following terms. Let Y^t be the matrix containing mean vectors of different word classes along its columns, ϕ be the feature map, B^t be the content vectors corresponding to different word labels and A^t be the set of style dependent basis vectors in the high dimensional feature space. Any observation y^{tc} corresponding to style t and label c can be represented in the feature space as

$$\phi(y^{tc}) = A^t b^c. \qquad (4)$$

To obtain style basis vectors A^t and content vectors B^t, we solve the following optimization problem

$$\min_{A^t, B^t} \left|\left|\phi(Y^t) - A^t B^t\right|\right|^2 + \beta \text{Trace}(A^{t^T} A^t). \qquad (5)$$

Here the first term is the data fitting term and second term is the regularizer which controls overfitting. Since style basis vectors lie in the same feature space as the observation vectors, each basis vector (each column of A^t) can be expressed as a linear combination of the mapped observation vectors, hence A^t can be represented as: $A^t = \phi(Y^t)\alpha$.

Using these, the above optimization problem can be rewritten as

$$\min_{\alpha, B^t} \mathcal{K} - B^{t^T} \alpha^T \mathcal{K} - \mathcal{K} \alpha B^t + B^{t^T} \alpha^T \mathcal{K} \alpha B^t + \beta \text{Trace}(\alpha^T \mathcal{K} \alpha). \quad (6)$$

This problem is convex in α if B^t is kept constant and vice-versa. We solve this optimization problem by alternately keeping one of the two factors as constant and optimizing for the other factor. Any standard QP solver [20], [21] can be used for solving this optimization problem.

To learn the nonlinear model from the available profile feature representation of training dataset word images, we solve the optimization problem given in 6. This gives us the coefficient matrix α and the content matrix B^t. Any observation in the feature space can now be represented as $\phi(y^{tc}) = \phi(Y^t)\alpha b^c$.

Dataset	# Distinct Words	#images
D1	200	19472
D2	200	4923
D3	200	8463
D4	200	13557
D5	200	2868
Dlab	500	5000

TABLE I. DATASET: TABLE GIVES INFORMATION ABOUT DIFFERENT DATASETS USED IN OUR EXPERIMENTS. D5 HAS A VERY DIFFERENT FONT IN COMPARISON TO D1 - D4. DLAB CONSISTS OF WORD IMAGES RENDERED IN 10 DIFFERENT FONTS.

Now, to use these nonlinear basis vectors to perform retrieval on the target dataset, we represent all the word images from the target dataset by solving $\min_{b^{ir}} ||\phi(y^{ir}) - \phi(Y^t)\alpha b^{ir}||^2$, where y^{ir} is the profile feature representation of i^{th} image from target dataset. We use the closed form expression of this problem and obtain the content vectors corresponding to all the images from the target dataset. Now the retrieval is performed on target dataset on the basis of distance between the content vector of query word images and content vector of target dataset word images.

Since the nonlinear model is more robust, the basis vectors computed from the training dataset can represent word image features from the target dataset also. Hence, we need not adapt the nonlinear model using word images from the target dataset.

V. EXPERIMENTS, RESULTS AND DISCUSSIONS

In this section, we compare the retrieval performance for the following three cases:

1) Query word images from training dataset are used directly to perform retrieval on target dataset (i.e. font independent feature definitions).
2) Semi-supervised style transfer as discussed in Sec. III.
3) Asymmetric kernel bilinear model as discussed in Sec. IV.

A. Data Sets, Implementation and Evaluation Protocol

To validate the performance of our approaches, we create datasets D1 - D5 comprising of five books varying in font. These datasets, detailed in Table I, comprise scanned English books from a digital library collection. We manually created the ground truth at word level for the quantitative evaluation of our proposed retrieval approaches. Each of the datasets D1 - D5 are subdivided into training, testing and validation sets, with each set containing one-third of word images for each word label. Apart from these datasets obtained from scanned books, we also create a multifont dataset Dlab by rendering 500 words in 10 different fonts. Few of the example images from this dataset has been shown in Fig 3. Bilinear models are learned from the examples in training set. Optimal value for kernel parameters and the regularization factors β and λ are found by performing retrieval on the validation set and these optimal parameters are then used while performing retrieval on the test set. We use RBF kernel for our experiments. The kernel function κ is defined as $\kappa(x_i, x_j) = exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$ where σ is the bandwidth of RBF kernel. For each word image in the dataset we extract the profile features [5] comprising of:

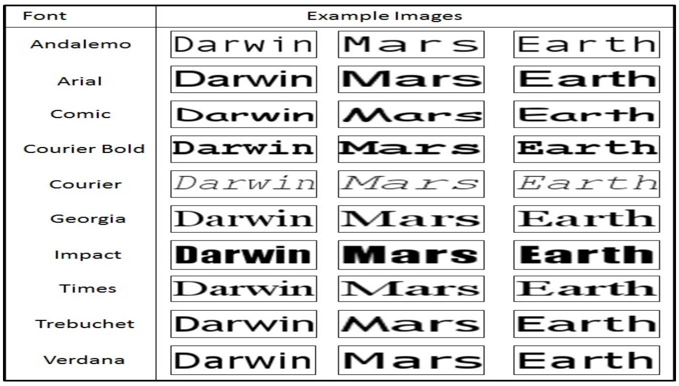

Fig. 3. Examples from each of the 10 fonts used in the Dlab.

1) Vertical projection profile, which counts the number of ink pixels in each column.
2) Upper and lower word profile, which encode the distance between the top (lower) boundary and the top-most (lower-most) ink pixels in each column.
3) Background/Ink transition which counts the number of background to ink transitions in each column.

B. Retrieval Experiments

In Table II, we compare the retrieval performance of font independent feature definitions (no transfer), semi-supervised style transfer (SSST) and asymmetric kernel bilinear model (AKBM). D1 - D4 are used for this set of experiments. 100 query word images are picked from the training dataset and retrieval is performed on the target dataset. Results are reported as the mAP values for these 100 queries. For SSST, we use asymmetric bilinear model for font transfer of query words from training dataset font to target dataset font. We learn asymmetric bilinear model using word images corresponding to 100 different word labels from training dataset. Then we do a nearest neighbor based search over the target dataset to find images similar to query words form training dataset. We assign the label of corresponding query word to the top retrieved results and use them to adapt the model. Using this updated bilinear model, we obtain feature vectors for the 100 word labels and use it for performing nearest neighbor based retrieval on the target dataset. For AKBM, we learn asymmetric kernel bilinear model using word images corresponding to 100 different word labels from training dataset. Using this kernel bilinear model, we obtain content vector representation for all of the target dataset word images and use them to perform nearest neighbor based retrieval on the basis of their distance with the content vectors corresponding to query labels from the training dataset. We observe that in majority of the cases, kernel based retrieval shows much better retrieval performance than the other two cases. It is able to achieve mAP gain of up to 0.33 over the no transfer case. In Figure 4, we show the Precision-Recall (PR) curves corresponding to 100 queries. For this experiment, two datasets are picked from D1 - D4 and used as training and target datasets. No transfer, AKBM and SSST cases are compared in the figure. Out of the three methods, AKBM has the maximum area under the PR curve, followed by SSST and no transfer case. In Figure 5, we show few query

Method	Training-Target dataset															
	D1,D1	D2,D1	D3,D1	D4,D1	D1,D2	D2,D2	D3,D2	D4,D2	D1,D3	D2,D3	D3,D3	D4,D3	D1,D4	D2,D4	D3,D4	D4,D4
No Transfer	0.97	0.69	**0.78**	0.55	0.63	0.81	**0.83**	0.63	0.55	0.68	**0.99**	0.85	0.68	0.76	0.92	0.82
SSST	**0.99**	0.71	0.64	0.74	0.67	0.91	0.75	0.81	0.59	0.76	0.95	0.84	0.70	0.83	0.89	0.91
AKBM	**0.99**	**0.85**	0.69	**0.88**	**0.88**	**0.94**	0.79	**0.92**	**0.72**	**0.83**	0.97	**0.95**	**0.84**	**0.91**	**0.96**	**0.99**

TABLE II. SHOWS THE MAP VALUES FOR 100 QUERIES WHEN USING NO TRANSFER, SSST AND AKBM. IN TRAINING-TARGET PAIR (D1, D2), D1 IS TRAINING DATASET AND D2 IS TARGET DATASET.

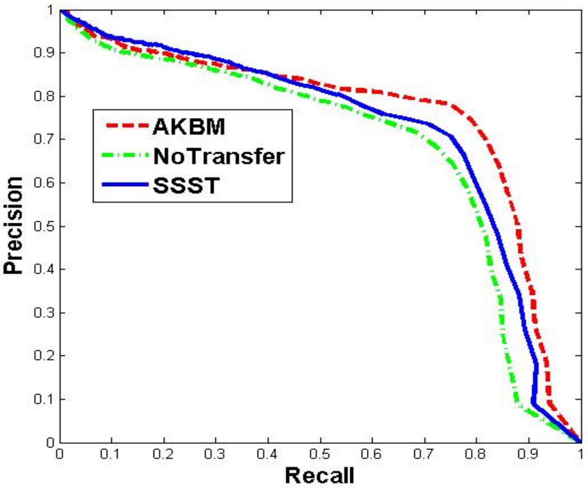

Fig. 4. Precision- Recall (PR) curves corresponding to 100 queries is given. For training and target dataset, two datasets are picked from D1 - D4.

Training dataset	mAP values over 100 queries		
	No Transfer	SSST	AKBM
D1	0.52	0.57	**0.84**
D2	0.43	0.47	**0.66**
D3	0.32	0.38	**0.52**
D4	0.44	0.52	**0.68**

TABLE III. RETRIEVAL PERFORMANCE ON D5.

images and the corresponding retrieval results, on D1 - D4, obtained using AKBM. The experiment is done in a multi-font scenario, i.e. one of the datasets is chosen for training, and retrieval is performed on dataset obtained by combining multiple datasets (D1 - D4). We also show retrieved results corresponding to a failure case in the last row which shows that visually similar words may sometimes create confusion while retrieval. We conduct another set of retrieval experiments where we test our proposed approach in case of large font variations between the training dataset and target dataset. We perform retrieval on D5 while training on one of the datasets D1 to D4 every time. We report the results in Table III. In this experiment, since the training and target fonts are too dissimilar, retrieval performance of all three approaches goes down, however, the performance of AKBM is still much better than the other two approaches. Thus, the kernelized version of the bilinear model is able to achieve font independence and improved mAP scores by up to 0.30 for word image retrieval.

Training dataset	Test dataset	mAP values over 100 queries	
		Semi-supervised Transfer	Supervised Transfer
D1	D2	0.67	0.68
D1	D3	0.59	0.59
D1	D4	0.70	0.70
D2	D1	0.71	0.69
D2	D3	0.76	0.74
D2	D4	0.83	0.82
D3	D1	0.64	0.63
D3	D2	0.75	0.76
D3	D4	0.89	0.89
D4	D1	0.74	0.74
D4	D2	0.81	0.81
D4	D3	0.84	0.85

TABLE IV. COMPARISON BETWEEN SEMISUPERVISED STYLE TRANSFER (SSST) AND SUPERVISED TRANSFER.

In Table IV we compare the semi-supervised style transfer strategy (SSST) with supervised style transfer. For doing supervised style transfer using Equation 3, we use a single labeled example per word class from the target domain instead of doing nearest neighbor based label propagation. SSST performs comparably to the supervised style transfer in this case. However, further increasing the labeled examples from the target dataset will result in improvement for the supervised case.

We also conduct an experiment on the dataset Dlab to observe retrieval performance of AKBM in presence of multiple widely varying fonts in the target dataset. Results of the experiment are given in Fig 6. For the retrieval experiment, query image is picked from one of the fonts and retrieval is performed on all the remaining fonts. For the baseline in this experiment, we directly use the query image for retrieval on the target fonts. Results are reported as average mAP values along with corresponding standard deviation for 10 runs, taking each of the fonts as source font once. As the number of the target fonts is increased, the retrieval performance of AKBM as well as the baseline decreases, however, AKBM outperforms the baseline in all the cases. The large values for the standard deviations can be attributed to the large font variations.

Results show that among the different approaches considered for handling cross-font and multi-font retrieval, our kernel based AKBM gives the best retrieval performance in the majority of cases. Superiority of this approach over the style-transfer approach could be attributed to the fact that style-content separation of word images is a complex task and using a linear model for this task may be rather restrictive.

VI. CONCLUSION

In this work, we have proposed strategies for doing word image retrieval in a multi-font database. To deal with the style variations between different documents, we have proposed a semi-supervised style transfer strategy. We have also suggested a font independent retrieval strategy by representing words

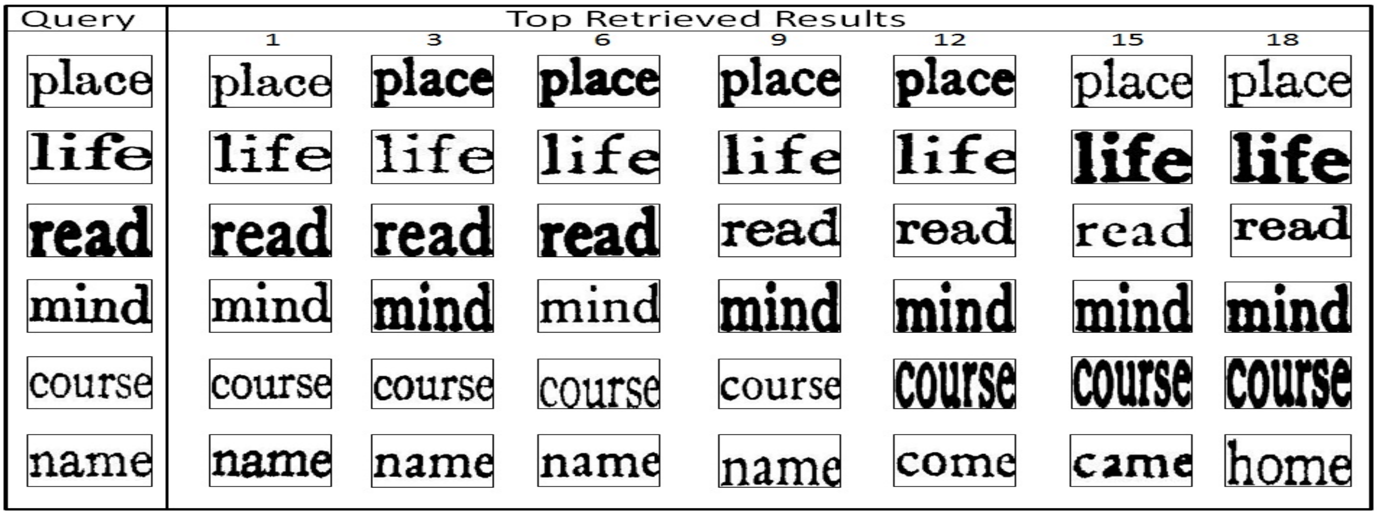
Query
Top Retrieved Results
1
3
6
9
12
15
18
place place place place place place place place
life life life life life life life life
read read read read read read read read
mind mind mind mind mind mind mind mind
course course course course course course course course
name name name name name come came home

Fig. 5. Retrieval results using AKBM on combined multi-font dataset comprising D1 - D4. The retrieval results are organized into columns showing images at rank 1, 3, 6, 9, 12, 15 and 18.

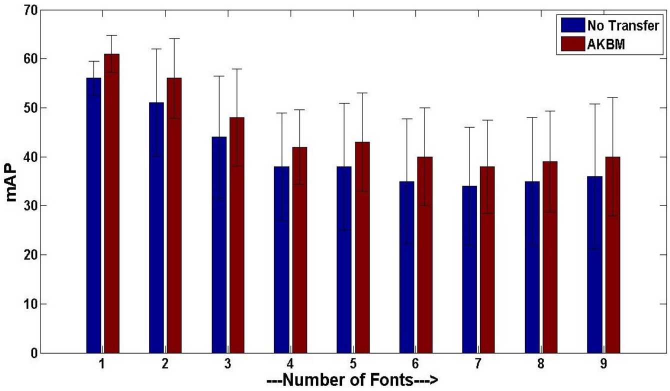

Fig. 6. Comparison of AKBM with no transfer case is given for the multifont dataset Dlab. Performance is given as mAP values with increasing number of fonts used in the target dataset. The rectangular bars represent the average mAP values for 10 runs and the error bars show the corresponding standard deviations.

from all the documents using the same set of high dimensional basis vectors. We have shown results on various datasets varying in font. Our future work will be to learn the font/style independent features from a large collection of document images.

References

[1] S. Marinai, E. Marino, G. Soda, Font adaptive word indexing of modern printed documents, in: IEEE Trans. Pattern Anal. Mach. Intell., 2006.

[2] S. D. Connell, A. K. Jain, Writer adaptation for online handwriting recognition, in: IEEE Trans. Pattern Analysis and Machine Intelligence, 2002.

[3] M. Szummer, C. M. Bishop, Discriminative Writer Adaptation, in: Proc. IEEE Int'l Workshop Frontiers in Handwriting Recognition (ICFHR), 2006.

[4] R. Jain, C. V. Jawahar, Towards more effective distance functions for word image matching, in: Proc. IAPR Workshop Document Analysis Systems (DAS), 2010.

[5] T. M. Rath, R. Manmatha, Word image matching using dynamic time warping, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2003.

[6] M. Meshesha, C. V. Jawahar, Matching word images for content-based retrieval from printed document images, in: International Journal of Document Analysis and Recognition (IJDAR), 2008.

[7] A. Kumar, C. V. Jawahar, R. Manmatha, Efficient search in document image collections, in: ACCV, 2007.

[8] S. J. Pan, Q. Yang, A survey on transfer learning, in: IEEE Trans. Knowledge and Data Engineering, 2010.

[9] X.-Y. Zhang, C.-L. Liu, Style Transfer Matrix Learning for Writer Adaption, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2011.

[10] J. B. Tenenbaum, W. T. Freeman, Separating style and content with bilinear models, in: Neural Computation, 2000.

[11] C. Leggetter, P. Woodland, Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models, in: Computer Speech and Language, 1995.

[12] L. Jin, K. Ding, Z. Huang, Incremental learning of lda model for chinese writer adaptation, in: Neurocomputing, 2010.

[13] N. Matic, I. Guyon, J. Denker, V. Vapnik, Writer-adaptation for on-line handwritten character recognition, in: Proc. IEEE Int'l Conf. Document Analysis and Recognition (ICDAR), 1993.

[14] W. Kienzle, K. Chellapilla, Personalized Handwriting Recognition via Biased Regularization, in: Proc. IEEE Int'l Conf. Machine Learning (ICML), 2006.

[15] V. Frinken, H. Bunke, Evaluating retraining rules for semi-supervised learning in neural network based cursive word recognition, in: Document Analysis and Recognition, 10th International Conference on, IEEE, 2009.

[16] A. Brakensiek, A. Kosmala, G. Rigoll, Comparing Adaptation Techniques for On-Line Handwriting Recognition, in: Proc. IEEE Int'l Conf. Document Analysis and Recognition (ICDAR), 2001.

[17] K. P. Shankar, C. V. Jawahar, R. Manmatha, Nearest neighbor based collection ocr, in: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, 2010.

[18] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, in: Neural computation, 1998.

[19] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K. Mullers, Fisher discriminant analysis with kernels, in: Neural Networks for Signal Processing, 1999.

[20] M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, `http://cvxr.com/cvx` (2013).

[21] M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in: Recent Advances in Learning and Control, 2008.