Synthesizing Classifiers for Novel Settings

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS
in
Computer Science

by

Viresh Ranjan
201250875
viresh.ranjan@students.iiit.ac.in

Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2015

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled " Synthesizing Classifiers for Novel Settings" by Viresh Ranjan, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Dr. C. V. Jawahar

To my family

Acknowledgments

First of all, I would like to show my deepest gratitude towards my adviser, Dr. C. V. Jawahar, for his guidance, support, motivation and for keeping me from going off track. Many thanks to Dr. Gaurav Harit, my co-adviser, without whom this work would not have been the same. I would also like to wholeheartedly thank my family, my parents and my brother, for their love and unwavering support in all my ventures. A special thanks to my brother for taking out time from his busy schedule to proofread all my research papers as well as my thesis. I am thankful to my lab mates and friends at IIIT for all the fruitful (as well as not so fruitful) discussions, Anand, Devendra, Udit, Praveen, Sanchit, Aniket, Naveen, Ajeet, Sandeep, Pritish, Arunava, Saurabh, Yashaswi, Nataraj, Nagender, Vijay, Swagatika, Suriya, Mohak, Rajvi, Abhinav, Pramod, Pulkit, Varun, Koustav. A special thanks to all the faculty members at CVIT, my adviser Dr. Jawahar as well as Dr. P. J. Narayanan, Dr. Anoop Namboodiri and Dr. Jayanthi Sivaswamy for creating a wonderful environment in CVIT to do research, and CVIT administration Satya, Phani, Rajan and Nandini for helping me on numerous occasions.

Abstract

Computer vision systems have been developed which perform well at recognizing and retrieving natural images as well as document images. However, these systems might not work well under certain scenarios, for instance, when the distribution of the training and the test data do not match. For example, an OCR system may not work on those target fonts which are very different from the fonts used while training. Moreover, such systems might not be able to tackle previously unseen categories. These scenarios limit the real world applications of computer vision systems to some extent. In this thesis, we tackle these problems by designing classifiers that work even in novel scenarios. We design algorithms for retrieving document images, as well as recognizing objects and digits in novel settings.

For the document image retrieval task, we consider two different scenarios. In the first scenario, we tackle the issue of novel query words in a classifier based retrieval system. We present a one-shot learning strategy for the learning of discriminative classifiers given a novel query word. This strategy utilizes the classifiers learned at the training time in order to obtain the classifier corresponding to the underlying query class. This extends the classifier based retrieval paradigm to an unlimited number of classes (words) present in a language. We validate our method on multiple datasets, and compare it with popular alternatives like OCR and wordspotting. In the second scenario, we tackle the problem of mismatch between the source and the target style(font). We tackle this problem by style(font)-content(word label) factorization strategy. Based on the style-content factorization, we present a semi-supervised style transfer strategy to transfer word images in the source font to the target font. We also present a nonlinear style content factorization for obtaining style independent representation of word images. We validate both these strategies on scanned document collections as well as multi-font synthetic datasets. We show mean average precision gains of upto 0.30 over the baseline using our nonlinear factorization strategy.

For the recognition task, we tackle the challenging problem of dataset shift between the source and the target data. Dataset shift is the scenario where the joint input-output distribution for the training and the test data are different. In such a scenario, the classifiers trained on the source data might perform poorly on the target data. We tackle two different tasks in this scenario, i.e. digits recognition and object recognition. The two domains we consider for the digits recognition task are handwritten and printed digits. To tackle the digits recognition task, we present a subspace alignment based strategy. In this approach, labeled source domain data and unlabeled target domain data is used to learn transformations for the two domain which reduces the mismatch between the two domains. A source domain classifier learned after applying the transformations would work well even on the target domain data. We consider

the simple nearest neighbor based classifier for validating this claim. For the object recognition task, we present a sparse representation based strategy. A dictionary learned from the source data might not be suitable for sparsely representing target domain data and vice-versa. Hence, we present a partially shared dictionary learning strategy which results in dictionaries which are suitable for representing the source as well as the target domains. We show our results on popular benchmark datasets and show improvement over the state of art approaches.

Contents

List of Figures

Figure Page

List of Tables

Chapter 1

Introduction

Human visual system possesses an astounding capability for pattern recognition and generalization. Humans can easily learn the concept of a previously unseen object category by just looking at few example images of that object category. One of the primary goals of computer vision has been to endow computers with such pattern recognition and generalization capabilities. Even though achieving human like visual capabilities still seems like a distant goal, still we have been able to achieve commendable performance in tasks such as optical character recognition, face detection and recognition, pedestrian detection etc.

A major portion of computer vision research has been directed towards recognizing entities such as objects, handwritten and printed text, faces etc. in the images. A variety of techniques exist for recognizing these entities automatically. These techniques typically consist of extracting interesting/relevant features from a large number of labeled images and feeding these to a machine learning algorithm. The machine learning algorithms in turn, learn classifiers which are able to recognize the presence of the underlying categories in the test images. These classifiers are then deployed in the real world. The performance of such systems depends vitally upon two assumptions

1. The training and the test data follow the same underlying distribution.

2. Sufficient labeled data is available for each of the categories.

In case either one, or both of these assumptions are not valid, the system might not perform well on the real world. It turns out that these assumptions need not always be true, we elaborate further upon this in Section 1.2. But first, in section 1.1, we describe the computer vision tasks tackled in this thesis in detail. In section 1.2, we discuss the novel scenario challenges which need to be addressed while designing strategies for the computer vision tasks. In section 1.3, we present few machine learning approaches which could be used for tackling the novel scenarios in the computer vision tasks. In Section 1.4, we discuss our problem statement. The primary technical contributions of this thesis have been highlighted in Section 1.5.

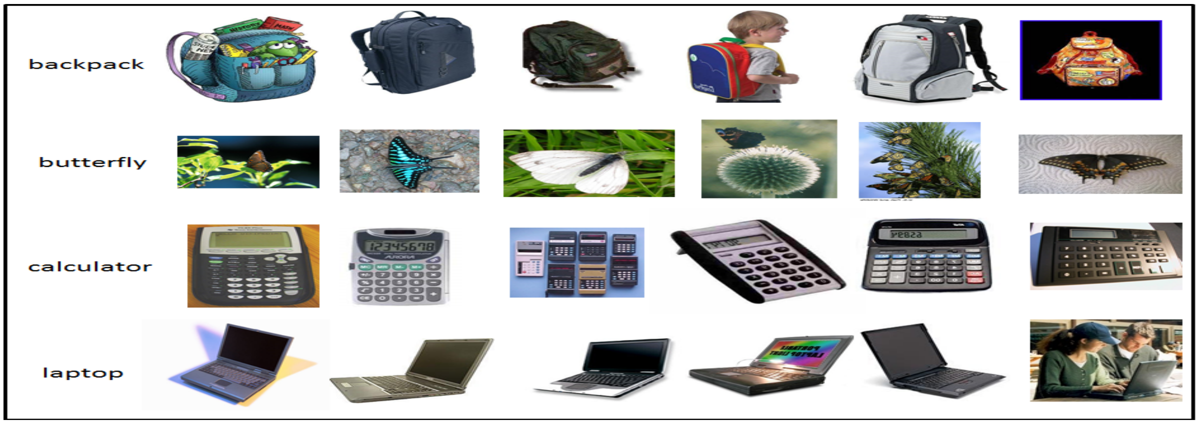


Figure 1.1 Figure shows few images and their corresponding class labels from the popular objection recognition dataset Caltech-256. Large intra-class variations, which can be clearly observed from the shown images, makes object recognition a challenging problem.

1.1 Computer Vision tasks

Here, we define the two computer vision tasks tackled in this thesis.

1.1.1 Recognition

Human visual system is good at recognizing and locating objects, faces, handwritten or printed digits or text present in images. For a computer, however, it is extremely difficult to perform these tasks. One of the longstanding goals of computer vision has been to make the computers *see* as humans do, i.e. localize and recognize the various entities present. Intra-class variations, inter-class similarity, change in pose and lighting conditions, occlusion, are some of the factors which make this task extremely difficult.

Recognition deals with the *what* and the *where* problems, i.e. what objects are present in an image and where is the location of the objects in the images. Some specific examples of the *what* problems are object recognition, face recognition, handwritten and printed digit recognition etc whereas some examples of the *where* problems are object detection, pedestrian detection, face detection etc. In this thesis, we tackle two recognition tasks, i.e. object recognition and handwritten/printed digits classification. In Fig 1.1, we give a few sample images from Caltech-256.

1.1.2 Retrieval

Image retrieval deals with retrieving relevant images from a database, given an image query. It consists of extracting, storing and indexing the features of the images in the database. Given a query image, its features are extracted and compared with the stored database features and a ranked list of

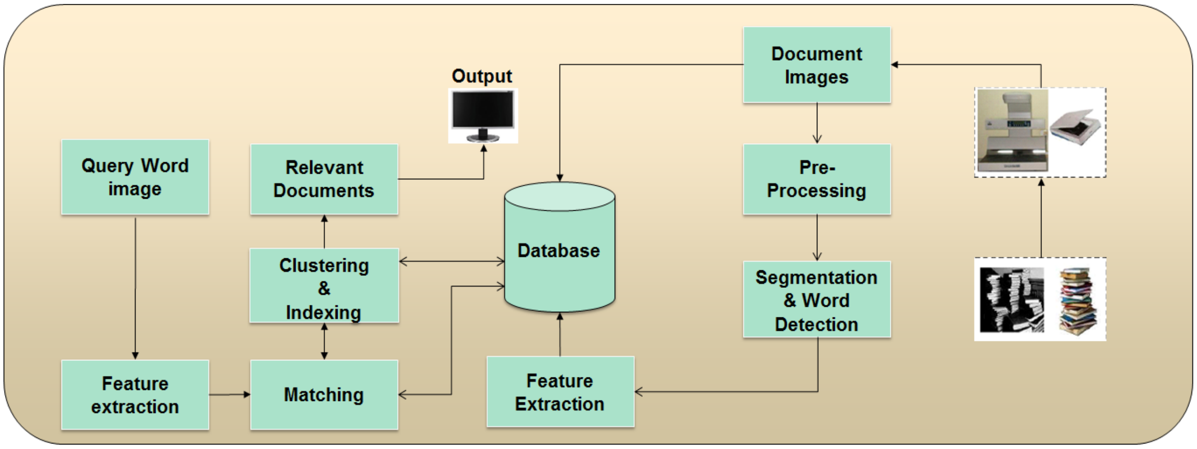


Figure 1.2 Figure shows a typical worspotting based document image retrieval system. User can give query in the form of a word image, which would result in retrieval of relevant documents from the database. Pic courtesy: Praveen Krishnan, CVIT.

images relevant to the query is retrieved. In this thesis, we specifically deal with document image databases.

Reduction in the cost of imaging devices, as well as digital storage devices , have resulted in large digital libraries, consisting of millions of document images. Given a query, document image retrieval considers the task of retrieving relevant documents (pages, paragraphs or words) from the digital libraries. Most of the present day digital libraries, use Optical Character Recognizers (OCRs) for the recognition of digitized documents, and thereafter employ a text-based solution for the information retrieval. Though OCRs have become the *de facto* preprocessing for the retrieval, they are realized as insufficient for degraded books [51], incompatible for older print styles [34], unavailable for specialized scripts [87] and very hard for handwritten documents [5]. To overcome these demerits of an OCR based document retrieval system, recognition free word spotting techniques have been proposed which take a word image as query and retrieve relevant images from the database on the basis of their similarity to the query image. A typical worspotting based document image retrieval system has been shown in Figure 1.2.

Consider a large document image database consisting of millions of documents. Manual annotation of all the documents in such a large collection of documents is practically impossible. Hence, to access desired documents from such databases, document image retrieval techniques can be employed. As discussed previously, an OCR based retrieval system might not be very useful in a number of scenarios. Hence, a recognition free word image retrieval system can be employed for accessing relevant documents from the database. In such systems, query is given in the form of a word image, and documents can be retrieved from the database based on the their similarity to the query word image. Such systems typically employ a nearest neighbor classifier based retrieval strategy as this classifier does not need any training. However, results show that SVM or LDA classifier based retrieval strategies(where each word is taken as a different category)can outperform a nearest based classifier for the word image retrieval task(See Table 3.2 of Chapter 3). Hence, in this work, we use such a classifier based retrieval strategy.

1.2 Challenges in Computer Vision tasks

Most of the tasks in computer vision face a number of challenges which make them rather difficult. Some of these challenges include: variation in pose and lighting conditions of objects, scarcity of labeled data, dataset shift, intra-class dissimilarity, inter-class similarity, image quality etc. In this thesis, focus on the challenges presented below.

1.2.1 Dataset Shift

Let us consider the handwritten digit recognition task. Also, let us assume that we do not have any labeled handwritten data for training the classifiers, but we do have labeled printed digits available in a large number of fonts. We can use the classifiers trained on the printed digits to classify the handwritten digit images. However, the performance of the classifiers might be very low because of possible dissimilarities between the handwritten and the printed digits. Such changes in the dataset, where the joint input-output distribution changes across the training and the test data are called dataset shift. Some other dataset shift examples are difference in fonts/styles across the source and the target documents for a classifier based document image retrieval system. In object recognition, factors such as quality of imaging device, viewing angle and intra-class variations can also give rise to dataset shift. In Figure 1.3, we present same category images from various datasets. Large intra-class variations can be clearly observed across the datasets. A classifier trained on one of the datasets might not be suitable for classification task over the other target datasets because of the large intra-class variations.

1.2.2 Labeled data scarcity

We use the term *Labeled data scarcity* to imply two scenarios, the first being there may be only a few labeled examples available for some categories, and the second being no labeled examples may be available for some categories. The labeled data scarcity scenario, which we tackle in this thesis, is the latter one, i.e. missing labeled examples for some categories. For example, consider a classifier based retrieval system which has been trained for the frequently occurring words. However, as the total number of words in a language such as English is typically very large, the query words might come from a previously unseen word class. If any query indeed pertains to a category for which we do not have any labeled data, it might not be possible to carry out the query.

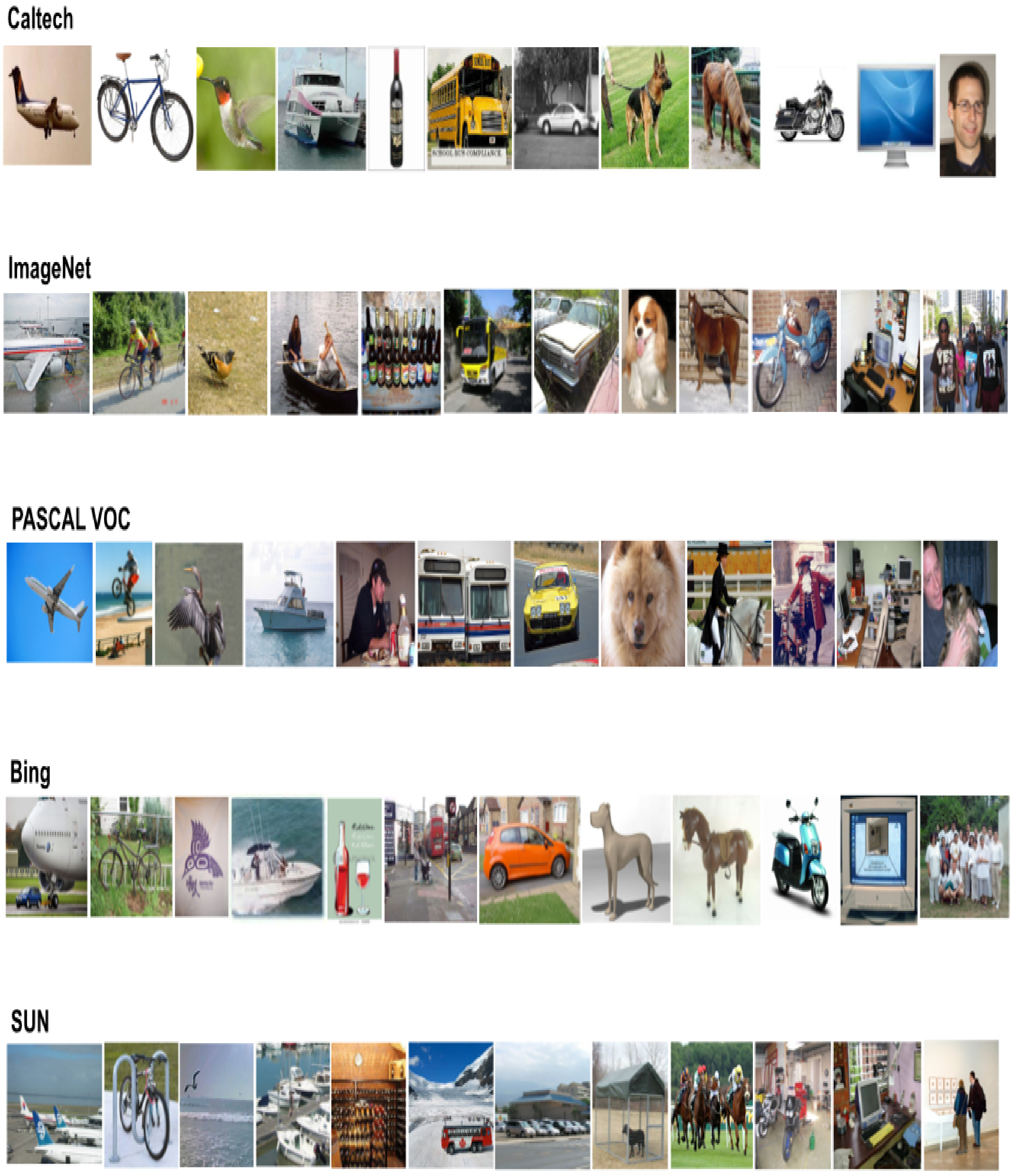


Figure 1.3 Figure shows images corresponding to 12 different categories from various datasets. The large intra-class dissimilarities across the datasets can be clearly observed from the shown images.

1.3 Handling novel scenarios

A number of machine learning strategies can be used for handling novel scenarios. We briefly describe some of these strategies below.

1.3.1 Semi-Supervised Learning

The goal of semi-supervised learning strategies is to utilize labeled as well as unlabeled data in order to obtain better performance than that obtained by using just the labeled data. These strategies are useful in the scenarios where a large amount of unlabeled data is also available along with the labeled data. The large amount of unlabeled images and videos, available over the internet, makes this scenario very relevant for various computer vision tasks. In Figure 1.4, we present the general idea behind a semi-supervised classification strategy. Unlabeled data is also considered while learning the classifier.

Semi-supervised algorithms can be broadly classified into two types

- Algorithms which allow usage of unlabeled data in supervised approaches, for example, self-training algorithms [105].
- Algorithms which allow for use some form of supervision in originally unsupervised algorithms, for example constrained clustering algorithms such as constrained k-means [25].

Self-training models are a popular class of semi-supervised classification strategies. In self-training models, a classifier is trained using the labeled examples and the unlabeled examples over which the classifier is confident are labeled using the classifier output. These examples are then added to the labeled set and the classifier is retrained and the entire process is repeated a number of times. For a detailed description of other semi-supervised learning strategies, we refer the reader to Zhu and Goldberg [105].

Constrained k-means algorithms [25] typically consist of must-link and cannot-link constraints. As the name suggests, must-link constraints enumerate such samples which must always occur in the same clusters, and cannot-link constraints specify those samples which must never be assigned to the same clusters.

1.3.2 Domain Adaptation

In presence of dataset shift, the classifiers trained on the source domain might perform badly on the target domain. Domain adaptation(DA) techniques handle the dataset shift scenario and allow use of the labeled source data for building classifiers which are effective for the target domain. DA techniques generally assume that the source domain and the target domain has sufficient number of labeled and unlabeled examples respectively. Some labeled data might be available in the target domain also.

Domain adaptation techniques typically transform either the classifiers or the feature representation. The classifier based DA techniques such as [101] modify the source domain classifiers using some

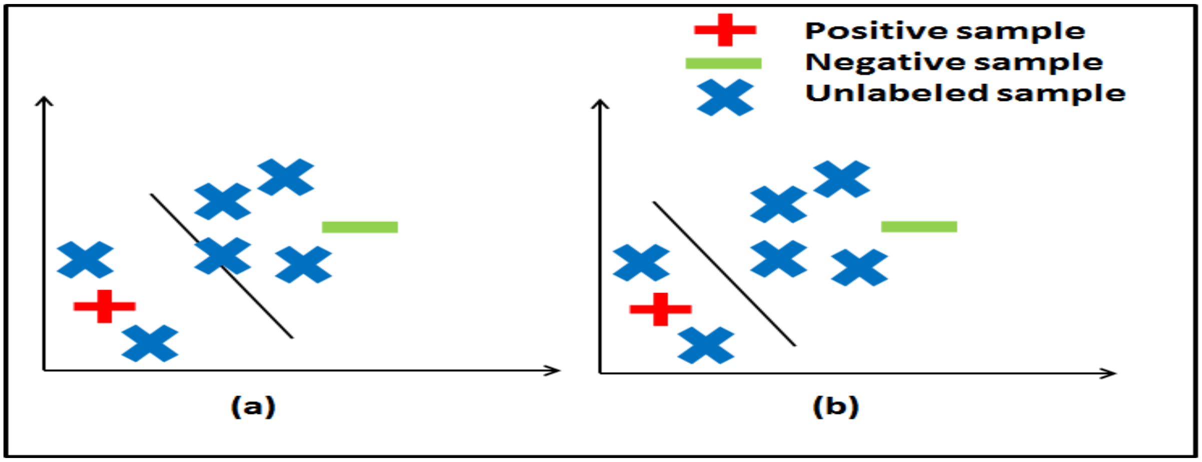


Figure 1.4 Figure shows how taking into account the unlabeled samples may guide the supervised learning algorithm in case only a few labeled samples are available. Fig(a) shows classifier learned using just the labeled examples whereas Fig(b) shows classifier which also considers the unlabeled examples.

labeled data from the target domain. The feature representation based DA approaches such as [86] transform the features from the source and/or target domain so that the intra-class dissimilarities across the source and the target domain are reduced as a result of the transformation.

1.3.3 Transfer Learning

Consider the object detection task in computer vision. Assume we have already trained a detector for one object category, say car and want to learn the detector for a new category, say car tyres. As the two categories share some similarities, i.e. all the cars have tyres, it might be a good idea to utilize the car detector in some way while learning the car-tyre detector. Using this prior knowledge might reduce the number of labeled examples needed to train the car-tyre detector. This idea has been presented in Figure 1.5.

Transfer learning techniques deal with transferring knowledge learned from one or more source tasks to a related target task. If application of a transfer learning strategy improves the classifier performance then *positive transfer* is said to occur whereas if the classifier performance deteriorates then *negative transfer* is said to occur. One of the challenges of transfer learning is to allow for positive transfer for related tasks while keeping in check the negative transfer between the unrelated tasks.

Transfer learning techniques have been presented for computer vision tasks such as object category recognition [9]. In this work, while training a support vector machine(SVM) for a new category by using just a few examples, the SVM classifiers trained previously for related categories are used to provide regularization.

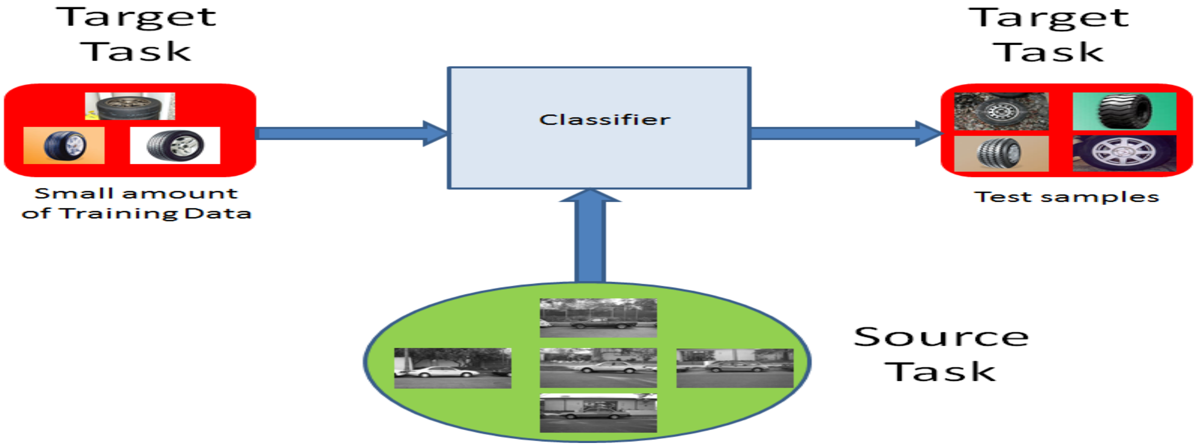


Figure 1.5 Figure shows central idea behind transfer learning. Knowledge from source task, in the form of the trained source classifiers, can be utilized alongwith data from related target task to learn classifers that perform well on the target task.The source and the target task should be related, in order for the transfer to work well.

1.3.4 One-Shot Learning

One-shot learning, similar to Transfer learning, also uses knowledge obtained from previously learned classifiers while training a novel classifier. However, a major difference between the two is that in transfer learning, knowledge from similar categories is used while training classifier for a novel category, whereas in one-shot learning, knowledge learned from all the previously trained classifiers is used.

Some of the popular one-shot learning strategies in computer vision employ a Bayesian approach [31]. They represent the previously learned object categories by probabilistic models. Prior knowledge from these categories is modeled as a probability distribution over the parameters of the models. Now given a single or a few examples from a novel category, the prior is updated to give the posterior model for the novel category.

1.4 Problem Statement

In this thesis, we look into the problem of performing recognition and retrieval tasks in presence of novel scenarios. To be specific, we tackle two novel scenarios

1. Given a set of discriminative classifiers, how to handle test images from previously unseen categories.

2. Given labeled data in source domain and unlabeled data in target domain, how to classify/retrieve images from the target domain.

In the first scenario, we tackle the problem of handling out of vocabulary queries in a classifier based word image retrieval system. For this, we present a one-shot classifier synthesis strategy for linear discriminant classifiers(LDA). Given a query word image from a previously unseen word class(labeled training data from the class is not available)and a set of pre-trained LDA classifiers, a classifier corresponding to the query is synthesized by extracting relevant information from the pre-trained classifier set. We have discussed this in Chapter 3.

In the second scenario, we look into retrieval as well as recognition tasks. For the retrieval task, we tackle the cross-font retrieval problem in a multi-font database. In this problem, the query and the target document fonts are different. We present two approaches based on style(font)-content(underlying word label) factorization of word images to tackle this problem. A popular approach of style-content factorization is via the use of bilinear models proposed by Freeman and Tenenbaum [95]. We utilize their asymmetric bilinear model and present a retraining strategy to style transfer query words from one style to another in a semi-supervised manner. The asymmetric bilinear model, is a linear model and hence it cannot capture the nonlinearities in the data. Hence, to capture the nonlinearities in the data, we propose asymmetric kernel bilinear model(AKBM), a kernelized style-content factorization strategy. This strategy has been discussed in Chapter 4. AKBM is a general style-content factorization strategy and can be employed for doing nonlinear feature extraction in other computer vision tasks also. For the recognition task, we tackle recognition of handwritten digits, as well as object recognition in presence of dataset shift. For handwritten digit recognition, handwritten and printed digits are taken as the two domains. We wish to classify handwritten images from the target domain (say handwritten digits) using labeled data from the source domain (printed digits). We propose a subspace alignment based strategy for this task. This strategy has been discussed in Chapter 5. For the object recognition task, we wish to recognize objects present in the target domain images by utilizing labeled data from the source domain, and unlabeled data from the target domain. We propose a novel shared dictionary learning strategy which learns dictionaries which are suitable for representing images from both the domains. The shared dictionary results in similar sparse representation for same category images across the two domains. This strategy has been discussed in Chapter 6.

1.5 Contributions

The main contributions of this thesis are

1. **Problem:***Given* LDA *classifiers corresponding to a set of words, how to design a* LDA *classifier for a novel query, even when there are no labeled examples available.*
   We present a one-shot classifier synthesis strategy which synthesizes a LDA classifier for a novel query word by utilizing the pre-trained classifiers.

2. **Problem:***Given document images in two different style, how to use images from one style(source) to retrieve images from the other style(target).*
   We propose a kernalized style-content factorization strategy for obtaining a style independent

representation. We also present a semi-supervised style transfer strategy for transferring query word images from the source style to the target style.

3. **Problem:***Given labeled examples of objects in source domain, how to design classifiers to recognize the same categories of objects in the target domain.*
   We propose a shared dictionary learning strategy that results in dictionaries which are suitable for representing images from the source as well as the target domains. Using these dictionaries, we separate the domain specific information and the information which is common across the domains. We use the latter for training cross-domain classifiers, i.e. , we build classifiers that work well on a new target domain while using labeled examples only in the source domain.

4. **Problem:***Given labeled examples of printed digits, how to recognize handwritten digits and vice-versa.*
   We propose a subspace alignment based domain adaptation technique to tackle this mismatch. Unlike previous subspace alignment approaches, we introduce a strategy to effectively utilize the training labels in order to learn discriminative subspaces.

1.6 Thesis Outline

In Chapter 2, we briefly discuss the technical background needed for the later chapters in the thesis. In Chapter 3, the problem of classifier design for novel query words has been tackled. In Chapter 4, we discuss strategies to use the nearest neighbor classifier in presence of novel styles. In Chapter 5 and Chapter 6, we tackle classification in presence of dataset shift. In Chapter 5, we present a semi-supervised subspace alignment based approach for digits classification. In Chapter 6, we present a partially shared dictionary learning strategy for the object recognition task. Conclusions, and future directions of our work have been given in Chapter 7.

Chapter 2

Background

In this chapter, we briefly discuss the machine learning tools which have been used in the thesis. In Section 2.1, we look at the popular linear as well as non-linear feature extraction strategies. For the linear techniques, we look into Principal Component Analysis (PCA) in Section 2.1.1 and Locality Preserving Projections in Section 2.1.2. For nonlinear techniques, we look into sparse representation in Section 2.1.3 and Kernelized version of LPP in Section 2.1.4. In Section 2.2, we discuss the classification strategies being used in the thesis. In Section 2.3, we describe the different datasets and the performance measure being used in this thesis.

2.1 Feature Extraction

2.1.1 Principal Component Analysis

Principal Component Analysis (PCA) is a very popular dimensionality reduction approach proposed by Hotelling [43]. Apart from dimensionality reduction, PCA is also used for data visualization, feature extraction and lossy data compression [15]. PCA is the projection of data onto a orthogonal set of basis vectors such that the variance of the data after the projection is maximized. These basis vectors are obtained by solving an eigenvalue problem involving the covariance matrix of the data samples. Consider a dataset consisting of n observations $x_1, x_2, ...x_n$ such that $x_i \in R^d$. Let the first principal component be obtained by projecting the data along u_1, i.e. the direction of maximum variance of the data. The variance of the data along the direction u_1 can be represented as

$$u_1^T S u_1 \tag{2.1}$$

where S is the covariance matrix of the observations given by

$$S = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T. \tag{2.2}$$

Here the column vectors x_i have been centered around the dataset mean. In order to maximize the variance given in Equation 2.1, we need to maximize the objective with respect to u_1. However, unconstrained maximization of this objective will result in u_1 having very large magnitude. In order to keep the l_2 norm of u_1 under check, a unit norm constraint is added to objective. This constrained objective can be converted to an unconstrained one using the method of Lagrange multipliers . Using this approach, the unconstrained objective can be rewritten as

$$u_1^T S u_1 \quad + \lambda_1(1 - u_1^T u_1), \tag{2.3}$$

where λ_1 is the Lagrange multiplier. This objective can be maximized by setting its derivative with respect to u_1 as zero. This results in the following eigenvalue problem

$$S u_1 = \lambda_1 u_1. \tag{2.4}$$

Hence u_1 is the eigenvector of S which corresponds to its largest eigenvalue. The other eigenvectors can be found similarly with added constraint that the eigenvectors must be orthogonal to one another. We have described here the maximum variance formulation for PCA as in Bishop [15].

2.1.2 Locality Preserving Projections

PCA aims at finding projection directions along which the variance of the data after projection is maximized. However, it does not try to preserve the local structure of the data points. He and Niyogi [42] present Locality Preserving Projections (LPP) which preserves the local neighborhood of data points during the projection. Although LPP is a linear dimensionality reduction algorithm like PCA, it shares many of the representation properties with nonlinear techniques such as Laplacian Eigenmaps [13]. In LPP, first a graph is constructed based on the neighborhood information of the data points. Then a transformation matrix is obtained using the graph Laplacian which preserves the neighborhood structure in the graph.

Given a dataset with m vectors $x_1, x_2, ..., x_m$ in R^n and their corresponding labels $y_1, y_2, ..., y_m$, LPP finds a set of basis vectors A (each column of A is a basis vector) so that the neighborhood of each of the m points is preserved after the transformation $z_i = A^T x_i$. Note that LPP does not use the labels while finding the basis vectors. To obtain the transformation matrix A, first an adjacency graph $G = (V, E)$ with m nodes is formed. Nodes i and j of G are connected by an edge if the vectors x_i and x_j are close to one another. Here, x_i and x_j are considered to be close based on either of these two conditions :

- $||x_i - x_j||^2 < \epsilon$ where $\epsilon \in R$.
- x_i and x_j are among the k nearest neighbors of one another.

Once the adjacency graph G is constructed, the undirected weights W_{ij} between the nodes corresponding to x_i and x_j can be obtained using either of these two strategies

- Heat kernel: if nodes i and j are connected, $W_{ij} = e^{-\frac{||x_i - x_j||^2}{t}}$, otherwise $W_{ij} = 0$, here $t \in R$.

- $W_{ij} = 1$ if nodes i and j are connected, otherwise $W_{ij} = 0$.

The columns a of the matrix A can be found by solving the following generalized eigenvalue problem:

$$XLX^T a = \lambda XDX^T a \tag{2.5}$$

where i^{th} column of X is x_i, D is a diagonal matrix such that $D_{ii} = \sum_j W_{ij}$ and $L = D - W$ is the Laplacian matrix. Solutions of this equation are the eigenvectors that form the columns of the transformation matrix A.

2.1.3 Sparse Representation of Signals

Consider the equation

$$Ax = b, \tag{2.6}$$

such that $A \in R^{n \times m}$ and $n < m$. Solution of such an equation exists if b lies in the column space of A, i.e. matrix A is full rank. In such a scenario, infinitely many x are possible which satisfy equation 2.6. From this set of solutions, we want to obtain a unique solution which is optimal in some sense. To obtain a unique solution, generally an $l2$ norm regularizer is added as

$$\min_x |x|_2^2 \quad \text{subject to} \quad Ax = b. \tag{2.7}$$

Some of the reasons for the popularity of the $l2$ norm regularizer are the existence of a closed form and unique solution of the above equation. However, in many scenarios, the $l2$ norm regularizer does not necessarily result in the best solution for a specific task.

Recently there has been a lot of interest in the sparse solutions of the Equation 2.6. Sparsity is defined as the number of nonzero elements of x. To ensure a sparse solution of the Equation 2.6, the $l1$ norm constraint is replacd by the $l0$ norm constraint as

$$\min_x |x|_0 \quad \text{subject to} \quad Ax = b. \tag{2.8}$$

However, it turns out that the above problem is not convex. Greedy algorithms such as Orthogonal Matching Pursuit (OMP) [97] can be used to solve this optimization problem. The $l0$ norm can be relaxed to a $l1$ norm resulting in the following optimization problem

$$\min_x |x|_1 \quad \text{subject to} \quad Ax = b. \tag{2.9}$$

It turns out that the $l1$ norm results in a convex optimization problem and also results in sparse solution under some constraints.

A lot of work has been done concerning the sparse representation of signals in the computer vision community. In this representation, signals are represented as a sparse linear combination of prototype signal-atoms, i.e. column vectors of an overcomplete dictionary [3]. Dictionary learning forms an integral part of sparse representation as the quality of representation depends upon the suitability of the dictionary elements to compactly represent the signals. Dictionary learning techniques generally consist of a sparse coding step and a dictionary update step [3, 30]. We describe methods for sparse decomposition of signals in Section 2.1.3.1 and dictionary learning approaches in Section 2.1.3.2.

2.1.3.1 Methods for Sparse Decomposition

The equation 2.7 can be solved by greedy approaches. The greedy approaches need the upper bound on the sparsity (number of nonzero elements in x). These greedy approaches view the problem as consisting of two parts

- **Determining the support** This consists of determining which of columns of A are to be used for representing b.
- **Least square** Once the support has been determined, the problem can be solved by doing simple least squares.

The greedy approaches start with an empty support and at each step, the column of A which minimizes the *residue* is added to the support. The residue is defined as the $l2$ norm of the error obtained by representing b using the active columns of A (which are in the current support).

2.1.3.1.1 Orthogonal Matching Pursuit The orthogonal matching pursuit algorithm (OMP) takes as input A, b and an error threshold ϵ_0. As discussed earlier, the algorithm starts with an empty support and iteratively keeps adding columns of A to the support. At the k^{th} step, the approximation x^k for x is found by solving the least square problem

$$\min_x \|Ax - b\|_2^2 \quad \text{subject to} \quad Support(x) = S^k, \tag{2.10}$$

where S^k is the support of the solution at the $k - th$ step. x^k is then used to obtain the residue r^k as

$$r^k = b - Ax^k. \tag{2.11}$$

If the $l2$ norm of the residue goes below the error threshold ϵ_0, the algorithm is terminated otherwise further iteration is started by finding the new support S^{k+1}.

2.1.3.2 Dictionary Learning

A signal $y \in R^n$ can be sparsely represented using a dictionary $D \in R^{n \times K}$ consisting of K atoms or prototype signals. The atoms of D can be pre-defined using discrete cosine transform basis [4], wavelets [60] or they can be learned from the available signals. The learned dictionaries have been shown to perform better than pre-defined dictionaries for tasks such as reconstruction [28]. For learning dictionaries from the data, several efficient dictionary learning strategies such as K-SVD [3] and

MOD [30] have been proposed in the past. These dictionary learning techniques solve the following optimization problem

$$\min_{D,A} \|Y - DA\|_F^2 \quad \text{subject to} \quad \forall i, \quad \|a^i\|_0 \leq T_0. \tag{2.12}$$

Here the signals are arranged along the columns of Y and the columns of A, i.e. a^i, contain the corresponding sparse representation. The dictionary learning techniques solve this problem by alternating between solving for A, i.e. sparse coding step and updating D, i.e. dictionary update step. In [30], the dictionary update consists of updating all the dictionary elements while keeping the sparse representation unchanged. The dictionary learning approach given in [3], however, updates a single dictionary atom at a time. The sparse coefficients also change during the update so that the number of nonzero coefficients further reduces or remains the same.

2.1.4 Kernelized Approaches

Sometimes the patterns inherent in the data are not apparent from the original feature representation. In such scenarios, it is a common practice to do a linear or a non-linear mapping/transformation of the data so that after the transformation, the patterns in the data become more obvious. Kernel based approaches are a non-linear feature mapping approach which map the data points in the original space to its mapping in a Reproducing Kernel Hilbert Space (RKHS).

Consider feature vector $x \in R^d$. A kernel function k is defined as

$$\begin{aligned} k : X \times X &\mapsto \mathbb{R}, \\ (x_i, x_j) &\mapsto k(x_i, x_j), \end{aligned} \tag{2.13}$$

where x_i and x_j objects from X. The kernel function k is said to be positive definite if it satisfies the following properties

- k is symmetric, i.e. $k(x_i, x_j) = k(x_j, x_i)$
- $\sum\limits_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0$, where $\alpha_i, \alpha_j \in \mathbb{R}$ and $n \in \mathbb{N}$

If the kernel function is positive definite, then there exists some mapping function ϕ such that

$$< \phi(x_i), \phi(x_j) >= k(x_i, x_j), \tag{2.14}$$

here ϕ is a mapping function which maps data points from original space to a Reproducing Kernel Hilbert Space (RKHS).

2.1.4.1 Kernelized LPP

If any algorithm can be expressed so that the data points always occur in dot-product, then we need not know the feature mapping ϕ explicitly. Using Equation 2.14, all occurrences of dot-products

$< \phi(x_i), \phi(x_j) >$ are replaced by the kernel function evaluation $k(x_i, x_j)$. This technique is called the kernel trick and has been used for kernelizing a number of algortihms such as PCA [89], LDA [66] and many other algorithms.

Now we show the application of kernel trick to obtain the kernelized version of locality preserving projections (LPP). Let us define a kernel matrix K as $K_{i,j} = k(x_i, x_j)$. Consider the generalized eigenvalue problem $XLX^Tv = \lambda XDX^Tv$. Let $\phi(X)$ correspond to the mapping of all the data points X. Hence, the equation can be represented as

$$\phi(X)L\phi(X)^Tv = \lambda\phi(X)D\phi(X)^Tv. \tag{2.15}$$

The generalized eigenvectors v can be represented as linear combinations of columns of the data points as

$$v = \phi(X)\alpha. \tag{2.16}$$

Using equation 2.17, equation 2.15 can be rewritten in terms of the Gram matrix (kernel matrix) K as

$$KLK\alpha = \lambda KDK\alpha. \tag{2.17}$$

α can be obtained by solving the above generalized value problem. The test samples can now be projected along the eigenvectors obtained by solving the above equation.

2.2 Classification

2.2.1 Linear Discriminant Analysis

PCA as well as LPP do not utilize the label information. As a result, the basis vectors obtained using these two techniques might result in projections which are not suitable for the classification task. Linear discriminant analysis (LDA) utilizes the class membership information of the samples while obtaining the projection vectors.

For a binary classification problem, LDA tries to find projection of vectors over such a direction along which the projection of the data samples from the two classes are linearly separable. Let such a direction be represented by $w \in R^d$. Also, let the covariates be represented by d-dimensional vectors x. Then the binary classification rule can be given as

$$y = w^Tx + w_0. \tag{2.18}$$

If $y \geq 0$, x belongs to the positive class otherwise it belongs to the negative class.

In order to obtain w, let the mean vectors of the positive and the negative classes be represented as μ_+ and μ_-. In order to keep the two classes far apart from one another, the direction w is chosen such that the projection of μ_+ and μ_- along w are as much far apart from another as possible. LDA also keeps the intra-class spread of each of the two classes along w small by minimizing the variance of each of the classes along w. These two result in the following objective function

$$J(w) = \frac{w^T S_B w}{w^T S_W w}, \tag{2.19}$$

where S_B is the between class covariance matrix given as

$$S_B = (\mu_+ - \mu_-)(\mu_+ - \mu_-)^T, \tag{2.20}$$

and S_W is the within class covariance matrix given as

$$S_W = \sum_{i \in C_+} (x_i - \mu_+)(x_i - \mu_+)^T \;\; + \sum_{i \in C_-} (x_i - \mu_-)(x_i - \mu_-)^T \;\; , \tag{2.21}$$

where C_+ corresponds to the positive class and C_- corresponds to the negative class. In order to maximize this objective function, we differentiate it with respect to w and set it to zero which gives the following equation

$$(w^T S_B w) S_W w = (w^T S_W w) S_B w. \tag{2.22}$$

As we are bothered only about the direction of w, we can remove $w^T S_B w$ and $w^T S_W w$ from the equation as these are scalar quantities. Now, $S_B w$ always points in the direction of the vector $\mu_+ - \mu_-$. Using these facts (and ignoring the scalar term in $S_B w$), we obtain the following eigenvalue problem which can be used to obtain w

$$w = S_W^{-1}(\mu_+ - \mu_-). \tag{2.23}$$

It is a common practice to assume the same within class covariance matrix S_W for all the classes and setting it equal to the combined covariance matrix of the entire data.

2.2.2 Support Vector Machine

Consider a binary classification problem where any feature vector $x \in R^d$ and corresponding class labels can be $y \in \{1, -1\}$. Support vector machine (SVM) constructs a decision surface in the form of a separating hyperplane:

$$w^T x + b = 0 \tag{2.24}$$

so that the positive and the negative examples are separated by the hyperplane. There can be an infinite number of hyperplanes which separates the positive & the negative examples (in case the data is linearly separable). Out of all such hyperplanes, SVM picks the hyperplane which maximizes the margin between the examples in the positive and the negative class. Such a hyperplane must satisfy:

$$w^T x_i + b \geq 1 \text{ for } y_i = 1, \tag{2.25}$$

$$w^T x_i + b < -1 \text{ for } y_i = -1 \tag{2.26}$$

The margin of separation ρ, between the positive and the negative examples is given by

$$\rho = \frac{2}{|w|} \tag{2.27}$$

The margin maximizing hyperplane can be found by solving the following constrained optimization problem

$$\min_{w,b} \frac{1}{2} w^T w \quad \text{subject to} \quad y_i(w^T x_i + b) \geq 1. \tag{2.28}$$

The above optimization problem is convex and hence, a unique hyperplane exists which is a global solution of the above constrained optimization problem. However, it turns out that many real world datasets are not linearly separable. Hence, in such a case, any feasible solution for the above objective does not exist. A *soft margin* version of SVM exists which can deal with such a scenario.

2.2.3 Nearest Neighbors

Nearest neighbors are one of the most classifiers, possibly because it does not involve any training. This technique simply consists of storing all the labeled training examples and given a test images, the label of closest training sample (or majority label of k-neighbors) is assigned to the test image. Nearest neighbors can be easily kernelized or coupled with metric learning [99].

For classification techniques such as LDA or linear SVM, only a single weight vector needs to be stored per class (in a one-vs-rest setting), however, a nearest neighbor classification strategy typically consists of storing all the training samples. This is one of the major demerits of nearest neighbors.

2.3 Dataset and Evaluation measures

2.3.1 Dataset

Printed books Our printed book datasets, used in Chapter 3 and Chapter 4, comprises scanned English as well as Telugu and Malayalam books from a digital library collection. We manually created ground truth at word level for the quantitative evaluation of the methods.

Dlab We created multi-font Dlab dataset under laboratory settings to conduct experiments on style transfer and style independence. This dataset consists of 500 different words rendered in Andalemo, Arial, Courier, Courier bold, Comic, Georgia, Impact, Verdana, Trebuchet and Times.

MNIST MNIST [54] is a popular handwritten digits dataset. It consists of $60,000$ training images and $10,000$ test images of digits $0-9$. We randomly sample three sets of images to form the training, test and validation sets for the experiments in Chapter 5.

Multi-font digits We created this multi-font digits dataset in laboratory setting for testing our domain adaptation approach in Chapter 5. We downloaded 1285 fonts from Googlefonts and rendered digits $0-9$ in these fonts.

Office-Caltech256 The Office-Caltech256 datset consists of 4 datasets, i.e. Amazon (images downloaded from online merchants), Webcam (images taken by a low resolution webcam), DSLR (images taken by a digital SLR camera) and Caltech (images taken from the Caltech-256 [40] dataset). The first three datasets were introduced in [86] whereas the fourth one was first studied by [36]. Each of the dataset are considered as a separate domain. Datasets consist of images pertaining to the following 10 classes BACKPACK, TOURING-BIKE, CALCULATOR, HEAD-PHONES, COMPUTER-KEYBOARD, LAPTOP, COMPUTER-MONITOR, COMPUTER-MOUSE, COFFEE-MUG, VIDEO-PROJECTOR. There are atleast 8 images and a maximum of 151 images per category in each domain. In total the datasets consist of 2533 images.

2.3.2 Evaluation Measures

We use classification accuracy for quantifying performance for the recognition tasks. For the retrieval tasks, we mean average precision (MAP). For defining classification accuracy for a binary classification task, we first define the following quantities

- **True Positive(tp):**number of correctly classified positive samples.
- **True Positive(tn):**number of correctly classified negative samples.
- **False Positive(fp):**number of positive samples misclassified as belonging to the negative class.
- **False Negative(fn):**number of negative samples misclassified as belonging to the positive class.

Now the accuracy can be defined as

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.29}$$

For the retrieval setting, tp is defined as number of correct samples in the retrieved results. Similarly, for tn, fp and fn. Now we define Precision and Recall:

- **Precision(P):**
$$P = \frac{tp}{tp + fp} \tag{2.30}$$
- **Recall(R):**
$$R = \frac{tp}{tp + fn} \tag{2.31}$$

PR curves are plotted by plotting precision as a function of recall obtained by varying parameters of the classifier. The area under this PR curve gives the average precision. For multi-class scenarios, the mean value of average precision for each of the classifiers is used.

Chapter 3

Synthesizing classifiers for Novel categories

In this section, we describe a classifier based retrieval scheme for efficiently and accurately retrieving relevant documents. We use SVM classifiers for word retrieval, and argue that the classifier based solutions can be superior to the OCR based solutions in many practical situations. We overcome the practical limitations of the classifier based solution in terms of limited vocabulary support, and availability of training data. We design a one-shot learning scheme for dynamically synthesizing classifiers. Given a set of linear discriminant classifiers, we appropriately join them to create novel classifiers. This extends the classifier based retrieval paradigm to an unlimited number of classes (words) present in a language. We validate our method on multiple datasets, and compare it with popular alternatives like OCR and wordspotting. Even on a language like English, where OCRs have been fairly advanced, our method yields comparable or even superior results. Our results are significant since we do not use any language specific post-processing for obtaining this performance. For better accuracy of the retrieved list, we use query expansion. This also allows us to seamlessly adapt our solution to new fonts, styles and collections.

3.1 Introduction

Retrieving relevant documents (pages, paragraphs or words) is a critical component in information retrieval solutions associated with digital libraries. Most of the present day digital libraries, use Optical Character Recognizers (OCRs) for the recognition of digitized documents, and thereafter employ a text-based solution for the information retrieval. Though OCRs have become the *de facto* preprocessing for the retrieval, they are realized as insufficient for degraded books [51], incompatible for older print styles [34], unavailable for specialized scripts [87] and very hard for handwritten documents [5]. Even for printed books, commercial OCRs may provide highly unacceptable results in practice. The best commercial OCRs can only give word accuracy of 90% on printed books [100] in modern digital libraries. This means that every 10th word in a book is not searchable. Recall of retrieval systems built on such erroneous text is thus limited.

In this chapter, we hypothesise that the words, even if degraded, can be matched and retrieved effectively with a classifier based solution. A properly trained classifier can yield an accurate ranked list of words since the classifier looks at the word as a whole, and uses a larger context (say multiple examples) for the matching. We show, later in this chapter, that a SVM based word retrieval can give mean average precision (mAP) as high as 1.0 even when the OCR based solution is limited to an mAP of 0.89. (See Figure 3.1 and Table 3.2.) Our results are significant since (i) we do not use any language specific post-processing for improving the accuracy (ii) even for a language like English, where OCRs are fairly advanced and engineering solutions were perfected, our simple classifier based solution is as good, if not superior to the best available commercial OCRs.

However, there are two fundamental challenges in using a classifier based solution for word retrieval. (i) A classifier needs good amount of annotated training data (both positive and negative) for training. Obtaining annotated data for every word in every style is practically impossible. (ii) One could train a set of classifiers for a given set of frequent queries. However they are not applicable for rare queries. In this chapter we introduce a oneshot learning scheme which enables direct design of a classifier for novel queries, without requiring any access to the annotated training data. i.e., classifiers are trained for a set of frequent queries, and seamlessly extended for the rare and arbitrary queries, as and when required. We call this as one shot learning of query classifiers.

3.1.1 Related Work

Recognition free retrieval was attempted in the past for printed as well as handwritten document collections [81, 34, 19]. Primary focus has been on designing appropriate features (eg. Profiles, SIFT-BoW), distance functions (eg. Euclidean, Earth Movers), matching schemes (eg. Dynamic Programing). Since most of these methods were designed for smaller collections (few handwritten documents as in [81]), computational time was not a major concern. Methods that extended this to larger collection [87, 88] used mostly (approximate) nearest neighbor retrieval. Besides new approaches to handwritten document retrieval, keyword spotting as an alternative approach of indexing and retrieving handwritten documents has been proposed in [61], [82], [102]. The idea is to search the document for a certain keyword by feature matching instead of recognition.

Konidaris *et al.* [51] retrieve words from a large collection of printed historical documents. A search keyword typed by the user is converted into a synthetic word image which is used as a query image. Word matching is based on computing the L_1 distance metric between the query feature and all the features in the database. The ranked results are further improved by relevance feedback.

Sankar and Jawahar [87] suggested a framework of probabilistic reverse annotation for annotating a large collection of images. Word images were segmented from 500 Telugu books. Matching of the word images was done using the DTW approach [81]. Hierarchical agglomerative clustering was used to cluster the word images. Exemplars for the keywords are generated by rendering the word to form a keyword-image. Annotation involved identifying the closest word cluster to each keyword cluster.

Almazan *et al.* [5] applied word spotting in historical handwritten documents for 19 word classes. Their dataset comprised 50 pages of a single volume from the same writer. They use an indexing method to get the interesting regions (which possibly contain the relevant word image) and then use a discriminative model to rank these regions.

Yalniz and Manmatha [100] have applied word spotting to scanned English and Telugu books. They are able to handle noise in the document text by the use of SIFT features extracted on salient corner points. The SIFT features are quantized using hierarchical k-means to obtain visual terms. Image retrieval follows in 2 stages: first, the images having visterms in common with the given query are retrieved using the inverted index; second, a final ranking of the word images is done based on a similarity score which takes into account the spatial arrangement of the common visterms between the query word and each test image. Resilience to noise, like underlining, stray ink strokes, etc. is achieved because salient points in the uncorrupted portions of the image are available for feature extraction.

Rath and Manmatha [81] used vertical projection profile and upper and lower word profile features in a DTW based matching technique. They showed that DTW matching achieves an average precision of around 65% on a set of 10 pages of good quality.

Gatos and Pratikakis [34] have proposed a segmentation free word spotting methodology that can be applied to historical printed documents. Block-based document image descriptors are extracted and used in a template matching process satisfying invariance in terms of translation, rotation and scaling. The approach can work on low-quality documents. In [56], Leydiera *et al.* present another segmentation free approach which finds informative parts of a word image. Differential features are extracted based on zones of interest. A cohesive elastic matching method is used to match only the informative parts of the words.

In [80, 81, 10, 51, 14], segmentation based approaches were presented. In [80, 81] Rath *et al.* represent word image as a sequence and use a dynamic programming based approach for matching the sequences. Balasubramanian *et al.* [10] improve over this approach by designing a novel partial sequence matching strategy which takes care of morphological variations while matching. Bhardwaj *et al.* [14] perform word spotting over multilingual document collections. They index the moment based features of all the word images. A keyword-feature representation mapping is used to map query keyword to image feature space and retrieval is done based on the cosine distance between query and indexed word image features.

Rath *et al.* [83] proposed the first automatic retrieval system for historical handwritten documents. Their dataset consisted of 1000 manuscript pages from the George Washington collection. They used statistical models for handwritten word image and page retrieval. Holistic features extracted from the word image were discretized to extract 52 features terms per word. These words were drawn from a discrete feature vocabulary of size 494. Relevance of a text query to a page or to a word image was formulated as a probabilistic framework which learns the joint probability of the query word and the features. Rath and Manmatha [83] proposed an OCR-free approach to historical manuscript retrieval that learned the joint probability of the query word and features of the word image.

Rothfeder *et al.* [84] present a point correlation voting algorithm which compares whole word-images based on their appearance. This algorithm recovers correspondences of points of interest in two images, and then uses these correspondences to construct a similarity measure. This similarity measure can then be used to rank word-images in order of their closeness to a querying image. They achieved an average precision of 62.57% on a set of 2372 images of reasonable quality and an average precision of 15.49% on a set of 3262 images from documents of poor quality that are even hard to read for humans.

Manmatha *et al.* [62] presented techniques for matching words by modeling the transformation between words. Correspondence between the pixels of the 2 candidate images is recovered by aligning for translational shift or by affine transform. The distance between the 2 images is computed as the sum of the distance between the closest corresponding pixels after computing the best alignment. The translational shift was computed using Euclidean Distance Mapping (EDM) and the affine transform was computed using SLH [90] algorithm.

Cao and Govindaraju [19] proposed a vector model based method for retrieval of degraded medical form images. The word segmentation errors are modeled by segmentation probabilities and the posterior probability of word recognition is modeled as a Guassian function of the edit distance (similarity between the word image and the entry in the lexicon) produced by the word recognizer.

Ambati *et al.* [7] have indicated that many of the digitized books in the Indian scripts are available in [7] [1] [2]. These book collections have been used for testing the recognition free retrieval approaches. Sankar *et al.* [88] propose a *collection OCR* which can handle noisy word images. Books are segmented into words and features are computed over them. A random set of word images are collected for which text-labels are obtained. A hierarhical k-means (HKM) tree is build over the labeled dataset. An unlabelled word-image is looked up in the HKM tree and is assigned to the nearest cluster. The label of the cluster centroid is assigned to the test word-image. A small set of labelled samples can be used to annotate a large set of word images. They used Profiles+DFT feautures and a Euclidean distance based NN classifier and achieve an accuracy of around 80% for a label set of 1000 Telugu words.

Kim and Govindaraju [50] have presented a fast method of handwritten word recognition suitable for real time applications. The method uses word models and involves the lexicon early in the recognition process. The word image is compared with only words present in the lexicon thus eliminating any need for post processing. They evaluated the performance of their algorithm on 3000 postal words (city names, firm names, personal names, street names, and state names). The word recognition rate of the forms using Word Model Recognizer (WMR)[50] is as low as 20% as reported in [67].

Tulyakov and Govindaraju [98] present a model for word recognizers based on over-segmentation of input image and recognition of segment combinations as characters in a given lexicon image.

Saabni and El-sana [85] present a word matching algorithm based on Chamfer Distance to compute the similarity between shapes of word-parts. To compute the distance between two word-part images, the algorithm subdivides each image into equal-sized slices (windows). A modified version of the Chamfer Distance, incorporating geometric gradient features and distance transform data, is used as a similarity distance between the different slices. Finally, the distance between two images of word-parts is computed by applying the Dynamic Time Warping (DTW) to a series of windows sliding horizontally over the images

Researchers [12],[22] have shown that the performance of OCR text retrieval is badly affected when dealing with short or low quality documents. In [48], [68], [72] different approaches modeling typical recognition errors were proposed. In [68] a probabilistic model for misrecognition was proposed and this model was used to design the term-weighting scheme of information retrieval. The approach that generates candidate terms for each "true" search term and adds the retrieval results of candidate terms into the final result was studied in [72]. In [48], a language model that took common recognition errors into account was built. This language model can then be used to approximate an "uncorrupted" version of a particular document, and it can be used for retrieval in a language modeling approach.

3.1.2 Contributions

For searching complex concepts in large databases, SVMs have emerged as the most popular and accurate solution in the recent past [59]. For linear SVMs, both training and testing have become very fast with the introduction of efficient algorithms and excellent implementations. Methods like Pegasos [91] and whitening [41] make the offline training, and incremental/online training really fast. Training a classifier on the fly [20] is now considered as quite feasible for reasonably large image collections. (Indeed these are still not yet comparable to the indexing schemes popularly used in the text IR tasks, specially for huge data.) In this chapter,

1. We demonstrate that the SVM based word retrieval performance is superior to that of OCRs, and also the popular nearest neighbour based word spotting solutions.

2. We design a one shot learning scheme, which allows to generate a novel classifier for rare/novel query words without any training data.

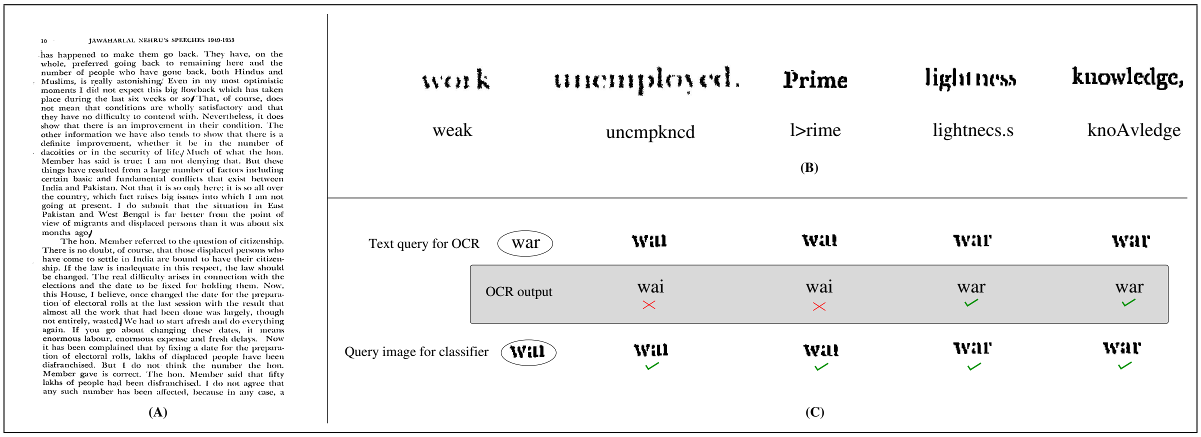
3. We demonstrate that with a simple retraining (with no extra supervision), the solution can adapt to a specific book or collection effectively.

4. We validate the performance on multiple books and demonstrate the qualitative and quantitative performance of the solution.

3.2 Accurate Classifiers for Frequent and Rare Queries

Our word-level retrieval scheme is a direct application of the SVM classifier. We train a linear SVM classifier with few positive examples and a set of randomly sampled negative examples. During retrieval, this classifier is evaluated over the dataset images, and a ranked list of word images is predicted.

For representing word images, we prefer a fixed length sequence representation of the visual content. i.e., each word image is represented as a fixed length sequence of vertical strips (of varying width



10 JAWAHARLAL NEHRU'S SPEECHES 1949-1953

has happened to make them go back. They have, on the whole, preferred going back to remaining here and the number of people who have gone back, both Hindus and Muslims, is really astonishing. Even in my most optimistic moments I did not expect this big flowback which has taken place during the last six weeks or so. That, of course, does not mean that conditions are wholly satisfactory and that they have no difficulty to contend with. Nevertheless, it does show that there is an improvement in their condition. The other information we have also tends to show that there is a definite improvement, whether it be in the number of dacoities or in the security of life. Much of what the hon. Member has said is true; I am not denying that. But these things have resulted from a large number of factors including certain basic and fundamental conflicts that exist between India and Pakistan. Not that it is so only here; it is so all over the country, which fact raises big issues into which I am not going at present. I do submit that the situation in East Pakistan and West Bengal is far better from the point of view of migrants and displaced persons than it was about six months ago.

The hon. Member referred to the question of citizenship. There is no doubt, of course, that those displaced persons who have come to settle in India are bound to have their citizenship. If the law is inadequate in this respect, the law should be changed. The real difficulty arises in connection with the elections and the date to be fixed for holding them. Now, this House, I believe, once changed the date for the preparation of electoral rolls at the last session with the result that almost all the work that had been done was largely, though not entirely, wasted. We had to start afresh and do everything again. If you go about changing these dates, it means enormous labour, enormous expense and fresh delays. Now it has been complained that by fixing a date for the preparation of electoral rolls, lakhs of displaced people have been disfranchised. But I do not think the number the hon. Member gave is correct. The hon. Member said that fifty lakhs of people had been disfranchised. I do not agree that any such number has been affected, because in any case, a

Figure 3.1 (A) A typical page from a document image from our data set. (B) OCRs make many errors. Examples of word images and the errors from a commercial OCR in a page. (C) Examples of a classifier based retrieval compared with OCR retrieval. OCR did not recognize the images marked with a cross, and failed to recall.

according to the aspect ratio of the word image). A set of features $\mathbf{f}_1, \ldots, \mathbf{f}_L$ where $\mathbf{f}_i \in R^M$ is the feature representation of the i^{th} vertical strip and L is the number of vertical strips. For an SVM classifier this can be considered as a single feature vector $\mathbf{F} \in R^d$ of size $d = LM$. However, we exploit the sequential nature of the feature representation for an on the fly synthesis of the novel classifiers in Section 3.2.2.1.

3.2.1 Efficient Classifier based Retrieval

Our classifier is basically a margin maximizing SVM classifier trained in a 1 vs rest setting. SVM gives maximum margin hyperplane separating the positive and negative instances. For a query word x_q, a SVM classifier $\mathbf{w}_q$ ($\mathbf{w}_q$ is the normal vector to the maximum margin hyperplane) is learned during the training, and for retrieval, database images are sorted based on the score $\mathbf{w}_q^T \mathbf{F}_i$. This evaluation is very efficient, since it requires only d multiplications and $d - 1$ additions. For frequent queries, this can in-fact be computed offline. However, traditional SVM implementations require many positive and negative examples to learn the weight vector $\mathbf{w}_q$.

In [59], Malisiewicz *et al.* proposed the idea of exemplar SVM (ESVM) where a separate SVM is learnt for each example. Almazan *et al.* [6] use ESVMs for retrieving word images. ESVMs are inherently highly tuned to its corresponding example. Given a query, it can retrieve highly similar word images. This constrains the recall, unless one has large variations of the query word available. Another demerit of ESVM is the large overall training time since a separate SVM needs to be trained for each exemplar. To tackle this issue, Gharbi *et al.* [35] provide a much faster alternative to train exemplar SVM. Assuming a Gaussian distribution over feature space, they give closed form expression for the normal vector to the Gaussian at query point x_q. This normal is given as $\Sigma^{-1}(x_q - \mu_0)$, where Σ is the global covariance matrix, μ_0 is global mean vector. This expression can also be interpreted in terms of linear discriminant classifiers as done by Hariharan *et al.* in [41]. We call the normal to this hyperplane as SVM weight and use this weight vector for retrieval. Generalized expression for LDA weights are given as $w = \Sigma^{-1}(\mu_+ - \mu_-)$ where μ_+ and μ_- are the means of the positive and negative examples respectively. This simple computation makes the training extremely efficient. It requires only few d^2 multiplications for the design of specific query classifiers. Also, the same method can be used independent of whether we have one example query or multiple examples from query class.

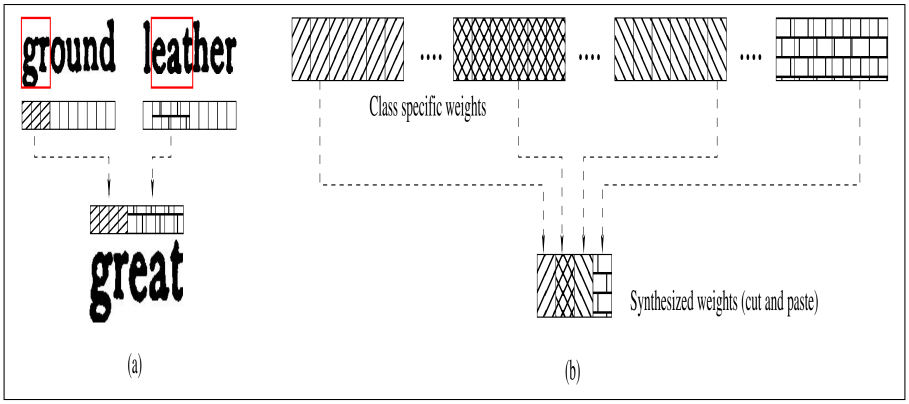


Figure 3.2 Synthesis of classifier for rare queries. (a) portions of the classifiers corresponding to "**gr**ound" and "l**eat**her" are joined to form a classifier for **great**. Note that the appropriate segments are automatically found. (b) In a general setting, a novel classifier gets formed from multiple constituent classifiers.

3.2.2 Classifier design for rare queries

The number of possible words in a language can be infinite. It is not practical to build classifiers for all the possible words. However, on a closer look, we realize that all these words are composed from a very small number of characters, and a reasonably small set of *ngrams*. In many practical applications related to text processing, a finite set of *ngrams* were used to cover the vocabulary, and the small vocabulary solutions were extended to unlimited vocabulary settings. In this chapter, we show that the SVM classifiers corresponding to the *ngrams* can be effectively composed to generate novel classifiers on the fly. Such synthesized classifiers, by simply concatenating *ngram* classifiers, could be inferior to the directly built classifiers. (i) Due to the nature of scripts and writing styles, the joining will not be ideal. One should prefer larger grams for better synthesis. (ii) Classifiers for smaller *ngrams* could be noisy. Since the classifiers need to be built for all the words, the overall performance could thus be poor. We address these limitations as follows. It is well known that the queries in any search engine follow an exponentially decaying structure (Zipf's law) like the frequency of occurrence of words in a language. We build SVM classifiers (Section 3.3) for the most frequent queries and use classifier synthesis only for rare queries. This improves the overall performance. When the synthesized classifiers are not as good as the originl one, we further use query expansion (QE) for the refinement of the query classifier (See Section 3.3.2). Our method does not build artificial *ngram* classifiers; we use complete word classifiers and dynamically decide what portions from what words to be cut and pasted to create the novel classifier. We refer to this solution as a Direct Query Classifier (DQC) design scheme. In Section 3.2.2.1, we describe DP DQC, a dynamic programming based approach for DQC synthesis.

3.2.2.1 DP DQC: DQC **Design using Dynamic Programming**

Given a set of linear classifiers $\mathcal{W}_w = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$ for most frequent N queries and a query feature vector $\mathbf{x}_q$, we would like to synthesize a novel classifier $\mathbf{w}_q$ as a piece-wise fusion of the parts from the available classifiers from $\mathcal{W}_w$ (See Figure 3.2). Let us assume that there are p portions that we need to select to form a novel classifier. Intervals (portions) are characterized by the sequence of indices a_1, ... a_{p+1} where $a_1 = 1$ and $a_{p+1} = L$. We formulate the classifier synthesis problem using a set of already available linear classifiers, as that of finding the optimal solution to

$$\max_{\{a_i\},\{c_i\}} \sum_{i=1}^{p} \sum_{k=a_i}^{a_{i+1}} w_{c_i}^k x_q^k \tag{3.1}$$

where w_{c_i} corresponds to the c_i^{th} classifier that we choose and the inner summation applies the index range (a_i, a_{i+1}) to use a portion from the classifier c_i and the index i in the outer summation identifies the classifier selected for each portion and p is the total number of portions we need to consider. The solution to the problem requires picking up the optimal set $\{c_i\}$ and the set of segment indices $\{a_i\}$ such that the $\{a_i\}$ form a monotonically increasing sequence of indices.

We use LDA as the linear classifier in our DQC solution. LDA weight $\mathbf{w}_q$ is given as

$$\mathbf{w}_q = \Sigma^{-1}(\mu_q - \mu_0) \tag{3.2}$$

where Σ and $\boldsymbol{\mu}_\mathbf{0}$ are covariance and mean computed over the entire dataset of word images. Since Σ and μ_0 are common for all classes, synthesizing w_q requires finding mean vector (μ_q) for unknown query class. We consider the problem of finding μ_q for the (unknown) query class as the classifier synthesis problem outlined above. Let the set of mean vectors of frequent words be defined as $\mathcal{W}_\mu = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_N\}$. We divide query vector x_q into p fixed length portions and match each cut portion x_q^k to the most similar feature portion (say $\boldsymbol{\mu}_q^k$) of equal length from the set $\mathcal{W}_\mu$. We solve the problem of selecting the optimal set $\{c_i\}$ by solving for the optimal alignment of the query feature portion x_q^k with the best matching portion $\mu_{c_i}^k$ of the mean vector μ_{c_i} picked from $\mathcal{W}_\mu$. We then combine each of the obtained $\boldsymbol{\mu}_q^k$ to compose the mean vector $\boldsymbol{\mu}_q$ for the query. We ensure monotonicity of the sequence $\{a_i\}$ by using a fixed sequence for $\{a_i\}$, thus avoiding optimization over the set of indices $\{a_i\}$.

The alignment of feature portions is done using subsequence Dynamic Time Warping(DTW) [70], which is a dynamic programming (DP) algorithm. The DTW takes care of the variability in the different instances (word images) of the same class. The time complexity of subsequence DTW is $O(l_1\, l_2)$, where l_1 and l_2 are the length of the two sequences given as input. In our case, the l_1 is the length of each small segment and $l_2 = N \cdot d$ is the length of the concatenated sequence of all the mean vectors in the set $\mathcal{W}_\mu$. If we use all the known mean vectors in the set $\mathcal{W}_\mu$, synthesizing the classifier for a single query could take long time. To reduce the time complexity, we compute the normalized dot product between the query vector and all the mean vectors of the known classes. We select the 10 most similar mean vectors and use them in the subsequence DTW.

3.3 Efficient and Accurate Retrieval

When a direct classifier is used for the frequent words, retrieval is efficient since this requires only the evaluation of the classifiers. In practice, these are also pre-computed. For the rare words, we use the DQC classifier which requires a DP based selection from multiple composite classifiers. This affects the efficiency and accuracy of the solution to some extent. We now discuss two refinements to the solution which can improve the efficiency and accuracy of the retrieval, NN DQC, which is an approximate nearest neighbor based implementation of DQC , and use of query expansion for adapting DQC to a previously unseen word collection without using any new training data.

3.3.1 NN DQC: DQC with Nearest Neighbor indexing

DP DQC synthesis is slow because it aligns the query vector portions with the mean vectors of the known classes. We can obtain a speed-up by using approximate nearest neighbor search instead of using DTW based alignment . This, in principle, compromises the optimality of the synthesis, however, in practice, it does not affect the quality of the classifier. We consider fixed portions (length R) of the mean vectors of all the known word classes and build an index over all such portions using FLANN [69].

The query $\mathbf{x}_q$ is also divided into portions of fixed length R and the approximate nearest neighbor match of each of these portions with the indexed portions is found using FLANN. The so obtained nearest neighbors are concatenated to give the mean vector $\boldsymbol{\mu}_q$, which is then used in Equation (3.2), to compute the LDA weight $\mathbf{w}_q$. FLANN has a time complexity $O(RBD)$ when using a hierarchical k-means for indexing, where B is the branching factor and D is the depth of the tree. This time complexity $O(RBD)$ is typically much smaller than $O(NRd)$ of subsequence DTW. Hence DQC using NN is much faster than using subsequence DTW.

3.3.2 Query expansion for DQC

Classifiers which are trained on one data set need not perform well on another data set due to the print and style variations. For adapting the query to a new collection, query expansion (QE) is used. We implement QE very efficiently. Query expansion is a concept used in information retrieval where the seed or primary query is reformulated to improve the retrieval performance. We use QE to further improve the performance of DQC . An index is built over all the database vectors and those vectors similar to query vector $\mathbf{x}_q$ are identified by performing approximate nearest neighbor search over the

Dataset					Partitioning		
Dataset	Language	Source	Type	#images	Training Set	Validation Set	Test Set
D1	English	1 Book	Clean	32371	11157	10472	10742
D2	English	2 Books	Clean	37376	12805	12454	12117
D3	English	1 Book	Noisy	3782	1321	1207	1254
D4	Telugu	1 Book	Clean	27980	9719	8955	9306
D5	Malayalam	1 Book	Clean	19246	6758	6112	6376

Table 3.1 Table gives details of the datasets.

index. The top k vectors closest to the query vector are averaged to give the reformulated query vector to be used in the DQC. We fixed $k = 5$ by using a validation dataset as explained in section 3.4. The reformulated query better captures the variations of the query class. We use FLANN for getting the approximate nearest neighbors, thus incurring an additional cost of $O(MBD)$ where M is the number of vertical strips in the word image.

This is much smaller compared to the time complexity $O(NRd)$ for subsequence DTW matching. Hence, adding QE step before DQC does not cause significant increase in computation time. However, it improves the accuracy.

3.4 Experiments, Results and Discussions

In this section, we validate the DQC classifier synthesis method on multiple word image collections and also demonstrate its quantitative and qualitative performance.

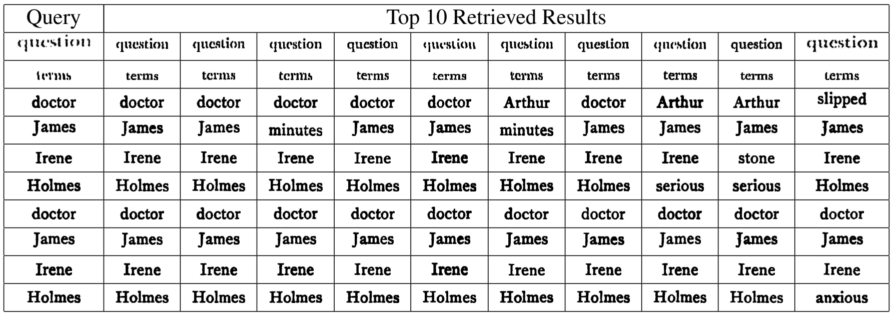
3.4.1 Data Sets, Implementation and Evaluation Protocol

Our datasets, detailed in Table 3.1, comprises scanned English as well as Telugu and Malayalam books from a digital library collection. We manually created ground truth at word level for the quantitative evaluation of the methods. The first collection (D1) of words is from a book which is reasonably clean. On this collection, commercial OCR (ABBYY Fine Reader 9.0) provides very high word accuracy. We use this collection to demonstrate that our method provides satisfactory performance on collections where the OCR is satisfactory. Second dataset (D2) is larger in size and is used to demonstrate the performance in case of heterogeneous print styles. Third data set (D3) is a noisy book, and is used to demonstrate the utility of the performance of our method in degraded collections. Fourth and fifth dataset (D4 and D5) are books from Telugu and Malayalam scripts and these are used to show that our approach is script independent. OCR [8] gives poor results for these datasets in comparison to the English datasets.

For the experiments, we extract profile features [81] for each of the word images. Profile features comprise of the following: (i) Vertical projection profile, which counts the number of ink pixels in each column (ii) Upper and lower word profile, which encode the distance between the top (lower)

Dataset	#queries	OCR	NN	LDA	SVM	ESVM	NN DTW
D1	100	0.97	0.94	0.99	1	0.98	0.88
D2	100	0.93	0.80	0.98	1	0.92	0.75
D3	100	0.89	0.77	1	0.94	0.93	0.79
D4	100	0.45	0.79	0.98	1	0.92	0.77
D5	100	0.78	0.97	1	1	0.97	0.92

Table 3.2 Table shows a comparison of various word retrieval schemes such as nearest neighbor (NN), LDA, SVM, EXEMPLAR SVM, DTW based NN with that of OCR based retrieval. Classifier based methods (specially SVM based methods) are much superior to the OCR based solution. Performance is reported as MAP.



Query	Top 10 Retrieved Results									
question	question	question	question	question	question	question	question	question	question	question
terms	terms	terms	terms	terms	terms	terms	terms	terms	terms	terms
doctor	doctor	doctor	doctor	doctor	doctor	Arthur	doctor	Arthur	Arthur	slipped
James	James	James	minutes	James	James	minutes	James	James	James	James
Irene	Irene	Irene	Irene	Irene	Irene	Irene	Irene	Irene	stone	Irene
Holmes	Holmes	Holmes	Holmes	Holmes	Holmes	Holmes	Holmes	serious	serious	Holmes
doctor	doctor	doctor	doctor	doctor	doctor	doctor	doctor	doctor	doctor	doctor
James	James	James	James	James	James	James	James	James	James	James
Irene	Irene	Irene	Irene	Irene	Irene	Irene	Irene	Irene	Irene	Irene
Holmes	Holmes	Holmes	Holmes	Holmes	Holmes	Holmes	Holmes	Holmes	Holmes	anxious

Figure 3.3 Figure shows few query words and corresponding top 10 retrieved results. First two rows show frequent query results. Row 3 - Row 6 show rare query results using DP DQC . Row 7 - Row 10 show rare query results using DP DQC with QE.

boundary and the top-most (lower-most) ink pixels in each column. (iii) Background/Ink transition counts the number of background to ink transitions in each column. All these features are extracted for 100 vertical strips. This results in a 400 dimensional representation for every word image. The features are normalized to $[0, 1]$ so as to avoid dominance of any specific feature. Instead of the query log of a search engine, which lists the frequent queries and rare queries, we have considered the frequency of occurrence of the words in the collection. We report the mAP score for 100 frequent queries as the retrieval performance measure in Table 3.2. In Table 3.2, we compare the performance of various methods as mAP of the retrieval. Some of the salient observations out of this experiment are:

1. OCR performance is inferior to the SVM based retrieval in all the cases.

2. Faster approximation in the form of LDA with multiple positive examples is quite close to the performance of SVM .

3. ESVM performs inferior due to the use only one example

4. Nearest neighbor methods (with DTW, Euclidean etc.) are inferior, in general, to SVM .

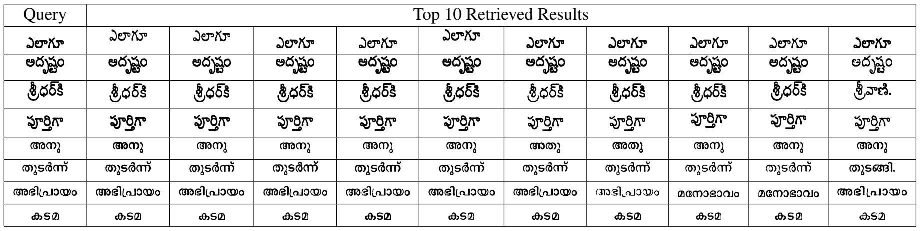

Query
Top 10 Retrieved Results

Figure 3.4 Figure shows few query words and corresponding top 10 retrieved results. First 4 rows show results for Telugu script and last 4 rows show results for Malayalam script

Dataset	queries	Freq	Freq(QE)	Rare		Rare(QE)		Overall	
				DP	NN	DP	NN	DP	NN
D1	100	0.99	0.99	0.82	0.85	0.91	0.90	0.98	0.98
D2	100	0.98	0.98	0.84	0.82	0.87	0.87	0.97	0.97
D3	100	1	1	0.71	0.73	0.80	0.82	0.98	0.98
D4	100	0.98	0.98	0.77	0.76	0.78	0.82	0.96	0.96
D5	100	1	1	0.95	0.96	0.96	0.95	1	1

Table 3.3 Retrieval performance: mAP scores for the evaluation of frequent and rare queries. For rare queries, mAP scores are given for DP DQC as well as NN DQC. Notice that QE improves the performance significantly.

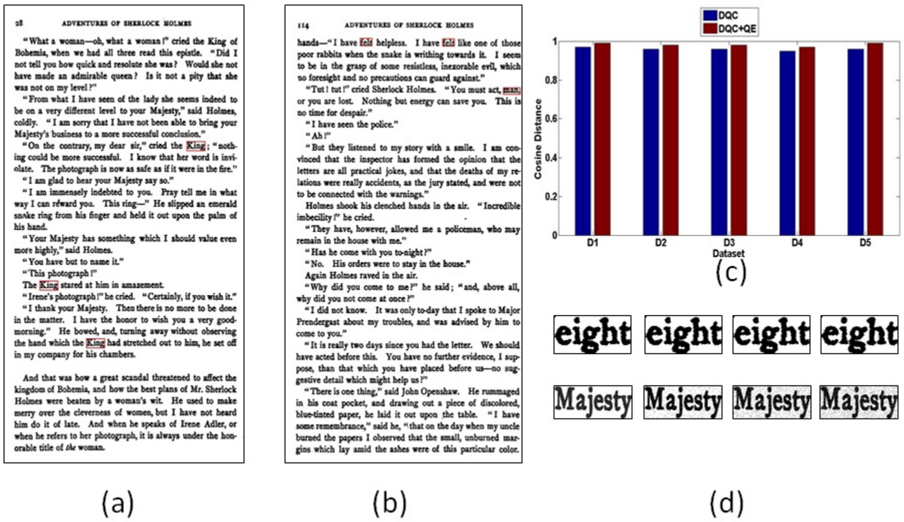
5. Nearest neighbour with DTW may be considered as equivalent to the word spotting.

6. For Telugu and Malayalam, OCR performs much worse than all other techniques.

Therefore, one could observe that this classifier based retrieval is superior to OCR and the DTW based retrieval. Note that DTW cannot be directly used in the SVM classifier since the corresponding kernel will not be positive semidefinite, and more over due to the computational complexity. Fig 3.1 depicts some of the qualitative examples of the retrieval. As can be seen that the classifier-based retrieved images are more relevant to the query. Classifiers are less sensitive to the degradations (eg. cuts, merges etc.) present in the word images. Figure 3.4 gives more qualitative examples of the retrieval for English datasets. We show examples of classifier based retrieval, and DQC based retrieval. Figure 3.4 gives qualitative results for Telugu and Malayalam. We also show in Table 3.3 that the quality of the retrieval improves with QE.

3.4.2 Performance of the DQC

In the initial experiment, we built classifiers for most frequent 1000 word categories. For the rest of the words, we find that a DQC based solution is appropriate. We implement the DQC based solution as discussed in the previous section. During the DQC evaluation, we discard the trained classifiers



28 ADVENTURES OF SHERLOCK HOLMES

"What a woman—oh, what a woman!" cried the King of Bohemia, when we had all three read this epistle. "Did I not tell you how quick and resolute she was? Would she not have made an admirable queen? Is it not a pity that she was not on my level?"

"From what I have seen of the lady she seems indeed to be on a very different level to your Majesty," said Holmes, coldly. "I am sorry that I have not been able to bring your Majesty's business to a more successful conclusion."

"On the contrary, my dear sir," cried the King; "nothing could be more successful. I know that her word is inviolate. The photograph is now as safe as if it were in the fire."

"I am glad to hear your Majesty say so."

"I am immensely indebted to you. Pray tell me in what way I can réward you. This ring—" He slipped an emerald snake ring from his finger and held it out upon the palm of his hand.

"Your Majesty has something which I should value even more highly," said Holmes.

"You have but to name it."

"This photograph!"

The King stared at him in amazement.

"Irene's photograph!" he cried. "Certainly, if you wish it."

"I thank your Majesty. Then there is no more to be done in the matter. I have the honor to wish you a very good-morning." He bowed, and, turning away without observing the hand which the King had stretched out to him, he set off in my company for his chambers.

And that was how a great scandal threatened to affect the kingdom of Bohemia, and how the best plans of Mr. Sherlock Holmes were beaten by a woman's wit. He used to make merry over the cleverness of women, but I have not heard him do it of late. And when he speaks of Irene Adler, or when he refers to her photograph, it is always under the honorable title of *the* woman.

114 ADVENTURES OF SHERLOCK HOLMES

hands—"I have felt helpless. I have felt like one of those poor rabbits when the snake is writhing towards it. I seem to be in the grasp of some resistless, inexorable evil, which no foresight and no precautions can guard against."

"Tut! tut!" cried Sherlock Holmes. "You must act, man, or you are lost. Nothing but energy can save you. This is no time for despair."

"I have seen the police."

"Ah!"

"But they listened to my story with a smile. I am convinced that the inspector has formed the opinion that the letters are all practical jokes, and that the deaths of my relations were really accidents, as the jury stated, and were not to be connected with the warnings."

Holmes shook his clenched hands in the air. "Incredible imbecility!" he cried.

"They have, however, allowed me a policeman, who may remain in the house with me."

"Has he come with you to-night?"

"No. His orders were to stay in the house."

Again Holmes raved in the air.

"Why did you come to me?" he said; "and, above all, why did you not come at once?"

"I did not know. It was only to-day that I spoke to Major Prendergast about my troubles, and was advised by him to come to you."

"It is really two days since you had the letter. We should have acted before this. You have no further evidence, I suppose, than that which you have placed before us—no suggestive detail which might help us?"

"There is one thing," said John Openshaw. He rummaged in his coat pocket, and drawing out a piece of discolored, blue-tinted paper, he laid it out upon the table. "I have some remembrance," said he, "that on the day when my uncle burned the papers I observed that the small, unburned margins which lay amid the ashes were of this particular color.

Figure 3.5 Fig(a) shows page retrieval for query ”king”. Fig(b) shows page retrieval with multiple query words. Here query words are ”man” and ”felt”. Fig(c)shows effect of query expansion on weights synthesized for 100 queries each from the five datasets. Blue bars are for DQC weights and Red bars are for DQC+QE weights. Fig(d) shows few example images over which noise was added artificially and query was performed.

and mean vectors for the chosen query word classes. DQC synthesizes the mean vector for the query, to be used to compute the LDA classifier. The DQC identifies the 10 most similar mean vectors (of the remaining 900 word classes) using the normalized dot product value. Subsequence DTW is used to find the best alignment of the cut-portions of the query feature vector with the concatenated mean vectors of the closest 10 word classes. We observe that the DQC performs somewhat inferior to the true classifiers designed with multiple examples. This is partly due to the joining process associated. In our implementation, we simply concatenated the segments coming from multiple classifiers. We consider 100 frequent and 100 rare queries for the evaluation and show the results in Table 3.3. One can notice that the method is superior for frequent queries. In both the cases, one can see that the query expansion improves the performance. Improvement is significant in the case of rare queries.

C	D1				D2				D3			
	mAP		Query Time		mAP		Query Time		mAP		Query Time	
	DP	NN	DP	NN	DP	NN	DP	NN	DP	NN	DP	NN
1	0.80	0.79	329	27	0.77	0.75	299	28	0.61	0.61	323	14
10	0.82	0.82	1340	15	0.77	0.78	1426	16	0.62	0.65	1330	5
20	0.82	0.84	1381	13	0.81	0.79	1461	11	0.61	0.66	1376	3
30	0.81	0.82	1399	10	0.75	0.78	1397	10	0.60	0.65	1454	3
40	0.79	0.80	1261	9	0.72	0.69	880	9	0.57	0.67	1401	2

Table 3.4 Table shows change in retrieval performance with change in cut length. mAP values and Average Query Time values are given for both DP DQC and NN DQC. Average Query time is given in milliseconds.

C	D4				D5			
	mAP		Query Time		mAP		Query Time	
	DP	NN	DP	NN	DP	NN	DP	NN
1	0.82	0.81	327	25	0.95	0.93	323	27
10	0.80	0.84	1290	15	0.93	0.95	1340	13
20	0.80	0.85	1438	10	0.95	0.94	1369	7
30	0.79	0.85	1396	9	0.94	0.93	1422	7
40	0.75	0.77	1138	8	0.94	0.91	1397	6

Table 3.5 Table shows change in retrieval performance with change in cut length. mAP values and Average Query Time values are given for both DP DQC and NN DQC. Average Query time is given in milliseconds.

Dataset	# queries	Cosine Distance	RMSE
D1	100	0.98	0.09
D2	100	0.98	0.09
D3	100	0.96	0.11
D4	100	0.97	0.09
D5	100	0.99	0.03

Table 3.6 Table gives comparison between DQC weights and actual weights for 100 rare queries. Cosine distance and RMSE are used for comparison between the two.

When the vocabulary covers 90% of the language, (which is relatively small for languages like English), the overall AP can be estimated as a weighed combination of these two mAPs. Please note that these are pure estimates and the actual measurement can be done only with a reasonably large query log. However, it can be seen that the performance of the system is quite competitive and also outperforms the OCR in many situations. In Figure 3.5, we show few page level retrieval results for single word query and multiple word query. In this figure, we also show results for similarity of the synthesized DQC weights to the corresponding original weights by measuring cosine distance between the two. To observe how DQC performs in presence of noise, we introduce noisy versions of few query words into the dataset and perform DQC retrieval over these query words. We degrade the images by adding Gaussian noise and by randomly performing flips for edge pixels of the words(fliping is done by randomly changing some of the foreground edge pixels to background pixels and vice-versa). We observe mAP of 0.81 for these noisy queries. Few sample images from the noisy set are shown in Fig3.5(d). Performance of DQC also depends upon the length of portions into which query vector is divided. In Table 3.5, we report the effect of variation of cut length on retrieval performance. For smaller cut lengths, mAP is relatively less.

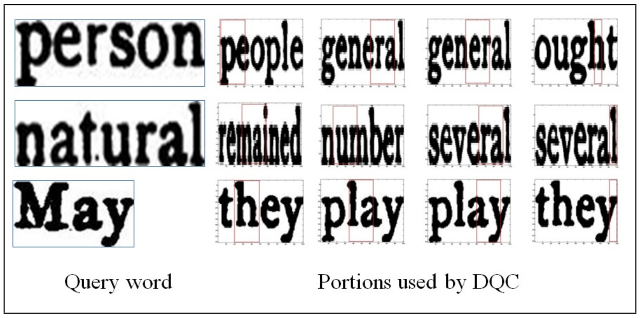


Figure 3.6 Figure shows few query images and corresponding portions used in generating DP DQC classifier.

To measure the goodness of the synthesized classifier, we compare it with the actually trained LDA classifier already available with us for the rare words we used. Table 3.6 shows the averaged cosine distance and average root mean squared error between the (DQC and trained) classifier weight vectors over the 100 queries. The cosine distance between actual weight w_q and synthesized weight $\hat{w}_q$ can be given as $\cos\theta = \frac{w_q^\top \hat{w}_q}{\|w_q\|\|\hat{w}_q\|}$. RMSE is the averaged root means square error between the weight vectors. It can be calculated by the formula $RMS = \sqrt{\frac{\sum_{i=1}^{d}(w_i - \hat{w}_i)^2}{d}}$.

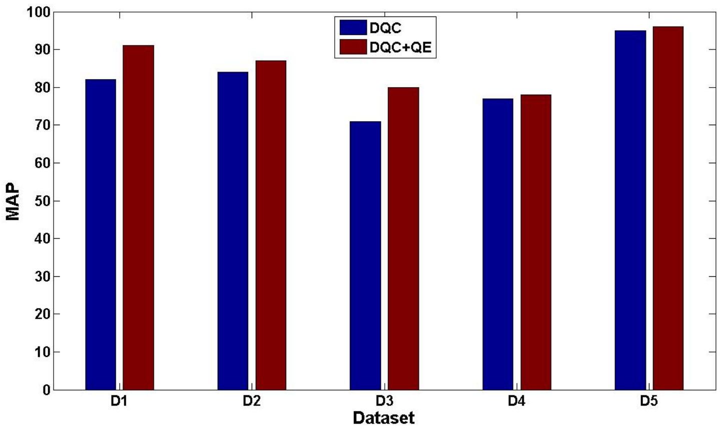


Figure 3.7 Barplot shows retrieval performance of DQC for rare queries. Improvement in mAP values with QE can be observed for all three datasets.

In Fig 3.6, we show a few queries and the corresponding portions used in synthesizing DQC . We also compare effect applying QE with DQC on all three datasets. We give corresponding results in Figure 3.7. We observe that in all three cases, QE further improves performance DQC . When QE is used with DQC , classifier performance also depends upon how many nearest neighbors (k)are used for reformulating the

Dataset	k=1		k=5		k=10		k=15	
	DP	NN	DP	NN	DP	NN	DP	NN
D1	0.79	0.79	0.84	0.84	0.70	0.72	0.59	0.62
D2	0.75	0.74	0.75	0.79	0.63	0.67	0.60	0.62
D3	0.64	0.65	0.78	0.74	0.79	0.66	0.74	0.63
D4	0.77	0.81	0.75	0.75	0.52	0.57	0.43	0.46
D5	0.93	0.94	0.91	0.91	0.70	0.72	0.61	0.63

Table 3.7 Table shows change in retrieval performance upon change in k (nearest neighbors) which is used in QE. mAP values are given for DP DQC as well as NN DQC

query. To determine k for each dataset, we vary k and observe retrieval performance over a validation dataset. We give corresponding results in Table 3.7 .

3.4.3 Discussions

Since DQC does not use any script specific information, our method can be used for any script. We have shown retrieval results for frequent and rare queries from English as well as two Indian scripts, Telugu and Malayalam. Our system can perform page retrieval and support multiple keyword queries. Page retrieval is performed based on the score given by query weight to different word images present in the page. Location of word in page can be marked using ground truth information. To deal with multiple query keywords, weights corresponding to all the keywords are obtained and multiple resulting lists are combined based on score of word images. Images showing page retrieval and multiple keyword query are given in Figure 3.5. To facilitate faster retrieval for frequent queries, we evaluate SVM weights of frequent queries over all the word images and store the resulting index. Hence, classifiers need to be evaluated only for the rare queries.

3.5 Summary

In this chapter, we have described a classifier based retrieval scheme for effectively retrieving word and document images from a collection. We argue that the classifier based method is superior to the OCR in practice. We introduce a novel classifier synthesis scheme which enable design of classifiers without any explicit training data. for this we exploit the fact that words in a language can be formed from fewer combination of character sequences. A major disadvantage of the classifier based scheme is the difficulty in indexing, which is important if the method needs to scale to millions of document images.

Chapter 4

Synthesizing classifiers for Novel styles

In this chapter, we investigate 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.

4.1 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 [63] 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 [63]. In some cases, a style clustering [21, 94] 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 [44] or the cost of alignment of two feature vector sequences with a Dynamic Time Warping (DTW) [81].

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

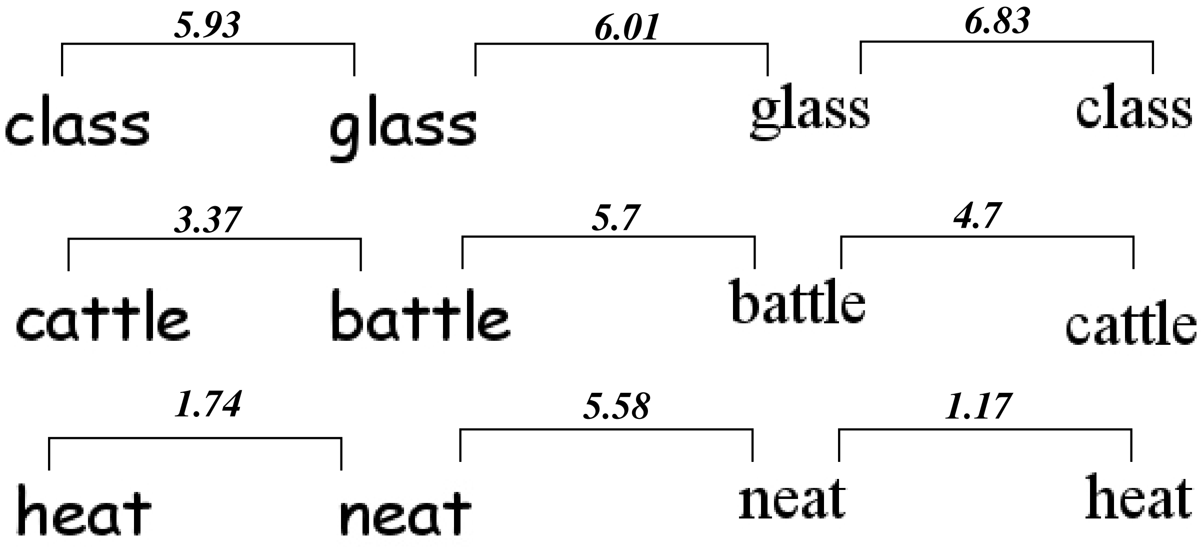


Figure 4.1 Euclidean distance between profile feature representation of pairs of word images. Euclidean distance could be affected more by font variation than a difference in underlying word labels, for example, distance between “battle” in the two fonts is more than “battle” and “cattle” in the same font.

chapter, 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 4.1, 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. 4.2).

Many efficient approaches for word image retrieval has been proposed in the recent past. Rath and Manmatha [81], as well as Meshesha and Jawahar [65] 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 [53]. 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 [76], and it has been widely used in applications like handwriting recognition [21, 103], face pose classification [95] etc. Transfer learning may involve (i) Feature transformations, e.g. updating the regression matrix [55], updating the LDA transformation matrix [47] (ii) Classifier adaptation, e.g. Retraining strategy for neural network [64], SVM [49], 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 [33, 17] to learn the style context in a group of patterns.

4.1.1 Contributions

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 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.

4.2 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 [81], [92] is one such popular feature. Profile features are considered to be reasonably robust to font variations (however see Figure 4.1). 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 [81] 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. 4.2.1.

4.2.1 Style Transfer

Style transfer strategy has been used in the past for handwriting recognition. Connell and Jain [21] do a general to specific adaptation of their model using few examples of handwritten words from each

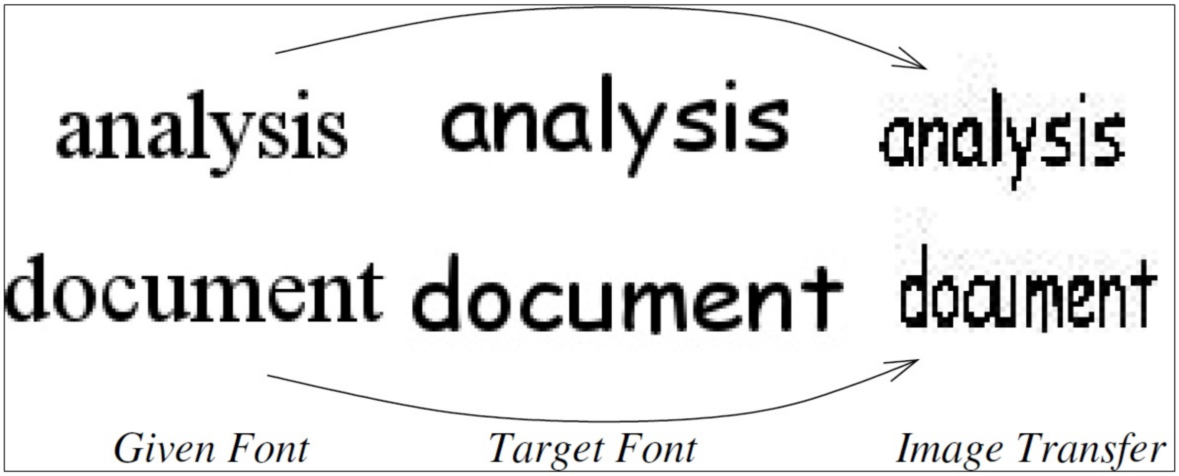


Figure 4.2 Application of bilinear model for transferring image from one font to another is shown. Content vectors corresponding to word images in the first font are transferred to the second font using style vectors of the second font.

user. This results in a specific model for each user. Zhang and Liu [103] 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 [95]. 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 [95], we represent the query observation y^{sc}, in style s and content c, as

$$y^{sc} = A^s b^c, \tag{4.1}$$

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 4.2. 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.

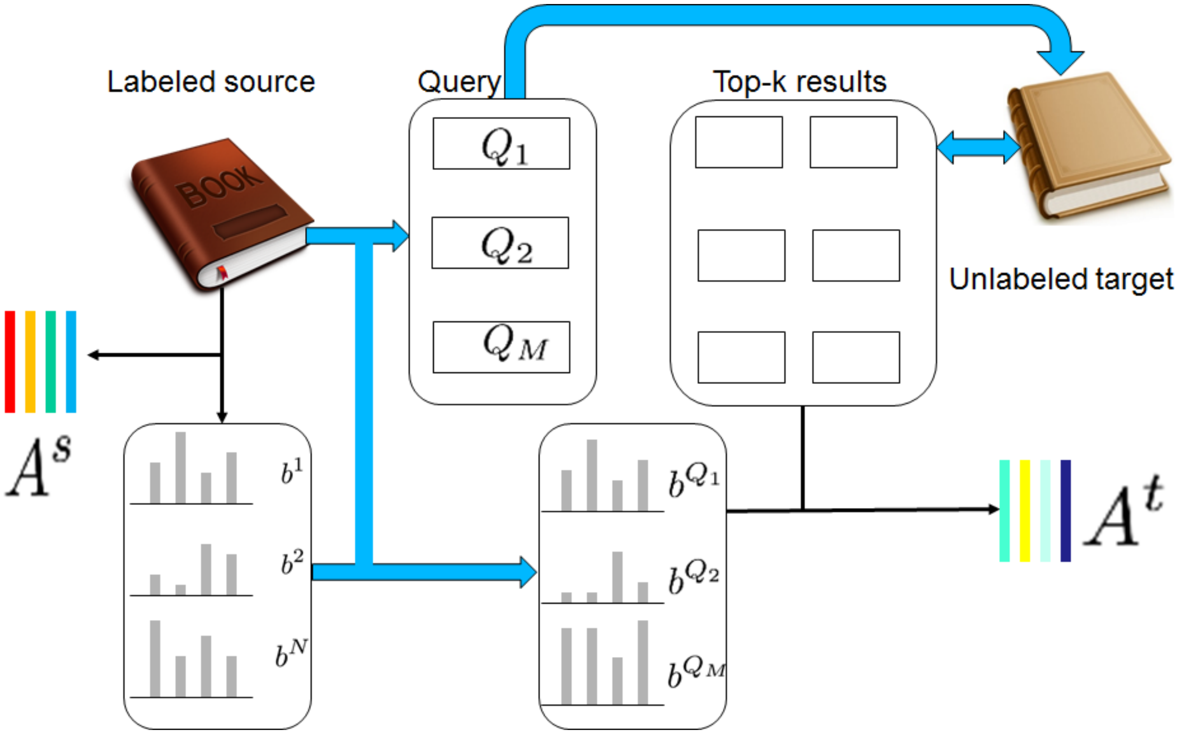


Figure 4.3 Figure shows the overview of SSST. Nearest neighbor search is used for label propagation. Content vectors are also propagated to the target domain. Style vectors corresponding to target domain is using the propagated content vectors and the target domain images.

4.3 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. 4.2.1 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

$$\min_{A^s,B^c} ||Y^s - A^s B^c||_F^2 \,. \tag{4.2}$$

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

$$\min_{A^r} ||Y^r - A^r B^r||_F^2 + \lambda \, ||A^r - A^s||_F^2 \,. \tag{4.3}$$

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 [81] 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 4.5) 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 4.1.

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 4.3 to obtain A^r.

6. Once we have obtained A^r, we use Equation 4.1 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.

In figure 4.3, we present the above steps for doing semi-supervised style transfer. 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.

4.4 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 $\phi : R^n \to H$ 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 [89]. This technique is known as the kernel trick and has been widely used for obtaining nonlinear versions of PCA [89], LDA [66] 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

$$\phi(y^{tc}) = \phi(Y^t)\alpha b^c. \tag{4.4}$$

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

$$\min_{A^t,B^t} \text{Trace}(\left|\left|\phi(Y^t) - A^t B^t\right|\right|^2 + \beta A^{t^T} A^t). \tag{4.5}$$

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

$$\min_{\alpha,B^t} \text{Trace}(\mathcal{K} - B^{t^T}\alpha^T\mathcal{K} - \mathcal{K}\alpha B^t + B^{t^T}\alpha^T\mathcal{K}\alpha B^t + \beta\alpha^T\mathcal{K}\alpha). \tag{4.6}$$

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 [39], [38] 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 4.6. This gives us the coefficient matrix α and the content matrix B^t. Any observation in the feature space can now be represented as $\phi(y^{tc}) = \phi(Y^t)\alpha b^c$.

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 $\min_{b^{ir}} \left|\left|\phi(y^{ir}) - \phi(Y^t)\alpha b^{ir}\right|\right|^2$, 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.

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

Table 4.1 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.

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.

4.5 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. 4.3.

3. Asymmetric kernel bilinear model as discussed in Sec. 4.4.

4.5.1 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 4.1, 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 4.4. Bilinear models are learned from the examples in training set. Optimal value for kernel parameters

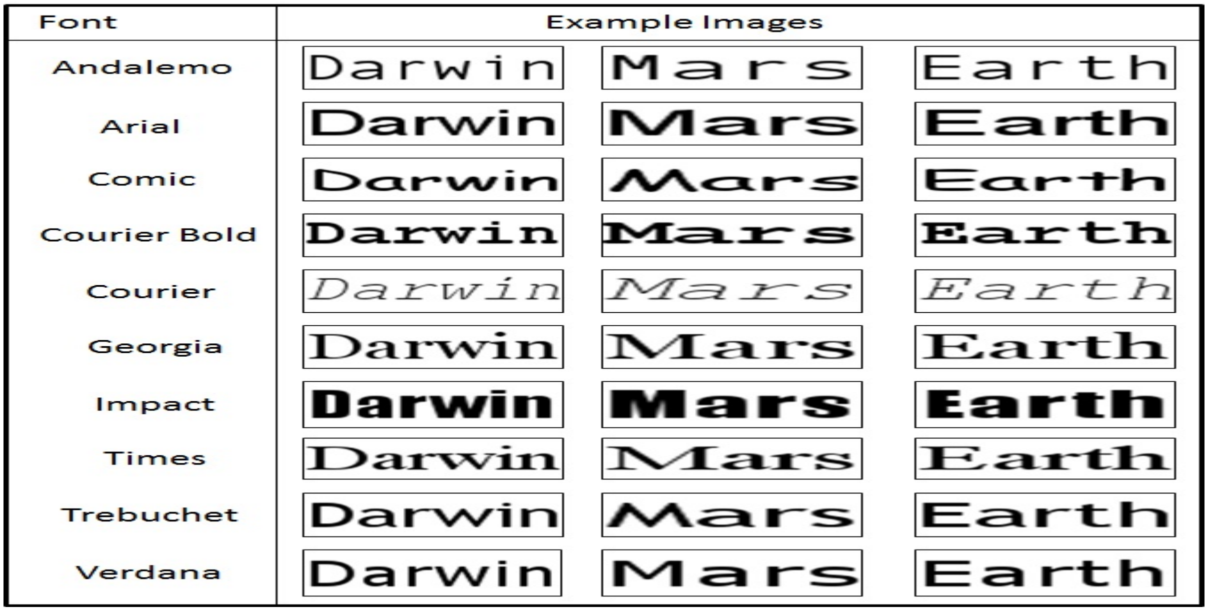

Font
Example Images
Andalemo
Arial
Comic
Courier Bold
Courier
Georgia
Impact
Times
Trebuchet
Verdana
Darwin
Mars
Earth

Figure 4.4 Examples from each of the 10 fonts used in the Dlab.

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 [81] comprising of:

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.

4.5.2 Retrieval Experiments

In Table 4.2, 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

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 4.2 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.

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 4.5, 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 4.6, we show few query 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

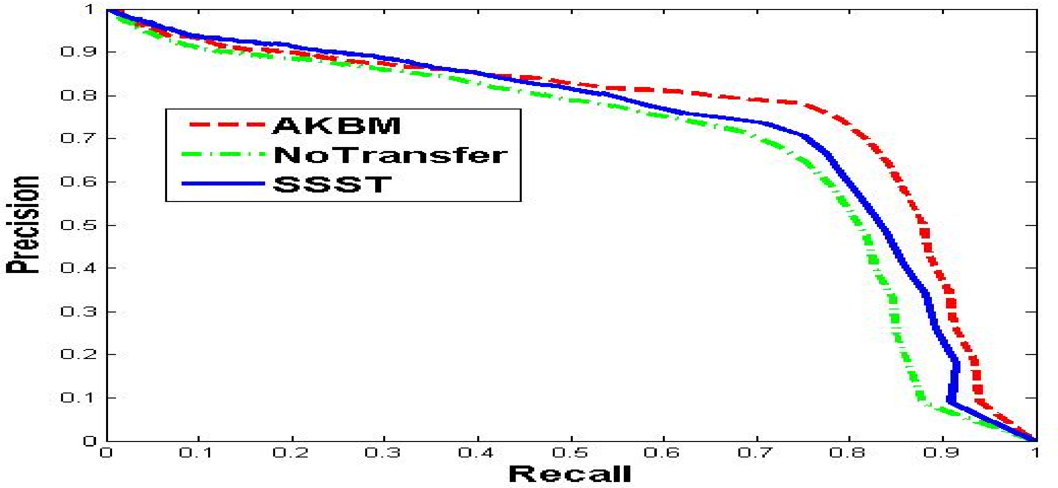


Figure 4.5 Precision- Recall (PR) curves corresponding to 100 queries is given. For training and target dataset, two datasets are picked from D1 - D4.

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 4.3 Retrieval performance on D5.

dataset. We perform retrieval on D5 while training on one of the datasets D1 to D4 every time. We report the results in Table 4.3. 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.

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 4.7. For the retrieval experiment, query image is picked from one of the fonts and retrieval is performed on

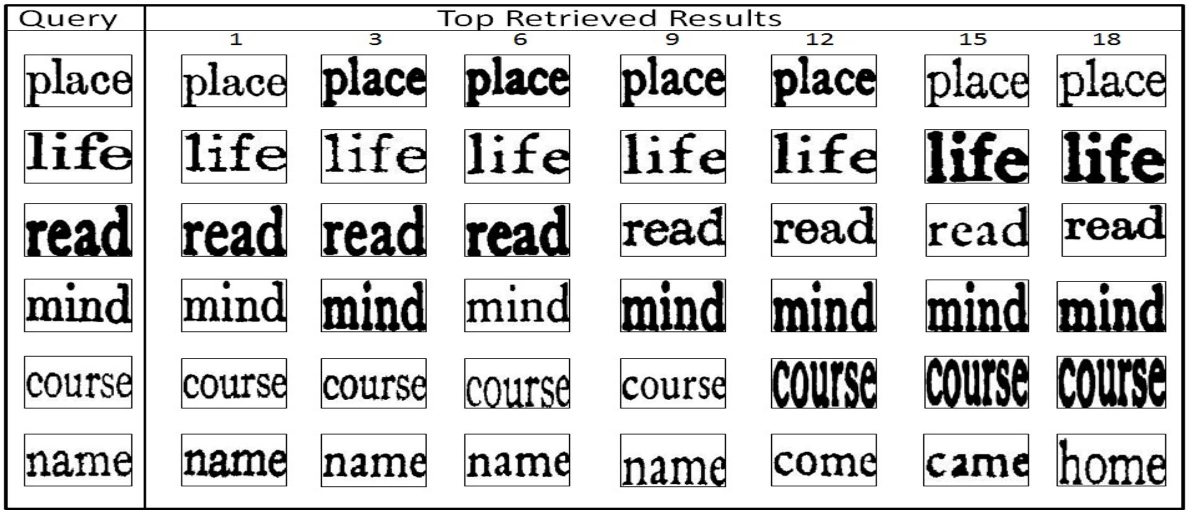

Query
Top Retrieved Results
1
3
6
9
12
15
18
place place place place place place place place
life life life life life life life life
read read read read read read read read
mind mind mind mind mind mind mind mind
course course course course course course course course
name name name name name come came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

Figure 4.6 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.

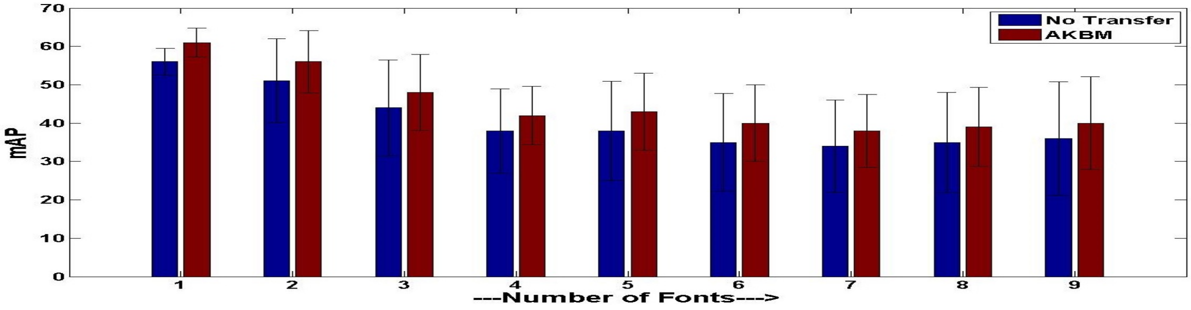


Figure 4.7 Comparison of AKBM with no transfer case is given for the multifont dataset Dlab. Performance is given as MAP values with increasing number of fonts used in the target dataset. The rectangular bars represent the average MAP values for 10 runs and the error bars show the corresponding standard deviations.

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.

In Table 4.4 we compare the semi-supervised style transfer strategy (SSST) with supervised style transfer. For doing supervised style transfer using Equation 4.3, we use a single labeled example per word class from the target domain instead of doing nearest neighbor based label propagation. SSST per-

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 4.4 Comparison between semisupervised style transfer (SSST) and supervised transfer.

forms 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.

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.

4.6 Summary

In this Chapter, 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 from all the documents using the same set of high dimensional basis vectors. We have shown results on various datasets varying in font.

Chapter 5

Classifier Design as Domain Adaptation

The mismatch between the distribution of the training data and the test data is a challenging issue while designing many practical machine learning systems. In this chapter, we propose an unsupervised domain adaptation technique to tackle this issue. We are interested in a domain adaptation scenario where source domain has large amount of labeled examples and the target domain has large amount of unlabeled examples. We align the source domain subspace with the target domain subspace in order to reduce the mismatch between the two distributions. We model the subspace using Locality Preserving Projections (LPP). Unlike previous subspace alignment approaches, we introduce a strategy to effectively utilize the training labels in order to learn discriminative subspaces. We validate our domain adaptation approach by testing it on a dataset with two domains, i.e. handwritten and printed digit images. We compare our approach with other existing approaches and show the superiority of our method.

5.1 Introduction

Dataset shift is a scenario when the training set and the test set do not follow the same underlying distribution [79]. It is a serious concern while designing machine learning algorithms for real world applications. For example, an OCR system trained on a few fonts might perform badly on a novel test font if the distribution of the characters in the test font is very different from that of the training fonts. The problem is far more challenging when one is interested in adapting a classifier trained on a printed data set to a handwritten character data set like MNIST [54]. In this chapter, we address this specific domain adaptation problem.

We first conduct a toy experiment to motivate how the performance of the simple k-nearest neighbor based classifier degrades in the presence of dataset shift. For this experiment, we consider the task of

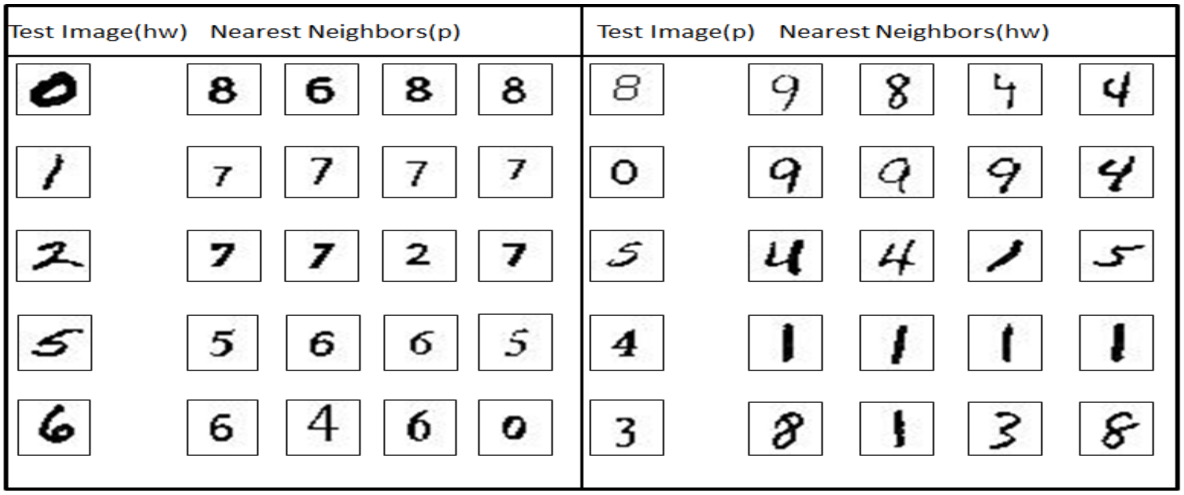


Figure 5.1 Test images from target domain and the corresponding nearest neighbors from the source domain are shown. Acronym hw stands for handwritten domain and p stands for printed domain.

classifying digit images in presence of dataset shift. The two domains we consider for the experiment are handwritten and printed digits. In order to classify test images from the target domain, we sort the labeled source domain images based on their Euclidean distance to the test image. The test image is then assigned the majority label of the k nearest sorted images. Few of the test images and their corresponding retrieved images has been shown in Figure 5.1. We observe that for the majority of cases, the source domain samples would misclassify the target domain test image. Hence, in presence of dataset shift, a source domain classifier might perform badly on the target domain.

Machine learning algorithms which are designed with mechanisms for tackling the dataset shift are called Domain Adaptation (DA) algorithms. In most of the DA algorithms, following scenario is assumed:

1. The training dataset (source domain) has plenty of labeled examples.

2. The learned model has to be tested on a test dataset (target domain) which may have a different distribution. Sufficient amount of unlabeled target domain data is available, apart from this few labeled examples from the target domain may be available while learning the model.

DA algorithms can be broadly categorized into two types: classifier adaptation based or feature adaptation based. Classifier based DA techniques such as [101] and [27] use plenty of labeled data from source domain and some labeled data from the target domain to learn a classifier which performs well on the target domain. Feature based DA approaches such as [52] try to reduce intraclass variations across the source and target domains. The feature based approaches can be further divided into two categories, semi-supervised or unsupervised, depending on whether few labeled examples from target domain are available or not. A review of different DA approaches for statistical classifiers can be found in [46].

Subspace based DA techniques such as [36] are becoming a popular means of doing unsupervised DA. Recently a subspace alignment based unsupervised DA approach has been presented in [32]. The central idea of their work is to align the source and target subspaces and then project all the data points to their respective aligned subspace before the classification. They model the source subspace by the eigenvectors obtained by doing PCA over the source domain and similarly for the target subspace. They align the source subspace with the target subspace by learning a transformation matrix. In their approach, however, label information present in the source domain is not being utilized while learning the subspaces. Also, PCA aims at maximizing the variance of the projected data but does nothing to preserve the local neighborhood inherent in the original space. Keeping in mind these two facts, using PCA for modeling subspaces might lead to less discriminative subspaces.

In this chapter, we present an unsupervised subspace alignment based DA approach, similar to [32]. We use Locality Preserving Projections (LPP) described by [42] for modeling the subspaces. LPP builds an adjacency graph using neighborhood information from the data set. Once the adjacency graph is formed, LPP finds those projection directions which keep the connected points in the graph as close as possible. This technique preserves the local neighborhood information present in the original space. Note that while forming the adjacency graph, LPP uses closeness of points based on Euclidean distance. Hence, it does not utilize any label information while finding the projection directions. To effectively use the labels to obtain a discriminative subspace, we use a supervised version of the LPP for learning the source subspace. As labeled examples are not available in the target domain, we use the original version of the LPP for learning the target subspace. Once the source and the target subspaces are obtained, we align the two subspaces by learning a transformation. The data points are then projected to their respective subspace before doing classification.

5.1.1 Contributions

Following are the contributions of our subspace alignment based DA approach:

1. We use label information from the source domain while learning the subspaces. This results in basis vectors which are discriminative in nature and hence more suitable for the classification task.

2. We use LPP for modeling the subspaces. This preserves the local neighborhood of the data points from the original space to the projected subspace.

3. The subspaces can be learned directly by solving a generalized eigenvalue problem.

4. We introduce a dataset comprising of two domains for validating our DA approach. The dataset has sufficient number of examples for each category.

For validating our DA algorithm, we pick the task of classifying digit images in the presence of dataset shift. The two different domains we use in our experiments are printed digits and handwritten digits. Handwritten digits are obtained by randomly sampling a subset of images from the MNIST database [54]. For the printed domain, we create a dataset of printed digits consisting of 300 fonts. Now, our DA problem can be stated as: Given labeled digit images from one domain and unlabeled digit images from another domain, classify the unlabeled images.

5.2 Related Work

There has been a lot of interest in DA techniques for the visual recognition task. [36] propose a geodesic flow kernel based approach where they integrate over infinite number of intermediate subspaces along the geodesic from source subspace to target subspace. [78] learn aligned dictionaries from multiple domains. This is a supervised approach as they use correspondence information across domains. [45] present a low rank reconstruction based DA strategy where source data points are transformed to an intermediate domain where they can be represented as a linear combination of the target domain data points. The intermediate representation is then used to transform the source domain data points to the target domain data points. In [74, 75], the difference between the source and target distributions is reduced by learning a latent feature representation. [101] learn a SVM classifier on source domain and adapt it for the target domain using some labeled data from the target domain.

5.3 Domain adaptation using Alignment of Locality Preserving Subspaces

To obtain source and target domain subspaces, we use Locality Preserving Projection (LPP) [42]. In its original form, LPP does not use any label information. Hence to utilize the label information present in the source domain to obtain a discriminative subspace, we use a supervised version of the LPP. We describe LPP and a supervised version of LPP in Section 5.3.1. This supervised version of LPP has been proposed in [18]. We refer to the supervised LPP as sLPP. Once the source domain and target domain subspaces are obtained using sLPP and LPP respectively, we align these two subspaces by learning a transformation matrix. The alignment technique has been discussed in Section 5.3.2. In Section 5.3.3, we describe our DA approach.

5.3.1 Locality Preserving Subspaces

Given a dataset with m vectors $x_1, x_2, ..., x_m$ in R^n and their corresponding labels $y_1, y_2, ..., y_m$, LPP finds a set of basis vectors A (each column of A is a basis vector) so that the neighborhood of each of the m points is preserved after the transformation $z_i = A^T x_i$. Note that LPP does not use the labels while finding the basis vectors. To obtain the transformation matrix A, first an adjacency graph $G = (V, E)$ with m nodes is formed. Nodes i and j of G are connected by an edge if the vectors x_i and x_j are close to one another. Here, x_i and x_j are considered to be close based on either of these two conditions :

- $||x_i - x_j||^2 < \epsilon$ where $\epsilon \in R$.
- x_i and x_j are among the k nearest neighbors of one another.

The edge strength W_{ij} between connected nodes i and j can be defined to be either $e^{-\frac{||x_i - x_j||^2}{t}}$ or simply 1, here $t \in R$. W_{ij} is assigned a value of 0 if the nodes i and j are not connected. The columns a of the matrix A can be found by solving the following generalized eigenvalue problem:

$$XLX^T a = \lambda XDX^T a \tag{5.1}$$

where i^{th} column of X is x_i, D is a diagonal matrix such that $D_{ii} = \sum_j W_{ij}$ and $L = D - W$ is the Laplacian matrix. Solutions of this equation are the eigen vectors that form the columns of the transformation matrix A.

Supervised LPP: Clearly LPP in its original form does not use any label information. Hence, if two vectors x_i and x_j belonging to different classes are close in original space R^n, their closeness would be preserved after the transformation also. Such scenarios would clearly have a negative impact on classification in the transformed space. To tackle this issue, we also consider the label of points x_i and x_j while forming the adjacency graph G. Hence the label aware closeness conditions can be given as:

- $||x_i - x_j||^2 < \epsilon$ and $y_i = y_j$; where $\epsilon \in R$.
- x_i and x_j are among the k nearest neighbors of one another and $y_i = y_j$.

here y_i and y_j are labels of x_i and x_j respectively. The remaining steps of sLPP are same as that of LPP. Clearly, sLPP would only preserve the intra class neighborhoods.

5.3.2 Aligning subspaces

Assume that both the source domain and the target domain data points lie in R^n. The m_s data points from source domain are arranged as column vectors of the $n \times m_s$ matrix X_s and similarly the m_t data points from the target domain are arranged in the $n \times m_t$ matrix X_t. The m_s dimensional column vector Y_s contains the labels of each of the source domain examples. Also, assume that the subspaces corresponding to the source domain and the target domain are known and each of the subspaces are represented using k basis vectors. Let the source subspace be represented by the $n \times k$ matrix A_s whose columns are the source domain basis vectors obtained by solving the generalized eigenvalue problem given in Equation 5.1. Similarly, the target subspace can be represented by the $n \times k$ matrix A_t whose columns are the target domain basis vectors. We want to find a transformation which aligns A_s with A_t. We model the transformation using a $n \times n$ matrix M. To obtain M, we minimize the following objective:

$$||MA_s - A_t||_F^2 \; + \; \beta \, ||M||_F^2 \tag{5.2}$$

where the first term tries to align the two subspaces, the second term is a regularizer and β is a constant. Solution to this equation can be obtained in closed form as:

$$M = A_t A_s^T (A_s A_s^T + \beta I)^{-1} \tag{5.3}$$

here I is an identity matrix. MA_s is the transformed source domain subspace which is aligned with the target domain subspace.

```
**Data**: source vectors $X_s$, source labels $Y_s$, target vectors $X_t$, constant $\beta$
**Result**: transformed vectors $Z_s$, $Z_t$
$A_s \leftarrow sLPP(X_s)$;
$A_t \leftarrow LPP(X_t)$;
$M \leftarrow A_t A_s^T (A_s A_s^T + \beta I)^{-1}$;
$Z_s^T \leftarrow {A_s}^T M^T X_s$;
$Z_t^T \leftarrow {A_t}^T X_t$;
```

Algorithm 1: DA by aligning Locality Preserving Subspaces

5.3.3 DA by aligning subspaces

In [32], an unsupervised domain adaptation technique is presented where the source and the target subspaces are aligned and the samples are then projected to their respective subspaces. In unsupervised scenario for domain adaptation, we have plenty of labeled data available in the source domain whereas only unlabeled data is available in the target domain. However, such an approach does not utilize any label information present in the source domain. Our unsupervised domain adaptation method, described below, uses labeled data from the source domain as well as unlabeled data from the target domain for learning the source and target subspaces respectively. The authors used eigenvectors induced by doing a PCA as the basis vectors of the subspaces. Although the eigenvectors obtained by PCA maximizes the overall variance of the data, they do not preserve the local neighborhood of the data points. Hence we use LPP for obtaining the source and target subspaces.

DA by aligning LPP subspaces: The goal of our DA approach is to use such subspaces where neighborhood of data points in the original space is preserved in the transformed space and also to utilize the label information present in the source domain while learning the source subspace. We describe our algorithm for doing these in Algorithm 1. Let X_s be the $n \times m_s$ matrix containing the source domain examples, where n is the dimension of each example and there are m_s such examples. Let Y_s be a m_s dimensional column vector containing the labels of the source examples. Also, X_t contains the target domain examples. The Algorithm 1 takes as input the source domain points X_s, target domain data points X_t and the labels of source domain data points Y_s and outputs the data vectors in the respective aligned subspaces, i.e. Z_s and Z_t. In order to utilize the source labels, the algorithm uses sLPP to learn the source subspace A_s. Target subspace A_t is learned by LPP. Once A_s and A_t are obtained, the two subspaces are aligned using the technique mentioned in Section 5.3.2. The source and target domain data points are now projected over the respective aligned subspaces represented by MA_s and A_t respectively as $Z_s^T = (MA_s)^T X_s$ and $Z_t^T = A_t^T X_t$.

5.3.4 Discussion

Most of the subspace based domain adaptation techniques, for example [36], [32] and [71] are unsupervised in nature. A majority of these techniques ([36, 71]) share a common theme wherein they try to obtain the representation of the data points across the intermediate subspaces between the source and the target subspace. This helps in obtaining a domain invariant representation of the data points. The work of [32] is different from these approaches as they do not obtain the intermediate representations of data points, but rather align the source and target domain subspaces and subsequently each data point is projected to a single subspace. All these approaches do not utilize the label information present in the source domain. Hence the subspaces over which they project the data points may not be discriminative enough for the classification task. Our approach, however, utilizes the source domain labels and finds such a source subspace which preserves the intra-class neighborhoods. Hence the source subspace in our approach is discriminative in nature. We find the target subspace and align it with the source subspace. Our approach preserves the geometry of data points from both the source and the target domains and also utilizes the label information from the target obtain to obtain discriminative subspaces which are suitable for classification.

5.4 Dataset and Experiments

In this section, we give details about the datasets used for the experiments and the features used for representing the images. We also validate our domain adaptation technique by doing nearest neighbor based classification experiments and compare our approach with related approaches.

5.4.1 Dataset and Representation

For our experiments, we use digit images $(0 - 9)$ from two domains, i.e. printed and handwritten. Handwritten digits are obtained by randomly sampling 300 images of each of the digits from the MNIST database. These images are equally subdivided into three sets, i.e. Train, Test and Validation set. All the images are binarized using the thresholding technique given in [73]. For printed digits, we obtained 300 different fonts from the internet and generated binary images of the digits in each of the fonts. To keep the image size same across both the domains, all synthetic images were generated in the same size as the images in the MNIST database, i.e. 28×28. Again, the synthetic dataset was equally subdivided into Train, Test and Validation sets.

	Handwritten	Printed
Handwritten	89.0	48.8
Printed	70.0	87.1

Table 5.1 Classification accuracies for cross domain experiments.

For image representation, we use histogram of oriented gradient features (HOG) described by [23]. The reason for using gradient features instead of raw pixel representation is that these features have been shown to give better results in handwritten digits classification task [58]. HOG features are obtained by dividing an image into square cells and computing a histogram of edge orientations for the pixels within each cell. We used the cell size 8 for our experiment. The HOG feature of each cell was concatenated to obtain a column vector representation for each image. All the feature vectors were normalized to have zero mean and unit variance.

5.4.2 Experiments

As described in Section 5.4.1, data from both the domains has been divided equally into Train, Test and Validation sets. For all the subsequent experiments, Train set examples are used as labeled source domain examples and Test set examples are used as unlabeled target domain examples. The optimal values for subspace dimension and β are learned by doing classification on the validation set. We conduct a cross domain classification experiment to observe how the classification accuracy of a nearest neighbor based classifier decreases in presence of dataset shift. In this experiment, we classify target domain samples using labeled examples from the source domain. We present the results in Table 5.1. We observe that accuracy is high when training and test set belong to same domain. Also, the classification accuracy decreases when training and test sets are from different domains. In Table 5.2, we compare our subspace based DA approach with existing techniques for the cross domain classification task. For the no adaptation case, we directly classify the target domain test points using labeled examples from the source domain.

We also show results for the trivial projection cases when samples from both of the domains are projected to a common subspace. We consider three common subspaces, source PCA subspace obtained by doing PCA for the source domain data points, target PCA subspace obtained by doing PCA of target data points and combined PCA subspace obtained by doing PCA of samples from both the domains.

Source	Target	Method	Accuracy
HW	Printed	NA	48.8
HW	Printed	PCA (source)	55.9
HW	Printed	PCA (target)	56.5
HW	Printed	PCA (combined)	56.5
HW	Printed	[32]	57.0
HW	Printed	Ours	**64.8**

Source	Target	Method	Accuracy
Printed	HW	NA	70.0
Printed	HW	PCA (source)	68.1
Printed	HW	PCA (target)	68.9
Printed	HW	PCA (combined)	70.2
Printed	HW	[32]	70.6
Printed	HW	Ours	**73.2**

Table 5.2 Classification accuracies for DA experiments. Here, HW is acronym for Handwritten domain, NA refers to No Adaptation scenario, PCA (source) refers to projecting all the samples to the source domain PCA subspace, similarly PCA (target) refers to projecting all the samples to target domain PCA subspace. PCA combined refers to projecting to the combined source and target PCA subspace. The left table shows results results for the case when the source and target domains are handwritten and printed respectively. For the table on the right, the source and target domains are printed and handwritten.

We also compare our approach with the recent subspace alignment approach of [32]. We observe that our approach outperforms all the other approaches. We also observe that improvement because of our DA approach is significantly better in the case when the source and target domains are handwritten and printed respectively.

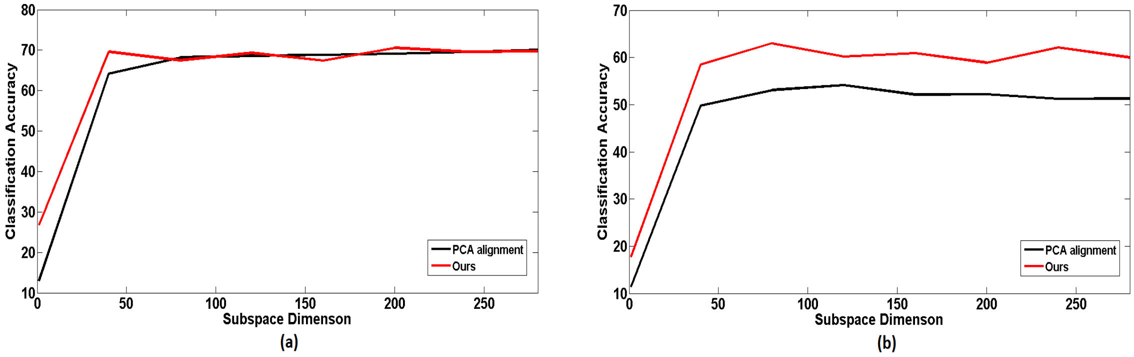


Figure 5.2 The plots for classifcation accuracy as a function of subspace dimension has been shown. Here, we compare our subspace alignment approach with the PCA subspace alignment approach proposed by [32]. Plot (a) corresponds to the case when source domain is Printed, target domain is handwritten. For Plot (b), source domain is handwritten and target domain is printed.

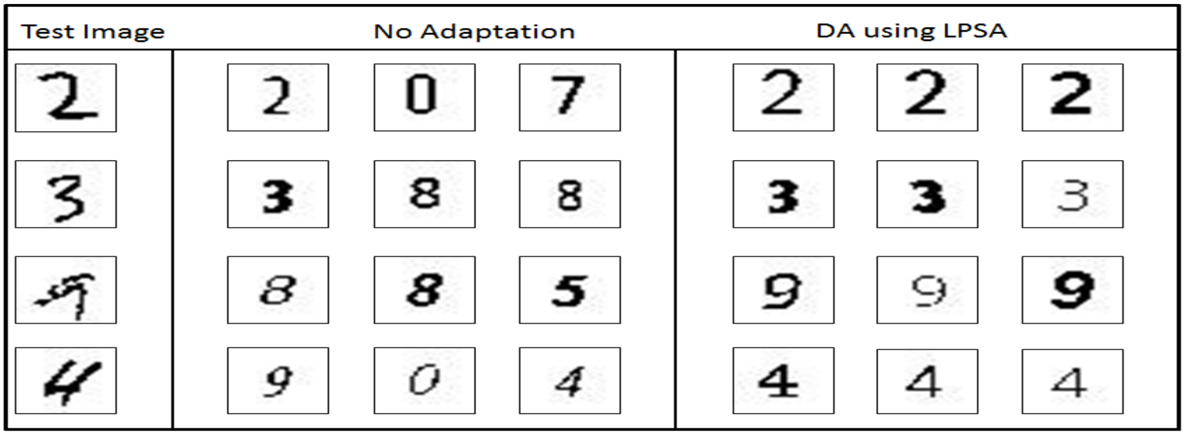


Figure 5.3 Qualitative comparison of nearest neighbor classifier for No Adaptation case with our DA approach. Here LPSA is acronym for locality preserving subspace alignment. The source and target domains are printed and handwritten respectively. The figure shows test image from target domain and corresponding nearest neighbors from source domain.

In Figure 5.2, we compare the performance of our subspace alignment approach with the approach of [32] as the subspace dimension is increased. We observe that for lower dimensional subspaces, both the approaches perform badly. For higher dimensional subspaces (around 50 and above), performance of both the approaches improve significantly. We observe that for the case when the source domain is handwritten and the target domain is printed, our method consistently outperforms [32] by a good margin. In the other case, when the source and target domains are printed and handwritten respectively, although our method outperforms [32], the difference between the two is not as prominent as the previous case. In Figure 5.3, we compare the qualitative results for no adaptation case with our subspace alignment approach for the cross domain nearest neighbor based classification task. In this figure, the source and target domains are printed and handwritten respectively. For the experiment, a test image is picked from the target domain and samples from source domain are sorted based on their distance to the test image. We can clearly observe the improvement in the results because of our approach. Although the source and target domain samples look visually very different from one another, our subspace alignment algorithm transforms the samples so that the intra-class variations across the domains is minimized. In Figure 5.4, we repeat the previous experiment taking the source and target domains as handwritten and printed respectively. Here also, our approach performs much better than the no adaptation case.

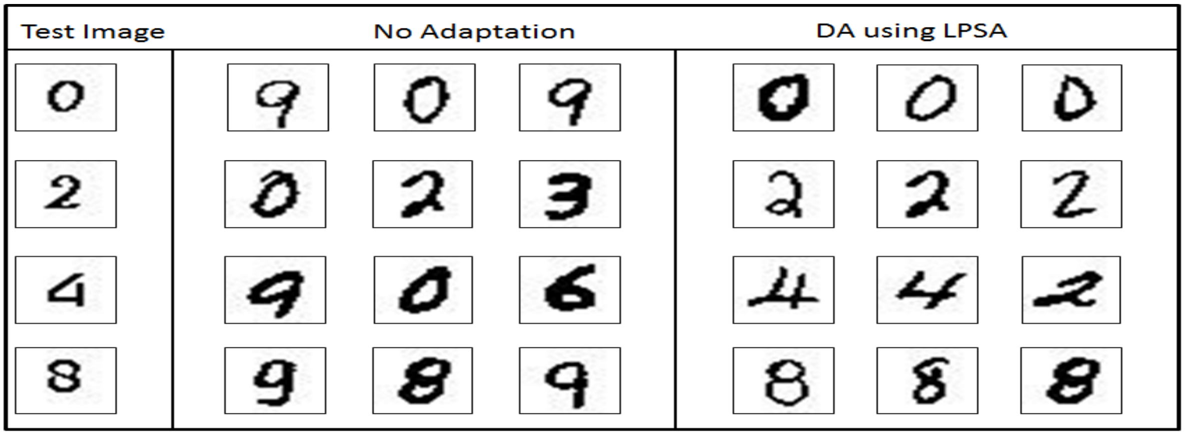

Test Image
No Adaptation
DA using LPSA

Figure 5.4 Qualitative comparison of nearest neighbor classifier for No Adaptation case with our DA approach. Here LPSA is acronym for locality preserving subspace alignment. The source and target domains are handwritten and printed respectively. The figure shows test image from target domain and corresponding nearest neighbors from source domain.

5.5 Summary

We presented an unsupervised DA strategy for classification in the presence of dataset shift. We have shown the application of our strategy for the task of digits classification, however, the approach is general and can be used for other tasks. Our approach not only learns subspaces by utilizing the source domain labels but also preserves the local neighborhood of data points. Hence the subspaces are discriminative in nature. We show the superiority of our approach over other existing approaches by showing significant improvement in classification over the other methods.

Chapter 6

Domain adaptation in object recognition

Real world applicability of many computer vision solutions is constrained by the mismatch between the training and test domains. This mismatch might arise because of factors such as change in pose, lighting conditions and quality of imaging devices. In this chapter, we present a dictionary learning based approach to tackle the problem of domain mismatch. In our approach, we jointly learn dictionaries for the source and the target domains. The dictionaries are partially shared, i.e. some elements are common across both the dictionaries. These shared elements can represent the information which is common across both the domains. The dictionaries also have some elements to represent the domain specific information. Using these dictionaries, we separate the domain specific information and the information which is common across the domains. We use the latter for training cross-domain classifiers. That is, we build classifiers that work well on a new target domain while using labeled examples only in the source domain. We conduct cross-domain object recognition experiments on popular benchmark datasets and show improvement in results over the existing state of art domain adaptation approaches.

6.1 Introduction

Visual object recognition schemes popularly use feature descriptor such as SIFT [57], HOG [23] followed by a classification strategy such as SVMs [77]. They train on a set of annotated training set images and evaluate on a set of similar images for quantifying the performance. However, such object recognition schemes may perform badly in the case of large variations between the source domain and the target domain [96]. Variations between the source and target domain might arise from changes in pose, illumination or intra-class variations inherent in object categories. In Figure 6.1, we show sample images of the categories chair and bottle from three different domains, namely Amazon, DSLR and

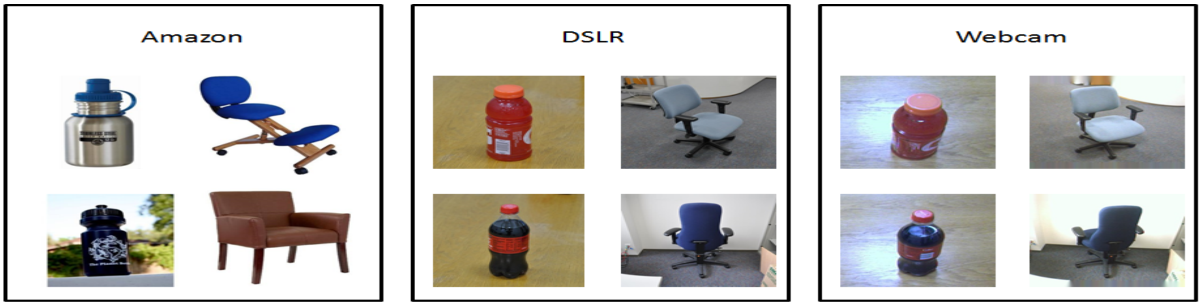


Figure 6.1 Sample images of categories “bottle” and “chair” from the domains Amazon, DSLR and Webcam [86]. Images from Amazon are visually very different in comparison to the other two domains. Visual mismatch between DSLR and Webcam is relatively less and arises from factors such as changes in pose, image resolution and lighting conditions.

Webcam. The domain Amazon is visually very different from the other two domain, the reason being large intra-class variations. The difference between the domains DSLR and Webcam arises because of change in pose, camera quality and lighting condition.

To tackle the issue of variations across the source and target domains, various domain adaptation (DA) techniques have been proposed in the natural language processing as well as computer vision communities. In Figure 6.2, we present the overall idea behind a general DA approach. The figure depicts the idea that a classifier trained on the source domain may need further adaptation in order to perform well on the target domain. In the natural language processing community, DA techniques have been applied for tasks such as sentiment classification, parts of speech tagging etc. Blitzer *et al.* [16] present a DA technique to modify discriminative classifiers from the source domain to the target domain. The primary aspect of their work is identifying the *pivot* features, i.e. those features which occur frequently and behave similarly across the two domains. Hal Daume [24] presents a feature augmentation approach where source, target and a common domain representation are obtained by replicating the original feature.

In recent years, there has been a surge of interest in the visual domain adaptation task. Several DA strategies have been proposed which adapt either the feature representation or the classifier. These strategies are semi-supervised or unsupervised depending on whether some labeled data from the target domain is available or not. Utilizing labeled examples from source as well as target domains, Saenko *et al.* [86] learn a transformation to map vectors from one domain to another. This transformation tries to bring closer the intra-class vectors from the two domains and push the inter-class vectors farther apart. [37] present an unsupervised DA approach where source and target subspaces are points on a Grassmann manifold. They sample points along the geodesic between the source subspace and the target subspace

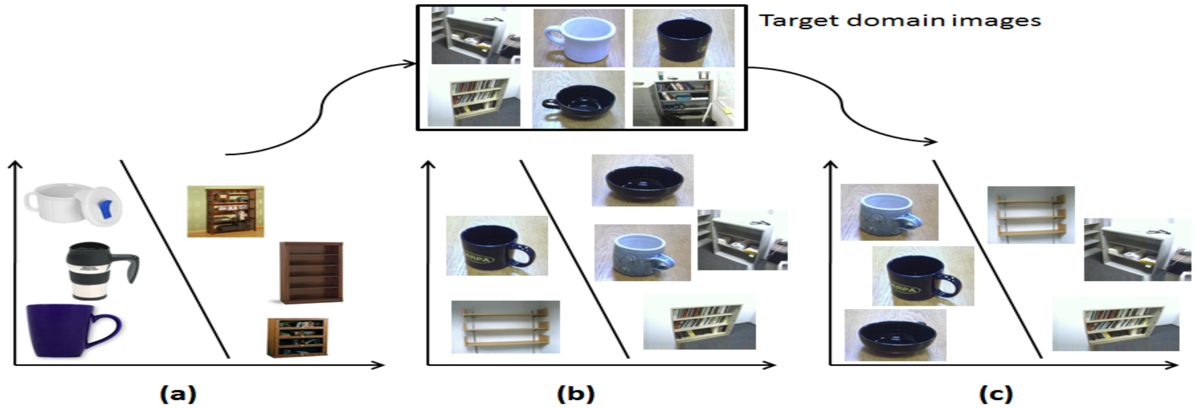


Figure 6.2 Overall idea behind a Domain Adaptation approach is shown. Source and target domains are Amazon and Webcam respectively. The two object categories are mug and bookcase. Fig(a) shows a classifier which perfectly separates the two object categories in the source domain. Fig(b) shows the same classifier misclassifies images from the target domain. Fig(c) shows the scenario after domain adaptation, the classifier now correctly classifies the target domain images. The target domain images aid the DA strategy. These examples can be labeled or unlabeled depending on whether the DA approach is semi-supervised or unsupervised.

to obtain intermediate subspaces. The data points are projected along all the intermediate subspaces to obtain a domain independent representation. Jhuo *et al.* [45] present a semi-supervised DA approach based on low-rank approximation. The samples from the source domain are mapped to an intermediate representation where the transformed source samples can be expressed as a linear combination of target samples. The authors consider single source domain as well as multiple source domain scenarios in this work.

Recently, sparse representation has been used for various visual DA tasks such as object recognition, face recognition [93, 71] and action recognition [104]. Zheng *et al.* [104] propose a dictionary learning approach for doing cross-domain action recognition. Given correspondence between videos from two domains, i.e. videos of same action shot from two different views, they learn two separate dictionaries while forcing the sparse representation for corresponding video frames from the two domains to be same. Using this view independent representation, action model learned from the source view video can be directly applied on the target view video. Ni *et al.* [71] present a dictionary learning based DA approach when correspondence information across domains is not available. Given a dictionary in one domain, say source, they iteratively modify the dictionary to be suitable for the target domain. They store all the intermediate dictionaries and use all of them to obtain a view independent representation of images from both the domains. Shekhar *et al.* [93] present a dictionary learning based approach where they map samples from both the domains to a low dimensional subspace and learn a common dictionary by minimizing reconstruction error for the projected samples in the low dimensional subspace. They also add regularization terms to their objective which try to minimize loss of information while projecting samples from both domains to the common subspace. They learn separate discriminative dictionaries for each category by encouraging low intra-class and high inter-class reconstruction error in the objective. Qiu *et al.* [78] present a dictionary transformation based DA approach to transfer signal from source domain to target domain. The transformation is done by adapting the dictionaries while keeping the sparse representation of the signal fixed. The domain dictionaries are modeled by a linear or nonlinear parametric function. The dictionary function parameters as well as the sparse codes are obtained jointly by solving an optimization problem.

In our current work, we present a dictionary learning approach for learning partially shared dictionaries across different domains. We learn separate dictionaries for the source and target domains. These dictionaries have some shared atoms which represent the common information which is present in both the domains. The dictionaries also have some domain specific atoms to represent the domain specific information. We show the effectiveness of our dictionary learning strategy by using it for the cross-domain classification task. The domain specific information can cause confusion while doing cross-domain classification. Hence, we ignore the domain specific information and use the representation obtained from the common dictionary elements for training cross-domain classifier.

6.1.1 Contributions

The contributions of our current work are

1. We present a strategy for jointly learning partially shared dictionaries across domains.

2. We design the dictionaries to have two types of elements, i.e. domain specific elements and domain independent elements. As the name suggests, the domain specific atoms represent the domain specific information whereas the domain independent elements capture the information common to both the domains.

3. A form of selective block sparsity arises naturally from the partially shared dictionary learning formulation. More specifically, depending on the underlying domain of the signal, a specific block of sparse coefficients is forced to consist only of zeros. A simple strategy for obtaining sparse representation in presence of selective block sparsity is given.

4. Our dictionary learning approach can be seen as making few modifications over an existing dictionary learning approach [30]. However, using this simple approach we obtain comparable results to the state of the art visual DA approaches.

6.2 Domain Adaptation using Partially Shared Dictionaries

6.2.1 Method Overview

A dictionary learned from the source domain might not be suitable for representing signals from the target domain. Using a dictionary for cross-domain representation might result in a scenario where the sparse representation obtained for the same class signals from the two domains are very different. Clearly, such a representation will lead to poor cross-domain classification performance. Hence, while designing dictionaries in the presence of domain mismatch, further steps are required to accommodate signals from the new domains. We have presented a partially shared dictionary learning strategy to tackle the issue of domain mismatch. Our strategy is based on the idea that there could be some commonalities between the source and target domains. The same set of dictionary atoms can be used to represent this common information. Also, the signals from a domain will have certain domain specific aspects. This domain specific information can be represented well by dictionary atoms which are exclusive to the particular domain. Using the common atoms for sparse decomposition will lead to similar representation for the same class signals from both the domains. Hence, such a representation is more suited for the cross-domain classification task. The overview of our approach is shown in Figure 6.3. As shown in the figure, some atoms are shared across the dictionaries from the source and the target domains. Apart from these common atoms, the dictionaries also have some domain specific atoms.

6.2.2 Sparse Representation of Signals

A signal $y \in R^n$ can be sparsely represented using a dictionary $D \in R^{n \times K}$ consisting of K atoms or prototype signals. The atoms of D can be pre-defined using discrete cosine transform basis [4], wavelets [60] or they can be learned from the available signals. The learned dictionaries have been shown to perform better than pre-defined dictionaries for tasks such as reconstruction [28]. For learning dictionaries from the data, several efficient dictionary learning strategies such as K-SVD [3] and

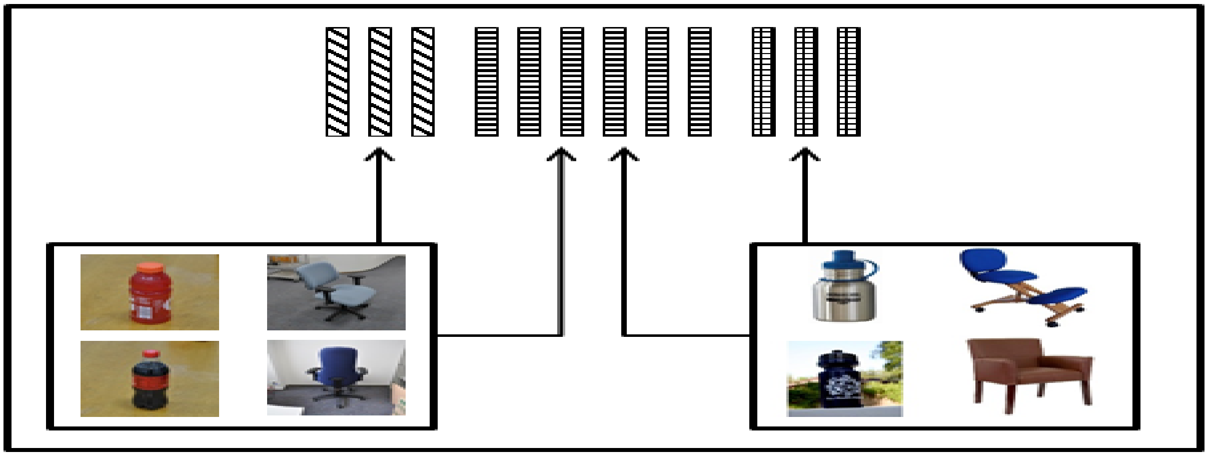


Figure 6.3 Overview of partially shared dictionary learning. Dictionary for each domain consists of two types of atoms, domain specific atoms and atoms shared across the domains. Shared atoms are learned using samples from both the domains whereas domain specific atoms are learned using samples from the corresponding domain.

MOD [30] have been proposed in the past. These dictionary learning techniques solve the following optimization problem

$$\min_{D,A} \|Y - DA\|_F^2 \quad \text{subject to} \quad \forall i, \quad \|a^i\|_0 \leq T_0. \tag{6.1}$$

Here the signals are arranged along the columns of Y and the columns of A, i.e. a^i, contain the corresponding sparse representation. The dictionary learning techniques solve this problem by alternating between solving for A, i.e. sparse coding step and updating D, i.e. dictionary update step. In [30], the dictionary update consists of updating all the dictionary elements while keeping the sparse representation unchanged. The dictionary learning approach given in [3], however, updates a single dictionary atom at a time. The sparse coefficients also change during the update so that the number of nonzero coefficients further reduces or remains the same. The sparse decomposition problem with the l_0 penalty is NP hard and greedy algorithms are used to solve this. When D is fixed, sparse representation a^i can be obtained using greedy pursuit algorithms such as OMP [97]. Sparse decomposition can also be done by relaxing the l_0 penalty and using a l_1 penalty in its place [26].

6.2.3 Partially Shared Dictionary Learning

Dictionary learned from one visual domain might not be suitable for representing signals from another visual domain. Hence, we propose a dictionary learning strategy which jointly learns a dictionary which is suitable for both the source as well as target visual domains. We believe that any domain can be represented effectively using a dictionary which has some domain specific atoms as well as some domain independent atoms, i.e. which are common across domains. This assumption is based on the fact that instances of same category across different domains generally have some similarity between them. Hence, the source domain dictionary D_s and the target domain dictionary D_t can be represented as

$$D_s = [D_{src} \;\; D_c] \; ; \quad D_t = [D_{tgt} \;\; D_c], \tag{6.2}$$

where D_{src}, D_{tgt} are source and target domain specific atoms and D_c are the common atoms across the two domains. Also, we represent the combined dictionary D as

$$D = [D_{src} \;\; D_c \;\; D_{tgt}]. \tag{6.3}$$

The objective for jointly learning D is given as given as

$$\begin{aligned} \min_{D,A,B} \quad & \|[Y_s \;\; Y_t] - D[A \;\; B]]\|_F^2, \\ \text{subject to} \quad & a^i_{tgt} = [0\,0 \,....\; 0]^T, \; b^i_{src} = [0\,0 \,....\; 0]^T, \\ & \|a^i\|_0 \le T_0, \; \|b^i\|_0 \le T_0, \end{aligned} \tag{6.4}$$

where $a^i = [a^i_{src} \;\; a^i_{com} \;\; a^i_{tgt}]^T$, $b^i = [b^i_{src} \;\; b^i_{com} \;\; b^i_{tgt}]^T$. Both the sparse coefficient vectors a^i and b^i can be seen as a concatenation of three blocks of coefficient vectors. Depending upon the underlying domain of the corresponding signal, one of these three blocks, i.e. a^i_{tgt} or b^i_{src}, is forced to have all elements as zero. The equality constraints thus give rise to a specific form of block sparsity [29], which we call selective block sparsity. The above optimization problem allows for jointly learning the source as well as the target domain dictionaries. The equality constraint $a^i_{tgt} = [0\,0 \,....\; 0]^T$ makes sure that the dictionary atoms D_{tgt} are used only for representing the target domain signals Y_t. Hence, D_{tgt} captures only the target domain information. Similarly, the equality constraint $b^i_{src} = [0\,0 \,....\; 0]^T$ makes sure that D_{src} captures only the source domain information. The block of sparse coefficients a^i_{com} and b^i_{com} correspond to the common dictionary atoms D_c. As both a^i_{com} and b^i_{com} can have non-zero terms, the dictionary atoms D_c are used while representing signals from both the domains, hence, these atoms capture the common information across the source and target domains.

To effectively solve the optimization problem given in Equation 6.4, we rewrite it as

$$\begin{aligned} \min_{D_s,D_t,A_s,B_t} \quad & \|Y_s - D_s A_s\|_F^2 \quad + \quad \|Y_t - D_t B_t\|_F^2, \\ \text{subject to} \quad & \|a^i_s\|_0 \le T_0, \; \|b^i_t\|_0 \le T_0, \end{aligned} \tag{6.5}$$

where $a_s^i = [a_{src}^i \;\; a_{com}^i]^T$, $b_t^i = [b_{tgt}^i \;\; b_{com}^i]^T$. We would like to point out here that to learn D_s and D_t using Equation 6.5, one might be tempted to use MOD and alternate between sparse coding and dictionary update by taking derivative of the energy term with respect to D_s and D_t. However, such an approach would not ensure the structure we desire to be present in the dictionaries D_s and D_t, as presented in Equation 6.2. We take a short digression to describe how to solve Equation 6.4 in case the dictionary structure given in Equation 6.2 is not present. In such a scenario, we can rewrite the optimization problem given in Equation 6.4 as Equation 6.5. For dictionary learning, we can use MOD. Obtaining a_s^i and b_t^i is straightforward and these can be obtained via OMP. If a_s^i and b_t^i are available, a^i and b^i can be obtained trivially by concatenating a vector of zeros at appropriate position.

Now we get back to our original dictionary learning formulation. To maintain the desired structure in the two dictionaries D_s and D_t, we further couple the two dictionaries D_s and D_t using the following relation between the two

$$D_t = D_s P, \tag{6.6}$$

where P is a square matrix. Since we want some elements to be common among D_s and D_t, we fix a set of the columns of P, i.e. Q, to have the diagonal elements as 1 and non-diagonal elements as 0. Hence P can be represented as

$$P = [Q \quad R]. \tag{6.7}$$

Using Equation 6.7, Equation 6.5 can be rewritten as

$$\begin{aligned} \min_{D_s, R, A_s, B_t} \quad & \|Y_s - D_s A_s\|_F^2 \quad + \quad \|Y_t - D_s Q B_{com} - D_s R B_{tgt}\|_F^2, \\ \text{subject to} \quad & \|a_s^i\|_0 \leq T_0, \; \|b_t^j\|_0 \leq T_0, \end{aligned} \tag{6.8}$$

where $B_t = \begin{bmatrix} B_{tgt} \\ B_{com} \end{bmatrix}$.

To solve this optimization problem, we alternate between updating D_s and R followed by sparse coding step. We set the first order derivative with respect to D_s equal to zero and obtain the following closed form expression for D_s

$$D_s = (Y_s A_s^T + Y_t B_t^T P^T)(A_s A_s^T + P B_t B_t^T P^T)^{-1}. \tag{6.9}$$

Similarly, the update for R is done using the following closed form expression.

$$R = (D_s^T D_s)^{-1} D_s^T E_t B_{tgt}^T (B_{tgt} B_{tgt}^T)^{-1}, \tag{6.10}$$

where $E_t = Y_t - D_s Q B_{com}$. In the sparse coding step, D_s and R is kept fixed and OMP is used to obtain the sparse representation.

6.2.4 Cross-Domain Classification using PSDL

Using unlabeled data from the source and the target dictionary, the dictionary D is learned as described in the previous section. The dictionary atoms which are common across the two domains, i.e. D_c, are then used to obtain sparse representations for signals from both the domains. The sparse decomposition using D_c maps signals from both the domains to a common subspace. The sparse representation of samples from source and target domain, thus obtained, are used directly for doing cross-domain classification. The coefficients corresponding to D_{src} and D_{tgt} are ignored while doing cross-domain classification.

The dictionary atom subsets D_{src} and D_{tgt} represent the domain specific information, hence, using their coefficients also for the cross-domain classification task will create confusion for the classifier. By using just the coefficients corresponding to D_c, we effectively extract only the common information which is shared across the source and target domains. This results in similar sparse representation for same class signals across the two domains. Clearly, such a representation is better suited for the crosss-domain classification task.

As stated before, we use just the coefficients corresponding to D_c for representing signals from the source and the target domains. The classifiers are trained using the sparse representation for plenty of labeled data from the source domain as well as a small amount of labeled data from the target domain. We use SVMs for the cross-domain classification task.

6.3 Results and Discussions

We validate our approach by conducting object recognition experiments in a cross-domain setting on benchmark datasets. We conduct the experiments using the same experimental setup as in [36, 71]. Our dictionary learning approach PSDL is unsupervised and does not use any label information from the source or the target domains. We compare our dictionary learning based DA approach with baseline dictionary learning approaches as well as a recently proposed dictionary learning based DA approach [71]. We also compare our approach with other DA techniques [37, 36].

6.3.1 Dataset and Representation

We conduct object recognition experiments on 4 datasets, i.e. Amazon (images downloaded from online merchants), Webcam (images taken by a low resolution webcam), DSLR (images taken by a digital SLR camera) and Caltech (images taken from the Caltech-256 [40] dataset). The first three datasets were introduced in [86] whereas the fourth one was first studied by [36]. Each of the dataset are considered as a separate domain. Datasets consist of images pertaining to the following 10 classes BACKPACK, TOURING-BIKE, CALCULATOR, HEAD-PHONES, COMPUTER-KEYBOARD, LAPTOP, COMPUTER-MONITOR, COMPUTER-MOUSE, COFFEE-MUG, VIDEO-PROJECTOR. There are atleast 8 images and a maximum of 151 images per category in each domain. In total the datasets consist of 2533 images.

Scale invariant interest points were detected in the images using the SURF detector [11]. A 64 dimensional SURF descriptor was used to describe the patch around the interest points. A codebook consisting of 800 visual words was constructed by clustering random descriptors from the Amazon using k-means clustering. A histogram representation was obtained for each of the images by getting the count of each of the visual words in the image. All the histograms were z-score normalized to have zero mean and unit deviation in each dimension.

6.3.2 Experiments

For experiments, two domains are picked from the datasets. We use one of them as the source domain and the other is used as the target domain. Goal of the experiments is to classify target domain data points. We conduct experiments in semi-supervised setting, i.e. we sample few labeled data-points from both the domains. When Webcam or DSLR are the source domains, we sample 8 labeled points from them. In case Amazon or Caltech are the source domains, 20 labeled examples are sampled. In semi-supervised setting, 3 labeled examples are sampled from the target domain. For dictionary learning using PSDL, we utilize unlabeled samples from both the domains. The optimal number of dictionary atoms to represent the domain specific as well as the common information is found by performing cross-validation on the few labeled samples available in the target domain. Sparse representation is obtained using Orthogonal Matching Pursuit (OMP) [97]. Following the previous works [36, 71, 37] , all the experiments are repeated 20 times and the mean classification accuracy over the 20 trials is reported in each case.

Method	C → A	C → D	A → C	A → W	W → C	W → A	D → A	D → W
MOD$_{source}$	44.9	50.5	39.2	46.6	27.3	38.5	37.6	67.2
MOD$_{target}$	49.2	53.6	39.4	50.7	34.2	44.4	44.3	72.0
SGF[37]	40.2	36.6	37.7	37.9	29.2	38.2	39.2	69.5
GFK[36]	46.1	55.5	39.6	56.9	32.8	46.2	46.2	80.2
Ni *et al.* [71]	50.0	57.1	41.5	57.8	40.6	51.5	50.3	87.8
PSDL	53.9	59.4	41.8	57.7	37.0	46.8	48.9	83.3

Table 6.1 Classification accuracies for PSDL is compared with baseline dictionary learning approaches as well as the DA approaches given in [37, 36, 71]. For baseline approaches we learn source and target domain dictionaries using [30]. The acronyms A, C, D, W represent the domains Amazon, Caltech, DSLR and Webcam respectively. In the notation C → A, C is the source domain and A is the target domain. Similar notation is followed for the other dataset pairs.

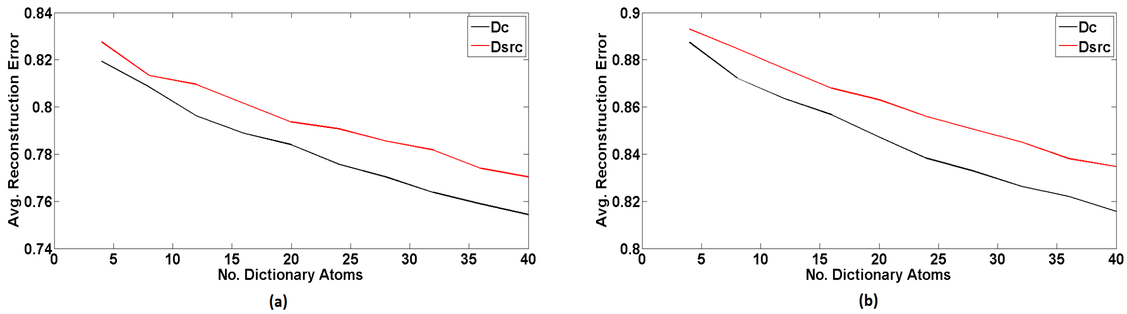


Figure 6.4 Average reconstruction error for target dataset is plotted as a function of number of dictionary atoms. Dc represents the shared dictionary and $Dsrc$ the source domain dictionary. The shared dictionary Dc does a better reconstruction of the target domain samples in comparison to $Dsrc$. For (a), the source and target domains are Amazon and Caltech respectively and for (b), Caltech and Amazon.

In Table 6.1, we compare our approach with other dictionary learning based approaches as well as other popular DA approaches. This experiment is done in a semi-supervised setting, i.e. we use few labeled examples from the target domain along with the labeled examples from the source domain. We use a SVM classifier for this experiment as in [71]. We use nonlinear SVM with RBF kernel. The optimal value for the kernel bandwidth parameter σ is learned with cross-validation. MOD based dictionary learning approach is taken as the baseline in this experiment. We use MOD for learning dictionaries from the source as well as the target domains. For MOD$_{source}$, sparse decomposition of signals from both the domains is done using the dictionary learned from just the source domain. Similarly, for MOD$_{target}$, the dictionary learned from the target domain is used for sparse representation. We also present the results given in the dictionary learning based DA approach given in [71]. We also compare our results with two other DA approaches [37, 36]. Our dictionary learning approach as well as [71] always outperform the baseline dictionary learning approaches as well as the DA approaches given in [37, 36]. For the dataset pair Webcam and DSLR, all the approaches perform better compared to the other dataset pairs. The reason for this high classification accuracy is the high similarity across these two domains, i.e. these datasets consist of images of the same object instances obtained using different imaging devices. On the other hand, all the approaches tend to show low accuracy for some dataset pairs, for example Amazon and Caltech. This can be explained by the large variations across these two domains. We observe that for the first four dataset pairs, our method performs almost as good or better than [71]. For the remaining dataset pairs, [71] outperforms our method. For the first four cases, training domain has 20 labeled examples whereas for the last 4 cases it has 8 examples. Hence, our method performs relatively better in presence of sufficient number of source domain examples.

In Figure 6.4, we show the comparison between the average reconstruction error obtained by dictionaries D_c and D_{src} while representing the target domain. Irrespective of the dictionary size, D_c results in less reconstruction error, thus showing that it is more suited for representing the target domain. In Figure 6.5, we show three test images from the target domain and corresponding top five nearest neighbors from the source domain. We provide nearest neighbors for two cases, i.e. results obtained via our dictionary learning based approach and no adaptation case using original features. In all the three examples, improvement because of our approach is clearly observable.

6.4 Summary

In this Chapter, we present a partially shared dictionary learning approach. Our approach allows dictionaries from the source and the target domains to share some atoms. We use the shared atoms to represent signals from both the domains. This results in similar sparse representation for same class signals across the two domains. Our results show that such a representation is better suited for cross-domain visual object recognition task.

We show the effectiveness of our approach by performing cross-domain object recognition on the benchmark datasets. We also compare our approach with various existing approaches and show improvement in results over the existing state of art approaches.

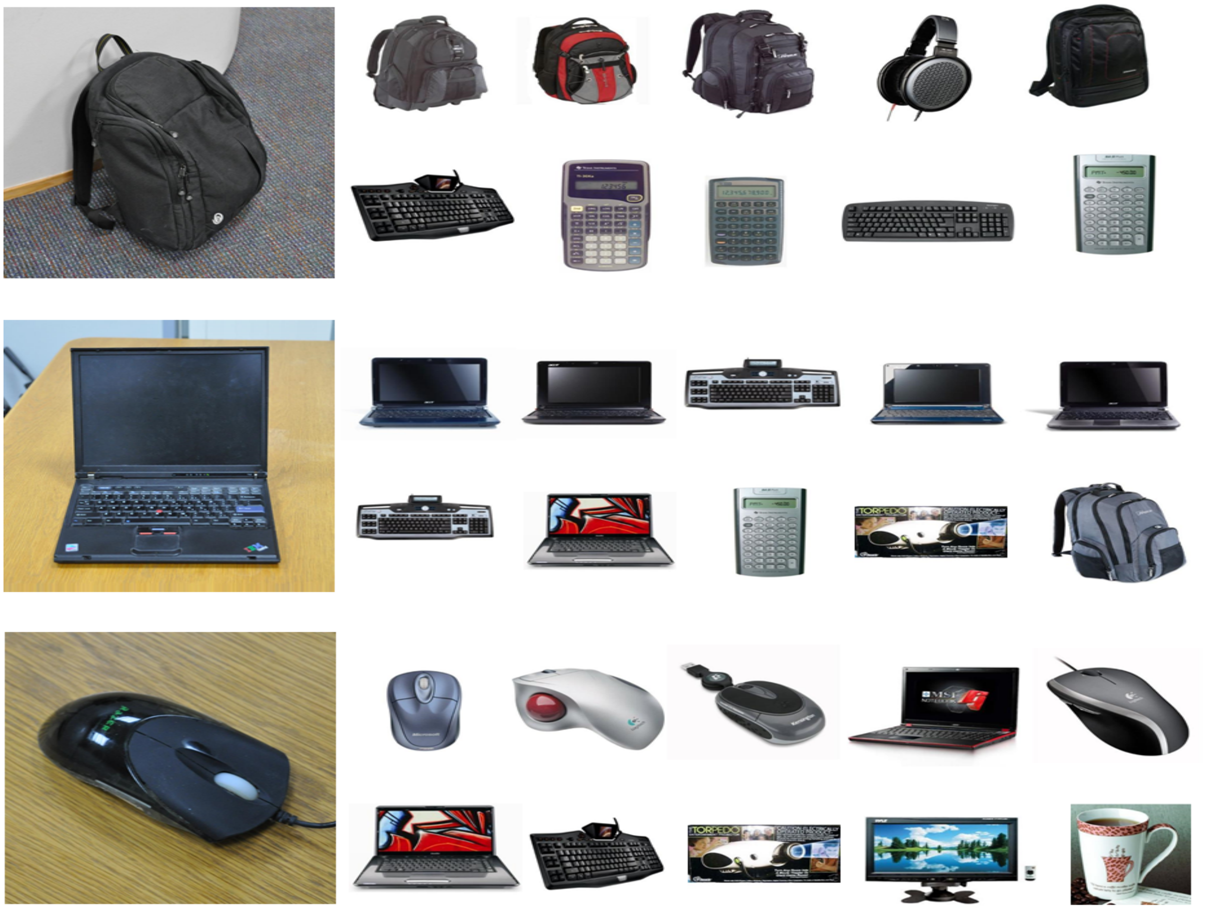


Figure 6.5 Test images from the Dslr domain on the left and two adjacent rows of corresponding nearest neighbors from the Amazon domain. Top row results are obtained using our dictionary learning approach, bottom row corresponds to nearest neighhbors obtained with original features.

Chapter 7

Conclusions and future work

In this theses, we have tackled the task of designing classifiers for novel scenarios. We have followed two broad approaches for doing this, by transforming available classifiers, and by transforming the source as well as target domain features. We have considered various Computer Vision tasks pertaining to document images as well as natural images. We have shown the superiority of our approaches by outperforming corresponding existing approaches. The following conclusions can be drawn from this theses

Synthesizing classifiers for Novel categories Classifier based retrieval systems perform better in comparison to other alternatives, however, such systems are limited by the out of vocabulary query words. In this chapter, we have presented a classifier synthesis strategy to instantly generate classifiers corresponding to novel query words. Classifier synthesis is done via a one-shot classifier learning strategy which utilizes the fact that words in a language can be formed from fewer combination of character sequences. Using the set of pre-trained classifiers for vocabulary words, and the classifier synthesis strategy for the out of vocabulary queries, we achieve state of the art performance on real world datasets pertaining to various languages. Since our classifier synthesis strategy does not need label information of the query word, our strategy might be useful in bringing down the annotation cost in large word image retrieval systems.

Synthesizing classifiers for Novel styles Style variations affect the retrieval performance in large document image databases. In this Chapter, we have proposed strategies for handling style variations in multi-font databases. To deal with the style variations between the source and the target documents, we have proposed a semi-supervised style transfer strategy based on style-content factorization. The semi-supervised style transfer strategy uses a cross-font label propagation strategy, which might not be very effective in presence of large font variations. For handling such scenarios, we have suggested a font independent retrieval strategy by representing words from all the documents using the same set of high dimensional basis vectors. The kernelized style-content factorization strategy is robust and performs well even in presence of large font variations.

Although we have used AKBM for word image retrieval task in this thesis, the model is general and can also be used as a nonlinear feature extraction strategy for other recognition and retrieval tasks also.

Classifier Design as Domain Adaptation Classifiers in source domain might not be effective in target domain because of the distribution mismatch. Obtaining labeled data for each new domain seems to be a costly practice. We present an unsupervised domain adaptation strategy which needs labeled data only in the source domain. We align the source and the target subspaces to reduce the mismatch between the source and the target domains. Our approach not only learns subspaces by utilizing the label information present in the source domain, but also preserves the local neighborhood of the source as well as target data points. Hence the subspaces are discriminative in nature.

We show the superiority of our approach over other existing approaches by showing significant improvement in classification over previous methods. We have shown the application of our approach for the digit classification task, however, our approach can be used for domain adaptation in other tasks also.

Domain adaptation in object recognition Dataset shift poses serious challenges for the object recognition task. We have presented a partially shared dictionary learning approach for tackling this challenge. Our approach is built on the intuition that same category images across the domains do have some similarities/common features. We try to extract such common features from both the domains. A source classifier that is trained using such common features will be effective while classifying images from the target domain.

Our approach allows dictionaries from the source and the target domains to share some atoms. We use the shared atoms to represent signals from both the domains. The atoms that are not shared represent the domain specific information, and hence, are ignored while training cross-domain classifiers. We show the effectiveness of our approach by performing cross-domain object recognition on the benchmark datasets. We also compare our approach with various existing approaches and show improvement in results over the existing state of the art approaches.

Our ultimate goal is to design Computer Vision systems that need to be trained just once, and could perform well in the real world without any further human intervention. In this thesis, we have suggested approaches which take us further towards achieving this goal. We have shown the effectiveness of our approaches by improving over the state of the art approaches in the various tasks considered in this thesis. However, we further need to develop approaches which result in acceptable performance in majority of scenarios, and scale these approaches to large datasets. We will be focusing on these issues in our future work.

Future Work

- The object recognition datasets which has been used in this theses are relatively small in size in comparison to the real world image databases. We would like to scale up the techniques presented in this theses in order to tackle novel scenarios in large datasets.
- We would also try to learn the font/style independent features from a large collection of document images.
- Another future work direction is to learn discriminative shared dictionaries by utilizing available label information.

Related Publications

1. Viresh Ranjan, Gaurav Harit and C.V. Jawahar: Enhancing World Image Retrieval in Presence of Font Variations, International Conference on Pattern Recognition, 2014 (Oral)

2. Viresh Ranjan, Gaurav Harit and C.V. Jawahar: Document Retrieval with Unlimited Vocabulary , IEEE Winter Conference on Applications of Computer Vision(WACV), 2015

3. Viresh Ranjan, Gaurav Harit and C.V. Jawahar: Learning Partially Shared Dictionaries for Domain Adaptation , 12th Asian Conference on Computer Vision (ACCV 2014) (Workshop: FSLCV 2014)

4. Viresh Ranjan, Gaurav Harit and C.V. Jawahar: Domain Adaptation by Aligning Locality Preserving Subspaces, 8th International Conference on Advances in Pattern Recognition(ICAPR 2015)

Bibliography

[1] Digital library of india. http://dli.iiit.ac.in.

[2] Universal library. http://www.ulib.org.

[3] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 2006.

[4] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, 1974.

[5] J. Almazán, D. Fernández, A. Fornés, J. Lladós, and E. Valveny. A coarse-to-fine approach for handwritten word spotting in large scale historical documents collection. In *Proc. ICFHR*, 2012.

[6] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Efficient Exemplar Word Spotting. In *Proc. BMVC*, 2012.

[7] V. Ambati, N. Balakrishnan, R. Reddy, L. Pratha, and C. Jawahar. The digital library of india project: process, policies, and architecture. In *International Conference on Digital Libraries (ICDL)*, 2006.

[8] D. Arya, C. Jawahar, C. Bhagvati, T. Patnaik, B. Chaudhuri, G. Lehal, S. Chaudhury, and A. Ramakrishna. Experiences of integration and performance testing of multilingual OCR for printed indian scripts. In *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*. ACM, 2011.

[9] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011.

[10] A. Balasubramanian, M. Meshesha, and C. Jawahar. Retrieval from document image collections. In *Proc. Document Analysis Systems (DAS)*, 2006.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer vision and image understanding*.

[12] S. M. Beitzel, E. C. Jensen, and D. A. Grossman. A survey of retrieval strategies for OCR text collections. In *Proceedings of the Symposium on Document Image Understanding Technologies*, 2003.

[13] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2001.

[14] A. Bhardwaj, D. Jose, and V. Govindaraju. Script independent word spotting in multilingual documents. In *Proceedings of the 2nd workshop on Cross Lingual Information Access (CLIA) Addressing the Information Need of Multilingual Societies*, 2008.

[15] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[16] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2006.

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

[18] D. Cai, X. He, J. Han, and H.-J. Zhang. Orthogonal laplacianfaces for face recognition. *Image Processing, IEEE Transactions on*, 2006.

[19] H. Cao and V. Govindaraju. Vector Model Based Indexing and Retrieval of Handwritten Medical Forms. In *Proc. ICDAR*, 2007.

[20] K. Chatfield and A. Zisserman. Visor: Towards on-the-fly large-scale object category retrieval. In *Proc. ACCV*, 2012.

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

[22] W. B. Croft, S. M. Harding, K. Taghva, and J. Borsack. An evaluation of information retrieval accuracy with simulated OCR output. In *Proceedings of the Symposium on Document Analysis and Information Retrieval*, 1994.

[23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition. CVPR. IEEE Computer Society Conference on*. IEEE, 2005.

[24] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[25] I. Davidson and S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. SIAM, 2005.

[26] D. L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 2006.

[27] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank. Domain transfer SVM for video concept detection. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2009.

[28] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*. IEEE, 2006.

[29] Y. C. Eldar, P. Kuppinger, and H. Bolcskei. Block-sparse signals: Uncertainty relations and efficient recovery. *Signal Processing, IEEE Transactions on*, 2010.

[30] K. Engan, S. O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*. IEEE, 1999.

[31] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2006.

[32] B. Fernando, A. Habrard, M. Sebban, T. Tuytelaars, et al. Unsupervised visual domain adaptation using subspace alignment. *Proceedings of ICCV*, 2013.

[33] V. Frinken and 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.

[34] B. Gatos, N. Stamatopoulos, and G. Louloudis. ICDAR handwriting segmentation contest. *International Journal of Document Analysis and Recognition, special issue on performance evaluation*, 2011.

[35] M. Gharbi, T. Malisiewicz, S. Paris, and F. Durand. A Gaussian Approximation of Feature Space for Fast Image Similarity. Mit techinical report, 2012.

[36] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2012.

[37] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), IEEE International Conference on*. IEEE, 2011.

[38] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*. 2008.

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

[40] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[41] B. Hariharan, J. Malik, and D. Ramanan. Discriminative Decorrelation for Clustering and Classification. In *Proc. ECCV*, 2012.

[42] X. He and P. Niyogi. Locality preserving projections. In *NIPS*, 2003.

[43] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 1933.

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

[45] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2012.

[46] J. Jiang. A literature survey on domain adaptation of statistical classifiers. 2008.

[47] L. Jin, K. Ding, and Z. Huang. Incremental learning of lda model for chinese writer adaptation. In *Neurocomputing*, 2010.

[48] H. Jing. Using hidden markov modeling to decompose human-written summaries. *Computational Linguistics*, 2002.

[49] W. Kienzle and K. Chellapilla. Personalized Handwriting Recognition via Biased Regularization. In *Proc. IEEE Int'l Conf. Machine Learning (ICML)*, 2006.

[50] 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*, 1997.

[51] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. J. Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *International Journal of Document Analysis and Recognition (IJDAR)*, 2007.

[52] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2011.

[53] A. Kumar, C. V. Jawahar, and R. Manmatha. Efficient search in document image collections. In *ACCV*, 2007.

[54] Y. LeCun and C. Cortes. The MNIST database of handwritten digits, 1998.

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

[56] Y. Leydiera, F. Lebourgeoisb, and H. Emptozb. Text search for medieval manuscript images. *Pattern Recognition*, 2007.

[57] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, IEEE international conference on*. IEEE, 1999.

[58] S. Maji and J. Malik. Fast and accurate digit classification. *EECS Department, University of California, Berkeley, Tech. Rep.*, 2009.

[59] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *Proc. ICCV*, 2011.

[60] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[61] R. Manmatha and W. Croft. Word spotting: Indexing hand-written manuscripts. In *Intelligent Multi-media Information Retrieval Collection*, 1997.

[62] R. Manmatha, C. Han, and E. Riseman. Word spotting: A new approach to indexing handwriting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996.

[63] S. Marinai, E. Marino, and G. Soda. Font adaptive word indexing of modern printed documents. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006.

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

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

[66] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing*, 1999.

[67] R. Milewski and V. Govindaraju. Extraction of handwritten text from carbon copy medical image forms. In *Document Analysis Systems (DAS)*, 2006.

[68] E. Mittendorf, P. Schuble, and P. Sheridan. Applying probabilistic term weighting to OCR text in the case of a large alphabetic library catalogue. In *In ACM SIGIR Conference on R&D in Information Retrieval*, 1995.

[69] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. VISSAPP*, 2009.

[70] M. Müller. *Information retrieval for music and motion*. Springer, Heidelberg, 2007.

[71] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2013.

[72] M. Ohta, A. Takasu, and J. Adachi. Retrieval methods for english-text with missrecognized OCR characters. In *International Conference on Document Analysis and Recognition (ICDAR)*, 1997.

[73] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 1975.

[74] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *AAAI*, 2008.

[75] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Transactions on*, 2011.

[76] S. J. Pan and Q. Yang. A survey on transfer learning. In *IEEE Trans. Knowledge and Data Engineering*, 2010.

[77] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*. 2010.

[78] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *Computer Vision–ECCV*. Springer, 2012.

[79] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

[80] T. M. Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR)*, 2003.

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

[82] T. M. Rath and R. Manmatha. Word spotting for historical documents. *International Journal of Document Analysis and Recognition (IJDAR)*, 2007.

[83] T. M. Rath, R. Manmatha, and V. Lavrenko. A search-engine for historical manuscript images. In *Proceedings of 27th annual international ACM SIGIR conference on research and development in information retrieval*, 2004.

[84] J. L. Rothfeder, S. Feng, and T. M. Rath. Using corner feature correspondences to rank word images by similarity. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2003.

[85] R. Saabni and J. El-sana. Word spotting for handwritten documents using chamfer distance and dynamic time warping. In *Document Recognition and Retreival (DRR)*, San Francisco, USA, 2011.

[86] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*. Springer, 2010.

[87] K. P. Sankar and C. V. Jawahar. Probabilistic reverse annotation for large scale image retrieval. In *Proc. CVPR*, 2007.

[88] K. P. Sankar, C. V. Jawahar, and R. Manmatha. Nearest neighbor based collection OCR. In *Proc. DAS*, 2010.

[89] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. In *Neural computation*, 1998.

[90] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two patterns. In *Proc. Royal Society London*, 1991.

[91] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *Proc. ICML*, 2007.

[92] K. P. Shankar, C. V. Jawahar, and R. Manmatha. Nearest neighbor based collection OCR. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, 2010.

[93] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. Generalized domain-adaptive dictionaries. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2013.

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

[95] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. In *Neural Computation*, 2000.

[96] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2011.

[97] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 2007.

[98] S. Tulyakov and V. Govindaraju. Probabilistic model for segmentation based word recognition with lexicon. In *Proc. Sixth International Conference on Document Analysis and Recognition (ICDAR)*, 2001.

[99] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, 2005.

[100] I. Z. Yalniz and R. Manmatha. An efficient framework for searching text in noisy document images. In *Proc. DAS*. IEEE, 2012.

[101] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *Proceedings of the 15th international conference on Multimedia*. ACM, 2007.

[102] B. Zhang, S. N. Srihari, and C. Huang. Word image retrieval using binary features. In *Proc. SPIE 5296, Document Recognition and Retrieval XI, 45*, Dec 2003.

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

[104] J. Zheng, Z. Jiang, P. J. Phillips, and R. Chellappa. Cross-view action recognition via a transferable dictionary pair. In *BMVC*, 2012.

[105] X. Zhu and A. B. Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 2009.