WRITER IDENTIFICATION BASED ON COVARIANCE FEATURES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
ÇANKAYA UNIVERSITY

BY
TALHA KARADENİZ

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
COMPUTER ENGINEERING

NOVEMBER 2016

Title of the thesis: **Writer Identification Based on Covariance Features**
Submitted by: **Talha Karadeniz**

Approval of Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Halil Tanyer Eyyuboğlu
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Müslim Bozyiğit
Head of Department

This is to certify that we have read this thesis and in our opinion it is fully adequate, in scope and quality, as a thesis for the degree Master of Science.

Assoc. Prof. Dr. H. Hakan Maraş
Supervisor

Examination Date: 08.16.2016

Examining Members:

Assoc. Prof. Dr. H. Hakan MARAŞ	(Çankaya Univ.):	
Asst. Prof. Dr. Abdül Kadir GÖRÜR	(Çankaya Univ.):	
Asst. Prof. Dr. Bülent G. EMİROĞLU	(Kırıkkale Univ):	

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document is obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Talha Karadeniz

Signature:

Date: 08.16.2016

ABSTRACT

WRITER IDENTIFICATION BASED ON COVARIANCE FEATURES

KARADENİZ, Talha
M.Sc., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. H. Hakan Maraş

November 2016, 30 pages

Local descriptors have been widely utilized in image analysis for automatic object categorization. In this work, an algorithm based on empirical covariance estimation of region descriptor vectors is formulated and developed. This technique is then specialized in order solve to the task of writer identification via a tricky way of keypoint extraction. Experiment results are reported for ETH-80 and ICFHR 2012 Writer Identification Contest datasets.

Keywords: Vector Set Kernels, Local Descriptors, Object Categorization, Empirical Covariance Matrix, Writer Identification

ÖZET

KOVARYANS NİTELİKLERİNE DAYALI YAZAR BELİRLEME

KARADENİZ, Talha
Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. H. Hakan Maraş

Kasım 2016, 30 sayfa

Görüntü analizinde lokal tanımlayıcılar otomatik nesne tasnifi için kullanılagelmiştir. Bu çalışmada, bölge tanımlama vektörlerinin empirik kovaryans tahminine dayalı bir algoritma formüle edilmiş ve geliştirilmiştir. Daha sonra, bu teknik, yazar belirleme görevi için, hususi bir anahtar nokta çıkarma usulü vasıtasıyla, özelleştirilmiştir. ETH-80 ve ICFHR 2012 Yazar Belirleme Yarışması verisetleri üzerinde gerçekleştirilmiş deneylerin sonuçları raporlanmıştır.

Anahtar Kelimeler: Vektör Cümlesi Kernel'leri, Lokal Tanımlayıcılar, Nesne Tasnifi, Empirik Kovaryans Matrisi, Yazar Belirleme

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This work is based on **vector set analysis**, by which we mean statistical modeling on feature multitudes. These multitudes can be collections of fixed D-dimensional vectors. That is, a collection $\mathbf{U}$ is considered as a **vector set** if any $\mathbf{u} \in \mathbf{U}$ is in $\mathbb{R}^D$. We concentrate specifically on categorization task. This will be in two ways: in chapter 2 we will focus on the labeling of images via covariance features. In chapter 3 we will move towards the challenge of automatic writer identification.

1.1 Object Categorization and Writer Identification

Object categorization is the task of automatically classifying objects. One can list many applications of this field by stating security systems [20], aided repairment via augmented reality [22] and robotics technology[20]. Writer identification, on the other hand, is a forensic-focused study of labeling hand-written texts, which is, apart from the image data content, closely related to signature verification. Notwithstanding the naming intention behind object categorization, by which one generally think about rigid or real-life bodies, writer identification can be seen as an extended object categorization task; or rather, one can say that, these can be grouped closely under study on pattern categorization. In spite of the difference between the kind of uniformity of features searched in object categorization and writer identification, it is completely normal to expect a technique constructed for the first one works for the second.

1.2 Motivation

The main motivation behind this study was first 'simple' innovation – especially on object categorization, during a multimedia analysis project at Tubitak Uzay – and then an introduction of a security-focused or rather a can-be-adapted-to-security routine; hence Automatic Identification. Since an increasing need for writer identification is observed in the last years, organizations may benefit from such automatizations, for example, on the area of digital rights management and batch document operations[14]. Without a towards-security goal, some of the 'novelties' introduced in this work would not exist. [1]

1.3 Route

Most of the framework is given in chapter 2; chapter 4 is a simplified application of the ideas noted in the first one. Hence the reader should not be suprised by the volume and density difference of these two.

In chapter 2, a definition of object categorization task is given. Then, a brief explanation of Support Vector Machines is given along with a list of implementation strategies. Afterwards, formal definitions of kernel mappings is written to form the basics of vector set kernels. This is followed by the examples and formulations of two known vector set kernels. Finally, covariance feature based Power Series Kernel-Linear Kernel combination is introduced before the experiment results. Literature content is kept relatively narrow due to theoretical and influential reasons; focus is directed to the work on vector set kernels.

In the third chapter, generic formulation of writer identification task and a moderately uptodate literature review is given. This is followed by a chapter allocated to the algorithm details. 'Tricky' keypoint and SIFT-BRIEF descriptor extraction steps are given to summarize the solution route. Most of the notation is based on the work supplied in first chapter. SIFT feature calculation steps and examples are demonstrated in figures.

[1] 'novelties' instead of actual novelties due to the experimental-and-quasi-combinatorial-rather-than-rock-solid-analytic-and-constructive basis behind each.

A contest result as a document of feasibility is listed at the end.

In spite of the simple idea of featurization of covariance entries, when combined with other components such as matrix exponentiation and power series implicit mapping, introduced algorithm base is hopefully shown to be a candidate for future study. The work is voluntarily kept short to make the content as dense as possible.

CHAPTER 2

BACKGROUND STUDY

Assume that we have a collection of images $S = \{I_1, I_2, ..., I_N\}$ and the corresponding labels $C = \{c_1, c_2, ..., c_N\}$ where each I_i is of type category c_i. Given a training set S, object categorization task is the challenge of finding a classifier model which is suitable for the correct labeling of any test image. This is for sure an example of machine learning analysis where an automatic labeler routine is derived from the training set.

2.1 Vector Set Analysis

As noted in chapter 1, Vector Set Analysis is the job of featurization of vector sets or deriving kernels from these ready to be utilized in kernel machines. Actually, apart from the abstract mathematical sense of the word 'analysis', here we refer rather to a data inquiry kind. Any fixed dimensional unordered vector collection is a vector set and in the following sections, kernel machines forged upon vector set relations are explored.

2.2 Object Categorization

Given the definition at the start of this chapter, it is now suitable to list the components of a vector set categorization route.

2.2.1 Support Vector Machines

Support Vector Machine [SVM] is one of the state-of-the-art classification methods. It is based on finding an optimal hyperplane between two sample sets via using kernel functions [2]. For a given input space X, a function $K : X \times X \rightarrow \mathbb{R}$ is a kernel if and only if it corresponds to an inner product in some feature space F. That is, for any $\mathbf{u}, \mathbf{v} \in X$, one must have $K(\mathbf{u}, \mathbf{v}) = \langle \phi(\mathbf{u}), \phi(\mathbf{v}) \rangle$. The function $\phi : X \rightarrow F$ is considered as an **implicit mapping** since one does not need the explicit representations of $\phi(\mathbf{u})$ and $\phi(\mathbf{v})$ to use the kernel values in optimal hyperplane calculation.

Let $X = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_l}\}$, be the input vectors and y_i be the corresponding labels for $i = 1, 2, ..., l$. Let K be a kernel, C be the penalty parameter. Then SVM optimization is of the form [1]:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{l} \alpha_i + \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x_i}, \mathbf{x_j})$$

$$\text{subject to } 0 \leq \alpha_i \leq C \ \forall i \tag{2.1}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.2}$$

$$\tag{2.3}$$

One of the generic techniques for finding solutions of the stated optimization progam is known as 'chunking', where a chunk of dataset is chosen as the initial working set [2]. Then, a training is done with respect to this collection and the hypothesis is validated against remaining vectors. Samples, on which the most violation of Karush-Kuhn-Tucker conditions are encountered, are added to support vectors to begin a new training session. This loop is kept until the stopping criteria is satisfied. In each step, a generic quadratic optimization routine is ran.

Decomposition, on the other hand, is a more economic version of chunking, in which the number of support vectors are kept constant. Again, a generic quadratic optimization routine is applied to the dataset at each step [2].

Finally, a fast way to solve this quadratic optimization problem is Sequential Minimal Optimization, where the **analytical** solutions of narrower sub-problems are composed to arrive the final optimized coefficients [1]. Main difference of SMO from chunking and decomposition is the size of sub-problems handled: at each iteration only 2 coefficients are changed. Furthermore, this is done in an analytical way; an exact solution is found for the coefficients.

Support Vector Machines are one well-known example of kernel machines. At the heart of this kind of machines, as it can be deduced from the name, there exists the core of kernel functions. These are similarity measures between vectors – or data structures such as strings and graphs – which are calculated to optimize the hypothesis. Measures can be combined in various algebraic ways so that more complex affinity elements can be constructed. Before going on, let's give the fundamental definition of a kernel.

Definition 1. *A function $K(\mathbf{u}, \mathbf{v}) : X \times X \rightarrow \mathbb{R}$ is a kernel if and only if it corresponds to an inner product $K(\mathbf{u}, \mathbf{v}) = \langle \phi(\mathbf{u}), \phi(\mathbf{v}) \rangle$ in a feature space F. [2]*

2.2.2 Vector Set Kernels

Standard kernels such as linear, polynomial and rbf are utilized for fixed-dimensional vector categorization [8], i.e. all the inputs are elements of $\mathbb{R}^D$. Although this is the case for conventional SVM applications, there also exists kernels of graphs, strings and manifolds [5]. In [8] and [5], scientists concentrate on vector set kernels. That is, they explored the kernels of type $K : X \times X \rightarrow \mathbb{R}$, where the elements of X are vector sets, each consisting of D-dimensional vectors. The expression $K(\mathbf{U}, \mathbf{V})$ represents the kernel between vector sets $\mathbf{U}$ and $\mathbf{V}$, which may have different cardinalities.

Pyramid Match Kernel [8] is calculated as weighted sum of multi-resolution histogram intersections. Let $\mathbf{U}$ and $\mathbf{V}$ be vector sets, $H_i(\mathbf{U})$, $H_i(\mathbf{V})$ be the i-th level histograms of $\mathbf{U}$ and $\mathbf{V}$, respectively. If $\mathcal{I}(H_i(\mathbf{U}), H_i(\mathbf{V}))$ is the histogram intersection function, L is the number of levels, then the kernel value is found by the following formula:

$$K_{pmk}(\mathbf{U},\mathbf{V}) = \sum_{i=0}^{L} \frac{1}{2^i}(\mathcal{I}(H_i(\mathbf{U}), H_i(\mathbf{V})) - \mathcal{I}(H_{i-1}(\mathbf{U}), H_{i-1}(\mathbf{V}))) \tag{2.4}$$

The meaning of the coefficient $\frac{1}{2^i}$ is that we assign lower weights to the coarse levels and higher importance to finer levels. At each level, we only count the new intersection counts and hence the difference operator. Since each intersection is a kernel, sum of all intersection values is also a kernel [2].

Bhattacharyya kernel on the other hand is calculated via fitting of Gaussians to vector sets [5]. Let $\mathbf{U} = \{\mathbf{u_1}, \mathbf{u_2}, ..., \mathbf{u_{N_U}}\}$, $\mathbf{V} = \{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_{N_V}}\}$. We fit two probability distributions p_U and p_V to $\mathbf{U}$ and $\mathbf{V}$ respectively. Bhattacharyya distance between these vectors are found via the integral:

$$K(\mathbf{U},\mathbf{V}) = \int p_U(\mathbf{w}) p_V(\mathbf{w}) \mathrm{d}\mathbf{w} \tag{2.5}$$

Let

$$\mu_{\mathbf{U}} = (1/N_U) \sum_{i=1}^{N_U} \mathbf{u_i} \tag{2.6}$$

,

$$\mu_{\mathbf{V}} = (1/N_V) \sum_{i=1}^{N_V} \mathbf{v_i} \tag{2.7}$$

be the mean vectors of sets $\mathbf{U}$, $\mathbf{V}$, respectively.

Let

$$\mathbf{\Sigma_U} = (1/N_U) \sum_{i=1}^{N_U} (\mathbf{u_i} - \mu_{\mathbf{U}})(\mathbf{u_i} - \mu_{\mathbf{U}})^T \tag{2.8}$$

$$\mathbf{\Sigma_V} = (1/N_V) \sum_{i=1}^{N_V} (\mathbf{v_i} - \mu_{\mathbf{V}})(\mathbf{v_i} - \mu_{\mathbf{V}})^T \tag{2.9}$$

be the empirical covariance matrices of $\mathbf{U}$ and $\mathbf{V}$. Define

$$\mathbf{\Sigma}' = (\frac{1}{2}\mathbf{\Sigma_U^{-1}} + \frac{1}{2}\mathbf{\Sigma_V^{-1}})^{-1} \tag{2.10}$$

and

$$\mu^{'} = (\frac{1}{2}\mathbf{\Sigma_U^{-1}}\mu_U + \frac{1}{2}\mathbf{\Sigma_V^{-1}}\mu_V) \tag{2.11}$$

Then the explicit form of this kernel is as follows [5]:

$$\begin{aligned} K(\mathbf{U},\mathbf{V}) = |\mathbf{\Sigma_U}|^{\frac{-1}{4}}|\mathbf{\Sigma_V}|^{\frac{-1}{4}}|\mathbf{\Sigma}^{'}|^{\frac{1}{2}} \\ exp(\frac{-1}{4}\mu_\mathbf{U}{}^T\mathbf{\Sigma_U^{-1}}\mu_\mathbf{U}+ \\ \frac{-1}{4}\mu_\mathbf{V}{}^T\mathbf{\Sigma_V^{-1}}^{\mu_\mathbf{V}}+ \\ \frac{1}{2}\mu^{'T}\mathbf{\Sigma}^{'-1}\mu^{'}) \end{aligned} \tag{2.12}$$

Influenced from works [5], [8], we have found that the following naive kernel yields accurate results in object categorization.

Let $\theta : \mathbb{R}^{D\times D} \rightarrow \mathbb{R}^{D^2}$ be the row-concatenation function

$$\begin{aligned} \theta(\begin{bmatrix} a_{11} & a_{12} & ... & a_{1D} \\ a_{21} & a_{22} & ... & a_{2D} \\ ... & & & \\ a_{D1} & a_{D2} & ... & a_{DD} \end{bmatrix}) = (a_{11}, a_{12}, ..., a_{1D}, \\ a_{21}, a_{22}, ..., a_{2D}, \\ ..., \\ a_{D1}, a_{D3}, ..., a_{DD}) \end{aligned} \tag{2.13}$$

Define the mapping $\phi_1 : X \rightarrow \mathbb{R}^{D^2}$ with

$$\phi_1(\mathbf{U}) = \theta(\Sigma_U) \tag{2.14}$$

and the kernel

$$K_1(\mathbf{U},\mathbf{V}) = \langle\phi_1(\mathbf{U}), \phi_1(\mathbf{V})\rangle = \langle\theta(\Sigma_U), \theta(\Sigma_V)\rangle \tag{2.15}$$

Actually, this route is a second-level-feature one, by which we mean, not a well-crafted-specifically-for-vector-sets but a tricky-way-of-transforming-vector-set-task-to-vector-task. It is **second-level** because in the first level we have D-dimensional vectors extracted from images. In the second level, we have covariance features extracted from these vector sets.

Okay. We are in the domain of vectors so that we can apply the standard routes or routes influenced from standard ones. We have noted that there exist conventinal kernels such as linear, polynomial and rbf. Now, it will be appropriate to introduce **Power Series Kernel** by which we obtained a slight but not-negligble improvement on the final categorization accuracy [1].

Now, suppose the mapping $\phi_2 : \mathbb{R}^{D^2} \rightarrow \mathbb{R}^{\infty}$ is defined as

$$\begin{aligned}\phi_2(\mathbf{u}) = \phi_2((u_1, u_2, ..., u_{D^2})) = (&u_1, {u_1}^2, ..., {u_1}^k, ..., \\ &u_2, {u_2}^2, ..., {u_2}^k, ..., \\ &..., \\ &u_{D^2}' u_{D^2}^2, ..., u_{D^2}^k, ...)\end{aligned} \tag{2.16}$$

and assume that $|u_i|, |v_i| < 1$ to make the function

$$K_2^{'}(\mathbf{u}, \mathbf{v}) = \langle \phi_2(\mathbf{u}), \phi_2(\mathbf{v}) \rangle = \sum_{i=1}^{D} (1/(1 - u_i v_i) - 1) \tag{2.17}$$

valid.

Remark 1. $K_2^{'}(\mathbf{u}, \mathbf{v})$ *is a kernel on* X *where* X *is the* D*-dimensional space of vectors* $\mathbf{u}$ *for which* $|u_i| < 1$ *holds* .

Proof. Since the feature representation is explicitly given in 2.16, by definition, $\langle \phi_2(\mathbf{u}), \phi_2(\mathbf{v}) \rangle = \sum_{i=1}^{D} \sum_{k=1}^{\infty} (u_i v_i)^k = \sum_{i=1}^{D} (1/(1 - u_i v_i) - 1)$ is a kernel. □

$|u_i|, |v_i| < 1$ condition is necessary for the series $\sum_{k=1}^{\infty} (u_i v_i)^k$ to converge [17]. Ap-

[1] Although the one in this thesis is formulated independently, an analysis on Power Series Kernels can be read here: Zwicknagl, Barbara. "Power series kernels." Constructive Approximation 29.1 2009: 61-84.

plying this to our covariance features, we get,

$$K_2(\mathbf{U}, \mathbf{V}) = \langle \phi_2(\phi_1(\mathbf{U})), \phi_2(\phi_1(\mathbf{V})) \rangle = \langle \phi_2(\theta(\Sigma_U)), \phi_2(\theta(\Sigma_V)) \rangle \tag{2.18}$$

Lemma 1. *Let K_a, K_b be kernels on some input space X. Then K_1K_2 is also a kernel.*

Proof. Assume that $K_a(\mathbf{u}, \mathbf{v}) = \langle \phi_a(\mathbf{u}), \phi_a(\mathbf{v}) \rangle$ and $K_a(\mathbf{u}, \mathbf{v}) = \langle \phi_b(\mathbf{u}), \phi_b(\mathbf{v}) \rangle$, where $\phi_a : X \to \mathbb{R}^{D_a}$ and $\phi_b : X \to \mathbb{R}^{D_b}$ are the corresponding feature space mappings. Consider $K_{ab}(\mathbf{u}, \mathbf{v}) = \langle \phi_a(\mathbf{u}), \phi_a(\mathbf{v}) \rangle \langle \phi_b(\mathbf{u}), \phi_b(\mathbf{v}) \rangle$. Let ϕ_{ai} and ϕ_{bj} be the projection functions; i.e. $\phi_{ai}(\mathbf{u}) = \phi_a(\mathbf{u})_i$ and $\phi_{bj}(\mathbf{u}) = \phi_b(\mathbf{u})_j$. Then explicit form of $K_a(\mathbf{u}, \mathbf{v})$ is

$$\begin{aligned}
&(\sum_{i=1}^{D_a} \phi_{ai}(\mathbf{u})\phi_{ai}(\mathbf{v}))(\sum_{j=1}^{D_b} \phi_{bj}(\mathbf{u})\phi_{bj}(\mathbf{v})) = \\
&\sum_{i=1}^{D_a} \sum_{j=1}^{D_b} \phi_{ai}(\mathbf{u})\phi_{ai}(\mathbf{v})\phi_{bj}(\mathbf{u})\phi_{bj}(\mathbf{v}) = \\
&\sum_{i=1}^{D_a} \sum_{j=1}^{D_b} \phi_{ai}(\mathbf{u})\phi_{bj}(\mathbf{u})\phi_{ai}(\mathbf{v})\phi_{bj}(\mathbf{v})
\end{aligned} \tag{2.19}$$

, which is an inner product in $\mathbb{R}^{D_a \times D_b}$, under the mapping

$$\begin{aligned}
\phi_{ab}(\mathbf{u}) = (&\phi_{a1}(\mathbf{u})\phi_{b1}(\mathbf{u}), \phi_{a1}(\mathbf{u})\phi_{b2}(\mathbf{u}), ..., \phi_{a1}(\mathbf{u})\phi_{bD_b}(\mathbf{u}), \\
&\phi_{a2}(\mathbf{u})\phi_{b1}(\mathbf{u}), \phi_{a2}(\mathbf{u})\phi_{b2}(\mathbf{u}), ..., \phi_{a2}(\mathbf{u})\phi_{bD_b}(\mathbf{u}), \\
&..., \\
&\phi_{aD_a}(\mathbf{u})\phi_{b1}(\mathbf{u}), \phi_{aD_a}(\mathbf{u})\phi_{b2}(\mathbf{u}), ..., \phi_{aD_a}(\mathbf{u})\phi_{bD_b}(\mathbf{u}))
\end{aligned}$$

□

That is, finally, we can state our second kernel as:

$$K(\mathbf{U}, \mathbf{V}) = K_1(\mathbf{U}, \mathbf{V})K_2(\mathbf{U}, \mathbf{V}) \tag{2.20}$$

When it comes to complexity, after asymptotic analysis on (2.12), (2.4) and (2.20), speed complexity list can be obtained as $O(D^3+ND^2)$, $O(NDlog(r))$ [8] and $O(ND^2)$, respectively. Slowness derived from 2.12 is due to matrix inversion. On the other hand, in experiments, duration results yielded by machines of 2.20 were quite comparable to that of 2.4.

As a lower-dimensional version of this covariance analysis, we now introduce an additional kernel, which is based on the exponential matrix:

Definition 2. *Let* $\mathbf{A} \in \mathbb{C}^{DxD}$*, then the limit of the sum* $\mathbf{S_{A,n}} = \mathbf{I_D} + \sum_{k=1}^{n} \frac{\mathbf{A^k}}{k!}$ *is described as* $\lim_{n\to\infty} \mathbf{S}_{\mathbf{A},n} = e^{\mathbf{A}}$ *[15].*

Now, define the mapping $\phi_e : X \to \mathbb{R}^{D^2}$ with

$$\phi_e(\mathbf{U}) = \theta(e^{\mathbf{\Sigma_U}}) \tag{2.21}$$

Remember that θ is the concatenation function (2.13). Additionally, let $\phi_c : X \to \mathbb{R}^{D^2}$ such that $\phi_c(\mathbf{U}) = [\mathbf{\Sigma_U}_{ij}]$. At the last step, concatenate the original empirical covariance estimation with the exponential and the cube-root features and normalize the vectors to obtain

$$(\phi_1(\mathbf{U})/|\phi_1(\mathbf{U})|_1) \oplus (\phi_e(\mathbf{U})/|\phi_e(\mathbf{U})|_1) \oplus (\phi_c(\mathbf{U})/|\phi_c(\mathbf{U})|_2) \tag{2.22}$$

Then, linear kernel for this induced space is

$$\begin{aligned} K_{e1}(\mathbf{U},\mathbf{V}) = \langle &(\phi_1(\mathbf{U})/|\phi_1(\mathbf{U})|_1) \oplus (\phi_e(\mathbf{U})/|\phi_e(\mathbf{U})|_1) \oplus \\ &(\phi_c(\mathbf{U})/|\phi_c(\mathbf{U})|_2), \\ &(\phi_1(\mathbf{V})/|\phi_1(\mathbf{V})|_1) \oplus (\phi_e(\mathbf{V})/|\phi_e(\mathbf{V})|_1) \oplus \\ &(\phi_c(\mathbf{V})/|\phi_c(\mathbf{V})|_2)\rangle \end{aligned} \tag{2.23}$$

which is equal to the statement

$$
\begin{aligned}
&\langle \phi_1(\mathbf{U})/|\phi_1(\mathbf{U})|_1, \phi_1(\mathbf{V})/|\phi_1(\mathbf{V})|_1\rangle + \\
&\langle \phi_e(\mathbf{U})/|\phi_e(\mathbf{U})|_1, \phi_e(\mathbf{V})/|\phi_e(\mathbf{V})|_1\rangle + \\
&\langle \phi_c(\mathbf{U})/|\phi_c(\mathbf{U})|_2, \phi_c(\mathbf{V})/|\phi_c(\mathbf{V})|_2\rangle
\end{aligned}
\tag{2.24}
$$

and the Power Series Kernel is

$$
\begin{aligned}
K_{e2}(\mathbf{U},\mathbf{V}) = \langle &\phi_2(\phi_1(\mathbf{U})/|\phi_1(\mathbf{U})|_1) \oplus \phi_2(\phi_e(\mathbf{U})/|\phi_e(\mathbf{U})|_1) \oplus \\
&\phi_2(\phi_c(\mathbf{U})/|\phi_c(\mathbf{U})|_2), \\
&\phi_2(\phi_1(\mathbf{V})/|\phi_1(\mathbf{V})|_1) \oplus \phi_2(\phi_e(\mathbf{V})/|\phi_e(\mathbf{V})|_1) \oplus \\
&\phi_2(\phi_c(\mathbf{V})/|\phi_c(\mathbf{V})|_2)\rangle
\end{aligned}
\tag{2.25}
$$

Similar to (2.20), the matrix exponential version is:

$$
K_e(\mathbf{U},\mathbf{V}) = K_{e1}(\mathbf{U},\mathbf{V})K_{e2}(\mathbf{U},\mathbf{V}) \tag{2.26}
$$

To our knowledge, the matrix exponential is not utilized by anyone so far in this context; that is, in conjunction with empirical covariance estimation of local descriptors.

As a last kernel candidate, let's concentrate on a substitue of component covariation measure. Normally, if we have two variables X and Y, we measure the covariance by

$$
C_{XY} = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y}) \tag{2.27}
$$

where $\bar{X}$, $\bar{Y}$ and N are sample mean of X, sample mean of Y and total number of observations, respectively. The rationale behind this calculation is summing up the centered values of each variable to arrive a scalar of co-linearity; if $C_{XY} > 0$ then one can say that he has two variables changing 'together'. Otherwise, if $C_{XY} < 0$, then he can deduce that the change is again 'together' but this time at opposite directions. If $C_{XY} = 0$, then there exists no covariation.

Remark 2 (Triangle Inequality). *For any x, $y \in \mathbb{R}$, we have $|x + y| \leq |x| + |y|$.*

Proof. If $x + y \geq 0$ then $|x + y| = x + y \leq |x| + |y|$ since $x \leq |x|$ and $y \leq |y|$. Otherwise if $x + y < 0$ then $|x + y| = -x + -y \leq |x| + |y|$ because since $-x \leq |x|$ and $-y \leq |y|$. □

Lemma 2. *If $sgn(x) = sgn(y)$ then $|x+y| - |x| - |y| = 0$. Otherwise $|x+y| - |x| - |y| < 0$.*

Proof. Assume that $sgn(x) = sgn(y) = -1$. $\Rightarrow |x+y| = -x - y$ since $sgn(x+y) = -1$. Thus, $|x + y| - |x| - |y| = -x - y - (-x) - (-y) = -x - y + x + y = 0$. Suppose that $sgn(x) = sgn(y) = 1$. $\Rightarrow |x + y| - |x| - |y| = x + y - x - y = 0$. If $sgn(x) \neq sgn(y)$, then either $|x| > |y|$ or $|y| < |x|$. $\Rightarrow |x + y| < |x|$ because we have opposite signs for x and y. Therefore $|x + y| < |x| \leq |x| + |y|$. This can also applied to the case $|y| < |x|$. □

Thus, one can integrate absolute value comparisons to strengthen the covariation measurement, for a sign difference implies a negative $|x + y| - |x| - |y|$ value and hence a negative covariation. We have found that the following estimate is complementary to the covariance summation stated in (2.27):

$$C'_{XY} = \frac{1}{N} \sum_{i=1}^{N} (|X_i - \bar{X} - (Y_i - \bar{Y})| - |X_i - \bar{X}| - |(Y_i - \bar{Y})|)^2 - \\ (|X_i - \bar{X} + (Y_i - \bar{Y})| - |X_i - \bar{X}| - |(Y_i - \bar{Y})|)^2 \tag{2.28}$$

At the formation of the modified covariance matrix, we take $C_{X_i X_j}$ (2.27) if $i > j$, otherwise we choose $C'_{X_i X_j}$ (2.28).

Now, let's concentrate on object categorization: for an image I_i, consider SIFT [6] descriptors extracted from uniformly sampled grids of I_i. Let $\mathbf{U}$ be this descriptor collection. Similarly, let $\mathbf{V}$ be the multitude extracted from another image I_j. It is obvious that we can use our kernel stated in (2.20) between these two images. Note that we do not define a standard kernel between vectors-extracted-from-images – which is the case for the standard route – but rather formulated one between vector-sets-extracted-from-images. By formulations (2.4), (2.12) and (2.20), one can construct kernels between vector sets of different cardinalities, albeit the constant-cardinality nature of uniformly sampled grids noted here.

For the lower-dimensional case, instead of direct SIFT, PCA-SIFT descriptors [7, 8] are taken as the base vector sets. Then these sets are categorized through the exponentiated, standard or cube-rooted empirical covariance features (2.26).

Each kernel matrix, that is, the matrix of all pairwise kernel, is normalized according to the formula [8]:

$$K_{\Delta}(\mathbf{U}, \mathbf{V}) = \frac{K(\mathbf{U}, \mathbf{V})}{\sqrt{K(\mathbf{U}, \mathbf{U}) K(\mathbf{V}, \mathbf{V})}} \tag{2.29}$$

2.3 Experiments

In the former section, some known vector set kernels are formulated and examples based on covariance matrices are built with or without extensions of Power Series Kernels. In this section, short reports on an idealized-environment dataset will be given.

Figure 2.1: Apple

Figure 2.2: Car

2.3.1 Dataset

For experiments, we followed the route explained in [8]. ETH-80 dataset is used for categorization accuracy measurements. The experiment set is a collection of 400 images, containing 8 category of objects. Each category is formed by 10 objects, where each object's 5 different poses are tested in an experiment. For each category, a one-vs-all SVM is trained. The total number of experiments run is 80.

Figure 2.3: Cup

Each image's vector set is composed of uniformly sampled grid SIFT descriptors [2]. Experiment code is written as an extension of libpmk-2.5 [3] via noweb [4]. Noweb is chosen for documentation since it is a flexible literate programming [5] tool.

2.3.2 Results

The results are evaluated with respect to categorization accuracy. The ratio of number of correct labelings to the total number of tests is shown in Table 2.1 and Table 2.3. Corresponding accuracy measures are given in Table 2.2 and Table 2.4. Abbreviations:

- **CFLK**: Covariance Features with Linear Kernel (2.15)
- **CFCK**: Covariance Features with Combined Kernel (2.20)
- **ECFCK**: Exponential Covariance Features with Combined Kernel (2.26)
- **ECFCK-II**: Exponential Covariance Features with Combined Kernel via alternating triangle inequality measures (2.28)
- **PMK**: Pyramid Match Kernel (2.4)

2.3.3 Analysis

In the case of low-dimensionality, that is, the task of classifiying PCA-SIFT collections, from these tables, one should not deduce that **ECFCK** is absolutely more accurate than **PMK**. These PCA-SIFT results are reported according to the basic configuration of PMK: **number of branches** and **number of levels** are taken 3 and 11, respectively [6]. When these values are increased, numbers ranging between $[0.80, 0.85]$ can be observed; the total run-time also increases, though.

[2] http://people.csail.mit.edu/jjl/libpmk/samples/eth.html

[3] http://people.csail.mit.edu/jjl/libpmk/

[4] https://www.cs.tufts.edu/~nr/noweb/

[5] Knuth, Donald Ervin. "Literate programming." The Computer Journal 27.2 (1984): 97-111.

[6] An example run from the engineers of PMK is logged at: http://people.csail.mit.edu/jjl/libpmk/samples/eth.html

Table 2.1: Rates of correct categorization

Object Category	CFLK	CFCK	PMK
#1	42/50	**45/50**	38/50
#2	50/50	**50/50**	49/50
#3	40/50	**43/50**	40/50
#4	45/50	**48/50**	45/50
#5	**45/50**	44/50	33/50
#6	**48/50**	47/50	33/50
#7	50/50	**50/50**	50/50
#8	**50/50**	49/50	47/50

Table 2.2: Average accuracy

CFLK	CFCK	PMK
0.925	**0.940**	0.837

When it comes to note the reason behind the success obtained through exponentiation of covariance matrix, as a first step, one can show that an element of the infinite series is a polynomial kernel collection of the covariance vectors. This collection is weighted with a decreasing coefficient so that the original covariance structure is conserved. One interesting side of the technique is the exploitation of variance-covariance kernel measures as features of Support Vector Machines. Note that these kernels are not directly transfered to SVM but rather calculated as vector components ready to be inputs of Power Series Kernel-Linear Kernel combination.

One drawback related to reliability of **CFCK** results, is the cube-root feature transformation and scaling of input components. Although in experiments, any combination

Table 2.3: Rates of correct categorization (PCA-SIFT)

Object Category	ECFCK	ECFCK-II	PMK
#1	40/50	**42/50**	36/50
#2	49/50	**49/50**	47/50
#3	**37/50**	37/50	33/50
#4	44/50	**46/50**	45/50
#5	**30/50**	29/50	29/50
#6	34/50	**36/50**	30/50
#7	**50/50**	50/50	49/50
#8	42/50	**43/50**	42/50

Table 2.4: Average accuracy (PCA-SIFT)

ECFCK	ECFCK-II	PMK
0.815	**0.830**	0.777

of **PMK** parameters yielded a lower accuracy rate, one cannot generalize this to an absolute success devised from machines of **CFCK** over to that of **PMK**. One may see this as a mandatory or usual item of data mining experiments [3], but since such a grid-search is not applied on the **PMK** machines, it can be thought as a drawback of the technique stated and formulated in this work, albeit not a major one.

CHAPTER 3

LITERATURE REVIEW

Assume that we have a collection of hand-written paragraph images $S = \{\{I_{11}, I_{12}, ..., I_{1K}\}, \{I_{21}, I_{22}, ..., I_{2K}\}..., \{I_{N1}, I_{N2}, ..., I_{NK}\}\}$, the corresponding writer labels $C = \{\{c_1, c_1, .., c_1\}, \{c_2, c_2, .., c_2\}, ..., \{c_N, c_N, .., c_N\}\}$ where each I_{ij} is composed by user c_i. A total of K distinct paragraphs are exampled by N writers; that is, all I_{ij} is of the same content T_i. Writer identification is the challenge of finding a classifier model induced from S which is suitable for the correct identification of any test image.

To automatically identify writers, several methods have been engineered so far. Along with the classical approaches consisting segmentation-based combination of macro-micro attributes [4, 10], bag of features [BoF] techniques have also been tested: a framework built upon K-adjacent Segment features [13] or a system based on SIFT [6] descriptors [12] are examples of modelling the writer style. The technique evaluated in this work, by its generic basis of local descriptors, is similar to [12] and have been successfully used at ICFHR 2012 Writer Identification Contest [16].

3.1 Macro-Micro Combination

In [4], macro attributes such as gray-level distribution, gray-level threshold and contour variations are calculated on overall image; hence the adjective 'macro'. Micro attributes, on the other hand, are found by an analysis of low-level shape structure. Gradient and concavity bits are these kind of attributes. Since this algorithm is dependent on segmentation and character-level feature extraction, the efficiency derived can be highly affected by the noted steps. When the time of release considered [2002], the work may be seen as novel but if one takes generalization to multi-languages into account, he can conclude that the stated route must be adapted each time to target the input kind.

A text-independent flow is reported at [10], where, similar to [4] a combination of micro and macro attributes is formed to automatically identify the writers. Engineers involved state that distribution functions of directional, grapheme and run-length measurements are utilized successfully to build a model. Final identification and verification is done by nearest-neighbor search with Hamming distance. One advantage of employing this route is independency from cursive-isolated variation.

3.2 Bag of Features Models

Assume we have a collection of feature vectors $U_{all} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N\}$ and a vector set extracted from an image $V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{|V|}\}$. Bag of Features is mainly done in two steps:

1. Calculate k clusters with k-means or any other algorithm [13] on U_{all}.
2. For all $\mathbf{v_i} \in V$ assign the vector to a cluster after distance comparison. Form Bag of Features representation as $R_V = \{V_{c_1}, V_{c_2}, ..., V_{c_k}\}$ where V_{c_i} is the number of vectors in V assigned to the cluster i.

3.2.1 K-Adjacent Segments

A technique of Bag of Features from adjacent line segment features is introduced in [13]. Here, a Canny edge detection followed by line fitting is done in order to calculate the location, orientation and length attributes

$$(r_{x_1}, r_{y_1}, r_{x_2}, r_{y_2}, ..., r_{x_k}, r_{y_k}),$$

$$(\theta_1, \theta_2, ..., \theta_k),$$

$$(l_1, l_2, ..., l_k),$$

respectively. Codebooks of these vectors are then extracted to build the final labeling engine. For each kind of attribute a different weight is taken into account so that a stable metric can be devised. Normalization is done with respect to the largest segment length. In our humble opinion, this is the most simple-but-elegant-and-effective route noted in this review.

3.2.2 SIFT

[12] is another Bag of Features kind modelling of writer style: Features in this case are SIFT descriptors rather than adjacent segments. Codebooks of descriptors are used to calculate the final vectors for classification routine. Several keypoint detectors are tested to get the best labelling. Originality of [13] is replaced in this case with the well-known scale invariant descriptor components.

3.2.3 Contour Moments

An uptodate example of Bag of Features kind can be seen in [18]. Here, engineers compose a method by Vector of Locally Aggregated Descriptors [VLAD] on Contour Moments. Descriptors based on region-to-orthogonal polynomial mappings are employed to generate a representation of the writer style. Results on two datasets are reported. Actually, VLAD is not an exact Bag of Features routine. It is constructed upon residuals [21] rather than cluster counts.

A robust scheme is established by combining VLAD with Moment Descriptors.

VLAD is a clean way for representing images and can be seen as an advanced or fine-tuned version of Bag of Features model. Here, assume again we have again U_{all} as defined in Bag of Features explanation 2. Let $C = \{\mathbf{c_1}, \mathbf{c_2}, ..., \mathbf{c_k}\}$ be the cluster centers. If a vector set $V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{|V|}\}$ extracted from an image, representation is written as $R_V = V_{c_1} \oplus V_{c_2} \oplus ... \oplus V_{c_k}$, where $V_{c_i} = \sum_{\mathbf{v_t}:l(\mathbf{v_t})=i} \mathbf{v_t} - \mathbf{c_i}$ and l is the cluster assignment function.

CHAPTER 4

INTRODUCED TECHNIQUE

Introduced algorithm is based on keypoint extraction followed by fixed-scale descriptor calculation and formation of covariance features.

4.1 Components

Given a binary image I, coarse keypoint-edge detection may yield an over-analyzed and over-generalized path, carrying the risk of a two-fold decrease in performance: i) Accuracy. ii) Complexity (of parameter control and of the algorithm). On the other hand, since the stated classification context is text-independent, dense sampling, where descriptors are extracted over the whole image by uniformly locating keypoints at pre-determined areas, is not feasible for the problem. Therefore, although it is proven to be a useful tool for object categorization, which was the topic of Chapter 2, grid sampling, is not suitable for writer identification task.

4.1.1 Keypoints

Following the line of reasoning stated above, one can focus on a quasi-dense strategy, where 1 of each M black pixel, is marked for descriptor extraction. Once the coordinates are located, local region vectors are calculated via SIFT [6] and BRIEF [11].

4.1.2 SIFT

In SIFT, the gradient information is gathered to describe a region. The circular area is considered as a 4×4 union of grids, where from each grid, an orientation histogram of 8 bins is extracted. A final vector of dimension $4 \times 4 \times 8 = 128$ is formed by concatenating the histogram data of each cell.

Taken from Hubble [1], in 4.1, the samples of astronomy images and corresponding SIFT desciptors are shown [2]. For demonstration reasons, number of bins is restricted to 16.

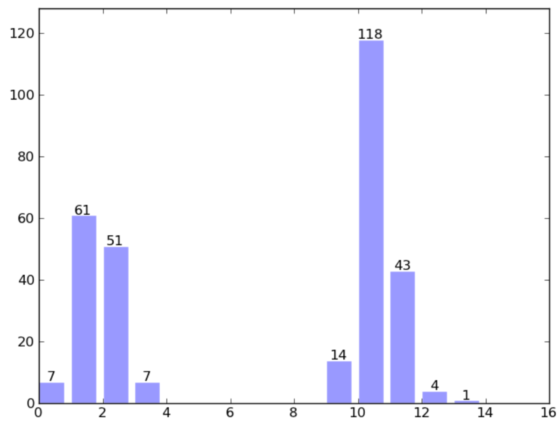
4.1.3 BRIEF

The second vector set source is BRIEF, where intensity tests of sampled point-pairs are encoded in a binary string manner to establish a memory-efficient and fast region descriptor.

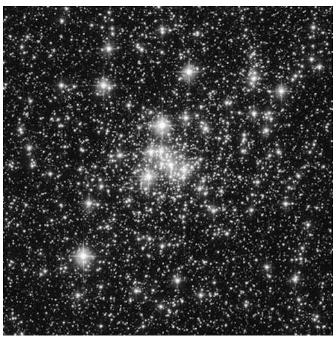
4.1.4 Covariance Features

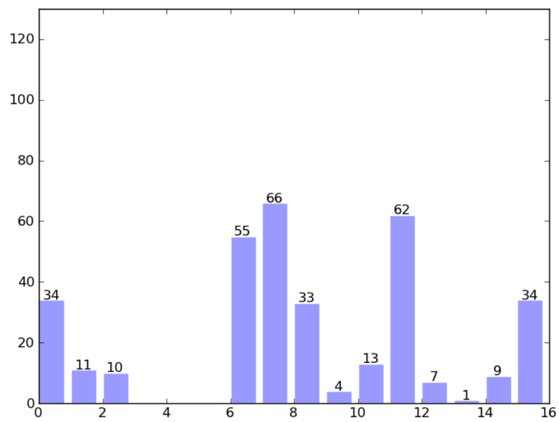
As in 2, covariance features are formed via the mapping (2.14). Let $\mathbf{U_s}$ be the SIFT vector set of a binary image I, $\mathbf{U_b}$ be the fitude for BRIEF. Assume that $\theta_s(\Sigma_{U_s})$, $\theta_b(\Sigma_{U_b})$ are the vectorizations of SIFT and BRIEF covariance matrices, similar to θ defined in 2.14. Then the covariance features are obtained by the concatenation of $\theta_s(\Sigma_{U_s})$ and $\theta_b(\Sigma_{U_b})$ = $\theta_s(\Sigma_{U_s}) \oplus \theta_b(\Sigma_{U_b})$.

[1] http://hubblesite.org/
[2] These are extracted via binaries from http://vlfeat.org
[3] Image is adapted from https://fr.wikipedia.org/wiki/Scale-invariant_feature_transform



(a) M 101





(b) NGC 290

Figure 4.1: Astronomy images and corresponding SIFT descriptors

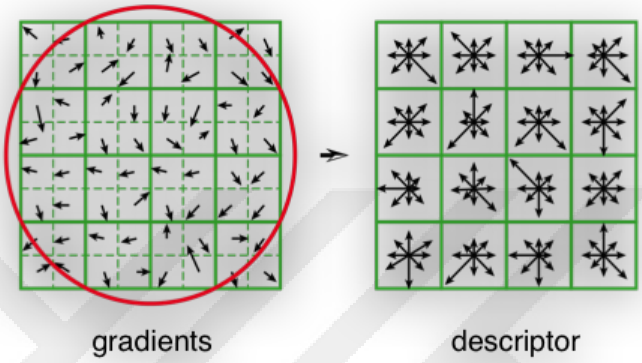


Figure 4.2: SIFT flow summary[3]

4.1.5 Identification

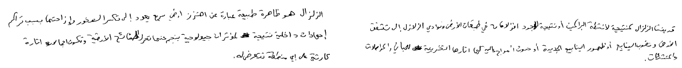
A 1-NN classifier [9] is built due to the low-cardinality of experiment text content set.

4.2 Results

In this section, competition structure and identification rates are noted.

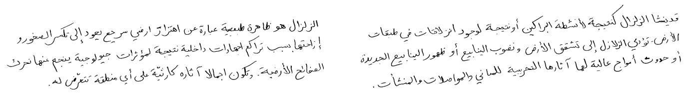
4.2.1 Competition Setup

ICFHR 2012 competition on writer identification challenge dataset [16] is selected to evaluate the introduced technique. This collection is a union of 206 writers' handwritten paragraphs. The same three textual content is written by each of the writers; the



(a) **(b)**

Figure 4.3: Writer 1



(a) **(b)**

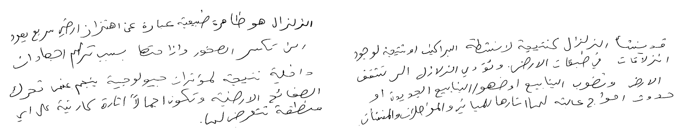
Figure 4.4: Writer 2

first two of these is used for training, the remaining ones are separated for testing.

In the training step, a two-fold cross validation scheme is followed to measure the success of the features. For each writer, a text is selected for training and the other for validation. This resulted with an average score of 0.95 identification rate.

4.2.2 Scores

An identification rate of 91.95% is achieved via covariance features. Our team **ihata** and its standing compared with other teams is given in table 4.1 [16].



(a) **(b)**

Figure 4.5: Writer 3

Table 4.1: Test results

Team	Identification Rate
Wayne Zhang	95.30%
Newell and Griffin	95.30%
YT	93.29%
ihata	91.95%
bfs	91.95%
AWReS	91.28%
cess_northumbria	91.28%
William Cukierski	91.28%
Marcos Sainz	89.93%
Sashi Dareddy	89.26%
D33B	87.92%
Han & Kilian	83.89%
Foxtrot	82.55%
Ben Hammer	81.88%
steinke	81.21%
Luciferase	77.18%

CHAPTER 5

CONCLUSION

To sum up, a simple technique based on empirical covariance estimation of local descriptors is tested successfully on writer identification, yielding a 4-th rank result in a competition with approximately 50 contestants.

5.1 Summary

Support Vector Machines is a technique of kernel engineering by which challenges of classification and regression are solved in an optimization framework. In this scheme, vector affinities with respect to high-dimensional vector space scalar products are calculated to categorize input features. Currently, the most efficient SVM implementation is SMO; relying on a fast analytical solution of sub-problems, it is the common choice for the final hypothesis formation.

In addition to solid quadratic optimization and statistical risk analysis grounds, while working through SVM, one can benefit from the modularity advantage via kernel mappings. It is not necessary for employing these functions to have fixed dimensional vectors as inputs at hand; graphs, strings and vector sets are also suitable structures. Influenced from [8] and [5], in the introduction and background of this study, an emphasis on vector sets can be observed.

Notwithstanding the explicit featurization kind through empirical covariance matrices, two main kernel machines for local descriptor set categorization are formulated: first of these is built upon direct covariance estimation combined with Power Series Kernels while the second is based on covariance matrix exponentiation. It has been shown that, both of the techniques can be regarded as alternatives to commonly known vector set kernels. Since matrix exponentiation is $O(D^3)$, its application is limited to low-dimensional cases, though.

After object categorization background, a derived version of the core notion is applied to the task of writer identification. Choosing the classic, well-known descriptor SIFT – which is robust to affine transformations – as the vector set generation way and supporting this with BRIEF, covariance feature identification is demonstrated to be an alternative system of automatic writer style modelling.

5.2 Future Work

During the evaluation of ICFHR 2012 Writer Identification Contest entries, there was the case of 'unknown writers' which corresponds to the task of writer verification. Our system, where any given input is necessarily assigned to a writting model in the training dataset, is lacking in detection of these writers. Thus, in future, a verification step shall be added to form a complete scheme.

Another extension can be accomplished by conducting experiments on datasets of different languages. This is also valid for object categorization in the aspect that the algorithm may be tested on more cluttered environment conditions or larger sample sets.

Moreover, Power Series Kernel-Linear Kernel combination can be adapted to writer identification simply by modifying the distance metric with respect to the final scalar product.

REFERENCES

[1] **Platt, J. C. (2009),** *"12 fast training of support vector machines using sequential minimal optimization"*, Advances in kernel methods: pp. 185-208.

[2] **Cristianini, N. and Shawe-Taylor J. (2000),** *"An introduction to support vector machines and other kernel-based learning methods"*, Cambridge University, Cambridge.

[3] **Friedman, J. H. (2001),** *"Greedy function approximation: a gradient boosting machine"*, Annals of statistics, pp. 1189-1232.

[4] **Sangjik, L., Cha, S.H. and Srihari, S. N. (2011),** *"Combining macro and micro features for writer identification"*, Electronic Imaging 2002, pp. 155-166.

[5] **Risi K., Jebara, T. (2003),** *"A kernel between sets of vectors."*, ICML, vol. 20, pp. 361.

[6] **Lowe, D. G (2004),** *"Distinctive image features from scale-invariant keypoints." International journal of computer vision"*, vol. 60 no. 2, pp. 91-110.

[7] **Ke, Y. and Sukthankar, R. (2004),** *"PCA-SIFT: A more distinctive representation for local image descriptors"*, in Computer Vision and Pattern Recognition, vol. 2, pp. II-506.

[8] **Grauman, K. and Darrell T. (2005),** *"The pyramid match kernel: Discriminative classification with sets of image features"*, in Tenth IEEE International Conference on Computer Vision, vol. 2, pp. 1452-1465.

[9] **Bishop, C. (2006),** *"Pattern Recognition and Machine Learning"*, Springer, New York.

[10] **Bulacu, M. and Schomaker, L. (2007),** *"Text-independent writer identification and verification using textural and allographic features."*, IEEE transactions on pattern analysis and machine intelligence, vol 29 no. 4, pp. 701-717.

[11] **Calonder, M., et al. (2010),** *"Brief: Binary robust independent elementary features."*, in Computer Vision, pp. 778-792.

[12] **Woodard, J., et al. (2010),** *"Writer recognition of Arabic text by generative local features."*, in Fourth IEEE International Conference on Biometrics: Theory Applications and Systems, pp. 1-7.

[13] **Jain, R. and Doermann, D. (2011),** *"Offline writer identification using k-adjacent segments."*, in IEEE International Conference on Document Analysis and Recognition, pp. 769-773.

[14] **Sreeraj, M. and Mary Idicula, S. (2011),** *"A survey on writer identification schemes."*, in International Journal of Computer Applications, vol. 26 no. 2, pp. 23-33.

[15] **Basar, F. (2012),** *Lineer Cebir*, Sürat, İstanbul.

[16] **Hassaine, A. and Al Maadeed, S. (2012),** *"ICFHR 2012 competition on writer identification challenge 2: Arabic scripts."*, IEEE International Conference on Frontiers in Handwriting Recognition, pp. 835-840.

[17] **Adams, R. and Essex C. (2013),** *Calculus: A Complete Course*, Pearson, Toronto.

[18] **Christlein, V., Bernecker D., Angelopoulou, E. (2015),** *"Writer identification using VLAD encoded contour-Zernike moments."*, in 13th IEEE International Conference on Document Analysis and Recognition, pp. 906-910.

[19] **Frejlichowski, D., et al. (2015)** *"Application of foreground object patterns analysis for event detection in an innovative video surveillance system."*, Pattern Analysis and Applications, vol. 18, no. 3, pp. 473-484.

[20] **Ruiz-Sarmiento, J. R., Galindo, C. and González-Jiménez, J. (2015)** *"Joint categorization of objects and rooms for mobile robots."*, in IEEE/RSJ International Conference onIntelligent Robots and Systems, pp. 2523-2528 .

[21] **VLAD description, VLFeat homepage,** http://www.vlfeat.org/api/vlad-fundamentals.html (Data Download Date: 10.22.2016)

[22] **ARMAR homepage,** http://monet.cs.columbia.edu/projects/armar/ (Data Download Date: 10.22.2016)

APPENDICES A

A. CURRICULUM VITAE

PERSONAL INFORMATION

Nationality: Turkish
Surname, Name: KARADENİZ, Talha
Date and Place of Birth: 21 August 1984, Adapazarı
Marital Status: Single
Phone: +905355087384
Email: talhakaradeniz@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
B.Sc.	METU, Mathematics	2012
High School	Gazi Anatolian High School	2002

WORK EXPERIENCE

Year	Place	Enrollment
2006-2011	Tubitak Uzay	Software Engineer
2012-2013	Simtek	Software Engineer
2013-2015	Freelance	Software Engineer
2015-2016	Çankaya University	Software Engineer

FOREIGN LANGUAGES

English

HOBBIES

Swimming, Writing