The Role of Riemannian Manifolds in Computer Vision: From Coding to Deep Metric Learning

Masoud Faraki

A thesis submitted for the degree of
Doctor of Philosophy of
The Australian National University

April 2018

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma at ANU or any other educational institution, except where due acknowledgment has been made.

I also declare that all sources used in this thesis have been fully and properly cited. Parts of this thesis have been published in

- A Comprehensive Look at Coding Techniques on Riemannian Manifolds, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, IEEE Transactions on Neural Networks and Learning Systems, 2018.
- Large Scale Metric Learning, A Voyage From Shallow to Deep, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, IEEE Transactions on Neural Networks and Learning Systems, 2017.
- No Fuss Metric Learning, a Hilbert Space Scenario, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, Pattern Recognition Letters, 98(C):83-89, 2017.
- Image Set Classification by Symmetric Positive Semi-Definite Matrices, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, IEEE Winter Conference on Applications of Computer Vision, Lake Placid, NY, March 7-9, 2016.
- More About VLAD: A Leap From Euclidean to Riemannian Manifolds, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA, June 7-12, 2015.
- Approximate Infinite-Dimensional Region Covariance Descriptors for Image Classification, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, 40th IEEE International Conference on Acoustics, Speech and Signal Processing, Brisbane, Australia, April 19-24, 2015.
- Material Classification on Symmetric Positive Definite Manifolds, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, IEEE Winter Conference on Applications of Computer Vision, Waikoloa Beach, HI, Jan 6-9, 2015.

- Fisher Tensors for Classifying Human Epithelial Cells, Masoud Faraki, Mehrtash Harandi, Arnold Wiliem, and Brian Lovell, Pattern Recognition, 47(7):2348-2359, 2014.

- Log-Euclidean Bag of Words for Human Action Recognition, Masoud Faraki, Maziar Palhang, and Conrad Sanderson, IET Computer Vision, 9(3):331-339, 2014.

- Bag of Riemannian Words for Virus Classification, Masoud Faraki and Mehrtash Harandi, CRC press Taylor and Francis group, 2014.

Masoud Faraki
11 April 2018

To my beloved family.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Mehrtash Harandi, for his kind support and continuous encouragement to make my Ph.D experience productive and stimulating. I appreciate all his contributions of time, ideas and patience during my candidature. I am also grateful to my other supervisory panel members, Prof. Fatih Porikli and Prof. Richard Hartley. I feel very proud to had the opportunity to work under their supervision and received constructive comments at different stages of my PhD.

I would like to thank NICTA/Data61 and the Australian National University (ANU) for financially supporting my PhD and providing a great scientific and friendly environment. Moreover, I would like to use this opportunity and thank my wonderful friends who have contributed immensely to my personal and professional time. My time at NICTA/Data61 and ANU was made enjoyable due to the many friends and groups that became an important part of my life.

Last but by no means least, I am sincerely grateful to my beloved family who have provided me through moral and emotional support in my life. This thesis would not be possible without their unconditional love and support and I would like to devote all my research achievements to them.

Masoud Faraki
11 April 2018

Abstract

A diverse number of tasks in computer vision and machine learning enjoy from representations of data that are compact yet discriminative, informative and robust to critical measurements. Two notable representations are offered by Region Covariance Descriptors (RCovD) and linear subspaces which are naturally analyzed through the manifold of Symmetric Positive Definite (SPD) matrices and the Grassmann manifold, respectively, two widely used types of Riemannian manifolds in computer vision.

As our first objective, we examine image and video-based recognition applications where the local descriptors have the aforementioned Riemannian structures, namely the SPD or linear subspace structure. Initially, we provide a solution to compute Riemannian version of the conventional Vector of Locally aggregated Descriptors (VLAD), using geodesic distance of the underlying manifold as the nearness measure. Next, by having a closer look at the resulting codes, we formulate a new concept which we name Local Difference Vectors (LDV). LDVs enable us to elegantly expand our Riemannian coding techniques to any arbitrary metric as well as provide intrinsic solutions to Riemannian sparse coding and its variants when local structured descriptors are considered.

We then turn our attention to two special types of covariance descriptors namely infinite-dimensional RCovDs and rank-deficient covariance matrices for which the underlying Riemannian structure, i.e. the manifold of SPD matrices is out of reach to great extent. To overcome this difficulty, we propose to approximate the infinite-dimensional RCovDs by making use of two feature mappings, namely random Fourier features and the Nyström method. As for the rank-deficient covariance matrices, unlike most existing approaches that employ inference tools by predefined regularizers, we derive positive definite kernels that can be decomposed into the kernels on the cone of SPD matrices and kernels on the Grassmann manifolds and show their effectiveness for image set classification task.

Furthermore, inspired by attractive properties of Riemannian optimization techniques, we extend the recently introduced Keep It Simple and Straightforward MEtric learning (KISSME) method to the scenarios where input data is non-linearly distributed. To this end, we make use of the infinite dimensional covariance matrices and propose techniques towards projecting on the positive cone in a Reproducing Kernel Hilbert Space (RKHS). We also address the sensitivity issue of the KISSME to the input dimensionality. The KISSME algorithm is greatly dependent on Principal Component Analysis (PCA) as a preprocessing step which can lead to difficulties, especially when the dimensionality is not meticulously set. To address this issue, based on the KISSME algorithm, we develop a Riemannian framework to jointly learn a mapping performing dimensionality reduction and a metric in the induced space. Lastly, in line with the recent trend in metric learning, we devise end-to-end learning of a generic deep network for metric learning using our derivation.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Making use of structured descriptors has been shown to be effective in a wide range of computer vision tasks. A notable example is the Diffusion Tensor Imaging (DTI) technique which represents each voxel in 3-D brain scans by a 3×3 Symmetric Positive Definite (SPD) matrix. It is now an accepted fact that analyzing the resulting diffusion tensors by vectorizing them deteriorates the performances heavily and can lead to solutions that are physically meaningless Alexander et al. [2007]. Another example of a structured descriptor is the Region Covariance Descriptor (RCovD) Tuzel et al. [2008], successfully used in human detection Tuzel et al. [2008], texture classification Faraki et al. [2015b] , human head pose estimation Tosato et al. [2013] and face recognition Wang et al. [2012]; Harandi et al. [2016]. RCovDs offer compact and rich visual content representations by fusing various features while reducing the impact of noisy samples Tuzel et al. [2008]; Faraki et al. [2015b]. Similarly, linear subspaces as structured descriptors offer a convenient platform to compensate for a wide range of image variations and have been used with promising results in image set and video classification Faraki et al. [2016]; Harandi et al. [2015a].

Despite their intriguing properties, analyzing the aforementioned structured descriptors is not straightforward as a result of their non-Euclidean geometry. More specifically, diffusion tensors and RCovDs belong to the manifold of SPD matrices Pennec et al. [2006] and linear subspaces are points on the Grassmann manifold Edelman et al. [1998]. Although the two manifolds are Riemannian (i.e., equipped with metrics), the lack of a vector space structure is a barrier for developing inference methods Arsigny et al. [2007]; Pennec et al. [2006]; Tuzel et al. [2008]. The difficulty will increase when special types of the descriptors such as infinite-dimensional RCovDs Harandi et al. [2014a] or rank-deficient covariance descriptors Wang et al. [2012] are considered. This has been demonstrated by many previous works Arsigny et al. [2007]; Pennec et al. [2006]; Tuzel et al. [2008].

On a related note, at the heart of many Mahalanobis metric learning algorithms lie notions of Riemannian geometry and optimization on a curved Riemannian manifold to find a metric $\boldsymbol{M}$ and/or a projection $\boldsymbol{W}$ in the presence of constraints Weinberger and Saul [2009]; Mignon and Jurie [2012]; Harandi et al. [2017]. In better words, an exact formulation of the objective functions of such algorithms is obtained through viewing $\boldsymbol{M}$ and $\boldsymbol{W}$ as a point on a Riemannian manifold. For example, an ideal metric matrix is an instance of a point on the manifold of SPD matrices and hence a proper optimization technique on this manifold promises a valid solution. As another example, the Stiefel manifold provides a natural way to handle orthogonality constraints on $\boldsymbol{W}$ which is of great interest in many metric learning methods.

It is worth mentioning that we endeavor to present universal techniques which with subtle modifications, are applicable to wide range of applications. Therefore, throughout the thesis we will utilize various types of datasets and applications to evaluate our proposals.

1.1 Contributions

1.1.1 Riemannian Coding

In our first contribution, we extend well-known aggregation/coding techniques (such as sparse coding Wright et al. [2009]) in curved and non-Euclidean spaces, i.e., Riemannian manifolds. Unlike many existing non-vectorial coding approaches, we do not base our algorithms on the restrictive assumption that a holistic representation of images or videos is at hand. In particular, we consider structured local descriptors from visual data, namely RCovDs and linear subspaces that lie on the manifold of SPD matrices and the Grassmannian manifolds, respectively. We provide a comprehensive mathematical framework that facilitates the aggregation problem of such manifold data into an elegant solution.

To this end, we start by the simplest form of coding, namely bag of words. Then, inspired by the success of Vector of Locally Aggregated Descriptors (VLAD) Jégou et al. [2012] in addressing computer vision problems, we will introduce its Riemannian extensions. Finally, we study Riemannian form of sparse coding, locality-constrained linear coding Wang et al. [2010] and collaborative coding Zhang et al. [2011].

1.1.2 Infinite-dimensional RCovDs

It has been shown in some studies that infinite-dimensional RCovDs offer better discriminatory power over their low-dimensional versions Harandi et al. [2014a]; Quang et al. [2014]. However, the underlying Riemannian structure, i.e., the manifold of SPD matrices, is out of reach to great extent for the infinite-dimensional RCovDs and one is confined to perform implicit analysis. To overcome this difficulty, we propose methods to approximate infinite-dimensional RCovDs by exploiting two feature mappings, namely random Fourier features Rahimi and Recht [2007] and the Nyström method Baker [1977]. By approximating the infinite-dimensional RCovDs with finite-dimensional ones, one could seamlessly exploit the rich geometry of RCovDs and tools developed upon that to do the inference. We will empirically show that the proposed finite-dimensional approximations of infinite-dimensional RCovDs consistently outperform the low-dimensional RCovDs for image classification task, while enjoying the Riemannian structure of the SPD manifolds.

1.1.3 Symmetric Positive Semi-Definite Matrices for Image Set Classification

Although representing visual contents by RCovDs and leveraging the inherent manifold structure lead to enhanced performances in various visual recognition tasks, the resulting RCovD is often rank-deficient when image set classification is deemed. Thus, most existing approaches adhere to blind perturbation with predefined regularizers just to be able to employ inference tools Wang et al. [2012]; Faraki et al. [2014b]. To overcome this problem, we introduce novel similarity measures specifically designed for rank-deficient RCovDs, or in other words, Symmetric Positive Semi-Definite (SPSD) matrices. In particular, we derive positive definite kernels that can be decomposed into the kernels on the cone of SPD matrices and kernels on the Grassmann manifolds.

1.1.4 Metric Learning

We add an extension to the recently introduced Keep It Simple and Straightforward MEtric learning (KISSME) Koestinger et al. [2012] method by devising a kernel version of the algorithm, hence making it applicable in scenarios where input data is not linearly distributed. With the aid of infinite dimensional covariance matrices, we propose two techniques towards projecting on the positive cone in a Reproducing Kernel Hilbert Space (RKHS). The first method, enjoys a closed-form formulation and is more suitable when computational load is important. The second solution is more accurate and requires Riemannian optimization techniques. Our experiments evidence that, compared to the state-of-the-art metric learning algorithms, working directly in reproducing kernel Hilbert space, leads to more robust and better performances.

Furthermore, we address the sensitivity issue of the KISSME to the input dimensionality.

To this end, based on the KISSME algorithm, we develop a Riemannian framework to jointly learn a mapping performing dimensionality reduction and a metric in the induced space. In line with the recent metric learning methods, we also devise end-to-end learning of a generic deep network for metric learning using our derivation.

1.2 Thesis Outline

The remaining parts of this thesis are organized into five chapters as follows. Chapter 2 introduces preliminary concepts which are of essential interest in later chapters. The notation used throughout the thesis is defined in the chapter as well. In Chapter 3, we extend state-of-the-art coding/aggregation methods onto an extensive space of curved Riemannian manifolds by providing a comprehensive mathematical framework that formulates the coding/aggregation problem into an elegant solution. In chapter 4, we consider the two special types of covariance descriptors, namely infinite-dimensional RCovDs and SPSD matrices and propose our methods to estimate the infinite-dimensional RCovDs and analyze SPSD matrices by novel similarity measures. We address some limitations of the KISSME method in Chapter 5, where Riemannian optimization techniques become useful. We first propose a kernel version of the KISSME, hence providing a solution to employ the algorithm on non-vectorized data (e.g., manifold-value data). We then suggest a dimensionality reduction technique along learning the metric. We conclude the chapter by devising end-to-end learning of a generic deep network for metric learning using our derivation. Finally, Chapter 6 concludes this thesis by a summary. Fig. 1.1 illustrates the connection between the main chapters in one diagram.

1.3 Publications

The contributions described in this thesis have previously appeared in the following publications.

- A Comprehensive Look at Coding Techniques on Riemannian Manifolds, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, IEEE Transactions on Neural Networks and Learning Systems, 2018.
- Large Scale Metric Learning, A Voyage From Shallow to Deep, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, IEEE Transactions on Neural Networks and Learning Systems, 2017.

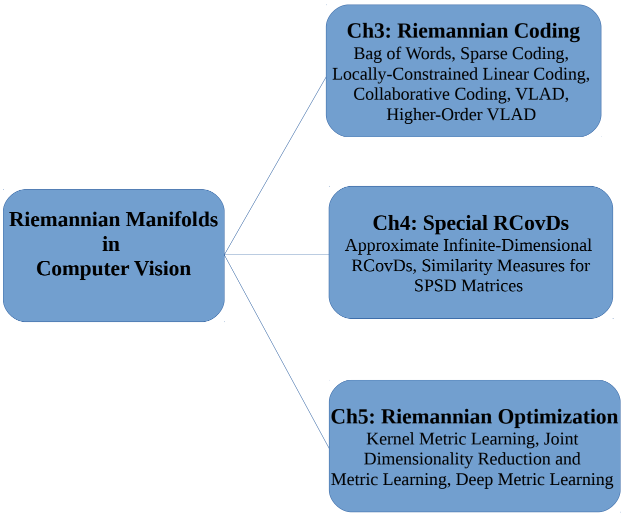
Ch3: Riemannian Coding
Bag of Words, Sparse Coding, Locally-Constrained Linear Coding, Collaborative Coding, VLAD, Higher-Order VLAD
Riemannian Manifolds in Computer Vision
Ch4: Special RCovDs
Approximate Infinite-Dimensional RCovDs, Similarity Measures for SPSD Matrices
Ch5: Riemannian Optimization
Kernel Metric Learning, Joint Dimensionality Reduction and Metric Learning, Deep Metric Learning

Figure 1.1: Depicted summary of the main contributions of this thesis.

- No Fuss Metric Learning, a Hilbert Space Scenario, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, Pattern Recognition Letters, 98(C):83-89, 2017.

- Image Set Classification by Symmetric Positive Semi-Definite Matrices, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, IEEE Winter Conference on Applications of Computer Vision, Lake Placid, NY, March 7-9, 2016.

- More About VLAD: A Leap From Euclidean to Riemannian Manifolds, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA, June 7-12, 2015.

- Approximate Infinite-Dimensional Region Covariance Descriptors for Image Classification, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, 40th IEEE International Conference on Acoustics, Speech and Signal Processing, Brisbane, Australia, April 19-24, 2015.

- Material Classification on Symmetric Positive Definite Manifolds, Masoud Faraki, Mehrtash Harandi, and Fatih Porikli, IEEE Winter Conference on Applications of Computer Vision, Waikoloa Beach, HI, Jan 6-9, 2015.

- Fisher Tensors for Classifying Human Epithelial Cells, Masoud Faraki, Mehrtash Harandi, Arnold Wiliem, and Brian Lovell, Pattern Recognition, 47(7):2348-2359, 2014.

- Log-Euclidean Bag of Words for Human Action Recognition, Masoud Faraki, Maziar Palhang, and Conrad Sanderson, IET Computer Vision, 9(3):331-339, 2014.
- Bag of Riemannian Words for Virus Classification, Masoud Faraki and Mehrtash Harandi, CRC press Taylor and Francis group, 2014.

Chapter 2

Background

In this part, we introduce some preliminary concepts such as Riemannian geometry which are of essential in our developments. Throughout the thesis, we use bold lower-case letters (e.g., $\boldsymbol{x}$) to show column vectors and bold upper-case letters (e.g., $\boldsymbol{X}$) to show matrices. $[\cdot]_i$ is used to denote the i-th element of a vector. $\mathbf{1}_n$ and $\mathbf{I}_n$ show vector of ones in $\mathbb{R}^n$ and the $n \times n$ identity matrix, respectively. ℓ_1 and ℓ_2 norms of a vector are denoted by $\|\boldsymbol{x}\|_1 = \sum_i |[\boldsymbol{x}]_i|$ and $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$, respectively. The Frobenius norm of a matrix is shown by $\|\boldsymbol{X}\|_F = \sqrt{\operatorname{Tr}(\boldsymbol{X}^T\boldsymbol{X})}$, with $\operatorname{Tr}(\cdot)$ indicating the matrix trace. The determinant of a matrix is shown by $\det(\boldsymbol{X})$. $\log(\boldsymbol{X})$ is the principal logarithm of matrix $\boldsymbol{X}$. $\det(\cdot)$ shows the matrix determinant. Finally, $[\cdot]_+$ indicates the hinge loss function, i.e., $max(0, \cdot)$.

Let $\mathcal{X}$ be a nonempty set. Then,

Definition 1. *A pair $(\mathcal{X}, g)$ identifies a metric space when $g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ is a global distance function such that $\forall x, y, z \in \mathcal{X}$ the following properties hold*

- $g(x, y) \geq 0$*, i.e., non-negativity*
- $g(x, y) = g(y, x)$*, i.e., symmetry*
- $g(x, y) \leq g(x, z) + g(z, y)$*, i.e., triangle inequality*
- $g(x, y) = 0$ *iff* $x = y$*, i.e., distinguishability*

Definition 2 (Real-valued Positive Definite Kernels)**.** *A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite (pd) kernel on $\mathcal{X}$ if and only if $\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) > 0$ for any $n \in \mathbb{N}$, $x_i \in \mathcal{X}$ and non-zero vector $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)^T \in \mathbb{R}^n$.*

According to Mercer's theorem, for any pd kernel $k(\cdot, \cdot)$, there exists a mapping to a Reproducing Kernel Hilbert Space (RKHS), $\phi : \mathcal{X} \to \mathcal{H}$, such that: $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}, k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle_{\mathcal{H}}$.

2.0.1 Riemannian Geometry

A *manifold* $\mathcal{M}$ is a Hausdorff topological space which locally resembles a Euclidean space $\mathbb{R}^m$. The focus of this work is analytic manifolds (see Subbarao and Meer [2009] for definition). Let f and g be two arbitrary continuous functions acting on a manifold $\mathcal{M}$ and c_1 and c_2 be two scalars. Then a *tangent* Δ at a point on the manifold is a real-valued operator on continuous functions satisfying following properties

$$\begin{aligned} \Delta(c_1 f + c_2 g) &= c_1 \Delta(f) + c_2 \Delta(g), \\ \Delta(f g) &= f \Delta(g) + g \Delta(f) \end{aligned}$$

Intuitively, Δ represents a direction (vector) in which the value assigned to $\Delta(f)$ can be thought as the derivative of f in that direction.

The *tangent space* attached to a point $\boldsymbol{P} \in \mathcal{M}$, $T_{\boldsymbol{P}}\mathcal{M}$, is a vector space that consists of the tangent vectors of all possible curves on the manifold passing through $\boldsymbol{P}$ Pennec et al. [2006]. A *Riemannian manifold* is a differential manifold with a metric defined on the tangent spaces. The structure of a Riemannian manifold is specified by the metric. A *Riemannian metric* is a continuous collection of dot products on the tangent space at each point of the manifold. It is usually chosen to provide robustness to some geometrical transformations. Furthermore, it enables one to define lengths and angles on the manifold.

Smooth curves connect points on a Riemannian manifold. Having the Riemannian metric at the disposal, one can compute instantaneous speed (direction and magnitude) and length of a given curve. The curves yielding the minimum distance for any two points of the manifold are called *geodesics* and their length is the *geodesic distance*.

On a Riemannian manifold $\mathcal{M}$, let $\overrightarrow{\boldsymbol{pq}} \in T_{\boldsymbol{P}}\mathcal{M}$ be a tangent vector. For geodesically complete manifolds (the case in our work), there exists a unique geodesic starting at $\boldsymbol{P}$ associated with this tangent vector and hence $\overrightarrow{\boldsymbol{pq}}$ can identify a point $\boldsymbol{Q} \in \mathcal{M}$. The *exponential map* $\exp_{\boldsymbol{P}}(\cdot) : T_{\boldsymbol{P}}\mathcal{M} \to \mathcal{M}$, guarantees that the length of the tangent vector is equal to the geodesic distance. The logarithm map $\log_{\boldsymbol{P}}(\cdot) = \exp_{\boldsymbol{P}}^{-1}(\cdot) : \mathcal{M} \to T_{\boldsymbol{P}}\mathcal{M}$, is the inverse of the exponential map and maps a point on the manifold to the tangent space $T_{\boldsymbol{P}}\mathcal{M}$, i.e., $\overrightarrow{\boldsymbol{pq}} = \log_{\boldsymbol{P}}(\boldsymbol{Q})$. We note that, the exponential and logarithm maps vary as the point $\boldsymbol{P}$ moves along the manifold.

2.0.2 The Manifold of Symmetric Positive Definite Matrices

A real $d \times d$ matrix $\boldsymbol{C}$ is Symmetric Positive Definite (SPD) if and only if $\boldsymbol{z}^T\boldsymbol{C}\boldsymbol{z} > 0$ for every non-zero vector $\boldsymbol{z} \in \mathbb{R}^d$. The space of real $d \times d$ SPD matrices, $\mathcal{S}_{++}^d$, forms a Lie group which has a manifold structure. This allows one to use the language of Riemannian manifolds, e.g., geodesics and all the relevant concepts of differential geometry when discussing $\mathcal{S}_{++}^d$. The tangent space at a point $\boldsymbol{X} \in \mathcal{S}_{++}^d$ is the set of all $d \times d$ symmetric matrices. Formally,

$$T_{\boldsymbol{X}}\mathcal{S}_{++}^d \triangleq \{\boldsymbol{\Delta} \in \mathbb{R}^{d\times d} : \boldsymbol{\Delta} = \boldsymbol{\Delta}^T\} \ . \tag{2.1}$$

Region Covariance Descriptors (RCovD) are SPD matrices and therefore it is essential to utilize Riemannian geometry to analyze them. Formally, a $d \times d$ RCovD can be constructed from a set of r observations $\mathbb{O} = \{\boldsymbol{o}_i\}_{i=1}^r,\ \boldsymbol{o}_i \in \mathbb{R}^d$, extracted from a region in an image (or a block in a video) as follows

$$\boldsymbol{C}_I = \frac{1}{r-1}\sum_{i=1}^{r}\left(\boldsymbol{o}_i - \overline{\boldsymbol{o}}\right)\left(\boldsymbol{o}_i - \overline{\boldsymbol{o}}\right)^T \ , \tag{2.2}$$

where $\overline{\boldsymbol{o}} = \frac{1}{r}\sum_{i=1}^r \boldsymbol{o}_i$.

$\mathcal{S}_{++}^d$ is mostly studied with the Riemannian structure induced by the Affine Invariant Riemannian Metric (AIRM) Pennec et al. [2006].

Definition 3. *The geodesic distance $\delta_G : \mathcal{S}_{++}^d \times \mathcal{S}_{++}^d \to \mathbb{R}^+$ induced by the AIRM is defined as*

$$\delta_G(\boldsymbol{X},\boldsymbol{Y}) \triangleq \|\log(\boldsymbol{X}^{-1/2}\boldsymbol{Y}\boldsymbol{X}^{-1/2})\|_F \ . \tag{2.3}$$

Beside the AIRM, two types of symmetric Bregman divergences, namely the Stein Sra [2012] and the Jeffrey Wang and Vemuri [2004] divergences are widely used to measure similarities on SPD manifolds.

Definition 4. *The Stein metric $\delta_S : \mathcal{S}_{++}^d \times \mathcal{S}_{++}^d \to \mathbb{R}^+$ is a symmetric type of Bregman divergence and is defined as*

$$\delta_S^2(\boldsymbol{X},\boldsymbol{Y}) \triangleq \ln\det\left(\frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right) - \frac{1}{2}\ln\det(\boldsymbol{X}\boldsymbol{Y}) \ . \tag{2.4}$$

Definition 5. *The Jeffrey divergence (also known as J or symmetric KL divergence) $\delta_J : \mathcal{S}_{++}^d \times$*

$\mathcal{S}_{++}^{d} \to \mathbb{R}^{+}$ is also a symmetric type of Bregman divergence and is defined as

$$\delta_J^2(\boldsymbol{X},\boldsymbol{Y}) \triangleq \frac{1}{2}\operatorname{Tr}(\boldsymbol{X}^{-1}\boldsymbol{Y}) + \frac{1}{2}\operatorname{Tr}(\boldsymbol{Y}^{-1}\boldsymbol{X}) - d\ . \tag{2.5}$$

2.0.3 The Grassmann Manifold

To have a better understanding of the Grassmann manifold, we first define Stiefel manifold. The set of $d \times p$, $0 < p < d$, matrices with orthonormal columns is a Riemannian manifold known as Stiefel manifold $S(p,d)$. More formally,

$$S(p,d) \triangleq \{\boldsymbol{X} \in \mathbb{R}^{d\times p} : \boldsymbol{X}^T\boldsymbol{X} = \mathbf{I}_p\}\ . \tag{2.6}$$

A point on the Grassmann manifold $\mathcal{G}_d^p$ is a subspace spanned by the columns of a $d \times p$ full rank matrix Edelman et al. [1998]. In other words, points on $\mathcal{G}_d^p$ are equivalence classes of $d \times p$ matrices with orthonormal columns where two matrices are equivalent if their columns span the same p-dimensional subspace. The tangent space at a point $\boldsymbol{X} \in \mathcal{G}_d^p$ admits

$$T_X\mathcal{G}_d^p \triangleq \{\boldsymbol{\Delta} \in \mathbb{R}^{d\times p} : \boldsymbol{X}^T\boldsymbol{\Delta} + \boldsymbol{\Delta}^T\boldsymbol{X} = 0\}\ . \tag{2.7}$$

The geodesic distance between two subspaces (points) is defined as the magnitude of the smallest rotation that takes one point to the other.

Definition 6. *For the Grassmannian, the geodesic distance between two points $\boldsymbol{X}$ and $\boldsymbol{Y}$ is given by*

$$\delta_G(\boldsymbol{X},\boldsymbol{Y}) \triangleq \|\Theta\|\ , \tag{2.8}$$

where $\Theta = [\theta_1, \theta_2, \cdots, \theta_p]$ is the vector of principal angles between $\boldsymbol{X}$ and $\boldsymbol{Y}$ Edelman et al. [1998].

In addition to the geodesic distance, a popular metric on $\mathcal{G}_d^p$ is the projection metric.

Definition 7. *The projection distance, $\delta_P : \mathcal{G}_d^p \times \mathcal{G}_d^p \to \mathbb{R}^{+}$, between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined as Hamm and Lee [2008]*

$$\delta_P^2(\boldsymbol{X},\boldsymbol{Y}) \triangleq \|\boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{Y}\boldsymbol{Y}^T\|_F^2\ . \tag{2.9}$$

Chapter 3

Riemannian Coding

3.1 Overview

An underlying assumption in traditional coding schemes (e.g., sparse coding) is that the data geometrically comply with the Euclidean space. In other words, the data is presented to the algorithm in vector form and Euclidean axioms are fulfilled. This is of course restrictive in machine learning, computer vision and signal processing as shown by a large number of recent studies. Our proposal takes a further step and provides a comprehensive mathematical framework to perform coding in curved and non-Euclidean spaces, i.e., Riemannian manifolds.

To this end, we start by the simplest form of coding, namely bag of words. Then, inspired by the success of Vector of Locally Aggregated Descriptors (VLAD) Jégou et al. [2012] in addressing computer vision problems, we will introduce its Riemannian extensions. Finally, we study Riemannian form of Sparse Coding (SC) Wright et al. [2009], locality-constrained linear coding (LLC) Wang et al. [2010] and Collaborative Coding (CC) Zhang et al. [2011]. Through rigorous tests, we demonstrate the superior performance of our Riemannian coding schemes against state-of-the-art methods on several visual classification tasks including head pose classification, video-based face recognition and dynamic scene recognition Faraki et al. [2018].

3.2 Introduction

In this chapter, we devise a frame-work to exploit state-of-the-art coding methods such as VLAD and SC where the local descriptors belong to a Riemannian manifold. Classical coding/aggregating techniques Jégou et al. [2012]; Sivic and Zisserman [2003]; Lazebnik et al. [2006] are designed to work only with vectors (i.e., local descriptors are points in $\mathbb{R}^n$). Lately, a few studies target the problem of coding/aggregation when the local descriptors are structured (e.g., subspaces) and non-vectorial Harandi et al. [2015a]. Inspired by the fact that describing images or videos by local descriptors is the method of choice Lazebnik et al. [2006]; Jégou et al. [2012]; Perronnin and Dance [2007] these days, we add a novel dimension to the applicability of such techniques by introducing a mathematical foundation for coding/aggregation of structured descriptors.

To put the discussion into perspective, describing images or videos by local descriptors is preferable to holistic representations when, for instance, the recognition problem pertains large intra-class variations, articulated shapes, self-occlusions, and changing backgrounds, to name a few. On a related note, structured representations such as Region Covariance Descriptors (RCovD) and linear subspaces have been shown to provide robust and efficient representations for a wide range of tasks Tuzel et al. [2008]; Faraki et al. [2014a]; Harandi et al. [2015a]. However, RCovDs and linear subspaces lie on connected Riemannian manifolds, the manifold of Symmetric Positive Definite (SPD) matrices and the Grassmann manifolds, respectively. Consequently, Euclidean geometry is not appropriate to analyze them as shown in several recent studies Tuzel et al. [2008]; Pennec et al. [2006]; Jayasumana et al. [2013]; Harandi et al. [2015a].

Here, we examine image and video-based recognition applications where the local descriptors have the aforementioned Riemannian structures, namely the SPD or linear subspace structure. To be precise, we provide answers to the two following questions

- can we encode the local structured descriptors into a fixed length and discriminative vector?
- can we derive a universal mathematical framework that helps us formulate the encoding problem into an elegant solution?

To this end, we begin by providing a solution to compute Riemannian version of the conventional VLAD, R-VLAD, using the geodesic distance of the underlying manifold as the nearness measure. Then, we clarify that the resulting codes are actually obtained from a new concept which we name Local Difference Vectors (LDV). Furthermore, analogues to the Higher-Order (HO-) VLAD Peng et al. [2014], we also leverage higher order statistics of local structured descriptors for R-VLAD codes and make them more discriminative. Lastly, with the aid of the LDVs, we expand our Riemannian coding techniques and provide intrinsic solutions to Riemannian SC (R-SC) and two of its variants, namely Riemannian version of the LLC (R-LLC) and the CC (R-CC).

With LDVs, we show that coding/aggregation with other metrics/closeness measures rather than geodesic distances is also possible. In other words, we do not confine ourselves to the geodesic distance case and develop the sister family of our methods by exploiting various well-known forms of similarity measures (e.g., divergences) defined on the underlying manifolds. Our motivation is the fact that one can seamlessly use our general formulation with a metric suitable for a specific task at hand. For example, one may choose a divergence over the geodesic distance if computing geodesics is demanding. In particular, we make use of the Stein Sra [2012] and Jeffrey Wang and Vemuri [2004] divergences on the manifold of SPD matrices and the projection distance Hamm and Lee [2008] on Grassmann manifolds to obtain new variants of our solution. Last but not least, our contributions enable one to aggregate local descriptors on curved spaces. Therefore, we show that conventional forms of coding/aggregation are indeed special cases of our universal scheme if the manifold is chosen to be the Euclidean space.

Our experiments demonstrate the superiority of the proposed approach against several state-of-the-art methods such as the Weighted ARray of COvariances (WARCO) of Tosato et al. Tosato et al. [2013] for head pose classification and the Deep Reconstruction Model (DRM) of Hayat et al. Hayat et al. [2015] for video-based face classification. To the best of our knowledge, using the standard protocol, our proposed methods achieve top results on standard benchmarks: 85.3% for HOCoffe Tosato et al. [2013], 92.9% for QMUL Tosato et al. [2013], 97.8% for Dyntex++ Ghanem and Ahuja [2010], 93.1% for Maryland Shroff et al. [2010] and 79.9% for YouTube Celebrities Kim et al. [2008].

3.3 Related Work

In this section, we review some relevant encoding methods to our proposals, such as Bag of Words (BoW), VLAD and SC. The reason behind the great surge of interest in these local models is twofold. Firstly, they can benefit from powerful local feature descriptors such as SIFT Lowe [2004] which (to some extent) provide robustness to image transformations such as scaling, translation and occlusion. Secondly, the output vector can be compared using the conventional Euclidean distance norms and utilized in powerful classifiers (e.g. Support Vector Machines (SVM)).

3.3.1 Bag of Words

While celebrating their third decade of birth, BoW Sivic and Zisserman [2003]and its extensions Lazebnik et al. [2006] continue to be the baseline image and video representations. Several variations have been proposed to improve the discriminatory power of the original

BoW model. Notable examples include Video Google Sivic and Zisserman [2003] in which the resulting vector components are weighted by inverse document frequency terms and spatio-temporal pyramid matching Lazebnik et al. [2006] which considers the information about the spatial layout of features in the final image representation. Another important variant is the work of Gemert et al. Van Gemert et al. [2010] which addresses the unsteady hard assignments in histogram generation process of the original BoW and generates a descriptor using multiple visual words in a soft assignment manner.

In a very compatible scenario, Super Vector Coding (SVC) Zhou et al. [2010] method generates a (non-linear) code by linearly approximating a sufficiently smooth function defined on a high dimensional space. The resulting code can be understood as a super vector aggregating zero and first order statistics of local descriptors. SVC may achieve a lower functional approximation error compared to the original BoW method.

Recently, a Riemannian version of BoW is proposed on the space of SPD matrices using the geodesic distance along with a simpler yet effective version, referred to as Log-Euclidean method Faraki and Harandi [2014]; Faraki et al. [2014a]. Moreover, a simpler yet effective version, referred to as Log-Euclidean BoW, is devised for action recognition Faraki et al. [2014b]. We will utilize these two methods as baselines in our experiments.

3.3.2 Vector of Locally Aggregated Descriptors

The VLAD descriptor, one of the main elements in this work, can be understood as a simpler version of the earlier Fisher Vectors (FV) derived from Fisher kernel Perronnin and Dance [2007]; Jaakkola and Haussler [1999]. Assuming that an incoming variable-sized set of descriptors follows a parametric generative model, FV is able to provide fixed-length codes by taking the gradients of the samples' likelihood with respect to the parameters of the distribution, weighted by the inverse square root of the Fisher information matrix. It has been shown that VLAD inherits the useful properties of FV by providing compact codes with relaxed assumptions on the origin of the samples and the scale of the output vector components (to be uniform) Jégou et al. [2012]. It is also worth noting that VLAD subtly departs from conventional BoW as it encodes the differences from the cluster centers rather than simply counting the number of assignments to them.

Peng et al. [2014] address the problem of enriching VLAD codes by higher-order statistics (called HVLAD) and supervised codebook learning (called SVLAD). The complimentary information in their HVLAD descriptor are second and third-order statistics which are obtained from covariance matrix and skewness measure of the points in each cluster. VLAD accuracy scores are further boosted by discriminatively learning the codebook in SVLAD.

Very recently, effective use of the aggregation with deep features has been proved to be beneficial on various benchmarks. Gong et al. [2014] propose to first extract deep Convolutional Neural Network (CNN) activations for local patches at multiple scales and then exploit VLAD coding vectors to pool the activations. As another example, Cimpoi et al. [2015] propose a descriptor obtained by pooling CNN features which substantially improves the state-of-the-art in texture/material classification and scene recognition tasks. Some other recent advances include kernel VLAD Harandi et al. [2015c], better normalization schemes for VLAD Arandjelovic and Zisserman [2013] and VLAD for action recognition Jain et al. [2013].

3.3.3 Sparse Coding

Encoding a vector as linear combination of a few elements of an over-complete codebook is recognized as SC and has led to notable performances in various computer vision tasks Wright et al. [2009]; Elad and Aharon [2006]. Another alternative to extend SC on non-linear spaces is through recasting the problem into Reproducing Kernel Hilbert Spaces (RKHS) via the kernel trick Harandi et al. [2016, 2015a]. This results in a convex quadratic problem which can be solved conveniently. Another advantage of this method is that one could benefit from SC while having more separable samples in the resulting higher dimensional RKHS. Nevertheless, one is always obliged to find a valid kernel to be able to work on the manifold. For instance, in Harandi et al. [2016] authors are limited to use Bregman divergence based Gaussian kernels on SPD manifolds since the popular widely used Affine Invariant Riemannian Metric (AIRM) Pennec et al. [2006] does not yield a valid positive definite Gaussian kernel.

In Xie et al. [2013], Xie et al. formulate the problem of sparse coding and dictionary learning on SPD manifolds using the Riemannian geodesic distances. To this end, they propose a coordinate-independent approach to reconstruct a given sample using affine linear combination of a small number of dictionary atoms. The optimization problem over sparse codes and dictionary atoms is solved by alternating between them. Further, the lack of global vector space structure is partially compensated by the local tangent space at each query point. We will elaborate on this method more in § 3.5.3.

3.4 Conventional Coding Methods

In this part, we review some standard coding methods related to our contributions. Let us assume that a set of local descriptors $\mathcal{X} = \{x_t\}_{t=1}^{m}, x_t \in \mathbb{R}^d$ (e.g., the dense SIFT features Lowe

[2004]) extracted from an image or video and a codebook $\mathcal{D}$ with atoms $\{\boldsymbol{d}_i\}_{i=1}^{k}$, $\boldsymbol{d}_i \in \mathbb{R}^d$ (e.g., obtained by the standard k-means algorithm) are at our disposal. Coding algorithms represent each query point $\boldsymbol{x}$ as some function of codebook atoms $\boldsymbol{d}_i$. Furthermore, some additional constraints might be added to objective functions to impose useful structure on the codes and subsequently obtain a more discriminative representation.

3.4.1 Vector of Locally Aggregated Descriptors

To review the VLAD method, we begin by studying its function in Euclidean spaces through its predecessor, i.e., the FV. FV encodes the set $\mathcal{X}$ into a high-dimensional vector representation by fitting a parametric generative model in the form of a Gaussian Mixture Model (GMM) with k components to the local descriptors, i.e.,

$$p(\boldsymbol{x}_t|\lambda) = \sum_{i=1}^{k} \omega_i \mathcal{N}(\boldsymbol{x}_t|\,\boldsymbol{\mu}_i, \Sigma_i)\ ,$$

where $\lambda = \{\omega_i, \boldsymbol{\mu}_i, \Sigma_i\}$ are the mixture weight, mean and covariance of the Gaussian components, respectively.

The FV descriptor is obtained by computing the gradients of the log-likelihood of the model with respect to its parameters (also known as the *score functions* in statistics). It leads to a representation that captures the contribution of the individual parameters to the generative process. Related to VLAD, is the first order differences between members of the set $\mathcal{X}$ and each of the GMM centers which has the following form

$$\nabla_{\mu_i} \log p(\mathcal{X}|\lambda) = \sum_{t=1}^{m} \gamma_i(\boldsymbol{x}_t)\Sigma_i^{-1}\left(\boldsymbol{\mu}_i - \boldsymbol{x}_t\right)\ , \tag{3.1}$$

where $\gamma_i(\boldsymbol{x}_t)$ is the soft-assignment of $\boldsymbol{x}_t$ to the i-th Gaussian component, i.e.,

$$\gamma_i(\boldsymbol{x}_t) = \frac{\omega_i \mathcal{N}(\boldsymbol{x}_t|\,\boldsymbol{\mu}_i, \Sigma_i)}{\sum_{j=1}^{k} \omega_j \mathcal{N}(\boldsymbol{x}_t|\,\boldsymbol{\mu}_j, \Sigma_j)}\ .$$

In VLAD, the input space $\mathbb{R}^d$ is first partitioned into k clusters by learning a codebook $\mathcal{D}$ with atoms $\{\boldsymbol{d}_i\}_{i=1}^{k}$. Then, for the aforementioned query set $\mathcal{X}$, the VLAD code $V \in \mathbb{R}^{kd}$ is obtained by stacking k Local Difference Vectors (LDV) $\boldsymbol{v}_i$ aggregating the differences $\boldsymbol{d}_i - \boldsymbol{x}_t$ in each cluster. More formally,

$$\boldsymbol{v}_i = \sum_{\boldsymbol{x}_t \in \boldsymbol{d}_i} \boldsymbol{d}_i - \boldsymbol{x}_t\ , \tag{3.2}$$

where $\boldsymbol{x} \in \boldsymbol{d}_i$ means that the local descriptor $\boldsymbol{x}$ belongs to the cluster defined by $\boldsymbol{d}_i$, i.e., the closest codeword to $\boldsymbol{x}$ is $\boldsymbol{d}_i$.

Comparing Eq. (3.1) to Eq. (3.2), one can observe the followings about VLAD

1. VLAD equally characterizes the distribution of local descriptors with respect to the centers. Hence, VLAD can be conceived as a non-probabilistic version of the FV.

2. In contrast to FV, in VLAD the covariance matrices of the mixture components are assumed to be diagonal and fixed, i.e., $\Sigma_i = \sigma \mathbf{I}_d, \ \forall i \in \{1, 2, \cdots, k\}$.

3.4.2 Sparse Coding

In Euclidean spaces, the idea of sparse coding is to reconstruct the query input $\boldsymbol{x}$ by a linear combination of codebook atoms, i.e., $\boldsymbol{x} = \sum_{i=1}^{k} \boldsymbol{d}_i [\boldsymbol{y}]_i$, such that a small number of codewords is involved Wright et al. [2009]. The problem of coding the single query input $\boldsymbol{x}_t$ can be formulated as solving the following minimization problem

$$\min_{\boldsymbol{y}} \ \left\| \boldsymbol{x}_t - \sum_{i=1}^{k} \boldsymbol{d}_i [\boldsymbol{y}]_i \right\|^2 + \lambda \left\| \boldsymbol{y} \right\|_1 , \tag{3.3}$$

where λ is the sparsity-promoter regularizer.

Since the codebook $\mathcal{D}$ is usually selected to be over-complete, i.e., $k > d$, the regularization is necessary to ensure that the under-determined system has a unique solution. Moreover, generally pooling methods such as average pooling or max pooling are performed on the resulting set $\mathcal{Y} = \{\boldsymbol{y}_t\}_{t=1}^{m}, \boldsymbol{y}_t \in \mathbb{R}^k$, to generate the final representation for the query set $\mathcal{X}$.

3.4.3 Locality-Constrained Linear Coding

LLC applies locality constraint to select similar atoms to the query and learns an affine combination of them to reconstruct the query Wang et al. [2010]. An approximated LLC algorithm is proposed by Wang et al. [2010] which first performs a K-nearest-neighbor (Knn) search and then analytically solves a constrained least squares problem. The affine combination of weights $\sum_{i=1}^{k} [\boldsymbol{y}]_i = 1$ (or equivalently $\mathbf{1}^T \boldsymbol{y} = 1$) is considered to ensure a shift invariant code is obtained

$$\min_{\boldsymbol{y}} \left\| \boldsymbol{x}_t - \sum_{\boldsymbol{d}_i \in Knn(\boldsymbol{x}_t)} \boldsymbol{d}_i [\boldsymbol{y}]_i \right\|^2, \quad \text{s.t.} \quad \mathbf{1}^{\mathrm{T}} \boldsymbol{y} = 1. \tag{3.4}$$

3.4.4 Collaborative Coding

Zhang et al. [2011] show that collaboratively reconstructing the query vector by codewords is effective for face recognition problem. To generate the face representation, a regularized least squares problem is solved as follows

$$\min_{\boldsymbol{y}} \ \left\| \boldsymbol{x}_t - \sum_{i=1}^{k} \boldsymbol{d}_i [\boldsymbol{y}]_i \right\|^2 + \lambda \left\| \boldsymbol{y} \right\|^2 , \tag{3.5}$$

where λ is the regularizer parameter.

Similar to the LLC coding, an analytic solution is obtained by zeroing out the derivative with respect to the variable $\boldsymbol{y}$. The induced sparsity is weaker than the original sparse coding method as the ℓ_2 norm is used for regularization.

3.5 Riemannian Coding Methods

In this section, we present our coding methods on Riemannian manifolds. In what follows, we assume that $\mathcal{X} = \{\boldsymbol{X}_t\}_{t=1}^{m}$, $\boldsymbol{X}_t \in \mathcal{M}$ and $\mathcal{D} = \{\boldsymbol{D}_i\}_{i=1}^{k}$, $\boldsymbol{D}_i \in \mathcal{M}$, are a set of local descriptors (extracted from a query image or video) and codewords on a Riemannian manifold $\mathcal{M}$, respectively. Moreover, let $\delta(\cdot,\cdot) : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ be a measure of similarity (e.g., geodesic distance) defined on $\mathcal{M}$.

3.5.1 Riemannian Bag of Words

In its most straightforward and simplest form, for the query set $\mathcal{X}$ and the codebook $\mathcal{D}$, a representation $\boldsymbol{y}$ is obtained by BoW algorithm using the hard assignment strategy Sivic and Zisserman [2003]. In this case, a histogram $\boldsymbol{y} \in \mathbb{R}^k$ is obtained by assigning each query point $\boldsymbol{X}_t$ to its closest codeword from the set $\mathcal{D}$ using the given measure δ in $\mathcal{M}$. The i-th dimension of $\boldsymbol{y}$, $[\boldsymbol{y}]_i$, is obtained using $[\boldsymbol{y}]_i = \#\big(\boldsymbol{X}_t \in \boldsymbol{D}_i\big)$, where $\#(\cdot)$ denotes the number of occurrences.

This obviously requires $m \times k$ comparisons. In the end, in order to add robustness to the number of extracted local descriptors, the resulting histogram is ℓ_2 normalized via $\widehat{\boldsymbol{y}} = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|}$.

3.5.2 Riemannian Vector of Locally Aggregated Descriptors

The key inspiration in VLAD coding is that it has been successfully used in addressing many challenging tasks such as image retrieval Jégou et al. [2012]; Gong et al. [2014], scene recognition Gong et al. [2014] and texture classification Cimpoi et al. [2014] , significantly raising the interest of the community in VLAD. The interest has even influenced the deep learning community Gong et al. [2014]; Cimpoi et al. [2015]. Besides, the discriminative representation obtained by VLAD is the result of rudimentary vector addition and subtraction. Another important merit is the reliance on small codebooks which further simplifies the learning stage and increases the popularity of VLAD.

In this section, we derive a general formulation for Riemannian VLAD (R-VLAD). To this end, we first start by devising R-VLAD on $\mathcal{M}$ when the similarity measure is the geodesic distance, i.e., $\delta_G : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$. We then discuss our universal solution in which any arbitrary similarity measure can take the role of δ_G and derive faster variants of R-VLAD. We conclude this section by introducing an approach to enrich R-VLAD by encoding more information about the distribution of the local descriptors and name it Higher Order R-VLAD (HO-R-VLAD).

3.5.2.1 R-VLAD: the geodesic distance scenario

A closer look at the signature generation steps of the conventional VLAD reveals that the LDVs are indeed the gradient of the ℓ^2 norm (or simply the Euclidean distance). By discarding the associated normalization terms in the FV algorithm we arrive at equity of the FV and VLAD. Having said that, it is easy to conclude that R-VLAD signature on $\mathcal{M}$ is obtained once we have the following tools at our disposal

- a metric δ required to determine how the local descriptors should be assigned to the codewords.
- operators to perform the role of vector addition or subtraction on $\mathcal{M}$.

Since a Riemannian manifold is a metric space, it is natural to choose the geodesic distance $\delta_G : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ to address the first requirement. As for the second requirement, we note that on a Riemannian manifold, one can see a vector $\overrightarrow{AB}$ (attached at point $\boldsymbol{A}$) as a vector of the tangent space at $\boldsymbol{A}$, i.e., $T_{\boldsymbol{A}}\mathcal{M}$. Therefore, subtraction on a Riemannian manifold can be attained through the logarithm map, $\log_{\boldsymbol{A}}(\cdot) : \mathcal{M} \to T_{\boldsymbol{A}}\mathcal{M}$. This concept has been used widely in the literature. For example, vector subtraction through the logarithm map was used to address the problem of interpolation and filtering Pennec et al. [2006], sparse coding Ho et al. [2013] and dimensionality reduction Goh and Vidal [2008], to name a few. The aforementioned discussion hints towards devising the R-VLAD as follows

- exploit the geodesic distance to determine the closest local descriptors to each codeword.
- build a Riemannian LDV per codeword using the tangent space attached to each codeword on the manifold.

Since the pole of the tangent space ($\boldsymbol{D}_i$ in our case) is fixed, the outputs of the logarithm map are compatible with each other and no further special care (e.g., parallel transport) is required[1]. Therefore, Eq. (3.2) on a curved Riemannian manifold boils down to

$$\boldsymbol{v}_i = \sum_{\boldsymbol{X}_t \in \boldsymbol{D}_i} \log_{\boldsymbol{D}_i}(\boldsymbol{X}_t) \,, \tag{3.6}$$

where $\log_{\boldsymbol{D}_i}(\cdot)$ is the logarithm map to the tangent space $T_{\boldsymbol{D}_i}$.

Although being perfectly accurate, the computational load of δ_G seems to be the sticking point as it leads to complex and slow algorithms, especially in our case where we have several local descriptors per query image/video. To alleviate this limitation, several studies recommend faster alternatives with excellent theoretical properties and similar results in practice Wang and Vemuri [2004]; Cherian et al. [2012]; Arsigny et al. [2007]; Hamm and Lee [2008]. This motivates us to engage other valid metrics and devise a universal form for our R-VLAD.

3.5.2.2 R-VLAD: arbitrary metric scenario

Obviously, for a new metric δ, we only need to take care of the second requirement. Since the LDV can be understood as the gradient of the distance function in the Euclidean case (see § 3.4.1), it is tempting to define the LDV on $\mathcal{M}$ as $\sum_{\boldsymbol{X}_t \in \boldsymbol{D}_i} \nabla_{\boldsymbol{D}_i} \delta^2(\boldsymbol{D}_i, \boldsymbol{X}_t)$[2]. The following theorem reinforces this idea even more.

[1] To be precise, this argument is valid as long as $\boldsymbol{x}_t$ is not in the cut locus of $\boldsymbol{c}_i$. This is of course not a very restricting assumption as in many manifolds (e.g., the SPD manifold) the cut locus is indeed empty.

[2] On an abstract Riemannian manifold $\mathcal{M}$, the gradient of a smooth real function f at a point $\boldsymbol{X} \in \mathcal{M}$, denoted by $\nabla_X f$, is the element of $T_X\mathcal{M}$ satisfying $\langle \nabla_X f, \zeta \rangle_X = Df_X[\zeta]$ for all $\zeta \in T_X\mathcal{M}$, where $Df_X[\zeta]$ denotes the directional derivative of f at x in the direction of ζ.

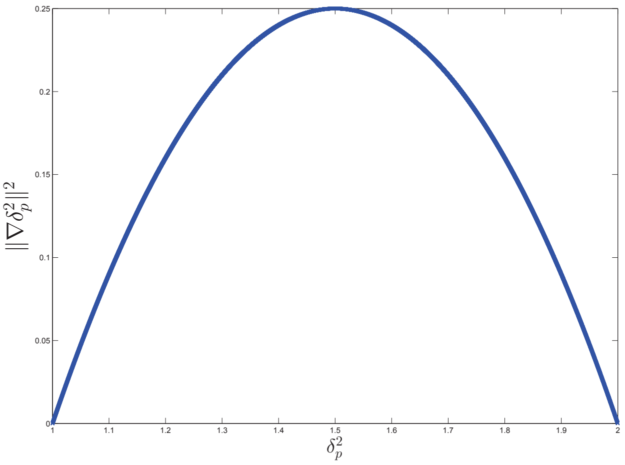

Figure 3.1: Illustration of the squared norm of the gradients vs distance for the projection distance on $\mathcal{G}_3^2$.

Theorem 1. *For a Riemannian manifold $\mathcal{M}$, the gradient of the geodesic distance function, $\delta_G : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ is*

$$\nabla_{\boldsymbol{X}}\delta_G^2(\boldsymbol{X},\boldsymbol{Y}) = -2\log_{\boldsymbol{X}}(\boldsymbol{Y}). \tag{3.7}$$

Proof. The interested reader is referred to Subbarao and Meer [2009] for the proof of this theorem. □

Unfortunately, choosing $\nabla_{\boldsymbol{D}_i}\delta^2(\boldsymbol{D}_i,\boldsymbol{X}_t)$ for LDV will not work in practice. The main reason being that for δ_G, the norm of $\nabla_{\boldsymbol{X}}\delta_G^2(\boldsymbol{X},\boldsymbol{Y})$ is related directly to the metric, i.e.,

$$\|\nabla_{\boldsymbol{X}}\delta_G^2(\boldsymbol{X},\boldsymbol{Y})\|^2 = 4\|\log_{\boldsymbol{X}}(\boldsymbol{Y})\|^2 = 4\delta_G^2(\boldsymbol{X},\boldsymbol{Y}).$$

This is of course inherited to the Euclidean space when the metric is chosen to be the geodesic distance, i.e., the Euclidean distance. However, this will not generalize to other metrics as shown by the following example.

Example 1. *Fig. 3.1 shows the behavior of $\|\nabla_{\boldsymbol{X}}\delta^2(\boldsymbol{X},\boldsymbol{Y})\|^2$ by varying $\delta^2(\boldsymbol{X},\boldsymbol{Y})$ for the projection metric on the Grassmann manifold $\mathcal{G}_3^2$ (see § 3.5.2.4 for the equations). Interestingly, the norm of the gradient will start decreasing while the point $\boldsymbol{Y}$ gets farther away from $\boldsymbol{X}$. During coding, a point which should contribute greatly to the descriptor, acts as an insignificant point, hence deteriorating the discriminatory power.*

The aforementioned example provides us with the following guideline for constructing an LDV on $\mathcal{M}$.

Algorithm 1 The proposed R-VLAD algorithm

```
Input:
    • local descriptors $\mathcal{X} = \{\boldsymbol{X}_t\}_{t=1}^{m}, \boldsymbol{X}_t \in \mathcal{M}$, extracted from a query image or video,
    • codebook $\mathcal{D} = \{\boldsymbol{D}_i\}_{i=1}^{k}, \boldsymbol{d}_k \in \mathcal{M}$
Output:
    • $\boldsymbol{V}(\mathcal{X})$ the Riemannian VLAD representation of $\mathcal{X}$
for $i = 1 \to k$ do
    Find $\boldsymbol{X}_t \in \boldsymbol{D}_i$, all nearest query points from $\mathcal{X}$ to $\boldsymbol{D}_i$
    Compute $\boldsymbol{v}_i$, i-th Local Difference Vector (LDV), using Eq. (3.8)
end for
Concatenate the resulting LDVs to form the final descriptor, i.e., $\boldsymbol{V}(\mathcal{X}) = \left[\boldsymbol{v}_1^T, \boldsymbol{v}_2^T, \cdots, \boldsymbol{v}_k^T\right]^T$
```

- the length of the LDV should represent the metric considered on $\mathcal{M}$.

As such, we propose the following form of LDV for our general R-VLAD descriptor (see Algorithm 1 for a step-by-step on the R-VLAD technique)

$$\boldsymbol{v}_i = \sum_{\boldsymbol{X}_t \in \boldsymbol{D}_i} \psi_\delta(\boldsymbol{D}_i, \boldsymbol{X}_t) \ , \tag{3.8}$$

where $\psi_\delta(\boldsymbol{D}, \cdot) : \mathcal{M} \times \mathcal{M} \to T_{\boldsymbol{D}}\mathcal{M}$ is defined as

$$\psi_\delta(\boldsymbol{D}_i, \boldsymbol{X}_t) = \delta(\boldsymbol{D}_i, \boldsymbol{X}_t) \frac{\nabla_{\boldsymbol{D}_i} \delta^2(\boldsymbol{D}_i, \boldsymbol{X}_t)}{\|\nabla_{\boldsymbol{D}_i} \delta^2(\boldsymbol{D}_i, \boldsymbol{X}_t)\|} \ .$$

Remark 1. *In line with the recommendations Jégou et al. [2012], post-processing of VLAD codes could increase the discriminatory power of the codes. In practice, we normalize the R-VLAD codes in two steps. First, an element-wise power normalization is performed using the transfer function $y : \mathbb{R} \to \mathbb{R}$, $y(x) = sign(x)\sqrt{|x|}$, where x is the element of VLAD vector and $|\cdot|$ denotes absolute value. This is to avoid having a concentrated distribution around zero. The power normalization is followed by an ℓ_2 normalization to make the energy of descriptors uniform.*

In the following two sections, we develop the R-VLAD for two widely used manifolds in computer vision, i.e., the SPD and the Grassmannian manifolds (see Table 3.1 for a quick peak at the studied metrics and the associated gradients as required by Eq. (3.8)).

Table 3.1: Metrics and associated gradients on the SPD and Grassmannian manifold.

Manifold	**Metric**	$\delta^2(X, Y)$	$\nabla_X \delta^2$
$\mathcal{S}_{++}^d$	geodesic	$\|\log(X^{-1/2}YX^{-1/2})\|_F^2$	$2X^{1/2}\log(X^{-1/2}YX^{-1/2})X^{1/2}$
$\mathcal{S}_{++}^d$	Stein	$\ln\det\left(\frac{X+Y}{2}\right) - \frac{1}{2}\ln\det(XY)$	$X(X+Y)^{-1}X - \frac{1}{2}X$
$\mathcal{S}_{++}^d$	Jeffrey	$\frac{1}{2}\operatorname{Tr}(X^{-1}Y) + \frac{1}{2}\operatorname{Tr}(Y^{-1}X) - d$	$\frac{1}{2}X\left(Y^{-1} - X^{-1}YX^{-1}\right)X$
$\mathcal{G}_d^p$	geodesic	$\|\Theta\|^2$	No analytic form
$\mathcal{G}_d^p$	projection	$2p - 2\|X^TY\|_F^2$	$-4\left(\mathbf{I}_d - XX^T\right)YY^TX$

3.5.2.3 R-VLAD on SPD Manifold

The gradient of a function $f : \mathcal{S}_{++}^d \to \mathbb{R}$ at X has the following form on $\mathcal{S}_{++}^d$ Sra and Hosseini [2014]

$$\nabla_X f = X\mathrm{sym}(Df)X, \tag{3.9}$$

where $\mathrm{sym}(X) = 0.5(X + X^T)$ and Df is the derivative of the function $f : \mathbb{R}^{d\times d} \to \mathbb{R}$ with respect to X.

The derivatives of $D\delta_S^2$ and $D\delta_J^2$ are reported in Cherian et al. [2012][3]. From Cherian et al. [2012] we can deduce the gradients required in the R-VLAD algorithm as depicted in Table 3.1.

Computational Cost

The computational load of coding in R-VLAD is dominated by the complexity of the used metric δ^2 and its gradient. On top of this, one should pay attention to the complexity of Riemannian codebook learning. As long as the complexity of coding is considered, the computational loads of computing δ_G^2, δ_J^2 and δ_S^2 are $4d^3$, $8/3d^3$ and d^3, respectively Cherian et al. [2012]. Computing the gradient of δ_G^2 requires an eigenvalue decomposition (for computing principal matrix logarithm) which adds up to a total of $9d^3$ flops for δ_G^2 (considering the matrix multiplications). For δ_J^2 and δ_S^2, computing gradient just requires a matrix inversion which is $O(d^3)$. As such, the computational load of R-VLAD using δ_J^2 and δ_S^2 is $O(17/3d^3)$ and $O(4d^3)$, respectively.

[3] Note that in Table 3 of Cherian et al. [2012] a scalar factor of 0.5 is wrongly dropped from the Jeffrey divergence (KLDM according to Cherian et al. [2012]). Also please note that the gradient reported in Cherian et al. [2012] is the Euclidean gradient not the Riemannian as required here.

3.5.2.4 R-VLAD on Grassmannian

The gradient of a function on Grassmannian, i.e., $f : \mathcal{G}_d^p \to \mathbb{R}$ has the form

$$\nabla_{\boldsymbol{X}} f = \left(\mathbf{I}_d - \boldsymbol{X}\boldsymbol{X}^T\right) Df, \tag{3.10}$$

where Df is a $d \times p$ matrix of partial derivatives of f with respect to the elements of $\boldsymbol{X}$, i.e.,

$$[Df]_{i,j} = \frac{\partial f}{\partial [\boldsymbol{X}]_{i,j}}.$$

The logarithm map (and also the exponential map) on Grassmannian does not have an analytic form. However, numerical methods for computing both mappings do exist. In particular, we will use the formulation introduced in Begelfor and Werman [2006] to compute R-VLAD using the geodesic distance. As for the projection metric, using Eq. (3.10) and noting that $\delta_P^2(\boldsymbol{X},\boldsymbol{Y}) = 2p - 2\|\boldsymbol{X}^T\boldsymbol{Y}\|_F^2$ leads to the following analytic form for the gradient as required in Eq. (3.8)

$$\nabla_{\boldsymbol{X}} \delta_P^2(\boldsymbol{X},\boldsymbol{Y}) = -4\left(\mathbf{I}_d - \boldsymbol{X}\boldsymbol{X}^T\right)\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{X}. \tag{3.11}$$

Computational Cost

We note that δ_G^2 on Grassmannian is obtained through Singular Value Decomposition (SVD). As such, computing δ_G^2 requires $dp^2 + p^3$ flops on $\mathcal{G}_d^p$. In contrast, the complexity of computing δ_P^2 on $\mathcal{G}_d^p$ is dp^2. Computing the gradient of δ_G^2 (or logarithm map) using a very efficient implementation requires a matrix inversion of size $p \times p$, two matrix multiplications of size $d \times p$, and a thin SVD of size $d \times p$. Computing thin SVD using a stable algorithm such as the Golub-Reinsch Golub and Van Loan [1996] requires $14dp^2 + 8p^3$ flops. This adds up to a total of $O\big(10p^3 + 17dp^2\big)$ flops for one local descriptor. As for δ_p^2, computing the gradient according to the Table 3.1 demands $4dp^2$ operations. This results in a total of $5dp^2$ flops for the projection metric.

To give the reader a better sense on the computational complexity of R-VLAD using δ_G^2 and δ_P^2, we measured the coding time for 1000 videos each with its own set of local descriptors on $\mathcal{G}_{177}^6$ (this is an example of the Grassmannian we will use in our experiments later). On a quad-core machine using Matlab, coding time for δ_P^2 and δ_G^2 were observed to be around 155 and 440 seconds, respectively.

3.5.2.5 Boosted R-VLAD

In this section, we introduce a variant of R-VLAD which in most cases further boosts the classification accuracy. We first note that the original VLAD formulation only considers simple first-order statistics of the LDVs to generate the final descriptor. Peng et al. [2014] address this issue and introduce coding of higher-order statistics into the VLAD framework. Assuming training data is clustered using a codebook, the idea is to compute two additional super vectors associated to each cluster, capturing the deviation of the LDVs from qualitative measures, namely the diagonal elements of the covariance matrix and the skewness of the training samples. Similar in spirit to VLAD, the two forms of high-order statistics are coded as complementary information.

Here, we further expand this idea to exploit complementary information and adapt it to our R-VLAD descriptor. To this end, we use the definition of LDV in § 3.5.2.2. Let the vector $\boldsymbol{\sigma_i}$ denotes diagonal elements of a covariance matrix constructed from the LDVs associated to $\boldsymbol{D}_i$ (training samples that are the closest to $\boldsymbol{D}_i$). In our case, the j-th element of the second-order super vector is computed as follows

$$[\boldsymbol{v}_i^{o^2}]_j = \frac{1}{\#(\boldsymbol{X}_t \in \boldsymbol{D}_i)} \sum_{\boldsymbol{X}_t \in \boldsymbol{D}_i} \big[\psi_\delta(\boldsymbol{D}_i, \boldsymbol{X}_t)\big]_j^2 - \big[\sigma_i\big]_j^2 \,, \tag{3.12}$$

with ψ defined below Eq. (3.8).

As for encoding the third-order statistics, skewness takes up the role of the diagonal elements of $\boldsymbol{\sigma_i}$

$$[\boldsymbol{v}_i^{o^3}]_j = \frac{\frac{1}{\#(\boldsymbol{X}_t \in \boldsymbol{D}_i)} \sum_{\boldsymbol{X}_t \in \boldsymbol{D}_i} \big[\psi_\delta(\boldsymbol{D}_i, \boldsymbol{X}_t)\big]_j^3}{\Big[\frac{1}{\#(\boldsymbol{X}_t \in \boldsymbol{D}_i)} \sum_{\boldsymbol{X}_t \in \boldsymbol{D}_i} \big[\psi_\delta(\boldsymbol{D}_i, \boldsymbol{X}_t)\big]_j^2\Big]^{\frac{3}{2}}} - \big[\Gamma_i\big]_j \,, \tag{3.13}$$

where Γ_i is the skewness vector of the training LDVs belonging to the i-th codeword.

The two super vectors $\boldsymbol{v}_i^{o^2}$ and $\boldsymbol{v}_i^{o^3}$ are concatenated and augmented to the original R-VLAD to form the final image/video signature. The power normalization is also performed in the end. We will dub this solution as Higher Order R-VLAD (HO-R-VLAD) in our experiments.

3.5.3 Riemannian Sparse Coding

As discussed earlier (see § 3.4.2), the goal of SC is to find a sparse vector of coefficients $\boldsymbol{y}$ in a way that a query point $\boldsymbol{x}$ is as close as possible to the linear combination $\sum_{i=1}^{k} \boldsymbol{d}_i [\boldsymbol{y}]_i$. While in $\mathbb{R}^n$, this problem seems to be well formulated, the difficulty arises when the query point (and subsequently each $\boldsymbol{d}_i$) belongs to $\mathcal{M}$, mainly because a universal coordinate system does not exist on $\mathcal{M}$. One natural modification to the notion of usual sparse coding is introduced by Xie et al. [2013] in which the term $\boldsymbol{x}_t - \sum_{i=1}^{k} \boldsymbol{d}_i[\boldsymbol{y}]_i$ in Eq. (3.3) is generalized for $\boldsymbol{X} \in \mathcal{M}$. The affine constraint $\mathbf{1}^T\boldsymbol{y} = 1$ is imposed to the code to avoid having the trivial solution $\boldsymbol{y} = 0$. The SC using the geodesic distance is cast as

$$\begin{aligned} &\min_{\boldsymbol{y}} \sum_{i=1}^{k} \left\| \log_{\boldsymbol{X}} \left(\boldsymbol{D}_i\right) \right\|^2 [\boldsymbol{y}]_i + \lambda \|\boldsymbol{y}\|_1 \,, \\ &\text{s.t.} \;\; \mathbf{1}^{\mathrm{T}}\boldsymbol{y} = 1. \end{aligned} \tag{3.14}$$

where $\log_{\boldsymbol{X}}(\cdot)$ is the logarithm map to the tangent space $T_{\boldsymbol{X}}$ and λ is the sparsity-promoter regularizer Wright et al. [2009].

With the aid of LDVs defined in § 3.5.2.2, we generalize the affine SC scheme to be used with an arbitrary metric δ. Our idea is to perform coding by minimizing the following objective function

$$\begin{aligned} &\min_{\boldsymbol{y}} \sum_{i=1}^{k} \left\| \psi_\delta(\boldsymbol{X}, \boldsymbol{D}_i) \right\|^2 [\boldsymbol{y}]_i + \lambda \|\boldsymbol{y}\|_1, \\ &\text{s.t.} \;\; \mathbf{1}^{\mathrm{T}}\boldsymbol{y} = 1. \end{aligned} \tag{3.15}$$

where ψ is defined below Eq. (3.8).

Similar to $\mathbb{R}^n$, the final descriptor of the set $\mathcal{X}$ is obtained by pooling the resulting $\{\boldsymbol{y}_t\}_{t=1}^{m}$ codes. We refer to this method as Riemannian Sparse Coding (R-SC) in our experiments.

3.5.4 Riemannian Locality-constrained Linear Coding

Similar in spirit to SC is the LLC Wang et al. [2010] in which the sparsity is a by-product of the locality constraint. LLC is easy to compute and gives superior image classification performance than many sophisticated approaches Wang et al. [2010]. The locality constraint is applied to select similar atoms of a codebook for coding. Like sparse coding, the goal is to learn a linear combination of the chosen atoms to reconstruct each query point.

Similar to LLC in $\mathbb{R}^n$, we have the luxury of a closed-form solution for our non-linear LLC. Having a metric δ at our disposal, for a query point $\boldsymbol{X} \in \mathcal{M}$, we first find $n \ll k$ nearest neighbors from atoms of $\mathcal{D}$ and then construct matrix $\boldsymbol{C}$ by stacking $\psi_\delta(\boldsymbol{X}, \boldsymbol{D}_i)$ selected vectors as its columns, i.e., $\boldsymbol{C} = \left[\psi_\delta(\boldsymbol{X}, \boldsymbol{D}_1) | \psi_\delta(\boldsymbol{X}, \boldsymbol{D}_2) | \cdots | \psi_\delta(\boldsymbol{X}, \boldsymbol{D}_n)\right]$. Then, the LLC code $\boldsymbol{y}$ is obtained by solving the following constrained least squares problem

$$\min_{y} \left\| Cy \right\|^2, \tag{3.16}$$
$$\text{s.t.} \quad \mathbf{1}^{\mathrm{T}} y = 1.$$

Here, again the affine constraint $\mathbf{1}^T y = 1$ is imposed to avoid having the trivial solution $y = \mathbf{0}$. As such, using the Lagrange multipliers technique, the code y is obtained in closed-form as

$$y = \frac{\left(C^T C\right)^{-1} \mathbf{1}}{\mathbf{1}^T \left(C^T C\right)^{-1} \mathbf{1}} . \tag{3.17}$$

In practice, a numerically stable way to minimize Eq. (3.17) is obtained through solving the set of n linear equations $C^T C y = 0$ followed by rescaling the coefficients y_i to ensure that $\mathbf{1}^T y = 1$ Saul and Roweis [2003]. We will dub this solution as Riemannian Locality-constrained Linear Coding (R-LLC) in our experiments.

3.5.5 Riemannian Collaborative Coding

In contrast to LLC, CC uses all dictionary atoms to represent the query sample. In the original CC, a regularized least squares problem is solved. Using δ, for a query point $X \in \mathcal{M}$, we first construct matrix C by stacking $\psi_\delta(X_t, D_i)$ vectors as its columns, i.e., $C = \left[\psi_\delta(X, D_1) | \psi_\delta(X, D_2) | \cdots | \psi_\delta(X, D_k)\right]$. Then, the code $y \in \mathbb{R}^k$ is obtained by solving the following constrained regularized least squares problem

$$\min_{y} \left\| Cy \right\|^2 + \lambda \left\| y \right\|^2, \tag{3.18}$$
$$\text{s.t.} \quad \mathbf{1}^{\mathrm{T}} y = 1.$$

To obtain the solution, again we use the Lagrange multipliers technique. Following a similar procedure to R-LLC, we obtain y as

$$y = \frac{\left(C^T C + \lambda I_k\right)^{-1} \mathbf{1}}{\mathbf{1}^T \left(C^T C + \lambda I_k\right)^{-1} \mathbf{1}} . \tag{3.19}$$

We will dub this solution as Riemannian Collaborative Coding (R-CC) in our experiments.

3.6 k-Means on Riemannian Manifolds

Before delving into experiments and for the sake of completeness, we provide details of learning a Riemannian codebook using different metrics introduced previously in §2.0.1. Like many other codebook learning algorithms, mean computation is a fundamental building block in our proposal. Therefore, we define the Fréchet mean which is incorporated in our Riemannian codebook learning.

Definition 8. *The Fréchet mean for a set of points $\{\boldsymbol{X}_i\}_{i=1}^n$, $\boldsymbol{X}_i \in \mathcal{M}$ is the minimizer of the cost function*

$$\boldsymbol{D}^* \triangleq \underset{\boldsymbol{D}}{\arg\min} \sum_{i=1}^{n} \delta^2(\boldsymbol{D}, \boldsymbol{X}_i) \, , \tag{3.20}$$

where $\delta : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ is the associated metric.

Generally, an analytic solution for Eq. (3.20) may not exist and hence iterative schemes that exploit the logarithm and exponential maps must be employed Pennec et al. [2006]. For high-dimensional manifolds, this could easily become overwhelming. Therefore, one reason in generalizations introduced in the previous section is that for some metrics Eq. (3.20) has analytic solution.

We train a codebook similar to the standard k-means algorithm using an iterative approach. The algorithm initiates by selecting k points from the training data randomly and calling them cluster centers. In one step, all the training samples are assigned to their nearest cluster center using the metric δ. In the next step, the cluster centers are re-estimated using the Fréchet mean.

On the SPD manifold and for the δ_G, the Fréchet mean is obtained using an iterative approach (see Pennec et al. [2006] for more details). For the Stein metric, we make use of the following theorem.

Theorem 2. *The Fréchet mean of a set of SPD matrices $\{\boldsymbol{X}_i\}_{i=1}^n \in \mathcal{S}_{++}^d$ with δ_S is obtained iteratively via*

$$\boldsymbol{\mu}^{(t+1)} = \left[\frac{1}{n} \sum_{i=1}^{n} \left(\frac{\boldsymbol{X}_i + \boldsymbol{\mu}^{(t)}}{2} \right)^{-1} \right]^{-1} . \tag{3.21}$$

Proof. See Cherian et al. [2012] for the proof. □

Unlike δ_G and δ_S which do not have an analytic form for the Fréchet mean, with the Jeffrey divergence, we have the luxury of obtaining the Fréchet mean analytically.

Theorem 3. *The Fréchet mean of a set of SPD matrices $\{\boldsymbol{X}_i\}_{i=1}^n \in \mathcal{S}_{++}^d$ with δ_J is*

$$\boldsymbol{\mu} = \boldsymbol{P}^{-1/2}(\boldsymbol{P}^{1/2}\boldsymbol{Q}\boldsymbol{P}^{1/2})^{1/2}\boldsymbol{P}^{-1/2} \;, \tag{3.22}$$

where $\boldsymbol{P} = \sum_i \boldsymbol{X}_i^{-1}$ and $\boldsymbol{Q} = \sum_i \boldsymbol{X}_i$.

Proof. The solution is obtained by zeroing out the derivative of $\sum_i^n \delta_J^2(\boldsymbol{X}_i, \boldsymbol{\mu})$ with respect to $\boldsymbol{\mu}$. At $\boldsymbol{\mu}$, $\frac{\partial \delta_J^2(\boldsymbol{X}_i,\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = \frac{1}{2}(\boldsymbol{X}_i^{-1} - \boldsymbol{\mu}^{-1}\boldsymbol{X}_i\boldsymbol{\mu}^{-1})$, we get

$$\begin{aligned}\frac{\partial \sum_i^n \delta_J^2(\boldsymbol{X}_i, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}} &= \sum_{i=1}^n \boldsymbol{X}_i^{-1} - \sum_{i=1}^n \boldsymbol{\mu}^{-1}\boldsymbol{X}_i\boldsymbol{\mu}^{-1} = 0 \\ &\Rightarrow \boldsymbol{\mu}\sum_{i=1}^n \boldsymbol{X}_i^{-1}\boldsymbol{\mu} = \sum_{i=1}^n \boldsymbol{X}_i. \end{aligned} \tag{3.23}$$

The quadratic equation $\boldsymbol{ABA} = \boldsymbol{C}$ is called a *Riccati* equation Bhatia [2007] and has the following unique and closed form solution for $\boldsymbol{B} \succ 0$ and $\boldsymbol{C} \succeq 0$

$$\boldsymbol{A} = \boldsymbol{B}^{-1/2}\big(\boldsymbol{B}^{1/2}\boldsymbol{C}\boldsymbol{B}^{1/2}\big)^{1/2}\boldsymbol{B}^{-1/2}$$

Comparing the form of Eq. (3.23) with the Riccati equation concludes the proof. We note that a different proof is also provided in Wang and Vemuri [2004]. □

Similarly, the projection metric has the following interesting property.

Theorem 4. *The Fréchet mean for a set of points $\{\boldsymbol{X}_i\}_{i=1}^n$, $\boldsymbol{X}_i \in \mathcal{G}_d^p$ based on δ_P admits a closed-form solution. That is the p largest eigenvectors of $\sum_{i=1}^n \boldsymbol{X}_i\boldsymbol{X}_i^T$.*

Proof. We need to solve

$$\begin{aligned}\boldsymbol{\mu}^* &= \underset{\boldsymbol{\mu}}{\arg\min} \sum_{i=1}^m \left\|\boldsymbol{\mu}\boldsymbol{\mu}^T - \boldsymbol{X}_i\boldsymbol{X}_i^T\right\|_F^2, \\ &\text{s.t. } \; \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu} = \mathbf{I}_{\mathrm{p}}.\end{aligned} \tag{3.24}$$

We note that with the orthogonality constraint on points, i.e., $\boldsymbol{\mu}^T\boldsymbol{\mu} = \boldsymbol{X}_i^T\boldsymbol{X}_i = \mathbf{I}_p$

$$\begin{aligned}&\sum_{i=1}^n \left\|\boldsymbol{\mu}\boldsymbol{\mu}^T - \boldsymbol{X}_i\boldsymbol{X}_i^T\right\|_F^2 = 2mp - 2\sum_{i=1}^n \mathrm{Tr}\{\boldsymbol{\mu}^T\boldsymbol{X}_i\boldsymbol{X}_i^T\boldsymbol{\mu}\} \\ &= 2mp - 2\,\mathrm{Tr}\{\boldsymbol{\mu}^T\Big(\sum_{i=1}^n \boldsymbol{X}_i\boldsymbol{X}_i^T\Big)\boldsymbol{\mu}\}.\end{aligned}$$

Therefore to minimize Eq. (3.24), one should maximize $\mathrm{Tr}\{\boldsymbol{\mu}^T\Big(\sum_{i=1}^n \boldsymbol{X}_i\boldsymbol{X}_i^T\Big)\boldsymbol{\mu}\}$ by taking into account the constraint $\boldsymbol{\mu}^T\boldsymbol{\mu} = \mathbf{I}_p$, i.e.,

$$\boldsymbol{\mu}^* = \underset{\boldsymbol{\mu}}{\arg\max}\ \mathrm{Tr}\{\boldsymbol{\mu}^T\Big(\sum_{i=1}^n \boldsymbol{X}_i\boldsymbol{X}_i^T\Big)\boldsymbol{\mu}\},\qquad(3.25)$$
$$\text{s.t.}\ \ \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu} = \mathbf{I}_{\mathrm{p}}.$$

The solution of Eq. (3.25) is obtained by computing the p largest eigenvectors of $\sum_{i=1}^n \boldsymbol{X}_i\boldsymbol{X}_i^T$ according to the Rayleigh-Ritz theorem Horn and Johnson [2012]; Faraki et al., which concludes the proof. □

3.7 Experiments

In this section, we present empirical evaluation of our proposal against the baseline and state-of-the-art for a number of visual recognition problems defined on the SPD and Grassmannian manifolds. In all our experiments, a set of overlapping blocks/cubes are extracted from images/videos. Then, each block/cube is represented by an RCovD or a linear subspace, hence it corresponds to a point on the SPD or the Grassmannian manifold, respectively.

The straightforward Log-Euclidean alternative of devising VLAD on a Riemannian manifold constitutes our first type of base-line. Here, we follow the terminology introduced in Arsigny et al. [2007] and label this as Log-Euclidean modeling or LE for short. Basically in the LE modeling, the manifold is embedded into a vector space through a fixed tangent space (centered at the identity matrix in our case). Therefore, one advantage of this ahead transformation of the points is that the rest of an encoding algorithm can be done by vector space algebraic operations, i.e., without concerning about the underlying structure.

Furthermore, we will consider the popular BoW representation of an image or video as another base-line method. Since the main step in BoW is measuring the distance of the query points from codewords, both intrinsic and LE variants are imaginable and evaluated in our experiments. In the intrinsic scenario, the codebook is learned by Riemannian k-means algorithm as described in § 3.6. Different algorithms tested in this section are referred to as

BoW$_{LE}$: Riemannian BoW model trained by flattening the manifold through the identity tangent space.

R-BoW$_{G}$: Riemannian BoW model using geodesic distance.

R-VLAD$_{LE}$: Similar in concept to the **BoW**$_{LE}$ but instead of BoW, we assess the performance of VLAD.

R-VLAD$_{G/J/S/P}$: R-VLAD using geodesic distance, the Jeffrey, Stein, or projection metrics.

HO-R-VLAD$_{G/J/S/P}$: Higher-Order R-VLAD using geodesic distance, the Jeffrey, Stein, or projection metrics.

R-SC$_{G/J/S/P}$: Riemannian sparse coding[4] using geodesic distance, the Jeffrey, Stein, or projection metrics.

R-LLC$_{G/J/S/P}$: Riemannian LLC coding using geodesic distance, the Jeffrey, Stein, or projection metrics.

R-CC$_{G/J/S/P}$: Riemannian CC coding using geodesic distance, the Jeffrey, Stein, or projection metrics.

Besides the Log-Euclidean and **R-BoW**$_G$ methods that serve as baseline methods, we will exclusively consider previous state-of-the-art algorithms for each studied problem to demonstrate the power of our methods. Here, our motivation is to provide a comprehensive study of the most popular coding techniques on a Riemannian manifold. Having the comparison at the disposal, one will be able to pick the most suitable methods to address the problem at hand.

3.7.1 SPD Manifold

For tests on the SPD manifold, an image is described by a set of RCovDs. More specifically, given a block $I(x,y)$ of size $W \times H$, let $\mathbb{O} = \{\boldsymbol{o}_i\}_{i=1}^{r}, \ \boldsymbol{o}_i \in \mathbb{R}^d$ be a set of r observations extracted from $I(x,y)$, e.g., $\boldsymbol{o}_i$ concatenates intensity values, gradients, filter responses, etc. for image pixel i. Then, block I can be represented by the $d \times d$ RCovD using Eq. (2.2). For classification, the descriptors (e.g., Log-Euclidean or R-VLAD) are fed to a simple Nearest Neighbor (NN) classifier which clearly shows the benefits of our proposal.

3.7.1.1 Head Pose Classification

We study the problem of head pose (orientation) classification utilizing two datasets, namely Heads Of Coffee break (HOCoffee) and Queen Mary University of London (QMUL) datasets Tosato

[4] In our experiments, as we have seen slight differences in accuracy values using different pooling methods, we maintain simplicity and adopt the average pooling method in constructing final descriptors.

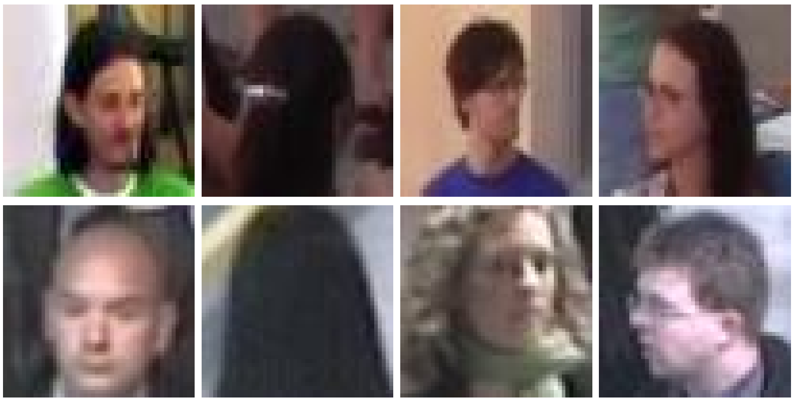

Figure 3.2: Examples of the HOCoffee and the QMUL datasets Tosato et al. [2013].

et al. [2013]. The HOCoffee dataset presents 18,117 low-resolution outdoor images, captured by a head detector for the purpose of automatically detecting social interactions. The QMUL head dataset is composed of 19,292 images, captured in an airport terminal. Images of both datasets are of size 50×50 pixels and split into a predefined training/test partition. There are 9,522 training and 8,595 test images in the HOCoffee dataset spanning six different classes (orientations): back, front, front-left, front-right, left and right. While for the QMUL dataset, 10,517 images are used for training and the remaining 8,775 images are considered for testing. The images are uniformly partitioned into five classes: back, front, left, right and background. The classification task is quite challenging since the datasets feature non-homogeneous illumination and severe occlusions.

As for image descriptor, similar to Tosato et al. [2013], we used a Difference Of Offset Gaussian (DOOG) filter-bank along color and image gradients for both datasets. More specifically, the feature vector assigned to each pixel in the image is

$$\boldsymbol{o}_{x,y} = \Big[I_L(x,y), I_a(x,y), I_b(x,y), \sqrt{I_x^2 + I_y^2}, \\ arctan\Big(\frac{|I_x|}{|I_y|}\Big), G_1(x,y), G_2(x,y), \cdots, G_8(x,y) \Big] ,$$

where $I_c(x,y)$, $c \in \{L, a, b\}$, denotes the CIELab color information at position (x,y), I_x and I_y are luminance derivatives, and $G_i(x,y)$ denotes the response of the i-th DOOG centered at $I_L(x,y)$. Therefore, each local RCovD is on $\mathcal{S}_{++}^{13}$.

In the first column of Table 3.2, we report the recognition accuracies of all the studied methods for the HOCoffee dataset. Several conclusions can be drawn here. First of all, even the simple **R-BOW**$_G$ outperforms the previous state-of-the-art method, demonstrating the ad-

Table 3.2: Recognition accuracies in % for the HOCoffee and QMUL datasets Tosato et al. [2013].

Method	**HOCoffee**	**QMUL**
WARCO	80.8 Tosato et al. [2013]	91.2 Tosato et al. [2013]
BOW$_{LE}$	81.6	87.2
R-BOW$_{G}$	81.8	87.6
VLAD$_{LE}$	82.4	87.8
R-SC$_{G}$	83.2	91.7
R-SC$_{S}$	83.1	91.6
R-SC$_{J}$	82.9	91.2
R-LLC$_{G}$	84.0	92.1
R-LLC$_{S}$	83.8	91.7
R-LLC$_{J}$	83.7	91.5
R-CC$_{G}$	82.7	90.6
R-CC$_{S}$	82.6	90.5
R-CC$_{J}$	82.4	90.1
R-VLAD$_{G}$	85.0	92.5
R-VLAD$_{S}$	84.9	92.5
R-VLAD$_{J}$	84.5	92.2
HO-R-VLAD$_{G}$	**85.3**	**92.9**
HO-R-VLAD$_{S}$	85.0	92.7
HO-R-VLAD$_{J}$	84.7	92.5

vantageous of local approaches. Compared to sparse coding techniques, R-VLAD coding with all studied metrics achieve higher performances (with a NN classifier), with **HO-R-VLAD**$_{G}$ being the overall winner in terms of classification accuracy. However, the performance of R-VLAD with the Stein and Jeffrey is on par or slightly worse than that of the geodesic solution while being at least 27 times faster in coding and 65 times faster (especially for the case of Jeffrey) in the training phase. We also observe that the proposed R-VLAD method is significantly superior as compared to the Log-Euclidean methods, which suggests that the underlying Riemannian structure is better exploited in R-VLAD.

Among sparse coding family methods, R-LLC using the geodesic distance obtains the highest accuracy. Here, collaborative construction of codes using all codebook atoms as in the variants of R-CC yields slightly inferior recognition accuracy. However, the accuracy numbers are still better than the previous state-of-the-art.

The second column of Table 3.2, reports recognition accuracies of all the studied methods for the QMUL dataset. Similar to the previous experiment, regardless of the metric, the performance is improved by considering the higher order information. The **HO-R-VLAD**$_G$ achieves the highest classification accuracy which is nearly 1.7 percentage points greater than Tosato et al. [2013]. Furthermore, the **R-VLAD**$_S$ works on par with the **R-VLAD**$_G$ while both are the preferred techniques to the **R-VLAD**$_J$ in terms of classification accuracy. Among the sparse coding family methods, the highest accuracy is obtained when the geodesic distance is used while sparse coding with the Jeffrey divergence yields the lowest accuracy number.

Moreover, we evaluated the performance of the VLAD in Euclidean space (**VLAD**$_E$) using very small to large codebook sizes to obtain signatures with the dimensionality similar or greater than that of R-VLAD's signatures. We observed that R-VLAD is significantly superior to **VLAD**$_E$. For instance, the best accuracy of **VLAD**$_E$ on the HOCoffee and QMUL datasets are 79.9% and 85.7%, respectively.

3.7.2 Grassmannian Manifold

For experiments on Grassmannian manifolds, we choose the application of recognition from videos by image-set modelling of the videos to create Grassmannian points. Similar to the experiments on the SPD manifolds, local descriptors of numerous small spatio-temporal blocks of a video are extracted. Then each cube is described by a linear subspace through SVD decomposition. We use a linear SVM classifier to further improve performances.

3.7.2.1 Dynamic Texture Classification

As our first experiment on Grassmannian manifolds, we tackled the task of dynamic texture recognition using the Dyntex++ dataset Ghanem and Ahuja [2010]. Dynamic textures are videos of moving scenes (such as Smoke, Waves, High way, Forest fire) that exhibit certain stationarity properties in time domain. The DynTex++ dataset contains 3600 ($50 \times 50 \times 50$) videos of moving scenes in 36 classes (see the first row in Fig. 3.3 for some examples).

To extract local Grassmannian points, each video was decomposed into 3D blocks of size $15 \times 15 \times 15$ with spatio/temporal overlap of 5 pixels/frames. Then, the 3D blocks were described by grouping their internal frames and describing each with the 3D extension of the Local Binary Pattern (LBP) Ojala et al. [2002], namely LBP in Three Orthogonal Planes (LBP-TOP) Zhao and Pietikainen [2007]. For each cube and from the LBPTOP features, we extracted a subspace of dimension 6 using SVD. This resulted in having local descriptors on $\mathcal{G}_{177}^{6}$. In total, we obtained 512 subspaces (Grassmannian points) from each video.

For this experiment, we followed the evaluation protocol used in Baktashmotlagh et al. [2014]. More specifically, half of the videos of each class were randomly chosen as training data and the remaining ones were used as test data. The process of random selection was repeated 10 times and average accuracy numbers along standard deviations are reported in the second column of Table 3.3.

Table 3.3 shows that using the projection metric, the proposed R-VLAD outperforms the state-of-the-art by more than 5 percentage points. Compared to Log-Euclidean solution, again R-VLAD is preferable though the gap is not as big as that of the previous experiment. Similar to the previous experiments, again R-VLAD is superior to the VLAD using the Log-Euclidean solution. However, the Log-Euclidean VLAD performs better than the state-of-the-art method of Baktashmotlagh et al. [2014]. Moreover, HO-R-VLAD boosts the accuracy values when the geodesic or projection metrics are utilized as similarity measures. HO-R-VLAD with projection metric, the **HO-R-VLAD**$_P$, achieves the highest average recognition rate of 97.8%.

3.7.2.2 Dynamic Scene Categorization

We conducted another experiment to classify videos of dynamic scenes (similar to the dynamic texture videos) using the Maryland "In-The-Wild" dataset Shroff et al. [2010]. The dataset is very challenging due to the web nature of videos, having significant camera motions, scene cuts, differences in appearance, frame rate, scale, viewpoint and illumination conditions. The videos span 13 categories (e.g., Avalanche) with 10 videos per each class. Some class examples are shown in Fig. 3.3.

Following the standard setup used in Faraki et al. [2016], we utilized the FC7 features of the CNN of Zhou et al. Zhou et al. [2014] trained on the Places dataset Zhou et al. [2014] with 205 scene classes and 2,5 million images. The utilized features are 4096 dimension which we subsequently reduce them to 400. To extract local Grassmannian points, we generated linear subspaces of order 6 by grouping every six consecutive frames with 90% overlap. As such each local descriptor belongs to $\mathcal{G}_{400}^{6}$. A leave-one-video-out validation protocol is used for consistency with previous study in Faraki et al. [2016]; Feichtenhofer et al. [2014].

The recognition accuracies for all the studied methods are shown in the third column of Table 3.3. To the best of our knowledge, the recent work of Faraki et al. [2016] has achieved the highest accuracy on this dataset. Our HO-R-VLAD using the grassmannian geodesic metric outperforms this state-of-the-art by 1.5%. Furthermore, the HO-R-VLAD using the projection metric achieves the highest accuracy, outperforming the state-of-the-art by more than 3 percentage points. Notably, HO-R-VLAD is superior to the R-VLAD using both metrics. Compared to the Log-Euclidean solution, R-VLAD is preferable, indicating the advantage of our proposal.

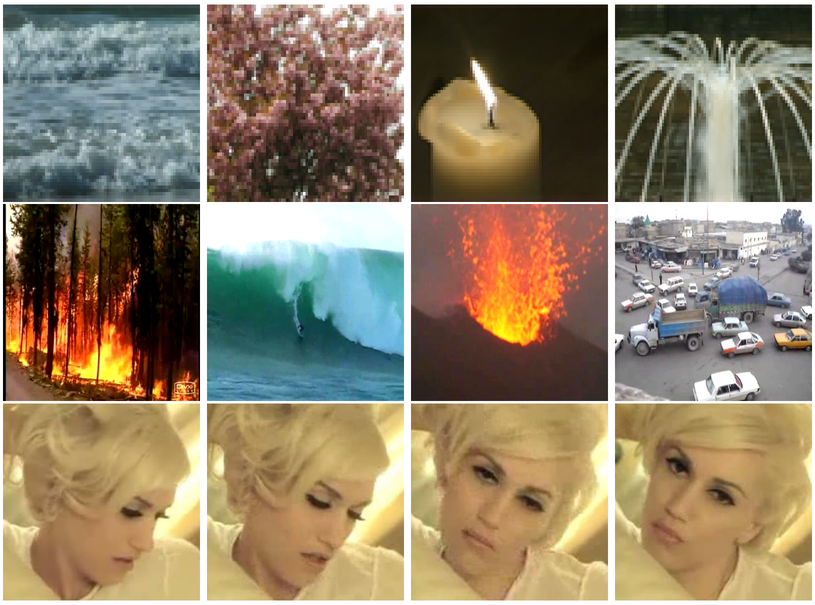

Figure 3.3: Examples of the DynTex++ Ghanem and Ahuja [2010] (first row), Maryland Shroff et al. [2010] (second row) and YTC Kim et al. [2008] (third row) datasets.

3.7.2.3 Face Recognition

As the last experiment, we considered the task of video-based face recognition. To this end, we considered the YouTube Celebrity (YTC) dataset Kim et al. [2008] which contains 1910 videos of 47 people (see Fig. 3.3). The large diversity of poses, illumination and facial expressions in addition to high compression ratio of face images provide significant challenges in this dataset.

For our evaluation, we followed the widely used setup in Deep Reconstruction Models (DRM) by Hayat et al. [2015]. More specifically, from each video, the face regions were first extracted using the tracker of Ross et al. [2008]. Then, without any further refinement, each face region was divided into distinct non-overlapping blocks and the histogram of LBP was extracted for each patch and concatenated to form the final frame descriptors.

As for the evaluation protocol, we note that various protocols were used by researchers on this dataset. Here again, we followed the five-fold cross validation protocol introduced in Hayat et al. [2015] which divides the whole dataset equally (with minimum overlap) into five folds with 9 videos per subject in each fold. Three of the videos were randomly selected for training while the remaining six were used for testing. We generated linear subspaces of order 6 by grouping features of every 6 consecutive frames. Therefore, each local descriptor belongs to $\mathcal{G}_{928}^{6}$.

The last column of Table 3.3 summarizes the average recognition rates and the standard

Table 3.3: Recognition accuracies in % for the Maryland Shroff et al. [2010], Dyntex++ Ghanem and Ahuja [2010] and YTC Kim et al. [2008] datasets. The previous best methods applied on the datasets are Baktashmotlagh et al. [2014], Faraki et al. [2016] and Hayat et al. [2014], respectively.

Method	**Dyntex++**	**Maryland**	**YTC**
Previous Best	92.4	90.0	72.6 $\pm$ 5.1
BOW$_{LE}$	81.1 $\pm$ 0.5	84.6	55.3 $\pm$ 2.9
R-BOW$_G$	92.4 $\pm$ 0.5	85.4	64.5 $\pm$ 5.1
VLAD$_{LE}$	93.3 $\pm$ 0.4	86.9	65.2 $\pm$ 2.8
R-SC$_G$	96.0 $\pm$ 0.4	87.7	74.1 $\pm$ 3.0
R-SC$_P$	96.1 $\pm$ 0.2	88.5	74.8 $\pm$ 5.2
R-LLC$_G$	96.3 $\pm$ 0.3	88.5	75.7 $\pm$ 3.2
R-LLC$_P$	96.3 $\pm$ 0.2	90.0	76.7 $\pm$ 4.8
R-CC$_G$	95.4 $\pm$ 0.5	86.9	75.2 $\pm$ 2.9
R-CC$_P$	95.8 $\pm$ 0.4	87.7	75.4 $\pm$ 2.0
R-VLAD$_G$	96.7 $\pm$ 0.3	90.0	75.6 $\pm$ 2.5
R-VLAD$_P$	97.6 $\pm$ 0.4	90.8	**79.9** $\pm$ **3.6**
HO-R-VLAD$_G$	97.0 $\pm$ 0.7	91.5	78.7 $\pm$ 3.8
HO-R-VLAD$_P$	**97.8** $\pm$ **0.4**	**93.1**	79.8 $\pm$ 3.7

Table 3.4: Computational loads of our coding methods in $\mathcal{S}_{++}^{13}$ (first row) and in $\mathcal{G}_{177}^{6}$ (second row) in seconds.

R-SC$_G$	**R-SC**$_S$	**R-SC**$_J$	**R-LLC**$_G$	**R-LLC**$_S$	**R-LLC**$_J$	**R-CC**$_G$	**R-CC**$_S$	**R-CC**$_J$	**R-VLAD**$_G$	**R-VLAD**$_S$	**R-VLAD**$_J$
7.1	5.9	6.3	6.8	5.0	6.1	7.0	5.2	6.2	2.1	0.4	0.1
R-SC$_G$	**R-SC**$_P$		**R-LLC**$_G$	**R-LLC**$_P$		**R-CC**$_G$	**R-CC**$_P$		**R-VLAD**$_G$	**R-VLAD**$_P$	
31.6	25.6		25.8	19.6		27.8	21.3		0.3	0.2	

deviations of all the studied methods. The results are self-explanatory. The R-VLAD with both geodesic and projection metric comfortably outperforms the state-of-the-art DRM algorithm. Furthermore, the accuracy gap between the Log-Euclidean solution and R-VLAD exceeds 10 percentage points. While encoding higher-order information improves the accuracy of R-VLAD using the geodesic distance, the maximum accuracy of 79.9% is achieved by R-VLAD when the projection metric is utilized.

Here, R-LLC coding is still the preferred coding method among the sparse coding schemes. Furthermore, collaborative coding improves the performance over simple sparse coding with both metrics, indicating this type of coding is more useful -as originally devised- for face recognition task.

3.7.3 Computational Load

To give the reader a better picture on the computational load of our coding methods, we recorded the average coding times for 100 descriptors on $\mathcal{S}_{++}^{13}$ and $\mathcal{G}_{177}^{6}$. These are indeed examples of the SPD and Grassmann manifolds which we had in our experiments. Table 3.4 shows the recorded times using Matlab on a quad-core computer, when different metrics are used in our coding methods. Since the extra computational load in **HO-R-VLAD** (over **R-VLAD**) is negligible, we removed that coding from the table.

3.8 Conclusions

Inspired by the recent success of coding/aggregating methods in Euclidean spaces and superior discriminative power of the descriptors on Riemannian manifolds, in this chapter we studied Riemannian coding methods such as VLAD and SC. In particular, we considered structured local descriptors from visual data, namely RCovD and linear subspaces that reside on the manifold of SPD matrices and the Grassmannian manifolds, respectively. In addition to a comprehensive formulation, we devised a family of methods that benefits from various forms of similarity measures defined on the underlying manifolds. An extensive set of experiments on several challenging vision tasks including head pose classification, face recognition from videos and dynamic scene categorization supported our proposal.

In the next chapter, we introduce some special types of covariance descriptors and similarity measures for them to perform inference methods.

Chapter 4

Infinite-Dimensional and Rank-Deficient Covariance Descriptors: Two Special Cases

4.1 Overview

In this chapter, we study two special types of covariance descriptors which do not readily conform to the usual development for Symmetric Positive Definite (SPD) matrices presented in § 2.0.1. We introduce methods to estimate infinite-dimensional Region Covariance Descriptors (RCovD) as the first special case. To do so, we exploit two feature mappings, namely random Fourier features Rahimi and Recht [2007] and the Nyström method Baker [1977]. Generally speaking, infinite-dimensional RCovDs offer better discriminatory power over their low-dimensional counterparts. However, the underlying Riemannian structure, i.e. the manifold of SPD matrices, is out of reach to great extent for infinite-dimensional RCovDs. To overcome this difficulty, we propose to approximate the infinite-dimensional RCovDs by making use of the aforementioned explicit mappings. We will empirically show that the proposed finite-dimensional approximations of infinite-dimensional RCovDs consistently outperform the low-dimensional RCovDs for image classification task, while enjoying the Riemannian structure of the SPD manifolds. Moreover, our methods achieve the state-of-the-art performance on three different image classification tasks.

Furthermore, we introduce novel similarity measures specifically designed for rank-deficient covariance descriptors, i.e. Symmetric Positive Semi-Definite (SPSD) matrices. In doing so, we are inspired by the fact that representing images and videos by covariance descriptors and leveraging the inherent manifold structure of SPD matrices leads to enhanced performances in various visual recognition tasks. However, when covariance descriptors are used to represent image sets, the result is often rank-deficient. Thus, most existing approaches adhere to blind perturbation with predefined regularizers just to be able to employ inference tools Wang et al. [2012]. To overcome this problem, we derive positive definite kernels that can be decomposed into the kernels on the cone of SPD matrices and kernels on the Grassmann manifolds. Our experiments evidence that, our method achieves superior results for image set classification on various recognition tasks including hand gesture classification, face recognition from video sequences, and dynamic scene categorization.

4.2 Introduction

The technical contribution in this chapter is twofold. Hence, to increase readability we present our novelties in two main parts.

Infinite-Dimensional RCovDs

We propose methods to approximate the recently introduced infinite-dimensional RCovDs Harandi et al. [2014a]; Quang et al. [2014]. The motivation here stems from the fact that the Riemannian geometry -which is essential in analyzing RCovDs- does not apply verbatim to the infinite-dimensional case. Hence, by approximating the infinite-dimensional RCovDs with finite-dimensional ones, one could seamlessly exploit the rich geometry of RCovDs and tools developed upon that to do the inference.

In an attempt to encode more information in RCovDs, harandi et al. have recently introduced infinite-dimensional RCovDs Harandi et al. [2014a]. To this end, a mapping from the low-dimensional Euclidean space to a Reproducing Kernel Hilbert Space (RKHS), i.e. $\phi : \mathbb{R}^d \to \mathcal{H}$, is used along the kernel trick to compute several forms of Bregman divergences between infinite-dimensional RCovDs in $\mathcal{H}$. In practice, infinite-dimensional RCovDs are rank deficient. This is because a valid d-dimensional RCovD requires more than d independent observations which translates into the impractical situation of having endless observations for the infinite-dimensional RCovDs. This difficulty, while partly resolved through regularization, deprives us from exploiting the geometry of the space. More specifically, tangent spaces, exponential and logarithm maps, and geodesics are out of reach to our best knowledge.

We overcome the aforementioned issue by introducing two methods to approximate infinite-dimensional RCovDs by finite-dimensional ones. To this end, we use random Fourier features and the Nyström method to learn a mapping $z : \mathbb{R}^d \to \mathbb{R}^D, d \leq D$ such that $\langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} \simeq$

$z(\boldsymbol{x}_i)^T z(\boldsymbol{x}_j)$. Having the mapping $z(\cdot)$ at our disposal, we approximate the infinite-dimensional RCovDs with $D \times D$ SPD matrices and take advantage of the Riemannian geometry of $\mathcal{S}_{++}^{D}$ to analyze the resulting RCovDs. We will show that both methods constantly outperform the low-dimensional RCovDs and achieve the state-of-the-art performance on three challenging image classification tasks, namely material categorization, virus cell identification, and scene classification. Moreover, our experiment shows that the RCovDs in the learned space could even outperform the infinite-dimensional ones. This is of course inline with findings in Lopez-Paz et al. [2014]; Rahimi and Recht [2009]; Le et al. [2013].

Symmetric Positive Semi-Definite Matrices

SPSD matrices naturally arise for applications where the number of observed samples is lower than the dimensionality of the samples, and a covariance matrix is used to represent the observations. One such application is image set classification where each set contains a number of images that belong to the same class. Compared to single image based classification, recognition from image sets has a significant advantage of efficiently exemplifying intra-class appearance variations such as pose changes, illumination differences, partial occlusions, and object deformations through multiple representatives Chen et al. [2013]; Hayat et al. [2014]. Therefore, proper modeling of image sets permits utilizing intra-class variation in the set as a complementary cue, thus enables discriminative representations Mahmood et al. [2014].

Covariance descriptors provide rich yet compact representations for image set modeling as they allow fusing various image cues while attenuating the impact of noisy samples through their averaging process Wang et al. [2012]; Faraki et al. [2014b]. Full rank covariance descriptors are symmetric and positive definite. Thus, from a geometrical point of view lie on the Riemannian manifold of SPD matrices. Given the power of modern inference frameworks on SPD manifolds (e.g., Faraki et al. [2015c,a]; Bonnabel and Sepulchre [2009]; Wang et al. [2012]; Faraki et al. [2015b]), several recent studies opt for modeling image sets using covariance descriptors Wang et al. [2012]; Faraki et al. [2014b].

Despite their success, a subtle point seems to be ignored. The covariance descriptor constructed from an image set is very unlikely to be full rank. This is simply because the dimensionality of the image descriptor is often greater than the number of available images in a set.

To overcome this difficulty, previous studies Wang et al. [2012]; Harandi et al. [2014b] adhere to ad-hoc solutions. One popular choice is to regularize the rank-deficient covariance

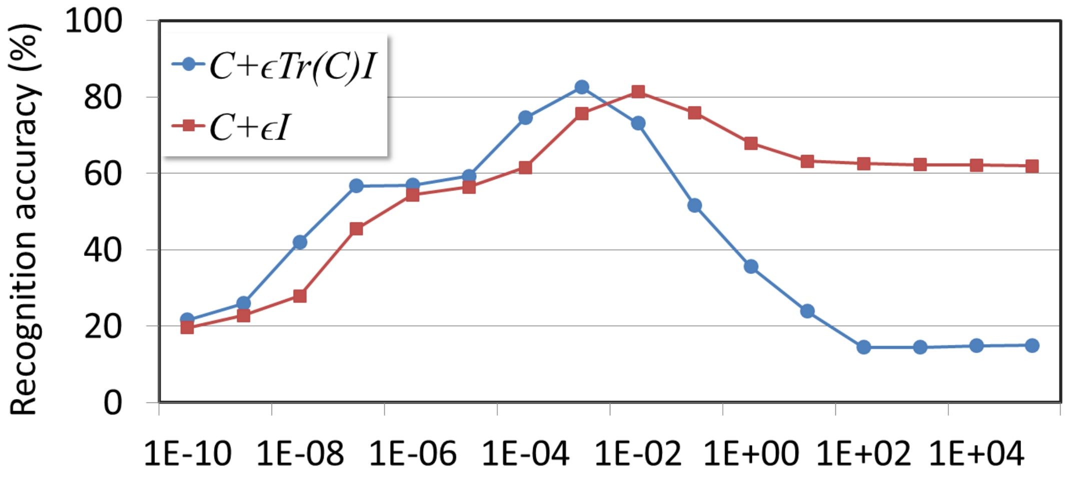

Figure 4.1: Recognition performance of a conventional nearest neighbor classifier using full rank matrices by regularizing the rank-deficient covariance descriptor. As seen, the performance changes drastically from 15% to 82% for different values of the regularization parameter ϵ.

descriptor (e.g., by adding a positive small value to the zero eigenvalues of the matrix). Such a regularization nonetheless may deteriorate the performance as shown in Figure 4.1. This is a very practical, albeit overlooked, problem lacking of a competent solution.

We propose a principle way of analyzing rank-deficient covariance descriptors by

- Making use of the distances defined on the manifolds of SPSD matrices.
- Introducing positive definite kernels on SPSD manifolds towards using kernel machines along rank-deficient covariance matrices.

Our experiments demonstrate the superiority of the proposed methods against several baseline and state-of-the-art methods. To the best of our knowledge, using the standard testing protocol, our method with the proposed kernels obtained in this new geometry equipped with kernel discriminant analysis classifier achieves the best reported results on standard image set classification benchmarks: 91.1% for Cambridge hand gesture recognition Kim et al. [2007b], 72.8% for YouTube celebrities face recognition Kim et al. [2008], and 90.0% for Maryland dynamic scene recognition Shroff et al. [2010].

4.3 Approximate Infinite-Dimensional RCovDs

We start this section by revisiting the region covariance descriptors Tuzel et al. [2008]. Let $X = \left[x_1|x_2|\cdots|x_n\right], x_i \in \mathbb{R}^d$ be a $d \times n$ matrix of n observations (extracted from an image

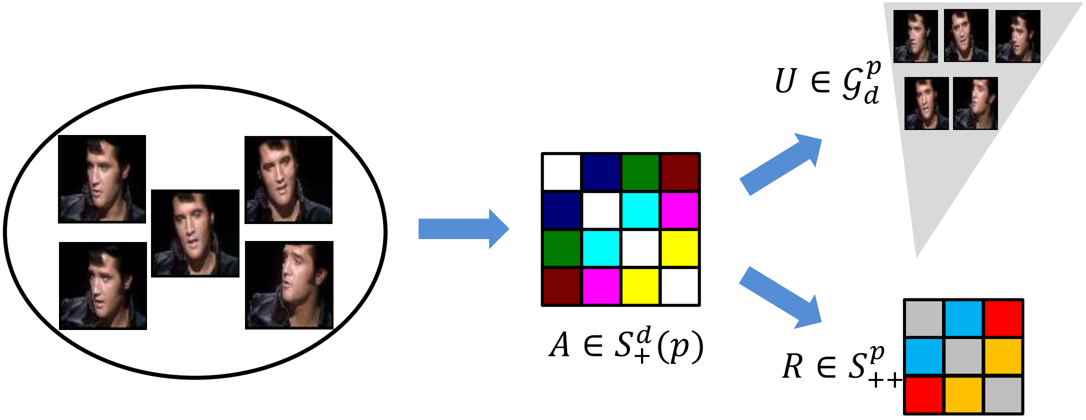

Figure 4.2: A conceptual example of our proposed image set representation. Image set is represented by an SPSD matrix A which is further decomposed to a linear subspace U and an SPD matrix R.

or a video). The RCovD $\boldsymbol{C} \in \mathcal{S}_{++}^d$ as its name implies is defined as

$$\boldsymbol{C} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T = \boldsymbol{X}\boldsymbol{J}\boldsymbol{J}^T\boldsymbol{X}^T, \tag{4.1}$$

where $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i$ is the sample mean of the observations, $\boldsymbol{J} = n^{-\frac{3}{2}}(n\mathbf{I}_n - \mathbf{1}_n\mathbf{1}_n^T)$, and $\mathbf{1}_n$ is a column vector of n ones.

Based on Eq. (4.1), an RCovD $\boldsymbol{C}_X$ in an RKHS $\mathcal{H}$ with dimensionality $|\mathcal{H}|$ can be defined as

$$\boldsymbol{C}_X = \Phi_X\boldsymbol{J}\boldsymbol{J}^T\Phi_X^T, \tag{4.2}$$

where $\Phi_X = \Big[\phi(\boldsymbol{x}_1)|\phi(\boldsymbol{x}_2)|\cdots|\phi(\boldsymbol{x}_n)\Big]$ and $\phi : \mathbb{R}^d \to \mathcal{H}$ is the implicit mapping to $\mathcal{H}$.

While embeddings into an RKHS seems preferable in many applications, the applicability of infinite-dimensional RCovDs is limited. This is evident by considering the situation where the dimensionality $|\mathcal{H}|$ approaches ∞, which leads to $\boldsymbol{C}_X$ being semi-definite. As a consequence, $\boldsymbol{C}_X$ is on the boundary of the positive cone and at infinite distance form SPD matrices.

In the following two sections, we will show how an infinite-dimensional $\boldsymbol{C}_X$ can be approximated by a finite $D \times D$ one.

4.3.1 Random Fourier Features

We start this section by providing a brief description of the method of random Fourier features for approximating $\phi(\cdot)$. Since in our experiments in § 4.3.3, we will only use RBF kernel, we limit the discussion here to this special kernel. The signature of other important kernels can be found in Rahimi and Recht [2007]; Vedaldi and Zisserman [2012].

According to the Bochner theorem Rudin [2011], a shift-invariant kernel[1] such as RBF kernel can be obtained by the following Fourier integral

$$k(\boldsymbol{x}_i - \boldsymbol{x}_j) = \int_{\mathbb{R}^d} p(\omega) e^{j\omega^T \boldsymbol{x}_i} e^{-j\omega^T \boldsymbol{x}_j} d\omega. \tag{4.3}$$

In other words, $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_i - \boldsymbol{x}_j)$ is the expected value of $\zeta_\omega(\boldsymbol{x}_i)\zeta^*_\omega(\boldsymbol{x}_j)$ according to the distribution $p(\omega)$ where $\zeta_\omega(\boldsymbol{x}) = e^{j\omega^T \boldsymbol{x}}$. As shown in Rahimi and Recht [2007], the function $z_F(\boldsymbol{x}) = \sqrt{2}\cos(\omega^T \boldsymbol{x} + b)$ satisfies the aforementioned criterion for real kernels, i.e., $E[z_F(\boldsymbol{x}_i) z_F(\boldsymbol{x}_j)] = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with ω and b being random variables drawn from $p(\omega)$ and $[0, 2\pi]$, respectively. For the RBF kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\|k(\boldsymbol{x}_i - \boldsymbol{x}_j)\|^2/2\sigma^2)$, $p(\omega) = \mathcal{N}(0, \sigma^{-2}\mathbf{I}_d)$ Rahimi and Recht [2007].

As such, let $\omega_1, \omega_2, \cdots, \omega_D$, $\omega_i \in \mathbb{R}^d$, be i.i.d samples drawn form the normal distribution $\mathcal{N}(0, \sigma^{-2}\mathbf{I}_d)$ and $b_1, b_2, \cdots, b_D$ be samples uniformly drawn from $[0, 2\pi]$. Then, the D dimensional estimation of $\phi(\boldsymbol{x})$ is given by

$$z_F(\boldsymbol{x}) = \sqrt{\frac{2}{D}} \Big[\cos(\omega_1^T \boldsymbol{x} + b_1), \cdots, \cos(\omega_D^T \boldsymbol{x} + b_D) \Big]. \tag{4.4}$$

Having the mapping $z_F : \mathbb{R}^d \to \mathbb{R}^D$ at our disposal, our first estimation of an infinite-dimensional RCovD can be obtained as

$$\hat{\boldsymbol{C}}_X = \Phi_X \boldsymbol{J}\boldsymbol{J}^T \Phi_X^T, \tag{4.5}$$

where $\Phi_X = \Big[z_F(\boldsymbol{x}_1) | z_F(\boldsymbol{x}_2) | \cdots | z_F(\boldsymbol{x}_n) \Big]$.

Algorithm 2 outlines the details of computing RCovDs using random Fourier features for the RBF kernel.

4.3.2 Nyström Method

While in § 4.3.1, an approximation to the embedding function $\phi(\cdot)$ was provided, we note that not only an arbitrary kernel $k(\cdot, \cdot)$ may not satisfy the Bochner theorem (e.g., if it is not shift-invariant), but even if it is, it may not be possible to obtain $p(\omega)$ analytically. To alleviate this limitation, we propose a data-dependent estimation of the RKHS $\mathcal{H}$ using the Nyström method Baker [1977].

[1] A kernel function is shift invariant if $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_i - \boldsymbol{x}_j)$.

Algorithm 2 Approximate infinite-dimensional RCovD using random Fourier features

```
Input:
    • $\boldsymbol{X} = \big[\boldsymbol{x}_1|\boldsymbol{x}_2|\cdots|\boldsymbol{x}_n\big]$, $\boldsymbol{x}_i \in \mathbb{R}^d$, matrix of $n$ feature vectors
    • $\sigma^2$, scale of the RBF kernel
    • $D$, target dimensionality
Output:
    • $\hat{\boldsymbol{C}}_X \in \mathcal{S}_{++}^D$, approximate infinite-dimensional RCovD
$\{\omega_i\}_{i=1}^D \leftarrow$ i.i.d samples drawn from $\mathcal{N}(0, \sigma^{-2}\mathbf{I}_{d\times d})$.
$\{b_i\}_{i=1}^D \leftarrow$ uniform samples drawn from $[0, 2\pi]$.
for $j = 1 \rightarrow n$ do
    Compute $z_F(\boldsymbol{x}_j)$ using Eq. (4.4).
end for
Compute $\hat{\boldsymbol{C}}_X$ using Eq. (4.5)
```

Given $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_M\}$ a collection of M training examples[2], a rank D approximation of $\boldsymbol{K} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{M\times M}$ can be written as $\boldsymbol{Z}^T\boldsymbol{Z}$. Here, $\boldsymbol{Z}_{D\times M} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{V}$ with $\boldsymbol{\Sigma}$ and $\boldsymbol{V}$ being the top D eigenvalues and corresponding eigenvectors of $\boldsymbol{K}$. Based on this low-rank approximation, one can obtain a D-dimensional vector representation of the space $\boldsymbol{K}$ as

$$z_N(\boldsymbol{x}) = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{V}\Big(k(\boldsymbol{x}, \boldsymbol{x}_1), \cdots, k(\boldsymbol{x}, \boldsymbol{x}_M)\Big)^T. \tag{4.6}$$

Given $\boldsymbol{X} = \Big[\boldsymbol{x}_1|\boldsymbol{x}_2|\cdots|\boldsymbol{x}_n\Big]$, a set of n observations, the corresponding RKHS region covariance descriptor estimation using the Nyström method is obtained as

$$\hat{\boldsymbol{C}}_X = \Phi_X \boldsymbol{J}\boldsymbol{J}^T\Phi_X^T, \tag{4.7}$$

where $\Phi_X = \Big[z_N(\boldsymbol{x}_1)|z_N(\boldsymbol{x}_2)|\cdots|z_N(\boldsymbol{x}_n)\Big]$.

Algorithm 3 summarizes the discussion about estimating RCovDs using the Nyström method in one pseudo-code.

4.3.3 Experiments

In this section, we evaluate the proposed approximate infinite-dimensional RCovDs on three different classification tasks, namely material categorization, virus cell identification, and scene

[2] Observations extracted from training images in our case.

Algorithm 3 Approximate infinite-dimensional RCovD using the Nyström method

```
Input:
    • $X = [x_1|x_2|\cdots|x_n]$, $x_i \in \mathbb{R}^d$, matrix of $n$ feature vectors
    • $\mathcal{D} = \{x_i\}_{i=1}^{M}, x_i \in \mathbb{R}^d$, a collection of training examples
    • $D$, target dimensionality
Output:
    • $\hat{C}_X \in \mathcal{S}_{++}^{D}$, approximate infinite-dimensional RCovD
Compute the kernel matrix $K = [k(x_i, x_j)]_{M \times M}$.
$\Sigma \leftarrow$ diagonal matrix of top $D$ eigenvalues of $K$.
$V \leftarrow$ associated eigenvectors of $\Sigma$.
for $j = 1 \rightarrow n$ do
    Compute $z_N(x_j)$ using Equation 4.6.
end for
Compute $\hat{C}_X$ using Equation 4.7.
```

classification. For benchmarking, we compare the accuracy of the Nearest Neighbor (NN) classifier on low-dimensional manifold against NN in higher-dimensional manifolds obtained by random Fourier features or the Nyström method.

Beside NN classifier, we will evaluate the performance of the state-of-the-art method of Covariance Discriminant Learning (CDL) Wang et al. [2012] for low and high-dimensional SPD manifolds. The CDL technique utilizes the identity tangent space of the SPD manifold to perform kernel Partial Least Squares (kPLS) Rosipal and Trejo [2002]. Partial Least Squares (PLS) can be understood as a dimensionality reduction technique that models relations between two sets of variables through a latent space. In the context of classification, PLS and its kernelized version can be used to model the relations between feature vectors and their representative classes.

The different algorithms evaluated in our experiments are referred to as

- **NN**: AIRM based NN classifier on low-dimensional RCovDs.
- $\mathbf{NN_F}$: AIRM based NN classifier on approximate infinite-dimensional RCovDs obtained by random Fourier features.
- $\mathbf{NN_N}$: AIRM based NN classifier on approximate infinite-dimensional RCovDs obtained by the Nyström method.
- **CDL**: CDL on low-dimensional RCovDs.

- $\mathbf{CDL_F}$: CDL on approximate infinite-dimensional RCovDs obtained by random Fourier features.
- $\mathbf{CDL_N}$: CDL on approximate infinite-dimensional RCovDs obtained by the Nyström method.

In what follows, we first elaborate on how rich RCovDs can be obtained for each task. This is followed by in-depth discussions on the performance of approximate infinite-dimensional RCovDs obtained through the processes described in § 4.3.1 and § 4.3.2, respectively.

4.3.3.1 Material Categorization

Material categorization is the task of classifying materials from their appearance in single images taken under unknown viewpoint and illumination conditions. For this experiment, we have used the UIUC material classification dataset Liao et al. [2013] which contains 18 classes of complex material categories "taken in the wild" (see Fig. 4.3 for sample images). The images were mainly selected to have various geometric fine-scale details. We split the database into training and test sets by randomly assigning half of the images of each class to the training set and using the rest as test data. The process of random splitting was repeated 10 times and the average recognition accuracies along standard deviations will be reported here.

To generate RCovDs, a feature vector is assigned to each pixel at position (x, y) in an image I by

$$\begin{aligned} F_{(x,y)} = \Big[& I_R(x,y), I_G(x,y), I_B(x,y), \left|\frac{\partial I}{\partial x}\right|, \left|\frac{\partial I}{\partial y}\right|, \left|\frac{\partial^2 I}{\partial x^2}\right|, \\ & \left|\frac{\partial^2 I}{\partial y^2}\right|, |G_{(0,0)}(x,y)|, \cdots, |G_{(u,v)}(x,y)| \Big], \end{aligned} \tag{4.8}$$

where $I_c(x, y), c \in \{R, G, B\}$, denotes color information, the next four entries are the magnitude of intensity gradients and the magnitude of Laplacians along x and y directions, and $G_{(u,v)}(x, y)$ is the response of a 2D Gabor wavelet Lee [1996] centered at (x, y) with orientation u and scale v. We extracted Gabor wavelets at four orientations and three scales. Therefore, each pixel is described by a 19 dimensional feature vector (i.e., 3 color, 4 gradients, and 12 Gabor features).

Table 4.1 shows the recognition accuracies for the studied methods. The correct classification rates obtained by simple NN clearly show that the proposed RCovDs are more discriminative than their low-dimensional counterparts. More specifically, when using the random

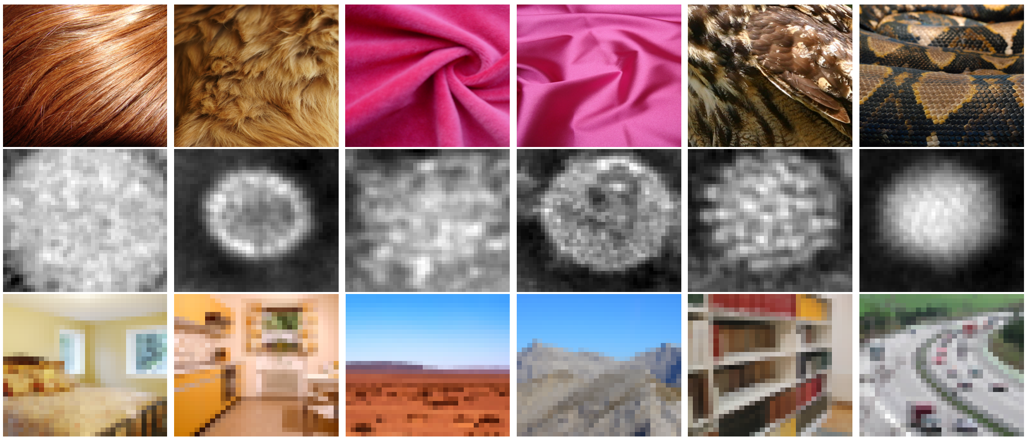

Figure 4.3: Sample images for datasets used in this work. Top: UIUC Liao et al. [2013], Middle: Virus Kylberg et al. [2011], Bottom: TinyGraz03 Wendel and Pinz [2007].

Fourier features and the Nyström methods to generate RCovDs, the average accuracy numbers boost from 26.5% to 35.9% and 35.6%, respectively. We also note that $\mathbf{NN_F}$ and $\mathbf{NN_N}$ achieve comparable performances to the more involved CDL in low-dimensional manifold.

The state-of-the-art performance on this dataset is 43.5% reported by Liao et al. [2013]. CDL on the proposed RCovDs (both random Fourier features and Nyström) outperforms the state-of-the-art performance by at least 2.8% percentage points.

4.3.3.2 Virus Classification

We performed an experiment to classify cell images using the Virus dataset Kylberg et al. [2011]. The dataset contains 1500 images of 15 different classes (100 samples per class). The images are formed from Transmission Electron Microscopy technique and re-sampled to 41×41 pixel gray-scale image (see Fig. 4.3 for examples). Here, RCovDs are obtained using the features described in Eq. (4.8) with one modification. For this task, we used Gabor wavelets at four orientations and five scales.

Our empirical results are reported in Table 4.1. The average correct recognition rate with both $\mathbf{CDL_F}$ and $\mathbf{CDL_N}$ is superior to the state-of-the-art performance of 81.2% reported in Harandi et al. [2014a] using infinite-dimensional RCovDs. We conjecture that computing the RCovDs with both random Fourier features and the Nyström method reveals the nonlinear patterns in data (as also evidenced in Lopez-Paz et al. [2014]). This is emphasized by the Riemannian structure of $\mathcal{S}_{++}^{D}$ (as CDL requires its tangent space) which is not available for the infinite-dimensional RCovDs.

Table 4.1: Recognition accuracies for the UIUC Liao et al. [2013], Virus Kylberg et al. [2011], and TinyGraz03 Wendel and Pinz [2007] datasets.

Method	UIUC	Virus	TinyGraz03
NN	26.5% ± 3.7	58.8% ± 5.4	34%
$\mathbf{NN_F}$	35.9% ± 3.0	67.1% ± 4.2	42%
$\mathbf{NN_N}$	35.6% ± 2.7	69.5% ± 4.8	44%
CDL	36.3% ± 2.0	75.5% ± 2.5	41%
$\mathbf{CDL_F}$	**47.4% ± 3.1**	**82.5% ± 2.9**	55%
$\mathbf{CDL_N}$	46.3% ± 2.6	81.4% ± 3.1	**57%**

4.3.3.3 Scene Classification

For the last experiment, we considered the task of scene classification using TinyGraz03 dataset Wendel and Pinz [2007]. The dataset contains 1148 indoor and outdoor images (see Fig. 4.3 for examples) with a spatial resolution of 32 × 32 pixels. The images are presented in 20 classes with at least 40 samples per class. This dataset is quite diverse, with scene categories being captured from various viewpoints and under various lighting conditions. We used the recommended train/test split provided by the authors. The correct recognition rate achieved by humans on this dataset is 30% Wendel and Pinz [2007].

The RCovDs for this task were obtained using the first 7 features in Eq. (4.8) (i.e., 3 color and 4 image gradients). Table 4.1 indicates that computing RCovDs using random Fourier features and the Nyström method offers notable enhancement in term of discriminatory power over the original RCovDs. We also note that $\mathbf{NN_F}$ and $\mathbf{NN_N}$ outperform the more involved **CDL**.

The state-of-the-art recognition accuracy on this dataset is reported to be 46% Wendel and Pinz [2007]. Interestingly, $\mathbf{CDL_F}$ and $\mathbf{CDL_N}$ significantly outperform the state-of-the-art method (more than 9 percentage points) and human performance (more than 25 percentage points).

4.4 Image Set Classification by Symmetric Positive Semi-Definite Matrices

Similar to previous section, we start by looking at how a covariance descriptor is made for image set classification task. Let $\mathbb{R}^{d \times p} \ni \mathbb{F} = \left[f_1 | f_2 | \cdots | f_p \right]$ denote a set containing the d-dimensional feature descriptors of p images of an image set. The covariance descriptor $\boldsymbol{C}$

representing the set $\mathbb{F}$ is

$$\boldsymbol{C} = \frac{1}{p-1} \sum_{i=1}^{p} (\boldsymbol{f}_i - \boldsymbol{\mu})(\boldsymbol{f}_i - \boldsymbol{\mu})^T, \tag{4.9}$$

where $\boldsymbol{\mu} = \frac{1}{p} \sum_{i=1}^{p} \boldsymbol{f}_i$ is the sample mean of the observations.

When $d > p$, $\boldsymbol{C}$ is rank-deficient, which means that the resulting matrix is on the boundary of the positive cone. As such, the machineries developed using SPD geometry to analyze such covariance descriptors will be no longer available. For instance, the distance from any SPD matrix to $\boldsymbol{C}$ would be infinite according to the AIRM. To overcome this issue, off-the-shelf treatment (for example proposed in Wang et al. [2012]) is through regularizing the original $\boldsymbol{C}$, i.e.,

$$\boldsymbol{C}^* = \boldsymbol{C} + \epsilon \mathrm{I}_d \ , \tag{4.10}$$

where ϵ is a constant and I_d is the $d \times d$ identity matrix.

As we will show in our experiments, the perturbation deteriorates the discriminatory power of covariance descriptors. Here, we are interested in taking the advantage of true geometry of the resulting covariance descriptors. To this end, we make use of the Riemannian structure of SPSD matrices of fixed rank introduced by Bonnabel and Sepulchre Bonnabel and Sepulchre [2009]. Below, we briefly discuss the natural metric and the geodesic distance for SPSD matrices and then turn our attention to create valid positive definite kernels.

4.4.1 Earlier Works

Almost all image set classification techniques have to make two major decisions: **1.** how to represent an image set, and **2.** what metric to use to measure the similarity between sets.

From the representation point of view, existing solutions can be divided roughly into model-driven and topology-driven approaches. As for the model-driven methods such as Li et al. [2009]; Nishiyama et al. [2007], it is usually assumed that the images within a set belong to a certain parametric form (e.g. distribution). Once, the model for each image set is determined, the similarity between sets can be obtained either as the distance between models (e.g. Kullback-Leibler divergence between Gaussian models) or more directly as the distance between the estimated parameters. The notable examples in this school of thought are modeling sets by Gaussian distribution Shakhnarovich et al. [2002]; Arandjelovic et al. [2005] and more recently with data-driven distributions Harandi et al. [2015b]. Clearly, the performance of model-driven methods will deteriorate if the set data is weakly correlated to the model.

To alleviate this difficulty, the topology-driven methods assume data establish a topological space and represent image sets by/on nonlinear manifolds Kim et al. [2007a]; Wang et al. [2008]; Hamm and Lee [2008]; Harandi et al. [2015a]; Wang et al. [2012]; Chen et al. [2013]. In Kim et al. [2007a] authors propose to learn a discriminant function that maximizes the canonical correlations of within-class sets while minimizing the canonical correlations of between-class sets. The concept of principal angles have been successfully utilized in Wang et al. [2008] for image sets matching through linear subspaces. More involved techniques exploit the geometry of the space of linear subspaces, i.e. Grassmann manifolds, to match image sets Harandi et al. [2015a]. Wang et al. [2012] model image sets by their natural second-order statistics, i.e. covariance matrices. Since nonsingular covariance matrices lie on a Riemannian manifold, a kernel function is used to explicitly embed the Riemannian structure into a Euclidean space. By exploiting the underlying geometrical structure, topology-driven methods provide robustness to noise and can operate with a relatively small number of samples per class.

In line with the success of deep learning architectures, Hayat et al. [2014] learn class-specific models by an Adaptive Deep Network Template (ADNT). Based on the minimum reconstruction error from the learned models, a majority voting strategy is used for classification. Furthermore, Lu et al. [2015] propose a multi-manifold deep metric learning approach which learns multiple sets of nonlinear transformations. Their method nonlinearly maps multiple sets of image instances into a shared feature subspace, hence more discriminative information is used for classification.

4.4.2 Geometry of SPSD space

Let us denote the set of SPSD matrices of rank p by $\mathcal{S}_{+}^{d}(p)$. For example $\boldsymbol{C}$ described in the previous section lies on $\mathcal{S}_{+}^{d}(p)$. We note that any $\boldsymbol{A} \in \mathcal{S}_{+}^{d}(p)$ can be decomposed as

$$\boldsymbol{A} = \boldsymbol{Z}\boldsymbol{Z}^{T} = (\boldsymbol{U}\boldsymbol{R})(\boldsymbol{U}\boldsymbol{R})^{T} = \boldsymbol{U}\boldsymbol{R}^{2}\boldsymbol{U}^{T} \,, \tag{4.11}$$

where $\boldsymbol{Z}$ is a full-rank $d \times p$ matrix, $\boldsymbol{U} \in \mathcal{S}_{p}^{d}$, and $\boldsymbol{R}^{2} \in \mathcal{S}_{++}^{p}$. Here, $\mathcal{S}_{p}^{d}$ denotes the Stiefel manifold, the set of orthogonal matrices, i.e., $\boldsymbol{U} \in \mathcal{S}_{p}^{d}$ iff $\boldsymbol{U}^{T}\boldsymbol{U} = \mathbf{I}_{p}$.

Eqn (4.11) remains unchanged under the transformation $\boldsymbol{Z} \rightarrow \boldsymbol{Z}\boldsymbol{O}$ for any matrix $\boldsymbol{O} \in \mathcal{O}_{p}$. Thus, one can deduce that the equivalence relation $(\boldsymbol{U}, \boldsymbol{R}^{2}) \equiv (\boldsymbol{U}\boldsymbol{O}, \boldsymbol{O}^{T}\boldsymbol{R}^{2}\boldsymbol{O})$ holds. As a result, the set $\mathcal{S}_{+}^{d}(p)$ admits a quotient manifold representation $\mathcal{S}_{+}^{d}(p) \cong \left(\mathcal{S}_{p}^{d} \times \mathcal{S}_{++}^{p}\right)/\mathcal{O}_{p}$.

Bonnabel and Sepulchre define the metric on $\mathcal{S}_+^d(p)$ based on the sum of infinitesimal displacements on $\mathcal{G}_d^p$ and $\mathcal{S}_{++}^p$ Bonnabel and Sepulchre [2009]. Let $\triangle$ and $\boldsymbol{D}$ represent the tangent vectors in Grassmannian and SPD manifolds, respectively. For $\mathcal{S}_+^d(p) \ni \boldsymbol{A} = \boldsymbol{U}\boldsymbol{R}^2\boldsymbol{U}^T$ and two pair of tangent vectors $(\triangle_1, \boldsymbol{D}_1)$ and $(\triangle_2, \boldsymbol{D}_2)$ the metric is defined as

$$\langle(\triangle_1, \boldsymbol{D}_1), (\triangle_2, \boldsymbol{D}_2)\rangle_A := \langle\triangle_1, \triangle_2\rangle + \lambda\langle\boldsymbol{R}^{-1}\boldsymbol{D}_1\boldsymbol{R}^{-1}, \boldsymbol{R}^{-1}\boldsymbol{D}_2\boldsymbol{R}^{-1}\rangle\,, \tag{4.12}$$

where $\langle\cdot,\cdot\rangle$ denotes the normal inner product and $\lambda \geq 0$ is a combination weight. The metric defined in Eqn (4.12) induces the following (squared) geodesic distance between $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{S}_+^d(p)$

$$\delta_g^2(\boldsymbol{A}, \boldsymbol{B}) = \|\Theta\|_F^2 + \lambda\|\log(\boldsymbol{R}_{\boldsymbol{A}}^{-1}\boldsymbol{R}_{\boldsymbol{B}}^2\boldsymbol{R}_{\boldsymbol{A}}^{-1})\|_F^2\,. \tag{4.13}$$

One can understand Eqn (4.13) as the sum of distances on $\mathcal{G}_d^p$ and $\mathcal{S}_{++}^p$. The first term refers to the squared geodesic distance between linear subspaces $\boldsymbol{U}_A$ and $\boldsymbol{U}_B$ while the second term is the squared geodesic distance between two SPD matrices $\boldsymbol{R}_A^2$ and $\boldsymbol{R}_B^2$. Moreover, the distance is invariant to the transformations that preserve angles (i.e., orthogonal transformations, scalings, and pseudoinversing) Bonnabel and Sepulchre [2009]. Here, our main motivation is to benefit from the manifold of SPSD matrices to overcome the limitations of the SPD manifolds in dealing with rank deficient matrices. As will become clear by our experiments, the induced geometry is more discriminative than both SPD and Grassmannian manifolds.

4.4.3 Kernels on SPSD Matrices

Though distances on SPSD manifolds can be used to measure similarities between image sets, the nonlinear structure of curved spaces (SPSD being an instance of) prohibits us from directly employing more involved machineries (e.g. discriminant analysis, large margin classification). A prominent way of getting around this difficulty is to make use of valid positive definite kernels on Riemannian manifolds Jayasumana et al. [2015]; Harandi et al. [2015a]. To define positive definite (*pd*) kernels on the SPSD manifold, we first introduce a negative definite (*nd*) function on $\mathcal{S}_+^d(p)$.

Theorem 5. *The function* $\delta^2 : \mathcal{S}_+^d(p) \times \mathcal{S}_+^d(p) \to \mathbb{R}_+$ *defined as*

$$\begin{aligned}\delta^2(\boldsymbol{A}, \boldsymbol{B}) &\triangleq \|\boldsymbol{U}_A\boldsymbol{U}_A^T - \boldsymbol{U}_B\boldsymbol{U}_B^T\|_F^2 + \lambda\|\log(\boldsymbol{R}_A) - \log(\boldsymbol{R}_B)\|_F^2 \\ &= 2p - 2\|\boldsymbol{U}_A^T\boldsymbol{U}_B\|_F^2 + \lambda\|\log(\boldsymbol{R}_A) - \log(\boldsymbol{R}_B)\|_F^2\,,\end{aligned} \tag{4.14}$$

is negative definite on $\mathcal{S}_+^d(p)$ *for* $\lambda \geq 0$.

Proof. We recall that a symmetric function $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ on a set $\mathcal{X}$ is *nd* if and only if $\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \leq 0$ for any $n \in \mathbb{N}$, $x_i \in \mathcal{X}$ and $c_i \in \mathbb{R}$ with $\sum_{i=1}^{n} c_i = 0$. As shown in Jayasumana et al. [2015], if $f : \mathcal{X} \to \mathcal{H}$ is a mapping from a set $\mathcal{X}$ to an inner product space $\mathcal{H}$, then the function $\|f(x_i) - f(x_j)\|_{\mathcal{H}}^2$ is negative definite for $\forall x_i, x_j \in \mathcal{X}$. Here $\| \cdot \|_{\mathcal{H}}$ denotes the norm in $\mathcal{H}$.

Now we note that $\pi_p : \mathcal{G}_d^p \to \mathrm{Sym}(d), \pi_p(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{X}^T$ is a mapping from the Grassmannian to the space of $d \times d$ symmetric matrices, hence the first term of Eqn. (4.14) is negative definite. Similarly, with $\log : \mathcal{S}_{++}^p \to \mathrm{Sym}(p)$, the second term of Eqn. (4.14) is negative definite. By invoking the definition of the negative definite kernels, it is easy to see that the summation of two negative definite kernels is also a negative definite kernel. □

Having an *nd* function at our disposal, we can make use of the following theorem to define a family of *pd* kernels on $\mathcal{S}_+^d(p)$.

Theorem 6 (Theorem 2.3 in Chapter 3 of Berg et al. [1984]). *Let μ be a probability measure on the half line $\mathbb{R}_+$ and $0 < \int_0^\infty t \mathrm{d}\mu(t) < \infty$. Let $\mathcal{L}_\mu$ be the Laplace transform of μ, i.e. $\mathcal{L}_\mu(s) = \int_0^\infty e^{-ts} \mathrm{d}\mu(t)$, $s \in \mathbb{C}_+$. Then, $\mathcal{L}_\mu(\beta f)$ is positive definite for all $\beta > 0$ if and only if $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ is negative definite.*

For example, by choosing μ to be the Dirac function at $t = 1$, we obtain the RBF kernel on $\mathcal{S}_+^d(p)$ as follows

$$k_R(\boldsymbol{A}, \boldsymbol{B}) \triangleq \exp\Big(-\beta\Big(\lambda \| \log(\boldsymbol{R}_A) - \log(\boldsymbol{R}_B)\|_F^2 - 2\|\boldsymbol{U}_A^T \boldsymbol{U}_B\|_F^2\Big)\Big)\,.$$

We notice that one could arrive to the same conclusion, i.e. $k_R(\cdot,\cdot)$ being *pd*, by observing that it is indeed the product of two *pd* kernels. However, our approach here is more principled and can be used to generate other types of *pd* kernels on $\mathcal{S}_+^d(p)$ by properly changing the measure μ in Thm.6.

Another widely used kernel in the Euclidean spaces is the Laplace kernel defined as $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\beta\|\boldsymbol{x} - \boldsymbol{y}\|)$. To obtain the Laplace kernel on the $\mathcal{S}_+^d(p)$, we make use of the following theorem for *nd* kernels.

Theorem 7 (Corollary 2.10 in Chapter 3 of Berg et al. [1984]). *If $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is negative definite and satisfies $f(\boldsymbol{x}, \boldsymbol{x}) \geqq 0$ then so is ψ^α for $0 < \alpha < 1$*

As a result both $\delta(\cdot,\cdot) = \sqrt{\delta^2(\cdot,\cdot)}$ is *nd* by choosing $\alpha = 1/2$ in Theorem 7 and hence the form of $\exp(-\beta\delta(\cdot,\cdot))$ is *pd*.

Table 4.2: The proposed SPSD kernels.

Kernel	**Equation**
Linear	$k_l(\boldsymbol{A},\boldsymbol{B}) = \|\boldsymbol{U}_A^T\boldsymbol{U}_B\|_F^2 + \lambda\,\mathrm{Tr}\left(\log(\boldsymbol{R}_A)\log(\boldsymbol{R}_B)\right)$
Polynomial	$k_p(\boldsymbol{A},\boldsymbol{B}) = \left(\beta + \|\boldsymbol{U}_A^T\boldsymbol{U}_B\|_F^2 + \lambda\,\mathrm{Tr}\left(\log(\boldsymbol{R}_A)\log(\boldsymbol{R}_B)\right)\right)^\alpha$
Laplace	$k_L(\boldsymbol{A},\boldsymbol{B}) = \exp\left(-\beta\sqrt{\lambda\|\log(\boldsymbol{R}_A)-\log(\boldsymbol{R}_B)\|_F^2-2\|\boldsymbol{U}_A^T\boldsymbol{U}_B\|_F^2}\right)$
RBF	$k_R(\boldsymbol{A},\boldsymbol{B}) = \exp\left(-\beta\left(\lambda\|\log(\boldsymbol{R}_A)-\log(\boldsymbol{R}_B)\|_F^2-2\|\boldsymbol{U}_A^T\boldsymbol{U}_B\|_F^2\right)\right)$

Before concluding this part, we also introduce the linear and polynomial kernels on $\mathcal{S}_+^d(p)$. The linear kernel $k_l(\boldsymbol{A},\boldsymbol{B}) = \|\boldsymbol{U}_A^T\boldsymbol{U}_B\|_F^2 + \lambda\,\mathrm{Tr}\left(\log(\boldsymbol{R}_A)\log(\boldsymbol{R}_B)\right)$ is interesting as it is a parameter-less kernel (discarding λ which defines the form of the linear combination of the two). To show that $k_l(\cdot,\cdot)$ is *pd*, we note that $k_l(\cdot,\cdot)$ is the summation of two *pd* kernels defined on the space of symmetric matrices Hamm and Lee [2008]; Jayasumana et al. [2015]. This will lead us to define the polynomial kernels as

$$k_p(\boldsymbol{A},\boldsymbol{B}) \triangleq \left(\beta + \|\boldsymbol{U}_A^T\boldsymbol{U}_B\|_F^2 + \lambda\,\mathrm{Tr}\left(\log(\boldsymbol{R}_A)\log(\boldsymbol{R}_B)\right)\right)^\alpha .$$

Table 4.2 summarizes all the aforementioned SPSD kernels.

4.4.4 Experiments

We present experiments on three benchmark image set classification tasks. Given the diversity of the studied problems, we deem to describe images differently based on the task in hand. However, please note that our goal is not to identify the best off the shelf image descriptors for each task.

In our experiments, we rely on two classifiers: **1.** a simple Nearest Neighbor (NN) classifier to demonstrate the benefits of the SPSD manifolds in comparison to SPD and Grassmann manifolds and **2.** an NN classifier on top of kernel Discriminant Analysis (kDA) to evaluate the positive definite kernels introduced in §4.4.3. Different algorithms tested in our experiments are referred to as

NN: NN classifier using the geodesic distance.

kDA$_{Linear}$: kDA classifier with linear kernel.

kDA$_{Polynomial}$: kDA classifier with polynomial kernel.

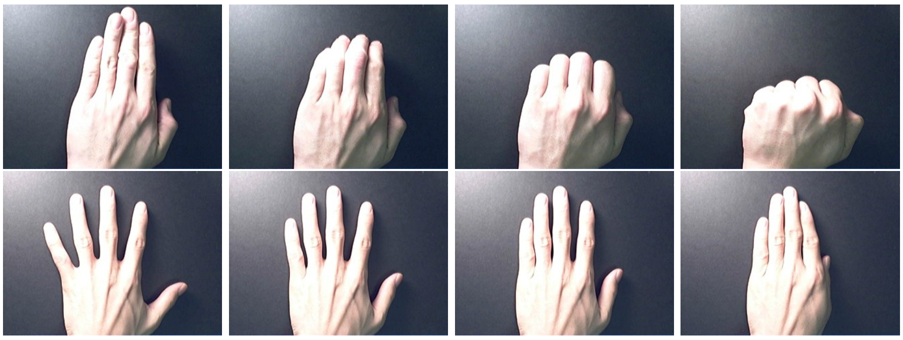

Figure 4.4: Examples of the Cambridge hand gesture dataset Kim et al. [2007b].

kDA$_{Laplace}$: kDA classifier with Laplace kernel.

kDA$_{RBF}$: kDA classifier with RBF kernel.

Before delving into details of each experiment, we note that the kernel parameters are found by cross-validating over training sets. As far as the sensitivity to the rank of matrices and the parameter λ are considered, we dedicate a separate section (§4.4.4.4) before concluding this part.

4.4.4.1 Hand Gesture Recognition

In our first experiment, we tackled the task of hand gesture classification from image sequences. To this end, we used the Cambridge hand gesture dataset Kim et al. [2007b] which contains 900 image sets of 9 gesture classes with large intra-class variations. The gestures are defined by 3 primitive hand shapes and 3 primitive motions (see Fig. 4.4 for examples). Therefore, the target task for this data set is to classify different shapes as well as different motions at a time.

We followed the experimental protocol suggested by Mahmood et al. [2014] in which 100 image sets of each class are divided into two parts, 81-100 used as train set and 1-80 as test set. For this dataset we made use of concatenated HOG features of 2×2 blocks of each frame. The state-of-the-art on this dataset Mahmood et al. [2014] obtains the accuracy score of 83.1% using an ensemble of 9 spectral classifiers.

Table 4.3 shows that all the proposed methods comfortably outperform the state-of-the-art algorithms. A kDA classifier when the kernel is RBF over the SPSD manifold significantly outperforms the state-of-the-art ensemble of classifiers Mahmood et al. [2014]. The difference is 8 percentage points.

Figure 4.5: Examples of the YouTube celebrities dataset Kim et al. [2008].

Figure 4.6: Examples of the Maryland dynamic scene dataset Shroff et al. [2010].

Table 4.3: Recognition scores for the Cambridge hand gesture dataset Kim et al. [2007b].

SANP	22.5 Hu et al. [2011]
CDL	73.4 Wang et al. [2012]
SSSC	83.1 Mahmood et al. [2014]
NN	87.4
kDA$_{RBF}$	**91.1**
kDA$_{Laplace}$	89.3
kDA$_{Polynomial}$	90.0
kDA$_{Linear}$	90.0

We conducted extra experiments using the regularized SPD matrices with the same classifiers. The recognition accuracies are 82.5%, 87.5%, 86.7%, 87.8%, and 78.2% using **NN**, **kDA**$_{RBF}$, **kDA**$_{Laplace}$, **kDA**$_{Polynomial}$, and **kDA**$_{Linear}$, respectively (see Li et al. [2013a] for more details about the kernels). Please note that all the utilized kernels are SPD. This clearly shows that the proposed geometry is significantly superior to that of SPD manifolds. Since the same trend is observed, we confine ourselves to report only the results of SPSD matrices for other datasets.

Table 4.4: Recognition scores for the YouTube celebrities Kim et al. [2008].

SANP	65.0 Hu et al. [2011]
CDL	70.1 Wang et al. [2012]
ADNT	71.4 Hayat et al. [2014]
NN	65.3
kDA$_{RBF}$	**72.8**
kDA$_{Laplace}$	71.8
kDA$_{Polynomial}$	70.6
kDA$_{Linear}$	70.4

4.4.4.2 Video-Based Face Recognition

We performed another experiment to classify human faces in videos. To this end, we considered the YouTube celebrity dataset Kim et al. [2008] which contains 1910 videos of 47 people (see Fig. 4.5 for a few examples). The large diversity of poses, illumination, and facial expressions in addition to high compression ratio of face images have made it the most challenging dataset for image set classification based face recognition.

For our evaluation, we followed the standard five-fold cross validation protocol used in Hu et al. [2011]; Wang et al. [2012]; Hayat et al. [2014] which divides the whole dataset equally (with minimum overlap) into five folds with 9 videos per subject in each fold. Three of the videos were randomly selected for training, while the remaining six were used for testing. We generated linear subspaces of order 6 by grouping features of individual frames.

From each video, we extracted the face regions using the tracker of Ross et al. [2008]. We considered Local Binary Patterns Ojala et al. [2002] as our feature. Each face region was divided into 2×2 distinct non-overlapping blocks and the features were extracted for each patch and concatenated to form the final frame descriptors. Therefore, each descriptor belongs to $\mathcal{S}_{++}^{6}$ and $\mathcal{G}_{232}^{6}$ for the covariance descriptors and linear subspaces.

Table 4.4 summarizes the average recognition rates of all the studied methods. Several conclusion can be drawn here. First of all, we note that in all cases the new SPSD manifold achieves descent accuracy scores. Furthermore, a single RBF kernel in the SPSD manifold comfortably outperform all the state-of-the-art algorithms. We achieve average accuracy score of 72.8% which outperforms the closest competitor by 1.4% percentage points.

4.4.4.3 Dynamic Scene Recognition

Finally, we considered the task of scene recognition from the videos using the Maryland "In-The-Wild"dataset Shroff et al. [2010] (see Fig. Kim et al. [2007b] for example classes). This

Table 4.5: Recognition accuracies for the Maryland dataset Shroff et al. [2010].

SFA	60.0 Theriault et al. [2013]
CSO	67.7 Feichtenhofer et al. [2013]
BoSE	77.7 Feichtenhofer et al. [2014]
NN	83.1
$\textbf{kDA}_{RBF}$	88.5
$\textbf{kDA}_{Laplace}$	82.3
$\textbf{kDA}_{Polynomial}$	**90.0**
$\textbf{kDA}_{Linear}$	89.2

dataset consists of 130 videos of natural scenes spanning 13 categories (e.g. Avalanche, Forest Fire, Waves) with 10 videos per class. The videos are collected from Internet-based video hosting sites, such as YouTube. Significant camera motions, differences in appearance, frame rate, scale, viewpoint, scene cuts, and illumination conditions exist in this dataset. A leave-one-video-out experimental protocol is used for consistency with previous evaluation in Feichtenhofer et al. [2014].

We made use of the FC7 features of Convolutional Neural Network (CNN) of Zhou et al. [2014]. The network is trained on the Places dataset Zhou et al. [2014] with 205 scene categories and 2,5 million images with a category label. Here, we extract the 4096 FC7 feature of each frame. We then reduce the dimension of the feature to 400 using Principal Component Analysis.

Results are reported in Table 4.5. The table is self explanatory. To the best of our knowledge, 77.7% classification accuracy by the recent Bag of Spatiotemporal Energy (BoSE) method of Feichtenhofer et al. [2014] is the highest accuracy score reported on this dataset. Our methods outperform the BoSE by a very large support.

4.4.4.4 Sensitivity to Rank and Weighting Parameter

We also studied the sensitivity of our proposed approach to the chosen subspace order as well as the value of λ. Figure 4.7 shows the accuracy against subspace order for the Cambridge hand gesture dataset using the pixel intensities as features and NN as classifier (i.e. using Eqn 4.13). As depicted in the figure for all the studied subspace order the accuracy of NN on the Grassmannian manifold is inferior to the SPSD cases.

More importantly, we observed that as the order of the subspaces increases the differences between the accuracy obtained on the Grassmannian drops significantly. In other words, most values of the parameter λ provides a consistently stable performance over a range of p values even if the number of subspaces varies considerably. This clearly justifies the use of SPSD

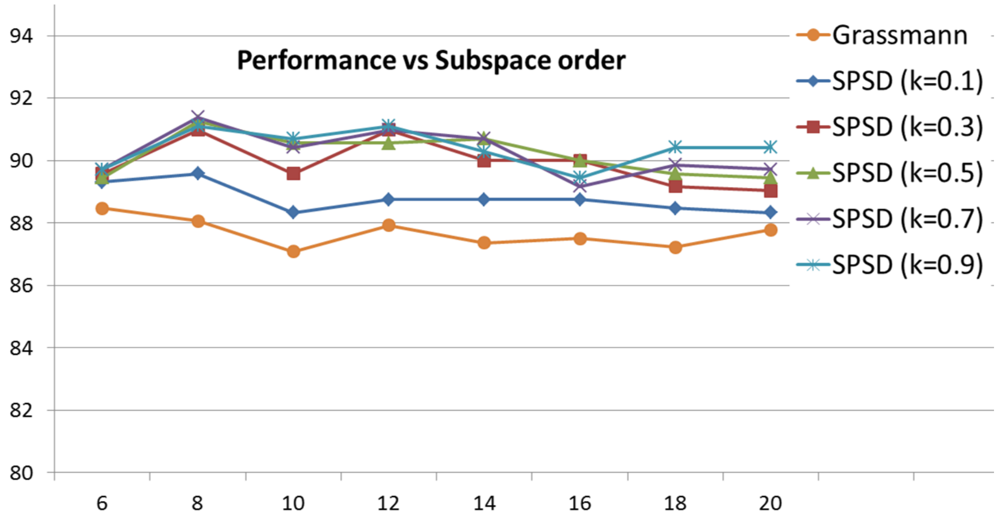

Figure 4.7: Accuracy against subspace order for the Cambridge hand gesture dataset. As visible inclusion of the SPD term significantly improves upon the use of Grassmannian only.

matrices.

4.5 Conclusions

In this chapter, we studied two special types of RCovDs, namely infinite-dimensional RCovDs and SPSD matrices which readily do not conform the usual development for SPD matrices. Firstly, we made use of random Fourier feature and the Nyström method to compute two approximations to infinite-dimensional RCovDs. Our experimental evaluation has demonstrated that the proposed RCovDs significantly outperform the low-dimensional ones on image classification task. More importantly, our RCovDs provide a framework in which the well-known Riemannian geometry of the SPD matrices can be taken into account.

Secondly, inspired by the recent success of image set representation as points on nonlinear Riemannian manifolds, we proposed SPSD matrices as descriptors for image set classification. The challenge lies in the fact that to measure the similarities, the usual metrics on the manifold of SPD matrices, such as the AIRM, are not valid due to rank-deficiency of the SPSD matrices. Hence, our main motivation to benefit from the SPSD matrices is to overcome the limitations of the SPD manifolds (rank deficiency being the most important one). We made use of a metric that can be decomposed as sum of infinitesimal distances on the Grassmannian manifold and the manifold of SPD matrices. Since our formulation enables us to utilize any distances on the two manifolds, we can integrate valid kernels for the image set classification task. A rigorous set of successful experiments on several challenging applications including gesture classification, video-based face recognition and dynamic scene recognition demonstrated the advantages of our method.

In the next chapter, we study how Riemannian optimization techniques assist us in finding solutions for metric learning.

Chapter 5

Metric Learning, A Riemannian Manifold Perspective

5.1 Overview

In this chapter, we first devise a kernel version of the recently introduced Keep It Simple and Straightforward MEtric learning (KISSME) Koestinger et al. [2012] method, hence adding a novel dimension to its applicability in scenarios where input data is non-linearly distributed. To this end, we make use of the infinite dimensional covariance matrices and show how a matrix in a Reproducing Kernel Hilbert Space (RKHS) can be projected onto the positive cone efficiently. In particular, we propose two techniques towards projecting on the positive cone in an RKHS. The first method, though approximating the solution, enjoys a closed-form and analytic formulation. The second solution is more accurate and requires Riemannian optimization techniques. Nevertheless, both solutions can scale up very well as our empirical evaluations suggest. For the sake of completeness, we also employ the Nyström method to approximate an RKHS before learning a metric. Our experiments evidence that, compared to the state-of-the-art metric learning algorithms, working directly in RKHS, leads to more robust and better performances Faraki et al. [2017b].

Furthermore, we devise a unified formulation for joint dimensionality reduction and metric learning based on the KISSME algorithm. Despite its attractive properties, the performance of the KISSME method is greatly dependent on Principal Component Analysis (PCA) as a preprocessing step. This dependency can lead to difficulties, e.g., when the dimensionality is not meticulously set. Our joint formulation is expressed as an optimization problem on the Grassmann manifold, hence enjoys properties of Riemannian optimization techniques.

Finally, following the success of deep learning in recent years, we also devise end-to-end learning of a generic deep network for metric learning using our derivation [Faraki et al., 2017a].

5.2 Introduction

In computer vision, determining a suitable metric plays a pivotal role in various applications such as person reidentification Xiong et al. [2014]; Chen et al. [2015]; Zheng et al. [2015b]; Cheng et al. [2011], face and kinship verification Koestinger et al. [2012]; Li et al. [2013b]; Lu et al. [2014]; Guillaumin et al. [2009]; Wolf et al. [2011], and image retrieval Song et al. [2016]; Hoi et al. [2006], to name a few. The commonly used Euclidean distance assumes that all features are of equal importance, which is almost never the case in practice.

On a related note, metric learning algorithms Weinberger et al. [2005]; Davis et al. [2007]; Koestinger et al. [2012]; Harandi et al. [2017] are of practical interest when learning from large number of categories (with limited training samples per category) is deemed, if the machinery is meant to deal with unseen classes (e.g., retrieval), or if weaker forms of supervision are considered. In such scenarios, conventional classification approaches are either not applicable or may fail miserably.

The KISSME algorithm Koestinger et al. [2012] is agnostic to the class labels and learns a metric purely from a set of equivalence constraints (similar/dissimilar pairs). Furthermore, the algorithm scales to large scale problems, making it a suitable -if not perfect- match for the aforementioned problems. To be more specific, the Mahalanobis distance is learned by one sweep over the data with the dominant computation being an eigenvalue decomposition. Given its attractiveness, we base our contributions on the KISSME method. We will discuss KISSME in detail in §5.4, but before that we review some notable examples of metric learning techniques below.

5.3 Related Work

We review some notable examples of conventional and deep metric learning techniques here. We start by two studies in the restricted mode and follow it up by a classical method devised for the unrestricted case. Note that that algorithms in restricted metric learning scenario do not have access to the class labels of the samples. These algorithms can also work in the unrestricted scenario while the opposite is not often the case. This makes the restricted algorithms more appealing as they can address a broader range of problems.

Mahalanobis Metric for Clustering (MMC) Xing et al. [2003] aims to minimize sum of distances over similar pairs while ensuring dissimilar pairs are far apart. The problem is formulated as an iterative gradient descent algorithm where at each iteration the obtained solution is projected back to the set of Positive Semi-Definite (PSD) matrices to ensure the metric is proper. Projection onto the PSD cone requires eigenvalue decomposition, making MMC computationally expensive when dealing with high-dimensional data.

Pairwise Constrained Component Analysis (PCCA) Mignon and Jurie [2012] learns a transformation to project similar pairs inside a ball while dissimilar pairs are pushed away. Optimization is performed by making use of the gradient descent method.

A goal common to the state-of-the-art metric learning techniques is to make use of discriminative information existing in training data. Neighborhood Component Analysis (NCA) Goldberger et al. [2004] learns a Mahalanobis distance to improve k-Nearest Neighbor (kNN) classification score in a supervised manner. To this end, NCA minimizes the expected value of a stochastic variant of the kNN error. The classification model is parameter free, without any assumptions about the shape of the class distributions or the boundaries between them, which makes NCA attractive and easy to use.

Large Margin Nearest Neighbor (LMNN), learns a global linear transformation of labeled input data to improve the kNN classification accuracy Weinberger and Saul [2009]. In doing so, the learned transformation (or equivalently the metric) is deemed to unite the k-nearest neighbors of each point sharing the same label while separating instances from different classes by a margin. Learning the linear transformation is formulated as a semi-definite programming problem and solved by iterating between a gradient descent step followed by projecting the solution onto the positive semi-definite cone.

Davis et al., leverage on the connection between the multivariate Gaussian distributions and the Mahalanobis metrics in their Information-Theoretic Metric Learning (ITML) method Davis et al. [2007]. The method seeks a metric to enforce the distance between similar pairs to be below the threshold δ_l while making the distance between dissimilar pairs exceeding the threshold δ_u with $\delta_l < \delta_u$. In ITML, the proximity between two Mahalanobis metrics is measured by the Kullback-Leibler divergence of their corresponding distributions.

Guillaumin et al. [2009] propose Logistic Discriminant Metric Learning (LDML) to tackle the problem of face verification. The key idea is to find a metric to make the distances between similar pairs smaller than the distances between dissimilar pairs. Thereby, a probabilistic estimate depicting whether a pair of face images belong to the same person or not is obtained using the Mahalanobis distance along a linear logistic discriminant model. The Mahalanobis metric is obtained by maximizing the log-likelihood of the logistic model.

In recent years, deep metric learning has received growing attention, following the trend of deep Convolutional Neural Networks (CNN) in solving large-scale classification problems Krizhevsky et al. [2012].

5.3.1 Metric Learning and Deep Nets

Similarity/distance metric learning using deep nets originates from the advent of Siamese networks Chopra et al. [2005]. In recent years, deep metric learning has received growing attention, following the trend of deep CNNs in solving large-scale classification problems. For example, in the spirit of PCCA, a two layer discriminative network for face verification is proposed in Hu et al. [2014]. This method is further extended in Lu et al. [2017] by collaboratively learning multiple neural networks. Another work in multiple metric learning is Duan et al. [2017] where multiple holistic and local subspaces are learned using Auto Encoders (AE). In order to train the local networks, training samples are first assigned to their nearest AE based on their reconstruction losses. Sun et al. [2014] uses an ensemble of networks, each operating on a different face patch for face verification. The networks are trained by making use of a combination of classification and verification cost functions. The cross-entropy loss is used for the classification loss. As for the verification loss, cost function encodes an l_2-margin between face images.

Very recent works in deep metric learning include the work of Hoffer and Ailon [2015] and Schroff et al. [2015] in which an LMNN based triplet loss layer is used to direct CNN parameter learning. Song et al. [2016] shows careful construction of batches such as including hard triplets during training, leads to better clustering and retrieval qualities. Huang et al. [2016] proposes a formulation to jointly perform similarity learning and hard sample selection. Finally, Hermans et al. [2017] shows promising results for the task of person re-identification using a variant of triplet loss function (see Eq. (5.31)).

Song et al. [2016] discuss drawbacks of pairwise constraints and triplets when combined with stochastic gradient descent updates in deep nets. In short, given the small size of batches, the full potential of pairwise or triplet information cannot be exploited in deep nets. As suggested in Song et al. [2016], careful construction of batches by the concept of lifted structured feature embedding, i.e., including hard triplets during training, leads to significant improvement in accuracy.

5.4 Background

We have already defined the notion of metric and the induced space in § 2. Let $\mathcal{X}$ be the set and $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$ the distance function. Choosing $\mathcal{X}$ to be the D-dimensional Euclidean space, the class of Mahalanobis distances can be defined as

$$d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^T \boldsymbol{M} (\boldsymbol{x}-\boldsymbol{y})}\,, \tag{5.1}$$

with $\boldsymbol{M} \in \mathcal{S}_{++}^D$.

The goal of Mahalanobis Metric Learning (MML) is to determine $\boldsymbol{M}$ such that $d_{\boldsymbol{M}}(\cdot,\cdot)$ endows certain useful properties. Let $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n$, $\boldsymbol{x}_i, \boldsymbol{y}_i \in \mathbb{R}^D$ be a set of n training pairs. Furthermore, let $l_i \in \{-1, 1\}$ denote the label of the i-th pair with $l_i = 1$ indicating that the corresponding pair is similar and $l_i = -1$ otherwise.

In KISSME algorithm, which our methods are built upon, a dissimilarity hypothesis is defined as

$$\delta(\boldsymbol{x}_i, \boldsymbol{y}_i) = log \left(\frac{\frac{1}{\sqrt{2\pi|\Sigma_d|}} \exp\Big(-\frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{y}_i)^T \Sigma_d^{-1} (\boldsymbol{x}_i - \boldsymbol{y}_i) \Big)}{\frac{1}{\sqrt{2\pi|\Sigma_s|}} \exp\Big(-\frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{y}_i)^T \Sigma_s^{-1} (\boldsymbol{x}_i - \boldsymbol{y}_i) \Big)} \right) \,, \tag{5.2}$$

where

$$\Sigma_d = \frac{1}{\#(l_i = -1)} \sum_{i, l_i = -1} (\boldsymbol{x}_i - \boldsymbol{y}_i)(\boldsymbol{x}_i - \boldsymbol{y}_i)^T, \tag{5.3}$$

$$\Sigma_s = \frac{1}{\#(l_i = 1)} \sum_{i, l_i = 1} (\boldsymbol{x}_i - \boldsymbol{y}_i)(\boldsymbol{x}_i - \boldsymbol{y}_i)^T \,. \tag{5.4}$$

where # denotes the number of samples.

Having a large $\delta(\boldsymbol{x}_i, \boldsymbol{y}_i)$ indicates that $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are dissimilar, and vice-versa. With this hypothesis, the Mahalanobis matrix is obtained as $\boldsymbol{M} = \mathrm{Proj}(\Sigma_s^{-1} - \Sigma_d^{-1})$ with $\mathrm{Proj}(\cdot)$ denoting projection to the cone of positive definite matrices. Such a projection is required to have a valid distance. In KISSME, the projection is obtained by clipping the spectrum of $\Sigma_s^{-1} - \Sigma_d^{-1}$. That is given the eigen-decomposition of $\Sigma_s^{-1} - \Sigma_d^{-1}$ as $\boldsymbol{UDU}^T$ then $\boldsymbol{M} = \boldsymbol{UD}_+\boldsymbol{U}^T$ where $\boldsymbol{D}_+ = \mathrm{diag}(\max(d_i, \epsilon))$ with $\boldsymbol{D} = \mathrm{diag}(d_i)$ and ϵ being a very small positive number.

5.5 KISSME in Hilbert Spaces

On the downside, the KISSME algorithm is designed to work with explicit and vectorized data. As such, the algorithm is unable to learn efficiently from non-linear data or if data is not in vector form (e.g., manifold-value data). This is also evidenced by some recent studies (e.g., Xiong et al. [2014]), stating that non-linearity associated with high-dimensional data cannot be captured by the KISSME algorithm. As a result, the algorithm falls short compared to the methods that are efficiently benefiting from such information. In this part, we provide solutions to both limitations in a principal way and present techniques to kernelize KISSME, making it applicable to a wider set of problems.

To kernelize KISSME algorithm while preserving its unique features, we make use of the recently introduced infinite dimensional covariance matrices Harandi et al. [2014a]; Quang et al. [2014]; Faraki et al. [2015a] and show how a matrix in an RKHS can be projected onto the positive cone efficiently. In particular, we propose two techniques towards projecting onto the positive cone in an RKHS. The first method, albeit approximating the solution, enjoys a closed-form and analytic formulation. The second solution is more accurate and requires Riemannian optimization techniques. Nevertheless, both solutions can scale up very well as our empirical evaluations suggest. Furthermore, to have the full package, we employ the Nyström method Baker [1977] to approximate an RKHS and formulate the Nyström KISSME accordingly.

In our experiments, we demonstrate the benefits of the presented kernelized KISSME approach over existing metric learning schemes on the task of person reidentification using the iLIDS Zheng et al. [2009] and the CAVIAR Cheng et al. [2011] datasets and kinship verification from unconstrained face images using the KinFace-I and the KinFace-II datasets Lu et al. [2014]. Before delving into more details, we emphasize that our method learns a metric purely from the equivalence constraints (similar/dissimilar pairs) and does not use class-labels as required by some other learning techniques (e.g., Song et al. [2016]; Ding et al. [2015]; Xiong et al. [2014]).

We now describe our approach to learning a Mahalanobis metric $\boldsymbol{M}_{\mathcal{H}}$ in an RKHS $\mathcal{H}$ using the KISSME algorithm. Let $\mathcal{X}$ and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a set and a (positive definite) pd kernel defined on $\mathcal{X}$, respectively. According to the Mercer theorem, a mapping $\phi : \mathcal{X} \to \mathcal{H}$ to an RKHS $\mathcal{H}$ exists for any pd kernel. Our aim in this section is to derive a Mahalanobis distance $d_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}_{+}$ in the feature space $\mathcal{H}$ with certain properties. Suppose $\{(\boldsymbol{x}_i, \boldsymbol{y}_i, l_i)\}_{i=1}^{n}$ with $\boldsymbol{x}_i, \boldsymbol{y}_i \in \mathcal{X}$ and $l_i \in \{-1, 1\}$ be a set of n training samples. Given a pd kernel k :

$\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ the Mahalanobis distance in $\mathcal{H}$ can be written as

$$d_{\mathcal{H}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \sqrt{\big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i)\big)^T \boldsymbol{M}_{\mathcal{H}} \big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i)\big)}\ . \tag{5.5}$$

To learn $\boldsymbol{M}_{\mathcal{H}}$, we define the likelihood ratio test of the pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ as

$$\begin{aligned} &\delta_{\mathcal{H}}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \\ &log \left(\frac{\frac{1}{\sqrt{2\pi|\Sigma_{\mathcal{H},d}|}} \exp\Big(-\frac{1}{2}\big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i)\big)^T \Sigma_{\mathcal{H},d}^{-1} \big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i)\big)\Big)}{\frac{1}{\sqrt{2\pi|\Sigma_{\mathcal{H},s}|}} \exp\Big(-\frac{1}{2}\big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i)\big)^T \Sigma_{\mathcal{H},s}^{-1} \big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i)\big)\Big)} \right)\ . \end{aligned} \tag{5.6}$$

Here, the covariance matrices are

$$\begin{aligned} \Sigma_{\mathcal{H},d} &= \frac{1}{\#(l_i = -1)} \sum_{i, l_i = -1} \big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i)\big)\big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i)\big)^T, \\ \Sigma_{\mathcal{H},s} &= \frac{1}{\#(l_i = 1)} \sum_{i, l_i = 1} \big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i)\big)\big(\phi(\boldsymbol{x}_i) - \phi(\boldsymbol{y}_i)\big)^T\ . \end{aligned} \tag{5.7}$$

With the same line of reasoning as Koestinger et al. [2012], the Mahalanobis form that maximizes $\delta_{\mathcal{H}}(\cdot, \cdot)$ over the training samples is obtained by choosing $\boldsymbol{M}_{\mathcal{H}} = \mathrm{Proj}_{\mathcal{H}}(\Sigma_{\mathcal{H},s}^{-1} - \Sigma_{\mathcal{H},d}^{-1})$. As such, we need to answer the following questions to extend the KISSME algorithm to work in $\mathcal{H}$:

1. *How $\Sigma_{\mathcal{H},s}^{-1}$ and $\Sigma_{\mathcal{H},d}^{-1}$ can be obtained in $\mathcal{H}$?*

2. *How the projection $\mathrm{Proj}_{\mathcal{H}}(\cdot)$ can be defined efficiently in $\mathcal{H}$?*

3. *Having answers to the previous questions at our disposal, how $d_{\mathcal{H}}(\cdot, \cdot)$ can be obtained efficiently $\mathcal{H}$?*

Below, we address these questions one-by-one.

5.5.1 Obtaining $\Sigma_{\mathcal{H},s}^{-1}$ and $\Sigma_{\mathcal{H},d}^{-1}$

In essence, obtaining $\Sigma_{\mathcal{H},s}^{-1}$ and $\Sigma_{\mathcal{H},d}^{-1}$ follow the same procedure. For the sake of simplicity, we describe how in general the inverse of a covariance matrix, namely $\Sigma_{\mathcal{H}}^{-1}$ in the RKHS $\mathcal{H}$, can be obtained. In doing so, we start with the familiar Euclidean space. Given a set of pairs

$\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, we have

$$\Sigma = \frac{1}{n} \sum_i (\boldsymbol{x}_i - \boldsymbol{y}_i)(\boldsymbol{x}_i - \boldsymbol{y}_i)^T = \boldsymbol{Z}\boldsymbol{J}\boldsymbol{J}^T\boldsymbol{Z}^T \;, \tag{5.8}$$

with $\boldsymbol{Z} = \left[\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n, \boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n\right]$ and

$$\boldsymbol{J}\boldsymbol{J}^T = \frac{1}{n} \begin{bmatrix} \mathbf{I}_n & -\mathbf{I}_n \\ -\mathbf{I}_n & \mathbf{I}_n \end{bmatrix} \;.$$

Accordingly, the covariance matrix $\Sigma_{\mathcal{H}}$ in the RKHS $\mathcal{H}$ with dimensionality $|\mathcal{H}|$ can be written as

$$\Sigma_{\mathcal{H}} = \Phi_{\boldsymbol{Z}}\boldsymbol{J}\boldsymbol{J}^T\Phi_{\boldsymbol{Z}}^T, \tag{5.9}$$

with $\Phi_{\boldsymbol{Z}} = \Big[\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \cdots, \phi(\boldsymbol{x}_n), \phi(\boldsymbol{y}_1), \phi(\boldsymbol{y}_2), \cdots, \phi(\boldsymbol{y}_n)\Big]$.

The difficulty in obtaining $\Sigma_{\mathcal{H}}^{-1}$ lies in the fact that for universal kernels (e.g., Gaussian kernel) the dimensionality of $\mathcal{H} \to \infty$. With limited data, $\Sigma_{\mathcal{H}}$ is positive semi-definite and hence $\Sigma_{\mathcal{H}}^{-1}$ does not theoretically exist. As such, we need to preserve the positive eigenvalues and the associated eigenvectors of $\Sigma_{\mathcal{H}}$ and regularize the zero ones. This can be understood as the best approximation to $\Sigma_{\mathcal{H}}$ given the set $\boldsymbol{Z}$.

In particular, let $\mathbb{K}_{\boldsymbol{Z}} \in \mathbb{R}^{2n \times 2n}$ be the kernel matrix of $\boldsymbol{Z}$, i.e.,

$$[\mathbb{K}_{\boldsymbol{Z}}]_{i,j} = \begin{cases} k(\boldsymbol{x}_i, \boldsymbol{x}_j), & i, j \leq n \\ k(\boldsymbol{y}_i, \boldsymbol{y}_j), & i, j > n \\ k(\boldsymbol{x}_i, \boldsymbol{y}_j), & \text{otherwise} \end{cases}$$

Let the SVD decomposition of $\boldsymbol{J}^T\Phi_{\boldsymbol{Z}}^T\Phi_{\boldsymbol{Z}}\boldsymbol{J} = \boldsymbol{J}^T\mathbb{K}_{\boldsymbol{Z}}\boldsymbol{J}$ be $\boldsymbol{V}_{\boldsymbol{Z}}\Lambda_{\boldsymbol{Z}}\boldsymbol{V}_{\boldsymbol{Z}}^T$. Then, we make use of the relationship between the eigenvalues and eigenvectors of the product $\boldsymbol{A}\boldsymbol{A}^T$ and $\boldsymbol{A}^T\boldsymbol{A}$ where $\boldsymbol{A} = \Phi_{\boldsymbol{Z}}\boldsymbol{J}$. The regularized estimate of $\Sigma_{\mathcal{H}}$ then can be written Harandi et al. [2014a]

$$\hat{\Sigma}_{\mathcal{H}} = \Phi_{\boldsymbol{Z}}\boldsymbol{W}_{\boldsymbol{Z}}\boldsymbol{W}_{\boldsymbol{Z}}^T\Phi_{\boldsymbol{Z}}^T + \rho\mathbf{I}_{\mathcal{H}} \;, \tag{5.10}$$

where $\boldsymbol{W}_{\boldsymbol{Z}} = \boldsymbol{J}\boldsymbol{V}_{\boldsymbol{Z}}\big(\mathbf{I}_{2n} - \rho\Lambda_{\boldsymbol{Z}}^{-1}\big)^{0.5}$ with ρ being a positive regularizor.

To obtain $\hat{\Sigma}_{\mathcal{H}}^{-1}$, we make use of the Woodbury matrix identity Golub and Van Loan [2012] to arrive at

$$\hat{\Sigma}_{\mathcal{H}}^{-1} = \left(\Phi_{\mathbf{Z}} \boldsymbol{W}_{\mathbf{Z}} \boldsymbol{W}_{\mathbf{Z}}^T \Phi_{\mathbf{Z}}^T + \rho \mathbf{I}_{\mathcal{H}}\right)^{-1} = \frac{1}{\rho}\mathbf{I}_{\mathcal{H}} - \frac{1}{\rho}\Phi_{\mathbf{Z}} \boldsymbol{W}_{\mathbf{Z}} \Lambda_{\mathbf{Z}}^{-1} \boldsymbol{W}_{\mathbf{Z}}^T \Phi_{\mathbf{Z}}^T . \tag{5.11}$$

This lets us answer the first question, i.e., obtaining $\Sigma_{\mathcal{H},s}^{-1} - \Sigma_{\mathcal{H},d}^{-1}$ as

$$\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1} = \frac{1}{\rho}\Phi_{\mathbf{Z}_d} \boldsymbol{W}_{\mathbf{Z}_d} \Lambda_{\mathbf{Z}_d}^{-1} \boldsymbol{W}_{\mathbf{Z}_d}^T \Phi_{\mathbf{Z}_d}^T - \frac{1}{\rho}\Phi_{\mathbf{Z}_s} \boldsymbol{W}_{\mathbf{Z}_s} \Lambda_{\mathbf{Z}_s}^{-1} \boldsymbol{W}_{\mathbf{Z}_s}^T \Phi_{\mathbf{Z}_s}^T . \tag{5.12}$$

5.5.2 Projection onto the Positive Cone in $\mathcal{H}$

We note that the form of $\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1}$ cannot be directly used to define a Mahalanobis distance in $\mathcal{H}$. This is because the difference of two positive definite matrices is not necessarily positive definite, violating the very basic definition of a metric given in §5.4.

In this part, we propose two methods to project $\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1}$ onto the positive cone in $\mathcal{H}$. In the first method, though being an approximation, the projection can be obtained in closed-form. The second method relies on Riemannian optimization techniques and is an iterative scheme. Our experiments suggest that the solution obtained by the second method is more reliable. As such, we recommend to use the first solution only if the burden of Riemannian optimization techniques is a concern.

Our main idea here is to define an implicit form of a positive definite matrix and then minimize a measure of similarity between the implicit form and $\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1}$. More specifically, with $\boldsymbol{C} \in \mathcal{S}_{++}^n$ and trn denoting a set of n training vectors, we propose to solve the following problem as a means of projection onto the cone of positive definite matrices in $\mathcal{H}$

$$\arg\min_{\boldsymbol{C} \succ 0} \mathcal{L}(\boldsymbol{C}) \triangleq \left\| \Phi_{trn} \boldsymbol{C} \Phi_{trn}^T + \Phi_{\mathbf{Z}_s} \boldsymbol{A}_s \Phi_{\mathbf{Z}_s}^T - \Phi_{\mathbf{Z}_d} \boldsymbol{A}_d \Phi_{\mathbf{Z}_d}^T \right\|_F^2 , \tag{5.13}$$

where $\boldsymbol{A}_s = \boldsymbol{W}_{\mathbf{Z}_s} \Lambda_{\mathbf{Z}_s}^{-1} \boldsymbol{W}_{\mathbf{Z}_s}^T$ and $\boldsymbol{A}_d = \boldsymbol{W}_{\mathbf{Z}_d} \Lambda_{\mathbf{Z}_d}^{-1} \boldsymbol{W}_{\mathbf{Z}_d}^T$.

Expanding the Frobenious norm and considering only the terms that include $\boldsymbol{C}$, we get

$$\begin{aligned} \mathcal{L}(\boldsymbol{C}) =& \operatorname{Tr}\left(\mathbb{K}_{trn} \boldsymbol{C} \mathbb{K}_{trn} \boldsymbol{C}\right) + 2\operatorname{Tr}\left(\boldsymbol{K}_{\mathbf{Z}_s,trn} \boldsymbol{C} \boldsymbol{K}_{\mathbf{Z}_s,trn}^T \boldsymbol{A}_s\right) \\ &- 2\operatorname{Tr}\left(\boldsymbol{K}_{\mathbf{Z}_d,trn} \boldsymbol{C} \boldsymbol{K}_{\mathbf{Z}_d,trn}^T \boldsymbol{A}_d\right) + const\ . \end{aligned} \tag{5.14}$$

5.5.2.1 First Solution (The Approximation)

Without considering the constraint $\boldsymbol{C} \succ 0$, a closed-form solution can be obtained by setting as

$$\begin{aligned}
&\nabla_{\boldsymbol{C}}\big(\mathcal{L}(\boldsymbol{C})\big) = 0 \qquad\qquad (5.15)\\
&\Rightarrow 2\mathbb{K}_{trn}\boldsymbol{C}\mathbb{K}_{trn} + 2\boldsymbol{K}_{\boldsymbol{Z}_s,trn}^T\boldsymbol{A}_s\boldsymbol{K}_{\boldsymbol{Z}_s,trn} - 2\boldsymbol{K}_{\boldsymbol{Z}_d,trn}^T\boldsymbol{A}_d\boldsymbol{K}_{\boldsymbol{Z}_d,trn} = 0\\
&\Rightarrow \boldsymbol{C}^* = \mathbb{K}_{trn}^{-1}\Big(\boldsymbol{K}_{\boldsymbol{Z}_d,trn}^T\boldsymbol{A}_d\boldsymbol{K}_{\boldsymbol{Z}_d,trn} - \boldsymbol{K}_{\boldsymbol{Z}_s,trn}^T\boldsymbol{A}_s\boldsymbol{K}_{\boldsymbol{Z}_s,trn}\Big)\mathbb{K}_{trn}^{-1}\,.
\end{aligned}$$

Unlike $\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1}$, which is implicit, $\boldsymbol{C}^*$ has an explicit form. As such, projecting onto the set of positive definite matrices can be attained by simply applying the $\mathrm{Proj}(\cdot)$ operator (see §5.4). We note that the proposed two step approach (minimizing followed by projection) does not necessarily provide the closest point inside the positive cone to $\hat{\Sigma}_{s,\mathcal{H}}^{-1} - \hat{\Sigma}_{d,\mathcal{H}}^{-1}$, hence the name approximation. In our experiments, we refer to this method as CF-K^{2}ISSME .

5.5.2.2 Second Solution (The Riemannian Approach)

Classical optimization methods generally turn a constrained optimization problem into a sequence of unconstrained problems for which unconstrained techniques can be applied. In contrast, recent advances in optimization on Riemannian manifolds offer an alternative if the constraints can be modeled by a Riemannian structure. This is indeed the case here.

Consider a constrained optimization problem in the form of minimizing $f(\boldsymbol{x})$ with the constraint that $\boldsymbol{x}$ should lie on a Riemannian manifold $\mathcal{M}$ (think of a Riemannian manifold as a smooth surface embedded in some Euclidean space). This problem can be understood as an unconstrained problem in the form $f : \mathcal{M} \to \mathbb{R}$. Optimization techniques on Riemannian manifolds (e.g., Riemannian Gradient Descent (RGD)) enjoy several unique properties (e.g., convergence, smooth behavior) that make them competent alternatives to classical techniques.

To apply RGD on $f : \mathcal{M} \to \mathbb{R}$, one ultimately needs to have the gradient of f at $\boldsymbol{x}$, i.e., $\mathrm{grad}_{\boldsymbol{x}} f \in T_{\boldsymbol{x}}\mathcal{M}$ with $T_{\boldsymbol{x}}\mathcal{M}$ denoting the tangent space of $\mathcal{M}$ at $\boldsymbol{x}$. For the problem of our interest, i.e., minimizing $\mathcal{L}(\boldsymbol{C})$ while satisfying $\boldsymbol{C} \succ 0$, the Riemannian structure that describes the constraint is $\mathcal{S}_{++}^n$, e.g. the manifold of SPD matrices. For a smooth function $f : \mathcal{S}_{++}^n \to \mathbb{R}$, the gradient $grad_{\boldsymbol{C}}\, f \in T_{\boldsymbol{C}}\mathcal{S}_{++}^n$ is given by

$$grad_{\boldsymbol{C}}\, f = \boldsymbol{C} sym\big(\nabla_{\boldsymbol{C}}(f)\big)\boldsymbol{C}\,, \qquad (5.16)$$

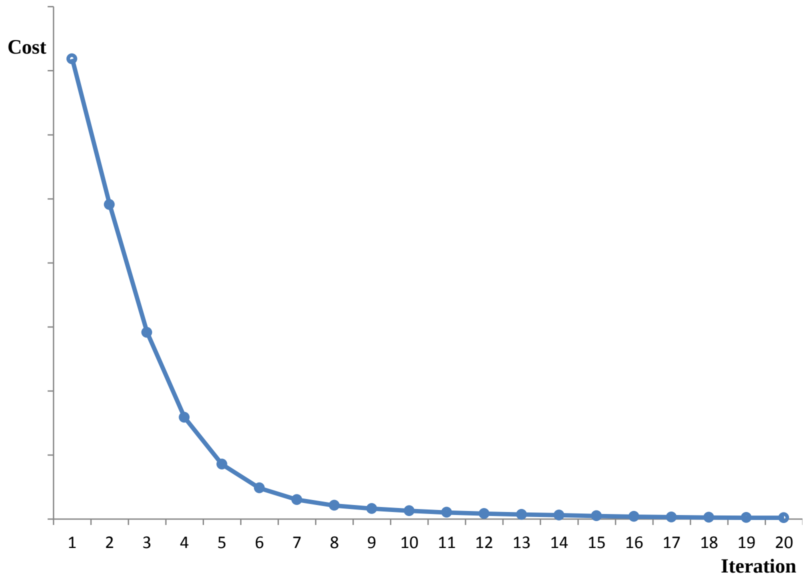

Figure 5.1: Convergence behavior of our R-K^{2}ISSME algorithm.

where $\nabla_C(\cdot)$ is the Euclidean gradient w.r.t $\boldsymbol{C}$ and

$$sym(\boldsymbol{X}) = \frac{\boldsymbol{X} + \boldsymbol{X}^T}{2} \,.$$

We have already computed $\nabla_C(\cdot)$ in the previous section, hence applying RGD is straightforward. In our experiments, we refer to this method as R-K^{2}ISSME . We use the implementation provided by the Manopt toolbox Boumal et al. [2014] to determine $\boldsymbol{C}$.

Figure 5.1 illustrates the convergence behavior of our R-K^{2}ISSME algorithm using the iLIDS dataset Zheng et al. [2009]. In all our experiments, we observed that the algorithm typically converges in less than 30 iterations, thus making it scalable to learning large metrics. To have a complete picture, we report the computational load of our proposal for our last experiment in §5.5.5. Averaging over 10 splits on a quad-core machine using Matlab, computing the kernel matrix for all samples takes about 110 seconds. Computing the metric matrix in the CF-K^{2}ISSME takes 0.7 seconds, making it the preferred technique when computational cost is important. Finally, performing 30 iterations in the R-K^{2}ISSME takes near 45 seconds.

5.5.3 Efficient Computation of the Mahalanobis Distances in $\mathcal{H}$

Once $\boldsymbol{C}$ is obtained either by the first method or the second solution, the Mahalanobis distance in $\mathcal{H}$ can be obtained as

$$\begin{aligned} d_{\mathcal{H}}(\boldsymbol{p},\boldsymbol{q}) &= \sqrt{\big(\phi(\boldsymbol{p})-\phi(\boldsymbol{q})\big)^T \Phi_{trn}\boldsymbol{C}\Phi_{trn}^T\big(\phi(\boldsymbol{p})-\phi(\boldsymbol{q})\big)} \\ &= \sqrt{\boldsymbol{k}_{p,trn}\boldsymbol{C}\boldsymbol{k}_{p,trn}^T - 2\boldsymbol{k}_{p,trn}\boldsymbol{C}\boldsymbol{k}_{q,trn}^T + \boldsymbol{k}_{q,trn}\boldsymbol{C}\boldsymbol{k}_{q,trn}^T}\ . \end{aligned} \tag{5.17}$$

which answers our third question.

5.5.4 The Nyström Solution

In the previous parts, we showed how the KISSME algorithm can be kernelized. Very related to our goal in this part is the concept of approximating the feature map ϕ of a pd kernel. For specific kernels (e.g., the Gaussian kernel), such approximations are known Vedaldi and Zisserman [2012]. Hence, one can obtain a vectorized representation of the kernel space towards kernelizing the KISSME algorithm.

For more complicated kernel functions, one can employ the Nyström method to kernelize KISSME. The Nyström method is a data-driven approach to estimate the RKHS induced by a kernel. Briefly, let $\mathcal{D} = \{\boldsymbol{t}_i\}_{i=1}^M$ be a collection of M training samples. A rank D approximation to $\boldsymbol{K} = [k(\boldsymbol{t}_i, \boldsymbol{t}_j)]_{M\times M}$ can be obtained using SVD as $\boldsymbol{K} \simeq \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T$. Here, $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$ is a diagonal matrix keeping the top D eigenvalues of $\boldsymbol{K}$ and $\boldsymbol{V} \in \mathbb{R}^{M\times D}$ is a column matrix storing the associated top eigenvectors. Having the low-rank representation at our disposal, a D-dimensional approximation to $\phi(\boldsymbol{x})$ is given by

$$\hat{\phi}(\boldsymbol{x}) = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{V}\Big(k(\boldsymbol{x},\boldsymbol{t}_1),\cdots,k(\boldsymbol{x},\boldsymbol{t}_M)\Big)^T. \tag{5.18}$$

We will call this solution, i.e., obtaining $\hat{\phi}(\cdot)$ followed by applying the original KISSME algorithm, the Nyström-KISSME method.

5.5.5 Experiments

In this section, we compare our proposed methods with several state-of-the-art metric learning techniques. In particular, we evaluate the performance of our R-K^{2}ISSME , CF-K^{2}ISSME , and Nyström-KISSME against LMNN Weinberger and Saul [2009], ITML Davis et al. [2007], LDML Guillaumin et al. [2009] and KISSME Koestinger et al. [2012]. As another indica-

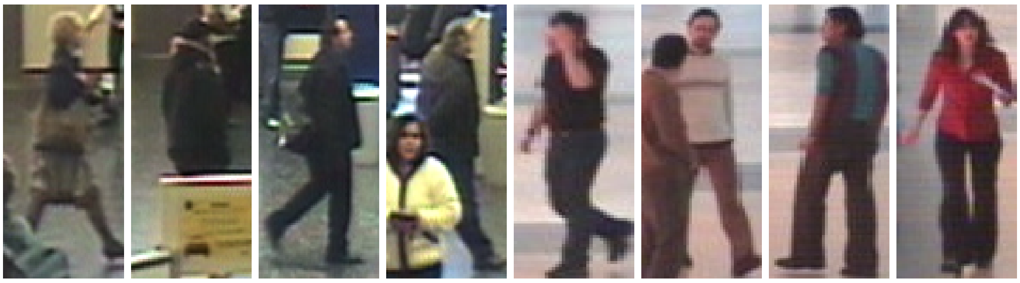

Figure 5.2: From left to right four sample images of the iLIDS Zheng et al. [2009] and the CAVIAR Cheng et al. [2011] datasets are shown, respectively.

Table 5.1: CMC at rank r on the iLIDS dataset with $p = 60$ test individuals.

Method	r = 1	r = 5	r = 10	r = 20
kLFDA-χ^2	36.5%	64.1%	76.5%	88.5%
MFA-χ^2	32.6%	58.5%	71.5%	84.5%
LMNN	32.6%	56.2%	68.9%	83.0%
ITML	29.5%	50.3%	62.6%	76.4%
LDML	27.8%	53.2%	67.0%	82.5%
KISSME	30.3%	54.8%	68.3%	83.6%
Nyström-KISSME	33.1%	60.6%	73.2%	86.2%
CF-K^{2}ISSME	37.8%	64.3%	76.5%	88.7%
R-K^{2}ISSME	**38.1**%	**65.0**%	**78.2**%	**89.4**%

tor, we also measure our performance to dataset-specific baselines. For all the baselines, we carefully tune their parameters and report their maximum accuracies here.

In all the experiments, we follow the so-called restricted protocol, where only the set of similar/dissimilar pairs is available during training. Furthermore, we utilize the parameter-free Chi-squared kernel depicted below in R-K^{2}ISSME , CF-K^{2}ISSME and Nyström-KISSME .

$$k_{\chi^2}(x, y) = \sum_i \frac{2x_i y_i}{x_i + y_i} \ . \tag{5.19}$$

5.5.5.1 Person Reidentification

As our first experiment, we tackled the task of person reidentification using two widely used datasets, namely iLIDS Zheng et al. [2009] and CAVIAR Cheng et al. [2011]. The iLIDS dataset contains images of 119 pedestrians captured by 8 cameras with different view points in an airport. Each individual has 2 to 8 images, and the dataset exhibits severe occlusions caused by people and their luggage. The CAVIAR4REID (CAVIAR) dataset includes 1220 images of 72 different persons captured from two different cameras in an indoor shopping mall. The number of images per individual varies from 10 to 20. Sample images of both datasets are shown in Fig. 5.2.

In our experiments, we followed the standard single-shot protocol. That is, the dataset was randomly partitioned into two exclusive subset of individuals, with p individuals constituting the test set and the remaining ones forming the training data. The random partitioning was repeated 10 times. In each partition, one image from each individual in the test set was randomly selected as the reference image and the rest of the images were used as query images. This process was repeated 20 times.

As for features, we used the histogram based descriptors provided by Xiong et al. [2014] for fair comparisons[1]. More specifically, each image in the dataset is described by 16-bin histogram of RGB, YUV and HSV color channels, as well as texture histograms based on the Local Binary Patterns (LBP) Ojala et al. [2002] extracted from 6 non-overlapping horizontal bands. This leads to a 2580 dimensional descriptor for each image.

Aside from the aforementioned MML baselines, we compare our proposed algorithms with the state-of-the-art kernel Local Fisher Discriminant Analysis (kLFDA) Xiong et al. [2014] and Marginal Fisher Analysis (MFA) Xiong et al. [2014]. Assuming Gaussian distribution for each class and using the Fisher discriminant objective, kLFDA finds a projection matrix to maximize the between-class scatters while minimizing the within-class scatters. MFA is a graph embedding dimensionality reduction method which allows to maximize the marginal discriminant even when the class distributions are not Gaussian.

We report performances in terms of the Cumulative Match Characteristic (CMC) curves for different rank values in Tables 5.1 and 5.2. To obtain CMC curves, a hit for rank k is considered if the correct class is identified among the k-nearest points of a query. From Table 5.1, we observe that our R-K^{2}ISSME achieves the highest scores for all the studied ranks. On the CAVIAR dataset, the best reported performance was achieved using the CF-K^{2}ISSME , while R-K^{2}ISSME works on par with that. It is worth mentioning that both kLFDA and MFA require the subject identities during training (i.e., they are unrestricted approaches) while our proposals do not require such additional information.

A parameter to take care of in R-K^{2}ISSME and CF-K^{2}ISSME is the number of eigenvalues and eigenvectors used to establish W_Z (see Eq. (5.11)). A similar parameter in conventional KISSME and Nyström-KISSME is the dimensionality of PCA (required as a preprocessing step) and rank of Nyström approximation, respectively. In Fig. 5.3, we analyze the sensitivity of R-K^{2}ISSME , CF-K^{2}ISSME , Nyström-KISSME and KISSME over the aforementioned

[1] https://github.com/NEU-Gou/kernel-metric-learning-reid

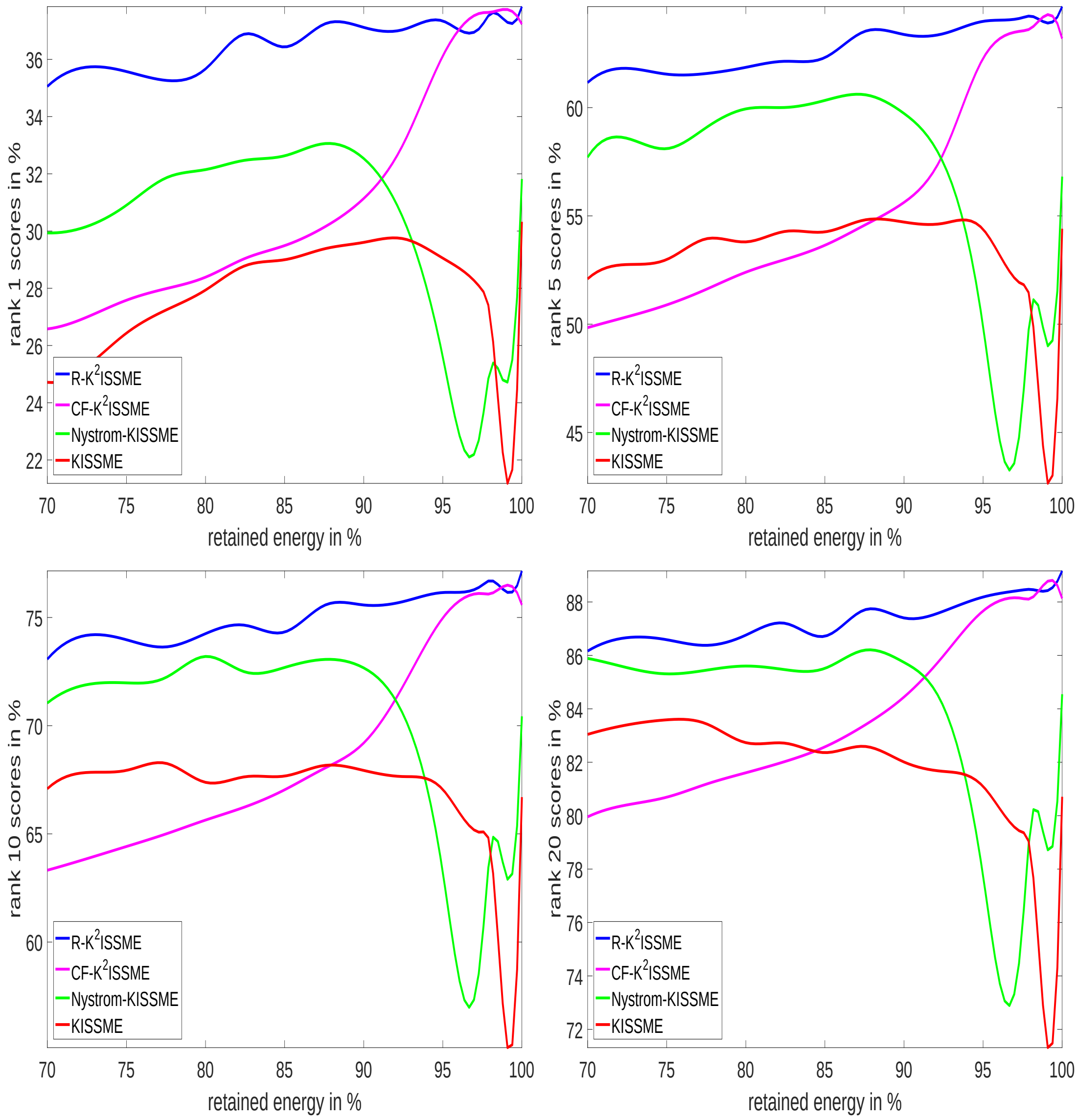

Figure 5.3: Rank x scores vs retained variance of the data on the iLIDS dataset Zheng et al. [2009] where x is 1, 5, 10, 20.

parameters on the iLIDS dataset. Both R-K^{2}ISSME and CF-K^{2}ISSME demonstrate robust and increasing performances when most of the energy is preserved. In contrast, the performance of Nyström-KISSME and KISSME may drop if more than 90% of energy is preserved. This is inline with other studies (such as Xiong et al. [2014]) that show the input dimensionality must be set carefully to enable KISSME to perform effectively.

As an indicator, on the iLIDS dataset, the deep net proposed in Ding et al. [2015] achieves

Table 5.2: CMC at rank r on the CAVIAR dataset with $p = 36$ test individuals.

Method	$\mathbf{r = 1}$	$\mathbf{r = 5}$	$\mathbf{r = 10}$	$\mathbf{r = 20}$
kLFDA-χ^2	36.2%	64.0%	78.7%	92.2%
MFA-χ^2	37.7%	67.2%	82.1%	94.6%
LMNN	33.8%	61.9%	78.6%	92.0%
ITML	29.1%	61.4%	75.8%	92.0%
LDML	30.4%	62.5%	77.8%	91.2%
KISSME	31.4%	61.9%	77.8%	92.5%
Nyström-KISSME	37.5%	67.5%	82.5%	95.0%
CF-K^{2}ISSME	**38.7%**	**68.2%**	**82.9%**	**95.4%**
R-K^{2}ISSME	**38.7%**	67.1%	80.9%	95.0%

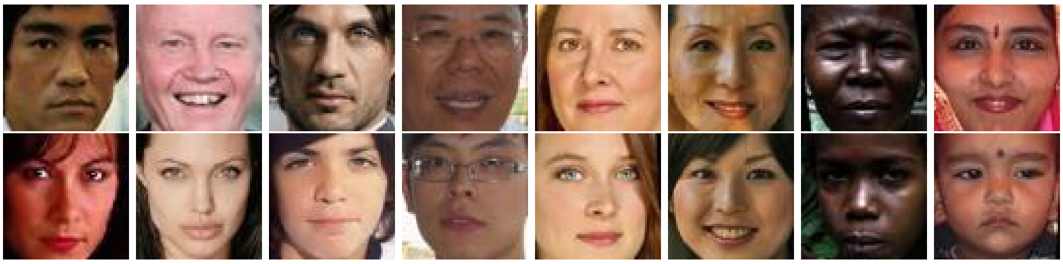

Figure 5.4: Examples of the KinFace-I and KinFace-II datasets Lu et al. [2014]. From left to right two examples are shown in each column for kinship relations: F-D, F-S, M-D, and M-S, respectively.

52.1%, 68.2%, 78.0%, and 88.8% at rank 1, 5, 10, and 20, respectively. Interestingly, our method performs on par or better than the deep solution for rank 5, 10, and 20 while underperforming at rank 1. This shows a potential research direction by incorporating the proposed technique in a deep net to benefit from deep architectures.

5.5.5.2 Kinship Verification

We performed another experiment to verify kinship relations from facial images. To this end, we made use of the KinFace-I dataset Lu et al. [2014] (see Fig. 5.4). The dataset contains images of four kin types: Father-Son (F-S), Father-Daughter (F-D), Mother-Son (M-S), and Mother-Daughter (M-D).

The coordinates of eyes in each face image are manually labeled, and facial regions are cropped and aligned into 64×64 templates. Then, histogram equalization is applied to mitigate the illumination variation. We have used the provided LBP features in our experiments. More specifically, each face image is divided into blocks of size 16×16 and for each block a 256 dimensional LBP histogram is extracted. The extracted histograms are finally concatenated to form a 4096 dimensional descriptor.

Table 5.3: Classification accuracies on various subsets of the KinFace-I dataset.

Method	**F-D**	**F-S**	**M-D**	**M-S**	**Mean**
NRML	65.2%	64.7%	65.4%	59.4%	63.7%
LMNN	63.2%	62.7%	63.4%	57.4%	61.7%
ITML	55.2%	58.3%	56.7%	55.6%	56.5%
LDML	57.1%	60.5%	57.4%	57.4%	58.1%
KISSME	65.4%	72.8%	66.7%	65.5%	67.6%
Nyström-KISSME	69.8%	**79.8**%	70.1%	68.5%	72.1%
CF-K^{2}ISSME	70.9%	**79.8**%	69.4%	66.0%	71.5%
R-K^{2}ISSME	**71.3**%	79.5%	**73.7**%	**69.4**%	**73.5**%

Table 5.4: Classification accuracies on various subsets of the KinFace-II dataset.

Method	**F-D**	**F-S**	**M-D**	**M-S**	**Mean**
NRML	69.5%	69.0%	69.0%	69.8%	69.5%
LMNN	68.5%	68.0%	67.0%	68.8%	68.2%
ITML	63.6%	69.2%	63.4%	64.2%	65.1%
LDML	65.6%	68.0%	66.0%	65.8%	66.4%
KISSME	72.0%	68.6%	68.6%	68.6%	70.4%
Nyström-KISSME	62.6%	64.1%	**72.6**%	70.2%	67.4%
CF-K^{2}ISSME	73.0%	77.5%	69.2%	70.2%	72.5%
R-K^{2}ISSME	**75.6**%	**78.4**%	68.6%	**73.2**%	**74.0**%

In Table 5.3, we compare our proposed algorithms against the baselines and the state-of-the-art NRML Lu et al. [2014] on the KinFace-I dataset Lu et al. [2014]. R-K^{2}ISSME , CF-K^{2}ISSME and Nyström-KISSME outperform the state-of-the-art NRML by a large margin. For example, the gap between R-K^{2}ISSME and NRML is near 10%. We also note that R-K^{2}ISSME , CF-K^{2}ISSME and Nyström-KISSME are superior to the other metric learning baselines.

In Table 5.4, we provide the results on the KinFace-II dataset Lu et al. [2014]. Here, our R-K^{2}ISSME again achieves the highest accuracy with CF-K^{2}ISSME being the second best. Both R-K^{2}ISSME and CF-K^{2}ISSME comfortably outperform the state-of-the-art NRML Lu et al. [2014] method[2].

5.5.5.3 Action Similarity Matching

As our last experiment, we considered the task of action similarity recognition using the ASLAN dataset Kliper-Gross et al. [2012]. The dataset contains 3,697 unique human action clips collected from YouTube, spanning 432 categories (see Fig. 5.5 for example frames). The

[2] We note that a recent study by López et al. discusses the bias in the KinFace dataset. Since our main goal here is to compare our proposal with other metric learning techniques, the bias does not harm the conclusions made here.

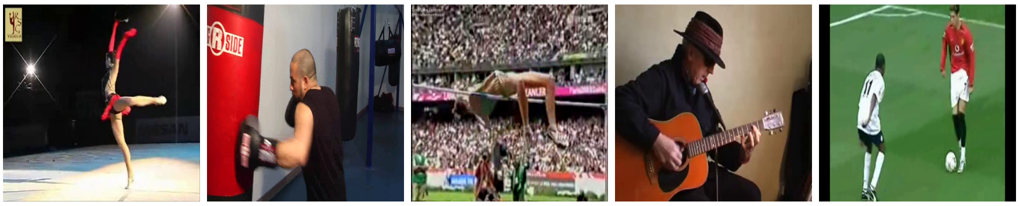

Figure 5.5: Examples of the ASLAN dataset Kliper-Gross et al. [2012].

Table 5.5: Matching accuracies on various descriptors of the ASLAN dataset Kliper-Gross et al. [2012].

Method	HoG	HoF	Hnf
Baseline Kliper-Gross et al. [2012]	54.2%	54.0%	54.5%
LMNN	55.9%	53.5%	56.0%
ITML	55.6%	53.9%	55.9%
LDML	57.3%	56.5%	58.0%
KISSME	55.2%	52.8%	55.7%
Nyström-KISSME	55.6%	53.3%	56.0%
CF-K^{2}ISSME	57.3%	57.8%	57.5%
R-K^{2}ISSME	**57.9%**	**58.3%**	**58.2%**

benchmark protocol is a binary pair matching and the goal is to decide whether two videos present the same action or not. The sample distribution across the categories in the benchmark is quite unbalanced, with 116 categories possessing only one video clip. Furthermore, categories included in the test sets are not available during training.

An action is represented by spatio-temporal bag-of-words descriptor Laptev et al. [2008] with a codebook of size 5,000 evaluated individually on three different types of descriptors, namely Histogram of Oriented Gradients (HoG), Histogram of Optical Flow (HoF) and a combination of both (HnF). We followed the standard matching protocol on this dataset which makes use of 10 predefined splits of data. There are 12,000 samples including 5,400 training and 600 testing pairs of action videos in each split.

In Table 5.5, we compare our proposed algorithms against the baselines on the ASLAN dataset. Here, our R-K^{2}ISSME again achieves the highest accuracies, while the closed-form solution works on par with it. Compared to the conventional KISSME, the Nyström-KISSME offers a better recognition rate, demonstrating benefits of analysis in the estimated RKHS in this method.

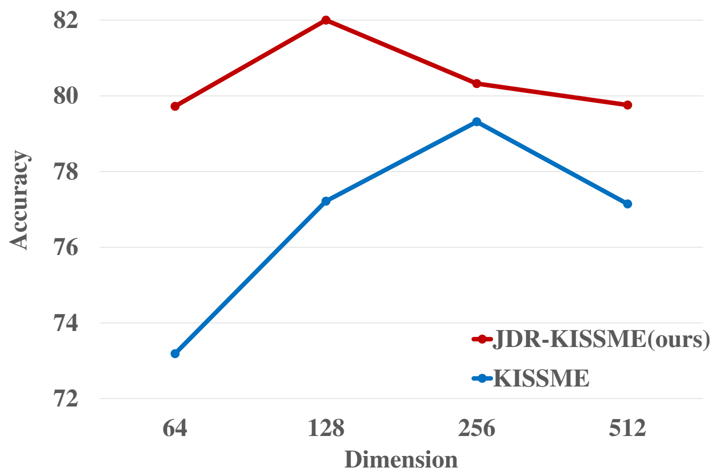

Figure 5.6: Verification accuracy against dimensionality of the space for an experiment using the surveillance nature images of the CompCars dataset (see §5.6.3.5 for more details).

5.6 Dimensionality Reduction for KISSME

Some recent studies (e.g., Xiong et al. [2014]) show that the KISSME algorithm is successful only when its input is carefully processed and denoised using PCA. This in turn results in high degree of sensitivity to the dimensionality of the PCA step as shown in Fig. 5.6. In this section, we propose to learn a low-dimensional subspace along its metric in the spirit of the KISSME algorithm in a unified fashion. Furthermore, based on our derivation, we show end-to-end learning of a generic deep CNN for metric learning.

In short, our contributions here are

1. We propose a Joint Dimensionality Reduction formulation for KISSME algorithm (JDR-KISSME) that learns a low-dimensional space along its metric in the spirit of the KISSME. In doing so, we benefit from the optimization techniques on Riemannian manifolds Absil et al. [2009] and in particular the geometry of Grassmann manifolds.

2. Upon our development, we propose a few simple, yet effective steps, to train a deep network for metric learning using KISSME verification signal as supervision.

Our experimental validation show that the JDR-KISSME consistently improves the conventional KISSME performance and achieves state-of-the-art results when large scale metric learning problems are addressed. Furthermore, on challenging datasets, its deep extension achieves very promising results and comfortably outperforms other state-of-the-art approaches.

5.6.1 Joint Dimensionality Reduction And Metric Learning

In this part, we present our idea to jointly learn a low-dimensional space and its metric. The lemma below provides the basis of metric construction using $\delta(\cdot,\cdot)$ (i.e., Eq. (5.2)) as suggested in Koestinger et al. [2012] and is central to our developments presented in the following.

Lemma 8. *Let $\delta : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^+$ be the function defined in Eq.* (5.2)*. The* $\log(\delta)$ *approximates a form of Mahalanobis metric, up to a constant term in* $\mathbb{R}^D$*. The approximated Mahalanobis metric is recognized as* $\boldsymbol{M} = \mathrm{Proj}(\Sigma_s^{-1} - \Sigma_d^{-1})$ *with* $\mathrm{Proj}(\cdot) : \mathrm{Sym}(D) \to \mathcal{S}_{++}^D$.

Proof. We note that

$$\log\big(\delta(\boldsymbol{x}_i, \boldsymbol{y}_i)\big) = \frac{1}{2}\Big(\log\det(\Sigma_s) - \log\det(\Sigma_d) \\ + (\boldsymbol{x}_i - \boldsymbol{y}_i)^T(\Sigma_s^{-1} - \Sigma_d^{-1})(\boldsymbol{x}_i - \boldsymbol{y}_i)\Big)\ . \tag{5.20}$$

The first two terms are constant and hence can be removed without loss of generality. From $(\boldsymbol{x}_i - \boldsymbol{y}_i)^T \boldsymbol{M}(\boldsymbol{x}_i - \boldsymbol{y}_i), \boldsymbol{M} \in \mathcal{S}_{++}^D$, (i.e., general form of the squared Mahalanobis distance) and by noting that $\Sigma_s^{-1} - \Sigma_d^{-1}$ is not necessarily an SPD matrix, we conclude that an approximated Mahalanobis metric associated to $\log(\delta)$ has the form $\boldsymbol{M} = \mathrm{Proj}(\Sigma_s^{-1} - \Sigma_d^{-1})$. □

As discussed earlier, the metric $\boldsymbol{M}$, while scaling well to the number of (dis)similar pairs, is sensitive to the dimensionality of the space (see for example Xiong et al. [2014]). In practice, finding the optimal dimension (or equivalently the most discriminative subspace) is done through PCA. Obviously, finding the subspace along its metric is more appealing and promises better discriminatory power. As such, our goal is to find a lower dimensional space and its Mahalanobis metric by making use of lemma 8. Formally, we seek a linear mapping $h : \mathbb{R}^D \to \mathbb{R}^d$ and an SPD matrix $\boldsymbol{M}$ such that

$$d_M^2\big(h(\boldsymbol{x}_i), h(\boldsymbol{y}_i)\big) = \big(h(\boldsymbol{x}_i) - h(\boldsymbol{y}_i)\big)^T \boldsymbol{M}\big(h(\boldsymbol{x}_i) - h(\boldsymbol{y}_i)\big) \tag{5.21}$$

reflects the dissimilarity function in Eq. (5.2) better. In doing so, we base our derivations on Koestinger et al. [2012] and rewrite Eq. (5.20) using $h(\boldsymbol{x}) = \boldsymbol{W}^T\boldsymbol{x},\ \boldsymbol{W} \in \mathbb{R}^{D\times d}$ as

$$\log\big(\delta(h(\boldsymbol{x}_i), h(\boldsymbol{y}_i))\big) = \frac{1}{2}\Big(\log\det(\boldsymbol{W}^T\Sigma_s\boldsymbol{W}) \\ - \log\det(\boldsymbol{W}^T\Sigma_d\boldsymbol{W}) + (\boldsymbol{x}_i - \boldsymbol{y}_i)^T\boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{y}_i)\Big)\ . \tag{5.22}$$

This enables us to define the loss over all training pairs as

$$\mathcal{E}\big(\boldsymbol{W},\boldsymbol{M}\big) \triangleq \sum_{i=1}^{n} l_i \ \log\big(\delta(h(\boldsymbol{x}_i),h(\boldsymbol{y}_i))\big) \ . \tag{5.23}$$

Minimizing Eq. (5.23) indeed makes the distances between similar pairs smaller while simultaneously increases the distances between dissimilar pairs. For reasons become clear soon, we opt for an alternating optimization scheme to obtain $\boldsymbol{W}$ and $\boldsymbol{M}$. That is, we keep $\boldsymbol{W}$ fixed to update $\boldsymbol{M}$, followed by updating $\boldsymbol{W}$ by keeping $\boldsymbol{M}$ fixed. By fixing $\boldsymbol{W}$, we make use of lemma 8 to obtain a closed form update for $\boldsymbol{M}$. We note that the covariance matrix Σ in the space defined by $\boldsymbol{W}$ has the form of $\boldsymbol{W}^T\Sigma\boldsymbol{W}$. Using lemma 8, this in turn leads to the following metric in the latent space

$$\boldsymbol{M}^* = \operatorname{Proj}\Big(\big(\boldsymbol{W}^T\Sigma_s\boldsymbol{W}\big)^{-1} - \big(\boldsymbol{W}^T\Sigma_d\boldsymbol{W}\big)^{-1}\Big). \tag{5.24}$$

To update $\boldsymbol{W}$ while $\boldsymbol{M}$ is fixed, we add an orthogonality constraint to $\boldsymbol{W}$. The orthogonality constraint helps to avoid degeneracy in the solution and is inline with the general practice in dimensionality reduction Weinberger and Saul [2009]. As such, we can write the following constrained optimization problem

$$\begin{array}{ll} \min_{\boldsymbol{W}} & \mathcal{E}\big(\boldsymbol{W},\boldsymbol{M}^*\big) \\ \text{s.t.} & \boldsymbol{W}^T\boldsymbol{W} = \mathbf{I}_d \end{array} \tag{5.25}$$

To minimize (5.25), we make use of the recent advances in optimization over the matrix manifolds Absil et al. [2009]. In particular, the constrained optimization problem in (5.25) can be understood as a minimization problem on space of tall matrices with orthogonal columns which we solve using Riemannian Conjugate Gradient Descent (RCGD) on Grassmannian.

The geometrically correct setting to minimize a problem with the orthogonality constraint is by making use of the geometry of the Stiefel manifold $\mathcal{S}_d^D = \{\boldsymbol{W} \in \mathbb{R}^{D\times d}, \boldsymbol{W}^T\boldsymbol{W} = \mathbf{I}_d\}$. The Grassmannian manifold $\mathcal{G}_d^D$ consists of the set of all linear d-dimensional subspaces of $\mathbb{R}^D$ and is the quotient of $\mathcal{S}_d^D$ with the equivalence class being (see Absil et al. [2009]; Edelman et al. [1998] for details)

$$[\boldsymbol{W}] \triangleq \{\boldsymbol{W}\boldsymbol{R}, \boldsymbol{W} \in \mathcal{S}_d^D, \boldsymbol{R} \in \mathcal{O}(d)\} \ ,$$

with $\mathcal{O}(d)$ denoting the orthogonal group, i.e., $\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^T = \mathbf{I}_d$.

A constrained optimization problem with the orthogonality constraint is a problem on

Grassmannian if its objective is invariant to the right action of $\mathcal{O}(d)$. This is indeed the case as a result of the following theorem.

Theorem 9. *The objective defined in Eq.* (5.23) *is invariant to the right action of* $\mathcal{O}(d)$*, i.e.,* $\mathcal{E}\big(\boldsymbol{W},\boldsymbol{M}^*\big) = \mathcal{E}\big(\boldsymbol{W}\boldsymbol{R},\boldsymbol{M}^*\big), \boldsymbol{R} \in \mathcal{O}(d)$.

Proof. First, we show that the $\log\det(\cdot)$ terms in Eq. (5.22) are invariant to the action of $\mathcal{O}(d)$. Consider the first term for example. Direct insertion results in

$$\begin{aligned}\det(\boldsymbol{R}^T\boldsymbol{W}^T\Sigma_s\boldsymbol{W}\boldsymbol{R}) &= \det(\boldsymbol{R}^T)\det(\boldsymbol{W}^T\Sigma_s\boldsymbol{W})\det(\boldsymbol{R})\\ &= \det(\boldsymbol{W}^T\Sigma_s\boldsymbol{W}),\end{aligned}$$

where we used the fact that $\det(\boldsymbol{R}^T) = \det(\boldsymbol{R}^{-1}) = 1/\det(\boldsymbol{R})$.

Now we show that the term with the metric is invariant to the action of $\mathcal{O}(d)$ as well. Let $\boldsymbol{A}^+ = \mathrm{Proj}(\boldsymbol{A}), \forall \boldsymbol{A} \in \mathrm{Sym}(d)$. Using SVD, it is easy to see that $\boldsymbol{R}^T\boldsymbol{A}^+\boldsymbol{R} = \mathrm{Proj}(\boldsymbol{R}^T\boldsymbol{A}\boldsymbol{R}), \forall \boldsymbol{R} \in \mathcal{O}(d)$. This in turn leads to recognizing Eq. (5.24) by replacing $\boldsymbol{W}$ with $\boldsymbol{W}\boldsymbol{R}$ as

$$\begin{aligned}&\mathrm{Proj}\Big(\big(\boldsymbol{R}^T\boldsymbol{W}^T\Sigma_s\boldsymbol{W}\boldsymbol{R}\big)^{-1} - \big(\boldsymbol{R}^T\boldsymbol{W}^T\Sigma_d\boldsymbol{W}\boldsymbol{R}\big)^{-1}\Big)\\ &= \mathrm{Proj}\Big(\boldsymbol{R}^{-1}\big(\boldsymbol{W}^T\Sigma_s\boldsymbol{W}\big)^{-1}\boldsymbol{R}^{-T} - \boldsymbol{R}^{-1}\big(\boldsymbol{W}^T\Sigma_d\boldsymbol{W}\big)^{-1}\boldsymbol{R}^{-T}\Big)\\ &= \mathrm{Proj}\Big(\boldsymbol{R}^T\Big(\big(\boldsymbol{W}^T\Sigma_s\boldsymbol{W}\big)^{-1} - \big(\boldsymbol{W}^T\Sigma_d\boldsymbol{W}\big)^{-1}\Big)\boldsymbol{R}\Big)\\ &= \boldsymbol{R}^T\mathrm{Proj}\Big(\big(\boldsymbol{W}^T\Sigma_s\boldsymbol{W}\big)^{-1} - \big(\boldsymbol{W}^T\Sigma_d\boldsymbol{W}\big)^{-1}\Big)\boldsymbol{R}\,.\end{aligned}$$

where we used the fact that $\boldsymbol{R}^T = \boldsymbol{R}^{-1}$.

As such and again by replacing $\boldsymbol{W}$ with $\boldsymbol{W}\boldsymbol{R}$ for the term with $\boldsymbol{M}^*$ involved (the third term in Eq. (5.22)), we arrive at

$$\begin{aligned}&(\boldsymbol{x}_i - \boldsymbol{y}_i)^T\boldsymbol{W}\boldsymbol{R}\boldsymbol{M}^*\boldsymbol{R}^T\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{y}_i) = (\boldsymbol{x}_i - \boldsymbol{y}_i)^T\boldsymbol{W}\boldsymbol{R}\boldsymbol{R}^T\times\\ &\mathrm{Proj}\Big(\big(\boldsymbol{W}^T\Sigma_s\boldsymbol{W}\big)^{-1} - \big(\boldsymbol{W}^T\Sigma_d\boldsymbol{W}\big)^{-1}\Big)\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{y}_i)\\ &= (\boldsymbol{x}_i - \boldsymbol{y}_i)^T\boldsymbol{W}\mathrm{Proj}\Big(\big(\boldsymbol{W}^T\Sigma_s\boldsymbol{W}\big)^{-1} - \big(\boldsymbol{W}^T\Sigma_d\boldsymbol{W}\big)^{-1}\Big)\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{y}_i).\end{aligned}$$

This concludes the proof as it shows all the terms are invariant to the right action of $\mathcal{O}(d)$. □

To perform RCGD on $\mathcal{G}_d^D$, we need to compute the Riemannian gradient of the loss $\mathcal{E}(\boldsymbol{W},\boldsymbol{M})$ with respect to $\boldsymbol{W}$. For a smooth function $f : \mathcal{G}_d^D \to \mathbb{R}$, the Riemannian gradient at $\boldsymbol{W}$ denoted by $\mathrm{grad}_{\boldsymbol{W}}f$ is an element of the tangent space $T_{\boldsymbol{W}}\mathcal{G}$ and is given by

$$\mathrm{grad}_{\boldsymbol{W}}f = \Big(\mathbf{I}_d - \boldsymbol{W}\boldsymbol{W}^T\Big)\nabla_{\boldsymbol{W}}f, \tag{5.26}$$

Algorithm 4 The proposed JDR-KISSME algorithm

```
Input: $\{\boldsymbol{x}_i, \boldsymbol{y}_i, l_i\}_{i=1}^n$ a set of training pairs in $\mathbb{R}^D$ with their similarity labels, the target dimensionality $d$
Output: Projection $\boldsymbol{W}$ and metric $\boldsymbol{M}$
Compute $\Sigma_d$ and $\Sigma_s$ using Eq. (5.3) and Eq. (5.4)
Initialize $\boldsymbol{W}$ to an orthonormal matrix (e.g., truncated identity)
Compute $\boldsymbol{M}^*$ using Eq. (5.24)
repeat
    $\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} \mathcal{E}\big(\boldsymbol{W}, \boldsymbol{M}^*\big)$ using RCGD on Grassmannian
    $\boldsymbol{W} \leftarrow \boldsymbol{W}^*$
    Update $\boldsymbol{M}^*$ using Eq. (5.24)
    $\boldsymbol{M} \leftarrow \boldsymbol{M}^*$
until convergence
```

where $\nabla_{\boldsymbol{W}} f$ is a $D \times d$ matrix of partial derivatives of f with respect to the elements of $\boldsymbol{W}$, i.e., $[\nabla_{\boldsymbol{W}} f]_{i,j} = \frac{\partial f}{\partial W_{i,j}}$. Below, we derive $\nabla_{\boldsymbol{W}} \mathcal{E}(\boldsymbol{W}, \boldsymbol{M})$. For a symmetric matrix Σ

$$\nabla_{\boldsymbol{W}} \log\det(\boldsymbol{W}^T \Sigma \boldsymbol{W}) = 2\Sigma \boldsymbol{W}\big(\boldsymbol{W}^T \Sigma \boldsymbol{W}\big)^{-1} .$$

Also,

$$\nabla_{\boldsymbol{W}} (\boldsymbol{x}_i - \boldsymbol{y}_i)^T \boldsymbol{W}\boldsymbol{M}\boldsymbol{W}^T (\boldsymbol{x}_i - \boldsymbol{y}_i) = \\ 2(\boldsymbol{x}_i - \boldsymbol{y}_i)(\boldsymbol{x}_i - \boldsymbol{y}_i)^T \boldsymbol{W}\boldsymbol{M}. \tag{5.27}$$

Therefore,

$$\nabla_{\boldsymbol{W}} \mathcal{E}(\boldsymbol{W}, \boldsymbol{M}) = \sum_{i=1}^{N} l_i \Big(\Sigma_s \boldsymbol{W}\big(\boldsymbol{W}^T \Sigma_s \boldsymbol{W}\big)^{-1} \\ - \Sigma_d \boldsymbol{W}\big(\boldsymbol{W}^T \Sigma_d \boldsymbol{W}\big)^{-1} + (\boldsymbol{x}_i - \boldsymbol{y}_i)(\boldsymbol{x}_i - \boldsymbol{y}_i)^T \boldsymbol{W}\boldsymbol{M} \Big) . \tag{5.28}$$

An alternating optimization solution not only lets us obtain $\boldsymbol{M}$ in closed form according to Eq. (5.24), but also justifies the use of Grassmannian, making the search space more confined. Putting everything together, the algorithm to learn $\boldsymbol{W}$ and $\boldsymbol{M}$ is depicted in Alg. 4. In our experiments, we observed that the algorithm typically converges in less than 30 iterations.

5.6.2 Incorporating the Solution into Deep Nets

In this part, we elaborate on how the previous developments can be incorporated into deep networks. The goal is to learn a mapping from images to a compact Euclidean space such that distances correspond to a notion of semantics between the images. Let us assume that a generic network provides us with an embedding from the image space to $\mathbb{R}^d$. We denote the

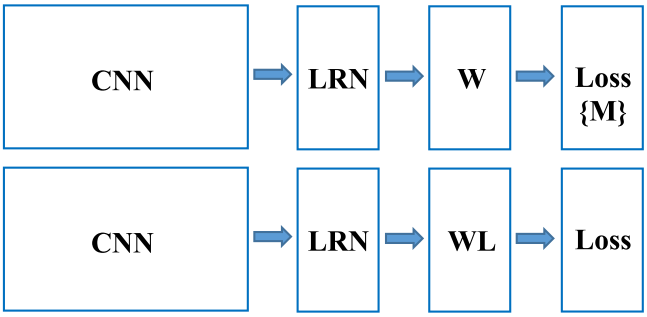

Figure 5.7: Incorporating JDR-KISSME into deep nets. The dimensionality reduction can be seen as an FC layer immediately before a loss layer. One can either have the metric $\boldsymbol{M}$ in computing the loss (top panel) or since $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$, combine it with the dimensionality reduction layer (bottom panel). Empirically, we found the first solution to be more stable.

functionality of this network on an input image x by $f(x)$.

Since a metric $\boldsymbol{M}$ is an SPD matrix, it can be decomposed as $\boldsymbol{M} = \boldsymbol{L}^T\boldsymbol{L}$. As a result, the distance between two images x_i and y_i passing through the network can be written as

$$\begin{aligned} d^2_M(x_i, y_i) &= \big(f(x_i) - f(y_i)\big)^T \boldsymbol{M}\big(f(x_i) - f(y_i)\big) \\ &= \|\boldsymbol{L}\big(f(x_i) - f(y_i)\big)\|_2^2 \,. \end{aligned} \tag{5.29}$$

This lets us incorporate the metric $\boldsymbol{M}$ into a deep net in the form of a Fully Connected (FC) layer immediately before a loss layer. The whole setup can be trained via BackPropagation (BP). Generally speaking, training a deep CNN for metric learning is cast as one of the following forms (see Song et al. [2016] and Schroff et al. [2015] for more details).

- **Pairwise**: training data consists of pairs of similar and dissimilar images. Given a predefined margin τ, training is guided by a loss function which seeks to learn an embedding such that distances between similar samples are smaller than τ while those between dissimilar ones are greater than τ. In this manner, the cost function for a batch with n pairs $\{x_i, y_i\}$ and their corresponding similarity label $l_i \in \{1, -1\}$ can be written as

$$\sum_{i=1}^{n} \Big[\big(\|f(x_i) - f(y_i)\|_2^2 - \tau\big) l_i \Big]_+ \tag{5.30}$$

- **Triplewise**: training data consists of triplets of images: one anchor x, a sample in the same class x^+, and one differently labelled sample x^-, and a predefined margin τ. Then, a loss function supervises training such that for each triplet, the distance between x and

x^- becomes greater than the distance between x and x^+ plus τ. Thus, the cost function for a batch with N triplets is

$$\sum_{i=1}^{n} \Big[\|f(x_i) - f(x_i^+)\|_2^2 - \|f(x_i) - f(x_i^-)\|_2^2 + \tau \Big]_+ \tag{5.31}$$

An important difference between the two categories is that only methods in the first group can work in the restricted metric learning scenario. We present our extension to deep networks utilizing the pairwise protocol, making our method applicable to wider set of problems. To this end, we start with an initial orthonormal $\boldsymbol{W}$ and compute $\boldsymbol{M}$ using Eq. (5.24), relying on the network to provide features in the low-dimensional space. To tune $\boldsymbol{W}$ and $\boldsymbol{M}$ via BP, two possibilities are

- Initialize the weights of the last FC layer to be $\boldsymbol{W}$ and engage the metric $\boldsymbol{M}$ directly in computing the distances in the loss layer. To this end, we perform Stochastic Gradient Descent (SGD) while $\boldsymbol{M}$ is kept fixed. We update $\boldsymbol{M}$ after a number of SGD iterations or when the network reaches to a reasonably good representation. In this case, BP updates the network according to the KISSME loss (i.e., Eq. (5.23)) while $\boldsymbol{M}$ is learned in a closed form manner using the output of the network and Eq. (5.24) (see the top panel in Fig. 5.7). We refer to this solution as "pairwise+KISSME".

- Since $\boldsymbol{M}$ is an SPD matrix (i.e., $\boldsymbol{M} = \boldsymbol{L}^T\boldsymbol{L}$), it can be absorbed in the last FC layer. Here, we initialize the weights of the last FC layer to be $\boldsymbol{W}\boldsymbol{L}$. Then, we train the network using BP. If the explicit form of the metric is required, the weights of the FC layer can be factorized into an orthogonal matrix $\boldsymbol{W}$ and a full-rank matrix $\boldsymbol{L}$ using any spectral decomposition such as QR decomposition (see the bottom panel in Fig. 5.7). We refer to this solution as "pairwise+KISSME-Compact".

Empirically, we observed that pairwise+KISSME solution is more stable and works better. We conjecture that the separation of learning $\boldsymbol{W}$ and $\boldsymbol{M}$ is the reason here. In § 5.6.3.4, we compare the two scenarios in more details. Before concluding this part, we would like to mention that placing a Local Response Normalization (LRN) block (see Fig. 5.7) before the dimensionality reduction block helps the convergence in our solution. This is inline with other deep metric learning models Liu et al. [2016]; Schroff et al. [2015].

5.6.3 Experiments

In this section, we assess and contrast the performance of our proposal against the KISSME baseline and several state-of-the-art methods. We begin by evaluating JDR-KISSME using the Comprehensive Cars (CompCars) Yang et al. [2015] and Market-1501 Zheng et al. [2015a] datasets. We then demonstrate the strength of the solution when incorporated into deep networks using the CUB200-2011 Wah et al. [2011] and Cifar100 Krizhevsky [2009] datasets.

5.6.3.1 Car Verification (Shallow Experiment)

The CompCars dataset is one of the largest benchmarks for image verification containing 214,345 images of 1,687 car models from two significantly different scenarios: web nature and surveillance nature. Web nature data is split into three subsets without overlap. Related to verification is part II and part III of the dataset. Part II contains 4,454 images in 111 models (classes) while there are 22,236 images spanning 1,145 models in part III. We followed Yang et al. [2015], the standard verification protocol on this dataset, which splits part III to three sets with different levels of difficulty, namely easy, medium, and hard. Each set contains 20,000 pairs including equal number of similar and dissimilar pairs. Each image in the easy pairs is chosen from the same viewpoint, while each pair in the medium pairs is selected from a random viewpoint. Each dissimilar pair in the hard subset is selected from the same car make. As for feature extraction, again following Yang et al. [2015], we utilized their available GoogLeNet Szegedy et al. [2015] fine tuned on part II of the CompCars.

In Table 5.6, we compare our JDR-KISSME method against the conventional KISSME algorithm and several state-of-the-art methods. First, we note that the JDR-KISSME shows consistent improvements over the KISSME. For example, the accuracy gap between the JDR-KISSME and the KISSME over the hard subset reaches 4.4%. The JDR-KISSME achieves the state-of-the-art verification accuracies on all the protocols. Moreover, we note that the work of Yang et al. [2015] (the closest competitor to our JDR-KISSME) utilizes class labels for training (and hence not applicable to the restricted metric learning scenarios) while our method is more general and does not rely on the availability of such information.

5.6.3.2 Person Re-Identification (Shallow Experiment)

Person re-identification is the practice of matching a probe image of an individual in a database of (gallery) images from non-overlapping views. The Market-1501 dataset is one of the largest datasets for this task containing over 32,000 bounding boxes of 1,501 identities captured by at

Table 5.6: Verification accuracy on the CompCars dataset.

Method	Easy	Medium	Hard	Mode
PCCA Mignon and Jurie [2012]	86.7%	81.4%	72.6%	Restricted
XQDA Liao et al. [2015]	87.8%	82.6%	74.4%	Unrestricted
MixedDiff+CCL Liu et al. [2016]	83.3%	78.8%	70.3%	Unrestricted
BoxCars Sochor et al. [2016]	85.0%	82.7%	76.8%	Unrestricted
Yang et al. Yang et al. [2015]	90.7%	85.2%	78.8%	Unrestricted
KISSME Koestinger et al. [2012]	88.9%	83.3%	75.4%	Restricted
JDR-KISSME(ours)	**91.0%**	**86.3%**	**79.8%**	Restricted

Table 5.7: CMC at rank-1 and mAP on the Market-1501 dataset.

Method	Rank@1	mAP	Mode
PCCA Mignon and Jurie [2012]	76.0%	52.8%	Restricted
XQDA Liao et al. [2015]	76.0%	53.0%	Unrestricted
KISSME Koestinger et al. [2012]	77.5%	53.9%	Restricted
JDR-KISSME(ours)	**79.2%**	**54.6%**	Restricted

least two (and at most six) cameras. The dataset is further enlarged using a distractor set of over 500,000 irrelevant (not belonging to the identities) images.

We adopted the experimental setting in Zheng et al. [2015a, 2016] which utilize the ResNet-50 He et al. [2016] to generate id-discriminative embedding for Market-1501 dataset. Here, the original dimensionality is 2048 and the target dimensionality is set to 200. Using the gallery size of 19,732 images and single-query evaluation, the mean average precision score (mAP) and the Cumulative Matching Curve (CMC) at rank-1 of our JDR-KISSME and baseline methods are reported in Table 5.7. Here, again our proposal comfortably outperforms the baseline methods.

5.6.3.3 Deep Experiments

In this part, we show the effectiveness of pairwise+KISSME, our deep metric learning method. To this end, we perform comparisons with the two most common ways of training a CNN for metric learning, i.e., pairwise metric learning (Eq. (5.30)) and triplet solution (Eq. (5.31)). We note that the triplet solution utilizes class labels while our method like the pairwise is more general and does not require the labels. We recall from § 5.6.2 that the training starts by initializing two matrices, $\boldsymbol{W}$ (or equivalently the last FC layer) and $\boldsymbol{M}$. To initialize $\boldsymbol{W}$, we rely on the initial CNN to provide embeddings of training images up to the last FC layer. Then, the FC layer is initialized with PCA (of a certain dimensionality). This is consistent with the original KISSME algorithm. Next, the metric $\boldsymbol{M}$ is initialized using the output of the FC layer (and Eq. (5.24)). This is required in the loss layer to perform BP.

To measure the performances, we randomly generate 30,000 similar and 30,000 dissimilar pairs from our test data. We report verification accuracy, Area Under the ROC Curve (AUC), and Equal Error Rate (EER) values for our algorithm and of the baselines for comparisons. We use Matconvnet package Vedaldi and Lenc [2015] for implementation. Before delving into experiments, we note that our aim here is to present a better, yet general way of metric learning for deep networks. In doing so, we are not chiefly concerned about best mining practices as suggested in Song et al. [2016]; Schroff et al. [2015].

Bird Verification

As our first experiment for deep metric learning, we considered the task of image verification using CUB200-2011 dataset Wah et al. [2011]. The CUB200-2011 has 200 classes of birds with 11,788 images. We used images of the first 100 classes as training and validation sets and the remaining classes for testing. As the CNN, we utilized the VGG-CNN-M-1024 Chatfield et al. [2014] pretrained on the ImageNet Deng et al. [2009]. The network contains 5 convolutional layers followed by 3 FC layers with 87 millions learnable parameters in total.

Fig. 5.9 summarizes the three score metrics of our deep metric learning technique and of our two baselines for various embedding sizes (or equivalently subspace dimensionality). Similar to the previous experiment, our solution is consistently superior to the baseline techniques for all embedding sizes. For example, the difference in the verification accuracy between our method and its closest competitor, i.e., the triplet solution, is about 2% for the size 128 over the 60,000 test pairs.

Fig. 5.8 shows the Barnes-Hut t-SNE Van Der Maaten [2014] visualization on our 128 dimensional embedding of the test split of the CUB-200-2011 dataset. Although a 2D mapping does not directly translate to the original high dimensional embedding, we can observe that similar species are projected together.

Tiny Image Verification

As another expperiment for deep metric learning, we studied the task of image verification using the Cifar100 dataset Krizhevsky [2009] which has 60,000 images of size 32×32. To this end, we trained the LeNet-5 LeCun et al. [1998] network (which has 2 convolutional layers followed by 2 FC layers) on the Cifar10 dataset Krizhevsky [2009] for 22,500 iterations of SGD. We then cropped the pretrained network at the fourth layer and fine tuned it on the Cifar100 similar to the previous experiment. We kept the embedding size to 32 for this experiment (i.e., $\boldsymbol{W}$ was 64×32 and $\boldsymbol{M}$ was 32×32). All other experimental details (e.g., train/test split, number of test pairs, etc) were the same as the bird verification experiment.

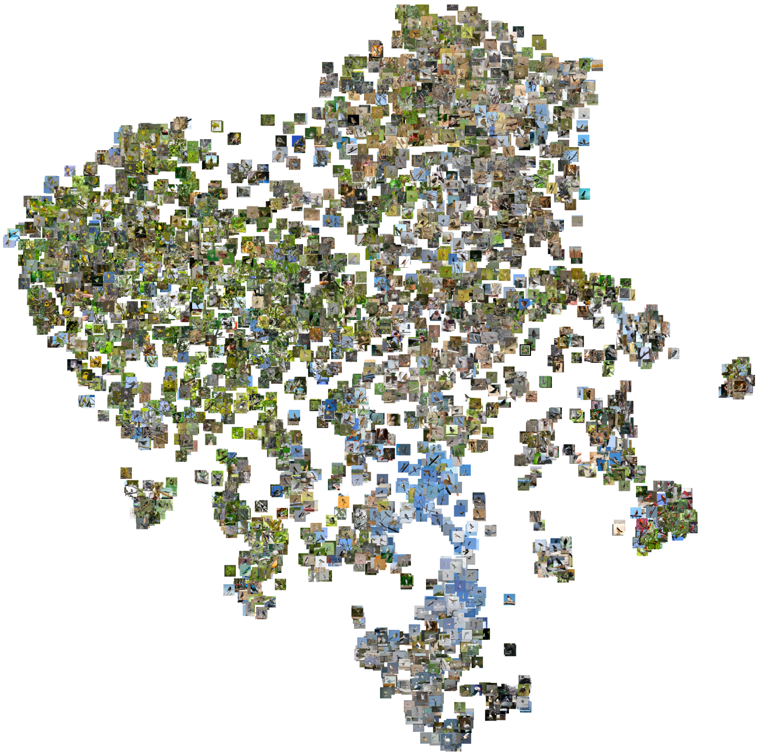

Figure 5.8: Barnes-Hut t-SNE visualization of our embedding on a subset of the test set of the CUB-200-2011. Best viewed when zoomed in.

Table 5.8: Accuracy, AUC, and EER on the Cifar100 dataset.

Method	Accuracy	AUC	EER
KISSME Koestinger et al. [2012]	63.1%	69.2%	36.4%
Pairwise	64.3%	70.0%	35.4%
Triplewise	65.8%	72.0%	34.2%
Pairwise+KISSME(ours)	**68.2%**	**75.2%**	**31.7%**

In Table 5.8, we compare our method against the so called pairwise and triplet methods as well as the original KISSME on the pretrained network (i.e., without fine tuning). Here again, our solution comfortably outperforms the other methods for all the studied metrics over the 60,000 test pairs.

5.6.3.4 Further Analysis

In this part, we empirically compare the pairwise+KISSME method to the pairwise+KISSME-Compact solution discussed in § 5.6.2. To this end, we conducted further experiments on the CUB200-2011 (bird verification) dataset. The accuracy, AUC and EER values for various

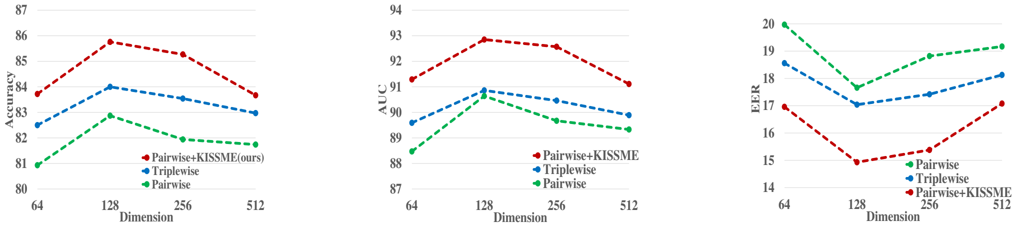

Figure 5.9: Verification accuracy, AUC, and EER score metrics on the CUB200-2011 dataset.

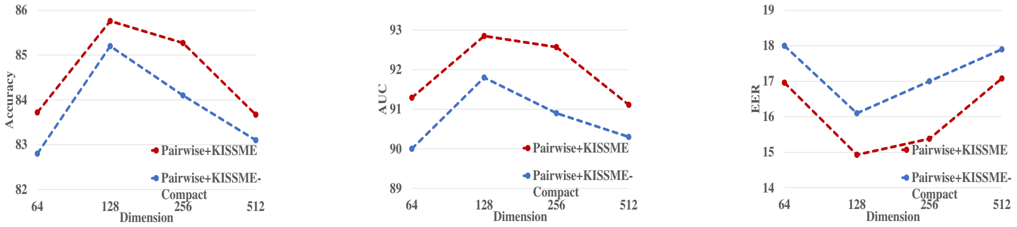

Figure 5.10: Comparison between the pairwise+KISSME and pairwise+KISSME-Compact deep solutions on the CUB200-2011 dataset.

Table 5.9: Accuracy, AUC, and EER when $\boldsymbol{M}$ is not incorporated into the networks.

Dataset	Accuracy	AUC	EER
CUB200-2011	84.2%	91.5%	16.6%
Cifar100	65.3%	71.7%	34.6%

embedding sizes are depicted for the two solutions in Fig. 5.10. From the Figure, we conclude that the pairwise+KISSME solution leads to superior performances and is more stable, hence our proposal. This is a consistent extension to the JDR-KISSME, our developments in the shallow mode, where we have an alternating algorithm to find the two matrices $\boldsymbol{W}$ and $\boldsymbol{M}$.

We conjecture that the separation of learning $\boldsymbol{W}$ and $\boldsymbol{M}$ is the reason. More specifically, assume the ideal projection is $\boldsymbol{W}^*$. From Lemma 8, we conclude that the ideal metric is obtained as $\boldsymbol{M}^* = \Gamma(\boldsymbol{W})$ where the function $\Gamma(\cdot)$ is a nonlinear function as a result of the projection to the positive definite cone. The pairwise+KISSME is more aligned with Lemma 8 as the metric is explicitly obtained from the representation. On the other hand, the pairwise+KISSME-Compact removes the dependency of $\boldsymbol{M}^*$ on $\boldsymbol{W}^*$ in the hope of learning $\boldsymbol{M}^*$, $\boldsymbol{W}^*$ and the underlying nonlinear projection together.

Furthermore, we studied the effect of keeping the metric $\boldsymbol{M}$ fixed while the network is trained. To this end, similar to the pairwise+KISSME, we fix $\boldsymbol{M}$ using the network outputs and tune the filters and FC layers according to the KISSME loss (i.e., Eq. (5.23)). Table 5.9 shows the performance measures for the CUB200-2011 (with $\boldsymbol{M} \in \mathcal{S}_{++}^{128}$) and Cifar100 datasets.

Comparing to the pairwise+KISSME, the performance gaps are about 1.7% and 2.9%, respectively, demonstrating that joint learning yields higher accuracies.

5.6.3.5 Experimental setup

For fine tuning the CNNs, we randomly generated 300,000 similar pairs and 300,000 dissimilar pairs (and equal number of triplets) and fed them to the CNNs. We exhaustively searched for all possible similar pairs within a batch and randomly sampled equal number of dissimilar pairs. For all experiments, we set maximum number of SGD iterations to 20,000, margin to $\tau = 1.0$, momentum to $\mu = 0.9$, and learning rate to $\eta = [10^{-4}, 10^{-7}]$ in log-space range. We observed that increasing the learning rate of the fully connected layer by a factor of 10 helps faster convergence. A similar observation is reported in Song et al. [2016]. To augment the data, we resorted to only flipping the images at random.

To have a complete picture, we report the computational burden of our methods here. Performing 30 iterations by the JDR-KISSME for the task of car verification takes about 90 seconds on a quad-core machine using Matlab. As for the pairwise+KISSME and two baselines, namely the pairwise and triplewise metric learning using the CUB200-2011 dataset (§ 5.6.3.3), in average for an embedding size of 128, performing 10,000 iterations of SGD takes 41,050 and 40,803 seconds for the pairwise+KISSME and pairwise, respectively, using a moderate NVIDIA Quadro M4000 GPU. The slight difference is because computing the metric $\boldsymbol{M}$ is the only extra step required for training in the pairwise+KISSME. For the case of triplewise this time is 42,345 seconds.

To do complete justice, we also provide details of the experiment used to generate Fig. 5.6. We used the surveillance images of the CompCars dataset. There exist 44,481 images in 281 classes (car models). We randomly split the dataset into 140 classes for training and used the remaining 141 classes for testing. This was to ensure that there is no overlap between the training and testing images. We generated 200,000 training pairs and 60,000 testing pairs, randomly from the training and testing sets. To extract image descriptors, we computed SIFT features on a dense grid and then computed Bag Of Word representations using a dictionary of size 4096, trained by the k-means algorithm. As shown in the figure, our JDR-KISSME is superior to the base line KISSME for all embedding sizes.

5.7 Conclusions

In this chapter, we first kernelized the recently introduced KISSME algorithm. This not only enables us to deal with non-linearity in data but also provides a principal way to employ KISSME on non-vectorized data (e.g., manifold-value data). Along the way, we developed a method to project a matrix into the positive cone in an RKHS. We also developed an approximated solution based on the Nyström method towards kernelizing KISSME. Our experiments demonstrate consistent improvements of the kernelized solutions over the original KISSME and other baselines.

Furthermore, we introduced a joint dimensionality reduction technique for the KISSME algorithm, namely JDR-KISSME. Our motivation stems from the fact that the KISSME fails badly when its input is not meticulously denoised using PCA. Along the way, we formulated the solution as a Riemannian optimization problem. Based on our proposal, we also showed an end-to-end learning of a generic deep network for metric learning. Our experiments demonstrate consistent improvements of the JDR-KISSME and its deep extension over the original KISSME and state-of-the-art methods.

Chapter 6

Conclusions

In this thesis, inspired by the recent success of compact representations of data which are sources of geometric information, we first proposed a list of Riemannian coding techniques including Riemannian Vector of Locally Aggregated Descriptors (R-VLAD) Jégou et al. [2012] and Riemannian Sparse Coding (R-SC) Wright et al. [2009] for image and video classification tasks. In particular, we studied structured local descriptors from visual data, namely Region Covariance Descriptors (RCovD) Tuzel et al. [2008] and linear subspaces that reside on the manifold of Symmetric Positive Definite (SPD) matrices and the Grassmannian manifolds, respectively. In our frame-work, we not only provided a comprehensive formulation but also incorporated various well-known metrics defined on the two manifolds into our models.

We then expanded our investigations on structured descriptors by considering infinite-dimensional RCovDs Harandi et al. [2014a]; Quang et al. [2014] and Symmetric Positive Semi-Definite (SPSD) matrices, two special types of covariance based descriptors for visual data. More specifically, we made use of random Fourier feature Rahimi and Recht [2007] and the Nyström method Baker [1977] to compute to approximate the infinite-dimensional RCovDs. Using our derivation, one can seamlessly exploit the rich geometry of RCovDs and tools developed upon that such as tangent spaces to do the inference. As for the SPSD matrices, we considered their role as image set descriptors. We devised similarity measures that can be decomposed as sum of infinitesimal distances on the Grassmannian manifold and the manifold of SPD matrices. We supported our technical contributions with successful experiments on a rigorous list of challenging applications including gesture classification, video-based face recognition and dynamic scene recognition.

Lastly, we provided a principal way to employ the Keep It Simple and Straightforward MEtric learning (KISSME) Koestinger et al. [2012] algorithm on non-vectorized data. To this end, we made use of the aforementioned infinite-dimensional RCovDs and Riemannian optimization technique to obtain an exact solution along with approximated variants of our proposal. Also, we addressed the sensitivity problem of the KISSME algorithm to the input dimensionality by introducing a joint dimensionality reduction technique for the algorithm. Along the way, we formulated the solution as a Riemannian optimization problem. This is followed by end-to-end learning of a generic deep network for metric learning using our derivation.

6.1 Future Work

Since our formulation for Riemannian coding allows us to utilize any metric to construct the codes, in the future, we are interested to extend our framework to other types of Riemannian structures such as Kendall shape manifolds. Furthermore, we intend to explore how the infinite-dimensional descriptors can be extended to other types of Riemannian manifolds, such as Grassmannian manifolds.

Given the importance of metric learning in real-world scenarios, we plan to devise other variants of training a deep network using the KISSME verification signal such as triplewise+ KISSME. To further improve the accuracy of our proposed method, we also plan to benefit from more advanced mining techniques such as multi-class N-pair loss Sohn [2016].

Bibliography

ABSIL, P.-A.; MAHONY, R.; AND SEPULCHRE, R., 2009. *Optimization algorithms on matrix manifolds*. Princeton University Press. (cited on pages 79 and 81)

ALEXANDER, A. L.; LEE, J. E.; LAZAR, M.; AND FIELD, A. S., 2007. Diffusion tensor imaging of the brain. *Neurotherapeutics*, 4, 3 (2007), 316–329. (cited on page 1)

ARANDJELOVIC, O.; SHAKHNAROVICH, G.; FISHER, J.; CIPOLLA, R.; AND DARRELL, T., 2005. Face recognition with image sets using manifold density divergence. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 581–588. IEEE. (cited on page 50)

ARANDJELOVIC, R. AND ZISSERMAN, A., 2013. All about vlad. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1578–1585. IEEE. (cited on page 15)

ARSIGNY, V.; FILLARD, P.; PENNEC, X.; AND AYACHE, N., 2007. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 29, 1 (2007), 328–347. (cited on pages 1, 20, and 30)

BAKER, C. T., 1977. *The numerical treatment of integral equations*. Clarendon press. (cited on pages 3, 39, 44, 66, and 93)

BAKTASHMOTLAGH, M.; HARANDI, M.; LOVELL, B. C.; AND SALZMANN, M., 2014. Discriminative non-linear stationary subspace analysis for video classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36, 12 (2014), 2353 – 2366. (cited on pages xvii, 34, 35, and 37)

BEGELFOR, E. AND WERMAN, M., 2006. Affine invariance revisited. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2087–2094. (cited on page 24)

BERG, C.; CHRISTENSEN, J. P. R.; AND RESSEL, P., 1984. *Harmonic Analysis on Semigroups*. Springer. (cited on page 53)

BHATIA, R., 2007. *Positive Definite Matrices*. Princeton University Press. (cited on page 29)

BONNABEL, S. AND SEPULCHRE, R., 2009. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31, 3 (2009), 1055–1070. (cited on pages 41, 50, and 52)

BOUMAL, N.; MISHRA, B.; ABSIL, P.-A.; AND SEPULCHRE, R., 2014. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research (JMLR)*, 15 (2014), 1455–1459. http://www.manopt.org. (cited on page 71)

CHATFIELD, K.; SIMONYAN, K.; VEDALDI, A.; AND ZISSERMAN, A., 2014. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. British Machine Vision Conference (BMVC)*. (cited on page 88)

CHEN, J.; ZHANG, Z.; AND WANG, Y., 2015. Relevance metric learning for person reidentification by exploiting listwise similarities. *IEEE Transactions on Image Processing (TIP)*, 24, 12 (Dec 2015), 4741–4755. doi:10.1109/TIP.2015.2466117. (cited on page 62)

CHEN, S.; SANDERSON, C.; HARANDI, M. T.; AND LOVELL, B. C., 2013. Improved image set classification via joint sparse approximated nearest subspaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 452–459. IEEE. (cited on pages 41 and 51)

CHENG, D. S.; CRISTANI, M.; STOPPA, M.; BAZZANI, L.; AND MURINO, V., 2011. Custom pictorial structures for re-identification. In *Proc. British Machine Vision Conference (BMVC)*, vol. 1, 6. (cited on pages xv, 62, 66, and 73)

CHERIAN, A.; SRA, S.; BANERJEE, A.; AND PAPANIKOLOPOULOS, N., 2012. Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35, 9 (2012), 2161–2174. (cited on pages 20, 23, and 28)

CHOPRA, S.; HADSELL, R.; AND LECUN, Y., 2005. Learning a similarity metric discriminatively, with application to face verification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 539–546. IEEE. (cited on page 64)

CIMPOI, M.; MAJI, S.; KOKKINOS, I.; MOHAMED, S.; ; AND VEDALDI, A., 2014. Describing textures in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 19)

CIMPOI, M.; MAJI, S.; AND VEDALDI, A., 2015. Deep filter banks for texture recognition and segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3828–3836. (cited on pages 15 and 19)

DAVIS, J. V.; KULIS, B.; JAIN, P.; SRA, S.; AND DHILLON, I. S., 2007. Information-theoretic metric learning. In *Proc. Int. Conference on Machine Learning (ICML)*, 209–216. ACM. (cited on pages 62, 63, and 72)

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; AND FEI-FEI, L., 2009. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255. (cited on page 88)

DING, S.; LIN, L.; WANG, G.; AND CHAO, H., 2015. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition (PR)*, 48, 10 (2015), 2993–3003. (cited on pages 66 and 75)

DUAN, Y.; LU, J.; FENG, J.; AND ZHOU, J., 2017. Deep localized metric learning. *IEEE Transactions on Circuits and Systems for Video Technology*, (2017). (cited on page 64)

EDELMAN, A.; ARIAS, T. A.; AND SMITH, S. T., 1998. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20, 2 (1998), 303–353. (cited on pages 1, 10, and 81)

ELAD, M. AND AHARON, M., 2006. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing (TIP)*, 15, 12 (2006), 3736–3745. (cited on page 15)

FARAKI, M. AND HARANDI, M., 2014. Bag of riemannian words for virus classification. *Case Studies in Intelligent Computing: Achievements and Trends*, (2014), 271–284. (cited on page 14)

FARAKI, M.; HARANDI, M. T.; AND PORIKLI, F. More about vlad: A leap from euclidean to riemannian manifolds supplementary material. (cited on page 30)

FARAKI, M.; HARANDI, M. T.; AND PORIKLI, F., 2015a. Approximate infinite-dimensional region covariance descriptors for image classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 1364–1368. IEEE. (cited on pages 41 and 66)

FARAKI, M.; HARANDI, M. T.; AND PORIKLI, F., 2015b. Material classification on symmetric positive definite manifolds. In *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, 749–756. (cited on pages 1 and 41)

FARAKI, M.; HARANDI, M. T.; AND PORIKLI, F., 2015c. More about vlad: A leap from euclidean to riemannian manifolds. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4951–4960. IEEE. (cited on page 41)

FARAKI, M.; HARANDI, M. T.; AND PORIKLI, F., 2016. Image set classification by symmetric positive semi-definite matrices. In *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, 749–756. (cited on pages xvii, 1, 35, and 37)

FARAKI, M.; HARANDI, M. T.; AND PORIKLI, F., 2017a. Large-scale metric learning: A voyage from shallow to deep. *IEEE Transactions on Neural Networks and Learning Systems*, (2017). doi:10.1109/TNNLS.2017.2761773. (cited on page 61)

FARAKI, M.; HARANDI, M. T.; AND PORIKLI, F., 2017b. No fuss metric learning, a hilbert space scenario. *Pattern Recognition Letters*, 98 (2017), 83–89. (cited on page 61)

FARAKI, M.; HARANDI, M. T.; AND PORIKLI, F., 2018. A comprehensive look at coding techniques on riemannian manifolds. *IEEE Transactions on Neural Networks and Learning Systems*, (2018). doi:10.1109/TNNLS.2018.2812799. (cited on page 11)

FARAKI, M.; HARANDI, M. T.; WILIEM, A.; AND LOVELL, B. C., 2014a. Fisher tensors for classifying human epithelial cells. *Pattern Recognition (PR)*, 47, 7 (2014), 2348–2359. (cited on pages 12 and 14)

FARAKI, M.; PALHANG, M.; AND SANDERSON, C., 2014b. Log-euclidean bag of words for human action recognition. *IET Computer Vision*, 9, 3 (2014), 331–339. (cited on pages 3, 14, and 41)

FEICHTENHOFER, C.; PINZ, A.; AND WILDES, R. P., 2013. Spacetime forests with complementary features for dynamic scene recognition. In *Proc. British Machine Vision Conference (BMVC)*. (cited on page 58)

FEICHTENHOFER, C.; PINZ, A.; AND WILDES, R. P., 2014. Bags of spacetime energies for dynamic scene recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. IEEE. (cited on pages 35 and 58)

GHANEM, B. AND AHUJA, N., 2010. Maximum margin distance learning for dynamic texture recognition. In *Proc. European Conference on Computer Vision (ECCV)*, 223–236. (cited on pages xv, xvii, 13, 34, 36, and 37)

GOH, A. AND VIDAL, R., 2008. Clustering and dimensionality reduction on riemannian manifolds. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–7. IEEE. (cited on page 20)

GOLDBERGER, J.; HINTON, G. E.; ROWEIS, S. T.; AND SALAKHUTDINOV, R., 2004. Neighbourhood components analysis. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 513–520. (cited on page 63)

GOLUB, G. H. AND VAN LOAN, C. F., 1996. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA. ISBN 0-8018-5414-8. (cited on page 24)

GOLUB, G. H. AND VAN LOAN, C. F., 2012. *Matrix computations*, vol. 3. JHU Press. (cited on page 68)

GONG, Y.; WANG, L.; GUO, R.; AND LAZEBNIK, S., 2014. Multi-scale orderless pooling of deep convolutional activation features. In *Proc. European Conference on Computer Vision (ECCV)*, 392–407. Springer. (cited on pages 15 and 19)

GUILLAUMIN, M.; VERBEEK, J.; AND SCHMID, C., 2009. Is that you? metric learning approaches for face identification. In *Proc. Int. Conference on Computer Vision (ICCV)*, 498–505. IEEE. (cited on pages 62, 63, and 72)

HAMM, J. AND LEE, D. D., 2008. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proc. Int. Conference on Machine Learning (ICML)*, 376–383. ACM. (cited on pages 10, 13, 20, 51, and 54)

HARANDI, M.; HARTLEY, R.; SHEN, C.; LOVELL, B.; AND SANDERSON, C., 2015a. Extrinsic methods for coding and dictionary learning on grassmann manifolds. *International Journal of Computer Vision*, 114, 2 (2015), 113–136. (cited on pages 1, 11, 12, 15, 51, and 52)

HARANDI, M.; SALZMANN, M.; AND BAKTASHMOTLAGH, M., 2015b. Beyond Gauss: Image set matching on the Riemannian manifold of PDFs. In *Proc. Int. Conference on Computer Vision (ICCV)*, 4112–4120. (cited on page 50)

HARANDI, M.; SALZMANN, M.; AND HARTLEY, R., 2017. Joint dimensionality reduction and metric learning: A geometric take. In *International Conference on Machine Learning*, EPFL-CONF-229290. (cited on pages 2 and 62)

HARANDI, M.; SALZMANN, M.; AND PORIKLI, F., 2014a. Bregman divergences for infinite dimensional covariance matrices. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. (cited on pages 1, 2, 40, 48, 66, 68, and 93)

HARANDI, M.; SALZMANN, M.; AND PORIKLI, F., 2015c. When vlad met hilbert. *arXiv preprint arXiv:1507.08373*, (2015). (cited on page 15)

HARANDI, M. T.; HARTLEY, R.; LOVELL, B.; AND SANDERSON, C., 2016. Sparse coding on symmetric positive definite manifolds using bregman divergences. *IEEE transactions on neural networks and learning systems*, 27, 6 (2016), 1294–1306. (cited on pages 1 and 15)

HARANDI, M. T.; SALZMANN, M.; AND HARTLEY, R., 2014b. From manifold to manifold: geometry-aware dimensionality reduction for spd matrices. In *Proc. European Conference on Computer Vision (ECCV)*, 17–32. Springer. (cited on page 41)

HAYAT, M.; BENNAMOUN, M.; AND AN, S., 2014. Learning non-linear reconstruction models for image set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1915–1922. IEEE. (cited on pages xvii, 37, 41, 51, and 57)

HAYAT, M.; BENNAMOUN, M.; AND AN, S., 2015. Deep reconstruction models for image set classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37, 4 (2015), 713–727. (cited on pages 13 and 36)

HE, K.; ZHANG, X.; REN, S.; AND SUN, J., 2016. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. (cited on page 87)

HERMANS, A.; BEYER, L.; AND LEIBE, B., 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, (2017). (cited on page 64)

HO, J.; XIE, Y.; AND VEMURI, B., 2013. On a nonlinear generalization of sparse coding and dictionary learning. In *Proc. Int. Conference on Machine Learning (ICML)*, 1480–1488. (cited on page 20)

HOFFER, E. AND AILON, N., 2015. Deep metric learning using triplet network. In *International Conference on Learning Representations Workshops*, 84–92. Springer. (cited on page 64)

HOI, S. C.; LIU, W.; LYU, M. R.; AND MA, W.-Y., 2006. Learning distance metrics with contextual constraints for image retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2072–2078. IEEE. (cited on page 62)

HORN, R. A. AND JOHNSON, C. R., 2012. *Matrix analysis*. Cambridge University Press. (cited on page 30)

HU, J.; LU, J.; AND TAN, Y.-P., 2014. Discriminative deep metric learning for face verification in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1875–1882. (cited on page 64)

HU, Y.; MIAN, A. S.; AND OWENS, R., 2011. Sparse approximated nearest points for image set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 121–128. IEEE. (cited on pages 56 and 57)

HUANG, C.; LOY, C. C.; AND TANG, X., 2016. Local similarity-aware deep feature embedding. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 1262–1270. (cited on page 64)

JAAKKOLA, T. AND HAUSSLER, D., 1999. Exploiting generative models in discriminative classifiers. *Proc. Advances in Neural Information Processing Systems (NIPS)*, (1999), 487–493. (cited on page 14)

JAIN, M.; JÉGOU, H.; AND BOUTHEMY, P., 2013. Better exploiting motion for better action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2555–2562. IEEE. (cited on page 15)

JAYASUMANA, S.; HARTLEY, R.; SALZMANN, M.; LI, H.; AND HARANDI, M., 2015. Kernel methods on riemannian manifolds with gaussian rbf kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37, 12 (2015), 2464–2477. (cited on pages 52, 53, and 54)

JAYASUMANA, S.; HARTLEY, R.; SALZMANN, M.; LI, H.; AND HARANDI, M. T., 2013. Kernel methods on the Riemannian manifold of symmetric positive definite matrices. In

Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 73–80. (cited on page 12)

JÉGOU, H.; PERRONNIN, F.; DOUZE, M.; SÁNCHEZ, J.; PÉREZ, P.; AND SCHMID, C., 2012. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34, 9 (2012), 1704–1716. (cited on pages 2, 11, 12, 14, 19, 22, and 93)

KIM, M.; KUMAR, S.; PAVLOVIC, V.; AND ROWLEY, H., 2008. Face tracking and recognition with visual constraints in real-world videos. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. IEEE. (cited on pages xv, xvii, 13, 36, 37, 42, 56, and 57)

KIM, T.-K.; KITTLER, J.; AND CIPOLLA, R., 2007a. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29, 6 (2007), 1005–1018. (cited on page 51)

KIM, T.-K.; WONG, K.-Y. K.; AND CIPOLLA, R., 2007b. Tensor canonical correlation analysis for action classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. IEEE. (cited on pages xv, xvii, 42, 55, 56, and 57)

KLIPER-GROSS, O.; HASSNER, T.; AND WOLF, L., 2012. The action similarity labeling challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34, 3 (2012), 615–621. (cited on pages xvi, xvii, 77, and 78)

KOESTINGER, M.; HIRZER, M.; WOHLHART, P.; ROTH, P. M.; AND BISCHOF, H., 2012. Large scale metric learning from equivalence constraints. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2288–2295. IEEE. (cited on pages 3, 61, 62, 67, 72, 80, 87, 89, and 93)

KRIZHEVSKY, A., 2009. Learning multiple layers of features from tiny images. (2009). (cited on pages 86 and 88)

KRIZHEVSKY, A.; SUTSKEVER, I.; AND HINTON, G. E., 2012. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems (NIPS)* (Eds. F. PEREIRA; C. J. C. BURGES; L. BOTTOU; AND K. Q. WEINBERGER), 1097–1105. Curran Associates, Inc. (cited on page 64)

KYLBERG, G.; UPPSTRÖM, M.; AND SINTORN, I.-M., 2011. Virus texture analysis using local binary patterns and radial density profiles. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 573–580. Springer. (cited on pages xv, xvii, 48, and 49)

LAPTEV, I.; MARSZALEK, M.; SCHMID, C.; AND ROZENFELD, B., 2008. Learning realistic human actions from movies. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. IEEE. (cited on page 78)

LAZEBNIK, S.; SCHMID, C.; AND PONCE, J., 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2169–2178. IEEE. (cited on pages 11, 12, 13, and 14)

LE, Q.; SARLÓS, T.; AND SMOLA, A., 2013. Fastfood: approximating kernel expansions in loglinear time. In *Proc. Int. Conference on Machine Learning (ICML)*. (cited on page 41)

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; AND HAFFNER, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 11 (1998), 2278–2324. (cited on page 88)

LEE, T. S., 1996. Image representation using 2d Gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 18 (1996), 959–971. (cited on page 47)

LI, P.; WANG, Q.; ZUO, W.; AND ZHANG, L., 2013a. Log-euclidean kernels for sparse representation and dictionary learning. In *Proc. Int. Conference on Computer Vision (ICCV)*, 1601–1608. IEEE. (cited on page 56)

LI, X.; FUKUI, K.; AND ZHENG, N., 2009. Boosting constrained mutual subspace method for robust image set based object recognition. In *Proc. Int. Joint Conference on Artificial Intelligence (IJCAI)*, 1132–1137. (cited on page 50)

LI, Z.; CHANG, S.; LIANG, F.; HUANG, T.; CAO, L.; AND SMITH, J., 2013b. Learning locally-adaptive decision functions for person verification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3610–3617. (cited on page 62)

LIAO, S.; HU, Y.; ZHU, X.; AND LI, S. Z., 2015. Person re-identification by local maximal occurrence representation and metric learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2197–2206. (cited on page 87)

LIAO, Z.; ROCK, J.; WANG, Y.; AND FORSYTH, D., 2013. Non-parametric filtering for geometric detail extraction and material representation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 963–970. (cited on pages xv, xvii, 47, 48, and 49)

LIU, H.; TIAN, Y.; YANG, Y.; PANG, L.; AND HUANG, T., 2016. Deep relative distance learning: Tell the difference between similar vehicles. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2167–2175. (cited on pages 85 and 87)

LÓPEZ, M. B.; BOUTELLAA, E.; AND HADID, A. Comments on the" kinship face in the wild" data sets. (cited on page 77)

LOPEZ-PAZ, D.; SRA, S.; SMOLA, A.; GHAHRAMANI, Z.; AND SCHÖLKOPF, B., 2014. Randomized nonlinear component analysis. *arXiv preprint arXiv:1402.0119*, (2014). (cited on pages 41 and 48)

LOWE, D. G., 2004. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision (IJCV)*, 60, 2 (2004), 91–110. (cited on pages 13 and 15)

LU, J.; HU, J.; AND TAN, Y.-P., 2017. Discriminative deep metric learning for face and kinship verification. *IEEE Transactions on Image Processing (TIP)*, (2017). (cited on page 64)

LU, J.; WANG, G.; DENG, W.; MOULIN, P.; AND ZHOU, J., 2015. Multi-manifold deep metric learning for image set classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1137–1145. (cited on page 51)

LU, J.; ZHOU, X.; TAN, Y.-P.; SHANG, Y.; AND ZHOU, J., 2014. Neighborhood repulsed metric learning for kinship verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36, 2 (2014), 331–345. (cited on pages xvi, 62, 66, 76, and 77)

MAHMOOD, A.; MIAN, A.; AND OWENS, R., 2014. Semi-supervised spectral clustering for image set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 121–128. IEEE. (cited on pages 41, 55, and 56)

MIGNON, A. AND JURIE, F., 2012. Pcca: A new approach for distance learning from sparse pairwise constraints. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2666–2672. (cited on pages 2, 63, and 87)

NISHIYAMA, M.; YUASA, M.; SHIBATA, T.; WAKASUGI, T.; KAWAHARA, T.; AND YAMAGUCHI, O., 2007. Recognizing faces of moving people by hierarchical image set matching.

In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. IEEE. (cited on page 50)

OJALA, T.; PIETIKAINEN, M.; AND MAENPAA, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24, 7 (2002), 971–987. (cited on pages 34, 57, and 74)

PENG, X.; WANG, L.; QIAO, Y.; AND PENG, Q., 2014. Boosting vlad with supervised dictionary learning and high-order statistics. In *Proc. European Conference on Computer Vision (ECCV)* (Eds. D. FLEET; T. PAJDLA; B. SCHIELE; AND T. TUYTELAARS), vol. 8691 of *Lecture Notes in Computer Science*, 660–674. Springer International Publishing. ISBN 978-3-319-10577-2. doi:10.1007/978-3-319-10578-9_43. (cited on pages 12, 14, and 25)

PENNEC, X.; FILLARD, P.; AND AYACHE, N., 2006. A riemannian framework for tensor computing. *Int. Journal of Computer Vision (IJCV)*, 66, 1 (2006), 41–66. (cited on pages 1, 8, 9, 12, 15, 20, and 28)

PERRONNIN, F. AND DANCE, C., 2007. Fisher kernels on visual vocabularies for image categorization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. IEEE. (cited on pages 12 and 14)

QUANG, M. H.; SAN BIAGIO, M.; AND MURINO, V., 2014. Log-Hilbert-Schmidt metric between positive definite operators on Hilbert spaces. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 388–396. (cited on pages 2, 40, 66, and 93)

RAHIMI, A. AND RECHT, B., 2007. Random features for large-scale kernel machines. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 1177–1184. (cited on pages 3, 39, 44, and 93)

RAHIMI, A. AND RECHT, B., 2009. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 1313–1320. (cited on page 41)

ROSIPAL, R. AND TREJO, L. J., 2002. Kernel partial least squares regression in reproducing kernel Hilbert space. *Journal of Machine Learning Research (JMLR)*, 2 (2002), 97–123. (cited on page 46)

Ross, D. A.; Lim, J.; Lin, R.-S.; and Yang, M.-H., 2008. Incremental learning for robust visual tracking. *Int. Journal of Computer Vision (IJCV)*, 77, 1-3 (2008), 125–141. (cited on pages 36 and 57)

Rudin, W., 2011. *Fourier analysis on groups*. John Wiley & Sons. (cited on page 44)

Saul, L. K. and Roweis, S. T., 2003. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research (JMLR)*, 4 (2003), 119–155. (cited on page 27)

Schroff, F.; Kalenichenko, D.; and Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823. (cited on pages 64, 84, 85, and 88)

Shakhnarovich, G.; Fisher, J. W.; and Darrell, T., 2002. Face recognition from long-term observations. In *Proc. European Conference on Computer Vision (ECCV)*, 851–865. Springer. (cited on page 50)

Shroff, N.; Turaga, P.; and Chellappa, R., 2010. Moving vistas: Exploiting motion for describing scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1911–1918. IEEE. (cited on pages xv, xvii, 13, 35, 36, 37, 42, 56, 57, and 58)

Sivic, J. and Zisserman, A., 2003. Video google: A text retrieval approach to object matching in videos. In *Proc. Int. Conference on Computer Vision (ICCV)*, 1470–1477. IEEE. (cited on pages 11, 13, 14, and 18)

Sochor, J.; Herout, A.; and Havel, J., 2016. Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3006–3015. (cited on page 87)

Sohn, K., 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 1849–1857. (cited on page 94)

Song, H. O.; Xiang, Y.; Jegelka, S.; and Savarese, S., 2016. Deep metric learning via lifted structured feature embedding. (2016). (cited on pages 62, 64, 66, 84, 88, and 91)

Sra, S., 2012. A new metric on the manifold of kernel matrices with application to matrix geometric means. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 144–152. (cited on pages 9 and 13)

SRA, S. AND HOSSEINI, R., 2014. Conic geometric optimisation on the manifold of positive definite matrices. *arXiv:1312.1039*, (2014). (cited on page 23)

SUBBARAO, R. AND MEER, P., 2009. Nonlinear mean shift over Riemannian manifolds. *Int. Journal of Computer Vision (IJCV)*, 84, 1 (2009), 1–20. (cited on pages 8 and 21)

SUN, Y.; CHEN, Y.; WANG, X.; AND TANG, X., 2014. Deep learning face representation by joint identification-verification. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 1988–1996. (cited on page 64)

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; AND RABINOVICH, A., 2015. Going deeper with convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9. (cited on page 86)

THERIAULT, C.; THOME, N.; AND CORD, M., 2013. Dynamic scene classification: Learning motion descriptors with slow features analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2603–2610. IEEE. (cited on page 58)

TOSATO, D.; SPERA, M.; CRISTANI, M.; AND MURINO, V., 2013. Characterizing humans on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35, 8 (2013), 1972–1984. (cited on pages xv, xvii, 1, 13, 31, 32, 33, and 34)

TUZEL, O.; PORIKLI, F.; AND MEER, P., 2008. Pedestrian detection via classification on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30, 10 (2008), 1713–1727. (cited on pages 1, 12, 42, and 93)

VAN DER MAATEN, L., 2014. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research (JMLR)*, 15, 1 (2014), 3221–3245. (cited on page 88)

VAN GEMERT, J. C.; VEENMAN, C. J.; SMEULDERS, A. W.; AND GEUSEBROEK, J.-M., 2010. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32, 7 (2010), 1271–1283. (cited on page 14)

VEDALDI, A. AND LENC, K., 2015. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, 689–692. ACM. (cited on page 88)

VEDALDI, A. AND ZISSERMAN, A., 2012. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34, 3 (2012), 480–492. (cited on pages 44 and 72)

WAH, C.; BRANSON, S.; WELINDER, P.; PERONA, P.; AND BELONGIE, S., 2011. The caltech-ucsd birds-200-2011 dataset. (2011). (cited on pages 86 and 88)

WANG, J.; YANG, J.; YU, K.; LV, F.; HUANG, T.; AND GONG, Y., 2010. Locality-constrained linear coding for image classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3360–3367. IEEE. (cited on pages 2, 11, 17, and 26)

WANG, R.; GUO, H.; DAVIS, L. S.; AND DAI, Q., 2012. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2496–2503. IEEE. (cited on pages 1, 3, 40, 41, 46, 50, 51, 56, and 57)

WANG, R.; SHAN, S.; CHEN, X.; AND GAO, W., 2008. Manifold-manifold distance with application to face recognition based on image set. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. IEEE. (cited on page 51)

WANG, Z. AND VEMURI, B. C., 2004. An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 223–228. IEEE. (cited on pages 9, 13, 20, and 29)

WEINBERGER, K. Q.; BLITZER, J.; AND SAUL, L. K., 2005. Distance metric learning for large margin nearest neighbor classification. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 1473–1480. (cited on page 62)

WEINBERGER, K. Q. AND SAUL, L. K., 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)*, 10 (2009), 207–244. (cited on pages 2, 63, 72, and 81)

WENDEL, A. AND PINZ, A., 2007. Scene categorization from tiny images. In *31st Annual Workshop of the Austrian Association for Pattern Recognition*, 49–56. (cited on pages xv, xvii, 48, and 49)

WOLF, L.; HASSNER, T.; AND MAOZ, I., 2011. Face recognition in unconstrained videos with matched background similarity. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 529–534. IEEE. (cited on page 62)

WRIGHT, J.; YANG, A. Y.; GANESH, A.; SASTRY, S. S.; AND MA, Y., 2009. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31, 2 (2009), 210–227. (cited on pages 2, 11, 15, 17, 26, and 93)

XIE, Y.; HO, J.; AND VEMURI, B., 2013. On a nonlinear generalization of sparse coding and dictionary learning. In *Proc. Int. Conference on Machine Learning (ICML)*, 1480. NIH Public Access. (cited on pages 15 and 26)

XING, E. P.; NG, A. Y.; JORDAN, M. I.; AND RUSSELL, S., 2003. Distance metric learning with application to clustering with side-information. *Proc. Advances in Neural Information Processing Systems (NIPS)*, 15 (2003), 505–512. (cited on page 62)

XIONG, F.; GOU, M.; CAMPS, O.; AND SZNAIER, M., 2014. Person re-identification using kernel-based metric learning methods. In *Proc. European Conference on Computer Vision (ECCV)*, 1–16. Springer. (cited on pages 62, 66, 74, 75, 79, and 80)

YANG, L.; LUO, P.; CHANGE LOY, C.; AND TANG, X., 2015. A large-scale car dataset for fine-grained categorization and verification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3973–3981. (cited on pages 86 and 87)

ZHANG, L.; YANG, M.; AND FENG, X., 2011. Sparse representation or collaborative representation: Which helps face recognition? In *Proc. Int. Conference on Computer Vision (ICCV)*, 471–478. IEEE. (cited on pages 2, 11, and 18)

ZHAO, G. AND PIETIKAINEN, M., 2007. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29, 6 (2007), 915–928. (cited on page 34)

ZHENG, L.; SHEN, L.; TIAN, L.; WANG, S.; WANG, J.; AND TIAN, Q., 2015a. Scalable person re-identification: A benchmark. In *Proc. Int. Conference on Computer Vision (ICCV)*, 1116–1124. (cited on pages 86 and 87)

ZHENG, L.; YANG, Y.; AND HAUPTMANN, A. G., 2016. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, (2016). (cited on page 87)

ZHENG, W.-S.; GONG, S.; AND XIANG, T., 2009. Associating groups of people. In *Proc. British Machine Vision Conference (BMVC)*, vol. 2, 6. (cited on pages xv, xvi, 66, 71, 73, and 75)

ZHENG, W.-S.; GONG, S.; AND XIANG, T., 2015b. Towards open-world person re-identification by one-shot group-based verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38, 3 (2015), 591–606. (cited on page 62)

ZHOU, B.; LAPEDRIZA, A.; XIAO, J.; TORRALBA, A.; AND OLIVA, A., 2014. Learning deep features for scene recognition using places database. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 487–495. (cited on pages 35 and 58)

ZHOU, X.; YU, K.; ZHANG, T.; AND HUANG, T. S., 2010. Image classification using super-vector coding of local image descriptors. In *Proc. European Conference on Computer Vision (ECCV)*, 141–154. Springer. (cited on page 14)