Exploring Geometric Constraints for Learning Representations for Visual Data

By

Ankita Shukla

under the guidance of

Dr. Saket Anand
Assistant Professor, IIIT-Delhi

Indraprastha Institute of Information Technology, Delhi
October, 2020

EXPLORING GEOMETRIC CONSTRAINTS FOR LEARNING REPRESENTATIONS FOR VISUAL DATA

By

ANKITA SHUKLA

Under the guidance of
Dr. Saket Anand

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

IIIT-DELHI
Electronics and Communications Engineering

OCTOBER 2020

CERTIFICATE

This is to certify that the thesis titled **Exploring Geometric Constraints for Learning Representations for Visual Data** being submitted by **Ankita Shukla** to the Indraprastha Institute of Information Technology-Delhi, for the award of the degree of **Doctor of Philosophy**, is an original research work carried out by her under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

October 2020
Dr. Saket Anand

Indraprastha Institute of Information Technology-Delhi
New Delhi-110020

To IIIT-Delhi:

The members of the committee appointed to examine the dissertation of ANKITA SHUKLA find it satisfactory and recommend that it be accepted.

Prof. Saket Anand, IIIT-Delhi, India.
Prof. Venu Madhav Govindu, IISC Bangalore, India.
Prof. Vinay P. Namboodri, IIT Kanpur, India.
Prof. C. Chandra Sekhar, IIT Madras, India.

ACKNOWLEDGMENT

First and foremost, I would like to thank my PhD advisor, Dr Saket Anand. During my PhD, he has not only taught me to think deeply about the problems, but also academic integrity. His dedication to produce top research is an inspiration. I would also like to thank my internal committee members Dr. Pravesh Biyani and Dr. Rajiv Raman for their insightful feedback and comments.

I would like to express my sincere and heartfelt thanks to Prof. Pavan Turaga at Arizona State University with whom I have interned twice and also collaborated with. I have learned a lot from the discussions and helpful suggestions. I owe my interest in geometrical methods to him.

I would also like to thank Dr. Ryan Farrell and his students at Brigham Young University for the successful collaboration. I also want to acknowledge a fellow lab member Gullal Singh Cheema for the brainstorming discussions as well as few undergrad students that contributed to my work and learning. I feel deeply indebted to Dr. Angshul Majumdar, my Master's thesis supervisor for the support during all these years and developing my interest in the research field.

Further, I also want to thank the faculty at IIIT-Delhi from whom I have learned many valuable lessons over the years. I am also greatful for the support from the Admin Staff and Admin Facilities at IIIT-Delhi.

Lastly and most importantly, I would like to thank my family: my mother, father and my siblings for their encouragement throughout my journey. Thank you for supporting me, even though I was never really able to explain to you what have I been doing all this time and the need for the international trips.

ABSTRACT

Representation of visual data is a connecting link between the perceptual world and machine based processing. Over the decades, the computer vision community is dedicated to improving these representations, so that it can assist humans in a wide range of applications from medical imaging to visual search and face recognition systems to name a few. In this thesis, we explore geometric constraints to aid in learning representations for various computer vision applications that either have access to only limited amount of labeled training data, abundant unlabeled training data or a combination of two.

We investigate two types of geometric constraints: manifold and semantic. The contribution in this thesis can be categorized into two parts based on the geometric constraints used. In the first category, we use the geometry of manifolds. First, we use the geometry of Stiefel manifold to learn a linear transformation of feature representations in the supervised setting, and we show improved generalization in low training-data settings. We also show the same manifold constraint to be effective in the unsupervised learning of disentangled representations, which can help improve the interpretability of deep networks. The third problem is that of defense against adversarial attacks on deep networks. Using the geometry of the Grassmann manifold, we show that our subspace based representations of an adversarially perturbed input sample are sufficiently close to their clean counterparts, and can serve as a defense strategy without the need of any retraining or fine-tuning of the network.

In the second category, we make use of semantic constraints and derive a loss term that leverages the statistical manifold, i.e., the space of probability distributions. We apply this loss term in two learning scenarios. First, we use it to combat over-fitting in supervised representation learning in case of limited labeled training data for visual animal biometrics task. We show that it improves the robustness and generalization of the representations for primate face recognition as well tiger re-identification problem. Secondly, we use it for learning clusterable representations in a semi-supervised setting, where it has access to limited labeled data along with abundant unlabeled data.

In this thesis, based on the improvements across different applications and settings, we conclude that the geometric information is useful for visual data representation learning regardless of the level of supervision.

Keywords: Representation learniing, Stiefel manifold, Grassmann manifold, Statistical manifold, Visual animal biometrics.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Notations

$\mathbf{X}$	Matrix (bold capital letter)
$\mathbf{x}$	Vector (bold small letter)
x	Scalar value (small letter)
$\mathcal{X}$	Set of points
$\mathbb{R}^n$	n dimensional vector space
$\mathbb{R}^n_+$	positive orthant of n dimensional space
$\mathbb{R}^{n\times p}$	space of $n \times p$ matrices
$\mathcal{S}_{n,p}$	Stiefel manifold $\mathcal{S}_{n,p} = \mathrm{U} : \mathrm{U}^\top \mathrm{s} = \mathrm{I}$
$\mathcal{G}_{n,p}$	Grassmann manifold of linear subspaces in $\mathbb{R}^n$ of dimension p
$\mathcal{T}_{\mathbf{x}}$	Tangent Space at a point $\mathbf{X}$ on the manifold
$\mathbf{I}_n$	Identity matrix of size $(n \times n)$
$\mathrm{Tr}(\mathbf{A})$	Trace of a square matrix
$\|\|\mathbf{A}\|\|_F$	Frobenius norm of the matrix $\mathbf{A}$, $\|\|\mathbf{A}\|\|_F = \sqrt{trace(\mathbf{A}^\top \mathbf{A})}$
f	Functional mapping (small italic letters)

To my family

"Imagination is more important than knowledge.

For knowledge is limited, whereas imagination embraces the entire world,

stimulating progress, giving birth to evolution".

-Albert Einstein

Chapter One

Introduction

Learning representations from raw pixel data is crucial for the success of machine learning and deep learning systems to perform tasks like image classification, recognition and retrieval. In the last decade, supervised learning of these representations has been very successful for various tasks and even generalized well when transferred to other tasks and datasets [90]. However, obtaining a large, diverse, well-curated dataset is usually tedious and becomes a critical bottleneck in supervised learning. The recent and emerging applications like self driving cars [121, 140] and visual wildlife monitoring [50] often have limited or no labeled data and require domain experts for manually labelling the data. This has triggered research that explores representation learning techniques for scenarios where there is limited labeled data in the supervised or semi-supervised settings, or no labeled data at all in the unsupervised setting.

In this thesis, we focus on exploring geometric constraints to assist representation learning in the aforesaid settings. In the unsupervised setting, we show that the quality of the learned feature representations improves when certain geometric properties hold in the feature space, which can be imposed through manifold constraints. On the other hand, in the supervised setting, the labeled data is used to derive semantic constraints that alter the geometry of the learned feature space, making it more discriminative. By leveraging these two constraints: manifold and semantic, for various computer vision applications, the central message of

this thesis is that geometric information is vital for learning representations and provides complementary benefits to human annotated data.

We now categorize our contribution in two heads based on the type of the geometric constraints as manifold constraints and semantic constraints

1.1 Manifold Constraints

Over the last decade, manifolds have been used successfully in machine learning and various computer vision applications [17, 168, 132]. In this thesis, we are specifically interested in two matrix manifolds, the Stiefel manifold and the Grassmann manifold. The use of the Stiefel manifold in this thesis is twofold. Firstly, we have used it to improve the interpretability of the representations learned by deep networks by parametrizing the representation space with the Stiefel manifold constraint. Secondly, the geometry of the Stiefel manifold is also used for developing efficient optimization schemes. On the other hand, the Grassmann manifold is used from purely an analysis perspective to validate the solution proposed for cleaning an image from adversarial noise. The three problems that have used manifold constraint are described in the following paragraphs.

1.1.1 Linear Transformation for Representations

Mahalanobis distance metric learning is one way to learn a linear transformation and is parameterized by a symmetric positive semi-definite (PSD) matrix. The PSD matrix essentially defines the rotation and scaling of basis vectors of the feature space, that encourages semantically consistent similarity and dissimilarity measures between the transformed representations. As opposed to most traditional approaches that directly learn the PSD matrix, we propose to learn the orthonormal basis vectors and the corresponding scaling factors. We propose a joint optimization strategy that leverages the manifold constraint to efficiently optimize for the matrix of orthonormal basis vectors whereas the scaling factors are opti-

mized over the positive orthant. The manifold constraint restricts the search space of basis vectors to the space of orthonormal matrices i.e., the Stiefel manifold, effectively reducing the number of parameters to be optimized from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^2 - \frac{p(p-1)}{2})$, where n is the dimensionality of the feature space and p is the rank of the PSD matrix. In limited labeled data settings, this formulation results in improved generalization due to fewer parameters and leads to $\sim 2\%$ improvement in image classification task on several datasets over popular traditional methods.

We further validate the capabilities of the proposed framework for figure/ground segmentation of patterned species such as leopards, tigers and zebras in images captured in the wild. Segmenting the foreground in the wild is challenging due to background clutter, complex illumination effects, non-rigid variations in animal pose and occlusions. Additionally, the limited labeled data poses a challenge in employing traditional supervised approaches for segmentation, whereas interactive approaches can not extend to large databases. Therefore, we design a simple pipeline based on handcrafted features that are transformed to a new representation space with the learned metric improving the discriminability between figure and ground representations. The proposed setup outperforms traditional segmentation approaches like GrabCut [145] and Random walk [57] across patterned species.

1.1.2 Disentangled Representation Learning

As the complexity of data increases, linear methods reach their limits and are often eschewed in favor of nonlinear representation learning methods like deep neural networks (DNN). Images encountered in real world computer vision applications are usually a result of a complex interaction of different factors like illumination, pose, object size etc. that can drastically affect the perceptibility and understanding of the image. This poses a challenge in learning meaningful representations that are invariant to the factors that do not contribute towards the task at hand. This has driven the research of learning representations that can con-

trol and/or separate these factors of variations in the representation space without the need for any additional attribute annotation. These representations are termed as disentangled representations as they capture different attributes along disjoint subsets of elements of the representation vector. In order to separate the attributes in the learned representations, we propose a manifold constraint driven approach that enforces orthonormality between these subsets. These constraints can be written in closed form as a Frobenius norm to account for deviation from orthonormality condition. Similar to our work on linear transformation learning, we leverage the geometry of Stiefel manifold to explicitly enforce the orthonormality constraint. We show that our approach improves the quality of disentanglement significantly across several datasets, while empirically improving the network training convergence owing to the explicit Stiefel manifold constraint.

1.1.3 Representations to Mitigate the Effect of Adversarial Attacks

Despite their success, a crucial limitation of deep network based representations is their vulnerability to small imperceptible changes in the input images. These small changes known as adversarial perturbations [55, 125, 26, 164] have shown to alter the representations that adversely affect the network performance. We define a subspace based input transformation that rectifies the effect of such perturbations of the image. The subspace is defined on the self-correlation of random corruptions of an input image and essentially corresponds to a point on the Grassmann manifold i.e. space of fixed dimensional subspaces. This allows us to leverage the geometry of the Grassmannian and develop a proximity relationship between the subspace obtained from an adversarial sample with that of its clean counterpart and show that the geodesic distance on the manifold is upper-bounded. Alongside, we empirically show that the subspaces corresponding to similar image pairs i.e. images from the same class are closer while the subspaces of dissimilar image pairs formed by different classes are far apart, establishing the ability of our approach to rectify the wrong predictions. The proposed

approach is both model as well as training data agnostic and relies on a low dimensional representation computed only from the input image that suppresses the perturbation effect when projected back to the image space. We report an absolute improvement of $\sim 4.5\%$ in accuracy on ImageNet dataset over state-of-the-art approaches.

1.2 Semantic Constraints

Semantic constraints are used to improve semantic consistency in the representation space so as to reflect task level similarities and dissimilarities when we have access to limited training data. In an image classification task, a DNN can be viewed as a nonlinear transformation of an image to a vector of probabilities that are used to predict the class label. Typically, these networks are over-parametrized and have more number of parameters than the number of training samples resulting in over-fitting [189]. We propose to use semantic constraints in low labeled data settings to assist in improving the consistency of probability vectors generated from data that directly impact the representations. The space of probability vectors defines a manifold known as the statistical manifold, where every point on the manifold is a probability vector. We propose to minimize an approximated measure of geodesic distance on this manifold known as the KL-divergence to drive the points from the same class together while maximizing the distance between points from the different classes. We have used semantic constraints in both supervised as well as semi-supervised settings as discussed below.

1.2.1 Representations for Visual Animal Biometrics

For image recognition tasks like visual animal biometrics, *i.e.*, recognizing a specific individual animal in an image, collecting a large annotated data for training DNNs is impractical. The network trained with a cross-entropy loss on such data only encourages predicting the correct class and ignores any other information present in the distribution over all other classes. This can contribute to overfitting and a lack of consistency between predictions of

different instances from the same class. Consequently, *semantic constraints* derived from class labels are leveraged that aid representation learning for small training datasets. Specifically, we use pairwise constraints in the probability space, to encourage similar distributions for pairs with matching labels and dissimilar ones for non-matching labels. This simple regularization on the probability vectors results in a learned feature space representation, which captures the fine-grained differences between individuals, and moreover delivers promising performance in all the three protocols of visual biometric tasks, i.e., closed-set, open-set and verification. We evaluate this individual identification approach for two visual biometrics applications: primate face recognition and tiger re-identification.

1.2.2 Semi-Supervised Representation Learning for Clustering

While clustering has traditionally been an unsupervised learning problem, many DNN based clustering approaches have incorporated varying degrees of supervision, which can substantially improve clustering performance. On one hand, fully supervised approaches [99, 157] perform well but are often limiting due to unavailability of sufficient labeled data. Unsupervised approaches, while very desirable, are oblivious to any semantic notions of data, and may lead to representations that are difficult to interpret and analyze. Weakly or semi-supervised approaches are often used as a reasonable middle-ground between the two extremes. For example, [76] uses pairwise constraints as a weaker form of supervision, and [49] exploits both, labeled as well as unlabeled data for learning. Inspired by the success of semantic constraints in the supervised setting, its applicability is further explored in learning clusterable representations in the semi-supervised setting. We propose an autoencoder based semi-supervised clustering framework in deep networks that has access to only a few labeled samples along with abundant unlabeled data. The proposed approach achieved state-of-the-art results on several datasets with annotated data less than 5%.

1.3 Thesis Organization

The thesis starts with an overview of concepts and tools in Chapter 2 that are used in this thesis.

Chapter 3: Learning Representations with Linear Transformation

This chapter presents a supervised approach for learning representations with linear transformation. The proposed approach uses manifold constraint for efficient optimization and is evaluated on a range of applications.

Related Papers:

- **A.Shukla**, S. Anand, Optimization on Stiefel Manifold for Mahalanobis Distance Metric Learning (Under Review).
- **A. Shukla** and S. Anand, Metric Learning Based Automatic Segmentation of Patterned Species, ICIP 2016.
- **A. Shukla** and S. Anand, Distance Metric Learning by Optimization on the Stiefel Manifold, DIFF-CV 2015.

Chapter 4: Representations for Visual Animal Biometrics

This chapter also presents a supervised approach for representation learning. Applications in visual wildlife monitoring have scarce training data that poses challenges in learning representations using deep networks. In this chapter, we propose an approach that utilizes semantic constraints to combat the effect of over-fitting during network training as well as improve the discriminability of the representations.

Related Papers:

- **A. Shukla**, G.S. Cheema, S. Anand, Q. Qureshi, Y. Jhala, Primate Face Identification in the Wild, PRICAI 2019.
- **A. Shukla**, C. Anderson, G. S. Cheema, P. Guo, S. Onda, D. Anshumaan, S. Anand, R. Farrell, A Hybrid Approach for Tiger Re-Identification (Challenge Paper), Computer Vision for Wildlife Conservation Workshop, ICCV 2019.

Chapter 5: Semi-Supervised Representation Learning for Clustering

This chapter develops on the semantic constraints used in Chapter 4 to learn clusterable representations in semi-supervised setting in deep networks. The performance is evaluated on several image datasets and shown to outperform existing approaches.

Related Paper:

- **A. Shukla**, G.S. Cheema, S. Anand, Semi-supervised Clustering with Neural Networks, BigMM 2020, (submitted as an invited paper).

Chapter 6: Unsupervised Disentangled Representation Learning

This chapter proposes to learn a specific class of representations known as disentangled representations in deep networks in unsupervised setting. The approach utilizes manifold constraint in the representation space to improve the expressability and interpretability.

Related Paper:

- **A. Shukla**, S. Uppal, S. Bhagat , S. Anand, P. Turaga, PrOSe: Product of Orthogonal Sphere Parametrization for Disentangled Representation Learning, BMVC 2019.

Chapter 7: Low Dimensional Representation for Adversarial Defense Strategy

This chapter proposes an unsupervised approach for defining a low dimensional representation for an adversarially perturbed input image such that the reconstructed image has reduced effect of adversarial noise. The effectiveness of the approach is established by theoretical and experimental results derived by analysis on the Grassmann Manifold.

Related Paper:

- **A. Shukla**, P. Turaga, S. Anand, GraCIAS: Grassmannian of Corrupted Images for Adversarial Security, ArXiv preprint arXiv:2005.02936 (2020).

Chapter 8: Geometry of Disentangled Representation Learning Models

This chapter presents the analysis developed using both quantitative and qualitative measures to understand the geometry of latent spaces learned with disentangled representation learning models.

Related Paper:

- **A. Shukla**, S. Uppal, S. Bhagat , S. Anand, P. Turaga, Geometry of Deep Generative Models for Disentangled Representations, ICVGIP 2018.

Chapter 9: Conclusion and Future Work

This chapter concludes the thesis and presents a brief summary of the contributions. It also presents some open problems that can be explored in future.

Chapter Two

Preliminaries

This chapter gives an overview of the essential differential geometry and optimization tools that are used throughout the thesis.

2.1 Riemannian Manifolds

A manifold is a generalization of curves and surfaces to higher dimensions that is locally Euclidean i.e. every point has a neighborhood that can be mapped to an openset in $\mathbb{R}^n$. Few examples of 2d manifolds that are embedded in $\mathbb{R}^3$ are shown in Figure 2.1. A manifold together with a Riemannian metric is known as a Riemannian manifold. A Riemannian metric defines inner product on the tangent space *i.e.* $\langle \mathbf{X}, \mathbf{Y} \rangle = g(\mathbf{X}, \mathbf{Y})$, here $\mathbf{X}, \mathbf{Y} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$. One simple example of a Riemannian manifold is the n-dimensional Euclidean space $\mathbb{R}^n$ with inner product as the metric on the tangent space that is essentially $\mathcal{T}_{\mathbf{x}}\mathbb{R}^n = \mathbb{R}^n$.

The Riemannian metric is also required to define geodesics on the manifold. A geodesic is analogous to a straight line connecting two points in the Euclidean space and gives the shortest path between the two. Formally, it is an arc-length parameterized curve that is locally shortest with respect to the chosen Riemannian metric.

Mathematically, given a curve $\gamma : [a, b] \to \mathcal{M}$, from an open interval to the manifold $\mathcal{M}$,

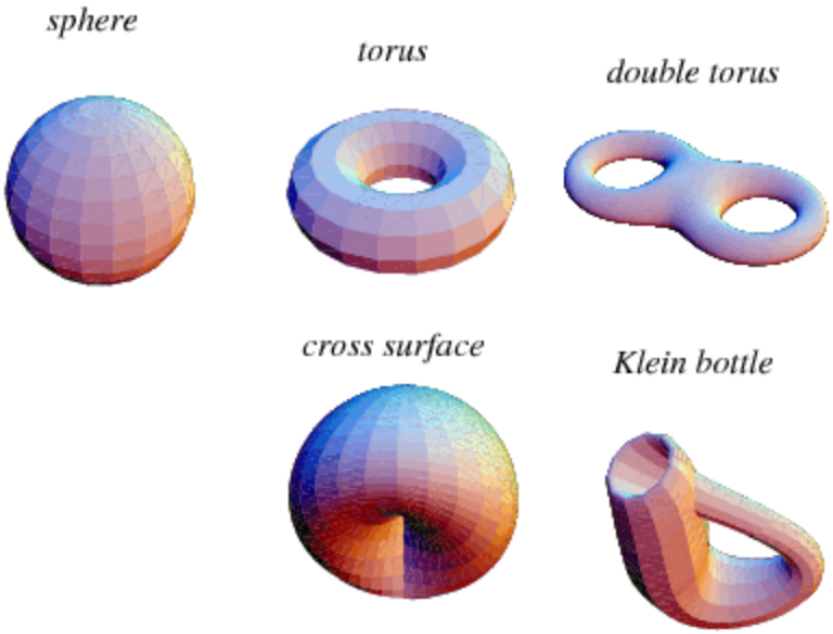


Figure 2.1 Few examples of 2d manifolds defined as non-intersecting closed surfaces in $\mathbb{R}^3$ [source:wolfram].

the arc length $L(\gamma)$ of γ is given by

$$L(\gamma) = \int_a^b |\gamma'(t)| dt \tag{2.1}$$

Here $(\gamma'(t))^2 = g(\gamma'(t), \gamma'(t))$, where $\gamma'(t)$ is the velocity vector of the curve. Following this, the geodesic distance between two points $\mathbf{X}, \mathbf{Y} \in \mathcal{M}$ is defined as the infimum of the arc length taken over all curves $\gamma : [a, b] \to \mathcal{M}$ such that $\gamma(a) = \mathbf{X}$ and $\gamma(b) = \mathbf{Y}$.

2.1.1 Stiefel Manifold

The set of $(n \times p)$ dimensional matrices with orthonormal columns, endowed with the Frobenius inner product forms a compact Riemannian manifold called the Stiefel manifold [46].

In other words, the Stiefel manifold $\mathcal{S}_{n,p}$ is defined for $n \times p$ matrices that satisfy

$$\mathcal{S}_{n,p} = \{\mathbf{U} \in \mathbb{R}^{n \times p} : \mathbf{U}^\top \mathbf{U} = \mathbf{I}_p, p \leq n\} \tag{2.2}$$

The tangent space $\mathcal{T}_\mathbf{U}$ at a point $\mathbf{U}$ on the Stiefel manifold is given by

$$\mathcal{T}_\mathbf{U}(\mathcal{S}_{n,p}) = \{\mathbf{Z} \in \mathbb{R}^{n \times p} : \mathbf{U}^\top \mathbf{Z} + \mathbf{Z}^\top \mathbf{U} = 0\} \tag{2.3}$$

2.1.2 Grassmann Manifold

The space of p-dimensional linear subspaces in $\mathbb{R}^n$ defines a Grassmann manifold [46] and is denoted by $\mathcal{G}_{n,p}$. Each point $\mathbf{U}$ on the manifold is a basis defined as a linear combination of the set of p orthonormal vectors $\mathbf{u}_1, \mathbf{u}_2 \cdots \mathbf{u}_p$. Mathematically, it can be written as

$$\mathcal{G}_{n,p} = \left\{ span(\mathbf{U}) : \mathbf{U} \in \mathbb{R}^{n \times p}, \mathbf{U}^\top \mathbf{U} = \mathbf{I}_p \right\} \tag{2.4}$$

Here, span denotes the space spanned by the column vectors of the matrix $\mathbf{U}$. So, different from the Stiefel manifold, two orthonormal matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ can denote the same point on the manifold if $span(\mathbf{U}_1) = span(\mathbf{U}_2)$. More systematically, we can say $\mathbf{U}_1\mathbf{R}_1 = \mathbf{U}_2\mathbf{R}_2$, where $\mathbf{R}_1, \mathbf{R}_2 \in \mathcal{O}_p$ where $\mathcal{O}_p = \{\mathbf{O} \in \mathbb{R}^{p \times p} : \mathbf{O}^\top \mathbf{O} = \mathbf{O}\mathbf{O}^\top = \mathbf{I}_p\}$

The tangent space $\mathcal{T}_{\mathbf{U}}$ at a point $\mathbf{U}$ on the Grassmann manifold is given by

$$\mathcal{T}_{\mathbf{U}}(\mathcal{G}_{n,p}) = \{\mathbf{Z} \in \mathbb{R}^{n \times p} : \mathbf{U}^\top \mathbf{Z} = 0\} \tag{2.5}$$

Given two orthonormal matrices $\mathbf{U}$ and $\mathbf{V}$, the principle angles $\theta_1, \theta_2 \cdots \theta_p$ are given by

$$\cos\theta_k = \max_{\mathbf{u} \in span(\mathbf{U})} \max_{\mathbf{v} \in span(\mathbf{V})} \mathbf{u}_k^\top \mathbf{v}_k \tag{2.6}$$

The cosine angles are used for computing geodesic distances. Thus, the geodesic distance between the two points $\mathbf{U}$ and $\mathbf{V}$ on the Grassmann manifold can be written as [46]

$$d_{\mathcal{G}}^2(\mathbf{U}, \mathbf{V}) = \Sigma_i \, \theta_i^2 \tag{2.7}$$

The principle angles can be computed by the singular value decomposition of the matrix $\mathbf{U}^\top \mathbf{V} = \mathbf{P}(\cos\Theta)\mathbf{Q}^\top$. Here, $\cos\Theta$ is a diagonal matrix of cosine principle angles. The columns of $\mathbf{P}$ and columns of $\mathbf{Q}$ are the left and right singular vectors of $\mathbf{U}^\top \mathbf{V}$ respectively.

2.1.3 Statistical Manifold

A statistical manifold [139] is a Riemannian manifold, where every point on the manifold is a probability distribution. Consider a family of probability functions parameterized by

$\theta = [\theta_1, \cdots, \theta_n]$. The collection of such probability functions, where each function is indexed by a point $\theta \in \mathbb{R}^n$ defines a statistical manifold given by

$$\mathcal{S} = \{p(x;\theta) : \theta \in \mathbb{R}^n\} \tag{2.8}$$

$\mathcal{S}$ is a subset of $\mathcal{P}(X)$, the set of all probability measures on X is given by

$$\mathcal{P}(X) = \left\{ p : X \to \mathbb{R},\ p(x) > 0, (\forall x \in X); \int_X p(x)dx = 1 \right\} \tag{2.9}$$

The tangent space at a point p_θ on the manifold is given by

$$\mathcal{T}_\theta(\mathcal{S}) = \left\{ \sum_{i=1}^{n} \alpha_i \partial_i : \alpha_i \in \mathbb{R} \right\} \tag{2.10}$$

Here, $\partial_i = \frac{\partial}{\partial \theta_i}$. The Riemannian metric defined on this manifold is the Fisher-Rao Information metric. It is proportional to the amount of information that the distribution contains about the parameters. So, the Fisher-Rao Information matrix is given by

$$g_{i,j}(\theta) = E[\partial_i l_\theta \partial_j] \tag{2.11}$$

Here, $l_\theta = \log p(x;\theta) = p_\theta$. Thus, the geodesic distance can be computed between points on the statistical manifold by substituting the metric in Eq. 2.1. However, in general, the parametrization of the probability density functions (pdfs) of the data is not known, making Fisher-Rao metric unsuitable for computing distances between probability distributions in practical scenarios [8].

Therefore, approximations of the Fisher-Rao metric have been developed [85] (e.g. Hellinger divergence, certain Ali-Silvey divergences and Kullback-Leibler-divergence). The one we are concerned in this thesis is the Kullback-Leibler divergence (known as the KL-divergence).

The KL-divergence provides an approximation of Fisher-Rao Information metric based geodesic distance. It can be used to compute the difference between two probability density functions in the infinite-dimensional function space or finite-dimensional vector space. Given two distributions $\mathbf{p}$ and $\mathbf{q} \in \mathbb{R}^n$, the KL-divergence is given by

$$d_{KL}(\mathbf{p}||\mathbf{q}) = \sum_{k=1}^{K} p_k \log \frac{p_k}{q_k} \tag{2.12}$$

The KL-divergence between two close distributions closely approximate the Fisher-Rao Information metric by the relation $2d_{KL}(\mathbf{p}||\mathbf{q}) \rightarrow (d_{FD}(\mathbf{p},\mathbf{q}))^2$, where d_{FD} is the Fisher-Rao distance [85].

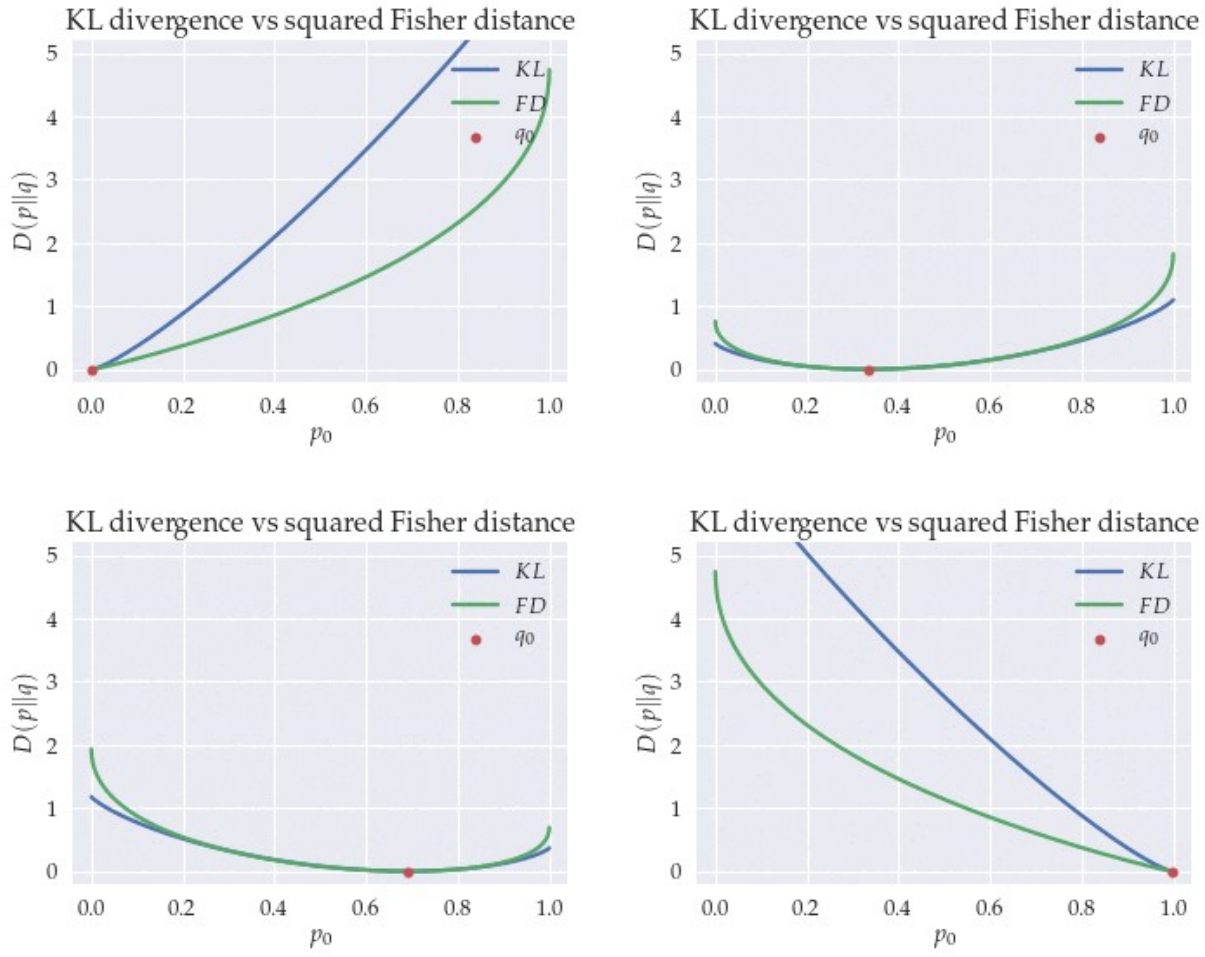


Figure 2.2 Comparison of KL-divergence (KLD) with Fisher Distance (FD) between two discrete probability distributions $\mathbf{p} = [p_0, p_1]$ and $\mathbf{q} = [q_0, q_1]$, where $\mathbf{q}$ is defined as a function of p_0 for different values of q_0 [1].

Even when $\mathbf{p}$ and $\mathbf{q}$ are different and far apart, the KL-divergence approximates the geodesic distance well. An example to illustrate the same is shown in Figure 2.2 for two discrete probability distributions $\mathbf{p} = [p_0, p_1]$ and $\mathbf{q} = [q_0, q_1]$. The distances are plotted by considering $\mathbf{q}$ as a function of p_0 for different values of q_0. The KL-divergence approximates the geodesic fairly well and drifts apart when KL-divergence goes to infinity i.e. with $p_0 =$

$1, q_0 = 0$.

2.2 Optimization on Manifolds

A Riemannian optimization approach consists of two steps in each iteration (i) find a tangent vector in the direction of descent (ii) apply a retraction that maps the tangent vector back to the manifold. A retraction is a projection mapping from the tangent space onto the manifold to keep the new point on the manifold. In the case of Stiefel manifold, polar decomposition and QR decomposition are the most commonly used retractions [4]. Recently, many practical algorithms like [83, 173] have been developed that are more computationally efficient. In our work, we use Cayley transform based gradient descent approach [173] for optimizing a loss function on the Stiefel manifold.

Given an optimization problem $\arg min_{\mathbf{X} \in \mathcal{S}_{n,p}} f(\mathbf{X})$, we want to solve for a $\mathbf{X} \in \mathcal{S}_{n,p}$ that minimizes the function $f(\mathbf{X})$.

The gradient on the manifold can be obtained by projecting the ambient Euclidean space gradient to the tangent space.

$$\nabla f = Proj_{\mathbf{X}} \nabla f(\mathbf{X}) \tag{2.13}$$

Here, the projection operator $Proj_{\mathbf{X}}$ is defined to project a matrix $\mathbf{Y}$ onto the tangent space as follows :

$$Proj_{\mathbf{X}} \mathbf{Y} = \mathbf{Y} - \mathbf{X} Sym(\mathbf{X}^\top \mathbf{Y}) \tag{2.14}$$

Therefore, the gradient on the manifold is defined as

$$\nabla f(\mathbf{X}) = \mathbf{G} - \mathbf{X} Sym(X^\top \mathbf{G}) \tag{2.15}$$

Here, $\mathbf{G} = \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}$ is the gradient in the Euclidean space. Sym() defines the symmetric matrix for any given matrix *i.e.* $Sym(B) = \frac{1}{2}(\mathbf{B} + \mathbf{B}^\top)$. Now, we adapt the Cayley transformation

to define the retraction. For all $\mathbf{Z} \in \mathcal{T}_{\mathbf{x}}\mathcal{S}_{n,p}$, it holds

$$\mathbf{Z} = \mathbf{W}\mathbf{X} \tag{2.16}$$

Here, $\mathbf{W}$ is given as follows

$$\mathbf{W}_{\mathbf{z}} = \mathbf{G}\mathbf{X}^{\top} - \mathbf{X}\mathbf{G}^{\top} \tag{2.17}$$

This is used to define the retraction on the Stiefel manifold, that is given by

$$\mathcal{R}_{\mathbf{x}}(\tau) = \big(\mathbf{I} + \frac{\tau}{2}\mathbf{W}\big)^{-1}\big(\mathbf{I} - \frac{\tau}{2}\mathbf{W}\big)\mathbf{X} \tag{2.18}$$

Here, τ is the step size. The curve defined by the retraction in the above equation is smooth. The step size is obtained by a curvilinear search algorithm in the direction of descent in the tangent space. The update on the manifold is then obtained using the Cayley transformation based retraction to minimize the objective function. We employed the method developed in [173] that proposed a monotone curvilinear algorithm and used Barzilai-Borwein step size [12] to achieve faster convergence.

2.3 Deep Learning Overview

A deep neural network (DNN) processes an input and applies a series of transformations to progressively extract representations suitable for a given task. These transformations are referred as layers with the first layer: input, last layer: output and the series of transformations as hidden layers.

2.3.1 CNNs

A Convolutional Neural Network (CNN) is the most widely used neural network that operates mainly on inputs that have grid structure such as the audio signals in 1d, images in 2d and videos in 3d. The key operation in a CNN is convolution. A CNN is made of multiple

convolution layers, to progressively learn higher level representations. Each layer consists of a set of filters, a nonlinear function and a pooling operation. The convolution layers are followed by a fully connected layer that produces the desired output. An example of CNN architecture is shown in Figure 2.3. The output of the network is used in a loss

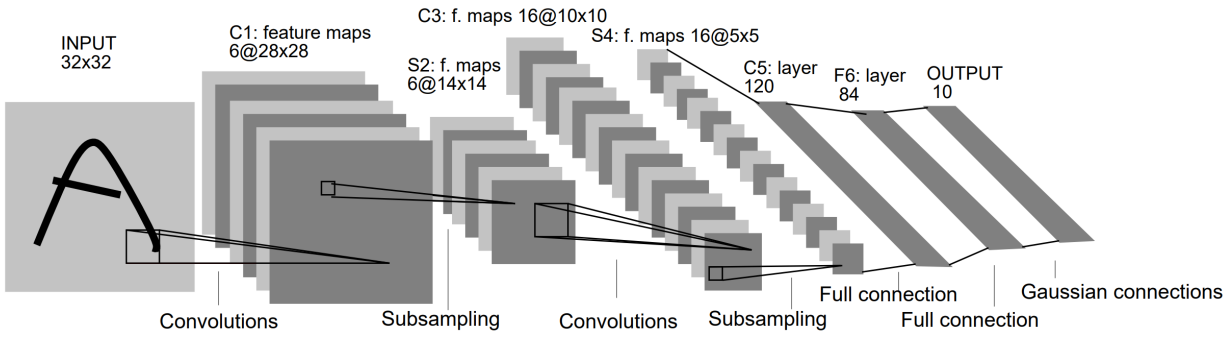


Figure 2.3 An example of CNN architecture, LeNet [100]

function that quantifies a measure of deviation from the desired output and is used to train the network. These parameters are learnt using a mini-batch stochastic gradient descent (SGD) that iteratively updates the network parameters starting with some initial (random) parameters.

2.3.2 Autoencoders

An autoencoder is a neural network defined with an encoder-decoder pair, where the encoder is designed to map an image to a representation $\mathbf{z} = f(\mathbf{x}, \theta)$ and the decoder function g transforms it back to input image $\hat{\mathbf{x}} = g(\mathbf{z}, \phi)$. The parameters are learnt to minimize the following loss function:

$$\mathcal{L}(\mathbf{x}, g(f(\mathbf{x}))) = ||\hat{\mathbf{x}} - \mathbf{x}||_2^2 \tag{2.19}$$

Autoencoders are trained in a similar way as convolution neural networks, but with the unsupervised loss given by the above equation. They have been used widely over several decades for a range of applications: layer wise greedy pre-training, dimensionality reduction

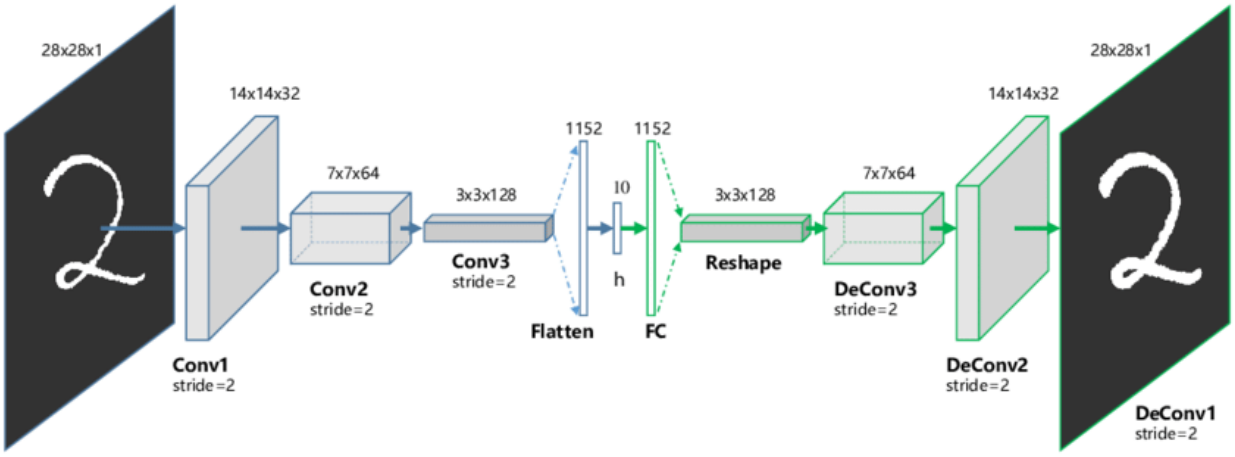


Figure 2.4 An example of convolutional autoencoder [65].

and image denoising, to name a few. In this thesis, we are concerned with convolutional autoencoders that have a sequence of convolution and deconvolution layers in the encoder and decoder network respectively. During network training, the encoder network learns to process an image to a low dimensional representation $\mathbf{z}$ and the decoder learns to reconstruct the image at the output. An example of a convolution autoencoder is shown in Figure 2.4.

2.3.3 Generative Models

Deep generative models learn to approximate the data distribution by modeling the latent variables and a generator function that maps the latent variables to the data manifold. The two generative models used in this thesis are: Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs).

Variational Autoencoders and Variants

As opposed to autoencoders that do not impose any constraint on latent variables, VAE's [88] add a generative probabilistic formulation to approximate the probability distribution of the training data. VAEs have a stochastic encoder network to model the posterior distribution $q(\mathbf{z}|\mathbf{x})$ and a decoder network or a generator that models the conditional log likelihood

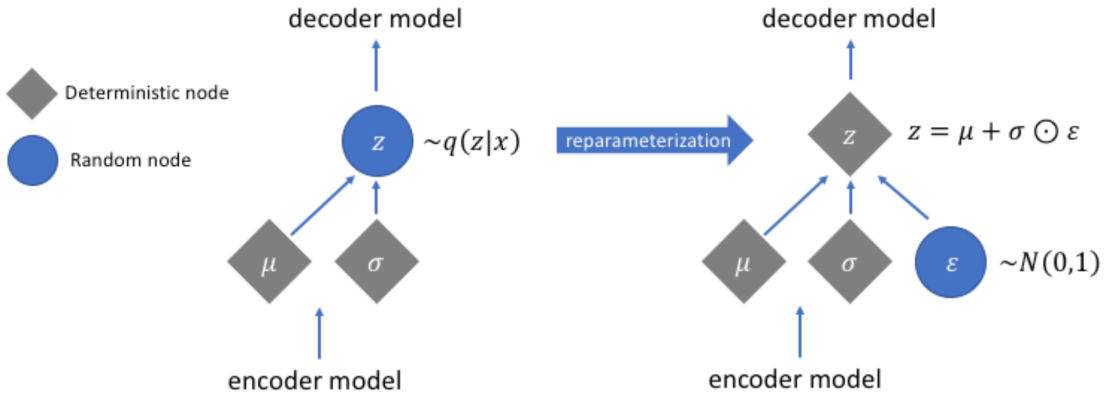


Figure 2.5 Illustration of the re-parametrization trick in VAEs [2].

$logp(\mathbf{x}|\mathbf{z})$.

The objective function that a VAE is trained to optimize is given by

$$\mathcal{L}(\mathbf{x};\theta,\phi) = -\mathbb{E}[log_{q_\phi(\mathbf{z}|\mathbf{x})}(p_\theta(\mathbf{x}|\mathbf{z}))] + d_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{2.20}$$

Here, the generator mapping parametrizes the data distribution and the posterior distribution $q_\theta(\mathbf{z}|\mathbf{x})$ parametrizes the encoder. The encoder is constrained to be a Gaussian and the unknown posterior distribution term $p_\phi(\mathbf{x}|\mathbf{z})$ defines the decoder. Here, θ and ϕ are the encoder and decoder parameters respectively. From an implementation point of view, the encoder of VAE will output the parameters of the distribution. Since we parametrize the latent space with a Gaussian distribution, it outputs mean and variance vectors describing the distribution. The decoder can then generate an image at the output by sampling from this distribution. But, the sampling process is not differentiable limiting the network training through the backpropagation algorithm. To tackle this, VAEs leverage *reparametrization trick* that instead samples η from a unit Gaussian which is then shifted by the latent distribution mean and scaled by the variance. This in-effect moves the non-differentiability out of the network, allowing to train the model with backpropagation. An illustration of the same is shown in Figure 2.5.

β-VAE [124] is a modified VAE that increases the weight on the KL-divergence between the variational posterior and the prior by $\beta > 1$ to enforce disentanglement. The modified

objective function is given as follows:

$$\mathcal{L}(\mathbf{x};\theta,\phi) = -\mathbb{E}[log_{q_\phi(\mathbf{z}|\mathbf{x})}(p_\theta(\mathbf{x}|\mathbf{z}))] + \beta d_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{2.21}$$

While it improves the quality of disentanglement, the reconstruction quality degrades as compared to VAE. Building on this work, Burgess *et. al.* viewed posterior distribution as information bottleneck and added a term to the loss function that works as an upper bound on the information that $\mathbf{z}$ captures of $\mathbf{x}$. The proposed modification is as follows

$$\mathcal{L}(\mathbf{x};\theta,\phi) = -\mathbb{E}[log_{q_\phi(\mathbf{z}|\mathbf{x})}(p_\theta(\mathbf{x}|\mathbf{z}))] + \beta|d_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - C| \tag{2.22}$$

Here, C is a real value that increases during training, allowing the divergence term to increase without impacting the overall loss.

Factor-VAE [86] improves β-VAE by augmenting the objective function with a term to enforce independence in the code distribution.

$$\mathcal{L}(\mathbf{x};\theta,\phi) = -\mathbb{E}[log_{q_\phi(\mathbf{z}|\mathbf{x})}(p_\theta(\mathbf{x}|\mathbf{z}))] + d_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) + \gamma d_{KL}(q(\mathbf{z})||\bar{q}(\mathbf{z})) \tag{2.23}$$

Here, the last term is known as TC (total correlation), where $\bar{q}(\mathbf{z}) = \prod_{j=1}^{d} q(z_i)$. Owing to the intractability of both $q(\mathbf{z})$ and $q(\hat{\mathbf{z}})$, the term is minimized using density ratio trick by training a discriminator jointly with the VAE.

Generative Adversarial Networks (GANs)

GANs [54] are generative models with two competing neural networks: a discriminator and a generator that are trained together to learn the data distribution in a min-max game. The discriminator tries to distinguish the samples generated from the data distribution from the noise samples while the generator gradually learns to generate samples that can fool the discriminator.

Chapter Three

Learning Representations with Linear Transformation

Given a set of images, one can compute representations either by naively vectorizing the images or by extracting the handcrafted features. These representations are often inadequate for a downstream task like image classification or clustering as they are unaware of the notion of similarity or dissimilarity specific to the task. Learning a linear transformation is one simple way of adapting these representations for the task at hand. This has been shown to improve the performance of many applications such as face verification [33], person re-identification [89], image classification [171] and image annotation [62].

Mahalanobis distance metric learning is one way to characterize this linear transformation. The Mahalanobis distance metric is a generalization of the Euclidean distance and is parametrized by a symmetric positive semidefinite (PSD) matrix. Learning the Mahalanobis distance amounts to learning a transformed input space that ensures that similar points are closer, while dissimilar points are farther apart. The notion of similarity and dissimilarity is based on the semantics of the application.

Most often, the positive semidefiniteness of the matrix is maintained by projecting it onto the PSD cone at every iteration [39, 80, 94, 98] that requires computing eigenvalue decomposition. However, doing so becomes intractable in medium and high dimensional

spaces due to the computational cost of eigenvalue decomposition $\mathcal{O}(n^3)$, where n is the dimension of the data. To this end, several approaches made use of special regularizers like the logDet divergence function [39], [93] that implicitly ensured positive semi-definiteness due to the scale invariance and range space preservation properties of the function. However, these approaches are limited in their choices of regularizer suitable for the task and the rank of the solution is restricted to that of the initial solution. This motivates the two requirements for a metric learning framework: that can efficiently maintain the positive semi-definiteness and the flexibility to different regularizers, making it suitable for the application at hand. In this chapter, we propose EigMetric framework for metric learning that fulfills these two requirements by using an eigenvalue decomposition parametrization and the geometry of the Stiefel Manifold.

Key Highlights

- *Propose an optimization framework to learn the PSD matrix by alternately optimizing for the eigenvectors on the constrained space of orthonormal matrices* i.e. *the Stiefel Manifold and the corresponding eigenvalues on the positive orthant.*
- *EigMetric is flexible to incorporate different convex spectral function based regularizers suitable for different applications with minimal changes in the update strategy.*
- *EigMetric is flexible to choice of distance constraints* i.e *triplet or pairwise distance constraints based on the requirement of given task.*
- *EigMetric achieves improved generalization performance in limited labeled data over traditional approaches for various applications.*

3.1 Background

In this section, we provide a brief overview of the Mahalanobis distance metric and the convex spectral functions required in our framework.

3.1.1 Mahalanobis Distance Metric

Given data points $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^n$, the Mahalanobis distance is given by

$$d_{\mathbf{M}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^\top \mathbf{M}(\mathbf{x} - \mathbf{y}) \tag{3.1}$$

For any function d to be a metric, it has to satisfy these conditions:

$$d_{\mathbf{M}}(\mathbf{x}, \mathbf{y}) \geq 0 \qquad \text{Non-negativity} \tag{3.2}$$

$$d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y} \qquad \text{identity of indiscernibles} \tag{3.3}$$

$$d_{\mathbf{M}}(\mathbf{x}, \mathbf{y}) = d_{\mathbf{M}}(\mathbf{y}, \mathbf{x}) \qquad \text{symmetry} \tag{3.4}$$

$$d_{\mathbf{M}}(\mathbf{x}, \mathbf{z}) = d_{\mathbf{M}}(\mathbf{x}, \mathbf{y}) + d_{\mathbf{M}}(\mathbf{y}, \mathbf{z}) \qquad \text{triangle inequality} \tag{3.5}$$

Now, for the Mahalanobis distance $d_{\mathbf{M}}$ to be a valid metric, it needs to fulfill the above conditions. A $n \times n$ real-valued symmetric positive definite matrix denoted as $\mathbf{M} \succ 0$ fulfills all the conditions. Therefore with a symmetric positive definite matrix $\mathbf{M}$, $d_{\mathbf{M}}(\mathbf{x}, \mathbf{y})$ is a distance metric.

However, strictly ensuring the symmetric positive definiteness is not easy to guarantee. For this reason the identity of indiscernibles condition is usually relaxed into a condition of identity that permits $d(\mathbf{x}, \mathbf{y}) = 0$ even when $\mathbf{x} \neq \mathbf{y}$ *i.e.* symmetric positive semidefinite matrix. This relaxed condition along with the rest of the conditions, define a pseudo metric. Thus, Mahalanobis distance with positive semi-definite (PSD) matrix *i.e.* $\mathbf{M} \succeq 0$ is a pseudo metric, which is shown to be enough in most applications.

3.1.2 Convex Spectral Functions

Most Mahalanobis metric learning approaches [80, 39, 126, 111] include a regularization term in the objective function to ensure desired properties on learned metric and impact model complexity. Popularly used regularizers in metric learning algorithms belong to the class of convex spectral functions.

Definition: A convex spectral function is defined as $f : \mathcal{S} \rightarrow \mathbb{R}$ for a $n \times n$ symmetric matrix $\mathbf{A}$, for which $f(\mathbf{A}) = \psi(\lambda_1, \ldots, \lambda_n)$ where $\lambda_i,\ i = 1, \ldots, n$ are the eigenvalues of $\mathbf{A}$ and $\psi : \mathbb{R}^n \rightarrow \mathbb{R}$ is a real-valued symmetric scalar function of n variables.

We show that our framework is flexible to the choice of any convex spectral function as regularizer. To this end, we show the effect with three convex spectral regularizers on $\mathbf{M}$ in our framework: Frobenius norm, trace norm and logDet divergence function.

$$\text{Frobenius norm} : \mathcal{R}(\mathbf{M}) = ||\mathbf{M}||_F^2 \tag{3.6}$$

$$\text{Trace norm} : \mathcal{R}(\mathbf{M}) = Tr(\mathbf{M}) \tag{3.7}$$

$$\text{LogDet divergence} : \mathcal{R}(\mathbf{M}) = Tr(\mathbf{M}\mathbf{M}_0^{-1}) - \log\det(\mathbf{M}\mathbf{M}_0^{-1}) - n \tag{3.8}$$

Here, in Eq. (3.6), the squared Frobenius norm regularizer is the matrix analog of standard l_2 regularizer on a vector. It enforces to learn $n(n+1)/2$ independent parameters, where n is the dimension of the data. While Frobenius norm is good for low dimensional data, it is prone to over-fitting when n is large and only has limited training data. *Trace norm* regularizer in Eq. (3.7) where $tr()$ denotes the trace of the matrix *i.e* sum of eigenvalues of the matrix, can limit the number of independent parameters by promoting a low rank solution. Low rank solutions exploit data correlation and are often desired in metric learning approaches. *LogDet Divergence* regularizer is given by Eq. (3.8). Here, $\mathbf{M}_0$ denotes an initial estimate of the solution. The regularizer focuses on learning a $\mathbf{M}$ that is close to $\mathbf{M}_0$. In Eq. (3.8), $\mathbf{M}_0$ is the identity matrix, but it can be generalized to arbitrary positive definite matrices. LogDet

divergence preserves the rank as defined by $\mathbf{M}_0$. The solution is also scale and translation invariant implicitly owing to the property of divergence function.

The flexibility of our approach to these regularizers is due to the proposed eigendecomposition parametrization for the PSD matrix $\mathbf{M}$ that further simplifies these functions. A more detailed description of the same will follow in Section 3.3.2.

3.2 Existing Literature

Most approaches in linear transformation learning focus on Mahalanobis distance metric that is parameterized by a PSD matrix. The survey articles [184, 93, 15, 14] discuss some of the important developments in traditional metric learning approaches and can be referred for a detailed understanding of the literature. In this chapter, we categorize a few works from the literature based on the way the positive semidefiniteness of the matrix is ensured under three heads: projection based approaches, projection free approaches and geometric approaches

Projection based methods The approaches in this category perform a projection onto the PSD cone at every iteration to ensure that the learned matrix is PSD. The projection step requires the computation of the eigenvalue decomposition, that scales cubic in the feature dimension. Xing *et al.* [179] proposed a convex objective function to maximize the sum of distances between dissimilar pairs while maintaining an upper bound on the sum of squared distance between similar pairs. On the other hand, Large Margin Nearest Neighbor (LMNN) [171] proposed to minimize the distances between intra-class neighbors while separating samples from different classes with a large distance margin. However, the lack of a regularization term in LMNN made it prone to over-fitting and restricted other desirable properties, e.g., learning a low rank metric. In practice, LMNN avoided over-fitting by applying cross validation for each dataset to carefully determine an early stopping criterion. The recent work [130] learned metric in the feature space induced by a nonlinear

function along with a low rank constraint. However, it resorted to a projection step at every iteration for the PSD constraint. These approaches are intractable for medium and high dimensional data. Therefore, [111] followed a two step strategy by first projecting it onto a low dimensional subspace, effectively reducing the dimension for metric learning. Several other methods used regularizer functions to promote a low rank solution. For example, the work in [192] used a regularizer to bias the solution to lie on a low dimensional manifold. Another work, Law *et. al* [98] instead used a regularizer to provide explicit control on the rank of the learned metric.

Projection Free Approaches To overcome the computational overhead of eigenvalue decomposition, methods like [39], [93] used logDet divergence function as a regularizer. These approaches learned a matrix $\mathbf{M}$ that had minimum divergence with a given PSD matrix. The learned metric is ensured to be a PSD matrix due to the one of the few properties of the LogDet function that includes scale invariance and preservation of range space. Thus, the projection step is omitted to ensure the PSD constraint. Further, similar to projection based approaches, a low rank matrix can also be learned in this framework as discussed in [94, 40].

Geometry Driven Approaches Sparse Compositional Metric Learning (SCML) [152] exploited the geometry of the training data to compute a set of basis vectors that are used to construct locally discriminative rank one matrices. The final learned metric is represented as a weighted sum of these rank one matrices, reducing the metric learning problem to solving for the sparse, non-negative weights. However, the lack of a regularization term and the dependency on a fixed basis set makes SCML prone to over-fitting, especially in low training data settings.

While SCML exploited data geometry, more recent approaches leveraged the geometry of Riemannian manifolds. The prime advantage of using manifolds is that a constrained optimization problem in Euclidean space is solved as an unconstrained problem by harnessing the intrinsic geometry of the manifold. For example, [126] optimized a logistic loss function

with Frobenius norm regularization over the manifold of fixed rank matrices. It combined gradient descent with an efficient retraction step to obtain fast updates on the Riemannian manifold of fixed rank matrices. However, the resulting matrix is not PSD and required an eigendecomposition at each iteration. [188] utilized the Riemannian geometry of PSD matrices and proposed a closed form solution for Mahalanobis distance metric. While this approach reduced the run time by many orders, the classification accuracy is not improved across various datasets. Later, [68] developed joint dimensionality reduction and metric learning framework by optimizing on the quotient space of the product space of two Riemannian manifolds with the orthogonal group. Following this, work in [106] developed a multi-modal geometric metric learning approach to learn distance metric for each view of the data. More recently, [56] developed a new representation for data using Grassmann manifold, followed by using LDA to improve the discriminability of the features. Further, to deal with more complicated data, they augmented the proposed method with LMNN [171] to employ the strengths of both the approaches.

Deep Learning Approaches Over the last decade, with deep learning approaches achieving state-of-the-art performances for various machine learning and computer vision applications, several deep metric learning approaches have been proposed [77, 186, 158, 108]. These approaches outperformed non-deep learning approaches due to their capability to learn complex nonlinear transformations of the data that makes data linearly separable. However, these approaches relied on huge training data to avoid over-fitting due to large number of parameters. However, these approaches are not compared with our approach due to following reasons. Firstly, our work learns a linear transformation, whereas deep networks learn nonlinear transformations. And secondly, we addressed the issue of learning in small training data setting and deep networks require pre-training on on large dataset owing to large number of parameters as opposed to the method compared in this work. Further, more recently the works [47, 127] pointed out that the progress in deep metric learning is biased with unfair evaluation caused due to change in network architecture and hyperparamter tuning. This

disparity among deep learning approaches further increases the gap between deep learning approaches and linear methods.

3.3 EigMetric Framework

3.3.1 Proposed Parametrization

Given data points $\mathbf{x}$, $\mathbf{x}' \in \mathbb{R}^n$, the squared Mahalanobis distance is given by

$$d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^\top \mathbf{M} (\mathbf{x} - \mathbf{x}') \tag{3.9}$$

Here, $\mathbf{M} \succeq 0$ is a symmetric positive semidefinite (PSD) matrix. A DML approach aims to learn this PSD matrix $\mathbf{M}$ to compute the Mahalanobis distance between the two points. The Mahalanobis distance function is equivalent to the Euclidean distance between data points under a linear transformation, which in turn is characterized by a rotation and scaling applied on the input space. To understand this aspect of the metric, we write the matrix $\mathbf{M}$ by its eigenvalue decomposition as follows:

$$\mathbf{M} = \mathbf{U}\mathbf{W}\mathbf{U}^\top \tag{3.10}$$

So, rewriting (3.9) with this substitution, the rotation and scaling components of Mahalanobis matrix $\mathbf{M}$ can be closely understood as follows:

$$\begin{aligned} d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') =& (\mathbf{x} - \mathbf{x}')^\top \mathbf{U}\mathbf{W}\mathbf{U}^\top (\mathbf{x} - \mathbf{x}') \\ =& (\mathbf{U}^\top \mathbf{x} - \mathbf{U}^\top \mathbf{x}')^\top \mathbf{W} (\mathbf{U}^\top \mathbf{x} - \mathbf{U}^\top \mathbf{x}') \end{aligned} \tag{3.11}$$

Thus, learning the PSD matrix $\mathbf{M}$ corresponds to learning an orthonormal matrix $\mathbf{U}$ *i.e.* the rotation matrix, that essentially removes the coupling among the feature dimensions. Whereas, learning $\mathbf{W}$ assigns weights to different feature dimensions to establish their importance. For convenience of notation, during the development of the optimization problem,

we will alternately, but unambiguously use the vector representation $\mathbf{w} \in \mathbb{R}^p_+$ such that $\mathbf{w} = Diag(\mathbf{W})$ and one can simply define $\mathbf{M}$ as

$$\mathbf{M} = \sum_{i=1}^{n} w_i \mathbf{u}_i \mathbf{u}_i^\top \tag{3.12}$$

Here, $w_i \geq 0$, are non-negative eigenvalues corresponding to eigenvectors $\mathbf{u}_i$'s of the PSD matrix $\mathbf{M}$. Based on the premise that most often data lies in a lower dimensional subspace with dimension $p \leq n$, we explicitly parameterize rank-p PSD matrix $\mathbf{M}$ in matrix notation as $\mathbf{M} = \mathbf{U}\mathbf{W}\mathbf{U}^\top$ where $\mathbf{U}$ is $n \times p$ matrix with eigenvectors as columns and $\mathbf{W}$ is a $p \times p$ diagonal matrix with eigenvalues along the diagonal.

We now present a simple example to show the effect of transformation defined by learning the eigenvectors and eigenvalues for 2 dimensional synthetic data in Figure 3.1. The data consists of 2 classes, the original spread of the data in ambient Euclidean space is shown in Figure 3.1(a), with canonical basis vectors along the 2 dimensions represented as blue and black arrows. This represents $\mathbf{M} = \mathbf{I}$ *i.e.* the canonical basis scaled with unit eigenvalues $\mathbf{w} = [1, 1]^\top$. Figure 3.1(c) shows transformed data with the learned metric after few iterations, the corresponding $\mathbf{U}$ and $\mathbf{w}$ are given in Figure 3.1(b). Here, the direction and magnitude of the vectors denote the rotation (eigenvectors) and scaling (eigenvalues) parameters respectively. The data transformed with the final learned metric is shown in 3.1(e). The larger magnitude in one of the basis directions shows that the class specific data can be distinguished along this direction.

Parameter Reduction

Mahalanobis distance metric learning algorithms focus on learning the matrix $\mathbf{M}$ such that the PSD constraint is satisfied on the distance matrix. Traditionally, one requires to solve for n^2 or np parameters for full rank and p-rank matrices respectively. With the eigenvalue decomposition parametrization, that utilizes the geometric constraint of the Stiefel manifold,

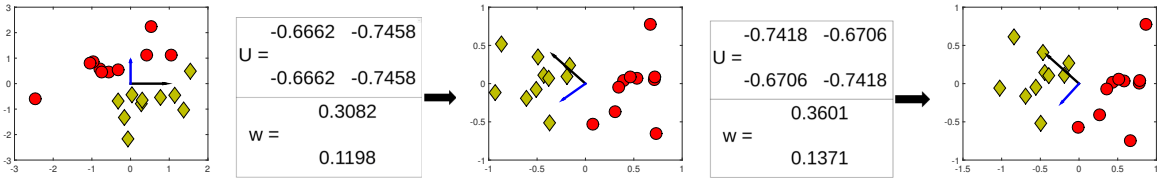


Figure 3.1 An illustration of 2 dimensional data with 2 classes (a) to show the effect of learned transformation in terms of rotation and scaling parameters. (b) shows the learned transformation after few iterations, (c) transformed points with (b), (d) transformation at convergence and (e) transformed points with final metric.

we need to learn only $np - \frac{p(p-1)}{2}$ parameters. It is worth noting that for $p > 1$, this number is strictly less than the number of parameters learned in traditional metric learning approaches. This reduction in the number of parameters provides better generalization as compared to other approaches when low training data is available.

3.3.2 Optimization Problem

With the PSD matrix parameterized as given by (3.10), we now describe our objective function for learning the metric from triplet constraints imposed on the training data. Using label information, we create triplets of data points $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \in \mathcal{T}$ such that points $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to same class while $\mathbf{x}_k$ comes from a different class and formulate the objective function as follows:

$$\min_{\mathbf{M} \succeq 0} \frac{1}{|\mathcal{T}|} \sum_{i,j,k \in \mathcal{T}} \left[1 + d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) - d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_k)\right]_+ + \alpha \mathcal{R}(\mathbf{M}) \tag{3.13}$$

Here, $d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)$ is the squared Mahalanobis distance between data points $\mathbf{x}_i$ and $\mathbf{x}_j$ with $\mathcal{R}$ as a regularizer, $\alpha \geq 0$ is a regularization parameter. The $[\cdot]_+ = \max(0, \cdot)$ is the hinge loss function that penalizes for a triplet where distance between x_i and x_k (samples from different class) is not greater than the distance between x_i and x_j (samples of same class) by atleast unit distance margin.

Now, rewriting (3.13) with the proposed parametrization $\mathbf{U}\mathbf{W}\mathbf{U}^\top$ as follows

$$\min_{\mathbf{U}\in\mathcal{S}_{n,p},\mathbf{w}\in\mathbb{R}^p_+} \frac{1}{|\mathcal{T}|}\sum_{i,j,k\in\mathcal{T}} \left[1 + d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) - d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k)\right]_+ + \alpha\mathcal{R}(\mathbf{U}\mathbf{W}\mathbf{U}^\top) \tag{3.14}$$

Here, $d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i-\mathbf{x}_j)^\top\mathbf{U}\mathbf{W}\mathbf{U}^\top(\mathbf{x}_i-\mathbf{x}_j)$ and the PSD constraint is replaced by the non-negativity constraint on $\mathbf{w}$ and orthonormality constraint on $\mathbf{U}$, $\mathbf{U}^\top\mathbf{U}=\mathbf{I}$ that are respectively optimized on positive orthant and the Stiefel manifold respectively.

In this work, we use convex spectral functions described in Section 3.1.2 as regularizer $\mathcal{R}$ in our framework. Due to the eigenvalue decomposition parametrization, the optimization problems with convex spectral function regularizers are reduced to simple expressions that are given as follows:

EigMetric-Fro

$$\min_{\mathbf{U}\in\mathcal{S}_{n,p},\mathbf{w}\in\mathbb{R}^p_+} \frac{1}{|\mathcal{T}|}\sum_{i,j,k\in\mathcal{T}} \left[1 + d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) - d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k)\right]_+ + \alpha\left\|\mathbf{w}\right\|_2^2 \tag{3.15}$$

EigMetric-Tr

$$\min_{\mathbf{U}\in\mathcal{S}_{n,p},\mathbf{w}\in\mathbb{R}^p_+} \frac{1}{|\mathcal{T}|}\sum_{i,j,k\in\mathcal{T}} \left[1 + d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) - d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k)\right]_+ + \alpha\sum_{i=1}^{p}\mathbf{w}_i \tag{3.16}$$

EigMetric-LogDet

$$\min_{\mathbf{U}\in\mathcal{S}_{n,p},\mathbf{w}\in\mathbb{R}^p_+} \frac{1}{|\mathcal{T}|}\sum_{i,j,k\in\mathcal{T}} \left[1 + d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) - d_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k)\right]_+ + \alpha\left(\sum_{i=1}^{p}\mathbf{w}_i - \sum_{i=1}^{p}\log\mathbf{w}_i - p\right) \tag{3.17}$$

We now describe our optimization strategy for the loss function defined with triplet constraints. However, our strategy is also flexible to adopt different distance constraints based loss functions with only minimal changes that are easy to accommodate in our framework.

3.4 Optimization Strategy

We develop an iterative alternating optimization strategy that alternately updates $\mathbf{U}$ and $\mathbf{w}$ using gradient based methods until the convergence. We observe that solving the individual

subproblems in Eq. (3.25) and Eq. (3.18) in each iteration for the exact solution of $\mathbf{U}$ and $\mathbf{w}$ alternately is computationally expensive and adds little value to the final solution, due to the dependency of $\mathbf{U}$ on the previous value of $\mathbf{w}$ and vice versa. Based on this observation, we alternate between $\mathbf{U}$ and $\mathbf{w}$ updates to move one step in the descent direction in every iteration.

In this section, while we illustrate the steps for the formulation in Eq. (3.15), that uses Frobenious norm regularizer, we point out the required changes to adapt the strategy for formulations in Eq. (3.16) and Eq. (3.17) as well.

3.4.1 Update Strategy for Eigenvalues $\mathbf{w} \in \mathbb{R}^p_+$

Given a solution for $\mathbf{U}$, we optimize for $\mathbf{w}$ in Eq. (3.15) by solving the following subproblem:

$$\min_{\mathbf{w}\in\mathbb{R}^p_+} \frac{1}{|\mathcal{T}|} \sum_{i,j,k\in\mathcal{T}} \left[1 + d_{\mathbf{w}}(\widehat{\mathbf{x}}_i, \widehat{\mathbf{x}}_i) - d_{\mathbf{w}}(\widehat{\mathbf{x}}_i, \widehat{\mathbf{x}}_k)\right]_+ + \alpha \left\|\mathbf{w}\right\|_2^2 \tag{3.18}$$

Here, $d_{\mathbf{w}}(\widehat{\mathbf{x}}_i, \widehat{\mathbf{x}}_j) = (\widehat{\mathbf{x}}_i - \widehat{\mathbf{x}}_j)^\top Diag(\mathbf{w})(\widehat{\mathbf{x}}_i - \widehat{\mathbf{x}}_j)$ where $\widehat{\mathbf{x}}_i, \widehat{\mathbf{x}}_j, \widehat{\mathbf{x}}_k \in \mathbb{R}^p$ and the transformed points given by $\widehat{\mathbf{x}}_i = \mathbf{U}^\top \mathbf{x}_i$.

We can further simplify and rewrite the above equation as follows:

$$\min_{\mathbf{w}\in\mathbb{R}^p_+} \frac{1}{|\mathcal{T}|} \sum_{m=1}^{|\mathcal{T}|} \left[1 + \mathbf{s}_m^\top \mathbf{w} - \mathbf{d}_m^\top \mathbf{w}\right]_+ + \alpha \left\|\mathbf{w}\right\|_2^2 \tag{3.19}$$

Here, $\mathbf{s}_m = (\widehat{\mathbf{x}}_i - \widehat{\mathbf{x}}_j) \circ (\widehat{\mathbf{x}}_i - \widehat{\mathbf{x}}_j)$ and $\mathbf{d}_m = (\widehat{\mathbf{x}}_i - \widehat{\mathbf{x}}_k) \circ (\widehat{\mathbf{x}}_i - \widehat{\mathbf{x}}_k)$ represent vectors corresponding to similar and dissimilar pair respectively for the m^{th} triplet in $\mathcal{T}$. This problem is convex in $\mathbf{w}$, but is not differentiable due to the hinge loss term. To tackle this, we introduce a dummy variable vector $\mathbf{c} \in \mathbb{R}^{|\mathcal{T}|}$ and rewrite the problem in Eq. (3.19) as follows :

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^p_+, \mathbf{c}\in\mathbb{R}^{|\mathcal{T}|}} \quad & \frac{1}{|\mathcal{T}|} \sum_{m=1}^{|\mathcal{T}|} \left[c_m\right]_+ + \alpha \left\|\mathbf{w}\right\|_2^2 \\ \text{s.t.} \quad & \mathbf{c} - \mathbf{1} - \mathbf{S}^\top \mathbf{w} + \mathbf{D}^\top \mathbf{w} = 0 \end{aligned} \tag{3.20}$$

Here, $c_m \in \mathbb{R}$ is the m^{th} entry in $\mathbf{c} \in \mathbb{R}^{|\mathcal{T}|}$ and $\mathbf{S}, \mathbf{D}$ are $p \times |\mathcal{T}|$ matrices where the columns

of the matrix correspond to vectors $\mathbf{s} \in \mathbb{R}^p$ and $\mathbf{d} \in \mathbb{R}^p$ for similar and dissimilar pairs of $\mathcal{T}$ respectively.

The function in Eq. (3.20) is now separable in $\mathbf{c}$ and $\mathbf{w}$ and can be solved effectively using Alternating Direction Method of Multipliers (ADMM) [20] approach by using a proximal operator for the hinge loss term.

So, using the solution at current iteration t, $\mathbf{U}^t$ and $\mathbf{w}^t$ we obtain updates for $(t+1)^{th}$ iteration for $\mathbf{c}^{t+1}$ and $\mathbf{w}^{t+1}$ and write the ADMM updates using Eq. B.4 as follows:

Update for c: The update for $\mathbf{c}^{t+1}$ is given by

$$\mathbf{c}^{t+1} = \underset{\mathbf{c}}{\operatorname{argmin}} \frac{1}{|\mathcal{T}|} \sum_{m=1}^{|\mathcal{T}|} \left[c_m^t\right]_+ + \frac{\rho}{2} \left\| \mathbf{c}^t - \mathbf{1} - \mathbf{S}^\top \mathbf{w}^t + \mathbf{D}^\top \mathbf{w}^t + \frac{\lambda^t}{\rho} \right\|_2^2 \tag{3.21}$$

where $\lambda \in \mathbb{R}^{|\mathcal{T}|}$ is a vector of Lagrange multipliers corresponding to every constraint in $\mathcal{T}$ and $\rho \geq 0$ is the penalty parameter. We apply the proximal operator proposed in [185] to handle $[\cdot]_+$ in Eq. (3.21) and obtain a closed form solution as

$$\mathbf{c}^{t+1} = T_{1/\rho} \left(\mathbf{1} - \mathbf{S}^\top \mathbf{w}^t + \mathbf{D}^\top \mathbf{w}^t - \frac{\lambda^t}{\rho} \right) \tag{3.22}$$

Here, the function $T_\theta(\omega)$ known as the proximal operator takes values $\omega - \theta$, 0 and ω for conditions $\omega > \theta$, $0 \leq \omega \leq \theta$ and $\omega < 0$ respectively and operates elementwise on vector arguments.

Update for w: Using $\mathbf{c}^{t+1}$, the $\mathbf{w}^{t+1}$ update is obtained by solving the following objective function.

$$\mathbf{w}^{t+1} = \underset{\mathbf{w} \in \mathbb{R}_+^p}{\operatorname{argmin}} \; \alpha \left\| \mathbf{w}^t \right\|_2^2 + \frac{\rho}{2} \left\| \mathbf{c}^{t+1} - \mathbf{1} - \mathbf{S}^\top \mathbf{w} + \mathbf{D}^\top \mathbf{w} + \frac{\lambda^t}{\rho} \right\|_2^2 \tag{3.23}$$

The problem in Eq. (3.23) is convex in $\mathbf{w}$ and can be solved using standard convex optimization techniques. We employed a simple gradient descent approach and took one step in the descent direction.

Update λ: The vector of dual variables, λ, is updated as follows

$$\lambda^{t+1} = \lambda^t + \rho \left(\mathbf{c}^{t+1} - \mathbf{1} - \mathbf{S}^\top \mathbf{w}^{t+1} + \mathbf{D}^\top \mathbf{w}^{t+1} \right) \tag{3.24}$$

3.4.2 Update Strategy for Eigenvectors $\mathbf{U} \in \mathcal{S}_{n,p}$

For updating $\mathbf{U}$, the intermediate solution $\mathbf{w}$ from Eq. (3.23) is now fixed and the problem in Eq. (3.15) is rewritten with the set of those violated triplet constraints $\mathcal{T}_1 \subseteq \mathcal{T}$, which trigger the hinge loss term as follows:

$$\mathbf{U}^{t+1} = \min_{\mathbf{U} \in \mathcal{S}_{n,p}} \frac{1}{|\mathcal{T}|} \sum_{i,j,k \in \mathcal{T}_1} \left(\mathbf{1} + {\mathbf{x}_{ij}}^{\top} \mathbf{U}\mathbf{W}\mathbf{U}^{\top}\mathbf{x}_{ij} - \mathbf{x}_{ik}^{\top}\mathbf{U}\mathbf{W}\mathbf{U}^{\top}\mathbf{x}_{ik} \right) \tag{3.25}$$

Here, $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ and $\mathbf{x}_{ik} = \mathbf{x}_i - \mathbf{x}_k$ are the difference vectors. In the problem above, we obtain a feasible $\mathbf{U} \in \mathcal{S}_{n,p}$ through a single update step that combines gradient descent with non-monotone line search [173]. Instead of computing the gradient with respect to the $n \times p$ matrix $\mathbf{U}$, we pick a subset of $q \leq n$ rows, effectively optimizing the objective on a lower dimensional Stiefel manifold. Without loss of generality, we can write $\mathbf{U}_{n\times p} = \begin{bmatrix} \mathbf{U}_s^{\top} & \mathbf{U}_n^{\top} \end{bmatrix}^{\top}$ where $\mathbf{U}_s \in \mathbb{R}^{q\times p}$ is randomly selected subset of rows from $\mathbf{U}$ and $\mathbf{U}_n \in \mathbb{R}^{(n-q)\times p}$ is the remaining set of rows of $\mathbf{U}$.

Optimization on $\mathcal{S}_{q,r}$

Since, $\mathbf{U}_s$ is no longer an orthonormal matrix, we again rewrite $\mathbf{U}_s = [\mathbf{U}_I \ \mathbf{U}_D]$, where $\mathbf{U}_I \in \mathbb{R}^{q\times r}$ is a matrix with r independent columns of $\mathbf{U}_s$ and $\mathbf{U}_D \in \mathbb{R}^{q\times(p-r)}$ represents dependent columns of $\mathbf{U}_s$. Now, following [35], we can parametrize these two components with $\mathbf{Q} \in \mathcal{S}_{q,r}$ as

$$\mathbf{U}_I = \mathbf{Q}\mathbf{H}^{1/2}, \qquad \mathbf{U}_D = \mathbf{U}_I\mathbf{T} \tag{3.26}$$

where, $\mathbf{H} = \mathbf{U}_I^{\top}\mathbf{U}_I$ is a PSD matrix and $\mathbf{T}$ is an appropriate linear transformation. With this re-parametrization of $\mathbf{U}$, the domain of the problem in Eq. (3.25) is now reduced to $\mathcal{S}_{q,r}$, leading to the following update:

$$\mathbf{Q}^{t+1} = \min_{\mathbf{Q} \in \mathcal{S}_{q,r}} \mathcal{F}(\mathbf{Q}) \tag{3.27}$$

which can be solved using the approach developed in [173]. However, since we are only interested in a single step, given an appropriate step size τ, gradient $\mathbf{G} = \nabla\mathcal{F}(\mathbf{Q})$ and $\mathbf{V} = \mathbf{G}\mathbf{Q}^\top - \mathbf{Q}\mathbf{G}^\top$, the update $\mathbf{Q}^{t+1}$ is

$$\mathbf{Q}^{t+1} = \left(\mathbf{I} + \frac{\tau}{2}\mathbf{V}\right)^{-1}\left(\mathbf{I} - \frac{\tau}{2}\mathbf{V}\right)\mathbf{Q}^t \tag{3.28}$$

which in turn leads to the update

$$\mathbf{U}^{t+1} = \begin{bmatrix} \mathbf{Q}^{t+1}\mathbf{H}^{1/2} & \mathbf{Q}^{t+1}\mathbf{H}^{1/2}\mathbf{T} \\ & \mathbf{U}_n \end{bmatrix} \tag{3.29}$$

From Eq. (3.29), an important observation is that an update on a lower dimensional Stiefel manifold affects all the columns of $\mathbf{U}$, which in turn drives the change in each w_i , $i = 1, \ldots, p$ in the following iteration. In practice, for sufficiently large q, we achieve significant computational gains without any noticeable loss in quality of the final solution.

3.5 Experimental Setup and Results

The performance of EigMetric framework is evaluated in different learning scenarios and across different applications. The performance and run-time comparisons of EigMetric with other approaches are presented in this section.

Choice of regularizers Firstly, we evaluate the flexibility of our framework to convex spectral functions as regularizers. We consider Frobenius norm, trace norm and logDet divergence functions and denote the corresponding objective functions as **EigMetric**-Fro, **EigMetric**-Tr and **EigMetric**-LogDet respectively.

3.5.1 Applications

The effectiveness of our framework is evaluated across three diverse applications that are described briefly below.

1. **Classification performance**: We performed experiments on several benchmark datasets of different sizes and feature dimensions and reported KNN classification accuracy.

2. **Person re-identification** : We evaluated the performance on two datasets and compared with widely used closed form solutions for the re-id task.

3. **Figure/ground segmentation of patterned species in images captures in the wild**: We pose figure/ground segmentation problem as a binary classification problem, where the set of features are extracted from the figure region form one class, while the background features correspond to the other class, thus casting the segmentation problem as a binary classification problem. We evaluated the performance on three different patterned species and compared them with both traditional segmentation approaches as well as other metric learning approaches.

We now discuss each of the applications and present details of the evaluation metric, datasets, experimental setup and results individually.

3.5.2 KNN Classification

We report the k-nearest neighbor classification error and run time on several datasets. The nearest neighbor classifier uses $k = 3$ nearest neighbors, breaking ties with the smallest distance. We compare our approach with various approaches: ITML [39], LMNN [171], SCML [152], FRML [126] and GMML [188]. These approaches have been discussed briefly in Section 3.2.

Datasets	Coil20	Ionosphere	Segment	USPS digits	MIT scene
# classes	20	2	7	10	8
# Samples	1440	351	1848	11000	2697
d	1024	33	19	256	512

Table 3.1 Description of Datasets: We report the number of classes and the total number of samples present in each of the datasets. The dimension (d) denotes the dimension in the raw pixel space obtained by vectorizing an image of a dataset.

Datasets and Parameter Settings

Ionosphere[1], Segment[1], MIT scene [2] and Coil20 [3]. All the datasets are transformed to a low dimensional representation by using PCA with 95% energy. The dimension of the data after the projection is the feature dimension n. The parameters used in different metric learning approaches are as follows:

EigMetric Method: The parameter p *i.e.* the rank of the PSD matrix is obtained by applying PCA with 95% energy on the training data. The value of q that defines the small Stiefel manifold $\mathcal{S}_{q,r}$ is calculated as $q = \frac{n}{4}$ and $q = n$ for $n > 20$ and $n \leq 20$ respectively. The penalty parameter ρ and weight for the regularization term α are fixed to $1e^4$ and $1e^{-4}$ respectively in all the experiments.

Baseline Methods: The parameters used in LMNN and GMML are obtained using the selection strategy provided with their code. For SCML and ITML, the optimal parameters are chosen by experimentation. The rank parameter in FRML uses the same p value as used in our EigMetric framework. For all the approaches, we use the implementation code available on the respective authors website.

[1] http://archive.ics.uci.edu/ml/

[2] http://people.csail.mit.edu/torralba/code/spatialenvelope/

[3] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

Datasets	USPS	Ionosphere	MIT	Segment	Coil20
Euclidean	13.42±0.47	17.22 ±3.68	50.27±1.73	14.49 ±1.02	23.03±2.02
LMNN	8.06±0.56	21.92±11.17	45.95±3.17	12.66 ±3.47	19.6 ±3.09
ITML	12.80 ±1.32	17.69± 3.31	49.86±1.68	12.68 ± 1.44	24.11±1.89
SCML	9.16± 0.55	18.64± 5.05	48.90± 1.55	18.27 ±5.41	19.58±2.20
FRML	15.04 ± 1.18	17.38± 5.09	43.51± 1.46	13.01 ±1.07	19.61 ± 1.50
GMML	10.70 ± 0.56	15.11 ± 3.71	47.92± 2.47	18.16±9.31	21.4±2.49
EigMetric-Fro	7.36 ± 0.40	**13.78± 4.01**	43.49±1.86	10.99± 1.22	**16.45±2.73**
EigMetric-Tr	**7.21± 0.35**	14.82± 3.66	43.16±1.43	**10.90 ±0.95**	16.46±2.55
EigMetric-LogDet	7.22±0.40	14.10±2.92	**42.97 ±1.41**	11.10 ± 1.34	16.60 ±2.79

Table 3.2 Small Training Data Setting (10 % data for training): 3-NN classification error (%) averaged over 10 random splits for 5 datasets. Comparison of three variants of EigMetric with traditional approaches: ITML, LMNN, SCML, FRML and GMML (best results in bold).

Classification Error

To compare the generalization performance of EigMetric with other baseline approaches, we perform experiments under two settings. Small training data setting with 10/90 split, where 10% data is used for training while 90% is used for testing and large training data setting with 60/40 split. All the results presented in this chapter are averaged over 10 random splits of the data, unless otherwise stated.

Small Training Data Setting: The classification error for different methods is reported in Table 3.2. As compared to existing approaches, EigMetric performs significantly better across all the datasets. EigMetric achieves small classification error and variance as compared to other approaches. The results point out that the flexibility in regularizer can lead to better performance for a given dataset. Also, the results on MIT and Ionosphere dataset show that both FRML and GMML perform better than other approaches. This improved performance is likely due to the geometry of the Riemannian manifold explored in these methods.

Datasets	USPS	Ionosphere	MIT	Segment	Coil20
Euclidean	6.38± 0.45	8.79 ±1.92	44.15±1.59	5.60± 0.73	4 ± 0.81
LMNN	4.1± 0.27	8.15 ±1.30	36.79±1.11	**4.51± 0.54**	1.15 ± 0.55
ITML	11.26 ±2.04	8.36 ±2.10	45.51±1.91	9.14 ±0.56	4.25 ± 0.80
SCML	4.42±0.40	7.80 ±2.90	39.5±1.59	5.64 ±1.32	2.6± 0.62
FRML	10.17 ±3.87	10.92±1.80	39.84 ±1.17	6.98 ± 0.77	4.63±1.28
GMML	5.28 ±0.41	13.90±3.60	41.06±2.33	6.16±2.17	3.12 ±0.81
EigMetric-Fro	**3.26±0.38**	**7.73±1.99**	**35.44 ±1.24**	4.79±0.95	**0.84±0.43**
EigMetric-Tr	3.86 ± 0.38	8.36±1.84	34.55±1.31	4.75±0.82	0.94±0.55
EigMetric-LogDet	3.30±0.35	8.01±1.73	34.77 ±1.25	4.77±0.87	0.93 ±0.53

Table 3.3 Large Training Data Setting (60% data for training): 3-NN classification error (in %) averaged over 10 random splits for 5 datasets. Comparison of three variants of EigMetric with traditional approaches: ITML, LMNN, SCML, FRML and GMML (best in bold).

Large Training Data Setting: The results for large training data setting are presented in Table 3.3. EigMetric performs at par with the other competing methods.

Effect of size of Training Data: Figure 3.2 shows average classification error on USPS digits dataset as a function of the number of training samples. The results show that EigMetric outperforms other methods by a significant margin when available data for training is scarce and also performs better in presence of large training data as well. On the other hand, other Riemannian manifold based methods like FRML and GMML achieve improved results over other traditional approaches in small training data setting, but drops in large training data setting.

3.5.3 Person Re-Identification

Person Re-Identification refers to the task of matching images of individuals across different camera views. We conducted experiments on two widely used public datasets and used different representations that have shown to perform well for re-id task.

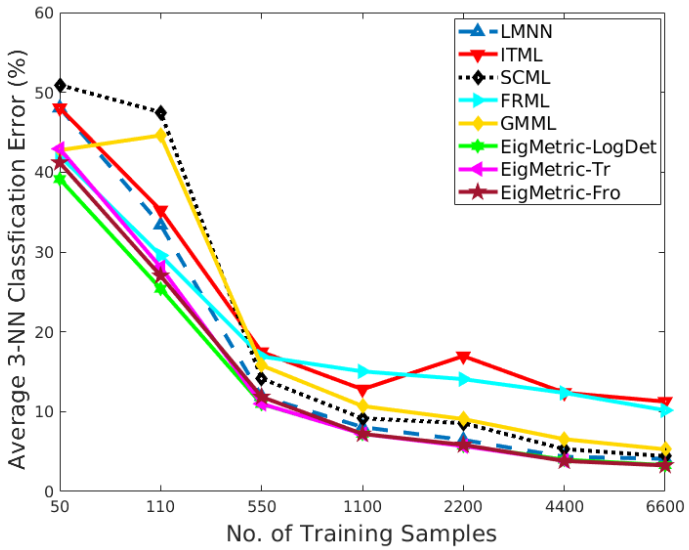


Figure 3.2 Effect of training data size on average 3-NN classification error (%) on USPS digits dataset for various metric learning approaches. EigMetric based framework consistently achieves lowest error rates.

Datasets and Setup

We evaluated on two publicly available datasets, VIPeR [59] and GRID [119]. VIPeR dataset has been used for benchmarking human re-id task and has 632 individuals with two images per person, one in each of the two camera views. On the other hand, QMUL underGround Re-Identification (GRID) is a challenging dataset captured from 8 disjoint camera views in an underground station. It contains 1275 images collected from 900 individuals with one image pair across two views for 250 individuals and an additional 775 images of other individuals to increase the gallery set.

In our experiments, we adopt two protocols for evaluation. First is the standard test/train setting where half of the image pairs are used for training *i.e.* 316 and 125 pairs in case of VIPeR and GRID datasets respectively. The second setting is used to evaluate the performance of our framework in small training data setting that uses only 25% of the total pairs for training while remaining are used for testing.

Results

We evaluate the performance using cumulative matching characteristic (CMC) that represents the expectation of finding the true match within the first r ranks. The results for large and small training data setting for the two datasets are reported in Table 3.4 and 3.5. The results show that the EigMetric consistently outperforms existing approaches in the standard train/test setting.

In the case of small training data setting, as most datasets are small and have very few images per individual with a few hundreds of individuals only, the number of samples for training are reduced drastically. We observe that it adversely affects the performance of closed form solution approaches used for re-id, more than EigMetric approach, as they fail to capture distinguishable information from data. This also highlights the merit of our framework in small training data that is due to the reduced number of parameters for optimization.

		Small No of Pairs					Standard Setting				
Features	Method	r=1	r=5	r=10	r=15	r=20	r=1	r=5	r=10	r=15	r=20
LOMO [107]	XQDA	27.30	53.52	66.98	74.81	79.60	40	68.13	80.5	87.37	91.07
	EigMetric	28.63	55.15	67.78	75.25	80.32	40.92	70.28	82.72	88.2	91.71
FTCNN [123]	XQDA	31.20	58.00	70.70	77.83	82.72	42.41	71.01	82.09	88.13	92.44
	EigMetric	34.06	60.50	71.96	78.53	82.96	43.2	71.9	83.10	89.01	92.94

Table 3.4 VIPeR dataset [59]: Performance comparison of our metric learning approach against baseline approach XQDA, when metric is learned on LOMO and FTCNN features. The cumulative matching scores (%) at rank (r) 1, 10, 15 and 20 are reported.

3.5.4 Figure/Background Segmentation of Patterned Species

Segmentation of patterned species in images captured in the wild is challenging due to several factors such as unavailability of large labeled data, complex illumination effects, non-rigid

		Small No of Pairs					Standard Setting				
Features	Method	r=1	r=5	r=10	r=15	r=20	r=1	r=5	r=10	r=15	r=20
LOMO	XQDA	12.61	27.61	36.65	42.82	47.07	18.40	37.28	47.20	53.52	57.76
	EigMetric	13.56	27.71	35.85	43.67	48.51	19.60	37.52	47.76	54.24	58.56
FTCNN	XQDA	21.60	39.26	48.19	53.72	58.35	24.08	43.28	51.36	57.60	62.80
	EigMetric	24.41	40.69	51.12	56.81	60.21	32.16	49.28	55.36	60.32	63.04

Table 3.5 GRID dataset [119]: Performance comparison of our metric learning approach against baseline approach XQDA, when metric is learned on LOMO and FTCNN features. The cumulative matching scores (%) at rank (r) 1, 5, 10, 15 and 20 are reported.

variations in animal pose and occlusions. The goal of segmentation is to produce a binary mask that separates the foreground object from the background. We first generate a super-pixel based feature representation of a given image that allows efficient processing at query time. As our domain consists of images of patterned animals, we create a feature space that captures the texture properties. The details for the same can be found in the Appendix C.1. The training data is used to learn a discriminative Mahalanobis distance based metric offline. At query time the learned metric is used with mean shift clustering [36] to perform clustering of the feature representations. Using a combination of mean shift and metric learning increases the robustness of our figure-ground segmentation process against clutter and illumination variations. The details of the same are present in Appendix C.2. Our method does not require any user input for the query image and relies only on a few training images for learning the metric.

Datasets and Setup

We evaluate the performance on three patterned species: tiger, leopard and zebra. The approaches are evaluated on 30 tiger and 30 zebra [96] images. Leopard images are collected from the web. The ground truths for all the images are created using interactive segmentation

tool [4]. We compare our metric learning method with other segmentation techniques: Graph cut [22], GrabCut [145] and Random Walker [57]. We also compare the effectiveness of our learned distance metric with Euclidean distance as a baseline as well as metrics learned by two popular approaches ITML [39] and LMNN [171].

The quality of segmentation is evaluated by computing average pixel-wise precision/recall and segmentation accuracy.

Training Setup

The metric learning problem for leopard and zebra datasets is formulated as a two class problem: figure and background. We use only one labeled image to extract 20 feature vectors from each class and generate similar and dissimilar image pairs for metric learning. For tiger images, these feature vectors are selected from two labeled images and the problem is formulated as a three class problem: figure, upper background and lower background. We divide the background into two subcategories to handle the visible similarity between the tiger and the illuminated ground conditions that appear due to flash in the camera traps.

Results

For tiger and zebra datasets, we report the average pixel-wise precision/recall and segmentation accuracy in Table 3.6. The results highlight the benefit of using a learning based method for image segmentation as opposed to user input dependent approaches. While this advantage is visible by the improvement in the results, it is more suitable for processing a large number of images. For the leopard dataset, since there is no publicly available database, we only report qualitative results in Figure 3.3.

[4]Available at http://kspace.cdvp.dcu.ie/public/interactive-segmentation

Method	Tiger (30 images)			Zebra (30 images)		
	Precision (%)	Recall (%)	Segmentation Accuracy(%)	Precision (%)	Recall (%)	Segmentation Accuracy(%)
GrabCut [145]	69.11	90.04	93.51	**98.10**	92.74	**96.43**
RW [57]	36.58	89.27	69.57	97.59	65.23	60.39
Graph Cut[21]	32.48	81.87	72.17	94	59.26	57.03
ITML[39]	30.71	64.57	65.01	64.92	92.77	66.76
LMNN [171]	51.75	67.93	72.19	50.03	70.93	73.14
Euclidean	26.37	33.98	71.51	75.86	80.95	78.61
EigMetric-FigSeg	**78.04**	**93.49**	**93.61**	90.33	**93.72**	94.59

Table 3.6 Comparison of Average Precision/Recall and Segmentation Accuracy on Tiger and Zebra datasets for different segmentation approaches (best results reported in bold).

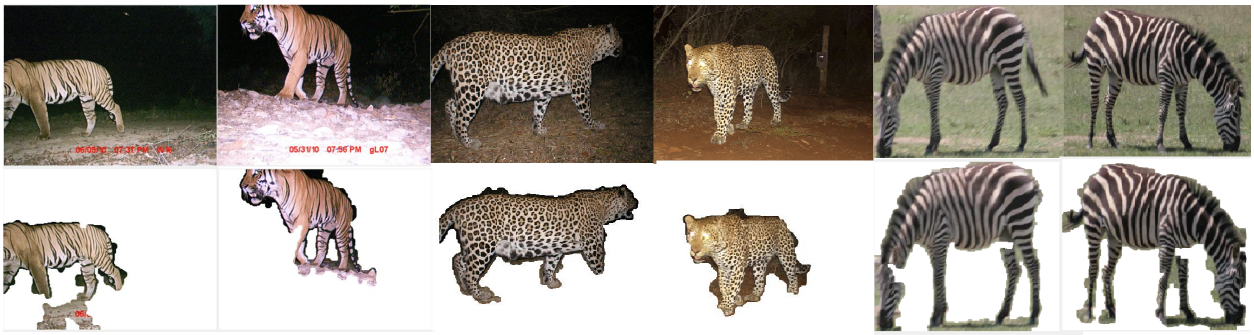


Figure 3.3 Qualitative segmentation results. The first row is the original images and the bottom row is the segmented foreground with our approach EigMetric-FigSeg.

3.6 Analysis and Ablation Study

In this section, we analyze different aspects of our proposed optimization strategy and the hyper-parameters in our formulation. We have also evaluated the objective function of existing metric learning approaches in our EigMetric framework to validate its flexibility to incorporate different objectives.

3.6.1 Row Selection for $\mathcal{S}_{q,r}$ Parametrization

As we parametrize the eigenvectors on the Stiefel manifold $\mathcal{S}_{n,p}$, it is further reduced to an optimization problem on a smaller Stiefel manifold $\mathcal{S}_{q,r}$. A smaller value of q essentially

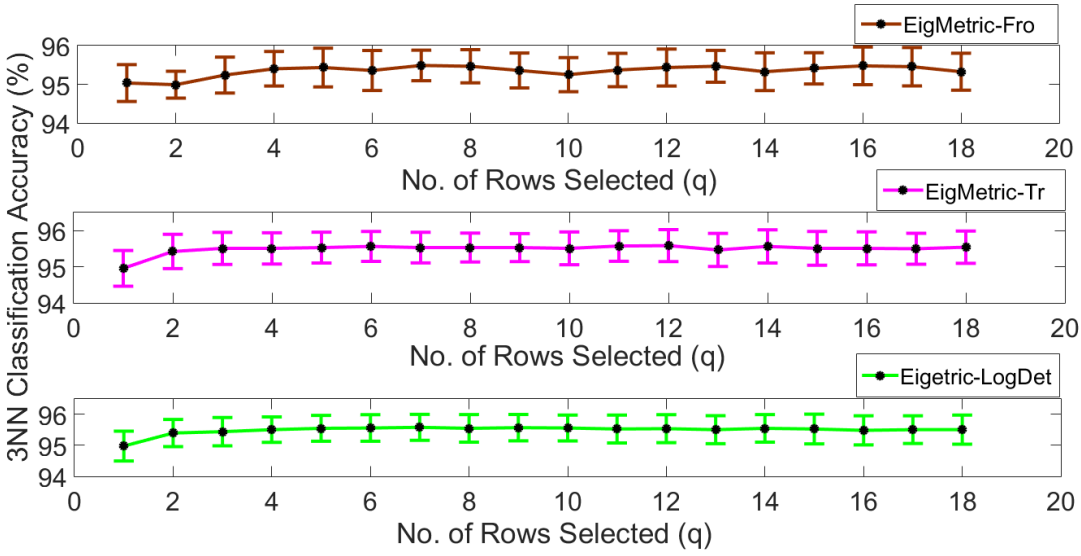
corresponds to a modification of a smaller set of dimensions to satisfy the given distance constraints on the training data. In the experiments, we fix q to $\frac{n}{4}$ and n for high dimension and low dimension data respectively. In Figure 3.4, we analyse the effect of this hyper-parameter on the performance of the classifier. The plot shows classification accuracy for Coil20 and Segment dataset with an increasing value of q. We observe that the performance is stable across different dimensions of the smaller manifold provided it is above the threshold value.

3.6.2 EigMetric for Existing Metric Learning Objective Function

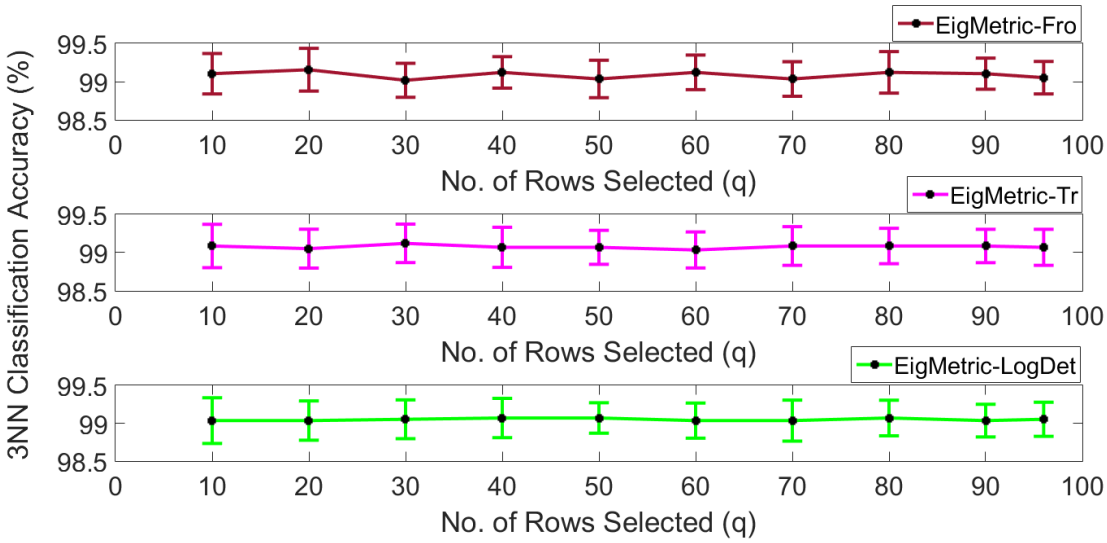
In the experiments for k nearest neighbor classification discussed in the previous section, EigMetric framework with triplet constraints is observed to consistently perform better than previous metric learning approaches. In this section, we also show that the objective functions of existing metric learning approaches in the EigMetric framework improve their performance as well. We present results for LMNN objective function [171] in EigMetric framework *i.e* we optimize the objective function with eigenvalue decomposition parametrization and our optimization strategy. The results are given in Table 3.7. Further, we also evaluate the effect of regularization in the same setup.

3.6.3 Comparison with Projected Gradient

Apart from the recent developments in metric learning that explore geometric information in the learning process, a large portion of metric learning methods comprises of a projection step at every iteration. The projection step computes an eigenvalue decomposition of the Mahalanobis matrix and ensures the positive semidefiniteness by clipping the negative eigenvalues to 0. We compare the performance of EigMetric with Projected Gradient (PrGrad) method in the Table 3.8.



(a) Segment Dataset



(b) Coil20

Figure 3.4 Average k nearest neighbor classification accuracy over 5 runs (in %) for different value of q that defines the dimension of the manifold $\mathcal{S}_{q,r}$ and $k = 3$ on different datasets.

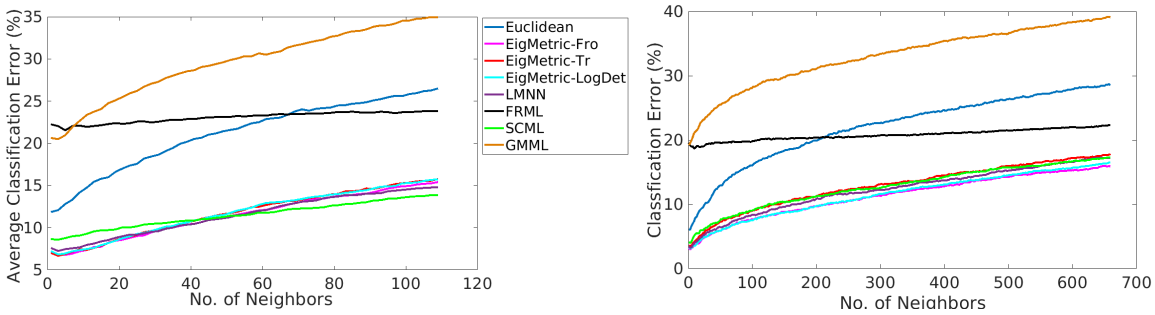
3.6.4 Performance with Neighborhood size in KNN

In order to evaluate the effectiveness of learned metric in capturing data semantics, we perform an experiment with varying neighborhood size in k nearest neighbor classifier. The

Dataset	Segment	USPS
LMNN	4.51 ± 0.54	4.1 ± 0.27
EigMetric-LMNN	4.48 ± 0.68	3.71 ±0.28
EigMetric-Fro-LMNN	**4.46 ± 0.83**	**3.42±0.32**
EigMetric-Tr-LMNN	4.48 ± 0.89	3.52± 0.39
EigMetric-LogDet-LMNN	4.46± 0.89	3.67± 0.26
EigMetric-Fro-Logistic	4.55 ± 0.98	3.33 ± 0.38
EigMetric-Tr-Logistic	**4.44 ± 0.86**1	**3.14 ± 0.28**
EigMetric-LogDet-Logistic	4.45 ± 0.85	3.21 ± 0.28

Table 3.7 Performance evaluation of EigMetric framework with different objective functions (60% training data). EigMetric-LMNN uses LMNN objective function, EigMetric-()-LMNN uses LMNN objective function along with a regularizer. EigMetric-()-Logistic uses logistic loss function used in FRML [126] with different regularizers.

results for USPS digits dataset in small and large training data setting are shown in Figure 3.5. While FRML shows a stable behaviour across neighbourhood size in both the settings, EigMetric sustains lower classification errors as opposed to other approaches.



(a) Small Training Data **(b)** Large Training Data

Figure 3.5 Comparison of k - nearest neighbor classification error vs neighbourhood size for USPS digits datasets for different metric learning approaches under small and large training data settings.

Dataset	Method	Classification Error		
		Frobenius	Trace	LogDet
Segment	PrGrad	11.76±0.95	11.35±1.67	11.40 ± 1.17
Small Training Data	EigMetric	**10.99± 1.22**	**10.90 ±0.95**	**11.10 ± 1.34**
Segment	PrGrad	4.89±0.73	6.69±1.64	4.88±0.86
Large Training Data	EigMetric	**4.79±0.95**	**4.75±0.82**	**4.77±0.87**
USPS	PrGrad	7.42±0.31	8.04 ±0.52	7.97 ± 0.43
(Small Training Data)	EigMetric	**7.36 ± 0.40**	**7.21± 0.35**	**7.22±0.40**
USPS	PrGrad	4.15 ±0.45	15.27 ±2.20	4.00 ± 0.43
(Large Training Data)	EigMetric	**3.26±0.38**	**3.86 ± 0.38**	**3.30±0.35**

Table 3.8 Comparison of average classification error (%) of Projected Gradient and EigMetric for different spectral regularizers in small and large training data setting.

3.6.5 Run Time Comparisons

Table 3.9 reports the run time of different approaches in small and large training data settings. EigMetric and other metric learning approaches except SCML are implemented in MATLAB and are available on the authors' webpage. All the experiments are run on a laptop with core i7 quad core processor and 8 GB RAM with only two cores enabled. The improvement in classification performance under both small and large training data comes at a cost of slightly higher running time.

Datasets	EigMetric	LMNN	SCML	FRML	GMML
USPS (Small Data)	21.58± 3.2	372 ±102.31	14.89 ±0.21	17.37 ±4.23	5.87±0.53
Coil20 (Small Data)	4.15 ±4.78	25.17 ±14.76	0.43 ±0.15	2.02 ±0.91	1.74±0.08
USPS (Large Data)	404.05 ± 45.33	3014 ±208	194.62 ±23.67	24.42 ±3.70	73.33±5.36
Coil20 (Large Data)	21.91 ±4.41	132.68 ±34.98	5.51 ±0.77	17.39 ±7.96	8.30±0.55

Table 3.9 Comparison of average run time (in secs) of EigMetric with existing metric learning approaches in small (10% training data) and large training data (60% training data) settings.

Dataset	MIT		USPS	
Strategy	Alternating Gradients	Our Approach	Alternating Gradients	Our Approach
EigMetric-Fro	1006.2 ± 140.33	112.97 ± 38.47	929.39 ± 126.30	404.05±45.33
EigMetric-Tr	917.39 ± 117.593	639.16 ± 163.27	1255 ±122.80	73.80± 9.50
EigMetric-LogDet	749.45± 58.0	602.30 ±83.16	1359 ±98.62	819.09±88.18

Table 3.10 Average run time comparison of EigMetric optimization strategy with Alternating Gradient approach for two datasets and across different regularizers. EigMetric achieves significantly lower runtime across different datasets as well as regularizer functions.

Alternating Gradients vs EigMetric Optimization Strategy

A simple solution to alternate between $\mathbf{U}$ and $\mathbf{w}$ updates is to do gradient step updates for both $\mathbf{U}$ and $\mathbf{w}$ subproblems *i.e.* the method of alternating gradients as opposed to our strategy that alternates between an ADMM update step for $\mathbf{w}$ subproblem and a gradient based update for $\mathbf{U}$ subproblem. We compared our strategy with the alternating gradient algorithm that is summarized in Algorithm 2, The comparison of run time on two datasets is presented in Table 3.10 and shows reduction in run time.

3.7 Conclusion

In this work, we parameterized the PSD matrix of Mahalanobis distance metric with orthonormal eigenvectors and the associated non-negative eigenvalues. We proposed a joint optimization strategy over $\mathcal{S}_{n,p} \times \mathbb{R}^p_+$ to learn the orthonormal matrix on the Stiefel manifold $\mathcal{S}_{n,p}$ and the corresponding eigenvalues on the positive orthant *i.e.* $\mathbb{R}^p_+$. As opposed to the existing approaches, the number of learnable parameters is reduced from n^2 to $np - \frac{p(p-1)}{2}$ owing to the parameterization, improving the generalization performance in scarce training data settings. Additionally, the proposed framework allows flexibility to incorporate any convex spectral function as a regularizer to promote a task specific solution. Furthermore,

we evaluated our framework to show the applicability of approach across different applications that include image classification, image segmentation as well as person re-identification task.

Chapter Four

Representations for Visual Animal Biometrics

Visual animal biometric is a challenging problem, where the task is not to just identify a species but also the specific identity of the individual. With the increasing use of visual sensors like camera traps for passive monitoring of wildlife, while data collection is substantially cheaper and more scalable, data labelling is still a challenge. Owing to very subtle differences among the individuals of a species, it is challenging even for the experts to label, making it a cumbersome and tedious process. This poses a bottleneck in training deep networks that would suffer from over-fitting in the absence of large training dataset. In this chapter, we propose to use the semantic constraints on the statistical manifold to define a regularizer that is augmented in the traditional loss function to mitigate the over-fitting problem. We evaluated our approach on two wildlife monitoring applications namely: primate face recognition and tiger re-identification.

Key Highlights

- *Used semantic pairwise constraints as a regularizer to learn robust and generalizable representations.*
- *The learned representations achieve state-of-the-art results on datasets of two different species of primates namely, chimpanzees and rhesus macaques.*
- *Used data transformations appropriate for the tiger Re-ID task.*
- *For tiger re-identification, combined SIFT-based matching together with the learned representations.*

4.1 Overview of the Contribution

In this chapter, we propose to learn representations for primate face recognition as well as tiger re-identification problem. An important role that CNN-based classifiers play is learning useful feature representations, even when simply trained for classification. In a typical classification setting, the network produces a probability distribution over all classes, and a cross-entropy loss is used to encourage the network to predict the correct class with high probability. The cross-entropy loss only encourages predicting the correct class and ignores any other information present in the distribution over all other classes. This can contribute to overfitting and a lack of consistency between predictions of different instances from the same class. We propose a pairwise KL-divergence penalty to encourage the network to produce similar distributions for instances of the same class, and dissimilar distributions for instances of different classes. In the case of primates, these representations alone have been shown to generalize well even when the evaluation data may consist of novel individuals

and environments. But in case of tiger re-identification task, for learning the representations we employ several types of data augmentation to account for a range of possible geometric, environmental, and image-quality transformations along with SIFT matching to tackle the subtle changes in the tiger identities.

4.2 Existing Literature

Work in visual animal biometrics has focused primarily on animals that have unique coat patterns like tigers or leopards, or on non-human primates like chimpanzees, gorillas and monkeys. Earlier techniques relied mostly on human input to get the Region of Interest (ROI) or key-points, while recent techniques utilized an end-to-end, automatic pipeline using a CNN for feature extraction or classification. We broadly categorize these approaches into two categories: Non-Deep Learning Approaches and Deep Learning Approaches and discuss developments for both patterned species as well as primates.

4.2.1 Non-Deep Learning Approaches

One of the earliest works in patterned species individual recognition developed the interactive software method *Extract-Compare* [71] for recognizing individuals by matching coat patterns for species like tigers, giraffes, frogs, etc. While the tool worked well in terms of accuracy, it required fifteen to twenty points to be manually marked in each image so that a 3D surface model can be fitted to the animal's body. This model is then used to unwarp the flank region to improve stripe matching. Sloop [45] was another interactive retrieval engine which, in addition to utilizing user input for key-points, pre-processed images for noise removal, extracted various key-point descriptors like SIFT [118] for matching and also had a relevance feedback loop for crowdsourcing. Hotspotter [37] and Wild-ID [18] also used SIFT features to match query images with a database of existing animals. Hotspotter also used efficient data structures like kd-trees, different scoring criteria and spatial re-ranking to rank

the matched descriptors obtained from database images. For aquatic animals like Saima ringed-seals, recent works [191, 29] used unsupervised segmentation to segment the body into superpixels, followed by foreground/background classification before using Hotspotter and Wild-ID for matching.

Similarly, early works in primate face recognition followed the standard pipeline for face recognition that consists of face alignment, followed by low level feature extraction and classification. The work in [117], adopted Randomfaces [176] technique for identifying chimpanzees in the wild and followed the standard pipeline for face recognition. Later, LemurID [38], additionally used manual marking of the eyes for face alignment. Patch-wise multi-scale Local Binary Pattern (LBP) features were extracted from aligned faces and used with LDA to construct a representation, that were then used with an appropriate similarity metric for identifying individuals.

4.2.2 Deep Learning Approaches

Recent methods for patterned species recognition like [28] and [129] used a detector network or unsupervised segmentation to crop the ROI and extracted CNN features from a pre-trained network that are used to train an SVM classifier for classification of individuals. A similar method was employed by [50, 24] for classifying chimpanzee and gorilla faces. Due to limited training data, they explored the use of different layers of a pre-trained AlexNet [90] as input to SVM, instead of fine-tuning the network. Freytag et al. [50] used CNNs for learning a feature representation for chimpanzee faces. For increasing the discriminative power, the architecture used a bilinear pooling layer after the fully connected layers (or a convolutional layer), followed by a matrix log operation. These features were then used to train a SVM classifier for classification of known identities. Later, [24] developed face recognition for the gorilla images captured in the wild. This approach fine-tuned a YOLO detector [142] for gorilla faces. For classification, a similar approach was taken as [50], where pre-trained CNN

features were used to train a linear SVM. More recently, [41] proposed PrimNet, that used the *Additive Margin Softmax* loss [170] and achieved state-of-the-art performance for identifying individuals across different primate species including lemur, chimpanzee and golden monkey. However, it required substantial manual effort in designing landmark templates for face alignment prior to the identification process, that can adversely affect the adoption rates in a crowdsourced mobile app setting. For human face recognition techniques, various approaches had improved performance by combining the standard cross entropy loss with other loss functions such as contrastive loss [161] and center loss [172] to learn more discriminative features.

4.3 Semantic Constraints on the Statistical Manifold

4.3.1 Motivation

A CNN for image classification task transforms an image to a probability vector of K dimensions, where K is the number of classes. The network is trained with a cross entropy loss that pushes this distribution to a one hot encoding vector. However, in absence of sufficient training data, the network suffers from over-fitting and the distributions are not indicative of the class label. In this chapter, we propose to utilize the notion of task dependent semantic information to improve these distributions in the absence of sufficient training data.

We leverage the geometry of space of probability distributions known as the statistical manifold to ensure that the images of the same class have similar distributions while images from different classes have diverse distributions. We use an approximation of distance measure defined on this manifold known as the KL-divergence measure to drive the similarity/dissimilarity between the distributions [85, 8]. We add semantic pairwise loss during network training that reduces the disparity between the distributions for points of the same class, while moving the points from different classes far apart. The familiarity with the

statistical manifold and the corresponding metric can be obtained by reading Section 2.1.3 in this thesis. We now describe the modified loss function used during network training in the following section.

4.3.2 Proposed Pairwise Semantic Loss

Given a training dataset of n samples $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ with $l_i \in \{1, 2, \ldots, K\}$ as the associated labels. We use the labeled training data to create the set of similar image pairs, $\mathcal{C}_s = \{(i,j) : \mathbf{x}_i,\mathbf{x}_j \in \mathcal{X}, l_i = l_j\}$, and dissimilar pairs, $\mathcal{C}_d = \{(i,j) : \mathbf{x}_i,\mathbf{x}_j \in \mathcal{X}, l_i \neq l_j\}$ for $i, j \in \{1, 2, \cdots, n\}$. The KL-divergence between two distributions $\mathbf{p}^i$ and $\mathbf{q}^j$ corresponding to points $\mathbf{x}_i$ and $\mathbf{x}_j$ is given by

$$KL(\mathbf{p}^i||\mathbf{q}^j) = \sum_{k=1}^{K} p_k^i \log \frac{p_k^i}{q_k^j} \tag{4.1}$$

For a similar pair $(i,j) \in \mathcal{C}_s$, we use the symmetric variant of (4.1) given by

$$\mathcal{L}_s^{ij} = KL(\mathbf{p}^i||\mathbf{q}^j) + KL(\mathbf{q}^j||\mathbf{p}^i) \tag{4.2}$$

and for a dissimilar pair $(i,j) \in \mathcal{C}_d$, we use its large-margin variant for improving the discriminative power

$$\mathcal{L}_d^{ij} = \max(0, m - KL(\mathbf{p}^i||\mathbf{q}^j)) + \max(0, m - KL(\mathbf{q}^j||\mathbf{p}^i)) \tag{4.3}$$

where m is the desired margin width between dissimilar pairs.

4.3.3 Primate Face Recognition

In case of primate face recognition, we combine the standard cross entropy loss with a guided pairwise KL-divergence loss imposed on similar and dissimilar pairs. Using pairwise loss terms ensure that the underlying features are more discriminative and generalize better. Our analysis in Sec. 4.4.4 shows empirical evidence that the learned features are more clusterable than when trained with the standard cross-entropy loss.

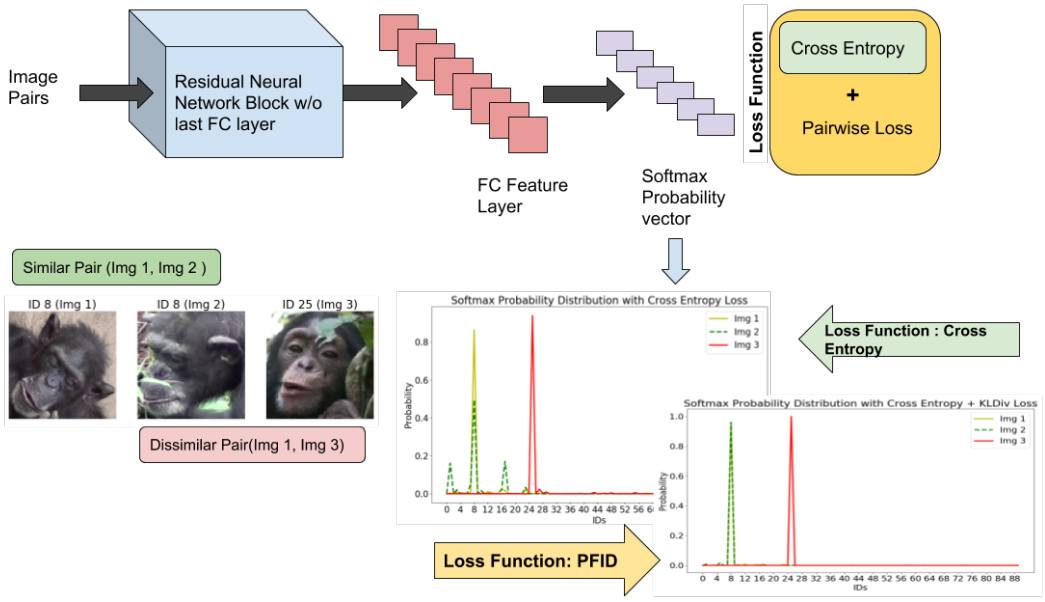


Figure 4.1 Illustration of proposed PFID loss function vs. the standard cross entropy loss on the learned class probability distributions with ResNet model.

An illustration of the effect of loss function is shown in Figure 4.1. A similar pair corresponds to images of the same individual, while a dissimilar pair corresponds to images from two different identities. The learned class probability distribution for a similar pair and dissimilar pair using two different loss functions is shown. In case of network trained with **P**rimate **F**ace **ID**entification (PFID) loss, the class probabilities are maximally similar for a similar pair as opposed to standard cross entropy loss.

We experimentally found that during training, when both $\mathbf{x}_i$ and $\mathbf{x}_j$ are mis-classified by the model, minimizing Eq. (4.2) may lead to an increase in the bias. We therefore considered the guided pairwise loss function that only considers those similar pairs, where atleast one of the images in a pair is correctly classified.

Guided Pairwise Loss Function Since we use class labels for the cross-entropy loss, we incorporate them in the pairwise loss terms to guide the training. Subsequently, we modify

the terms in Eq. (4.2) and Eq. (4.3) to get the following guided KL-divergence loss term:

$$\mathcal{L}_s = \sum_{i,j \in \mathcal{C}_s} a\mathcal{L}_s^{ij}, \qquad \mathcal{L}_d = \sum_{i,j \in \mathcal{C}_d} a\mathcal{L}_d^{ij} \tag{4.4}$$

where, $a = 1$ if either $\arg\max\ \mathbf{p}^i = l_i$ or $\arg\max\ \mathbf{q}^j = l_j$ and $a = 0$ otherwise. The loss function PFID is given by the sum of standard cross entropy ($\mathcal{L}_{CE}$) and the guided KL divergence loss

$$\mathcal{L}(\theta) = \mathcal{L}_{CE} + \frac{1}{|\mathcal{C}_s|} \sum_{i,j \in \mathcal{C}_s} a\mathcal{L}_s^{ij} + \frac{1}{|\mathcal{C}_d|} \sum_{j,k \in \mathcal{C}_d} a\mathcal{L}_d^{jk} \tag{4.5}$$

This loss function is used to train the network with a mini-batch gradient descent. Here $|\mathcal{C}_s|$ and $|\mathcal{C}_d|$ are the number of similar and dissimilar pairs respectively in a given batch. More details on the training are provided in Sec. 4.4.3.

4.3.4 Tiger Re-Identification

In the case of tiger data, the loss function is modified to include all pairs in Eq. (4.5) i.e. we set $a = 1$. However, the identification process involves several other steps along with a modified loss function used for primate recognition. An overview of our method is shown in Figure 4.2. The training images are transformed according to our augmentation scheme before being passed to the convolutional network. The network is trained to classify images according to their identity, using a combination of regular cross-entropy loss and a KL-divergence loss between pairs of class-probability vectors. At inference time we take the test set and treat each image as query, with all others as the gallery. The goal is to rank all the images in the gallery so that images of the same identity as the query get ranked highest. We use the class scores produced by the network as an image descriptor, and initially rank the gallery by cosine similarity to the query. We then reorder the ranking so that all images of the same flank as the query (facing left or right) get placed first. Finally, we re-rank the *top twenty* gallery images by matching SIFT descriptors to the query.

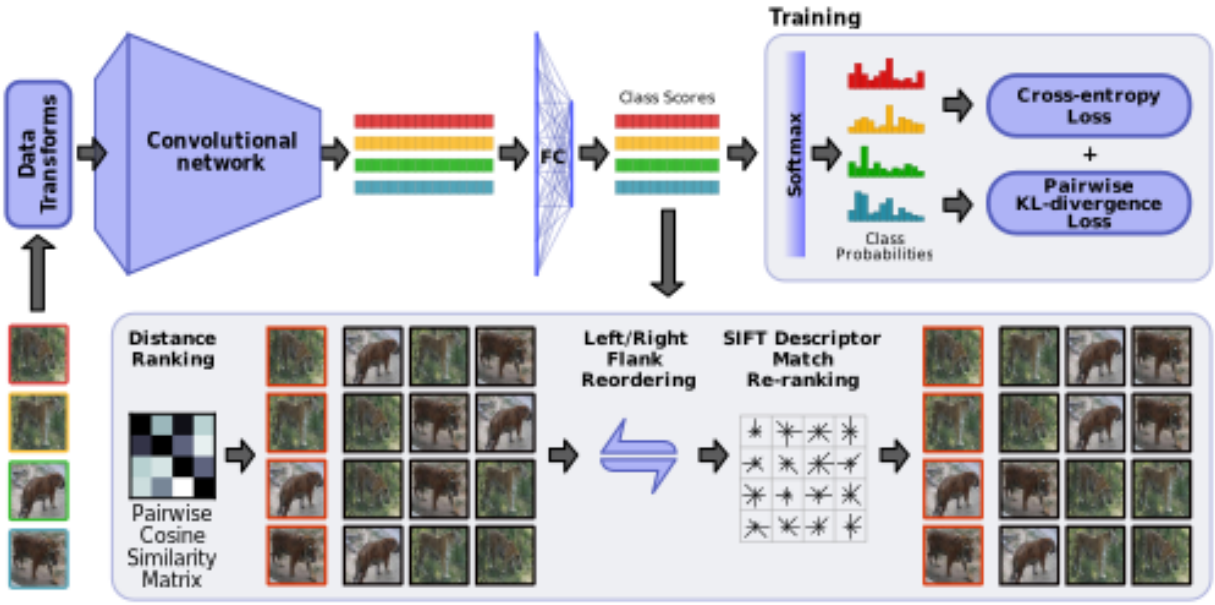


Figure 4.2 Overview of proposed approach: During training, a DenseNet121 network is finetuned using cross-entropy and pairwise KL-divergence losses on images that have been augmented through a variety of transforms. During evaluation, the class-score vectors are used as features for similarity ranking. The initial ranking is then modified by using flank information and SIFT descriptor matching.

We now discuss the details of our data augmentation method, the flank detection and the SIFT matching for re-ranking.

a) **Data Transformation**

We utilize several types of image transforms to improve the model's robustness to geometric, environmental, and image-quality variations. Common image augmentation techniques are random cropping, random rotations, random horizontal flipping, and random color-jitter (adjusting brightness, contrast, etc.). For handling geometric transformations, we do random rotations within $\sim 10^o$. We found that incorporating random crops gave poorer performance, so we did not use them. We also did not use horizontal flips, since the tiger identities are based on the visible flank (left or right side). We use small perturbations in brightness and contrast ($\sim 5\%$) to better handle lighting variations in the data. We also randomly convert some images to grayscale

Figure 4.3 Effect of JPEG compression: At the lower quality (higher compression), block-like color artifacts introduced by the compression are visible. While this change may seem insignificant to the human eye, it changes the internal statistics of the image. Compressing the images at different random quality values during training helps the network become robust to those statistical differences. Given the *fine-grained* nature of this visual recognition problem, we saw significant improvements in our empirical analysis

in order to reduce model dependency on color information. To help with variability in image quality, we adopt a random JPEG compression transform. This has been previously explored in the context of adversarial defences, but we hypothesize that it can have a regularizing effect against differences in internal image statistics caused by general image quality differences. Figure 4.3 shows the visual effect of different levels of JPEG compression. We randomly compress the images at each training iteration with compression quality values between 50 and 80. To the best of our knowledge, JPEG compression has not been used as a data augmentation technique. We validate its use in the experimental results.

b) **Flank Separation**

In the challenge dataset we also had access to key points, but we found that many of the images had missing or incorrect keypoint labels. Instead of using the ground-truth, we use keypoints generated from a keypoint-prediction network. The keypoint prediction network is an HR-net [160] trained using the noisy ground-truth keypoint annotations. To determine the flank orientation, we find the median x-value for the

fore keypoints (nose, ears, shoulders, front paws) and the hind keypoints (tail, hips, knees, back paws) and calculate the vector from hind to fore. If the vector points right we assign the image a right flank, otherwise a left flank. During evaluation, we re-rank the gallery so that all images with a flank orientation matching that of the query get put before those which don't match.

c) **Re-ranking with SIFT Matching**

Owing to the fact that SIFT [118] features are invariant to image scaling, rotation and partially invariant to viewpoint and illumination changes, they have been extensively used for individual recognition of zebras, jaguars and several other patterned animals in [18, 37, 28]. To avoid unnecessary keypoint detection and matching due to background clutter, all the previous works compute SIFT features on specific parts of the animal, like the cropped flank of the Jaguar. In addition, a query image is compared to all the database images to get the final match, making the process time consuming.

In our case, we compute features on the whole image but only use SIFT matching to re-rank the top 20 images ordered by the cosine similarity score, thus increasing the mAP and top-1 accuracy of the system in both single cam and cross cam scenarios. We also observed that using a larger number of images for re-ranking decreased the performance because of false matches in the background. For SIFT matching, we use the standard matching algorithm [118] that uses nearest neighbor matching, followed by Lowe's ratio test to reject false matches and finally computing the number of inliers by computing the homography.

4.4 Experimental Setup and Results for Primate Face Recognition

4.4.1 Dataset

We evaluate our model using three datasets, the details of which are given in Table 4.1. As is typical of wildlife data collected in uncontrolled environments, all the three datasets have a significant class imbalance as reported in Table 4.1.

Rhesus Macaque Dataset. The dataset is collected using DSLRs in their natural dwelling in an urban region in the state of Uttarakhand in northern India. The dataset is cleaned manually to remove images with no or very little facial content (e.g., extreme poses with only one ear or only back of head visible). The filtered dataset has 59 identities with a total of 1399 images. An illustrative set of pose variations for the datasets are shown using the cropped images in Figure 4.4. Due to the small size of this dataset, we combined our dataset with the publicly available dataset by Witham [174]. The combined dataset comprises 7679 images of 93 individuals. Note that we use the combined dataset only for the individual identification experiments, as the public data by Witham consists of pre-cropped images. On the other hand, the detection and the complete PFID pipeline is evaluated on a test set comprising full images from our macaque dataset.

Chimpanzee Dataset The C-Zoo and C-Tai dataset consists of 24 and 66 individuals with 2109 and 5057 images respectively [50]. The C-Zoo dataset contains good quality images of chimpanzees taken in a Zoo, while the C-Tai dataset contains more challenging images taken under uncontrolled settings of a national park. We combine these two datasets to get 90 identities with a total of 7166 images.

Dataset	Rhesus Macaques	C-Zoo	C-Tai
# Samples	7679	2109	5057
# Classes	93	24	66
# Samples/individual	[4,192]	[62,111]	[4,416]

Table 4.1 Datasets Summaries: The numbers in the brackets show the range of samples per individual ([min,max]), highlighting the imbalance in the datasets.

Figure 4.4 Pose variations in a Rhesus Macaque (Top) and a Chimpanzee face image (Below) from the dataset.

4.4.2 Evaluation Protocol

We evaluate and compare the performance of PFID system under four different experimental settings, namely: classification, closed-set identification, open-set identification and verification.

Classification. To evaluate the classification performance the dataset is divided into 80/20 train/test splits. We present the mean and standard deviation of classification accuracy over five stratified splits of the data. As opposed to other evaluation protocols discussed below, all the identities are seen during the training, with unseen samples of same identities in the test set.

Open and Closed-Set Identification. Both, closed-set and open-set performance is reported on *unseen* identities. We perform 80/20 split of data w.r.t. to identities, which leads

to a test set with 18 identities in test for both chimpanzee and macaque datasets. We again use five stratified splits of the data. For each split, we further perform 100 random trials for generating the probe and gallery sets. However, the composition of the probe and gallery sets for the closed-set scenario is different from that of open set.

Closed-Set: In case of closed-set identification, all identities of images present in the probe set are also present in the gallery set. Each probe image is assigned the identity that yields the maximum similarity score over the entire gallery set. We report the fraction of correctly identified individuals at Rank-1 to evaluate the performance.

Open-Set: In case of open-set identification, some of the identities in the probe set may not be present in the gallery set. This allows to evaluate the recognition system to validate the presence or absence of an identity in the gallery. To validate the performance, from the test of 18 identities, we used all the images of odd numbered identities as probe images with no images in the gallery. The rest of the even numbered identities are partitioned in the same way as closed-set identification to create probe and gallery sets. We report Detection and Identification Rate (DIR) at 1% FAR to evaluate open-set performance.

Verification. We compute positive and negative scores for each sample in the test set. The positive score is the maximum similarity score of the same class and negative scores are the maximum scores from each of the classes except the true class of the sample. In our case, where the test data has 18 identities, each sample is associated with a set of 18 scores, with one positive score from the same identity and 17 negative scores corresponding to the remaining 17 identities. The verification accuracy is reported as mean and standard deviation at 1% False Acceptance Rates (FARs).

4.4.3 Network Details and Hyper-Parameter Setting

We resize all the face images in macaque and chimpanzee dataset to 112×112. We add the following data augmentations: random horizontal flips and random rotations within 5 degrees

for both the datasets. We use the following base network architectures for PFID: ResNet-18 [70] and DenseNet-121 [79] and remove the first maxpool layer because of small image size. For CE setting, we fine-tuned the imagenet pre-trained networks with cross-entropy loss and a batch size of 16. For the PFID setting, for each image in a batch, a similar class image is sampled to make a batch size of 8 pairs (16 images in a batch). The dissimilar pairs are then exhaustively created from these pairs. We used SGD for optimization with an initial learning rate of 10^{-3} and weight decay of $5e-4$. We trained all the models for both datasets for 40 epochs with learning rate decay by 0.1 at 25^{th} and 35^{th} epoch. We observed better performance with batch size of 16 instead of 32 or higher especially in case of training with only cross-entropy loss. It is recommended to use a lower batch size given that the training data is less in both the datasets.

4.4.4 Results

We present the results corresponding to PFID and other state-of-the-art approaches for face recognition.

Baseline Results

For the baseline results, we extracted the penultimate (FC) layer features from both ResNet-18 and DenseNet-121 models. For all the evaluation protocols, the features are $l2$-normalized and in addition for classification, they are used to train a SVM (Support Vector Machine) classifier by performing a grid-search over the regularization parameter. The results are given in the first 2 rows of Table 4.2 and 4.3. We directly used the features and did not perform PCA (Principal Component Analysis) to reduce the number of feature dimensions because it had no impact on the performance in each evaluation.

Comparison with state-of-the-art approaches

We compare PFID with recent work PrimNet [41] that achieved state-of-the-art performance on chimpanzee face dataset. While we outperform PrimNet by a large margin, it is worth noting that our results are reported on non-aligned face images, that makes PFID better suited for the application of crowdsourced population monitoring by eliminating the need for manual annotations of fiducial landmarks. Since ResNet-18 and DenseNet-121 are pretrained on imagenet data, we additionally fined-tuned ArcFace [42] and SphereFace [112] models that are pre-trained on human face images, specifically on CASIA [187] dataset. We use ResNet-50 as the backbone for ArcFace and 20-layer network for SphereFace, and use the parameters given in the respective papers. We observed best performance with batch size 32 in all the three methods. We used a learning rate of 0.1, 0.01 and 0.001 for PrimNet (trained from scratch), SphereFace and ArcFace respectively and weight decay as $5e-4$. We trained all the models for 30 epochs to avoid over-fitting with learning rate decay by 0.1 at 15^{th} and 25^{th} epoch. The results are reported in Table 4.2 and 4.3 for both the datasets. The results highlight that the imagenet pre-trained models generalize well in our case where the training data is not huge. Further, it should be noted that the results reported for the three models ArcFace, SphereFace and PrimNet are also reported without face alignment as opposed to the results reported in the respective papers. While we report results with non-aligned face images, we would also like to point out that the performance dropped in all the approaches with aligned face images in case of chimpanzee dataset owing to loss of features in aligned faces.

PFID Results

To show the efficiency of our approach, we fine-tuned ResNet-18 and DenseNet-121 models with standard cross entropy (CE) loss and report results in Table 4.3 and 4.2 for macaque and chimpanzee datasets respectively and compared it with the PFID loss. We observe an

Method	Classification Rank-1	Closed-set Rank-1	Open-set Rank-1	Verification 1 % FAR
Baseline (ResNet-18 FC + SVM)	55.38 ± 1.18	70.51 ± 2.98	12.80 ± 5.73	37.10 ± 4.63
Baseline (DenseNet-121 FC +SVM)	61.78 ± 1.4	75.34 ± 3.98	30.51 ± 6.61	54.80 ± 3.65
ArcFace (ResNet-50)	85.47 ± 0.86	78.47 ± 5.81	41.24 ± 7.82	63.91 ± 5.37
SphereFace-20	78.38 ± 1.23	72.72 ± 3.44	35.49 ± 8.34	57.74 ± 6.38
PrimNet	70.86 ± 1.19	72.22 ± 5.33	37.27 ± 5.48	62.83 ± 5.98
CE (ResNet-18)	85.29 ±1.43	86.44 ± 5.42	48.62 ± 9.05	75.19 ± 8.16
CE (DenseNet-121)	86.74 ± 0.74	87.01± 5.39	53.60 ± 13.04	76.86 ± 9.55
PFID (ResNet-18)	88.98 ± 0.26	88.26 ± 5.01	59.36 ± 9.12	80.06 ± 6.62
PFID (DenseNet-121)	**90.78 ± 0.53**	**91.87 ± 2.92**	**66.24 ± 8.08**	**83.23 ± 6.07**

Table 4.2 Evaluation of Chimpanzee dataset for classification, closed-set, open-set and verification settings. Baseline results are reported by taking the penultimate layer features of the network and training a SVM for classification. For all the remaining settings the features are directly used for the evaluation protocol.

Method	Classification Rank-1	Closed-set Rank-1	Open-set Rank-1	Verification 1 % FAR
Baseline (ResNet-18 FC +SVM)	85.28 ± 0.25	88.29 ± 2.95	50.09 ± 7.35	66.98 ± 9.21
Baseline (DenseNet-121 FC +SVM)	88.3 ± 0.57	89.24 ± 3.63	53.93 ± 10.27	71.34 ± 8.88
ArcFace (ResNet-50)	98.23 ± 0.47	93.98 ± 2.99	67.07 ± 13.91	95.16 ± 1.56
SphereFace-20	97.61 ± 0.74	93.41 ± 2.19	95.62 ± 12.21	93.18 ± 1.95
PrimNet	97.11 ± 0.65	90.94 ± 2.54	65.98 ± 15.23	92.14 ± 2.82
CE (ResNet-18)	97.91 ± 0.58	95.94 ± 2.94	79.69 ± 8.12	96.35 ± 2.06
CE (DenseNet-121)	97.99 ± 0.69	96.24 ± 0.85	71.36 ± 10.05	96.01 ± 3.01
PFID (ResNet-18)	98.71 ± 0.41	96.18 ± 1.58	83.02 ± 7.36	97.71 ± 0.91
PFID (DenseNet-121)	**98.91 ± 0.40**	**97.36 ± 1.73**	**84.00 ± 7.43**	**98.24 ± 0.94**

Table 4.3 Evaluation of Rhesus Macaque dataset for classification, closed-set, open-set and verification settings. Baseline results are reported by taking the penultimate layer features of the network and training a SVM for classification. For all the remaining settings the features are directly used for the evaluation protocol.

increase in performance for the four evaluation protocols with PFID loss as compared to traditional cross entropy based fine-tuned network. Imposing a KL-divergence loss has im-

Model	Macaque	Chimpanzee
	NMI	NMI
CE	0.868 ± 0.008	0.686 ± 0.084
PFID	0.897 ± 0.030	0.715 ± 0.089

Table 4.4 Comparison of K-means clustering performance on the learned representations with DenseNet-121. The results highlight that PFID learns more clusterable space.

proved the discriminativeness of features by skewing the probability distributions of similar and dissimilar pairs. For chimpanzee dataset an improvement of **4.04%**, **4.86 %**, **12.64%** and **6.97 %** is achieved in case of classification, closed-set, open-set and verification settings respectively using DenseNet-121. The corresponding CMC (Cumulative Matching Characteristic) and TAR (True Acceptance Rate) vs FAR plots for the datasets are shown in Figure 4.5.

Feature Learning and Generalization

To further show the effectiveness of PFID loss function and robustness of features, we perform cross dataset experiments in Table 4.5. We used model trained on chimpanzee dataset and extracted features on macaque dataset to evaluate the performance for closed-set, open-set and verification task and vice versa. We compared the quality of the features learned with PFID with the features learned with cross entropy based fine-tuning. We also show the generalizability between two chimpanzee datasets captured in different environments i.e. C-Zoo and C-Tai. The results clearly highlight the advantage of PFID over cross entropy loss for cross data generalization. Additionally, to highlight the discriminativeness and clusterability of the class specific features, we cluster the feature representations of unseen (identities) test data using K-means clustering algorithm. We report the clustering performance in Table 4.4 and compare with the standard cross entropy loss.

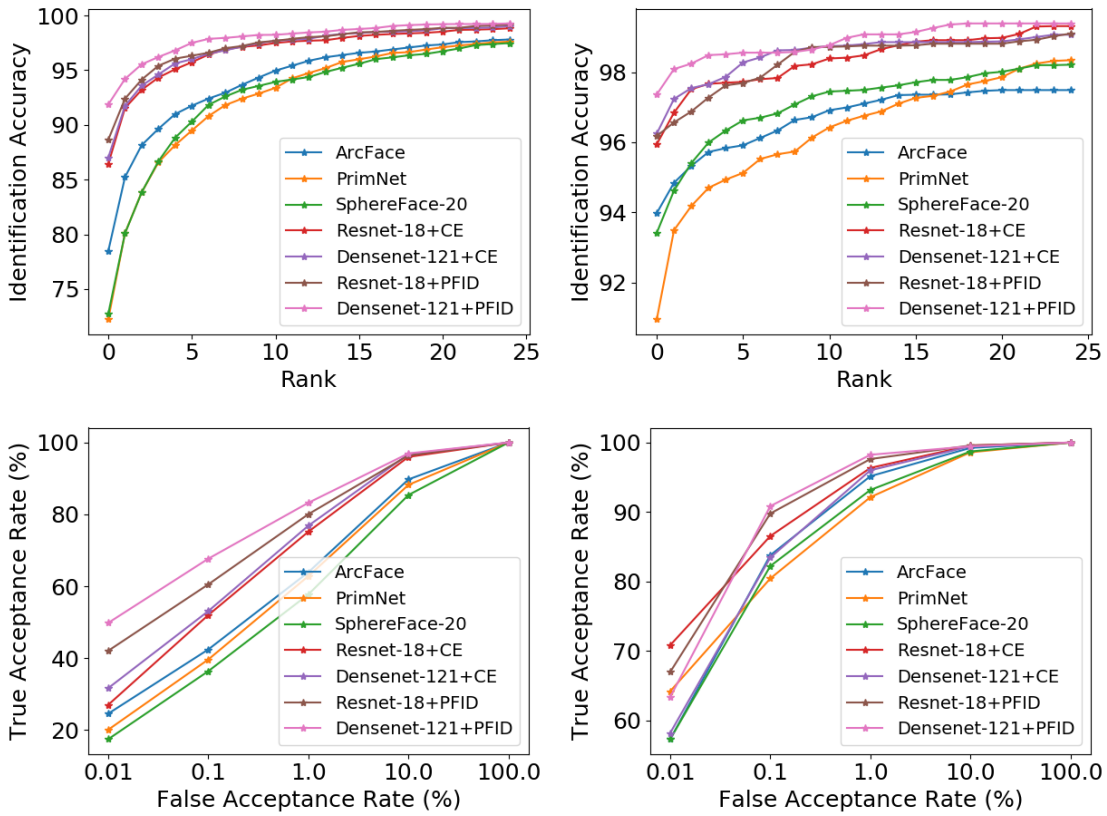


Figure 4.5 CMC (Top) and TAR vs FAR (Bottom) plots for (Left) C-Zoo+C-Tai and (Right) Rhesus Macaques datasets.

Comparison with Siamese Network based features

One might draw similarities between our approach and the popular siamese networks [150] that are trained on similar and dissimilar pairs to output a similarity score. We trained ResNet-18 on chimpanzee data in siamese setting with pairwise hinge-loss on features to show that the learned features in the classification setting are not discriminative as compared to our PFID. While training in siamese setting, we also observed that the network overfits on the training data and performs poorly on unseen classes. The results for different evaluation protocols are: Classification (83.97 ± 1.42), Closed-set (75.45 ± 5.51), Verification (57.28 ± 7.37) and Open-set (22.22 ± 8.07)

	Macq. → Chimp.		Chimp. → Macq.		C-Zoo → C-Tai		C-Tai → C-Zoo	
	CE	PFID	CE	PFID	CE	PFID	CE	PFID
Closed Set	54.58	**63.48**	83.02	**88.38**	59.92	**70.35**	87.54	**91.96**
Open Set	13.56	**34.29**	32.04	**43.00**	17.21	**27.21**	43.25	**64.75**
Verification	43.02	**63.77**	67.51	**75.37**	48.68	**60.57**	66.71	**82.22**

Table 4.5 Evaluation of learned model across datasets. Left of the arrow indicates the dataset on which the model was trained on, and right of the arrow indicates the evaluation dataset. All the results are reported for DenseNet-121 network.

Identification on Detected Face Images

The above results evaluated the performance of PFID on cropped face images *i.e.* the true bounding box of the test samples. As the captured images with handheld devices like cameras would also have background, we evaluate the performance of PFID on the detected faces on test samples. Since we had 1191 full images for the Macaque dataset, the detector is trained and tested with a split of 80/20. We fine-tune state-of-the-art Faster-RCNN [144] detector for detecting macaque faces and achieve highly accurate face detection performance. The identification results on the cropped faces obtained from the detector are shown in Table 4.6. For identification evaluation, we have 10 identities and 227 images for both closed-set and verification. For open-set, we extend the probe set by adding 8 identities and 1100 samples, which are not part of the dataset.

4.5 Experimental Setup and Results for Tiger Re-Identification

4.5.1 Dataset

The plain Re-ID dataset consists of 1887 training images distributed across 107 identities and 1764 images in the test set. The number of training images varies from minimum 10 to

Method	Closed-set Rank-1	Open-set Rank-1	Verification 1 % FAR
CE (ResNet-18)	95.00	70.78	89.22
PFID (ResNet-18)	97.20	78.80	91.11
CE (DenseNet-121)	95.30	80.67	91.56
PFID (DenseNet-121)	**97.80**	**89.67**	**95.11**

Table 4.6 Evaluation of detected macaque faces for closed set, open set and verification settings.

maximum 98 images per individual with an average of 18 images per individual. The wild Re-ID dataset consists of 1652 images in the test set which is the same as the detection track test set.

4.5.2 Network Details and Hyper-parameters

We use a pre-trained DenseNet-121 model and finetune the network with the objective function given in section 3.2, with an initial learning rate of 10^{-3} using SGD. The network is trained for 20 epochs with learning rate decay by 0.1 at 10 and 15 epochs. We use a batch size of 16 images. When training with KL-divergence, we sample 8 pairs of images, where each pair consists of two images from the same identity. The randomized JPEG transformation chooses a random value between 50 and 80 (maximum value 100).

4.5.3 Results for Plain Re-ID task

The description of compared methods is summarized in Table 4.7 and results for Plain Re-ID task are given in Table 4.8. The result in bold shows the performance of our approach in single as well as cross camera settings.

Method	Description
SIFT (Baseline)	No training is involved. The SIFT matching is done directly on the images without any pre-processing or data transformations.
CE	CE denotes finetuning the network with standard cross entropy loss, while using standard data augmentation that includes random affine, color jitter and random gray scale. During testing, the images are ranked based on cosine similarity.
CE+JPEG+LR+SIFT	Same as CE, with additional JPEG compression during training. During test, images are first ranked with cosine similarity, followed by Left/Right pose reordering and finally ranking top 20 entries with SIFT matching.
KLDiv+CE	This denotes finetuning the network with cross entropy loss augmented with the pairwise KL-divergence loss, using the standard data augmentation listed under CE. The testing is same as CE.
KLDiv+CE+JPEG+LR+SIFT (Proposed Method)	This denotes finetuning the network with cross entropy loss augmented with the pairwise KL-divergence loss, using the standard data augmentation along with JPEG. The testing is same as CE+JPEG+LR+SIFT.

Table 4.7 Brief description of various methods used in Tables 4.8 and 4.9 for the Re-ID task.

4.5.4 Ablation Study for Plain Re-ID task

In order to establish the efficacy of the proposed approach, we performed a set of experiments to gauge the relevance of different components that contribute to model performance.

SIFT Matching We use the standard SIFT matching to set up the baseline for tiger identification. Because the pre-cropped images in the Plain-ReID dataset have a lot of background clutter, the baseline matching causes a lot of false matches. When re-ranking only the top twenty images, we find that the SIFT matching in some cases improves the ranking of images which lie outside the top-5, giving much better performance across all metrics as seen in Tables 4.8, 4.9 and 4.10.

JPEG Transformations We also present the effect of randomized JPEG transformation in network finetuning. We observe that training with the proposed transformation improves performance, specifically in the cross camera setting where the images are more challenging, both in terms of image quality and pose variation. We also use JPEG compression during testing but with a fixed quality value of 65, so that noisy artifacts do not affect the test performance.

Left-Right Prioritizing We use the keypoints to identify the left and right flanks. We ob-

served that accounting for this information allows the system to avoid false matches between left and right flanks.

Relevance of KL-divergence Loss We present the standard cross entropy results with and without JPEG compression and re-ranking, to establish the improvement brought by adding the pairwise KL-divergence loss in both cases. The cross-entropy loss without KL-divergence, JPEG compression, or re-ranking performs much worse as can be seen in Table 4.8 and 4.9 for Test-dev and Full Test respectively.

Approach		**Single Cam**			**Cross Cam**		
	mmAP	mAP	Top-1	Top-5	mAP	Top-1	Top-5
SIFT (Baseline)	0.532	0.748	0.943	0.969	0.317	0.766	0.897
CE	0.603	0.754	0.920	0.966	0.453	0.806	0.931
CE+JPEG+LR+SIFT	0.657	0.817	0.977	0.983	0.498	0.851	0.937
KLDiv+CE	0.658	0.801	0.948	0.980	0.515	0.840	0.914
KLDiv+CE+JPEG+LR+SIFT	**0.691**	**0.847**	**0.986**	**0.986**	**0.535**	**0.891**	**0.940**

Table 4.8 Ablation Study for Plain Re-ID Task on Test-dev.

Approach		**Single Cam**			**Cross Cam**		
	mmAP	mAP	Top-1	Top-5	mAP	Top-1	Top-5
SIFT (Baseline)	0.538	0.749	0.930	0.970	0.327	0.768	0.909
CE	0.615	0.746	0.894	0.956	0.484	0.816	0.925
CE+JPEG+LR+SIFT	0.669	0.809	0.964	0.980	0.530	0.860	0.940
KLDiv+CE	0.662	0.791	0.923	0.969	0.533	0.833	0.926
KLDiv+CE+JPEG+LR+SIFT	**0.696**	**0.836**	**0.973**	**0.981**	**0.556**	**0.872**	**0.948**

Table 4.9 Ablation Study for Plain Re-ID Task for Full Test Data.

Approach	Detection	Data Split		Single Cam			Cross Cam		
			mmAP	mAP	Top-1	Top-5	mAP	Top-1	Top-5
CE+KLDiv+JPEG	0.8	Test-dev	0.64	0.74	0.85	0.92	0.54	0.84	0.90
		Full Test	0.65	0.75	0.86	0.92	0.55	0.85	0.92
CE+KLDiv+JPEG	0.5	Test-Dev	0.644	0.749	0.866	0.927	0.538	0.841	0.91
		Full Test	0.653	0.756	0.882	0.930	0.55	0.849	0.920
CE+KLDiv+JPEG+SIFT	0.8	Test-dev	0.654	0.773	0.902	0.925	0.535	0.834	0.918
		Full Test	0.662	0.777	0.913	0.932	0.547	0.844	0.926
CE+KLDiv+JPEG+SIFT	0.5	Test-dev	**0.658**	**0.780**	**0.916**	**0.937**	**0.536**	**0.835**	**0.920**
		Full Test	**0.667**	**0.787**	**0.927**	**0.946**	**0.548**	**0.845**	**0.928**

Table 4.10 Wild Re-ID Task Results. We report performance on the *Test-dev* and *Full Test* test sets at two different detection levels (0.8 and 0.5 detection confidence). Note that for wild Re-ID we don't use any pose information, including left-right flank filtering.

4.5.5 Results for Wild Re-ID

We also evaluate our approach on the wild Re-ID task. We use the same model trained for plain Re-ID on the plain Re-ID training dataset. We fine-tune an RFBNet [110] model on the detection dataset, and use detected bounding boxes with confidence scores greater than 0.5 and 0.8. We present the Re-ID results on both the Test-dev and Full Test datasets. Here, the benefit of using SIFT features during inference can be observed across all metrics in Table 4.10.

4.6 Relevance and Impact of Visual Wildlife Monitoring

Over the last several decades, technological progress has substantially improved human quality of life, albeit at a cost of rapid environmental degradation. Specifically, to meet the needs of the growing human population, various factors like urban and infrastructural development, agricultural land expansion and livestock ranching have resulted in soaring rates of deforestation. While on one hand it has caused the risk of extinction for many species, on the

other hand several other species have transitioned and adapted in urban dwellings, both of which is alarming for sustainable ecosystem.

4.6.1 Primate Face Recognition

Primates are one such species which have transitioned into a commensal relationship with humans, i.e., they rely on humans for food without causing direct harm. This species often dwell in close proximity to human settlements and this co-existence has led to indirect conflicts in the form of crop-raiding and property damage as well as occasional direct conflicts such as attacks or biting incidents. An example image of crop raiding and primates in close vicinity of humans is shown in Figure 4.6. Certain species like the rhesus macaque (*Macaca mulatta*) have become a cause of serious concern due to their resilience and ability to co-exist with humans in rural, semi-urban and urban areas. Their prolific breeding and short gestation periods lead to high population densities, thereby increasing the chances and extent of conflicts with humans.

Figure 4.6 Example images showing primates in human shared space and crop raiding [source: google images].

As a consequence, organizations have resorted to lethal conflict management measures like culling [5], which become infeasible when the conflicted species have declining populations, e.g., the human-primate conflict crisis in Sri Lanka where two of the responsible primate species are endangered: Toque macaques (*Macaca sinica*) and the purple faced langur (*Trachypithecus vetulus*) [25]. Besides, the effectiveness of lethal measures is well debated and poorly designed initiatives could have unexpected consequences like increased aggression

or even extinction of the conflicted species [131]. On the other hand, non-lethal approaches are easier to adopt across geographies as they avoid complex socio-religious issues [149]. Two recurring non-lethal themes in conflict management discussions are population monitoring and stakeholder engagement [131], both of which can be easily achieved with a combination of smartphone and AI technology. Pursuing a crowdsourcing approach to population monitoring and conflict reporting has two direct benefits: the cost and scalability of data collection for population monitoring can be improved drastically and active involvement of the affected community can help increase awareness, which in turn abates the human behavioral factors that often influence human-wildlife conflicts.

Inspired by the success and scalability of human face recognition, we propose a Primate Face Identification (PFID) system. Automatic identification capabilities could serve as a backbone for a crowdsourcing platform, where geo-referenced images submitted by users are automatically indexed by individuals, gender, age, etc. Such an indexed database could simplify downstream tasks like primate population monitoring and analysis of conflict reports, enabling better informed and effective strategies for conflict as well as conservation management.

4.6.2 Tiger Re-Identification

Tigers are among the many species that have become endangered over the last decade. There are several factors that have caused this population decline. While rapid deforestation is one factor, illegal poaching and trafficking is also an equal contributor. Conservation efforts are often driven by policy-level changes that may impose special restrictions on human activities like infrastructure building, logging, deforestation for agriculture and poaching and trafficking of endangered species and their body parts [3]. For example, using tiger skin and body parts are considered luxurious and prestigious items in many cultures and sold with huge prices both legally and illegally. Some example images for the same are shown in the

Figure 4.7 Example of tiger skin poaching and selling [3].

Figure 4.7. Active and frequent monitoring of endangered species populations is crucial in facilitating timely policy-level decisions, where delays may lead to species extinction. Based on population monitoring and census, specially targeted conservation efforts like captive breeding programs can be designed for effective recovery. However, traditional field-based methods of population monitoring like collaring are invasive, expensive, tedious and time-consuming, thus limiting their scalability and ultimately their success.

With increasing use of visual sensors like camera traps for passive monitoring of wildlife, data collection is substantially cheaper and more scalable. Advances in automated methods for population estimation could significantly reduce turn-around times, thus helping achieve the necessary conservation objectives for biodiversity preservation and sustaining the ecosystems in general. Individual identification is vital to population monitoring. Similar to unique faces in primates, tigers can be uniquely identified by the stripe pattern on their bodies. Therefore, we modified our primate face identification system and augmented it with traditional matching algorithms to develop a hybrid re-identification system for tigers.

4.7 Conclusion and Future Scope

As visual sensing becomes a preferred modality for monitoring wildlife, designing robust algorithms for applications like Re-ID of endangered species such as tigers and primates becomes important for scalable data analysis. In this work, we proposed a solution for tiger

Re-ID by fine-tuning a pretrained deep learning model while also leveraging standard SIFT-based image matching. In order to capture the wide range of data variations inherent in this task, such as pose, illumination, scale and image quality, we proposed to use a set of data transformations for augmentation during network fine-tuning. Additionally, to help mitigate the small number of samples per class, we enhanced the standard cross-entropy loss with a pairwise KL-divergence loss to explicitly enforce consistent semantically-constrained deep representations. We showed competitive results on the Plain Re-ID task using our approach, and further demonstrated its effectiveness when extended to the Wild Re-ID task, without using any pose information, thus highlighting the robustness of our Re-ID technique. We also showed through a series of ablation experiments that each component of our proposed approach helps contribute to a robust and general solution to the tiger re-identification problem. We foresee that this identification system could become a part of widely used wildlife management tools like SMART3.

Chapter Five

Semi-supervised Representation Learning for Clustering

Clustering is a classical unsupervised learning problem that seeks insights into the underlying structure of data by naturally grouping similar objects together. DNNs have proven effective in unsupervised learning with architectures like autoencoders, which learn representations that consistently reconstruct the input. This capability has led to deep clustering methods [178, 43, 64, 182] that rely on autoencoders, or on convolutional neural networks (CNNs) [76, 183]. For using DNNs to learn clusterable representations, two approaches are commonly used: methods that explicitly compute cluster centers [178, 64, 182] and methods that directly or indirectly model the data distribution [76, 43].

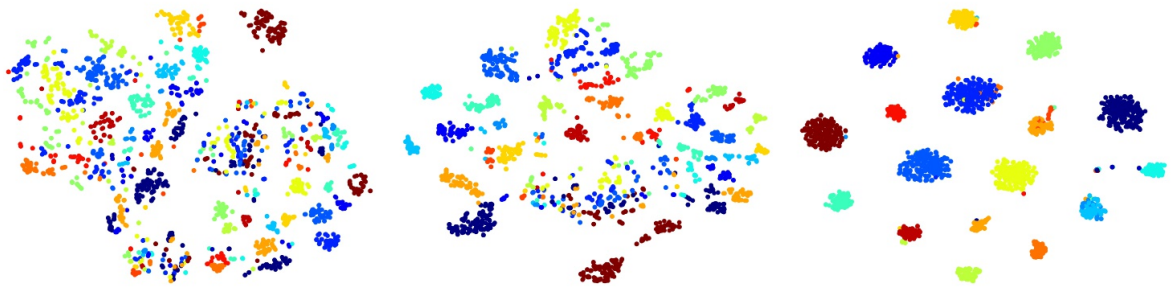


Figure 5.1 t-SNE plots for FRGC dataset: Raw data (Left Image), after unsupervised training of autoencoder (Center Image) and after training with clustering loss (Right Image, 2% labeled data)

While clustering has traditionally been an unsupervised learning problem, many DNN based clustering approaches have incorporated varying degrees of supervision, which can substantially improve the clustering performance. Based on the level of supervision, clustering approaches can be categorized as unsupervised, supervised and semi-supervised. Fully supervised approaches like [99, 157] perform well but were often limiting due to the unavailability of sufficient labeled data. On the other hand, approaches like [178, 64] are unsupervised and rely on unlabeled data alone for learning the network parameters, assuming only the knowledge of the number of clusters. Fully unsupervised approaches, while very desirable, are oblivious of any semantic notions of data, and may lead to learned representations that are difficult to interpret and analyze. Weakly or semi-supervised approaches are often used as a reasonable middle-ground between the two extremes. For example, [76] uses pairwise constraints as a weaker form of supervision, and [49] exploits both, labeled as well as unlabeled data for learning.

5.1 Overview of the Contribution

In this chapter, we introduce *ClusterNet*, an autoencoder based architecture that learns latent representations suitable for clustering. The training works in a semi-supervised setting utilizing the abundant unlabeled data augmented with pairwise constraints generated from a few labeled samples. An example is shown in Figure 5.1, that shows the effect of using a small amount of labeled data on the feature embedding space. In addition to the usual reconstruction error, ClusterNet uses an objective function that comprises two complementary terms: a k-means style clustering term that penalizes high intra-cluster variance and a pairwise KL-divergence based term that encourages similar pairs to have similar cluster membership probabilities.

Key Highlights

- *An approach that relies on a simple convolutional autoencoder architecture to learn clusterable latent representations in an end-to-end fashion.*
- *A loss function that combines a pairwise KL-divergence loss with a k-means loss to complement each other and simultaneously learn the cluster centers as well as clusterable representations.*
- *The approach requires minimal hyper-parameter tuning and is easy to train, due to an annealing strategy that adaptively trades off the importance of labeled and unlabeled data during the training.*

5.2 Existing Literature

In this section, we restrict our discussion to recent progress in clustering in a deep learning framework. Many of the clustering approaches leveraged the autoencoder due to its ability to learn representations in an unsupervised manner. Xie *et al.* [178] employed the encoder of a layer-wise pretrained denoising autoencoder to obtain both the latent space representations and cluster assignment simultaneously. Guo *et al.* [64] improved upon this approach by incorporating an autoencoder to preserve the local structure of the data. While these two approaches used fully connected multi-layer perceptron (MLP) for encoder and decoder architectures, work in [65] used a convolutional autoencoder to effectively handle image data. Apart from encoder-decoder architectures, (CNN) based architectures had also been applied to the clustering problem. For example, the work in [183] proposed JULE takes an agglomerative clustering approach that utilizes a CNN based architecture. While this approach performed well, it required a large number of hyper-parameter tuning, limiting

its applicability in real world clustering problems. Further, being fully unsupervised, these approaches may not learn a semantically meaningful representation, which can subsequently affect the clustering performance. Huang *et al.* [43] proposed an unsupervised approach, DEPICT, that also used a convolutional autoencoder stacked with a softmax layer in the latent space. A KL-divergence loss was used to optimize for the cluster assignments in addition to a regularization term for discovering balanced clusters. Such a loss tends to limit its capability to handle uneven class distributions. While this approach was extended to the semi-supervised learning scenario and showed out of sample generalization, it required fine-tuning the network with cross entropy loss as done in [141, 190].

More recently, several deep learning approaches [141, 190, 87] had utilized both labeled and unlabeled data for classification. However semi-supervised clustering was not well explored. Work in [76] used pairwise constraints on the data and enforced small divergence between similar pairs while increasing the divergence between dissimilar pair assignment probability distributions. However, this approach did not leverage the unlabeled data, leading to suboptimal performance in small labeled data settings. Fogel *et al.* [49] proposed a clustering approach using pairwise constraints, that were obtained by considering a mutual KNN graph on the unlabeled data for unsupervised clustering. The method further extended to semi-supervised approach by using few labeled connections that were defined using the distances defined on the embedding obtained from the initial autoencoder training. However, several parameters like distance threshold and number of neighbors required for mutual KNN (MKNN) were user defined and were crucial to the performance of the algorithm.

5.3 Proposed Approach

In this section, we present our approach, the network architecture and loss function, and explain the optimization strategy. We consider the following setup for ClusterNet. Let $\mathcal{X}^l = \cup_{k=1}^{K}\{\mathcal{X}_k^l\}$ denote the set of labeled data corresponding to K classes. Here $\mathcal{X}_k^l =$

$\{\mathbf{x}_1^l, \mathbf{x}_2^l..\mathbf{x}_p^l\} \in \mathbb{R}^n$ is the set of labeled points in k^{th} class. Let $\mathcal{X}^u$ denote a set of unlabeled data points. The goal is to learn a feature representation that is amenable to forming K clusters in a manner that similar pairs of points tend to belong to the same cluster while the dissimilar pairs do not.

5.3.1 Network Architecture

We build our clustering model using a convolutional autoencoder architecture. The parameters for encoder and decoder are represented by θ_e and θ_d respectively. The cluster centers are given by $\mu \in \mathbb{R}^d$, where d is the dimensionality of the latent space. We denote the cluster membership of the i^{th} point using a one-hot encoding vector $\mathbf{a}_i \in \{0,1\}^K$, such that $a_{i,j} = 1$ implies that the i^{th} point lies in the j^{th} cluster. Further, we use $\mathbf{z} = f(\theta_e; \mathbf{x})$ to denote latent space representations, where $f(\theta_e; \dot{)} : \mathbf{x} \rightarrow \mathbf{z}$ is a nonlinear mapping from the input space to the latent space. Lastly, $g(\theta_d, \mathbf{z})$ represents the output of the autoencoder that maps the latent space representations to the reconstructed input space. For all our experiments, we have fixed both the network architecture as well as the dimension of the latent space to show that clustering performance of ClusterNet is sustained across multiple datasets.

5.3.2 Objective Function

Our semi-supervised clustering loss function has three components: pairwise loss, cluster loss and reconstruction loss, and is defined as

$$\mathcal{L} = \mathcal{L}_{pair} + \mathcal{L}_{cluster} + \mathcal{L}_{recon} \tag{5.1}$$

The reconstruction loss is common for all data, but the cluster and pairwise losses are defined slightly differently for the labeled and the unlabeled data The reconstruction loss serves as a regularizer for ensuring that the representation has high fidelity, while the other two terms make the latent space more amenable to forming semantically consistent clusters. Next, we discuss each component of the loss term in detail.

Cluster Loss: Similar to k-means loss term, the cluster loss aims to minimize the distance between latent space representations of samples with the corresponding cluster centers to encourage clusterable representations. The corresponding loss for both labeled and unlabeled data are given by

$$\mathcal{L}_{cluster} = \frac{1}{|\mathcal{X}^l|} \sum_{\mathbf{x}_j \in \mathcal{X}^l} ||f(\theta_e; \mathbf{x}_j^l) - \mathbf{C}\mathbf{a}_j^l||_2^2 + \frac{\lambda}{|\mathcal{X}^u|} \sum_{\mathbf{x}_i \in \mathcal{X}^u} ||f(\theta_e; \mathbf{x}_i^u) - \mathbf{C}\mathbf{a}_i^u||_2^2 \tag{5.2}$$

Here, $\mathbf{C} \in \mathbb{R}^{d \times K}$ is a matrix with each column corresponding to one cluster center, $\mathbf{a}_i^u$ is the *predicted* label assignments for an unlabeled sample i and $\mathbf{a}_j^l$ is the *true* label for a labeled sample j. λ acts as a balancing coefficient and is important for the performance of our algorithm. A high value of λ suppresses the benefits of labeled data, whereas a very small value of λ fails to use the unlabeled data effectively during training. Therefore, we resort to a deterministic annealing strategy [58, 101], that gradually increases the value of λ with time and tends to avoid poor local minima.

Pairwise Loss: The KL-divergence loss utilizes similar and dissimilar pairs and encourages similar cluster assignment probabilities for similar point pairs, while ensuring a large divergence between assignment probabilities of dissimilar pairs. We write the pairwise loss as

$$\mathcal{L}_{pair} = \mathcal{L}_{pair}^l + \lambda \mathcal{L}_{pair}^u \tag{5.3}$$

For labeled data, the pairwise loss is given by

$$\mathcal{L}_{pair}^l = \frac{1}{|\mathcal{T}_{sim}^l|} \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{T}_{sim}^l} \mathcal{L}_{sim} + \frac{1}{|\mathcal{T}_{dsim}^l|} \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{T}_{dsim}^l} \mathcal{L}_{dsim} \tag{5.4}$$

where

$$\mathcal{L}_{sim} = d_{KL}(\mathbf{p}||\mathbf{q}) + d_{KL}(\mathbf{q}||\mathbf{p})$$

$$\mathcal{L}_{dsim} = [0, m - d_{KL}(\mathbf{p}||\mathbf{q}))]_+ + [0, m - d_{KL}(\mathbf{q}||\mathbf{p})]_+$$

Here, $\mathbf{p}$ and $\mathbf{q}$ are $K-$ dimensional vectors denoting the cluster assignment probabilities for points $f(\theta_e; \mathbf{x}_p)$ and $f(\theta_e; \mathbf{x}_q)$ obtained based on the distances from the cluster centers. The

margin m is introduced to impose a minimum separability between dissimilar pairs. The sets, $\mathcal{T}^l_{sim}$ and $\mathcal{T}^l_{dsim}$ are the similar and dissimilar image pairs respectively generated using the labeled data. Similarly, the labels based on cluster membership are used to compute the pairwise loss for unlabeled data as

$$\mathcal{L}^u_{pair} = \frac{1}{|\mathcal{T}^u_{sim}|} \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{T}^u_{sim}} \mathcal{L}_{sim} + \frac{1}{|\mathcal{T}^u_{dsim}|} \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{T}^u_{dsim}} \mathcal{L}_{dsim} \tag{5.5}$$

Here $\mathcal{L}_{sim}$ and $\mathcal{L}_{dsim}$ are defined as before.

Reconstruction Loss: While we want the latent space representations to form clusters, the cluster loss and pairwise loss can lead to a degenerate latent space where points tend to collapse to the cluster center, in an attempt to minimizing the cluster loss. This degeneracy may lead to poor generalization to unseen points. To mitigate this effect, we regularize the clustering loss by adding a reconstruction loss term for the autoencoder given by

$$\mathcal{L}_{recon} = \sum_{\mathbf{x} \in \{\mathcal{X}^u \cup \mathcal{X}^l\}} ||g(f(\mathbf{x}_i)) - \mathbf{x}_i||_2^2 \tag{5.6}$$

5.3.3 Network Optimization

We alternately optimize for the network parameters and the cluster center parameters. For fixed cluster center parameters *i.e.* $\mathbf{C}$ and $\mathbf{a}$, we update the network parameters θ_e and θ_d using standard backpropagation algorithm. The steps are described below and are summarized in Algorithm 1.

Cluster Center Parameters

Cluster Center Initialization: Motivated by constrained k-means, we initialize the cluster centers by the mean of the embedding corresponding to the samples in the corresponding labeled set $\mathcal{X}^l$

$$\mu_k = \frac{1}{|\mathcal{X}_k^l|} \sum_{\mathbf{x} \in \mathcal{X}_k^l} f(\theta_e; \mathbf{x}) \quad for\ k = 1, \cdots K \tag{5.7}$$

Cluster Assignment: For given cluster centers $\mathbf{C} = \{\mu_1, \mu_2 \cdots \mu_K\}$ and the embeddings in the latent space, similar to constrained k-means, the class or cluster associated with a data sample $\mathbf{x}_i$ is obtained by Eqs. (5.8) and (5.9) for unlabeled and labeled data respectively.

$$a_{j,i}^u = \begin{cases} 1 & \text{if } j = \underset{k=\{1,2,\cdots K\}}{\arg\min} ||f(\theta_e; \mathbf{x}_i^u) - \mu_k|| \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

$$a_{j,i}^l = \begin{cases} 1 & \text{for } j = k \quad \text{for} \quad \mathbf{x}_i \in \mathcal{X}_k^l \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

Cluster Center Updates: A naive way to update cluster centers is to follow the k-means strategy and compute the mean of the samples allocated to a cluster by looking at the current values of $\mathbf{a}^u$ and $\mathbf{a}^l$. This approach can pose a serious challenge due to a dynamic representation space. While $\mathbf{a}_i^l$ is informative and denotes the true cluster membership, the predicted unlabeled data assignments $\mathbf{a}_i^u$ may not be accurate and can introduce a significant bias in the updated centers.

Therefore, similar to [182], we update cluster centers using gradient updates with an adaptive learning rate. Further, we utilize the labeled data to update the corresponding true cluster center as

$$\mu_k \leftarrow \mu_k - \frac{1}{N_k^l}(\mu_k - f(\theta_e, \mathbf{x}_i^l)) \quad \text{for } \mathbf{x}_i^l \in \mathcal{X}_k^l \tag{5.10}$$

The unlabeled data update the cluster centers corresponding to the predicted cluster label in Eq. (5.8)

$$\mathbf{C}\mathbf{a}_i^u \leftarrow \mathbf{C}\mathbf{a}_i^u - \frac{1}{N_k^u}(\mathbf{C}\mathbf{a}_i^u - f(\theta_e, \mathbf{x}_i^u)) \tag{5.11}$$

Here, N_k^l and N_k^u are the number of labeled and unlabeled samples respectively, that have been assigned to cluster k in the current iteration.

Pairwise Constraints: The pairwise loss in Eqs. (5.4) and (5.5) is defined on similar and dissimilar image pairs. For labeled data, these pairs are created using the label information as follows:

$$\begin{aligned}
\mathcal{T}_{sim}^l &= \{(i,j) : \mathbf{x}_i \in \mathcal{X}_{k_1}^l, \mathbf{x}_j \in \mathcal{X}_{k_2}^l, k_1 = k_2\}, \\
\mathcal{T}_{dsim}^l &= \{(i,j) : \mathbf{x}_i \in \mathcal{X}_{k_1}^l, \mathbf{x}_j \in \mathcal{X}_{k_2}^l, k_1 \neq k_2\}, \\
& k_1,\ k_2 \in \{1, 2, \cdots, K\}
\end{aligned} \tag{5.12}$$

Similarly, the pairs are also computed for unlabeled data based on the predictions given by the Eq. (5.8)

$$\begin{aligned}
\mathcal{T}_{sim}^u &= \{(i,j) : \mathbf{x}_i \in \mathcal{X}_{k_1}^u, \mathbf{x}_j \in \mathcal{X}_{k_2}^u, k_1 = k_2\}, \\
\mathcal{T}_{dsim}^u &= \{(i,j) : \mathbf{x}_i \in \mathcal{X}_{k_1}^u, \mathbf{x}_j \in \mathcal{X}_{k_2}^l, k_1 \neq k_2\}, \\
& k_1,\ k_2 = \{1, 2, \cdots K\}
\end{aligned} \tag{5.13}$$

Assignment Probabilities: In many approaches, for a given sample the cluster assignment probabilities are obtained by a softmax layer with learnable parameters [76, 43]. Instead, we simply define probabilities based on distances from the cluster centers as

$$\begin{aligned}
p_{k,i} &= \frac{\exp(-d_{k,i})}{\sum_{k=1}^{K} \exp(-d_{k,i})};\ k = 1, 2, \cdots K \\
\text{where} \quad d_{k,i} &= ||f(\theta_e; \mathbf{x}_i) - \mu_k)||_2^2
\end{aligned} \tag{5.14}$$

Here, $p_{k,i}$ is the probability of assigning the i^{th} sample to the k^{th} cluster.

5.4 Experimental Setup and Results

In this section, we evaluate the performance of ClusterNet on several benchmark datasets and compare it with state-of-the-art approaches. We present evaluation results of our approach

Algorithm 1 ClusterNet Optimization

Input: $\mathcal{X}^u = \{\mathbf{x}_1^u, \mathbf{x}_2^u, \cdots \mathbf{x}_m^u\}$: set of unlabeled data points, $\mathcal{X}_k^l = \{\mathbf{x}_1^l, \mathbf{x}_2^l, \cdots \mathbf{x}_p^l\}$: set of labeled data in class k, $\mathcal{X}^l = \cup_{k=1}^K \mathcal{X}_k^l$, T : number of iteration

Output: $\mathbf{C} = \{\mu_1, \mu_2, \cdots \mu_k\}$: Cluster centers, θ_e: encoder parameters

Method:

1. Initialize cluster centers $\mu_k = \frac{1}{|\mathcal{X}_k^l|} \sum_{\mathbf{x} \in \mathcal{X}_k^l} \mathbf{x}$
2. for $t = 1 \cdots T$
   - I Create similar and dissimilar image pairs $\mathcal{T}_{sim}^l$ and $\mathcal{T}_{dsim}^l$ respectively from labeled data $\mathcal{X}^l$
   - II Compute labels for unlabeled data $\mathcal{X}^u$ using cluster assignments from Eq. 5.8
   - III Create similar and dismal pairs $\mathcal{T}_{sim}^u$ and T_{dsim}^u respectively using the label assignment
   - IV Compute assignment probabilities for both labeled and the unlabeled data using Eq. (5.14)
   - V Compute cluster center updates with labeled and unlabeled data using Eq. 5.10 and 5.11.
   - VI Update network parameters θ_e and θ_d
3. Return cluster centers $\mathbf{C} = \mu_1, \mu_2, \cdots, \mu_K$ and θ_e

for both clustering and classification problems.

5.4.1 Datasets

We use 5 datasets for experiments: two handwritten digits datasets; MNIST [100] and USPS [1] and three face datasets; FRGC [2], CMU-PIE [155] and YouTube Faces [175]. The details of the datasets are given in Table 5.1. For YTF dataset, we use first 41 subjects, sorted

[1] https://cs.nyu.edu/~roweis/data.html

[2] https://sites.google.com/a/nd.edu/public-cvrl/data-sets

according to their names in alphabetical order as in DEPICT [43]. For FRGC dataset, as in [43] we use 20 randomly selected subjects for our experiments.

	MNIST-train	USPS	FRGC	CMU-PIE	YouTube Faces
# Samples	60.000	11,000	2462	2856	10,000
# Classes	10	10	20	68	41
Size	32×32	16×16	$32 \times 32 \times 3$	$32 \times 32 \times 3$	$55 \times 55 \times 3$

Table 5.1 Dataset Description

	MNIST-test	USPS	FRGC	CMU-PIE	YouTube Faces
#Samples	10,000	1100	246	285	1000

Table 5.2 Samples to evaluate the performance of ClusterNet on unseen data samples. These samples are new to the network and have not been used by the network for training the network for clustering or in the pretraining stage.

5.4.2 Comparison with state-of-the-art approaches

ClusterNet is a semi-supervised approach that uses both labeled and unlabeled data. Since semi-supervised approaches are not explored for end-to-end clustering, we present comparisons with state-of-the-art unsupervised clustering approaches as well as approaches that can be adapted to operate in the semi-supervised setting.

Fogel *et al.* [49]: The approach works in both unsupervised as well as semi-supervised setting. In order to bring supervision in the framework, while they define pairwise connections from unlabeled data, they either manually label or use ground truth to generate some data pairs as similar and dissimilar. In Table 5.3, we present a comparative analysis of our results with the best reported results in [49] with $5k$ labeled connections.

Kira *et al.* [76]: This approach utilizes labeled data to create similar and dissimilar constraints, however do not use any unlabeled data. Therefore, their performance is suboptimal in case of low labeled data setting.

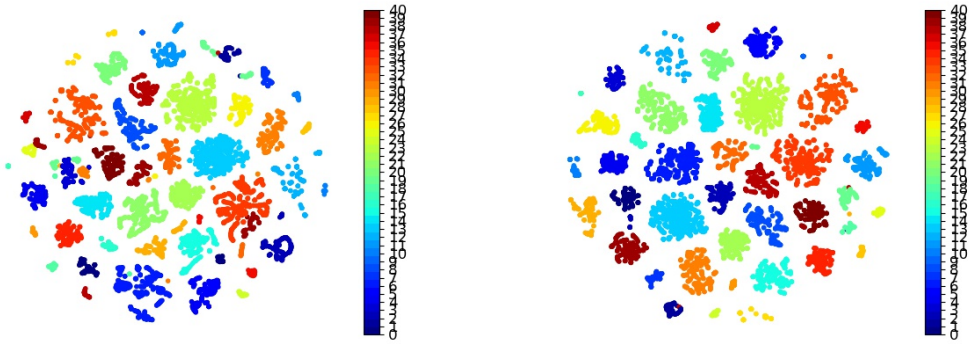
Datasets	Evaluation	DEC [178]	JULE-RC [183]	DEPICT [43]	CPAC-VGG [49]	ClusterNet	
MNIST-full	NMI	0.816	0.913	0.917	-	**0.926**	**0.94**
	ACC	84.4	96.4	96.50	-	**96.84**	**97.72**
Labeled Data		-	-	-	-	**0.5%**	**1%**
USPS	NMI	0.586	0.913	0.927	0.914	0.923	**0.936**
	ACC	61.9	95.0	96.4	86.7	**96.51**	**97.26**
Labeled Data	-	-	-	-	5k connect	**2 %**	**5 %**
FRGC	NMI	0.504	0. 574	0.610	0.799	**0.873**	**0.955**
	ACC	37.8	46.1	47.0	54.0	**82.24**	**95.77**
Labeled Data	-	-	-	-	5k connect	**2 %**	**5 %**
CMU-PIE	NMI	0.924	**1.00**	0.974	0.849	**1.00**	**1.00**
	ACC	80.1	**100**	88.3	68.8	**1.00**	**100**
Labeled Data	-	-	-	-	5k connect	**2%**	**5%**
YouTube Face	NMI	0.446	0.848	0.802	0.860	**0.932**	**0.988**
	ACC	37.1	68.4	62.1	54.2	**90.31**	**98.58**
Labeled Data	-	-	-	-	5k connect	**2 %**	**5 %**

Table 5.3 Performance comparison of different algorithms on several datasets based on NMI (normalized Mutual Information) and ACC (Accuracy in %). The results for various approaches are reported from the original paper. CPAC-VGG uses a number of labeled connections and ClusterNet uses % labeled images/class.

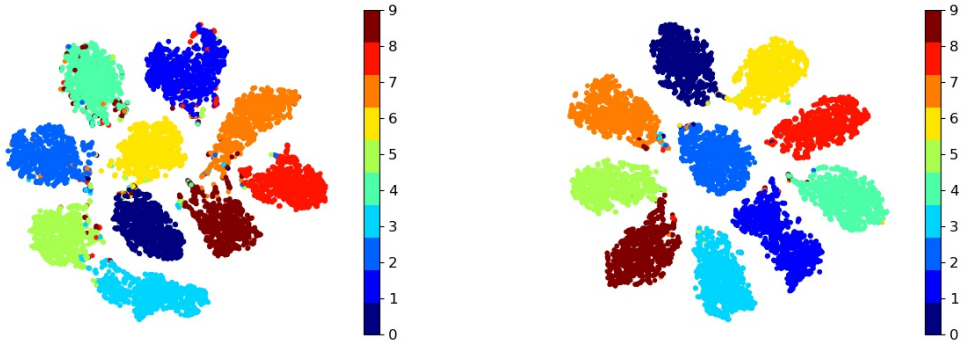
ClusterNet outperforms the best clustering accuracy reported for MNIST dataset in [76] by a margin of **15.71 %**, **3.26 %** and **0.26 %** respectively for 6, 60 and 600 samples/class respectively. The unsupervised approaches used for reference are: JULE [183], DEC [178] and DEPICT [43].

5.4.3 Network Details

We used a common architecture for ClusterNet across all the datasets, and show that its clustering performance generalizes well. We used three convolution layers with 32, 64 and 128 filters respectively, followed by a fully connected layer of dimension 32. Each convolution



(a) YTF dataset



(b) MNIST dataset

Figure 5.2 t-SNE plots for two different percent of labeled data (a) YTF (Left : 2% labeled data, Right: 5% labeled data) and (b) MNIST (Left: 0.1% labeled data, Right: 0.5% labeled data)

layer is followed by InstanceNorm layer and Leaky ReLU activation, while the fully connected layers in both encoder and decoder use a tanh activation function. The stride and padding values are chosen appropriately for the dataset. For each of the datasets, we pre-train the autoencoder end-to-end for 100 epochs using Adam optimizer, with a learning rate of 0.0001 and $\beta = (0.9, 0.999)$ to minimize the reconstruction error. We use a dropout of 10% at each layer, except the last layer of the decoder. We use this pre-trained network and fine tune it for clustering loss for 60 epochs, again with the Adam optimizer and a learning rate of

0.0001. The value of balancing coefficient is determined by eq. 5.15.

$$\lambda = \begin{cases} 0 & t < T_1 \\ \frac{t-T_1}{T_2-T_1} & T_1 \leq t < T_2 \\ 1 & T_2 \leq t \end{cases} \tag{5.15}$$

Here, t denotes the current epoch, $T_1 = 5$, $T_2 = 40$ for all the experiments.

5.4.4 Evaluation Metric

We use two standard metrics to evaluate the performance of proposed approach: NMI (normalized mutual information) that computes the normalized similarity measure between between true and predicted labels for the data and is defined as

$$NMI(c, c') = \frac{I(c; c')}{max(H(c), H(c'))} \tag{5.16}$$

Here, c and c' represents true and predicted class label, $I(c, c')$ is the mutual information c and c' and $H(\cdot)$ denotes the entropy. We also report the classification accuracy (ACC) that gives the percentage of correctly labeled samples. For our approach, the predicted label is the cluster with minimum distance from the given sample point.

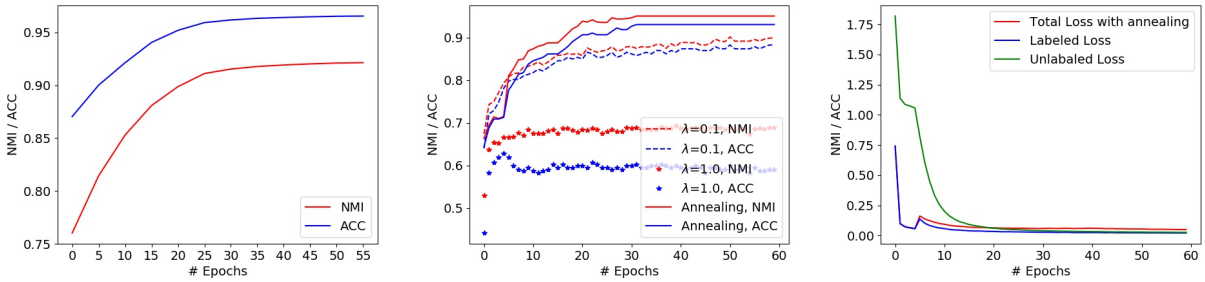
5.4.5 Clustering and Classification Results

We compare the performance of ClusterNet with other clustering approaches on all the datasets. For our approach, we show the average results achieved over five runs. For each of the datasets except MNIST, we first randomly split the data using stratified sampling with 90% for training and hold-out 10% for testing. We further generate five random splits from training data using stratified sampling into labeled and unlabeled data. For other approaches, we quote the results reported in the original reference. Table 5.3 summarizes both the classification as well as the clustering performance comparisons. The results show that *ClusterNet* outperforms state-of-the-art approaches by using small amounts of labeled

data (0.5, 2%and5%). The high NMI scores for ClusterNet can also be corroborated by the clusters formed in the t-SNE based latent space visualization shown in Figure 5.2 for the YouTube Face and MNIST datasets. The visible structure in the figures is typical and similar visualizations are obtained for other datasets as well. We also show the effect of training the pre-trained autoencoder with the clustering loss in Figure 5.1 and the corresponding NMI results in Table 5.4. As before, the t-SNE plots shown on the FRGC dataset is a typical indication of a clusterable latent space.

Datasets	Autoencoder	ClusterNet	
	All data	Train	Test
CMU-PIE	0.788 ± 0.006	1 ± 0.0	1 ± 0.0
FRGC	0.485 ± 0.012	0.945 ± 0.002	0.951 ± 0.003
YTF	0.761 ± 0.003	0.949 ± 0.0	0.953 ± 0.0
USPS	0.491 ± 0.001	0.931 ± 0.0	0.961 ± 0.0
MNIST	0.542 ± 0.0	0.942 ± 0.0	0.954 ± 0.0

Table 5.4 Comparison of K-means clustering performance (NMI) of autoencoder embeddings with ClusterNet embeddings



(a) (b) (c)

Figure 5.3 (a) Consistency in NMI and ACC over epochs for unlabeled USPS digits training data, (b) Effect of λ (balancing coefficient) on NMI and ACC on FRGC dataset (c) Loss function plot for YTF dataset with 2% labeled data

5.5 Analysis

5.5.1 Cluster Purity

At training time, we make use of predicted labels for the unlabeled data to compute the clustering loss. It is crucial for most predicted labels to be relatively consistent with the true class labels, or the training may diverge. In Figure 5.3a, we plot the test accuracy and NMI over the training epochs for the USPS dataset as a typical example. We point out that the annealing strategy Eq. (5.15) applied to the balancing coefficient λ is important for stabilizing the clustering.

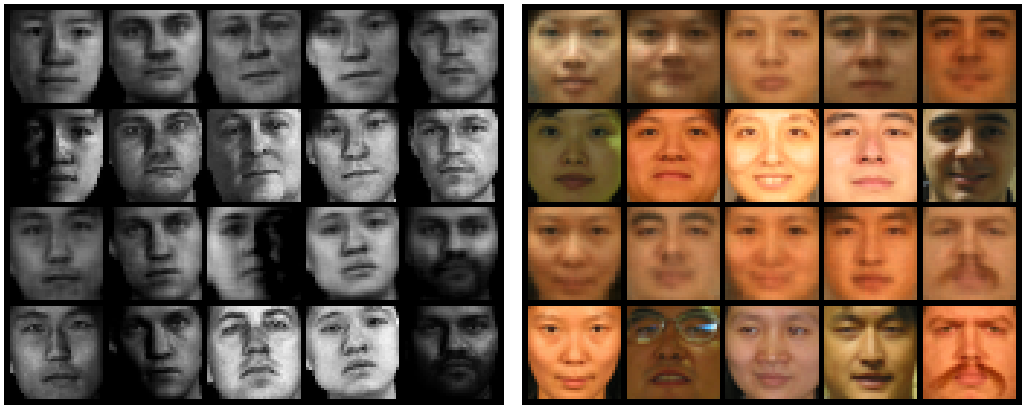
5.5.2 Effect of Balancing Coefficient

In Figure 5.3b, we show the effect of λ on the test accuracy and NMI values on the FRGC dataset. Since the labeled data is typically much smaller than the unlabeled data, it is important to balance the loss contributed by unlabeled data for achieving good representations and clustering performance. Too small a value of λ under-utilizes the unlabeled data, while too large a value may lead the training loss to be too high. The performance is significantly better when an annealing strategy is used to adapt λ throughout the training process that gradually increases the contribution of unlabeled data as training progresses. Further, we also observe a smooth convergence as shown in the loss function plot for Youtube Face dataset in the Figure 5.3c.

5.5.3 Cluster Centers

Since our approach uses labeled data for initializing the cluster centers and also update them using the corresponding labeled data samples over the epochs, each cluster center is representative of one of the true classes decided by the labeled samples. This one-to-one correspondence between clusters and classes provides the flexibility to use the cluster centers

directly for classification of unseen data points. We show the reconstructions of the learned cluster centers and a sample face image from the same class for two of the datasets in Figure 5.4.



(a) CMU-PIE **(b)** FRGC

Figure 5.4 Comparison of few reconstructed images corresponding to few cluster centers (1^{st} and 3^{rd} row) in the latent space with sample images (2^{nd} and 4^{th} row) from the respective clusters.

5.5.4 Performance with Varying Label Data

We present our results for the face and digit datasets, with varying number of labeled samples in Tables 5.5 and 5.6 respectively. The number of data samples used for testing for each of the datasets is given in Table 5.2. We use a maximum of 10% labeled data for training the network, but our technique significantly improves the performance over unsupervised approaches for most datasets with as low as 2% labeled data. Further, our approach is more suitable for practical clustering tasks as opposed to existing semi-supervised approaches, as supervision is provided with notably fewer samples, and yet boosting the performance significantly. However, as we see in Table 5.5, there is a huge deviation in performance for FRGC dataset with 2% samples/class due to large class imbalance. Here, the lowest number of samples in a class is only 6, which results in choosing only 2 samples from the class as labeled data.

Labeled Data /Class (%)	FRGC		CMU-PIE		YouTube Faces	
	NMI	ACC (%)	NMI	ACC (%)	NMI	ACC (%)
2	0.884 ± 0.61	83.32 ± 8.80	1.00 ± 0.00	100 ± 0.00	0.938 ± 0.016	90.54 ± 2.92
5	0.951 ± 0.010	94.57 ± 0.97	1.00 ± 0.00	100 ± 0.00	0.989 ± 0.004	98.54 ± 0.61
10	0.971 ± 0.0052	97.01 ± 0.46	1.00 ± 0.00	100 ± 0.00	0.997 ± 0.002	99.70 ± 0.25

Table 5.5 Performance of ClusterNet with different percentage of labeled data on completely unseen face data samples

Labeled Data /Class (%)	USPS		MNIST	
	NMI	ACC (%)	NMI	ACC (%)
1	0.933 ± 0.006	96.33± 0.35	0.939 ± 0.018	98.13 ± 0.24
2	0.959 ± 0.007	98.02 ± 0.44	0.958 ± 0.003	98.38 ± 0.18
5	0.965 ± 0.009	98.43 ± 0.46	0.963 ± 0.001	98.65 ± 0.05
10	0.971 ± 0.007	98.64 ± 0.39	0.97 ± 0.003	98.96 ± 0.12

Table 5.6 Performance of ClusterNet with different percentage of labeled data on completely unseen digits data samples

5.5.5 Ablation Experiments

Effect of Reconstruction Loss on Clustering Performance

To better understand the selection of autoencoder network over an encoding network, we report results in Table 5.7 with and without reconstruction loss in our objective function. The results show that adding the data fidelity term improves the performance both in terms of the average and standard deviation values.

Effect on Image Reconstruction

Additionally, ClusterNet also retains the quality of reconstructed images owing to the reconstruction loss in the objective function. We show some reconstructed images for the unseen test samples of CMU-PIE and FRGC dataset in Figure 5.5. This indicates that the learned

Approach		MNIST	FRGC	CMU-PIE	USPS	YouTube Face
ClusterNet	NMI	0.911 ± 0.039	0.866 ± 0.086	1.0 ± 0.0	0.924 ± 0.008	0.933 ± 0.016
w/o reconstruction loss	ACC	94.94 ± 4.47	81.4 ± 13	100 ± 0.0	96.42 ± 0.691	90.23 ± 2.73
ClusterNet	NMI	0.926 ± 0.018	0.873 ± 0.074	1 ± 0.0	0.923 ± 0.006	0.932 ± 0.016
with reconstruction loss	ACC	96.84 ± 1.2	82.24 ± 10.13	100 ± 0.0	96.51 ± 0.43	90.31 ± 2.69

Table 5.7 Clustering performance with and without reconstruction loss in ClusterNet objective function.

latent space is not degenerate and the decoder can successfully generate the images from the embeddings.

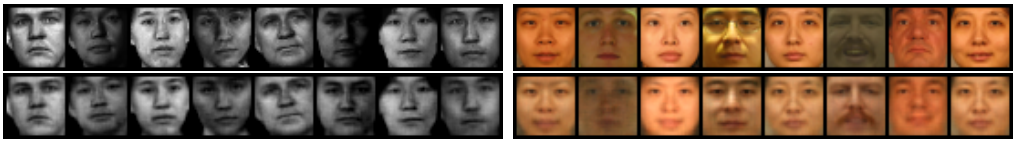


Figure 5.5 CMU-PIE and FRGC original images in top left and top right respectively with corresponding reconstructed images in bottom left and bottom right.

5.6 Conclusion and Future Scope

In this work, we introduced a convolutional autoencoder based semi-supervised clustering approach that works in an end-to-end fashion. The clustering and representation learning is driven by the few labeled samples as well as the unlabeled data and their predicted cluster-membership labels, as they evolve during the training process. The loss comprises a $k-$means style cluster loss and a pairwise KL-divergence term regularized by the autoencoder reconstruction loss. Further, experimental results show that ClusterNet achieves better performance over state of the art unsupervised and semi-supervised approaches using only 2% of labeled data. We achieve stability in training by employing an annealing based strategy to adjust the balancing coefficient.

In this chapter, using empirical analyses, we have demonstrated that with only a small amount of labeled data and a lot of unlabeled data, it is possible to learn representations that

achieve good clustering and classification performance. While the proposed representations do perform well in clustering, its classification accuracy is still lower than the state of the art semi-supervised *classifier* presented in [141]. In order to close this gap, one possible future direction is to use probabilistic predictions of labels and perform clustering in a Bayesian framework.

Chapter Six

Unsupervised Disentangled Representation Learning

6.1 Understanding Disentanglement and Human Perception

A lot of work in computer vision and graphics has focused on developing analytical forward (and inverse) models of image formation, written as combinations of various factors such as object size, type, pose, shape, illumination, reflectance, inter-reflections, shading, etc. In a physical sense, while many of these factors vary independent of each other like pose and object type, these variables often interact in highly non-linear ways, partly due to the intrinsic non-Euclidean nature of the space in which these variables reside for image formation. Common assumptions to simplify the generation model include assuming Lambertian reflectance properties, near-convex object shapes etc. Several results have been developed, which under different assumptions and different combinations of factors show that the data manifold generated is in fact non-linear [53, 181]. The non-linearity in the observed data manifold is due to both the intrinsic non-linearity of certain factors of variation (like pose), and the non-linearity in the forward model itself (due to illumination and shading for instance).

However, human cognition performs an invariant perception of objects, that allows to

identify and discriminate objects irrespective of object variations in terms of pose, size, orientation or background. This indicates that the human sensory system learns representations that encode these attributes disjoint from one another. The idea of disentangled representations (*dis-rep*) is to mimic human understanding of different attributes. Thus, dis-rep learning approaches focus on learning an interpretable representation that highlights and provide control on the different attributes defining the visual data. As opposed to supervised learning task in deep networks that learn representations focusing on a given attribute for the downstream task like classification, *dis-reps* are generic in their construction making them suitable for wider range of applications like few shot learning [109], transfer learning [16] and domain adaptation [169]. Most prominently dis-reps are used (i) to learn certain attribute independent/invariant representations that can improve the performance of various downstream tasks and (ii) to perform image synthesis or attribute transfer by controlling different factors of variation.

6.2 Overview of the Proposed Formulation

In this chapter, motivated by the principles of image formation we propose to learn factorized disentangled representations. Given the fact that an image is generated from a complicated manifold formed by interaction of different manifolds that define different *independent* factors of variations, the factorized aggregated representation is constrained to mimic this independence. To this end, we *explicitly* enforce orthonormality of the attribute vectors of an image as a proxy for independence of the factors. Furthermore, to improve the generalization of the latent space, we employ an orthonormality loss during training along with the explicitly enforced orthonormality of the stacked attribute representation. Additionally, to effectively ensure the orthonormality of the stacked representation, it is explicitly parametrized as a point on the Stiefel manifold. The proposed approach is modular in nature and hence can be easily augmented in existing approaches.

Key Highlights

- *Proposed a framework that imposes an explicit structure in the form of orthogonality constraint on the latent space to act as an inductive bias to improve the disentangling.*
- *Employed optimization on the Stiefel manifold to efficiently ensure the orthonormality of the attribute representations during network training in an end-to-end fashion.*
- *The learned representations corresponding to different subsets of the latent space are inherently separated on orthogonal subspaces.*
- *The proposed approach outperforms existing unsupervised approaches both in terms of disentanglement as well as in image synthesis tasks.*
- *The proposed framework is flexible and can be accommodated in any existing disentangling framework.*

6.3 Existing Literature

Disentangling factors of variation to extract meaningful representations has been an engaging area of interest in the research community and has been explored in various contexts of style-transfer for both text [84] and image domains [52], analogy-making [143] and domain adaptation [167, 134]. Most widely VAEs and GANs have been used to learn these representations in either unsupervised setting [31, 124, 86, 105, 30, 134, 153] or semi/weakly supervised setting [122, 82, 162, 19, 146, 104].

Approaches such as those proposed in Mathieu *et al.* [122] and Szabó *et al.* [162] successfully separate a specified attribute using semi-supervised adversarial training. Ruiz *et al.* [146] also employed adversarial training in weak supervision. On the other hand, approaches

like Jha *et al.* [82] eliminated adversarial training and resorted to cycle consistency in VAE framework to avoid degeneracy. The approach of [104] also used cycle consistency in a semi-supervised way, to disentangle pose and appearance for hand-pose tracking and estimation application.

While these approaches perform well to some extent, disentangling in an unsupervised setting is still a challenge. Several approaches like Press *et al.* [134] followed a domain adaptation approach using an encoder-decoder architecture to disentangle. Li *et al.* [105] utilized unsupervised sequence modeling to disentangle temporally dynamic features from the static ones in audio and video sequences. Another related approach, [153] disentangled shape from appearance using canonical and template-based coordinate systems in an unsupervised manner.

Whether semi-supervised or unsupervised, most of these approaches are often limited to one or two factors of variation. In order to improve image synthesis and attribute transfer, a more elaborate representation is required that can disentangle multiple factors simultaneously. More recently, there is an interest in disentangling multiple factors of variation simultaneously. Approaches like [32] condition the input on a favorable mask of factors to be replicated in the output image. On the other hand, Bouchacourt *et al.* [19], utilized group level supervision, where within a group the observations share a common but unknown value for one of the factors of variation. This in effect relaxes the observations from independent and identically distributed (i.i.d) condition imposed in VAE based models, and also improves generalization to unseen groups during inference.

Unsupervised approaches based on β-VAE [124], introduce a weighted KL-divergence term to ensure a restrained information bottleneck for the learned disentangled latent representations. β-TCVAE [30], a variant of β-VAE, decomposes the marginal KL-divergence term into dimension-wise KL-divergence and a total correlation term that exhibits the extent of disentanglement. Another variant of β-VAE called Factor-VAE [86] supports the idea of the latent code distribution to be factorial by introducing a discriminator that differentiates

a latent code picked from marginal code distribution from one that is a product of them. Chen *et al.* [31] tries to achieve mutual information maximization between latent codes and corresponding observations via an additional information regularization in an unsupervised fashion. Although it works in an unsupervised fashion, lack of encoder-decoder style architecture limits the re-usability of the latent codes of an image. We aim to propose a general approach that works well on any disentangling model, by making use of factorized latent spaces to enhance the quality of disentanglement.

More recently, over the last several years, geometry has been explored in deep learning approaches. Approaches like [151, 154] explored the Riemannian geometry of the latent space of generative models. [9] incorporates a geometry-aware adversarial approach to avoid mode collapse, while [103] adds a geometry-aware regularization in accordance with the manifold hypothesis. [6, 92] consider the curvature and non-linear manifold statistics in latent-space and induce a Riemannian metric to achieve better interpolation and distance functions. [23] proposes various metrics to measure disentanglement and flattening across layers of neural architectures.

6.4 Proposed Latent Space Parametrization

A naive approach to learn representations [74] in deep networks is by using autoencoder framework. An autoencoder, as the name suggests, consists of an encoder network that learns a function $f : \mathbf{x} \rightarrow \mathbf{z}$ to map an input to the representation space and a decoder network $g : \mathbf{z} \rightarrow \mathbf{x}$ that maps the representation back to the input space. By restricting the dimensionality of representation space to be fairly smaller than the input space, it allows one to capture the principal factors of variations in the data. However, it is not ensured that these factors are separated or disentangled to allow systematic data manipulation. On the other hand, disentangled representations facilitate latent space manipulation and closely align with human reasoning [11, 73, 72].

While most approaches resort to deep generative models like VAEs and GANs, there are only few developments using autoencoders. We develop a framework that is flexible to use any existing disentanglement method irrespective of the network choice as a backbone and improves its disentangling ability. We propose to explicitly enforce geometric structure in the latent space.

6.4.1 Enforcing Geometric Structure in the Latent Space

The three factors that contribute to our design choice for disentangled representation learning are as follows:

- **Explicit or Implicit control on the latent space** As pointed by recent developments in the field [115], unsupervised learning of disentangled representation is not possible. This suggests the need for supervision either by interaction in the latent space [166] or by using weak supervision e.g. data grouping [19] to achieve disentanglement.
- **Independent latent variables:** Assuming that the data is generated from independent factors of variation, the disentangled representation should capture different factors in independent dimensions of the latent space. This will allow control on different factors and improve image generation process.
- **Continuity of Space:** The latent representations should be evenly distributed in the latent space to obtain valid latent codes allowing smooth interpolation, resulting in valid images.

Product of Orthogonal Spheres We propose to model the latent space as product of constant curvature spaces, where each space is a hypersphere. This characterization of the latent space is trained with a disentangling objective function to capture different attributes in these hyperspheres.

6.4.2 Motivation for Proposed Parametrization

We would like to present a motivation for the choice of orthogonality in latent-spaces by first principles analysis of typical factors of image-formation. An image can be seen as a complex non-linear interaction between various factors such as lighting, pose, shape, and shading. In many cases, there is a wealth of literature that studies the basic geometric properties of each mode of variation. However, it has been found that the specific constraints are too specific for each factor, and generally not compatible with each other, or with contemporary machine learning models. In this section, we show that a slight relaxation of some of these classical findings will lead to a very natural compact model for latent-variables in the form of a product of orthogonal-spheres. This new model is very simple to implement and enforce as a simple loss-function in deep-learning modules. We start with a few common factors in image-formation, and develop the model.

1. **Lighting variables:** In the illumination cone model, assuming Lambertian reflectance, and convex object-shapes, one can show that the image space is a convex-cone in image space [53]. A relaxation of this model leads to identifying cones as linear-subspaces, which are seen as points on a Grassmannian manifold $\mathcal{G}_{n,k}$ (n = image-size, k = lighting dimensions, typically considered equal to number of linearly independent normals on the object shape) [116]. Under certain conditions of variance on the Grassmannian being low, a distribution of points on the Grassmann induces a distribution on a high-dimensional sphere (see [27]), whose dimension depends on n and k [27].

2. **Pose variables:** 3D pose is frequently represented as an element of the special orthogonal group $SO(3)$. For analytical purposes, it is convenient to think of rotations represented by quaternions [51], which are elements of the 3-sphere S^3 embedded in $\mathbb{R}^4$, with the additional constraint of antipodal equivalence. This makes rotations to be identified as points on a real-projective space $\mathbb{RP}^3$. Real-projective spaces are just a special case of the Grassmannian – in this case, of 1D sub-spaces in $\mathbb{R}^4$. Using similar

result as before [27], from a distribution on quaternions, we can induce a distribution in a higher-dimensional hyperspherical manifold.

3. **Deformation variables:** Deformation of underlying shape, or non-elastic deformation of image-grid (e.g. due to shape change, expression, or photoshop effects), can be modeled by the framework of diffeomorphic maps from $\mathbb{R}^2$ to $\mathbb{R}^2$. Diffeomorphic maps are continuous, smooth, and invertible maps that warp an image grid. Under additional conditions such as corner-point preservation, a square-root form of the diffeomorphic map can be viewed as a point on a infinite-dimensional Hilbert sphere [81]. The infinite-dimensionality is more a mathematical convenience, but in practice, the dimension of the hypersphere is defined by the image-resolution.

Generality: Now, the relevant question is whether hyperspherical relaxation is a good model for factors beyond the above? Consider for instance the factors listed in table 6.4. Most of the factors listed there are mid-level or semantic factors, rather than analytically definable physical factors. However, to a good degree of approximation such semantic factors can be explained as a result of combinations of low-level physical factors. Wavy hair for instance can be approximately described by deformation variables, makeup can be approximated by lighting variables, etc. While our model specifically is motivated by well-studied tractable factors like illumination, pose, and deformation, our conjecture is that the model is quite flexible and applicable beyond these factors.

6.4.3 Loss Function

Product of spheres is a generalization of tori geometry. The specification is incomplete without knowing the individual dimensions of the hyperspheres involved. For analytical and computational tractability, we choose to set the dimensions of the spheres to be equal to each other. This in effect imposes a simple orthonormality constraint on the latent space

blocks.

Definition 1: ***Spherical Space:*** A d dimensional hypersphere is an embedding in $\mathbb{R}^{d+1}$ space and can be defined as follows:

$$\mathbb{S}^d = \{\mathbf{x}^{d+1} : ||\mathbf{x}||_2 = 1\} \tag{6.1}$$

with spherical distance on the space given by $d_{\mathbb{S}}(x, y) = arccos(\langle x, y \rangle)$.

So, k such spheres $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \cdots \mathbf{S}_k$, to characterize k partitioned latent space representation for an input $\mathbf{x}$. So, the latent space representation $\mathbf{z}$ given as the product of spheres is obtained by the Cartesian product *i.e.* $\mathcal{S} = \mathbf{S}_1 \times \mathbf{S}_2 \times \mathbf{S}_3 \cdots \times \mathbf{S}_k$

Now, in order to enforce that these k embeddings capture different factors of variation, we propose to suppress their correlation *i.e.*, $\forall(i, j), i \neq j \langle \mathbf{z}_i, \mathbf{z}_j \rangle = 0$. Thus, we augment the disentanglement objective function with the following regularization term:

$$\mathcal{R}(\mathbf{Z}) = ||\mathbf{Z}^\top \mathbf{Z} - \mathbf{I}||_F^2 \tag{6.2}$$

Here, $\mathbf{Z} = [\mathbf{z}_1 \mathbf{z}_2 \cdots \mathbf{z}_k]$ is a $d \times k$ matrix obtained by stacking together the k embeddings corresponding to k hypershperes of the latent space and $||\cdot||_F$ is the matrix Frobenius norm.

Thus, the modified loss function is given by

$$\mathcal{L}_{disentangle} + \mathcal{R}(\mathbf{Z}) \tag{6.3}$$

Here, $\mathcal{L}_{disentangle}$ is the loss function imposed by any existing disentangled representation learning approach, $\mathcal{R}(\mathbf{Z})$ is the orthonormality regularizer on the latent space and λ is a trade-off parameter to weigh the importance of the added regularization term.

This effectively amounts to a structural constraint to overcome the limitation of unsupervised approaches. It also explicitly fulfils the requirement of independent latent variables for different attributes thus satisfying the first two design goals. The continuity of the latent space is achieved as a consequence of the joint effect of the disentangling loss and the orthogonality structure imposed by the objective function. An illustration of the proposed parametrization and update strategy is shown in the Figure 6.1.

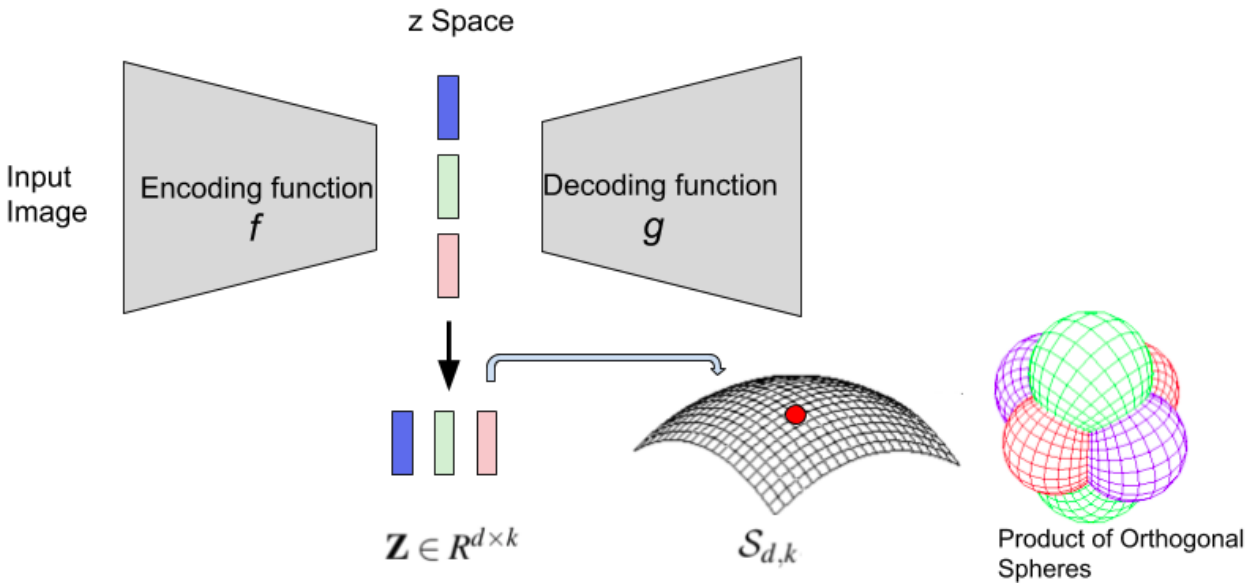


Figure 6.1 Illustration of proposed Product of Orthogonal Spheres as a latent space model

6.4.4 Parameter Updates and Explicit Orthonormality Constraint

The proposed regularizer in the loss function is a relaxation of strict orthogonality, also known as the soft orthonormality constraint. The loss function in Eq. (6.3) can be optimized with standard back-propagation algorithm to learn the network parameters. However, we find that optimizing the proposed loss function leads to slow convergence due to additional constraint on the latent space. Therefore, we explicitly enforce the orthogonality constraint that ensures that the latent space representation is strictly orthonormal that improves the convergence.

Explicit Orthonormality Projection

We utilize the geometry of Stiefel manifold to effectively ensure the orthonormality constraint. Therefore, our update strategy consists of two parts. Firstly, back-propagation works towards ensuring that the soft constraint is satisfied. And secondly, the hard constraint imposed by constraining the representation as a point on the Stiefel manifold to push the convergence along.

Definition 2: ***Stiefel manifold*** $\mathcal{S}_{d,k}$, with $d > k$ is defined as space of $d \times k$ matrices, that have orthonormal columns and is equipped with Frobenius inner product i.e.

$$\mathcal{S}_{d,k} = \{\mathbf{U} \in \mathbb{R}^{d \times k} : \mathbf{U}^\top \mathbf{U} = \mathbf{I}\} \tag{6.4}$$

Minimizing an objective function defined on the Stiefel manifold requires an update strategy that ensures feasibility of the solution with every gradient step. We use Cayley transform, a widely used approach for optimization on the Stiefel manifold because of the closed form update. We used this transform to obtain feasible iterates in the direction of descent.

Definition 3 : ***Cayley Transform*** Given a problem $\min_{\mathbf{U} \in \mathcal{S}_{d,k}} \mathcal{F}(\mathbf{U})$ computes a parametric curve on the Stiefel manifold to obtain a feasible iterate in the direction of iterate by a closed form solution using the gradient $\mathbf{G}$ and is given by

$$\mathbf{U}_{t-1} = \big(\mathbf{I} + \frac{\tau}{2}\mathbf{A}\big)^{-1}(\mathbf{I} - \frac{\tau}{2}\mathbf{A})\mathbf{U}_t, \tag{6.5}$$

where, $\mathbf{A} = \mathbf{G}\mathbf{U}^\top - \mathbf{U}\mathbf{G}^\top$ is a skew symmetric matrix and τ is the step size. More details on the Stiefel manifold and the optimization strategy can be found in Section 2.1.1 and Section 2.2.

So, correspondingly the updates for $\mathbf{Z}$ as an element on the Stiefel Manifold $\mathcal{S}_{d \times k}$ for optimizing the loss function given in (6.3) as follows:

$$\mathbf{Z} = \big(\mathbf{I} + \frac{\tau}{2}\mathbf{A}\big)^{-1}(\mathbf{I} - \frac{\tau}{2}\mathbf{A})\mathbf{Z}, \tag{6.6}$$

where, $\mathbf{A} = \mathbf{J}\mathbf{Z}^\top - \mathbf{Z}\mathbf{J}^\top$, and $\mathbf{J}$ is the network gradient, thus transforming the latent space representation before feeding it to generator for further processing.

It should be noted that the explicit orthonormality constraint is only enforced during the training time. We observe that the trained network achieves close to orthonormal representations at evaluation time as well.

We empirically demonstrate the advantage of the proposed product embeddings parametrization of the latent space. We show that it not only improves the quality of the disentangled representations but simultaneously improves the fairness of the representations as well. We also show improvement in the reconstruction quality across several synthetic and real datasets.

6.5 Experimental Setup and Results

We evaluate our approach on a number of datasets to validate its ability to learn better disentangled representations owing to the imposed orthonormality constraint in the latent space.

Datatset	Attributes
MNIST (28×28)	Slant Angle, Stroke Width, **Identity**
2D Sprites (64×64)	**Gender** , **Hair Type**, **Body Type**, **Armour Type**, **Greaves Type**, Pose, Weapon
Cars3d (128×128)	Azimuth, Elevation, **Identity**
Shapes3d (64×64)	**Floor Hue, Wall Hue, Object Hue, Scale, Shape, Orientation**

Table 6.1 Summaries of datasets with details of image size and the attributes. The labeled attributes are denoted in bold.

6.5.1 Datasets

We perform experiments on several datasets of various difficulties with multiple attributes. The details of different datasets along with the available label information are given below and summarized in Table 6.1 for quick reference. The table provides details about image sizes and different attributes present in the dataset. For ease of readability, we also highlight the attributes that have available annotations. These annotations are used for evaluating several aspects of disentangled representations.

2D Sprites [143] is a synthetic dataset with $143,040$ images of animated game characters

Network	Dataset and Architecture
	MNIST
Enc.	Conv(64,3,2) - Conv(128,3,2) - Conv(256,3,2) - FC(kd)
Dec.	FC(1024)- Dconv(128,4,2) - Dconv(64,3,2) - Dconv(1,4,2)
Disc	Conv(64,3,2) - Conv(138,3,2) - Conv(256,3,2) - Conv(512,1,nill) - FC(1)
	CelebA, Sprites2d
Enc.	Conv(64,3,2) - Conv(128,3,2) - Conv(256,3,2) - Conv(512,3,2) - Conv(512,3,2) - FC(kd)
Dec.	Dconv(512,4, nill) - Dconv(256,4,2) - Dconv(128,4,2) - Dconv(64,4,2) - Dconv(3,2,nill)
Disc	Conv(64,3,2)- Conv(128,3,2) - Conv(256, 3,4) - Conv(512,1, nill)- FC (2)
	Cars 3D
Enc.	Conv(64,3,2) - Conv(128,3,2) - Conv(256,3,2) - Conv(512,3,2) - Conv(512,3,2) - Conv(512,3, nill) - FC (kd)
Dec.	Dconv(512,3,2) - Dconv(512,3,2) - Dconv(256,3,2) - Dconv(128,3,2) - Dconv(64,3,2) - Dconv(4,4,2)
Disc.	Conv(64,3,2) - Conv(128,3,2) - Conv(256,3,2) - Conv(512,3,2) - Conv(512,3,2) - FC(1)
	Shapes 3D
Enc.	Conv(64,3,2)- Conv(128,3,2) - Conv(256,3,2) - Conv(256,3,2) - Conv(256,3, nill) - FC(kd)
Dec.	Dconv(256,3,2) - Dconv(256,3,2) - Dconv(128,3,2) - Dconv(64,3,2) - Dconv(3,4,2)
Disc.	Conv(64,3,2) - Conv(128,3,2) - Conv(256,3,2) - Conv(512,3,2) - Conv(512,3,2) - FC(1)

Table 6.2 Details of network architectures for Enc. (Encoder), Dec. (Decoder) and (Dis.) Discriminator used for different datasets.

(sprites) with 480 distinct characters. The training set consists of 320 characters while validation and test sets contain 80 identities each. Annotations for distinct factors of variation such as gender, hair type, body type, armour type and greaves type are provided.

MNIST [100] comprises of 28×28 grayscale images of hand written digits with 10 different classes. The training set consists of 60K images along with a test set of 10K images. A few tangible factors of variation that could be easily perceived from the data are stroke-width, slant angle, identity etc.

CelebA [114] is a celebrity face dataset with 202,599 images each annotated with 40 partitionary attributes like eyeglasses, wearing hat, bangs, wavy hair, mustache, smile, oval face etc. The number of identities are 10,177. We trained our model using a train-test split ratio of 4:1.

Shapes3d is a simulated dataset that consists of 48000 images of 64×64 color images of 3d shapes. The images are procedurally generated from 6 ground truth independent latent factors. These factors are floor colour, wall colour, object colour, scale, shape and orientation. The dataset has label information for all the six mentioned attributes.

Cars3d [48] consists of 128×128 color images of 3d renderings of car models. The dataset has three labeled attributes: azimuth, elevation and model type that have 4, 24 and 183 categories respectively.

6.5.2 Network Details and Hyper-Parameters

To fairly evaluate the different approaches and avoid induced bias, we use the same network architecture, hyper-paramters, optimizer and batch size. For 2D Sprites and CelebA dataset, the latent space is partitioned in 8 partitions with 64 dimensions each. For MNIST, due to fewer variations in the data, the latent space is partitioned into 3 each with dimension 8. For MNIST, the choice is due to the fact that it consists of 3 prime attributes: class identity, stroke width and slant angle. The other two datasets have considerably more factors of variation and therefore, we aspire to capture the ones that are the most obviously perceptible. For shapes3d, the latent space is partitioned into 6 to capture the ground truth factors. Additionally, the weights of individual loss terms used in our implementation is similar to [78]. We use Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and initial learning rate adapted to each dataset to ensure convergence. The step size (τ) for Cayley transformation update equation is chosen as 0.1 for all datasets. The details of the network architectures is given in Table 6.2 for different datasets.

6.5.3 Comparison with state-of-the art approaches

Our framework proposes a paramterization of the latent space that can be augmented with any disentangled representation approach. To evaluate this flexibility of our framework, we

Method	Gender	Skin	Vest	Hair	Arm	Leg	Avg.
MIX	66.5	77.2	90.0	56.2	63.1	89.4	73.7
+ **PrOSe**	70.1	75.5	88.5	63.2	72.1	94.4	7**7.3**
β-VAE	68.4	77.6	86.8	60.2	60.8	92.2	74.3
+ **PrOSe**	73.8	75.6	93.2	63.7	59.6	94.7	**76.8**
Factor-VAE	71.2	75.5	88.2	59.8	61.7	91.8	74.7
+ **PrOSe**	73.8	75.0	93.8	64.1	61.8	93.7	**77.0**

Table 6.3 mAP values for different attributes for 2D Sprites with various approaches with and without PrOSe

used β-VAE, Factor-VAE, as well as an autoencoder based approach [78] that we refer as MIX in our results.

6.5.4 Disentanglement Evaluation

We perform a series of experiments to qualitatively and quantitatively evaluate the disentangled representations learned with PrOSe framework.

Classification Performance

A disentangled representation captures different attributes in the different parts. In order to evaluate how well a factor is captured in the identified part of the latent space, we evaluate the classification performance with respect to different attributes. We report Mean Average Precision (mAP) values to quantify the classification, with higher mAP indicating better performance. The effect of using PrOSe framework with existing approaches results in improved performance across several datasets as given in Table 6.3, 6.4 and 6.5 for 2d Sprites, CelebA and Shapes3d datasets respectively. Here MIX, β-VAE and Factor-VAE denote the original method and + PrOSe denotes their PrOSe framework counterpart. We report mAP for each of the labeled attribute in the datasets as well the average mAP across

Attribute	MIX	+**PrOSe**	β-VAE	+**PrOSe**	Factor-VAE	+ **PrOSe**
Eyebrows	79.4	79.5	77.2	79.2	76.2	78.8
Attractive	72.6	80.4	69.8	76.7	76.7	72.6
Bangs	91.7	90.2	76.3	74.5	87.2	90.4
Black Hair	71.9	75.6	89.2	90.5	89.6	86.6
Blonde Hair	87.2	92.0	91.0	92.4	78.1	84.8
Makeup	76.5	78.0	72.1	72.4	70.8	76.7
Male	86.2	83.1	83.8	86.2	84.8	86.5
Mouth	72.0	80.6	72.2	72.4	80.2	78.5
No Beard	86.3	89.6	82.0	78.2	80.5	82.4
Wavy Hair	65.7	71.9	84.8	84.0	75.0	77.8
Hat	95.2	94.8	85.4	86.2	82.0	85.2
Lipstick	79.8	80.5	68.2	74.8	72.2	70.0
Average	80.3	**83.0**	79.3	**80.6**	79.4	**80.8**

Table 6.4 mAP values for different attributes for CelebA face dataset with various approaches with and without PrOSe parameterization.

all the attributes. The results highlight the benefits of employing ProSe framework over existing state-of-the-art methods.

Orthogonality of the Attribute Subspaces

We have imposed the orthonormality constraint for every image such that the representation in each of the subsets of the latent space is orthonormal to the other, improving the disentangled representations. This orthonormality allows us to capture different attributes that vary independent of each other. We show that this condition extends beyond image level and also results in different subspaces for different attributes and are orthogonal to each other. We compute the subspace angle between subspaces corresponding to every attribute

Attribute	MIX	+**PrOSe**	β-VAE	+**PrOSe**	Factor-VAE	+ **PrOSe**
Floor Hue	88.2	90.4	84.4	82.2	81.8	84.0
Wall Hue	88.4	91.0	88.4	90.0	89.4	90.2
Object Hue	89.6	92.2	86.7	85.9	84.5	82.6
Scale	81.2	78.4	75.8	80.1	82.5	86.5
Shape	82.3	86.5	88.0	89.7	86.5	88.7
Orientation	70.2	69.6	68.1	71.2	68.8	70.0
Average	83.3	**84.7**	81.9	**83.2**	82.2	**83.7**

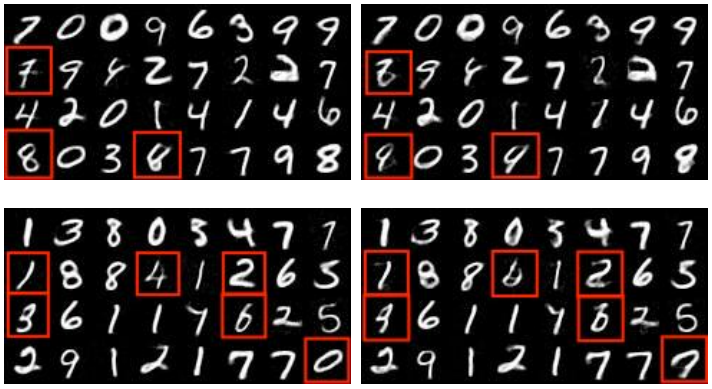
Table 6.5 mAP values for different attributes for Shapes3d dataset with various approaches with and without PrOSe parametrization.

Datasets	Number of Partitions	MIX	MIX + **PrOSe**
MNIST	3	83.42± 7.27	90.38 ±0.05
Sprites2d	8	85.97 ±29.22	90±0.0
Shapes3d	6	76.67±31.21	90.0±0.0
Cars3d	3	26.86±3.86	89.80± 0.72

Table 6.6 Quantitative evaluation of disentanglement by analyzing the relation between different subspaces corresponding to the partitions of the latent representation. We report the average angle between different subspaces corresponding to different subsets.

Dataset	PrOSe Framework	Only ortho-normality loss	Only Projection on Stiefel Manifold
Sprites2D	77.3	75.1	71.7
CelebA	83.0	80.9	76.8
Shapes3d	84.7	81.6	79.8

Table 6.7 Effect on mAP with different ways of imposing orthonormality constraint.



(a) MNIST

(b) Car3D

Figure 6.2 Results of predicting an attribute using representations of remaining $k-1$ subsets with MIX (top row) and MIX + PrOSe (bottom row) for each of the dataset. In both cases, (Left) shows the true class and (Right) shows the predicted class. Marking on the true class is shown for visual correspondence. The marked red boxes are the mis-classified images.

pair. For example, in case of MNIST, where latent space is partitioned into three parts, we compute the angle between the 4 subspace pairs. The average across all pairs is reported for each of the datasets in Table 6.6. We see that for all the datasets, this quantity is very close to 90^o with ProSe framework. This is also indicative of the fact the attribute subspaces are disentangled, leading to improved control over image synthesis and generation.

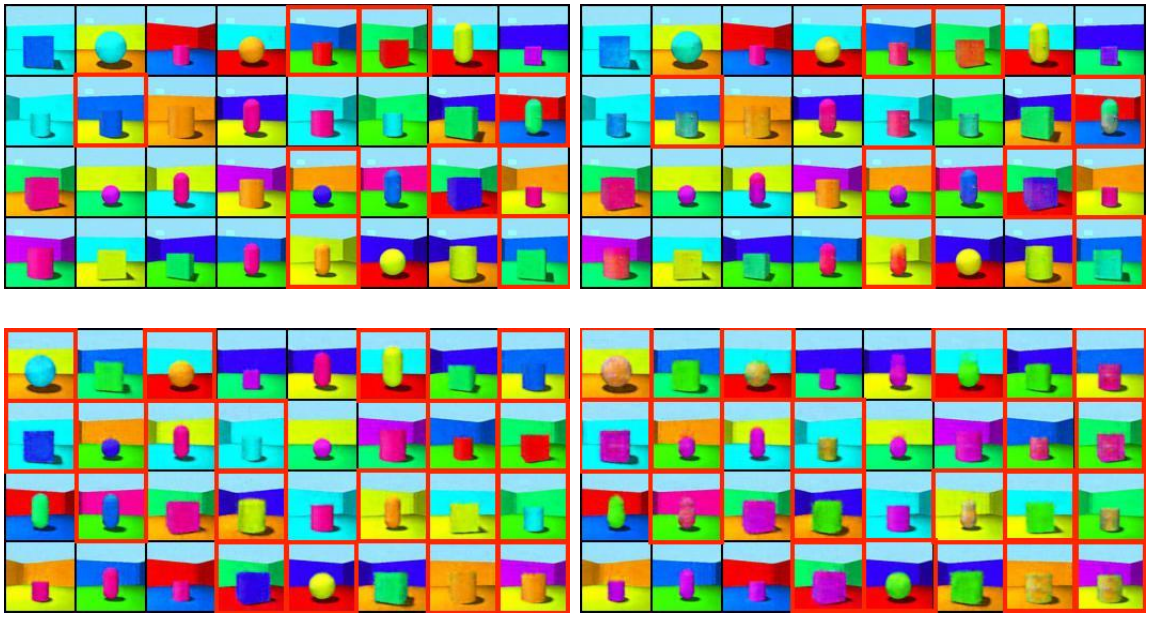


Figure 6.3 Shapes3D: Results of predicting an attribute using representations of remaining $k-1$ subsets with MIX (top row) and MIX + PrOSe (bottom row). In both the cases, (Left) shows the true class and (Right) shows the predicted class. Marking on the true class is shown for visual correspondence. The marked red boxes are the mis-classified images.

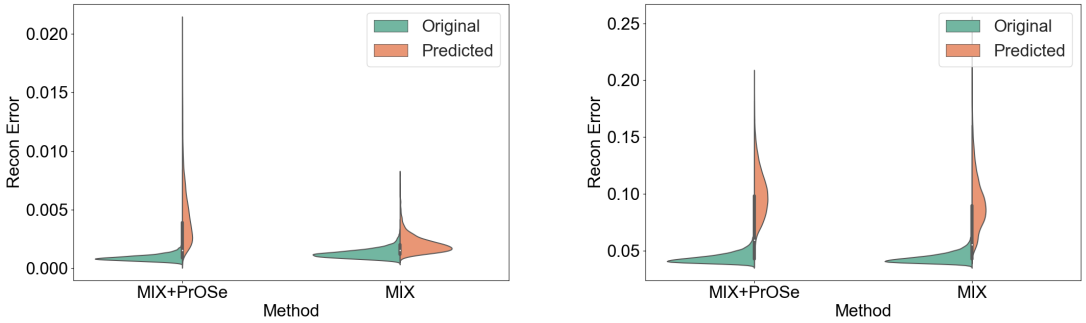


Figure 6.4 Distribution of reconstructed error with original output and with the regressor output for Sprites2D (Left) and MNIST (Right) datasets.

Prediction of Attribute

A disentangled representation ensures that the information of a particular attribute is only restricted to one subset of the space and there is no information leakage of the same to rest of the latent space. Therefore, to validate the quality of disentanglement with PrOSe, we trained a model using random set of $k-1$ attributes (k being the total number of

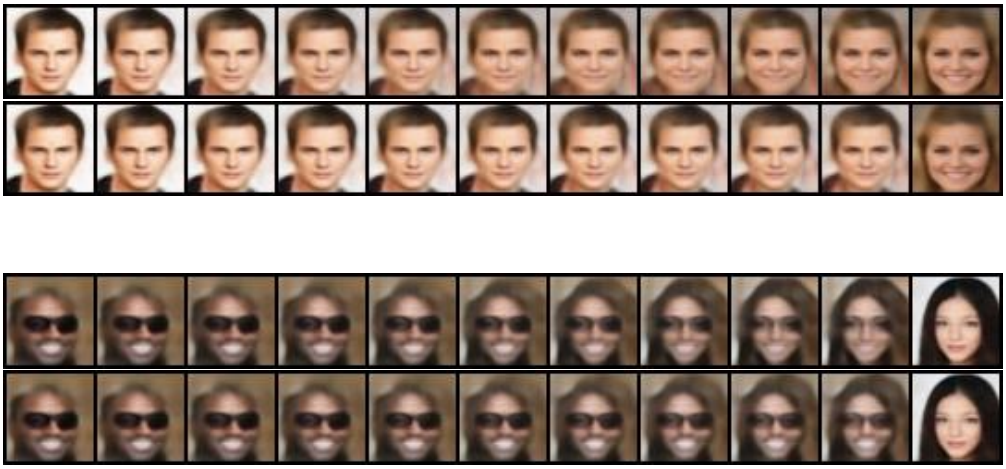


Figure 6.5 CelebA Dataset:(Top Pair) Interpolation across disentangled gender attribute and hair attribute (Bottom Pair) for CelebA dataset with MIX (top) and MIX + PrOSe (bottom). PrOSe achieves a well separated attribute spaces, evidenced by smoother and more meaningful interpolation without altering face shape, expressions etc.

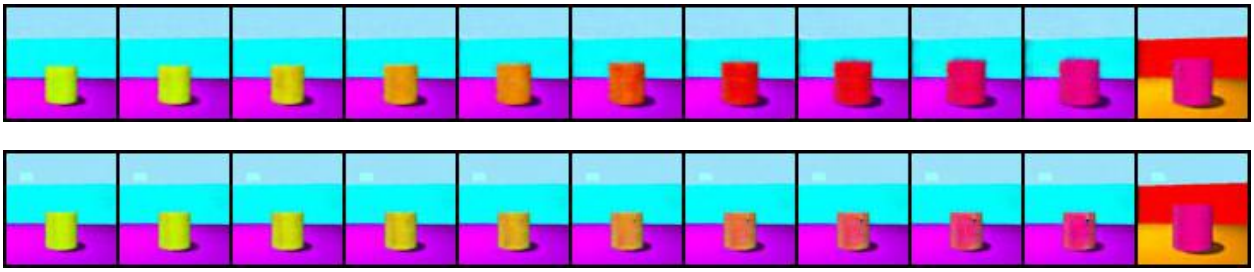
partitions in the latent space) to predict the remaining attribute. In an ideal scenario where attributes have been disentangled, the prediction model must fail, since the left-out attributes information is not captured by the other partitions. Thus, a higher number of mis-classifications signifies lesser information leakage across different partitions of the latent space that are dedicated to different attributes.

However, real world datasets often do not have attribute label information, hence reporting the mis-classifications is not feasible. Therefore, we empirically evaluated the effect on the generated image. Given that the prediction model has failed, the reconstructed image will deviate from the original image. We report the average reconstruction error on 10,000 samples of CelebA dataset. The results in Figure 6.4 show a comparison of reconstruction error with and without prediction model for several datasets.

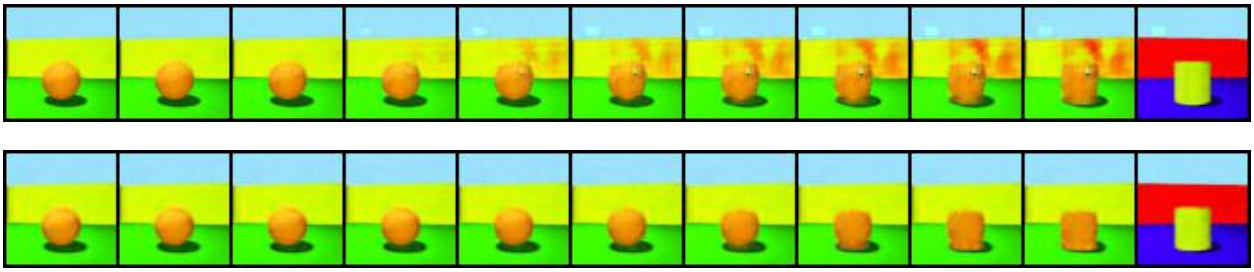
6.5.5 Applications of Disentangling

Image Manipulation and Interpolation

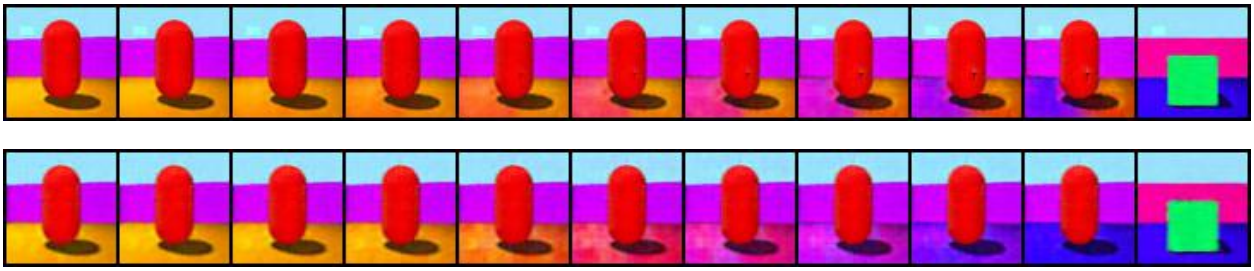
Interpolation is a qualitative measure to assess the purity of attribute captured in a partition. Linearly interpolating between a pair of images along one partition of one image in the



(a) Object Color



(b) Object Shape



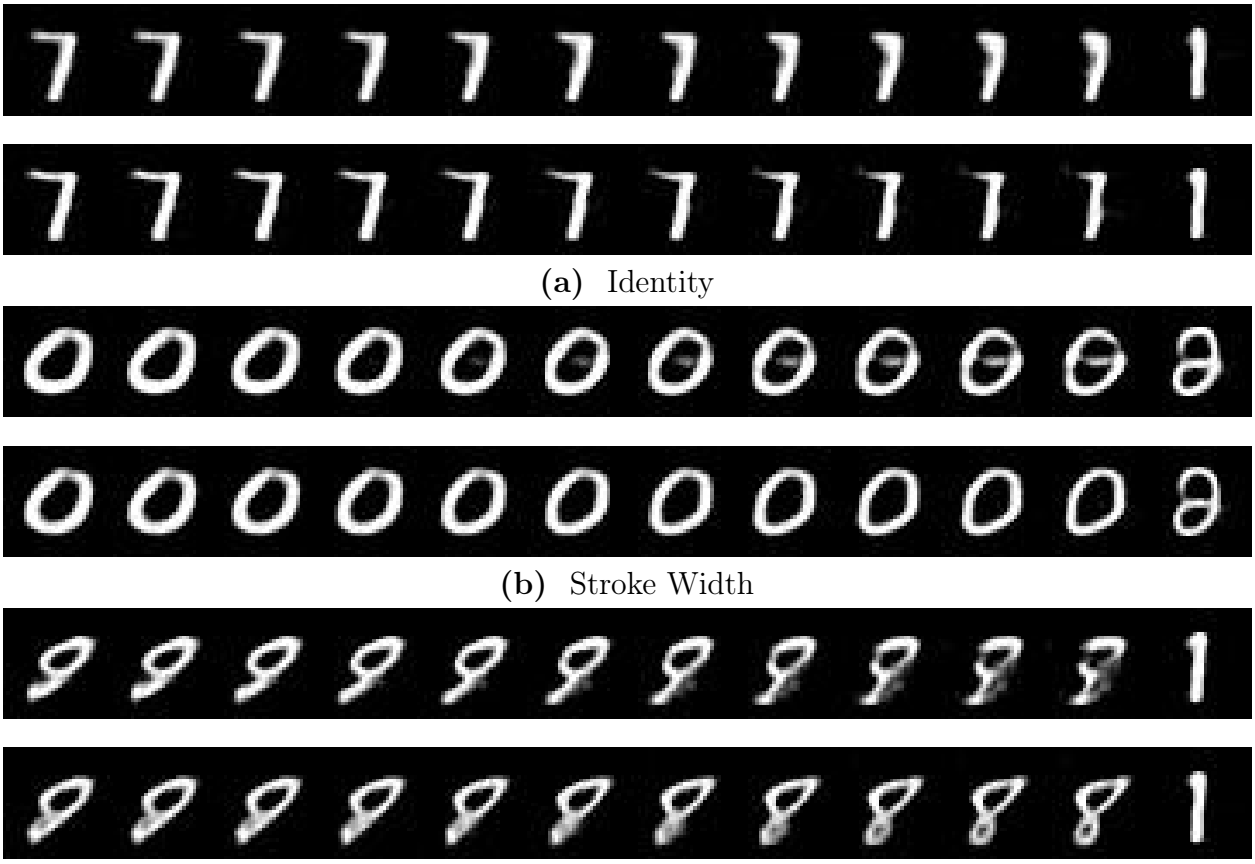
(c) Floor Color

Figure 6.6 Interpolation across a disentangled attribute, while others are fixed for Shapes3d dataset with MIX (top) and with MIX + PrOSe (bottom). The leftmost image is the starting point, whose all factors are fixed except for one specified as a, b or c that is interpolated in the direction of the target image in the rightmost image.

direction of other, while fixing others, demonstrates the purity of given partition. Figures 6.5 and 6.6 show that PrOSe achieves a smoother transition in the attribute of interest while others are unaltered. For example, in CelebA, interpolating in gender attribute space, also affects smile and even face shape, in the case of MIX [78].

Attribute Transfer and Image Synthesis

Figures 6.9 and 6.10 show results for attribute transfer to visually analyze the quality of disentanglement. The first row and the first column in each grid are randomly chosen samples



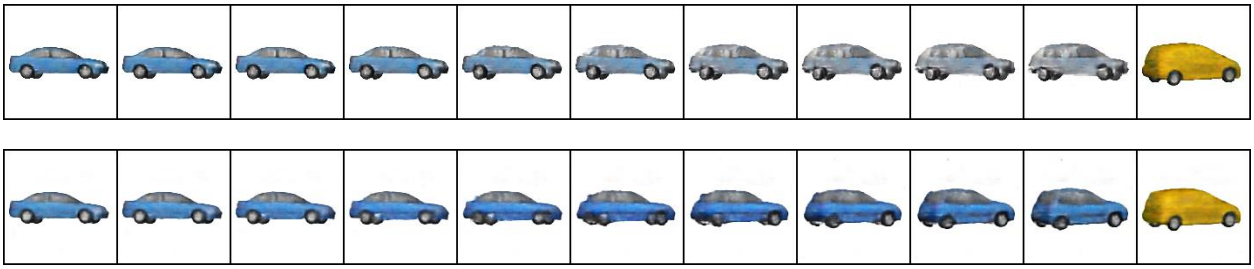
(a) Identity

(b) Stroke Width

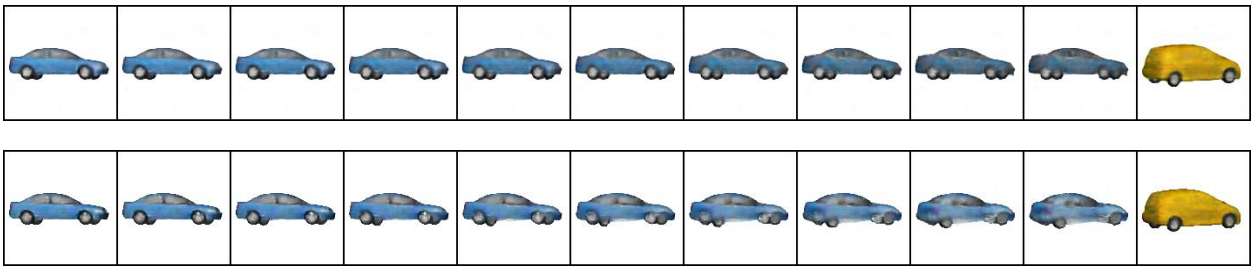
(c) Slant

Figure 6.7 Interpolation across a disentangled attribute, while others are fixed for MNIST dataset with MIX (top) and with MIX + PrOSe (bottom). The leftmost image is the starting point, whose all factors are fixed except for one specified as a, b or c that is interpolated in the direction of the target image in the rightmost image.

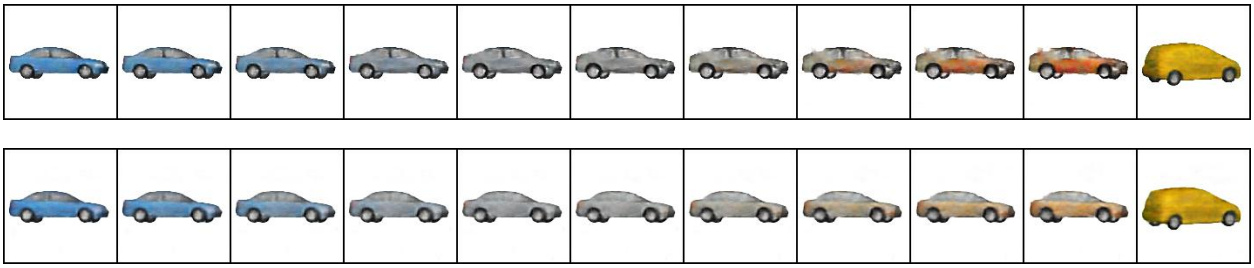
from the test-set. All other images are formed by picking one subset representation from the corresponding column lead while the remaining subsets representations are used from the corresponding row lead. This allows us to quantify how well an attribute is captured in a single partition. For example, in 2D Sprites, more than one factor changes simultaneously with MIX [78] *i.e.* along with complexion/hair colour, a variation in pose is seen, similarly, leg colour also varies along with armour colour. Similar trends are observed in other datasets as well where either multiple factors change or the specified attribute is not transferred.



(a) Car Model



(b) Orientation

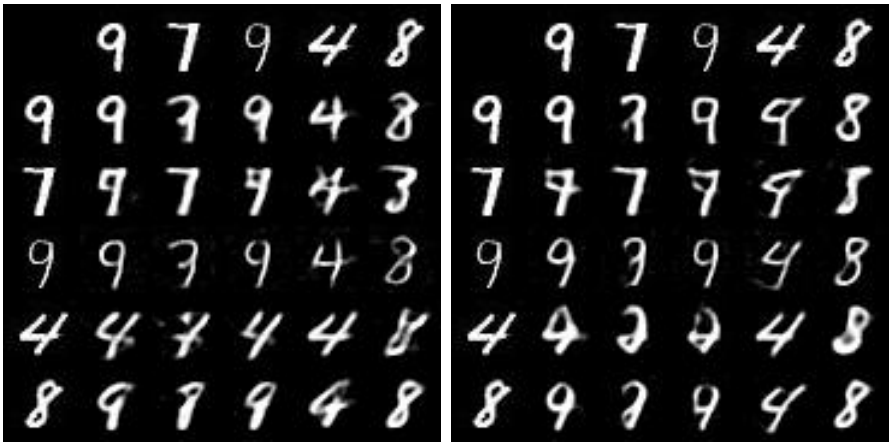


(c) Color

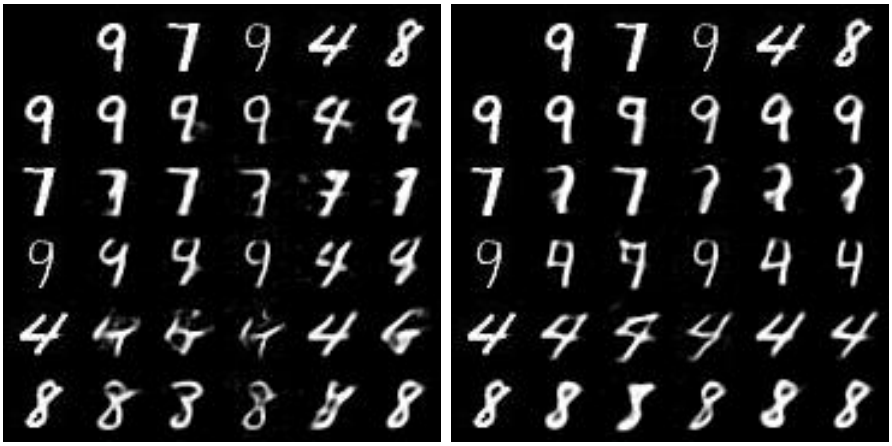
Figure 6.8 Interpolation across a disentangled attribute, while others are fixed for Cars3d dataset with MIX (top) and with MIX+ PrOSe (bottom). The leftmost image is the starting point, whose all factors are fixed except for one specified as a, b or c that is interpolated in the direction of the target image in the rightmost image.

6.5.6 Ablation Study

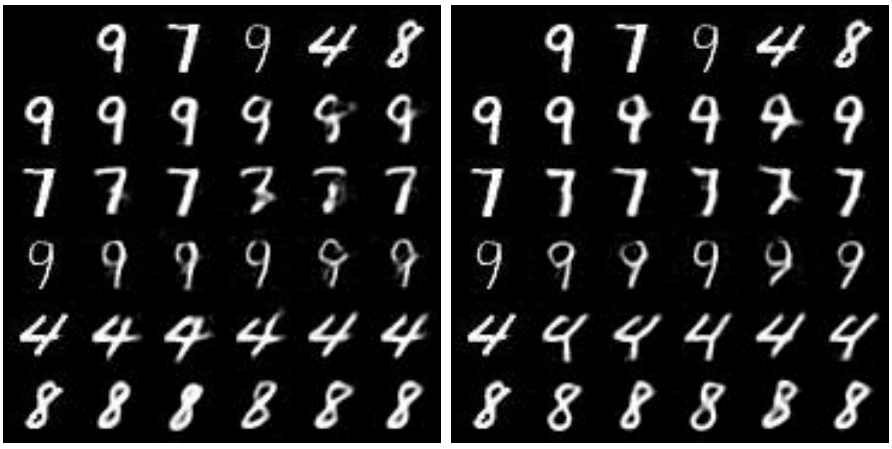
Explicit Stiefel Manifold Projection: The loss function given by Equation 6.3 has a Frobenius norm term that penalizes the deviation of orthonormal structure on the stacked attribute code vectors. In order to ensure orthonormality while training, the stacked attribute code matrix is also projected onto the Stiefel manifold. We observed that the training converges faster with this additional projection step, without any significant change in the reconstruction quality. We compute the mAP values in both the cases and report the results in Table 6.7. The results indicate that the projection step imposes a stronger prior on the



(a) Identity

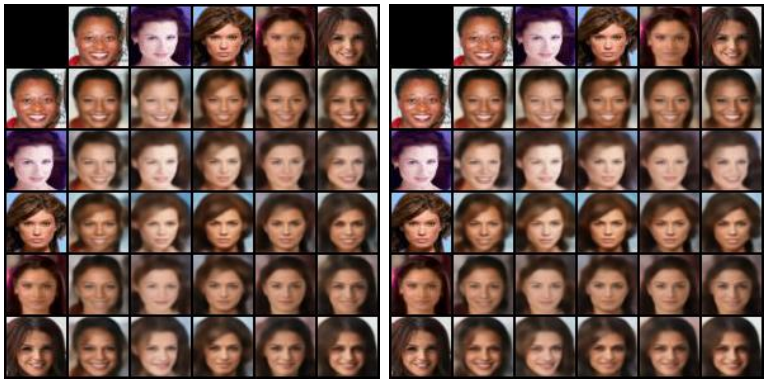


(b) Stroke Width

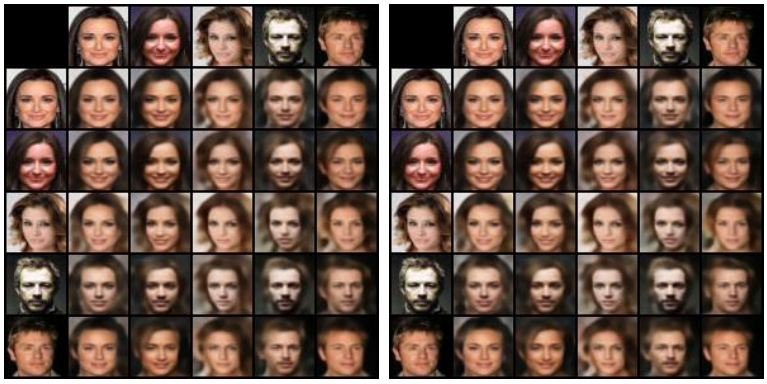


(c) Slant Angle

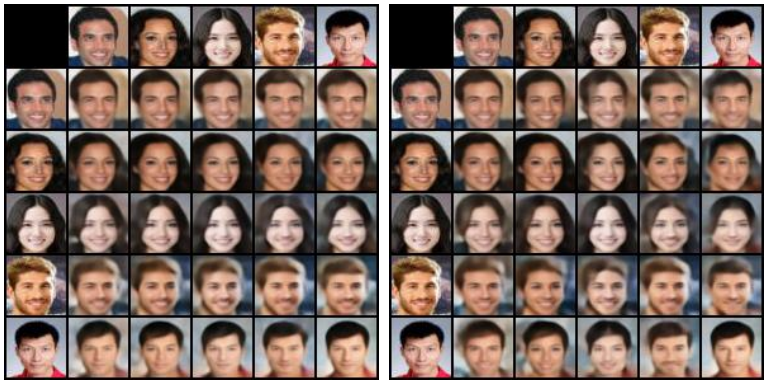
Figure 6.9 MNIST: A visualization grid of image synthesis using attribute transfer. In each grid of the subfigures, the top row and leftmost column images come from the test set. The other images are generated using code vector corresponding to one of the attributes from the image in the top row, while all the remaining attributes are taken from the leftmost column image. Results with MIX (left in each pair of the subfigures) and MIX + PrOSe (right in each pair) are shown.



(a) Complexion



(b) Hair



(c) Moustache

Figure 6.10 CelebA: A visualization grid of image synthesis using attribute transfer. In each grid of the subfigures, the top row and leftmost column images come from the test set. The other images are generated using code vector corresponding to one of the attributes from the image in the top row, while all the remaining attributes are taken from the leftmost column image. Results with MIX(left in each pair of the subfigures) and MIX + PrOSe (right in each pair) are shown.

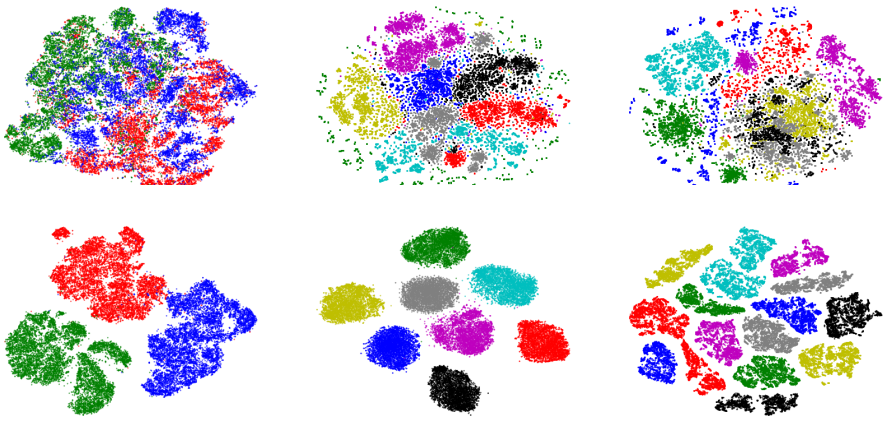


Figure 6.11 t-SNE plots for MNIST (Column 1), 2D Sprites (Column 2) and CelebA face (Column 3) datasets with MIX (top) and MIX + PrOSe (bottom). The different colors denote different attribute spaces. Clearer separation of attributes is seen in the case of PrOSe.

model, improving the results.

Visual Comparison of Latent Space: We show the latent space representations learned from different models in Figure 6.11. The well separated attribute/subset with PrOSe ensures smoother and meaningful interpolations in a given attribute space without affecting other attributes.

6.6 Conclusion and Future Scope

In this work, we used the product of orthogonal spheres parametrization for improving disentangled representations. The parametrization amounts to imposing an orthonormality constraint on the partitioned latent representation of a data sample. The proposed framework added a simple loss term to the disentangling loss, making it easy to incorporate in existing approaches. Directions for future work include investigating broadening the model to different-sized latent blocks, and the impact of the model on other tasks such as recognition, multi-task learning, and generalization to other unseen factors. We are hopeful that such geometric constraints and geometry aware directions will be taken up in the research

community to improve beyond standard deep learning setting.

Chapter Seven

Low Dimensional Representation for Adversarial Defense Strategy

Adversarial attacks are small imperceptible perturbations carefully crafted to modify an image that can mislead the classification ability of state-of-the-art classifiers. An example from ImageNet dataset is shown in Figure 7.1 that is perturbed with adversarial noise, gets mis-classified by GoogLeNet. Robustness of DNNs to such attacks [55, 125, 26, 164] rapidly became an active area of research due to the increasing adoption rates of deep learning based systems in practical applications often with high reliability and security requirements. Since these applications range from autonomous driving [180] to medical evaluations [10], the robustness of these models to adversarial perturbations is a crucial aspect for their reliability.

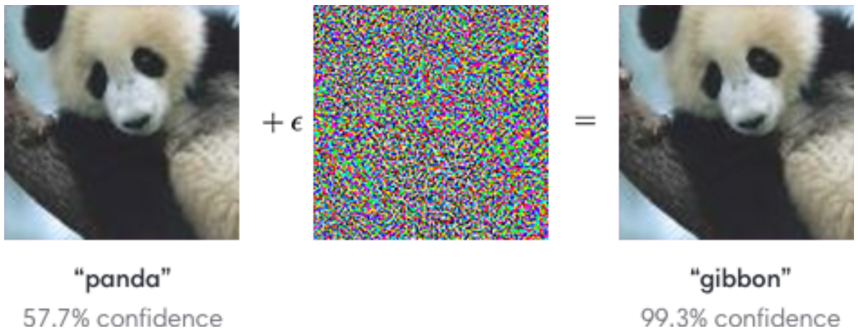


Figure 7.1 An illustration of an adversarial example generated by adding a small imperceptible vector that is obtained by multiplying a small value $\epsilon = 0.007$ with the sign of the gradient of the loss function with the input.

7.1 Limitations in Existing Defense Strategies

- **Requirement of Huge Compute:**

  In order to provide security against adversarial attacks, various approaches like [95, 120, 55, 177, 159, 165, 128, 135] have focused on improving the model's robustness by modifying either the network, the loss function and/or the training strategy. While these approaches have been successful to a great extent, many of them have either model or attack dependencies. Moreover, defenses that rely on adversarial training, i.e, the use of adversarially perturbed samples at train time, typically incur significant computational costs. For example, the recently proposed feature denoising approach [177] required synchronized training on *128 NVIDIA V100 GPUs* for 52 hours to train a baseline ResNet-152 model on ImageNet.

- **Vulnerable to Strong Attacker:**

  Several approaches for adversarial defense employ simple pre-processing or post processing strategies [63, 133, 148, 113] that can be augmented with the deployed models directly at inference time. The success of these transforms is due to *gradient obfuscation* that results in incorrect or undefined gradients, limiting the impact of the typical gradient based white-box attacker. However, work by Athalye *et al* [7] overcame this shortcoming by computing gradient approximations in such scenarios, drastically dropping the performance of many defense strategies, and often completely defeating them (0 % classification accuracy over adversarial samples).

7.2 Which Factors Contribute to a Successful Defense?

To achieve our goal of devising an input transformation based defense, from the discussion above, we make two crucial observations that contribute in undermining the white-box adversary's ability to generate an attack: First is the compounded randomness in the trans-

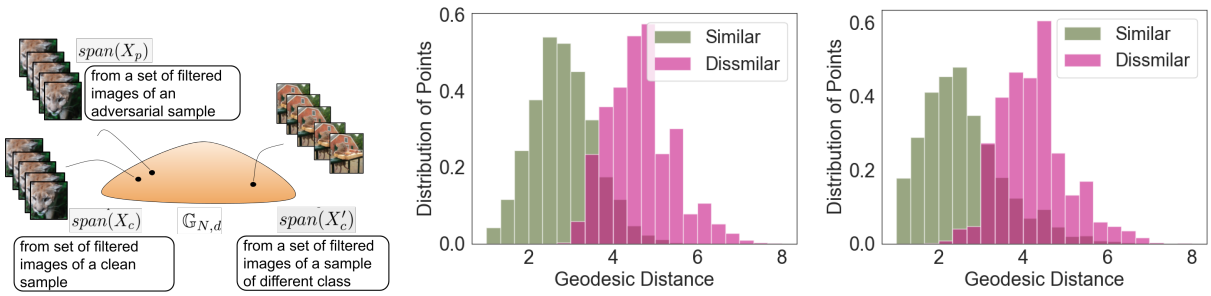


Figure 7.2 Left: Representation of subspaces as points on the Grassmannian manifold. The subspace corresponding to the perturbed sample X_p lies close to the subspace of its clean sample X_c counterpart. The distance between these two subspaces is shown to be upper bounded as given by Eq. (7.8). Centre and Right: The histograms show that subspaces of a pair of images of the same class are closer than subspaces of an image pair formed from different classes. Given an adversarial sample, the plot highlights that the geodesic distance between clean sample subspace $\mathcal{X}_c^l$ and its corresponding adversarially perturbed sample $\mathcal{X}_p^l$, is such that $d(\mathcal{X}_c^l, \mathcal{X}_p^l) < d(\mathcal{X}_c^{l'}, \mathcal{X}_p^l)$. Here l and l' represent two different classes. The plot is shown for 8000 similar $(\mathcal{X}_c^l, \mathcal{X}_p^l)$ and 8000 dissimilar pairs $(\mathcal{X}_c^{l'}, \mathcal{X}_p^l)$. The normalized histogram for these pairs is shown for two models on ImageNet dataset : InceptionV3 (Center) and ResNet50 (Right).

formations applied to the input image. Second is the access to a set of similar images (identified as neighbors in the web-scaled database in case of Dubey *et al* [44]) to have an averaging or smoothening effect over predictions, leading to more accurate predictions for an adversarial sample. Contrary to [138, 44], we take an approach that simply relies only on a given sample and leverages benefits from both the compounded random transformations as well as smoothing. However, the random transforms and smoothing are performed in a *principled manner* that reduces the impact of adversarial noise, without significant changes in the image. Thus, making our approach a model-agnostic, inference-time defense that does not rely on additional data or training to achieve the desired goal of adversarial security.

Key Highlights

- *The proposed input transformation based defense GraCIAS achieves state-of-the-art results on ImageNet dataset for ResNet50, InceptionV3, VGG16 and MobileNet models under different attacker strength in white box attack scenario.*
- *As opposed to state-of-the-art randomized input transformation approaches, GraCIAS benefits not only from its intrinsic random parameterization, but also from the theoretical motivation that suggests retention of task-relevant information and suppression of adversarial noise.*
- *Due to its simplicity and computational efficiency, the proposed defense can be integrated with existing weak defenses like JPEG compression to create stronger defenses, as shown in our experiments.*

7.3 Overview of the Contribution

Our proposed approach, **Gra**ssmannian of **C**orrupted **I**mages for **A**dversarial **S**ecurity (GraCIAS) applies a random number of randomized filtering operations to the input test image. These filtered images provide a basis for a lower dimensional subspace, which is used to smoothen the input image. Due to a principled structure of generated image corruptions used for defining the subspace, it permits projection and reconstruction of the input image without substantial loss of information. Furthermore, we can interpret these subspaces as points on the Grassmann manifold, enabling us to derive an upper bound on the geodesic distance between the subspaces obtained by filtering a clean sample and its adversarially perturbed counterpart.It is also supported by empirical analysis, which suggests that the geodesic distances between the subspaces corresponding to clean and adversarial examples

belonging to the same class are smaller than those corresponding to samples from different classes. Figure 7.2 illustrates the subspace representation and shows that the distribution of geodesic distances computed between subspace pairs of the same class and that of different classes are reasonably separable. This observation is central to our approach and validates that our choice of filters ensures a low-dimensional subspace that is representative of the test sample's original class and serves as an appropriate smoothing operator for the input samples. Through extensive experiments on the ImageNet dataset, we show the effectiveness of GraCIAS on several models under attack of various strengths.

7.4 Existing Literature

The vulnerability of neural networks to adversarial perturbations has led the growth of a large number of defense strategies. Given the large volume of work in this area, we categorize the literature into two broad groups and briefly review recent developments in them.

Robust Training and Network Modification. Robust training refers to strategies that retrain a model with an augmented training set. The most popular strategies use adversarial training [120, 55, 95], where adversarially perturbed samples are included in the training set on the fly, i.e., during the model training. These approaches, while effective, are very computationally expensive, as the attacks have to be regenerated multiple times during the entire training process. On the other hand, approaches [113, 165] modify network architecture to achieve adversarial robustness. Xie *et al* [177] added feature denoising blocks in the model to circumvent the impact of noisy feature maps caused due to adversarial perturbations at the input. Whereas, [165] transformed layerwise convolutional feature map into a new manifold using non-linear radial basis functions. However, both robust training and network modification, not only require access to model parameters and training data, but also necessary computational resources to perform the retraining, which may be nontrivial to obtain. This requirement poses a bottleneck in securing the already deployed systems for

various applications that are built on deep learning models. Therefore, in case of restrictions on computational resources or access to model parameters, add-on defenses in the form of pre-processing blocks at the input or output of a network are viable.

Input Transformations. The limitations of the previous category are addressed by the input transformation based approaches, which aim to denoise the image before feeding it to the network for classification. Most transformation based defenses like the ones proposed in [63], e.g., image compression, bit depth reduction, image quilting etc., lead to *obfuscated gradient*, a way of gradient masking that gives a false sense of security. Such defenses have limited robustness on a given model under powerful attacks [7]. Other approaches that have been successful to some extent under strong attacks like the Barrage of Random Transforms (BaRT) [138] follow a highly randomized approach to choose at random from an *enormous* pool of transformations, making it difficult for the attacker to break. BaRT requires the model to be finetuned on the input transformations to reduce the drop in performance on clean samples. Another defense in this category [113] improves the standard JPEG compression to tackle the adversarial attack without significant drop in performance of clean samples. Other approaches exist that assume access to the training set. They either learn valid range spaces of clean samples, or use the training set to approximate the image manifold, on which the input image is projected [148, 159].

Our proposed approach GraCIAS is also an input transformation based approach. We emphasize that unlike most existing approaches, our defense is agnostic to model and the training set used.

7.5 Proposed Approach

Given a trained deep network, an adversary can add an imperceptible perturbation to the input sample that forces the model to make a wrong prediction. For a given sample $\mathbf{x}$, an adversary generates a sample $\hat{\mathbf{x}} = \mathbf{x} + \delta$ such that its label $l(\cdot)$ does not match that of the

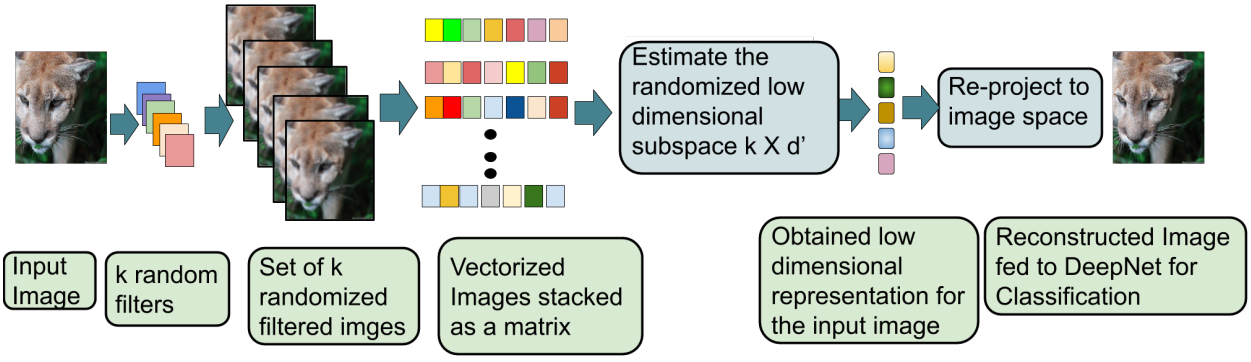


Figure 7.3 An overview of our defense applied on an adversarial sample. The number of k random filters are used for creating a set of corrupted images. These images are used to estimate a random d dimensional subspace that is used for obtaining the low dimensional representation followed by re-projection to image space to obtain a rectified image.

original sample, i.e., $l(\mathbf{x}) \neq l(\hat{\mathbf{x}})$. Thus, the objective of an attacker can be understood as follows $\arg\max \mathcal{L}(\mathbf{x}+\delta, y) \quad s.t. \|\delta\|_p < \epsilon$

Here, $y = l(\mathbf{x})$ is the ground truth label of sample $\mathbf{x}$, δ is the added perturbation and ϵ is the perturbation bound.

Design Goal: The goal of an input transformation based defense strategy is to 'clean' an adversarial sample before feeding it into a classification network. The 'cleaning' should reduce mis-classifications due to the perturbation, while maintaining performance on clean samples. In this work, we propose an input transformation-based *inference time* approach that is *simple* and *methodically randomized* to achieve effective defense.

We strive to achieve this design goal and develop our defense strategy *GraCIAS* combines simple randomized corruptions, subspace projections and a geometric perspective on the input transformations. Figure 7.3 shows an overview of our approach, which we describe in detail in the following sections.

7.5.1 Proposed Defense Strategy

The process of generating the transform to rectify an adversarial sample $\mathbf{x}_p$ is described as follows:

Step 1: Image Set for Subspace Approximation.

An input defense strategy aims to find a transform that can estimate a clean sample from a given adversarial sample. As opposed to Barrage of random transforms [138], we focus on developing a random transform that is minimal without compromising the effectiveness of the transform. Our transform comprises of a projection step from image space to a low dimensional space and a reconstruction step to project back to image space. The first step is to generate a database required for estimating the subspace for projecting the adversarial sample. To this end, we use random image filtering to generate several noisy versions of a given adversarial sample. For k such random filters, the set of k blurred images can be written as

$$\mathbf{X}_p = \{\mathbf{x}_p * \mathbf{h}_1, \mathbf{x}_p * \mathbf{h}_2, ...\mathbf{x}_p * \mathbf{h}_k\} \tag{7.1}$$

This step essentially corresponds to mixing a uniformly structured noise to the nonuniform noise in the image caused due to adversarial perturbation. This mixing is achieved by multiple convolutions with kernels $\{\mathbf{h}_1, \mathbf{h}_2...\mathbf{h}_k\}$ that have uniformly distributed weights normalized to have unit ℓ_1-norm *i.e.* $h_i(m,n) \sim \mathcal{U}[0,1]$ and $||Vec(\mathbf{h}_i)||_1 = 1$, where $Vec()$ is the vectorization operator and m, n are the indices of the kernel. In addition to random kernels, the *number* of such kernels, k to create the set of corresponding k images, is also picked at random from a fixed range of values k_{range}. This choice of randomizing filter kernels and their number is driven by our goal to increase randomness in the defense. Such random filters are more effective than other filters like Gaussian blur, where their parametric nature is much easier for attackers to approximate.

Step 2: Randomized Subspace for Low Dimensional Projection.

As $\mathbf{X}_p$ is derived from random filtering of the input image itself, the span of its elements is likely to retain some information relevant for the end task. So we simply find an orthogonal basis for the subspace spanned by $\mathbf{X}_p$. The projection of the input image on to this subspace has a blurring effect on the adversarial perturbation mixed with random noise. In the ab-

sence of the adversarial perturbations, i.e., for clean samples, a similar blurring is expected along with retention of task-relevant information. Computing the basis for the subspace is computationally inexpensive even for high resolution datasets like ImageNet as the set of corrupted images is fairly small.

Step 3: Re-projecting Low Dimensional Representation into Image Space.
The final step is to reconstruct the image from the low dimensional mapping obtained in the previous step. As in PCA based reconstruction using the low dimensional mapping and the inverse transform, we obtain the restored image. The basis of the subspace will capture relevant image content in the first few leading principal components while the noisy components are captured by the later. Thus, restoring to a low dimensional subspace will filter out the noisy components. However, using a fixed dimension for subspace can be easily estimated by a white box adversary, weakening the defense effectiveness. Therefore, the dimension of the subspace is defined based on retaining a specific value of variance in the data. The variance value is selected randomly from a predefined range of values. This adds another level of randomness in choosing the subspace dimension, making the defense effective in the presence of an adaptive attacker that is aware of the defense strategy.

7.5.2 Validity of Proposed Subspace

We now present our analysis to show that the subspace estimated with an adversarial example is close to subspace created with clean image counterpart. Our theoretical result is based on bounds obtained on the geodesic distance between the subspaces constructed by using the clean sample $\mathbf{x}_c$ and the adversarially perturbed sample $\mathbf{x}_p$.

The column space of $\mathbf{X}_p$ and $\mathbf{X}_c$ are the subspaces containing the set of convolutions of $\mathbf{x}_p$ and $\mathbf{x}_c$ with random kernels h_i i.e., $span(\mathbf{X}_p)$ and $span(\mathbf{X}_c)$ respectively. We represent the subspaces as $\mathcal{X}_p$ and $\mathcal{X}_c$ for perturbed and clean image respectively, which are both d-dimensional subspaces in $\mathbb{R}^N$, where N is the dimension of the vectorized image. Now,

we want to compare these linear d-dimensional subspaces in $\mathbb{R}^N$ for which we make use of the Grassmann Manifold, $\mathcal{G}_{N,d}$, which is an analytic manifold, where each point represents a d-dimensional subspace in $\mathbb{R}^N$ regardless of the specific bases of the subspace. The distance between the two subspaces is then given by the geodesic distance between the points on the Grassmann manifold. The normalized shortest geodesic distance is defined as follows:

$$d_{ng}(\mathcal{X}_c, \mathcal{X}_p) = \frac{1}{D} d_g(\mathcal{X}_c, \mathcal{X}_p) \tag{7.2}$$

Here, D is the maximum possible distance on $\mathcal{G}_{N,d}$ [61, 102]. It was shown in [61], that this normalized distance is upper bounded by the following expression:

$$d_{ng}(\mathcal{X}_c, \mathcal{X}_p) \leq \|\mathbf{X}_c\|_F \left\|\mathbf{X}_c^\dagger\right\|_2 \frac{\|\Delta\mathbf{X}\|_F}{\|\mathbf{X}_c\|_F} \tag{7.3}$$

$$\leq \left\|\mathbf{X}_c^\dagger\right\|_2 \|\Delta\mathbf{X}\|_F \tag{7.4}$$

Here, $\left\|\mathbf{X}_c^\dagger\right\|_2$ is the spectral norm of pseudo inverse of $\mathbf{X}_c$, $\Delta\mathbf{X} = \mathbf{X}_c - \mathbf{X}_p$ and $\|\cdot\|_F$ denotes the Frobenius norm. Rewriting, Eq. (7.3) with squaring on both sides as

$$d_{ng}^2(\mathcal{X}_c, \mathcal{X}_p) \leq \left\|\mathbf{X}_c^\dagger\right\|_2^2 \|\Delta\mathbf{X}\|_F^2 \tag{7.5}$$

Here, $\left\|\mathbf{X}_c^\dagger\right\|_2$ corresponds to the smallest eigenvalue of the $\mathbf{X}_c^\dagger$ *i.e.* inverse of the smallest singular value of $\mathbf{X}_c$. Hence, we can write the following:

$$\left\|\mathbf{X}^\dagger\right\|_2 = \frac{1}{\sigma_{min}(\mathbf{X})} \tag{7.6}$$

Here, σ_{min} denotes the smallest singular value. In case of natural images the σ_{min} is non-zero. The eigenvalues of a blurred image decay faster than the clean images, as the blurred images are dominated by low frequency components [66]. Similarly, the other factor in the right side of Eq. (7.5) is given by

$$\|\Delta\mathbf{X}\|_F^2 = \sum_i \|\mathbf{H}_i\|^2 \|\delta\|^2 \tag{7.7}$$

Here, $\mathbf{H}_i$'s are BTTB matrices that are full rank under zero boundary condition for convolution and δ is the adversarial noise added to the clean image. Thus, substituting Eq. (7.6) and Eq. (7.7) in Eq. (7.5), we get

$$d^2_{ng}(\mathcal{X}_c, \mathcal{X}_p) \leq \frac{1}{\sigma_{min}(\mathbf{X}_c)} \sum_i \|\mathbf{H}_i\|^2 \|\delta\|^2 \tag{7.8}$$

The bound in the above equation establishes that the subspaces of clean and adversarial sample are in close proximity, as long as the singular value term is bounded above. To show the latter, we adopt the following approach. It is not possible to provide a general bound on $\sigma_{min}(\mathbf{X})$ without assuming something about natural image statistics. On the other hand, it is easy to see that $\sigma_{min}(\mathbf{X})$ is small only for pathological examples. Consider the case of $\sigma_{min}(\mathbf{X}) = 0$. This happens if and only if the columns of $\mathbf{X}$ are linearly dependent. That is, there must exist non-zero scalars α_i such that:

$$\mathbf{x}_c = \sum_i \alpha_i \left(\mathbf{x}_c * h_i\right) \tag{7.9}$$

Using simple Fourier transform arguments, it can be shown that the above happens only under pathological cases such as when $\mathbf{x}_c$ is a constant-image, or the filters h_i are all just plain delta functions. For any general situation, $\sigma_{min}(\mathbf{X})$ is finite, although a general lower bound is hard to find.

The random filtering and the subspace projection reduces the effect of adversarial noise. Further, in the above discussion, we have established that the subspaces derived from clean samples and those from their adversarially perturbed counterparts are close to each other. If such a subspace $\mathcal{X}_c$ is representative of the clean sample, i.e., if it captures information relevant to the end task, then a nearby subspace like $\mathcal{X}_p$ is also likely to retain similar information. Therefore, a projection and reconstruction operation on such a subspace ($\mathcal{X}_c$ or $\mathcal{X}_p$) will achieve our objective of reducing adversarial noise and yet retaining relevant information. In addition to this proximity guarantee in Eq. (7.8), our empirical analysis of geodesics in Fig. 7.2 shows that the geodesic distances between subspaces obtained from an

adversarial sample is closer to its clean counterpart than that of another clean image from a different class.

This observation along with our proximity result is the basis of our main hypothesis: The subspaces resulting from our randomly corrupted versions of the input image are sufficiently representative to retain task-relevant information and yet are effective transformations in attenuating adversarial noise. In the following sections, we validate this hypothesis through extensive empirical evaluation and analysis experiments.

7.6 Experimental Setup and Results

In this section, we firstly evaluate our defense strategy as pre-processing at inference time on adversarial samples generated with different perturbation magnitude as well as attacker's knowledge. Secondly, we also present ablation experiments to thoroughly evaluate the choice of parameters used in the proposed defense strategy. Now, we list down the dataset, models and attack methods used to evaluate the performance of our defense strategy.

7.6.1 Experimental Setup

We present a series of experiments to evaluate the effectiveness of our defense strategy.

Datasets and Models

Datasets: ImageNet-50K[147] dataset contains images of different sizes distributed along 1000 categories. The images are processed to 256×256 dimensions encoded with 24 bit color. The validation set consists of 50,000 images. We represent this entire set as ImageNet-50K and a subset of first 10,000 images as **ImageNet-10K** for our experiments.

Models: For ImageNet [147], we evaluate the performance on InceptionV3 [163], Resent50[70], MobileNet [75] and VGG16 [156]. We used pre-trained models available in Tensorflow.

Comparison with other Approaches.

We compare our defense strategy with several input transformations that are summarized below:

1. **JPEG compression, BitDepth reduction [63] and JPEGDNN [113]**: These defenses are applied to an image only at the inference time. Like our approach, these approaches are model agnostic and can easily be integrated with existing systems. We used the framework of [7] and the authors' implementation from github for evaluating these defenses under different attack scenarios. The JPEG defense is performed at a compression quality level of 75 (out of 100) and the images are reduced to 3 bits for BitDepth defense in all our experiments. For JPEGDNN, we used default parameters available with the authors' implementation that is available on github.

2. **Barrage of Random Transformation [138]** We compared state-of-the-art performance of BaRT on ResNet50 model with our defense strategy on ImageNet50K. As the authors' implementation is not publicly available, we use results reported in their paper in our comparisons. While we outperform BaRT in accuracy of attacked images by a significant margin, our performance on clean images is lower. However, it is essential to point out that BaRT's model is finetuned for an additional 100 epochs using BaRT transformed images, whereas we use the original model.

Parameters in GraCIAS.

In our defense strategy, two steps require random parameter choice. First is the *number* of filters k to create the set of corrupted images as given by Eq. (7.1). To limit the computational cost, it is chosen from a fixed range k_{range}. This range is set to $\mathcal{U}[10, 60]$ i.e. a minimum of 10 corrupted images to a maximum of 60. The other random parameter that adds to the proposed defense strategy's robustness is the dimensionality of the subspace defined on the set of images mentioned earlier. The dimension is computed based on the %

of data variance retained while calculating the PCA basis. To avoid information loss and drastic image changes, the variance is selected between 60 % to 95% at random. The third parameter is the kernel size for the filtering operation. The filter size is fixed to 7×7 for ImageNet dataset.

7.6.2 White Box Attacker

The white box adversary can access model parameters, training data, and trained weights to generate adversarial samples. We evaluate the performance of different models under FGSM [55] and PGD [120] attacks with L_∞ distance. The different iterations of PGD attack are denoted by PGDk. For example, PGD with 10 iterations is denoted with PGD10.

We evaluate the performance of InceptionV3 model under FGSM and PGD attack. The results are present in Table 7.1 with perturbation value of $\epsilon = 16$. The table also presents results corresponding to standard JPEG compression and BitDepth reduction that are effective when the defense is not known to the attacker. These transformations are applied at the inference time to transform the sample before feeding it to a pre-trained InceptionV3 network. The recent work on DNN guided JPEG compression [113] was shown to achieve state-of-the-art results compared to other input transformations like image quilting, standard JPEG and bitDepth. While JPEGDNN performs well in small perturbation setting, the performance drops for high perturbation value of $\epsilon = 16$ as reported in the table. Our approach outperforms previously best known results using JPEGDNN as input transformation.

Performance across different Perturbation Magnitude. We evaluated our defense strategy's performance over a range of perturbation magnitude for both PGD and FGSM attacks and compared them with state-of-the-art approaches. The results are shown in Figure 7.4 for InceptionV3 and for VGG16 and ResNet50 in Figure 7.5. The results indicate that our approach sustains for larger range of perturbation before dropping at large magnitudes of perturbations.

Attack	ImageNet10K (InceptionV3)				
	No def	JPEG (ICLR'18)	BitDepth (ICLR'18)	JPEGDNN (CVPR'19)	GraCIAS (Our)
FGSM	22.88	25.32	24.76	26.21	**37.63**
PGD40	0.0	0.65	0.27	3.52	**42.49**
PGD100	0	0.0	0.23	3.01	**44.32**

Table 7.1 ImageNet 10K validation set: Comparison of different input transformation based defense on InceptionV3 model. The table reports defense classification accuracy under FGSM, PGD40 and PGD100 attacks with an attack magnitude of $\epsilon = 16$

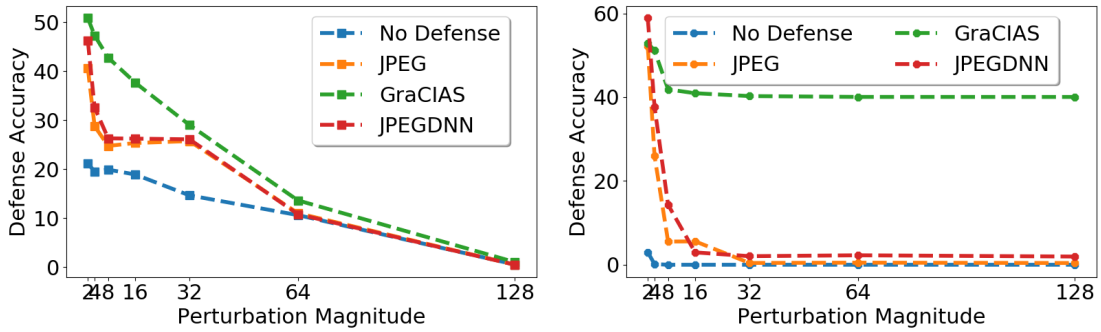


Figure 7.4 Performance Comparison of various defenses across different magnitudes of ϵ using (Left) FGSM and (Right) PGD10 attacks on InceptionV3 model.

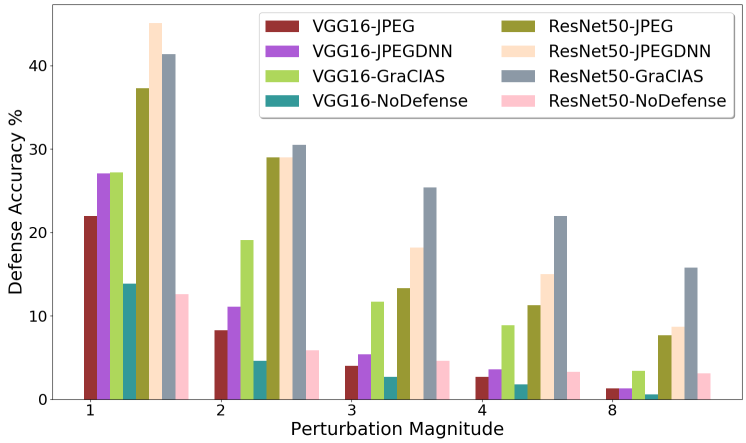


Figure 7.5 Performance comparison of different defense strategies under different magnitudes of FGSM attack on ImageNet dataset for VGG16 and ResNet50 models.

7.6.3 Adaptive Adversary

The recent work [7] showed that existing input transformation based defense strategies can be attacked with a strong attacker that has access to the defense strategy as well. A non-differentiable defense strategy can be defeated with BPDA (Backward Pass Differentiable Approximation) that approximates the gradient with identity while backpropagating through the transformation layer to develop strong attacks. We show that the proposed defense strategy can withstand such attacks as opposed to existing defense strategies. The results in Table 7.2 indicate that the proposed approach achieves state-of-the-art results on both InceptionV3 and ResNet50 models with approximately 4% improved over state-of-the-art results. We also validated the efficacy of our approach across different perturbation magnitudes and attack iterations in Fig 7.6 and show that GraCIAS achieves non-trivial accuracy as opposed to state-of-the-art input defense JPEGDNN.

Defense	Apply	Models			
		InceptionV3	ResNet50	MobileNet	VGG16
JPEG [63](ICLR '18)	Only Inference	9.82	0.0	0.0	0.0
JPEG DNN [113] (CVPR '19)	Only Inference	13.12	0.35	0	18.38
*BaRT k =5 [138] (CVPR '19) [	Finetune+ Inference	NA	16.0	NA	NA
*BaRT k =10 [138](CVPR'19)	Finetune+Inference	NA	36.0	NA	NA
GraCIAS	Only Inference	**19.65**	**41.94**	**35.6**	**21.5**

Table 7.2 ImageNet Validation Set: Performance comparison of defense classification accuracy under BPDA attack ($\epsilon = 16$, iteration 40) on InceptionV3, ResNet50, MobileNet and VGG16 models. * indicates that the results are quoted from the respective paper, in the absence of open source implementation.

To further ensure the effectiveness of proposed defense strategy, we also evaluate it against EOT (Expectation over Transformation). The attacker aims to capture the randomness in the transform by performing the transformation multiple times and using the average gradient. However, due to the presence of randomness at different steps of the transformation, the expectation over the transformations fails to capture the randomness in the transform, even

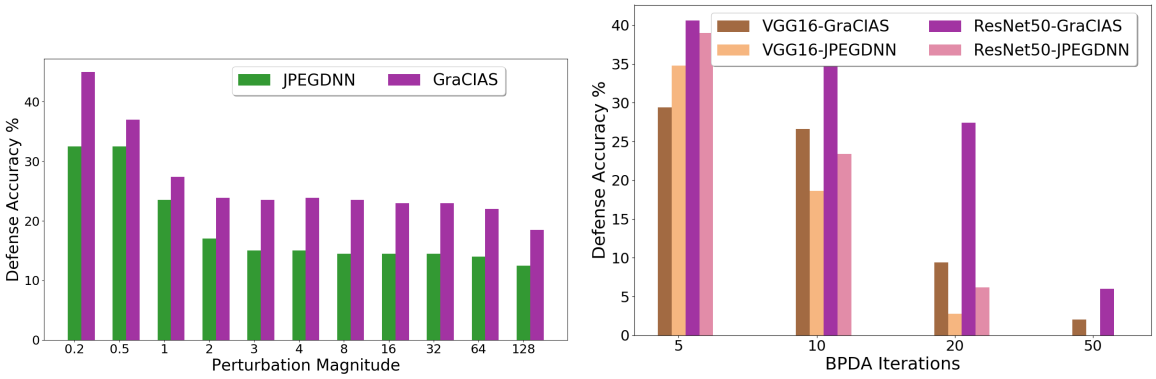


Figure 7.6 Left: InceptionV3 model under BPDA attack with different perturbation magnitude on ImageNet dataset. The plot highlights that GraCIAS achieves state of the art results over previously reported with JPEGDNN. Right: Performance of defense accuracy on ResNet50 model under different iterations of BPDA attack with $\epsilon = 8$. While both ResNet50 and VGG16 are completely defeated at increased attacker's strength, GraCIAS still achieves non-trival defense performance.

with as large as 100 runs. The InceptionV3 model achieved an accuracy of 40.19% over 100 steps of EoT. Further, we also investigated the combination of the two i.e. BPDA +EoT, where the defense achieved an accuracy of 19.1% similar to BPDA alone.

Effect on Clean Sample Accuracy. In the absence of *detection* of adversarial examples, the input transformation strategies are applied to both clean and adversarial samples, adversely effecting the performance on clean samples. To cope with this drop, the network is finetuned with the proposed defense, before applying defense at the inference time. We report that with GraCIAS, performance on clean samples drops by ∼16 % and ∼23 % for Inception and ResNet50 models on ImageNet without finetuning respectively. However, this drop reduces with network finetuning as suggested by BaRT that achieved 65% on clean samples against original clean sample accuracy of 76%, *i.e.* 11% drop. Due to limited hardware resources, we verified this on ResNet model for CIFAR10, where the finetuned model regained the drop by 8%, suggesting similar benefits for ImageNet dataset on finetuning with our defense strategy.

7.7 Ablation Study

We now investigate the choice of parameters in our defense strategy. While the range for the percentage of variance in data to be retained as well as the range of number of filters is fixed across all the experiments, the parameter like filter size can depend on the dimension of the image. Also, image operations' performance to create the corrupted image set is evaluated under adaptive attack settings to validate the effectiveness of random filters over others.

7.7.1 Effect of Filter Size

Our filtering operation aims to develop a diverse yet informative set of images to estimate a subspace that retains task-relevant information. This effect is verified from the results in Table 7.3 on ImageNet dataset for two different perturbation magnitudes across three different filter sizes.

Operation	Defense Accuracy
Gaussian Filter	17.11
Affine Transformation	11.86
Symmetric Transformation	7.29

Filter Size	$\epsilon = 8$	$\epsilon = 16$
3	17.11	16.41
5	17.43	18.90
7	22.38	19.65

Table 7.3 Left: Effect of selecting different transforms to create the set of corrupted image given in Eq. (7.1) needed for our GraCIAS defense. Right: Effect of filter size on defense performance on ImageNet dataset at different perturbation levels under adaptive adversary (BPDA+PGD) with 100 iterations.

The results are reported in the adaptive attack setting (PGD +BPDA) to show the effect in stronger attack scenario. The results also point to the fact that in most real world applications, higher image resolutions are encountered and the choice of filter size is not difficult.

7.7.2 Random Filters vs. Other Transforms

We evaluated the effect of different image transforms to create the set of images in Eq. 7.1 required for defining the subspace. The results for the same are given in Table 7.3. The performance suffers a significant drop with affine and symmetric transformation, due to the likely reason that the typically high frequency adversarial noise are still retained after such transformations.

7.7.3 GraCIAS as Pre-processing Prior to Other Defenses

JPEG compression and BitDepth reduction are simple input transformations that are very easy to incorporate in real world systems as they are model agnostic and can be implemented in hardware as well. However, they become ineffective in the presence of large perturbation as well as an adaptive adversary. Since the proposed defense is also inexpensive in terms of computation resources for its implementation, it can complement these defenses to retain their defense capabilities. The results in Table 7.4 show improvements on ImageNet dataset under PGD attack.

Defense	Accuracy		
	InceptionV3	ResNet50	VGG16
JPEG	5.58	39.2	18.9
GraCIAS → JPEG	**44.26**	**45.14**	**31.79**
BitDepth	0.42	39.38	27.95
GraCIAS → BitDepth	**25.97**	**43.79**	**29.89**

Table 7.4 ImageNet-10K: Performance of simple defenses with GraCIAS used as a pre-processing step at the inference time on various models with $\epsilon = 16$ under PGD10 attack. The boost in the performance is indicative of GraCIAS ability to restore the image details, making it easier for much simpler defense like JPEG and BitDepth to defend against the attack.

7.8 Conclusion and Future Scope

In this chapter, we proposed a simple randomized linear subspace-based input defense approach applied at inference time to mitigate the effect of adversarial noise. The proposed approach achieved state-of-the-art results on ImageNet dataset across four different deep classification networks. The proposed defense is extensively evaluated on attacks with different strengths and magnitudes and is effective in all cases. We have established both empirically and theoretically, the validity of our strategy in cleaning the adversarial perturbation. In future, it would be interesting to extend this approach further to develop the robustness guarantees [34].

Chapter Eight

Geometry of Disentangled Representation Learning Models

As most recent representation learning approaches use DNNs as the base models, there has been an increased interest in understanding their geometry. The work of Brahma et al. [23] focused on providing theoretical insights and justification behind the success of deep learning models for classification task. Owing to the recent success of deep generative models like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) in various applications like realistic image generation [137], learning interpretable features [162, 122] and unsupervised domain adaptation [136], several recent works have also focused on establishing the geometry of generative models [6, 92, 151, 97]. Shao et al. [151] suggested that while the manifolds learned by deep generative models are nonlinear, they have close to zero curvature. This conclusion further implied that the linear movements in the latent space result in movements approximately along the geodesics of the generated data manifold. Thus, statistics computed on a much smaller dimensional latent space, can be used along with the Riemannian geometry induced by the generator mapping function $f(\mathbf{z})$, to draw inferences in the high dimensional data manifold. Later, [6, 92] showed that the latent space is typically nonlinear and computations should appropriately account for its curvature. The analysis in these papers has been restricted to latent space of one type of generative model

i.e. VAEs.

The generator of a deep generative model like VAE can be seen as a mapping from the low dimensional latent space to the data manifold embedded in a much higher dimensional space. This permits us to define the Riemannian metric of the data manifold via the Jacobian of the mapping. The work of Shao et. al. [151] developed the Riemannian metric for deep generative models and analyzed the curvature of the generated data manifold. They empirically showed that for VAEs, the resulting manifold is non-linear, yet has a low curvature as indicated by the low disparity between the Euclidean and the geodesic distances. However, they do not directly provide a compact measure for quantifying the curvature. Aravanitdis et. al. [6] also used the Riemannian metric to define non-linear statistics in the latent space of VAEs, and argued that the latent space is a curved space and advance the use of geodesic interpolation.

As pointed out by these recent works, leveraging the Riemannian metric in generative models helped in smoother interpolations as well as more meaningful interpretation of distances as opposed to the default Euclidean metric. As in the previous works, we restrict our study to deep generative models. However, we focus our investigation on the geometric properties of VAE-based disentangled latent space models. Furthermore, we aim to quantify the curvature of the latent space and resort to different metrics for studying the following: Firstly, we validate and quantify the near zero curvature of VAE as addressed in [151]. Secondly, we extend the geometric analysis to disentangled representation learning models and quantify the effect of class separability on the curvature of the generated data manifold.

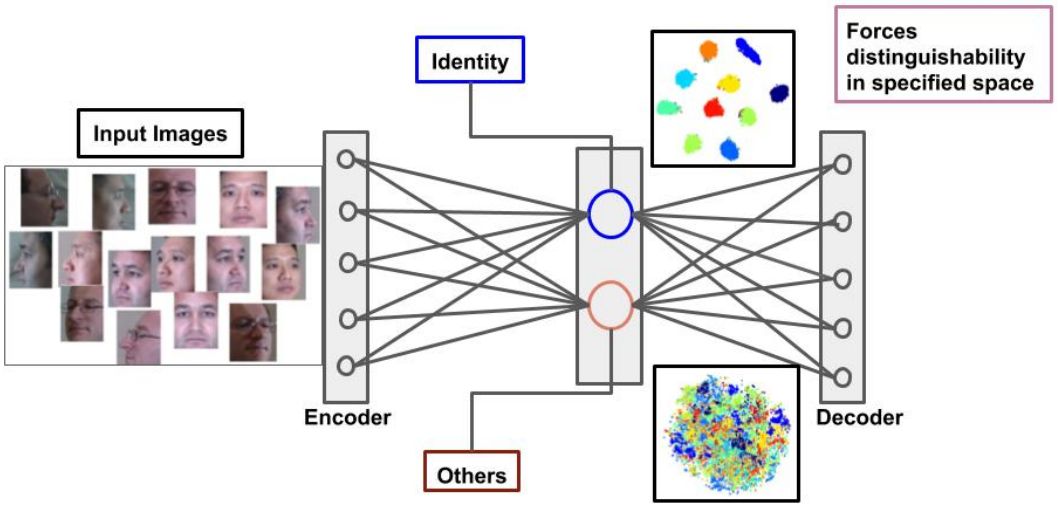


Figure 8.1 Model for learning factorized latent space representation for human face recognition. The identity constitutes the specified component while all other factors such as pose and illumination are considered in unspecified space.

8.1 Disentangled Representation Learning Models Under Study

The focus of this work is limited to factorized disentangled representations that partition the latent space into two codes: one code for the specified factor and the other code for all uncontrolled or unspecified factors. For example, the latent space of face data set can be partitioned with identity as specified factor whereas factors like illumination, pose and expression together as unspecified factor. An illustrative example is shown in Figure 8.1 with identity as the specified factor and all other attributes like pose, illumination, facial hair as the unspecified component. In this work, we study the geometry of three models proposed for disentangling [162], [122] and [82].

Szabo *et al.*[162] The model learns a disentangled latent representation using an encoder-decoder architecture with weakly labeled data in the form of triplets $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ have the same label, while $\mathbf{x}_3$ has a different label. Note that *absolute* labels are not required for this architecture, hence the weaker form of supervision. As shown

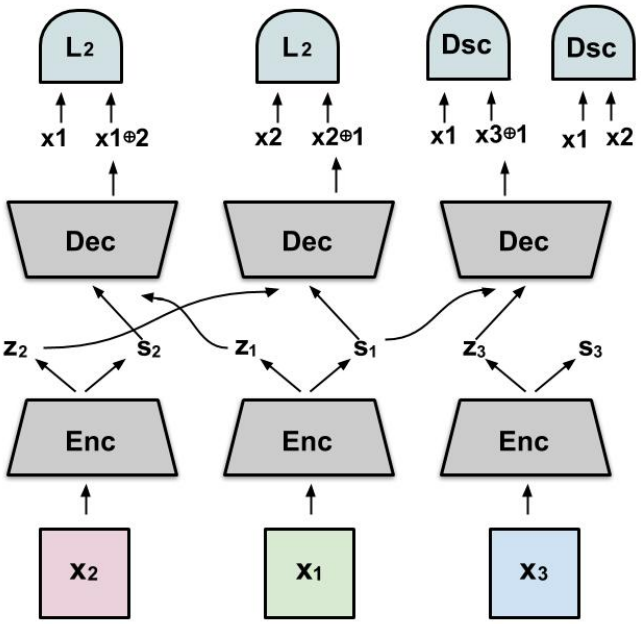


Figure 8.2 Network for disentangled representation learning given in [162].

in Figure 8.2, during the training, the specified and unspecified components are swapped and an appropriate loss function is optimized to ensure that the decoder learns to generate realistic samples. In the first step, since $\mathbf{x}_1$ and $\mathbf{x}_2$ have the same labels, the encoder should learn to satisfy $\mathbf{s}_1 \simeq \mathbf{s}_2$ (latent space representation corresponding to the specified attribute). Upon swapping the unspecified components $\mathbf{z}_1$ and $\mathbf{z}_2$ i.e. representations corresponding to the unspecified attributes, a simple ℓ_2 loss suffices to ensure that the generated samples$\mathbf{x}_{2\oplus1}$ obtained is similar to $\mathbf{x}_2$ as well as $\mathbf{x}_{1\oplus2}$ looks similar to $\mathbf{x}_1$. On the other hand, when $\mathbf{z}_3$ and $\mathbf{z}_1$ (or $\mathbf{z}_2$) are swapped, the loss cannot be an ℓ_2-norm, which Szabo et al. [162] circumvent by using a discriminator with an adversarial loss. The discriminator is trained over real pairs $(\mathbf{x}_1, \mathbf{x}_2)$ and fake ones $(\mathbf{x}_{3\oplus1}, \mathbf{x}_1)$, so as to enable the decoder to generate samples that resemble those from the distribution of $\mathbf{x}_1$ and $\mathbf{x}_2$.

The objective function consists of two terms: autoencoder loss and adversarial loss, and

is given by

$$\min_{Dec,Enc} \max_{Dsc} \mathcal{L}_{AE}(Dec, Enc) + \lambda \mathcal{L}_{GAN}(Dec, Enc, Dsc) \tag{8.1}$$

Here $\mathcal{L}_{AE}$ is given by

$$\mathcal{L}_{AE} = \mathbb{E}_{(\mathbf{x}_1,\mathbf{x}_2)} ||\mathbf{x}_1 - f(g_s(\mathbf{x}_1), g_z(\mathbf{x}_2))||_2^2 + ||\mathbf{x}_2 - f(g_z(\mathbf{x}_2), g_s(\mathbf{x}_1))||_2^2 \tag{8.2}$$

and, the adversarial loss is given by

$$\mathcal{L}_{GAN} = \mathbb{E}_{\mathbf{x}_1,\mathbf{x}_2}[\log(d(\mathbf{x}_1, \mathbf{x}_2))] + \mathbb{E}_{\mathbf{x}_1,\mathbf{x}_3}[\log(1 - d(\mathbf{x}_1, \mathbf{x}_3))] \tag{8.3}$$

The model by **Mathieu *et al.*** [122] is similar to the one described above. However, the key difference is they train the unspecified component as a VAE, i.e., impose a KL-divergence term such that the unspecified component has a standard normal distribution. The work by **Jha *et al.*** [82] improves upon the two approaches [122, 162] by substituting the adversarial training strategy with cycle-consistency of the unspecified latent space.

Geometric Perspective All the three methods, despite their differences, attempt to partition the space into specified and unspecified components. In context of the work in this chapter, the two important and common aspects of these methods are

- Encoder-decoder based architectures, that partition the latent space into specified and unspecified factors.
- Specified latent space comprises discriminative features of the specified attributes, while the unspecified space contains uninformative, nuisance variables.

We focus on analyzing the geometry of the two components of the learned latent space $(\mathbf{s}, \mathbf{z})$ and the corresponding generator mapping functions $(f_{\mathbf{S}}(\cdot), f_{\mathbf{Z}}(\cdot))$. We expect this analysis and quantitative validation to provide insights into the geometric aspects of learning disentangled latent spaces, which in-turn may help design better network architectures and training strategies.

8.2 Geometry of Latent Space of Factorized Representations

In this section, we present the different metrics used to compare the latent spaces of the learned disentangled representation.

8.2.1 Euclidean vs Riemannian Metric

The latent space of a generative model provides a low dimensional representation of the data manifold via nonlinear functions implemented by the encoder and decoder. Thus, the statistical computations based on manifold theory are more appropriate as opposed to the Euclidean space assumption. As pointed out by works in [6, 91], the latent space is the coordinates for the data manifold though a generator mapping function. Thus distances and other statistics are better defined with a Riemannian metric. This manifold assumption holds true provided the generator is a smooth function.

In deep generative models, the network is a composition of multiple convolutonal layers followed by activation layers that bring in the nonlinearity of the feature space. In order to obtain a smooth generator function, the activation function such as ELU (exponential linear unit) is used over the more popular ReLU (rectified linear unit).

Thus, the two conditions required for the generator mapping f to define a smooth manifold [151] are:

- The activation function is smooth and monotonic function.
- The weight matrices for the layers are full rank. This condition in effect translates to the full rank condition of Jacobian matrix at every point in the latent space
  $rank(\mathbf{J}_f(\mathbf{z})) = d$
  where $\mathbf{z}$ is a point in the latent space and d is the dimension of the latent space.

Provided that the above conditions are satisfied, every point in the coordinate latent space is mapped uniquely on to the data manifold. This connection allows one to perform operations in the low dimensional latent space, that is relatively computationally cheap as opposed to their higher dimensional counterpart, the data manifold. Given a point in the latent space, the Jacobian provides a mapping from the tangent space of latent space to the tangent space of the data manifold. The associated Jacobian at every point in the coordinate space allows to define a local metric at every point in the space that accounts for distortion brought by low dimensional latent space representations. Thus, the Riemannian metric is defined at every point $\mathbf{z}$ as a symmetric positive definite matrix $\mathbf{M_Z}$ as

$$\mathbf{M_Z} = \mathbf{J}_f(\mathbf{z})^\top \mathbf{J}_f(\mathbf{z}) \tag{8.4}$$

Here, $\mathbf{J}_f(\mathbf{z})$ is the Jacobian matrix at point $\mathbf{z}$ and $f()$ is the generator function.

8.2.2 Residual Normalized Cross Correlation

This metric has been used to establish the relation between geodesic and Euclidean distances in [23] for establishing the flattening achieved due to unsupervised pre-training of the deep network. While [6, 151] discuss the degree of nonlinearity (or flatness) of the approximated data manifold in a VAE, the curvature of the established manifold is not quantified. The residual cross correlation measures the similarity between Euclidean and Riemannian distance and hence can provide an indirect measure of the curvature of the manifold. For example, for a linear manifold, both the distances are equal, but nonlinearity or curvature of the space increases the difference between the two quantities. The residual cross correlation is given by

$$\begin{aligned} c_k &= 1 - \frac{(r_M(k) - \mu_{r_M})(r_E(k) - \mu_{r_E})}{\sigma_{r_M}\sigma_{e_E}} \\ \hat{c} &= \frac{2}{N(N-1)}\Sigma_k c_k \end{aligned} \tag{8.5}$$

Here, r_E and r_M are the vectors formed by the concatenation of upper-triangular matrices of pairwise distance matrix for Euclidean and manifold distance respectively. μ_{r_M}, μ_{r_E} and σ_{e_E}, σ_{e_M} are means and standard deviations for manifold and Euclidean distance respectively.

8.2.3 Normalized Margin

It measures the class separability in the latent space. A higher value of normalized margin means a larger separation between clusters of different classes and hence a higher classification and clustering accuracy. Therefore, the specified space of disentangled representations would typically have a higher margin than a standard VAE.

$$m_n = \frac{||\mathbf{x}_n - \mathcal{M}(\mathbf{x}_n)|| - ||x_n - \mathcal{H}(\mathbf{x}_n)||}{||\mathbf{x}_n - \mathcal{M}(\mathbf{x}_n)||} \tag{8.6}$$

Here, $\mathcal{M}(\mathbf{x}_n)$ is the nearest member from a different class other than $\mathbf{x}_n$ and $\mathcal{H}(\mathbf{x}_n)$ is the nearest member from the same class. A larger value signifies better separation between classes.

8.2.4 Tangent Space Alignment

For a flat manifold, the tangent spaces are aligned, so the angle between the two subspaces is zero. With the increase in the curvature of the space, the tangent spaces at two points in this space would have a larger angle between them.

For a given data point $\mathbf{x}$, the neighboring points are collected as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_k]$. The mean centered data is given by

$$\hat{\mathbf{X}} = \mathbf{X} - \frac{1}{k}\mathbf{X}\mathbf{1} \tag{8.7}$$

Here, $\mathbf{1}$ is a $d\times$ k matrix of all ones. The basis for the tangent space is obtained by the singular value decomposition of the covariance matrix given by

$$\mathbf{C} = \mathbf{X}^\top\mathbf{X} = \mathbf{U}\Sigma\mathbf{U}^\top \tag{8.8}$$

Given two points the principal angle between the subspaces defined by the tangent spaces gives a measure of curvature of the space.

8.3 Experimental Setup and Results

In this section, we present the qualitative and quantitative results to analyze the geometric properties of the VAE and disentangled representation models. We provide the details of the network architecture, datasets and the results for the different evaluation metrics.

8.3.1 Datasets

All the generative models are trained for 2 datasets: MNIST digits and MultiPIE face dataset. MNIST digits consist of 60000 training samples and 10000 test samples distributed over 10 class. The specified component is the class identity whereas the unspecified factors constitute digit slant, stroke width etc. We consider a subset of MultiPIE dataset with 25 identities for training and 5 identities for test, each with 3300 images. We use identity as the specified factor and other factors like pose, expression etc are considered as unspecified. Further, we additionally evaluate the models on 3d chairs dataset that consists of images of different chair models of different styles and large viewpoint variations. We selected a subset of 30 models in different viewing angles.

8.3.2 Network Architecture

We use Convolutional Neural Networks (CNNs) for all the models. The dimensions for the specified and unspecified spaces for MNIST digits are 16 and 64 respectively. For the Face dataset, the specified and unspecified dimensions are 512 and 64 respectively. We also trained a VAE with the same encoder decoder network as for the disentangled representation models and with latent space dimensions as 16 and 512 for MNIST and Face dataset respectively.

We used ELU activation function in all our models.

8.3.3 Normalized Margin

The normalized margin is a measure of distinguishability of class specific features. The specified latent space of these models is enforced to learn class specific features and are used for task like classification. Classification accuracy is one measure to evaluate the effectiveness of these features for the given task. As pointed in [82], the performance of the three models for MNIST digit classification is equally good, reflecting similar structure in the specified space of these models. The normalized margin value given in the Table 8.1 is consistent with the classification performance of these models.

Datasets	Szabo *et al*	Mathieu *et al.*	Jha *et al.*
MNIST	0.622	0.653	0.640
MultiPIE	0.462	0.496	0.488
3D chairs	0.492	0.529	0.52

Table 8.1 Normalized margin for MNIST, MultiPIE and 3D chairs datasets.

8.3.4 Residual Cross Correlation

The values for the two datasets across different models are given in Table 8.2. Smaller value suggests higher similarity between Euclidean and geodesic distances. For the MNIST dataset, the small value of $\hat{c}$ validates the claims of near zero curvature of VAEs discussed in [151]. However, we observe that there is significant disparity between the Euclidean and the Riemannian distances for the MultiPie dataset. Note that Szabo et al. [162] failed to converge on the MultiPie data, and hence are not included in the table.

Dataset	VAE	Szabo *et al.*	Mathieu *et al.*	Jha *et al.*
MNIST	0.071	0.142	0.178	0.167
MultiPie	0.65	-	0.72	0.71
3D chairs	0.162	0.262	0.311	0.315

Table 8.2 Comparison of $\hat{c}$ values for different disentangling models with VAE for MNIST digits, MultiPIE and 3D chairs datasets.

8.3.5 Curvature of Latent Spaces

In this section, we evaluate the curvature of the latent space manifolds of VAEs. The low curvature claimed [151] is reflected by the small residual cross correlation between Euclidean and Riemannian distance given in the Table 8.3 for different dimensions of the latent space. The stability of improvement in clustering performance across different dimensions is also indicative of the low curvature of the latent space regardless of the dimensionality. We further quantify the curvature of latent spaces of VAE and disentangled representation models in Table 8.4 by computing the angle between the tangent spaces of a pair of points. The results validate higher curvature for disentangled representation models over VAEs. As the specified spaces of the three disentangled representation models are constrained to learn class discriminative features, the similarity in the curvature also reflects the same.

Dimension	16	64	80
$\hat{c}$	0.065	0.069	0.071
F score (Euclidean)	83.32	85.22	87.38
F score (Riemannian)	91.74	92.33	96.23

Table 8.3 Effect of dimensionality on the nonlinearity of the latent space of VAE. The clustering performance: F score with Euclidean and Riemannian distance as metric in K-means clustering algorithm for MNIST digits dataset.

Datasets	VAE	Szabo *et al*	Mathieu *et al.*	Jha *et al.*
MNIST	21.45	34.12	32.25	32.76
MultiPIE	27.95	37.42	36.88	36.96
3D chairs	23.45	36.77	35.86	36.50

Table 8.4 Approximate curvature estimated with principal angles between tangent spaces.

8.3.6 Riemannian Distance vs Euclidean Distance

While the c values quantify the residual cross correlation between Euclidean and Riemannian distances, we also quote the magnitudes of these distances in Table 8.5, 8.6 and 8.7 for MNIST, 3D chairs and Face datasets respectively. Due to curvature of the specified space, using Riemannian distance over Euclidean distance is more appropriate for tasks like clustering as shown in the results. For both the datasets, a significant improvement in the clustering performance validates the high curvature of the latent space for the disentangled models.

	Models	Szabo *et al*	Mathieu *et al.*	Jha *et al.*
Distances	Euclidean	0.114	0.112	0.116
	Riemannian	0.297	0.355	0.336
Clustering	Euclidean	91.12	94.32	92.22
F score	Riemannian	94.56	98.00	96.60

Table 8.5 MNIST dataset: Comparison of average distance between randomly selected 100 pairs and clustering performance: Riemannian distance vs Euclidean distance. The large differences in the distance/ F score is a result of curvature in the latent space.

	Models	Szabo *et al*	Mathieu *et al.*	Jha *et al.*
Distances	Euclidean	0.158	0.160	0.156
	Riemannian	0.344	0.376	0.365
Clustering	Euclidean	91.16	94.33	94.24
F score	Riemannian	95.22	96.34	96.44

Table 8.6 3D chairs dataset: Comparison of average distance between randomly selected 100 pairs and clustering performance: Riemannian Distance vs Euclidean Distance. The large differences in the distance/ F score is a result of curvature in the latent space.

	Models	VAE	Mathieu *et al.*	Jha *et al.*
Distances	Euclidean	0.312	0.346	0.332
	Riemannian	1.142	1.784	1.602
Clustering	Euclidean	82.98	89.37	90.06
F score	Riemannian	89.04	94.45	95.60

Table 8.7 MultiPIE dataset: Comparison of average distance between randomly selected 100 pairs and clustering performance: Riemannian Distance vs Euclidean Distance. The large differences in the distance/ F score is a result of curvature in the latent space.

8.3.7 Interpolations

Further, we also investigate the effect of Riemannian metric on interpolations in the latent space of VAEs as well as the disentangling model. The images generated by linear and Riemannian interpolations for class *2* of MNIST dataset for VAEs are shown in Figure 8.3. The images generated with Riemannian metric are sharper as opposed to Euclidean metric reflecting the non-linear nature of the space. The two ends of the sequence are the images corresponding to the cluster center of class 2 and a point at the boundary of class 2 latent space representations. Thus, along with sharper image generation, the presence of curvature in class specific manifold is also highlighted. For samples from different classes, the images for Euclidean and Riemannian interpolation are shown in Figure 8.4 for VAE as well as for the

disentangling model. As shown in the figure, the transition from one class to other is abrupt with Euclidean distance while it changes more smoothly with the Riemannian counterpart. An example with large number of intermediate interpolants between classes 8 and 2 is shown in the Figure 8.5 for both VAE and disentangled representation model. The results show the effect of the higher curved space in case of disentangled model with a huge difference between Riemannian and Euclidean interpolants as opposed to VAE. Similar results are obtained for face and chair datasets and are shown in the Figure 8.6 and 8.7 respectively.

Figure 8.3 Interpolation between two samples from same class in the latent space of VAE using Euclidean (Left) and Riemannian Metric (Right).

8.3.8 Image Synthesis

Disentangling specified component from other components, allows one to generate new images with different variations of fixed specified component. Figure 8.8 shows the effect of using Euclidean and Riemannian metric for generating images of a specific class with different styles. These images are obtained by interpolation in the specified component between the cluster center and a sample of the same class, while randomly sampling the unspecified component from the Gaussian distribution imposed in the unspecified latent space. The samples generated are more realistic and preserve class identity in case of Riemannian metric.

Figure 8.4 Interpolation between two samples from different classes in the latent spaces of VAE (Top) and specified space of Mathieu *et al.*[122](Bottom) with fixed unspecified using Euclidean (Left) and Riemannian Metric (Right).

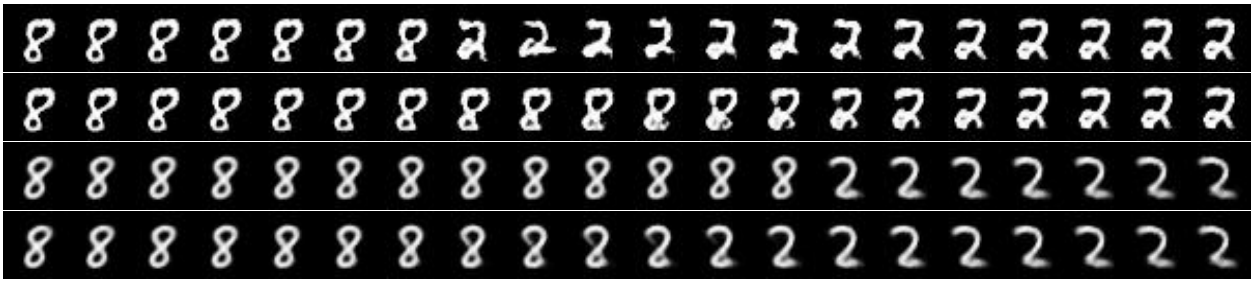


Figure 8.5 Interpolation between two samples from different classes in the latent spaces of VAE (1^{st} and 2^{nd} row)) and specified space of Mathieu *et al.*[122] (3^{rd} and 4^{th} row) with fixed unspecified using Euclidean (Odd rows) and Riemannian Metric (Even rows).

Figure 8.6 Interpolation between two samples from different classes in the latent spaces of Mathieu *et al.*[122] with fixed unspecified using Euclidean (Left) and Riemannian Metric (Right).

Figure 8.7 Interpolation between two samples from different classes in the latent spaces of VAE (left) and specified space of Jha *et al.*[82] (right) with fixed unspecified using Euclidean (Top) and Riemannian Metric (Bottom).

Figure 8.8 Interpolation between two samples from same class in the specified latent space of Mathieu *et al.* with randomly sampled unspecified component using Euclidean (Left) and Riemannian Metric (Right).

Figure 8.9 Effect of ReLU (left) and ELU (right) activation functions on the quality of generated images with Euclidean (top row) and Riemannian metric (bottom row) interpolations.

8.3.9 Rank of Jacobian

For the network to learn smooth generator function, the Jacobian matrix is required to be full rank. We compare the rank of Jacobian matrix on MNIST dataset for the two activation functions ReLU and ELU. The results given in the Table 8.8 show that the ELU results in full rank Jacobian matrices as opposed to ReLU. Further, the results in Figure 8.9 show the effect on the quality of the images generated with ReLU and eLU layers.

Activation function	VAE	Mathieu *et al.*
ReLU	8	15
ELU	16	16

Table 8.8 Rank of the Jacobian matrix for MNIST digits.

8.4 Conclusion

Latent spaces of deep generative models provide a low dimensional representation for data embedded in a high dimensional manifold. In this chapter, we studied the geometric properties of deep generative models that learn disentangled representations. We verified various recent claims about the non-linearity of the VAE based latent space representations and utilized several metrics to quantify its lower curvature. Using the same metrics, we also established that the specified components of the latent space of VAE-based disentangled models are substantially more curved. The proposed study concluded that the latent spaces

are curved. Thus an appropriate Riemannian metric as opposed to the Euclidean metric should be used for obtaining better distance estimates, performing interpolations as well as generating synthetic views.

Chapter Nine

Conclusion and Future Work

This thesis investigated geometric constraints for learning representations of visual data across different learning settings (supervised, semi-supervised and unsupervised). In particular, the geometric constraints took one of the two forms: manifold and semantic. In Chapter 3, we proposed eigenvalue decomposition parametrization of the PSD matrix for Mahalanobis distance metric learning problem. This effectively reduced the PSD matrix learning to learning of the orthonormal eigenvectors and the corresponding eigenvalues. We leveraged the geometry endowed by space of orthonormal matrices known as the Stiefel manifold to reduce the number of optimization parameters effectively. Thus, enabling the framework to perform well even in limited training data setting. Additionally, the parametrized appproach is flexible to convex spectral functions as regularizers, making it adaptable for the task at hand. The success of manifold constraint for metric learning motivated us to further explore the field of matrix manifolds and their properties to assist in learning and analysis. While Stiefel manifold was explored to develop efficient optimization strategy in Chapter 3, it was later used in deep generative models to impose structure in the latent space that improved the disentanglement of factors of variation in an unsupervised setting. Apart from the Stiefel manifold, the other matrix manifold used in this thesis is the Grassmann manifold that was used in Chapter 7 from analysis perspective. The low dimensional subspace computed from the adversarial sample used for removing the effect of adversarial perturbation was

compared with the subspace obtained from the clean image by analysing them as points on the Grassmann manifold to establish the proximity relation between the two. The approach outperformed state-of-the-art defense strategies across different attacker strengths. Lastly, we used semantic constraints on the statistical manifold to drive the same class distributions closer while pushing the distributions of different classes far. This had shown to improve the semantic consistency of learned representations in supervised settings with limited labeled training data and learning clusterable representations in semi-supervised settings.

The research presented in this thesis provides few possible avenues for future research and exploration. In chapter 6, we parametrized the disentangled representation as a point on the Stiefel manifold i.e. the subset representations are unit normed vectors that are orthogonal to each other. This parametrization, while improved the results, was restricted to unit normed vectors. In the future, it would be interesting to explore mixed curvature representations [60] instead of just positive curvature *i.e.* the Stiefel manifold parametrization. This would have the potential to impose an attribute specific structure on the partitioned representation, further improving the interpretability and control in image generation.

Further, the semi-supervised approach for representation learning in chapter 5 can also be extended to leverage the manifold structure of the latent space representations obtained from the unlabeled data to assist in clustering. This would allow us to get semantic understanding of the data and can lead to better representations without the need of labeled data.

Another direction to explore in future would be the expansion of approach developed for adversarial defense. Chapter 7 developed a low dimensional representation to obtain the clean image from an adversarially perturbed image. While, the method used random image corruptions to define the low dimensional subspace obtained using PCA, it would be interesting to further investigate other dimensionality reduction techniques that can perform at par with PCA and can help overcome the drop in performance on clean sample performance.

Publications

Journal

1. **A.Shukla**, S. Anand, Optimization on Stiefel Manifold for Mahalanobis Distance Metric Learning (Under Review).

Conferences

2. **A. Shukla**, S. Uppal, S. Bhagat , S. Anand, P. Turaga, PrOSe: Product of Orthogonal Sphere Parametrization for Disentangled Representation Learning, BMVC 2019.

3. **A. Shukla**, G.S. Cheema, S. Anand, Q. Qureshi, Y. Jhala, Primate Face Identification in the Wild, PRICAI 2019.

4. **A. Shukla**, S. Uppal, S. Bhagat , S. Anand, P. Turaga, Geometry of Deep Generative Models for Disentangled Representations, ICVGIP 2018.

5. A. Som, K. Thopalli, K.N. Ramamurthy, V. Venkataraman, **A. Shukla**, P. Turaga, Perturbation Robust Representations of Topological Persistence Diagrams, ECCV 2018.

6. **A. Shukla** and S. Anand, Metric Learning Based Automatic Segmentation of Patterned Species, ICIP 2016.

Workshops

7. **A. Shukla**, C. Anderson, G. S. Cheema, P. Guo, S. Onda, D. Anshumaan, S. Anand, R.

Farrell, A Hybrid Approach for Tiger Re-Identification (Challenge Paper), Computer Vision for Wildlife Conservation Workshop, ICCV 2019.

8. D. Kimothi, **A. Shukla**, P. Biyani, S. Anand. and J.M. Hogan, Metric Learning on Biological Sequence Embeddings, IEEE SPAWC 2017.

9. **A. Shukla** and S. Anand, Distance Metric Learning by Optimization on the Stiefel Manifold, DIFF-CV 2015.

Pre-prints

10. **A. Shukla**, P. Turaga, S. Anand, GraCIAS: Grassmannian of Corrupted Images for Adversarial Security, ArXiv preprint arXiv:2005.02936 (2020)

11. **A. Shukla**, G.S. Cheema, S. Anand, Semi-supervised Clustering with Neural Networks, ArXiv preprint arXiv:1806.01547 (2018).

REFERENCES

[1] http://boris-belousov.net/2017/07/11/distance-between-probabilities/.

[2] https://www.jeremyjordan.me/variational-autoencoders/.

[3] https://www.savewildtigers.org/facts/film.

[4] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

[5] C. J. Anderson, S. A. Johnson, M. E. Hostetler, and M. G. Summers. *History and Status of Introduced Rhesus Macaques (Macaca mulatta) in Silver Springs State Park, Florida*. 2016. URL: http://edis.ifas.ufl.edu/uw412 (visited on 09/02/2018).

[6] Georgios Arvanitidis, Lars Kai Hansen, and Soren Hauberg. "Latent Space Oddity: on the Curvature of Deep Generative Models". In: *International Conference on Learning Representations (ICLR)*. 2018.

[7] Anish Athalye, Nicholas Carlini, and David Wagner. "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples". In: *International Conference on Machine Learning*. 2018, pp. 274–283.

[8] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. "Domain adaptation on the statistical manifold". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 2481–2488.

[9] Duhyeon Bang and Hyunjung Shim. "MGGAN: Solving Mode Collapse using Manifold Guided Training". In: *CoRR* abs/1804.04391 (2018).

[10] Yaniv Bar, Idit Diamant, Lior Wolf, and Hayit Greenspan. "Deep learning with non-medical training used for chest pathology identification". In: *Medical Imaging 2015: Computer-Aided Diagnosis*. Vol. 9414. International Society for Optics and Photonics. 2015, p. 94140V.

[11] Horace B Barlow et al. "Possible Principles Underlying the Transformation of Sensory Messages". In: *Sensory Communication* 1 (1961), pp. 217–234.

[12] Jonathan Barzilai and Jonathan M Borwein. "Two-point step size gradient methods". In: *IMA Journal of Numerical Analysis* 8.1 (1988), pp. 141–148.

[13] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. "Semi-supervised Clustering by Seeding". In: *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*. Ed. by Claude Sammut and Achim G. Hoffmann. Morgan Kaufmann, 2002, pp. 27–34.

[14] Aurelien Bellet, Amaury Habrard, and MarcSebban. "Metric Learning". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 9.1 (2015), pp. 1–151.

[15] Aurelien Bellet, Amaury Habrard, and Marc Sebban. "A survey on metric learning for feature vectors and structured data". In: *arXiv preprint arXiv:1306.6709* (2013).

[16] Yoshua Bengio. "Deep learning of representations for unsupervised and transfer learning". In: *Proceedings of ICML workshop on unsupervised and transfer learning*. 2012, pp. 17–36.

[17] Alessandro Bissacco, Alessandro Chiuso, Yi Ma, and Stefano Soatto. "Recognition of human gaits". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 2. IEEE. 2001, pp. II–II.

[18] Douglas T Bolger, Thomas A Morrison, Bennet Vance, Derek Lee, and Hany Farid. "A computer-assisted system for photographic mark–recapture analysis". In: *Methods in Ecology and Evolution* 3.5 (2012), pp. 813–822.

[19] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. "Multi-Level Variational Autoencoder: Learning Disentangled Representations From Grouped Observations". In: *Proceedings of the Thirty-Second Conference on Artificial Intelligence (AAAI)*. 2018, pp. 2095–2102.

[20] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122.

[21] Y. Boykov and G. Funka-Lea. "Graph cuts and efficient ND image segmentation". In: *IJCV* 70.2 (2006), pp. 109–131.

[22] Y. Y Boykov and Marie-Pierre Jolly. "Interactive graph cuts for optimal boundary and region segmentation of objects in ND images". In: *ICCV*. Vol. 1. 2001, pp. 105–112.

[23] Pratik Prabhanjan Brahma, Dapeng Wu, and Yiyuan She. "Why Deep Learning Works: A Manifold Disentanglement Perspective." In: *IEEE Trans. Neural Netw. Learning Syst.* 27.10 (2016), pp. 1997–2008.

[24] Clemens-Alexander Brust, Tilo Burghardt, Milou Groenenberg, Christoph Käding, Hjalmar S Kühl, Marie L Manguette, and Joachim Denzler. "Towards automated visual monitoring of individual gorillas in the wild". In: *CVPR*. 2017, pp. 2820–2830.

[25] Surendranie Judith Cabral, Tharaka Prasad, Thulmini Pubudika Deeyagoda, Sanjaya Nuwan Weerakkody, Ashwika Nadarajah, and Rasanayagam Rudran. "Investigating

Sri Lanka's human-monkey conflict and developing a strategy to mitigate the problem". In: *Journal of Threatened Taxa* 10.3 (2018), pp. 11391–11398.

[26] Nicholas Carlini and David Wagner. "Adversarial examples are not easily detected: Bypassing ten detection methods". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM. 2017, pp. 3–14.

[27] Rudrasis Chakraborty and Baba C. Vemuri. "Recursive Frechet mean computation on the Grassmannian and Its Applications to Computer Vision". In: *IEEE International Conference on Computer Vision, (ICCV)*. 2015, pp. 4229–4237.

[28] Gullal Singh Cheema and Saket Anand. "Automatic Detection and Recognition of Individuals in Patterned Species". In: *ECML PKDD*. 2017. DOI: 10.1007/978-3-319-71273-4{_}3. URL: http://link.springer.com/10.1007/978-3-319-71273-4_3.

[29] T Chehrsimin, T Eerola, M Koivuniemi, M Auttila, R Levänen, M Niemi, M Kunnasranta, and H Kälviäinen. "Automatic individual identification of Saimaa ringed seals". In: *IET Computer Vision* 12.2 (2018), pp. 146–152. ISSN: 1751-9632. DOI: 10.1049/iet-cvi.2017.0082.

[30] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. "Isolating sources of disentanglement in variational autoencoders". In: *Neural Information Processing Systems (NeuRIPS)*. 2018, pp. 2610–2620.

[31] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". In: *International Conference on Neural Information Processing Systems*. 2016.

[32] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-

to-Image Translation". In: *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*. 2018, pp. 8789–8797.

[33] Sumit Chopra, Raia Hadsell, and Yann LeCun. "Learning a Similarity Metric Discriminatively, with Application to Face Verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. IEEE Computer Society, 2005, pp. 539–546. DOI: 10.1109/CVPR.2005.202. URL: https://doi.org/10.1109/CVPR.2005.202.

[34] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. "Certified Adversarial Robustness via Randomized Smoothing". In: *International Conference on Machine Learning*. 2019, pp. 1310–1320.

[35] Maxwell D Collins, Ji Liu, Jia Xu, Lopamudra Mukherjee, and Vikas Singh. "Spectral Clustering with a Convex Regularizer on Millions of Images". In: *in Proc. ECCV*. Springer, 2014, pp. 282–298.

[36] Dorin Comaniciu and Peter Meer. "Mean shift: A robust approach toward feature space analysis". In: *IEEE Transactions on pattern analysis and machine intelligence* 24.5 (2002), pp. 603–619.

[37] Jonathan P Crall, Charles V Stewart, Tanya Y Berger-Wolf, Daniel I Rubenstein, and Siva R Sundaresan. "HotSpotter - Patterned species instance recognition." In: *WACV*. 2013. ISBN: 978-1-4673-5053-2.

[38] David Crouse, Rachel L Jacobs, Zach Richardson, Scott Klum, Anil Jain, Andrea L Baden, and Stacey R Tecot. "LemurFaceID: a face recognition system to facilitate individual identification of lemurs". In: *BMC Zoology* 2.1 (2017), p. 2.

[39] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. "Information-theoretic metric learning". In: *Proc. ICML*. ACM. 2007, pp. 209–216.

[40] Jason V. Davis and Inderjit S. Dhillon. "Structured metric learning for high dimensional problems". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*. Ed. by Ying Li, Bing Liu, and Sunita Sarawagi. ACM, 2008, pp. 195–203. DOI: 10.1145/1401890.1401918. URL: https://doi.org/10.1145/1401890.1401918.

[41] Debayan Deb, Susan Wiper, Alexandra Russo, Sixue Gong, Yichun Shi, Cori Tymoszek, and Anil Jain. "Face Recognition: Primates in the Wild". In: *arXiv preprint arXiv:1804.08790* (2018).

[42] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. "Arcface: Additive angular margin loss for deep face recognition". In: *arXiv preprint arXiv:1801.07698* (2018).

[43] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. "Deep Clustering via Joint Convolutional Autoencoder Embedding and Relative Entropy Minimization". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 5747–5756. DOI: 10.1109/ICCV.2017.612. URL: https://doi.org/10.1109/ICCV.2017.612.

[44] Abhimanyu Dubey, Laurens van der Maaten, Zeki Yalniz, Yixuan Li, and Dhruv Mahajan. "Defense against adversarial images using web-scale nearest-neighbor search". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8767–8776.

[45] James Duyck, Chelsea Finn, Andy Hutcheon, Pablo Vera, Joaquin Salas, and Sai Ravela. "Sloop: A pattern retrieval engine for individual animal identification". In: *Pattern Recognition* 48.4 (2015), pp. 1059–1073. ISSN: 0031-3203. DOI: http://dx.doi.org/10.1016/j.patcog.2014.07.017.

[46] Alan Edelman, Tomás A Arias, and Steven T Smith. "The Geometry of Algorithms with Orthogonality Constraints". In: *SIAM journal on Matrix Analysis and Applications* 20.2 (1998), pp. 303–353.

[47] Istvan Fehervari, Avinash Ravichandran, and Srikar Appalaraju. "Unbiased evaluation of deep metric learning algorithms". In: *arXiv preprint arXiv:1911.12528* (2019).

[48] Sanja Fidler, Sven Dickinson, and Raquel Urtasun. "3d object detection and viewpoint estimation with a deformable 3d cuboid model". In: *Advances in Neural Information Processing Systems*. 2012, pp. 611–619.

[49] Sharon Fogel, Hadar Averbuch-Elor, Jacov Goldberger, and Daniel Cohen-Or. "Clustering-driven Deep Embedding with Pairwise Constraints". In: *arXiv preprint arXiv:1803.08457* (2018).

[50] Alexander Freytag, Erik Rodner, Marcel Simon, Alexander Loos, Hjalmar S Kühl, and Joachim Denzler. "Chimpanzee faces in the wild: Log-euclidean cnns for predicting identities and attributes of primates". In: *German Conference on Pattern Recognition*. Springer. 2016, pp. 51–63.

[51] Jean Gallier. "The Quaternions and the Spaces S^3, $SU(2)$, $SO(3)$, and $\mathbb{RP}^3$". In: *Geometric Methods and Applications*. Springer, 2001, pp. 248–266.

[52] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. "Image Style Transfer Using Convolutional Neural Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2414–2423.

[53] Athinodoros S Georghiades, Peter N Belhumeur, and David J Kriegman. "From few to many: Illumination cone models for face recognition under variable lighting and pose". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 23.6 (2001), pp. 643–660.

[54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[55] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015. URL: http://arxiv.org/abs/1412.6572.

[56] Zahra Goudarzi, Peyman Adibi, Rolf-Rainer Grigat, and Mohammad Saeid Ehsani. "Making metric learning algorithms invariant to transformations using a projection metric on Grassmann manifolds". In: *International Journal of Machine Learning and Cybernetics* 10.12 (2019), pp. 3407–3416.

[57] Leo Grady. "Random Walks for Image Segmentation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 28.11 (2006), pp. 1768–1783. DOI: 10.1109/TPAMI.2006.233. URL: https://doi.org/10.1109/TPAMI.2006.233.

[58] Yves Grandvalet and Yoshua Bengio. "Entropy Regularization". In: *Semi-Supervised Learning*. Ed. by Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. MIT Press, 2006, pp. 151–168. URL: http://www.iro.umontreal.ca/~lisa/pointeurs/entropy_regularization_2006.pdf.

[59] Douglas Gray, Shane Brennan, and Hai Tao. "Evaluating appearance models for recognition, reacquisition, and tracking". In: *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*. Vol. 3. 5. 2007.

[60] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. "Learning Mixed-Curvature Representations in Product Spaces". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=HJxeWnCcF7.

[61] Sun Ji-guang. "Perturbation of angles between linear subspaces". In: *Journal of Computational Mathematics* (1987), pp. 58–61.

[62] Matthieu Guillaumin, Thomas Mensink, Jakob J. Verbeek, and Cordelia Schmid. "TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation". In: *IEEE 12th International Conference on Computer Vision, ICCV 2009, Kyoto, Japan, September 27 - October 4, 2009*. IEEE Computer Society, 2009, pp. 309–316. DOI: 10.1109/ICCV.2009.5459266. URL: https://doi.org/10.1109/ICCV.2009.5459266.

[63] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. "Countering Adversarial Images using Input Transformations". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=SyJ7ClWCb.

[64] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. "Improved deep embedded clustering with local structure preservation". In: *International Joint Conference on Artificial Intelligence (IJCAI-17)*. 2017, pp. 1753–1759.

[65] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. "Deep Clustering with Convolutional Autoencoders". In: *International Conference on Neural Information Processing*. Springer. 2017, pp. 373–382.

[66] Per Christian Hansen, James G. Nagy, Dianne P. O'Leary, and Rodney Miller. "*Deblurring Images: Matrices, Spectra and Filtering*". In: *J. Electronic Imaging* 17.1 (2008), p. 019901. DOI: 10.1117/1.2900557. URL: https://doi.org/10.1117/1.2900557.

[67] Robert M Haralick, Stanley R Sternberg, and Xinhua Zhuang. "Image analysis using mathematical morphology". In: *IEEE transactions on pattern analysis and machine intelligence* 4 (1987), pp. 532–550.

[68] Mehrtash Harandi, Mathieu Salzmann, and Richard Hartley. "Joint Dimensionality Reduction and Metric Learning: A Geometric Take". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, June 2017, pp. 1404–1413. URL: http://proceedings.mlr.press/v70/harandi17a.html.

[69] John A Hartigan and Manchek A Wong. "Algorithm AS 136: A k-means clustering algorithm". In: *Journal of the royal statistical society. series c (applied statistics)* 28.1 (1979), pp. 100–108.

[70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[71] Lex Hiby, Phil Lovell, Narendra Patil, N Samba Kumar, Arjun M Gopalaswamy, and K Ullas Karanth. "A tiger cannot change its stripes: using a three-dimensional model to match images of living tigers and tiger skins". In: *Biology letters* 5.3 (2009), pp. 383–386.

[72] Irina Higgins, Loic Matthey, Xavier Glorot, Arka Pal, Benigno Uria, Charles Blundell, Shakir Mohamed, and Alexander Lerchner. "Early visual concept learning with unsupervised deep learning". In: *arXiv preprint arXiv:1606.05579* (2016).

[73] IV Higgins and SM Stringer. "The role of independent motion in object segmentation in the ventral visual stream: Learning to recognise the separate parts of the body". In: *Vision Research* 51.6 (2011), pp. 553–562.

[74] Geoffrey E Hinton and Ruslan R Salakhutdinov. "Reducing the dimensionality of data with neural networks". In: *science* 313.5786 (2006), pp. 504–507.

[75] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[76] Yen-Chang Hsu and Zsolt Kira. "Neural network-based clustering using pairwise constraints". In: *arXiv preprint arXiv:1511.06321* (2015).

[77] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. "Discriminative deep metric learning for face verification in the wild". In: *CVPR*. 2014, pp. 1875–1882.

[78] Qiyang Hu, Attila Szabó, Tiziano Portenier, Paolo Favaro, and Matthias Zwicker. "Disentangling factors of variation by mixing them". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 3399–3407.

[79] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. "Densenet: Implementing efficient convnet descriptor pyramids". In: *arXiv preprint arXiv:1404.1869* (2014).

[80] Prateek Jain, Brian Kulis, Jason V Davis, and Inderjit S Dhillon. "Metric and kernel learning using a linear transformation". In: *JMLR* 13.1 (2012), pp. 519–547.

[81] Ian H. Jermyn, Sebastian Kurtek, Eric Klassen, and Anuj Srivastava. "Elastic Shape Matching of Parameterized Surfaces Using Square Root Normal Fields". In: *European Conference on Computer Vision (ECCV)*. 2012, pp. 804–817.

[82] Ananya Harsh Jha, Saket Anand, Maneesh Singh, and V. S. R. Veeravasarapu. "Disentangling Factors of Variation with Cycle-Consistent Variational Auto-encoders". In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Vol. 11207. Lecture Notes in Computer Sci-

ence. Springer, 2018, pp. 829–845. DOI: 10.1007/978-3-030-01219-9_49. URL: https://doi.org/10.1007/978-3-030-01219-9%5C_49.

[83] Bo Jiang and Yu-Hong Dai. "A Framework of Constraint Preserving Uupdate Schemes for Optimization on Stiefel manifold". In: *Mathematical Programming* 153.2 (2015), pp. 535–575.

[84] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. "Disentangled Representation Learning for Text Style Transfer". In: *arXiv preprint arXiv:1808.04339* (2018).

[85] Robert E Kass. "The geometry of asymptotic inference". In: *Statistical Science* (1989), pp. 188–219.

[86] Hyunjik Kim and Andriy Mnih. "Disentangling by Factorising". In: *International Conference on Machine Learning (ICML)*. 2018, pp. 2654–2663.

[87] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. "Semi-supervised learning with deep generative models". In: *NIPS*. 2014, pp. 3581–3589.

[88] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[89] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. "Large scale metric learning from equivalence constraints". In: *CVPR*. IEEE. 2012, pp. 2288–2295.

[90] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *NIPS*. 2012.

[91] Line Kuhnel, Tom Fletcher, Sarang Joshi, and Stefan Sommer. "Latent Space Non-Linear Statistics". In: *arXiv preprint arXiv:1805.07632* (2018).

[92] Line Kühnel, Tom Fletcher, Sarang C. Joshi, and Stefan Sommer. "Latent Space Non-Linear Statistics". In: *CoRR* abs/1805.07632 (2018).

[93] Brian Kulis. "Metric learning: A survey". In: *Foundations and Trends in Machine Learning* 5.4 (2012), pp. 287–364.

[94] Brian Kulis, Matyas A Sustik, and Inderjit S Dhillon. "Low-rank kernel learning with Bregman matrix divergences". In: *JMLR* 10 (2009), pp. 341–376.

[95] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. "Adversarial Machine Learning at Scale". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=BJm4T4Kgx.

[96] M. Lahiri, C. Tantipathananandh, R. Warungu, D. I. Rubenstein, and T. Y. Berger-Wolf. "Biometric animal databases from field photographs: Identification of individual zebra in the wild". In: *1st ACM ICMR*. 2011, p. 6.

[97] Samuli Laine. *Feature-Based Metrics for Exploring the Latent Space of Generative Models*. 2018. URL: https://openreview.net/forum?id=BJslDBkwG.

[98] M.T. Law, N. Thome, and M. Cord. "Fantope Regularization in Metric Learning". In: *Proc. CVPR*. June 2014, pp. 1051–1058.

[99] Marc T Law, Raquel Urtasun, and Richard S Zemel. "Deep spectral clustering learning". In: *ICML*. 2017, pp. 1985–1994.

[100] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[101] Dong-Hyun Lee. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks". In: *Workshop on Challenges in Representation Learning, ICML*. Vol. 3. 2013, p. 2.

[102] Chenxi Li, Zelin Shi, Yunpeng Liu, and Baoshu Xu. "Grassmann manifold based shape matching and retrieval under partial occlusions". In: *International Symposium on Optoelectronic Technology and Application 2014: Image Processing and Pattern Recognition*. Vol. 9301. International Society for Optics and Photonics. 2014, 93012O.

[103] Qunwei Li, Bhavya Kailkhura, Rushil Anirudh, Yi Zhou, Yingbin Liang, and Pramod K. Varshney. "MR-GAN: Manifold Regularized Generative Adversarial Networks". In: *CoRR* abs/1811.10427 (2018).

[104] Yikang Li, Chris Twigg, Yuting Ye, Lingling Tao, and Xiaogang Wang. "Disentangling Pose from Appearance in Monochrome Hand Images". In: *arXiv preprint arXiv:1904.07528* (2019).

[105] Yingzhen Li and Stephan Mandt. "Disentangled Sequential Autoencoder". In: *International Conference on Machine Learning (ICML)*. 2018, pp. 5656–5665.

[106] Jianqing Liang, Qinghua Hu, Pengfei Zhu, and Wenwu Wang. "Efficient multi-modal geometric mean metric learning". In: *Pattern Recognition* 75 (2018), pp. 188–198.

[107] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. "Person re-identification by local maximal occurrence representation and metric learning". In: *CVPR*. 2015, pp. 2197–2206.

[108] Hao Liu, Jiwen Lu, Jianjiang Feng, and Jie Zhou. "Label-sensitive deep metric learning for facial age estimation". In: *IEEE Transactions on Information Forensics and Security* 13.2 (2018), pp. 292–305.

[109] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. "Few-shot unsupervised image-to-image translation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 10551–10560.

[110] Songtao Liu, Di Huang, and Yunhong Wang. "Receptive Field Block Net for Accurate and Fast Object Detection". In: *European Conference on Computer Vision*. Springer. 2018, pp. 404–419.

[111] Wei Liu, Cun Mu, Rongrong Ji, Shiqian Ma, John R. Smith, and Shih-Fu Chang. "Low-Rank Similarity Metric Learning in High Dimensions". In: *Proc. AAAI*. Austin, Texas, USA, 2015.

[112] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. "Sphereface: Deep hypersphere embedding for face recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 212–220.

[113] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. "Feature Distillation: DNN-Oriented JPEG Compression Against Adversarial Examples". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 860–868.

[114] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 3730–3738.

[115] Francesco Locatello, Gabriele Abbati, Tom Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. "On the Fairness of Disentangled Representations". In: *arXiv preprint arXiv:1905.13662* (2019).

[116] Suhas Lohit and Pavan Turaga. "Learning Invariant Riemannian Geometric Representations Using Deep Nets". In: *ICCV Workshop on Manifold Learning: From Euclid to Riemann*. 2017, pp. 1329–1338.

[117] Alexander Loos and Andreas Ernst. "Detection and identification of chimpanzee faces in the wild". In: *Multimedia (ISM), 2012 IEEE International Symposium on*. IEEE. 2012, pp. 116–119.

[118] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[119] Chen Change Loy, Tao Xiang, and Shaogang Gong. "Multi-camera activity correlation analysis". In: *CVPR*. IEEE. 2009, pp. 1988–1995.

[120] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=rJzIBfZAb.

[121] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. "Event-Based Vision Meets Deep Learning on Steering Prediction for Self-Driving Cars". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[122] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. "Disentangling factors of variation in deep representation using adversarial training". In: *Advances in Neural Information Processing Systems*. 2016, pp. 5040–5048.

[123] Tetsu Matsukawa and Einoshin Suzuki. "Person re-identification using CNN features learned from combination of attributes". In: *2016 23rd international conference on pattern recognition (ICPR)*. IEEE. 2016, pp. 2428–2433.

[124] Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *International Conference on Learning Representations (ICLR)*. 2017.

[125] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2574–2582.

[126] Yadong Mu. "Fixed-Rank Supervised Metric Learning on Riemannian Manifold". In: *Proc. AAAI*. 2016.

[127] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. *A Metric Learning Reality Check*. 2020 (Accepted to ECCV 2020). arXiv: 2003.08505 [cs.CV].

[128] Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. "Adversarial defense by restricting the hidden space of deep neural networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 3385–3394.

[129] Ekaterina Nepovinnykh, Tuomas Eerola, Heikki Kälviäinen, and Gleb Radchenko. "Identification of Saimaa Ringed Seal Individuals Using Transfer Learning". In: *Advanced Concepts for Intelligent Vision Systems*. Ed. by Jacques Blanc-Talon, David Helbert, Wilfried Philips, Dan Popescu, and Paul Scheunders. Cham: Springer International Publishing, 2018, pp. 211–222. ISBN: 978-3-030-01449-0.

[130] Bac Nguyen and Bernard De Baets. "Kernel-Based Distance Metric Learning for Supervised k-Means Clustering". In: *IEEE Transactions on Neural Networks and Learning Systems* 30.10 (2019), pp. 3084–3095.

[131] Philip J. Nyhus. "Human–Wildlife Conflict and Coexistence". In: *Annual Review of Environment and Resources* 41.1 (2016), pp. 143–171.

[132] Vic Patrangenaru and Kanti V Mardia. "Affine shape analysis and image analysis". In: *22nd Leeds Annual Statistics Research Workshop*. 2003, pp. 57–62.

[133] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. "Deflecting adversarial attacks with pixel deflection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8571–8580.

[134] Ori Press, Tomer Galanti, Sagie Benaim, and Lior Wolf. "Emerging Disentanglement in Auto-Encoder Based Unsupervised Image Content Transfer". In: *International Conference on Learning Representations (ICLR)*. 2019.

[135] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. "Adversarial robustness through local linearization". In: *Advances in Neural Information Processing Systems*. 2019, pp. 13824–13833.

[136] Daniel Quang, Yifei Chen, and Xiaohui Xie. "DANN: a deep learning approach for annotating the pathogenicity of genetic variants". In: *Bioinformatics* 31.5 (2014), pp. 761–763.

[137] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[138] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. "Barrage of Random Transforms for Adversarially Robust Defense". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6528–6537.

[139] C Radhakrishna Rao. "Information and the accuracy attainable in the estimation of statistical parameters". In: *Breakthroughs in statistics*. Springer, 1992, pp. 235–247.

[140] Qing Rao and Jelena Frtunikj. "Deep learning for self-driving cars: chances and challenges". In: *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*. 2018, pp. 35–38.

[141] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. "Semi-supervised learning with ladder networks". In: *Advances in Neural Information Processing Systems*. 2015, pp. 3546–3554.

[142] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection". In: *CVPR*. 2016, pp. 779–788.

[143] Scott E. Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. "Deep Visual Analogy-Making". In: *Advances in Neural Information Processing Systems (NIPS)*. 2015, pp. 1252–1260.

[144] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *NIPS*. 2015, pp. 91–99.

[145] C Rother, V. Kolmogorov, and A Blake. "Grabcut: Interactive foreground extraction using iterated graph cuts". In: *ACM Transactions on Graphics (TOG)* 23.3 (2004), pp. 309–314.

[146] Adrià Ruiz, Oriol Martinez, Xavier Binefa, and Jakob Verbeek. "Learning disentangled representations with reference-based variational autoencoders". In: *arXiv preprint arXiv:1901.08534* (2019).

[147] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y. URL: https://doi.org/10.1007/s11263-015-0816-y.

[148] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. "Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=BkJ3ibb0-.

[149] Raghav Saraswat, Anindya Sinha, and Sindhu Radhakrishna. "A god becomes a pest? Human-rhesus macaque interactions in Himachal Pradesh, northern India". In: *European Journal of Wildlife Research* 61.3 (June 2015), pp. 435–443.

[150] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[151] Hang Shao, Abhishek Kumar, and P Thomas Fletcher. "The Riemannian Geometry of Deep Generative Models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 315–323.

[152] Yuan Shi, Aurelien Bellet, and Fei Sha. "Sparse Compositional Metric Learning". In: *Proc. AAAI*. 2014.

[153] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Güler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. "Deforming Autoencoders: Unsupervised Disentangling of Shape and Appearance". In: *European Conference on Computer Vision (ECCV), Part X*. 2018, pp. 664–680.

[154] Ankita Shukla, Shagun Uppal, Sarthak Bhagat, Saket Anand, and Pavan K. Turaga. "Geometry of Deep Generative Models for Disentangled Representations". In: *Indian Conference on Computer Vision, Graphics, and Image Processing (ICVGIP)*. 2018.

[155] Terence Sim, Simon Baker, and Maan Bsat. "The CMU pose, illumination, and expression (PIE) database". In: *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE. 2002, pp. 53–58.

[156] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*. 2015.

[157] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. "Deep Metric Learning via Facility Location". In: *2017 IEEE Conference on Computer Vision*

and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society, 2017, pp. 2206–2214. DOI: 10.1109/CVPR.2017.237. URL: https://doi.org/10.1109/CVPR.2017.237.

[158] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. "Deep Metric Learning via Lifted Structured Feature Embedding". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 4004–4012. DOI: 10.1109/CVPR.2016.434. URL: https://doi.org/10.1109/CVPR.2016.434.

[159] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. "PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=rJUYGxbCW.

[160] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. "Deep high-resolution representation learning for human pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5693–5703.

[161] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. "Deep learning face representation by joint identification-verification". In: *Advances in neural information processing systems*. 2014, pp. 1988–1996.

[162] Attila Szabó, Qiyang Hu, Tiziano Portenier, Matthias Zwicker, and Paolo Favaro. "Challenges in Disentangling Independent Factors of Variation". In: *6th International Conference on Learning Representations (ICLR), Workshop Track Proceedings*. 2018.

[163] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308. URL: https://doi.org/10.1109/CVPR.2016.308.

[164] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).

[165] Saeid Asgari Taghanaki, Kumar Abhishek, Shekoofeh Azizi, and Ghassan Hamarneh. "A kernelized manifold mapping to diminish the effect of adversarial perturbations". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11340–11349.

[166] Valentin Thomas, Emmanuel Bengio, William Fedus, Jules Pondard, Philippe Beaudoin, Hugo Larochelle, Joelle Pineau, Doina Precup, and Yoshua Bengio. "Disentangling the independently controllable factors of variation by interacting with the world". In: *Workshop in NeurIPS* (2017).

[167] Hoang Tran Vu and Ching-Chun Huang. "Domain Adaptation Meets Disentangled Representation Learning and Style Transfer". In: *arXiv preprint arXiv:1712.09025* (2017).

[168] Ashok Veeraraghavan, Amit K Roy-Chowdhury, and Rama Chellappa. "Matching shape sequences in video with applications in human movement analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.12 (2005), pp. 1896–1909.

[169] Hoang Tran Vu and Ching-Chun Huang. "Domain adaptation meets disentangled representation learning and style transfer". In: *arXiv preprint arXiv:1712.09025* (2017).

[170] F. Wang, J. Cheng, W. Liu, and H. Liu. "Additive Margin Softmax for Face Verification". In: *IEEE Signal Processing Letters* 25.7 (July 2018), pp. 926–930.

[171] K.Q. Weinberger and L.K. Saul. "Distance metric learning for large margin nearest neighbor classification". In: *JMLR* 10 (2009), pp. 207–244.

[172] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. "A discriminative feature learning approach for deep face recognition". In: *European Conference on Computer Vision*. Springer. 2016, pp. 499–515.

[173] Zaiwen Wen and Wotao Yin. "A feasible method for optimization with orthogonality constraints". In: *Mathematical Programming* 142.1-2 (2013), pp. 397–434.

[174] Claire L Witham. "Automated face recognition of rhesus macaques". In: *Journal of Neuroscience Methods* 300 (2018), pp. 157–165.

[175] Lior Wolf, Tal Hassner, and Itay Maoz. "Face recognition in unconstrained videos with matched background similarity". In: *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*. IEEE Computer Society, 2011, pp. 529–534. DOI: 10.1109/CVPR.2011.5995566. URL: https://doi.org/10.1109/CVPR.2011.5995566.

[176] John Wright, Allen Y. Yang, Arvind Ganesh, Shankar S. Sastry, and Yi Ma. "Robust Face Recognition via Sparse Representation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 31.2 (2009), pp. 210–227. DOI: 10.1109/TPAMI.2008.79. URL: https://doi.org/10.1109/TPAMI.2008.79.

[177] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. "Feature denoising for improving adversarial robustness". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 501–509.

[178] Junyuan Xie, Ross Girshick, and Ali Farhadi. "Unsupervised deep embedding for clustering analysis". In: *International conference on machine learning*. 2016, pp. 478–487.

[179] Eric P. Xing, Michael I. Jordan, Stuart J Russell, and Andrew Y. Ng. "Distance Metric Learning with Application to Clustering with Side-Information". In: *NIPS*. 2002, pp. 521–528.

[180] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. "End-to-end learning of driving models from large-scale video datasets". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2174–2182.

[181] Y. Xu and A. K. Roy-Chowdhury. "Integrating Motion, Illumination, and Structure in Video Sequences with Applications in Illumination-Invariant Tracking". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 29 (May 2007), pp. 793–806. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.1047. URL: doi.ieeecomputersociety.org/10.1109/TPAMI.2007.1047.

[182] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. "Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering". In: *International Conference on Machine Learning*. 2017, pp. 3861–3870.

[183] Jianwei Yang, Devi Parikh, and Dhruv Batra. "Joint unsupervised learning of deep representations and image clusters". In: *CVPR*. 2016, pp. 5147–5156.

[184] Liu Yang and Rong Jin. "Distance metric learning: A comprehensive survey". In: (2006).

[185] Gui-Bo Ye, Yifei Chen, and Xiaohui Xie. "Efficient variable selection in support vector machines via the alternating direction method of multipliers". In: *International Conference on Artificial Intelligence and Statistics*. 2011, pp. 832–840.

[186] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. "Deep metric learning for person re-identification". In: *ICPR*. IEEE. 2014, pp. 34–39.

[187] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. "Learning face representation from scratch". In: *arXiv preprint arXiv:1411.7923* (2014).

[188] Pourya Zadeh, Reshad Hosseini, and Suvrit Sra. "Geometric Mean Metric Learning". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. Ed. by Maria-Florina Balcan

and Kilian Q. Weinberger. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 2464–2471. URL: http://proceedings.mlr.press/v48/zadeh16.html.

[189] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning requires rethinking generalization". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=Sy8gdB9xx.

[190] Junbo Zhao, Michael Mathieu, Ross Goroshin, and Yann Lecun. "Stacked what-where auto-encoders". In: *arXiv preprint arXiv:1506.02351* (2015).

[191] Artem Zhelezniakov, Tuomas Eerola, Meeri Koivuniemi, Miina Auttila, Riikka Levänen, Marja Niemi, Mervi Kunnasranta, and Heikki Kälviäinen. "Segmentation of Saimaa Ringed Seals for Identification Purposes". In: *Advances in Visual Computing*. Cham: Springer International Publishing, 2015, pp. 227–236. ISBN: 978-3-319-27863-6.

[192] Guoqiang Zhong, Kaizhu Huang, and Cheng-Lin Liu. "Low Rank Metric Learning with Manifold Regularization". In: *ICDM*. 2011, pp. 1266–1271.

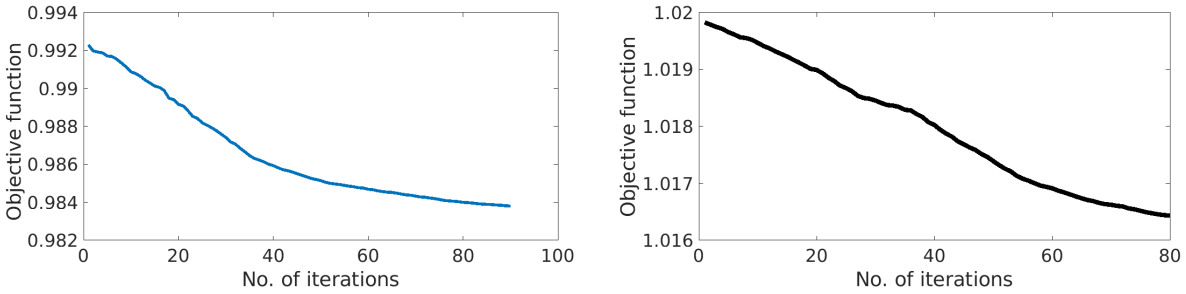
APPENDIX

Appendix A

Convergence Proof and Experiments in Chapter 3

A.1 Convergence Plots

EigMetric restores to an alternating optimization strategy that jointly optimizes on the Stiefel Manifold and the positive orthant. We derive the convergence proof for our Algorithm 2 in this section. We additionally support our proof with empirical evidence of convergence for USPS and segments datasets in Figure A.2 and VIPeR dataset in Figure A.1 that show monotonic decrease in the objective function.



(a) Small Training Data **(b)** Large Training Data

Figure A.1 Objective function plot over iterations for EigMetric framework for VIPeR dataset in small training data (10 %) and large training data (60 % data) settings to show the convergence.

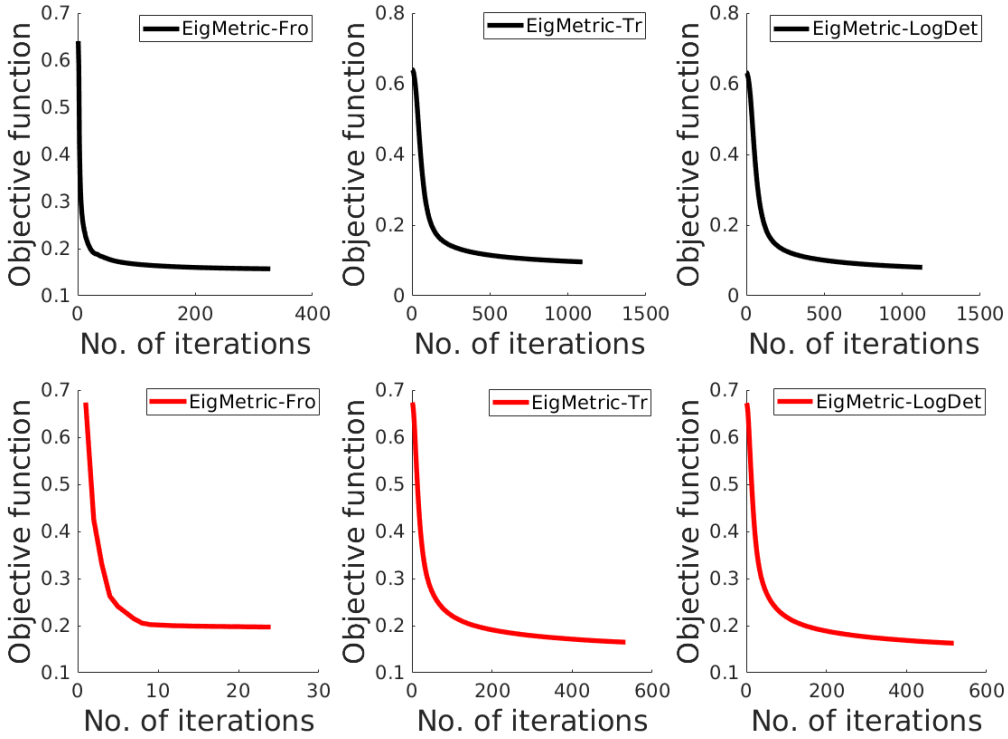


Figure A.2 Objective function plots of EigMetric with different regularizers for USPS (first row) and Segment dataset (second row) to show the convergence of the proposed algorithm.

A.2 Proof of Convergence

Our optimization strategy alternates between the convex subproblem for $\mathbf{w}$ and the manifold constrained subproblem for $\mathbf{U}$ that is shown to monotonically reduce the objective function value over the iterations. We present Theorem 1 that develops on the theoretical convergence of ADMM strategy to establish that the proposed strategy leads to a minima of the objective function.

Theorem 1. *For $\alpha > 0$ and any convex spectral regularizer, the sequence $\{\mathbf{w}^t, \mathbf{U}^t\}$ generated by Algorithm 1 monotonically decreases the objective function starting with initial $\mathbf{w}^0 = \mathbf{1}$, $\mathbf{U}^0 = \mathbf{U}_{\mathcal{X}}$. Here $\mathbf{U}_{\mathcal{X}}$ is the matrix of eigenvectors extracted from the training data.*

Proof. The objective function for our metric learning formulation is given in 3.13. The optimization strategy comprises of ADMM updates to solve the $\mathbf{w}$ subproblem and gradient

updates to solve for **U** subproblem.

For a fixed U: We first show that the optimization strategy converges to a solution that minimizes the problem in (3.20), as it reduces to standard ADMM strategy.

We rewrite the subproblem for $\mathbf{w}$ with the dummy variable vector $\mathbf{c}$ and a regularizer $\mathcal{R}(\mathbf{w})$ to account for spectral functions as follows

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^p_+,\mathbf{c}\in\mathbb{R}^{|\mathcal{T}|}} \quad & \frac{1}{|\mathcal{T}|}\sum_{k=1}^{|\mathcal{T}|}\left[\mathbf{c}_k\right]_+ + \alpha\mathcal{R}(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{c} = \mathbf{1} + \mathbf{S}^\top\mathbf{w} - \mathbf{D}^\top\mathbf{w} \end{aligned} \tag{A.1}$$

Here, we replace, $g(\mathbf{c}) = \sum_{k=1}^{|\mathcal{T}|}\left[\mathbf{c}_k\right]_+$, $h(\mathbf{w}) = \mathcal{R}(\mathbf{w})$ for simplicity and show that the problem in (A.1), satisfies the two conditions required for ADMM convergence as given by [20].

Condition 1: Both the functions $g(\mathbf{c})$ and $h(\mathbf{w})$ are convex, proper and closed.

Condition 2: The function $\mathcal{L}_\rho(\widehat{\mathbf{c}},\widehat{\mathbf{w}},\widehat{\lambda})$ has a saddle point.

The condition 1 holds for $g(\mathbf{c})$. For $h(\mathbf{w})$, in our case, since variable $\mathbf{w}$ denotes the eigenvalues of a PSD matrix that are non-negative *i.e.* $\mathbf{w}\in\mathbb{R}^p_+$ and $h(\mathbf{w})$ is restricted to convex spectral functions, conditions 1 also holds for $h(\mathbf{w})$ as well.

For condition 2, we write the augmented Lagrangian as

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{c},\mathbf{w},\lambda) = \frac{1}{|\mathcal{T}|}\sum_{k=1}^{|\mathcal{T}|}\left[\mathbf{c}_k\right]_+ + \alpha\mathcal{R}(\mathbf{w}) + \langle\lambda, \mathbf{c} - \mathbf{1} - \mathbf{S}^\top\mathbf{w} + \mathbf{D}^\top\mathbf{w}\rangle \\ + \frac{\rho}{2}\left\|\mathbf{c} - \mathbf{1} - \mathbf{S}^\top\mathbf{w} + \mathbf{D}^\top\mathbf{w}\right\|_F^2 \end{aligned} \tag{A.2}$$

The domains of the variables $\mathbf{c}_i$,$\mathbf{w}$ and λ_i are convex sets given by $\mathbb{R}$, $\mathbb{R}^p_+$ and $\mathbb{R}_+$ respectively. So, for the fixed λ_i's, the $\mathcal{L}_\rho(\mathbf{c},\mathbf{w},\lambda)$ is convex in $\mathbf{c}$ and $\mathbf{w}$. Therefore, there exists $\{\widehat{\mathbf{c}},\widehat{\mathbf{w}},\widehat{\lambda}\}$ such that

$$\mathcal{L}_\rho(\widehat{\mathbf{c}},\widehat{\mathbf{w}},\widehat{\lambda}) \leq \mathcal{L}_\rho(\mathbf{c},\mathbf{w},\widehat{\lambda}) \tag{A.3}$$

Similarly, for a fixed $(\mathbf{c},\mathbf{w})$, $\mathcal{L}_\rho(\mathbf{c},\mathbf{w},\lambda)$ is a concave function of λ_i's. Hence, there exists $\{\widehat{\mathbf{c}},\widehat{\mathbf{w}},\widehat{\lambda}\}$ such that

$$\mathcal{L}_\rho(\widehat{\mathbf{c}}, \widehat{\mathbf{w}}, \widehat{\lambda}) \geq \mathcal{L}_\rho(\widehat{\mathbf{c}}, \mathbf{w}, \widehat{\lambda}) \tag{A.4}$$

This proves that the $\mathcal{L}_\rho(\mathbf{w}, \mathbf{c}, \lambda)$ has a saddle point.

This implies that there exists a $\widehat{\mathbf{w}}$, that minimizes the objective function in (3.13) .

With U updates: Now, to prove the convergence of the objective function in (3.13), the $\mathbf{U}$ updates are also considered in addition to the $\mathbf{w}$ updates with ADMM strategy. Since at every iteration t, $\mathbf{U}$ is updated with a [173] Crank-Nicholson update scheme that (1) ensures the orthogonality of $\mathbf{U}$ at every iteration and (2) employ a non-monotone line search approach to determine the appropriate step size in the direction of descent. This implies that with every iteration t, there exists a $\mathbf{U}^t$ that satisfies the following condition

$$\mathcal{J}(\mathbf{w}, \mathbf{U}^t) \leq \mathcal{J}(\mathbf{w}, \mathbf{U}^{(t-1)}) \tag{A.5}$$

So, with the domain defined as the Stiefel manifold, the update for $\mathbf{U}$ essentially leads to further decrease in the objective function in addition to the decrease achieved by ADMM updates for $\mathbf{w}$ subproblem. Thus, the convergence properties of ADMM still hold with $\mathbf{U}$ updates and leads to the conclusion the sequence of updates monotonically decreases the objection function. Furthermore, a practical solution $(\widehat{\mathbf{w}}, \widehat{\mathbf{U}})$ is obtained when both objective function and $\mathbf{M}$ stop changing in l_2 sense from the last iteration.

A.3 Algorithms

Algorithm 2 Joint Optimization over $\mathcal{S}_{n,p} \times \mathbb{R}_+^p$

Input: $\mathcal{X} \in \mathbb{R}^{n \times m}$:Training data matrix, $\mathcal{T}$: Triplet constraints matrix, α: regularization parameter, q : no of rows, $\rho \geq 0$

Output: $\mathbf{M}$: Learned Distance Matrix

1. Initialize $\mathbf{U}$ and $\mathbf{w}$

   p: dimension after PCA containing 95% of the energy of $\mathcal{X}$.

   $\mathbf{w}^0$: $\mathbf{1}_{p \times 1}$ (vector of ones)

   $\mathbf{U}^0$: $n \times p$ orthonormal matrix corresponding to r eigenvectors of training data

2. Initialize $\mathbf{c}$ and λ

   $\mathbf{c}^0 \leftarrow \mathbf{0}_{|\mathcal{T}| \times 1}$, $\lambda^0 \leftarrow \mathbf{0}_{|\mathcal{T}| \times 1}$ (vectors of all zeros)

3. **repeat** for $t = 1, \ldots T$ do

   (a) Compute $\mathbf{S}$ and $\mathbf{D}$ using $\mathbf{U}^{t-1}$

   (b) Update dummy variable $\mathbf{c}^t$ using (3.22)

   (c) Update $\mathbf{w}^t$ by solving least squares in (3.23)

   (d) Update $\lambda^t = \lambda^{t-1} + \rho(\mathbf{c}^t - \mathbf{1} - \mathbf{S}^\top \mathbf{w}^t + \mathbf{D}^\top \mathbf{w}^t)$

   (e) Update $\mathbf{U}^t \in S_{n,p}$

      i. Randomly select q row indices from n rows

      ii. Compute $\mathbf{H}$ and $\mathbf{T}$ for parametrization on $\mathcal{S}_{q,r}$

      iii. Update $\mathbf{Q}$ according to (3.28)

      iv. Update $\mathbf{U}^t$ with $\mathbf{Q}$ as in (3.29)

4. **until** convergence

return $\widehat{\mathbf{M}} = \widehat{\mathbf{U}}\ \mathrm{Diag}(\widehat{\mathbf{w}})\widehat{\mathbf{U}}^\top$

Algorithm 3 Alternating Gradient Method: Joint Optimization over $\mathcal{S}_{n,p} \times \mathbb{R}^p_+$ with alternating gradients

Input: $\mathcal{X} \in \mathbb{R}^{n \times m}$:Training data matrix, $\mathcal{T}$: Triplet constraints matrix, α: regularization parameter, q : no of rows, $\rho \geq 0$

Output: $\mathbf{M}$: Learned Distance Matrix

1. Compute $\mathcal{T}_0$ (active constraints)
2. Update $\mathbf{w}$ with $\mathcal{T}_0$ constraints to obtain $\mathbf{w}^t$

   by computing the gradient update from the following

   $\frac{1}{|\mathcal{T}|} \sum_{m=1}^{|\mathcal{T}_0|} \left[1 + \mathbf{s}_m^\top \mathbf{w} - \mathbf{d}_m^\top \mathbf{w}\right]_+ + \alpha \left\|\mathbf{w}\right\|_2^2$
3. Update $\mathbf{U}^t$

   Follow the steps (e) in Algorithm 2
4. **until** convergence

return $\widehat{\mathbf{M}} = \widehat{\mathbf{U}}\ \mathrm{Diag}(\widehat{\mathbf{w}})\widehat{\mathbf{U}}^\top$

Appendix B

Algorithms Required for Chapter 3 and 5

B.1 Constrained K-Means

k-means [69] is a popular unsupervised approach for clustering data. Constrained K-Means [13] is a modification that uses some amount of labeled data along with the the unlabeled data to improve the clusterability. It utilizes some labeled data to guide the unsupervised k-means clustering. As opposed to random initialization of cluster centers in traditional k-means, labeled samples are used to initialize the cluster centers in constrained k-means. Secondly, at each iteration of k-means the cluster re-assignment is restricted to the unlabeled samples, while the membership of labeled samples is fixed. This procedure of constrained k-means has shown performance improvement over k-means algorithm. The steps in the procedure are summarized in the Algorithm 4

B.2 Alternating Direction Method of Multipliers

Alternating Direction Method of Multipliers (ADMM) [20] is used to solve large convex optimization problems by decomposing the problem into smaller tractable problems that can be easily solved. Given an optimization problem that is separable in variables allows one to employ ADMM and takes an alternating update strategy by optimizing with respect to each of the variable while other variables are kept fixed. While we discuss this approach briefly here, more details of the method and related proofs are present in [20].

Algorithm 4 Constrained KMeans [13]

Input: $\mathcal{X}^u = \{\mathbf{x}_1^u, \mathbf{x}_2^u, \cdots \mathbf{x}_m^u\}$: set of unlabeled data points, $\mathcal{X}_k^l = \{\mathbf{x}_1^l, \mathbf{x}_2^l, \cdots \mathbf{x}_p^l\}$: Set of labeled data in class l, $\mathcal{X}^l = \cup_{k=1}^K \mathcal{X}_k^l$: set of labeled data points

Output: Disjoint K clusters $\{\mathcal{C}_k\}_{k=1}^K$

Method:

1. $t = 0$
2. Initialize cluster centers $\mu_k = \frac{1}{|\mathcal{X}_k^l|} \sum_{\mathbf{x} \in \mathcal{X}_k^l} \mathbf{x}$
3. Repeat till convergence
   - Assign data to clusters

     For labeled data :

     $\mathbf{x} \in \mathcal{X}_k^l$ assign $\mathbf{x}$ to cluster $\mathcal{C}_k^{t+1}$ cluster

     For unlabeled data:

     for $\mathbf{x}_i^u \in \mathcal{X}^u$ assign to $\mathcal{C}_k^{t+1}$ cluster obtained by $k = \arg\min_k ||\mathbf{x}_i^u - \mu_k^t||^2$
   - Update centers : $\mu_k^{t+1} = \frac{1}{|\mathcal{C}_k^t|} \sum_{\mathbf{x} \in \mathcal{C}_k^t} \mathbf{x}$
   - $t \leftarrow t + 1$

For the optimization problem, given as follows:

$$\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \tag{B.1}$$

$$\text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c} \tag{B.2}$$

Here, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$ are the variables, $\mathbf{A} \in \mathbb{R}^{p \times n}$, $\mathbf{B} \in \mathbb{R}^{p \times m}$ and $\mathbf{c} \in \mathbb{R}^p$. The functions $f(\mathbf{x})$ and $g(\mathbf{z})$ are convex functions.

We first write the augmented Lagrangian of the problem in (B.1) as follows

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \lambda) = f(\mathbf{x}) + g(\mathbf{z}) + \lambda^\top(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{1}{2}\rho||\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}||_2^2 \tag{B.3}$$

Here, $\lambda \in \mathbb{R}^p$ represents Lagrange variable and $\rho > 0$ is the penalty parameter. Now, the

steps involved in ADMM strategy are given as

$$\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{x}, \mathbf{z}^k, \lambda^k) \tag{B.4}$$

$$\mathbf{z}^{k+1} = \underset{\mathbf{z}}{\operatorname{argmin}} \mathcal{L}_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \lambda^k) \tag{B.5}$$

$$\lambda^{k+1} = \lambda^k + \rho(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}) \tag{B.6}$$

Appendix C

Patterned Species Segmentation Setup

C.1 Feature Extraction

All the images used for testing and training are first oversegmented using SLIC superpixels [18]. The texture features are extracted using a filter bank [19] of 48 filters at differ-ent scales and orientations. Thus, every pixel in the imagehas a 48-dimensional response vector. We use the labeled training images and map pixels from each class to this feature space and perform k-means clustering with $k = 20$. The cluster centers from each class are then concatenated for vector quantization of the feature space. Each superpixel in the oversegmented image is represented by a histogram of texture features. The texture features corresponding to each constituent pixel is assigned to a histogram bin based on its closest cluster center. This texture histogram is normalized to have unit l_1-norm and is our final feature vector.

C.2 Segmentation Strategy

We describe the steps involved in processing a query image.

- **Mean Shift Segmentation**

  The feature vectors representing the superpixels are concatenated with the centroid of the superpixel in image space. Mean shift clustering is used in this concatenated feature space, with the learned metric $\mathbf{M}$ used to compute distance between texture feature vectors, while the spatial distance is computed as the l_2 distance. For all experiments,

we use available bandwidth mean shift with the pointwise bandwidth as the 3-nearest neighbor distance. The step forms clusters of superpixels exhibiting texture and spatial similarity.

- **Distance Map Generation**

  To identify the cluster that corresponds to patterned species, we generate a distance map by computing the average Mahalanobis distance of each cluster with the foreground feature vector extracted from the training images. The cluster with the least average distance is marked as the region containing animal while others are marked as background to generate binary mask.

- **Morphological Operations**

  The cluster marked as animal is often effected by illuminated vegetation and clutter of some background superpixels. We perform two morphological operations on the binary image to obtain the final segmentation mask [67]. The binary image is first operated with a dilation operation then a connected component operation is performed on the dilated image. The largest connected component corresponds to the animal, while other connected components occur due to illuminated background superpixels and hence are discarded.