A study of discriminative data representations for image set classification

March 2022

Naoya Sogi

A study of discriminative data representations for image set classification

Graduate School of Systems and Information Engineering
University of Tsukuba

March 2022

Naoya Sogi

Abstract

For the last decade, image-set based classification methods have gained substantial attention in various applications of multi-view images or videos. The essence of image-set based classification is on how to effectively represent intrinsic information of a set and measure the similarity between two image sets. Among the conventional methods, this thesis focuses on subspace-based methods due to their efficiency, the compactness of a subspace model, and practical and efficient computation. In this type of methods, a set of images is compactly modelled by a subspace in a high-dimensional vector space. After converting each image set to a subspace, the similarity between two sets to be compared can be calculated by using the canonical angles between their subspaces. While the subspace representation has remarkable representation ability of a set, there is still room for the improvement for the classification. This thesis develops accurate classification algorithms by obtaining a discriminative representation of a set. First, we introduce a metric space in the calculation of canonical angles and learning it to obtain suitable subspace representations. This allows us to transform subspace representations such that the new subspaces are still efficient representations of sets and more suitable for the classification than the original one, by obtaining a discriminative metric space through the proposed optimization problem. Then, we introduce convex cone representation as an accurate representation compared with the subspace representation. A convex cone is a set of vectors by a non-negative linear combination of bases. Due to the non-negativity, the convex cone can be a more compact and accurate representation than the subspace. To establish a convex cone-based framework, we mathematically define multiple angles between two convex cones and then use the angles to define the geometric similarity between them. Moreover, to enhance the framework, we introduce two types of feature extraction methods based on the relationship of multiple convex cones. Finally, we show that the similarity can be used as a kernel function and then propose the powerful classification methods using the kernel trick.

Acknowledgements

This thesis is the culmination of efforts from many people around me. Therefore, I would like to use this opportunity to express my sincere gratitude to those who supported me during this PhD journey.

First, I would like to express my deepest gratitude to my advisor Prof. Kazuhiro Fukui for his support and guidance in my research. It was a great privilege to learn about theories and applications of subspace-based methods from him. Besides sharing his knowledge with me, he offered encouragement and was always willing and enthusiastic to assist me. In addition to the advice in the research process, he gave me a lot of advice to appeal research results to experts and un-experts. His cheerful attitude and dedication will continue to be a reference for my future career.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Takahito Kuno, Prof. Keisuke Kameyama, Prof. Jun Sakuma, and Prof. Taro Tezuka, for their insightful comments and for the questions which led me to broaden my research from various perspectives.

Throughout my research journey, I had the pleasure of working with many other knowledgeable researchers. My sincere thanks go to Prof. Jing-Hao Xue, Prof. Rui Zhu And Prof. Tat-Jun Chin, for discussing and supporting me to expand my research. The discussions with Prof. Xue and Prof. Zhu at the UCL and through emails were essential to conduct my research. Without their precious support, it would not be possible to conduct this research.

I would like to express my sincere gratitude to Mrs. Hiroko Sawabe, the Computer Vision laboratory's administrative staff, for supporting me with paperwork and the process of the research fund throughout six years. Without her support, it would not possible to present the research results at conferences and finalize the thesis in the period.

I thank my fellow actual and former labmates in the Computer Vision Lab.: Dr. Bernardo Gatto, Dr. Lincon Souza and Ms. Erica Kido, for research collaborations and for supporting me. I also thank them together with all other members for the countless paper proofreadings and for having interesting and pleasant discussions, not only about research but about many facets of life. Thank you for making this journey an enjoyable experience.

A warm thank you goes to my friends. They put up with my distractions, listened to me unburdened, and did not mind my absence. I am forever grateful for their patience and understanding. I hope to have time now to reconnect with each of them. I extend my gratitude to my parents for their moral support. Their unwavering faith, honesty, and effort took me to where I stand today.

Contents

List of Figures

Chapter 1

Introduction

This thesis proposes image-set classification methods using subspace and convex cone representations. Image-set based classification method has been known as an efficient approach for various types of classification problems, as an image set can capture precise information about the target object than one image. Here, the important step of image-set classification is on how to effectively represent intrinsic information of a set.

We seek to construct accurate classification methods by using rich information of a set. To reach this goal, we propose six methods to classify image sets represented as subspaces or convex cones. To establish the motivation and position of this work, we first overview the image-set classification methods.

1.1 Overview of image-set classification methods

For the last decade, image-set based classification methods have gained substantial attention in various applications of multi-view images or videos, such as 3D object recognition and motion analysis. The essence of image-set based classification is on how to effectively represent intrinsic information of a set and measure the similarity between two image sets, i.e., representation models. To this end, several types of methods using different representations, e.g. covariance matrix, normal distribution, subspace, have been proposed [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. In the following, we briefly describe image set classification methods while focusing on subspace representation based methods, which are representative methods and the main interest in this thesis, while referring to Figure 1.1.

The Mutual Subspace Method (MSM) [1] is the pioneer for image set classification, as shown at the top of Figure 1.1. In MSM, an image set is converted to a subspace representation for extracting intrinsic information and classified based on similarities between input and dictionary subspaces. There are two main reasons to use the subspace representation: 1) it is empirically known that images belonging to the same category distribute in a low-dimensional subspace [16, 17, 18], and 2) it is theoretically known that images taken in a specific condition form a low dimensional subspace [19, 20, 21].

MSM required similarity between two subspaces. To calculate the similarity, [1] used the minimum angles (or the first canonical angle [22]) between two subspaces. [23] proposed to utilize multiple canonical angles to deal with complicated shapes and confirmed its effectiveness on the 3D face recognition task.

The well-known limitation of the MSM is that it does not have any mechanisms to separate subspaces of different categories. This limitation induces poor discrimination ability, and thus decreases its classification performance. To solve the problem, the early attempt is to utilize geometric relationships among subspaces to obtain a discriminative transformation of the subspaces (feature extraction), typified by discriminative canonical correlation (DCC) [24], Constrained MSM (CMSM) [25, 26] and Orthogonal MSM (OMSM) [27, 28], as shown in the upper left of Figure 1.1. These methods were employed successfully in various tasks, and further improved by the kernel trick, i.e., by utilizing non-linear subspace representation [29, 30, 28].

The recent advances in image set classification were mainly made by the introduction of the Grassmann manifold, which is defined as the set of m-dimensional linear subspaces, as shown in the upper right of Figure 1.1. For instance, Grassmann Discriminant Analysis (GDA) [31, 32] is a popular extension method of the MSM. [31] defined the kernel functions on the Grassmann manifold and employed them to generalize the powerful feature extraction method, the Fisher discriminant analysis, for subspaces. GDA has been used as a basic tool to construct novel methods [33, 34]. The representative extensions are Graph-Embedding GDA (GGDA) and Grassmann Dictionary Learning (GRDICT) [35, 36]. GGDA incorporated graph structure inherited in the Grassmann manifold, and GRDICT introduced the concept of dictionary learning and sparse constraint to GDA.

The state of the art methods in subspace-based methods are the projection metric learning (PML) [37], and Riemann manifold metric learning on Grassmann manifold (RMML-GM) [38]. PML learns a transformation matrix of subspaces from the original Grassmann manifold to the Grassmann manifold consisting of the lower-dimensional subspaces, such that the transformed subspaces have the high discriminative ability. To this end, PML defined the optimization problem, and solve it by the gradient descent method. RMML-GM revealed that the global optimum of the optimization problem in the PML can be obtained by Riccati's equation. Furthermore, there are a few studies to attempt to utilize neural network models with subspace representation [39, 40].

Instead of the subspace-based methods, classification methods using other representations have been investigated. Especially, covariance, sparse and affine hull representations showed competitive results on the same tasks succeeded by the subspace-based methods [41, 42, 43]. These methods have the same development as the subspace-based methods [44, 42, 45, 46, 38, 47, 48]. The advantages of the above representations are that they can model a distribution of an image-set accurately compared with subspace. Therefore, if parameters of a representation can be estimated stably, these methods may achieve the same or superior results to the subspace-based methods. However, as the number of images in an image set is typically less than the dimension of image features, it is a bit difficult to estimate the parameters accurately.

1.2 Motivations

The motivation to focus on subspace-based methods is to their efficiency, the compactness of a subspace model, a simple geometrical relationship of class subspaces, and practical and efficient computation. The validity of the subspace representation is also supported by the evidence based on physical characteristics. For example, a low-dimensional subspace (with at most nine dimensions)

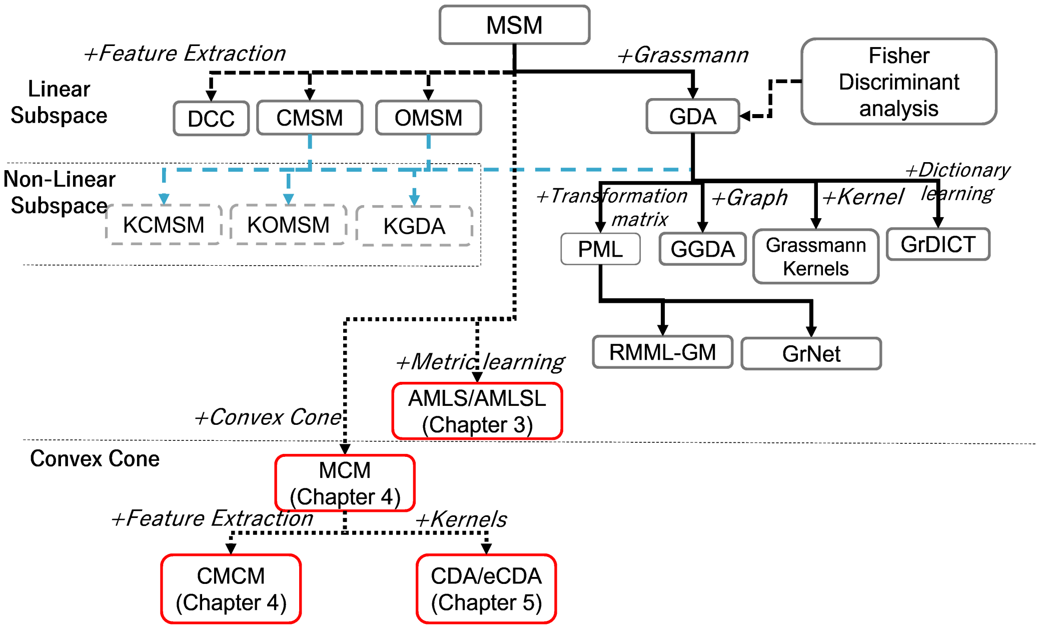


Figure 1.1: Diagram of the algorithms proposed in this thesis (in red boxes) and the relationship with conventional algorithms.

can represent a set of images of a convex object with Lambertian reflectance under a fixed camera view and various illumination conditions [19, 20, 21].

It has been empirically shown that the subspace representation works effectively, even when the above assumptions are not strictly satisfied. In fact, many studies have supported the effectiveness of the subspace representation in various problems [1, 31, 37] as described in the previous section. While the subspace representation has remarkable representation ability of a set, there are still problems for the improvement for the classification.

First problem is that the previous subspace-based methods calculate canonical angles in a metric vector space with a simple scalar product, i.e., Euclidean space. The previous methods have achieved considerable results by designing a method to efficiently utilize similarity (or dissimilarity) defined by canonical angles [49, 22] in Euclidean space. Although this calculation is simple and easy to implement, there is still large room for the improvement of the representation ability in which an identity matrix is naively used as the metric matrix. PML and RMML-GM may have learning mechanisms of the metric matrix, as learning a transformation matrix and scalar product have a close relationship. However, they ignored the orthogonality of basis vectors to simplify the learning process, resulting in that these methods could not fully utilize the geometric relationship among subspaces.

The second problem is that the subspace representation includes redundant space. Various image features, such as LBP, HoG and CNN features, have only non-negative values. This characteristic induces the additivity of feature vectors [50]. Furthermore, the additivity allows only the linear combination with non-negative coefficients of feature vectors. Accordingly, a set of features forms a convex cone instead of a subspace in a high-dimensional vector space, where a convex cone is mathematically defined as a subset of a subspace that is closed under the linear combination with non-negative coefficients. Besides, it is well known that a set of front-facing images under various illumination conditions forms a convex cone, referred to as an illumination cone [19, 20, 21]. The illumination cone has an advantage over the illumination subspace as it has a more accurate representation ability. These facts suggest that subspace representation is rough approximation of convex cone representation.

1.3 Objective

This thesis aims to produce two novel approaches; 1) introducing a metric space and learning it to obtain suitable subspace representations, and 2) introducing the novel representation, convex cone, to the image-set based classification, as an extension of the subspace representation. The goal is to develop classification algorithms that achieve high classification performance by solving the above problems.

1.4 Contributions

Based on the above problems, we enumerate the contributions of this thesis as follows:

1. We propose metric learning methods for image set recognition using subspace representation (Chapter 3). The proposed methods learn a general scalar product space (metric space) that produces more valid canonical angles, i.e., discriminative subspaces. To realize this idea, we first introduce an $\mathbf{A}$-based scalar product instead of the standard scalar product, where $\mathbf{A}$ is a symmetric positive definite matrix and the canonical angles between two subspaces are measured through the $\mathbf{A}$-based scalar product. We learn a discriminative metric space by optimizing metric $\mathbf{A}$ in terms of the Fisher ratio from local Fisher discriminant analysis. Besides, we introduce a mechanism to automatically reduce the dimension of the metric space by imposing a low-rank constraint on metric $\mathbf{A}$.

2. We propose convex cone-based frameworks for image-set classification (Chapter 4) to accurately and compactly represent a set of features. To establish a convex cone-based framework, we mathematically define multiple angles between two convex cones, and then use the angles to define the geometric similarity between them. Moreover, to enhance the framework, we introduce two discriminant spaces. We first propose a discriminant space that maximizes gaps between cones and minimizes the within-class variance. We then extend it to a weighted discriminant space by introducing weights on the gaps to deal with complicated data distribution.

3. We propose convex cone kernels for effectively handling image-set data, where each set of images is compactly represented by a convex cone (Chapter 5). Then, with these convex cone kernels, we propose a framework under which we properly visualize the geometrical relationship among convex cones and build strong classifiers for multiple convex cones.

1.5 Thesis organization

The rest of this thesis is organized as follows.

- Chapter 2 provides the preliminaries for the proposed methods.
- Chapter 3 introduces metric learning method for subspaces.
- Chapter 4 introduces convex cone-based classification frameworks; mutual convex cone method (MCM), Constrained MCM (MCM) and extension of the CMCM.
- Chapter 5 describes the convex cone kernels and cone discriminant analysis.
- Chapter 6 concludes the thesis by providing summaries and future works.

Chapter 2

Preliminary

This section describes the basis for our methods, 1) a subspace representation of an image set, 2) the definition of the similarity between subspaces based on canonical angles in Euclidean space, 3) the two fundamental subspace-based image set classification methods, MSM and CMSM, and 4) convex cone representation of an image set, and the projection of a vector onto a convex cone.

2.1 Subspace representation and canonical angles

Let $\mathbf{X} \in \mathbb{R}^{d \times n}$ be an image set, where the image set has n images, each image is expressed as a d dimensional vector. The image set can be compactly and accurately represented by a low dimensional subspace. The orthonormal basis vectors $\mathbf{S} \in \mathbb{R}^{d \times m}$ of the m dimensional subspace can be obtained as eigenvectors corresponding to m largest eigenvalues of the matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$.

The similarity between two subspaces is measured by the canonical angles. Given two m dimensional subspaces $\mathbf{S}_1$ and $\mathbf{S}_2$ in d-dimensional vector space, the canonical angles $\{0 \leq \theta_1, \cdots, \theta_m \leq \frac{\pi}{2}\}$ between $\mathbf{S}_1$ and $\mathbf{S}_2$ are recursively defined as follows [49, 22]:

$$\cos\theta_i = \max_{\mathbf{u} \in \mathbf{S}_1} \max_{\mathbf{v} \in \mathbf{S}_2} \mathbf{u}_i^{\mathrm{T}} \mathbf{v}_i, \tag{2.1}$$
$$s.t.\ \|\mathbf{u}_i\|_2 = \|\mathbf{v}_i\|_2 = 1,\ \mathbf{u}_i^{\mathrm{T}} \mathbf{u}_j = \mathbf{v}_i^{\mathrm{T}} \mathbf{v}_j = 0,\ i \neq j,$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the canonical vectors producing the i-th smallest canonical angle θ_i between $\mathbf{S}_1$ and $\mathbf{S}_2$. The j-th canonical angle θ_j is the smallest angle in the orthogonal direction to the canonical angles $\{\theta_k\}_{k=1}^{j-1}$. A conventional similarity applying the canonical angles is defined as follows:

$$f_p(\mathbf{S}_1, \mathbf{S}_2) = \sum_{i=1}^{m} \cos^2\theta_i. \tag{2.2}$$

This similarity can be easily obtained as $||\mathbf{S}_1^{\mathrm{T}} \mathbf{S}_2||_F^2$ without calculating individual angles [31].

2.2 Mutual subspace method

Mutual subspace method (MSM) [1] is a classification method of an image set based on its subspace representation. The essence of MSM is to use the structural similarity between subspaces as the similarity between input and reference image sets. The subspace similarity is defined by the canonical angles between two subspaces to precisely compare the whole structures of them, as defined in the previous section.

In MSM, an input subspace $\mathscr{S}_{in}$ is classified by comparison with class subspaces $\{\mathscr{S}_c\}_{c=1}^{C}$ by measuring their similarity using this similarity.

2.3 Constrained MSM

MSM was extended to Constrained MSM (CMSM) [25, 51] by introducing projection of subspaces onto a constraint space. As a constraint space, generalized difference subspace (GDS) [51] is typically used. GDS consists of only difference components among subspaces $\{\mathscr{S}_c\}_{c=1}^{C}$. Thus, the projection of class subspaces onto GDS can enlarge the separability among the class subspaces, substantially enhancing the classification performance of MSM [51].

2.4 Convex cone representation

In this subsection, we describe the definition of a convex cone and the projection of a vector onto a convex cone. A convex cone $\mathscr{C}$ in the d-dimensional vector space $\mathbb{R}^d$ is defined by a finite number of generators (basis vectors) $\{\mathbf{b}_i \in \mathbb{R}^d\}_{i=1}^{r}$ as $\mathscr{C} = \{\mathbf{a}|\mathbf{a} = \sum_{i=1}^{r} w_i\mathbf{b}_i, w_i \geq 0\}$. As indicated by the definition, a convex cone has non-negative constraints on the combination coefficients, unlike a subspace.

Given a set of feature vectors $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^{N}$. We obtain the basis vectors $\{\mathbf{b}_i\}_{i=1}^{r}$ of a convex cone by non-negative matrix factorization (NMF) [52, 53] to suppress noise and remove redundant bases of a cone model. Let $\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2 \dots \mathbf{x}_N] \in \mathbb{R}^{d\times N}$ and $\mathbf{B} = [\mathbf{b}_1\mathbf{b}_2 \dots \mathbf{b}_r] \in \mathbb{R}^{d\times r(<N)}$. NMF generates the basis vectors $\mathbf{B}$ by solving the following optimization problem:

$$\begin{aligned} &\underset{\mathbf{B},\mathbf{W}}{\arg\min} \|\mathbf{X} - \mathbf{B}\mathbf{W}\|_F, \\ &s.t.\ \ (\mathbf{B})_{i,j}, (\mathbf{W})_{i,j} \geq 0, \end{aligned} \tag{2.3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and the number of basis vectors r is a hyperparameter. We use the alternating non-negativity-constrained least squares-based method [53] to solve this problem.

Although the basis vectors can be computed by NMF, the projection of a vector $\mathbf{x}$ onto the convex cone is slightly complicated by the non-negative constraints. In [50], the projection is defined with the non-negative least squares (NNLS) method [54] as follows:

$$\begin{aligned} &\underset{\{w_i\}}{\arg\min} \|\mathbf{x} - \sum_{i=1}^{r} w_i\mathbf{b}_i\|_2, \\ &\qquad\qquad s.t.\ \ w_i \geq 0. \end{aligned} \tag{2.4}$$

The projected vector $\hat{\mathbf{x}}$ is obtained as $\hat{\mathbf{x}} = \sum_{i=1}^{r} w_i\mathbf{b}_i$. In the end, the angle θ between a vector $\mathbf{x}$ and the convex cone can be calculated as follows:

$$\cos\theta = \frac{\hat{\mathbf{x}}^{\mathrm{T}}\mathbf{x}}{\|\hat{\mathbf{x}}\|_2\|\mathbf{x}\|_2}. \tag{2.5}$$

Chapter 3

Metric learning for subspace based method

This chapter discusses a method that enables applying metric learning for subspace-valued data to generating subspace representation which is suitable for the classification. In Section 3.1, we describe the background of the proposed method. In Section 3.2, we elaborate the proposed metric learning method. In Section 3.3, we present the details of the automatic dimension reduction method of the metric space. In Section 3.4, we demonstrate the effectiveness of the proposed methods through classification experiments. Finally, Section 3.5 concludes this chapter.

3.1 Background

Conventional subspace-based methods have achieved considerable results by designing a method to efficiently utilize similarity (or dissimilarity) defined by canonical angles [49, 22] between two subspaces measured in the vector space [1, 25, 29, 30, 28, 31, 55, 56]. In these methods, canonical angles are calculated in a metric vector space with a simple scalar product, i.e., Euclidean space. Although this calculation is simple and easy to implement, there is still large room for the improvement of the representation ability in which an identity matrix is naively used as the metric matrix. To enhance the representation ability of canonical angles, we introduce a generalized concept of canonical angles with a scalar product space based on $\mathbf{A}$-based scalar product [57]. In our metric space, the canonical angles between two subspaces are measured through a scalar product defined by a symmetric positive definite matrix $\mathbf{A}$.

Then, we propose a method for learning a valid metric vector space for more discriminative canonical angles between two given subspaces by searching for a suitable $\mathbf{A}$. The conceptual diagram of the proposed methods is shown in Fig. 3.1. This learning method is equivalent to the deformation problem of the metric vector space, while fixing all the class subspaces. In this sense, the proposed method can be regarded as a kind of dual problem of learning subspace method [58], where the class subspaces are moved in reverse, while fixing the metric vector space.

Our idea of deforming a metric vector space is simple, yet effective. However, the optimization of the metric matrix $\mathbf{A}$ is not trivial, since $\mathbf{A}$ is required to be a symmetric positive definite matrix. This means that we need to solve an optimization problem on a Riemann manifold consisting of

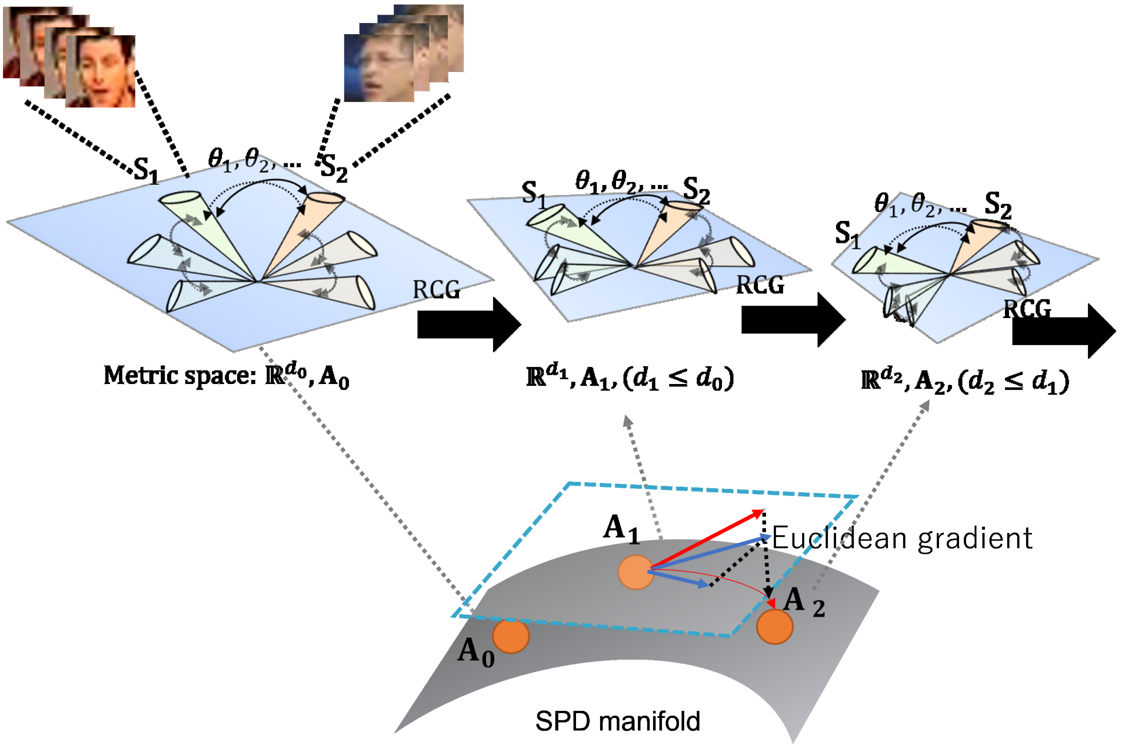

S_1
$\theta_1, \theta_2, \dots$
S_2
RCG
Metric space: $\mathbb{R}^{d_0}, A_0$
S_1
$\theta_1, \theta_2, \dots$
S_2
RCG
$\mathbb{R}^{d_1}, A_1, (d_1 \leq d_0)$
S_1
$\theta_1, \theta_2, \dots$
S_2
RCG
$\mathbb{R}^{d_2}, A_2, (d_2 \leq d_1)$
A_1
Euclidean gradient
A_2
A_0
SPD manifold

Figure 3.1: Conceptual diagram of the proposed metric learning method. The similarity between two subspaces $\mathbf{S}_1$ and $\mathbf{S}_2$ is calculated by the canonical angles $\{\theta_i\}$ in $\mathbf{A}_t$-based scalar product space (metric space). The metric space is updated step by step through the optimization of the metric $\mathbf{A}_t$, using Riemann conjugate gradient (RCG) method such that the different category subspaces are more separated while the same subspaces are more close, with reference to the local relationship between subspaces. The rank of $\mathbf{A}_t$ corresponds to a dimension of t-th metric space. In the optimization, the low-rank constraint on $\mathbf{A}_t$ has the effect of sequentially reducing the dimension of the metric space.

symmetric positive definite matrices. To learn the effective metric space, we define an optimization problem and solve it by the Riemann conjugate gradient method (RCG) [59, 60], as RCG converges in a small number of iterations. Especially, we utilize the Dai-Yuan-type RCG method [61].

Since RCG optimizes $\mathbf{A}$ by using the Euclidean gradient of the objective function, we compute the gradient of the objective function, including the computation of canonical angles with respect to $\mathbf{A}$. Then, with this gradient, a metric space is sequentially updated by the RCG to provide efficient canonical angles for classification. We call this learning algorithm $\mathbf{A}$-based Metric Learning for Subspace representation (AMLS).

A few studies have proposed reliable metric learning methods using subspace representation [37, 38]. The proposed methods is essentially different from these methods in what is the object to be learned. The proposed method focuses on the metric space, in which class subspaces exist. We try to adjust the space itself as mentioned previously. In contrast, the conventional methods focus on each class subspace and rotate them by some orthogonal transformation. Besides, the conventional methods can not fully utilize the geometry of subspaces, as they ignore the orthogonality of basis vectors to simplify their learning processes.

Furthermore, we propose a variation of the proposed method, motivated by the following fact: the rank of $\mathbf{A}$ corresponds to the dimension of a metric space. This characteristic enables us to reduce the dimension d of the metric space sequentially by decreasing the rank of $\mathbf{A}_0$ in optimization steps. To this end, we impose a low-rank constraint on $\mathbf{A}$ by adding a term of trace norm regularization to the cost function. This constraint induces sparseness on the singular values of $d \times d$ matrix $\mathbf{A}_0$. As a result, several singular values are set to zero so that only d' singular values are non-zero. As a result, the original d-dimensional metric space has shrunk to a $d'(< d)$-dimensional metric space, according to the rank of $\mathbf{A}_0$, although the dimension of the metric space still appears to remain d. To extract the actual d' metric space, we perform the dimension reduction from d to d' by applying PCA to a set of learning data in the $\mathbf{A}_0$-based metric space. After that, we project the learning data onto the new d' metric space. A new metric $d' \times d'$ matrix $\mathbf{A}_1$ is optimized for the projected learning data in the same way mentioned above. We call this enhanced method AMLS with the Low-rank constraint (AMLSL).

The same set of steps can be sequentially repeated in the optimization process. In this way, a set of these is sequentially repeated in the optimization process. Besides, our dimension reduction based on sparseness provides a more discriminative space, as demonstrated in the experiment section. Although the process in this idea may seem complicated, it is simple and scalable compared to previous dimension reduction methods.

The main contributions of this chapter are as follow:

1. Introducing the generalized concept of canonical angles based on a metric space for subspace-based image set classification.

2. Deriving the gradient of the similarity computation using the generalized canonical angles, and then proposing a learning method, called $\mathbf{A}$-based metric learning method for subspace representation (AMLS), for generating a discriminant metric space.

3. Incorporating the low-rank constraint on $\mathbf{A}$ into AMLS for sequential dimension reduction of metric space.

4. Demonstrating the fundamental performance of AMLS in tasks of image set based recognition using public databases.

3.2 A-based metric learning for subspace representation

In this section, we first provide the concept of canonical angles in an $\mathbf{A}$-based inner product, which are the necessary fundamental techniques for constructing our method. Then, we describe the details of the proposed $\mathbf{A}$-based metric learning for subspace representation (AMLS). First, we formulate the optimization problem of $\mathbf{A}$ in order to learn a suitable metric space for calculating the canonical angles. Then, we derive the optimization method based on Riemann conjugate method with the gradient of the similarity using the canonical angles.

3.2.1 Canonical angles with an A-based scalar product

The canonical angles described in the previous chapter are measured in a space equipped with the standard Euclidean scalar product. These angles can be generalized by introducing an $\mathbf{A}$-based scalar product, i.e., scalar product is calculated on the metric space defined by the symmetric positive-definite matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ [57].

Let $(\mathbf{x}, \mathbf{y})_{\mathbf{A}} = \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{y}$ be an $\mathbf{A}$-based scalar product, and $\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{(\mathbf{x}, \mathbf{x})_{\mathbf{A}}}$ be the corresponding vector norm. Given two subspaces $\mathbf{S}_1$, $\mathbf{S}_2$, the canonical angles in the $\mathbf{A}$-based scalar product between them are defined as follows:

$$\begin{aligned} \cos\theta_i &= \max_{\mathbf{u}\in\mathbf{S}_1}\max_{\mathbf{v}\in\mathbf{S}_2}(\mathbf{u}_i, \mathbf{v}_i)_{\mathbf{A}}, \\ s.t.\ \|\mathbf{u}_i\|_{\mathbf{A}} = \|\mathbf{v}_i\|_{\mathbf{A}} &= 1,\ (\mathbf{u}_i, \mathbf{u}_j)_{\mathbf{A}} = (\mathbf{v}_i, \mathbf{v}_j)_{\mathbf{A}} = 0,\ i \neq j. \end{aligned} \tag{3.1}$$

The corresponding similarity $f_{\mathbf{A}}$ and dissimilarity $d_{\mathbf{A}}$ are defined as follows:

$$f_{\mathbf{A}}(\mathbf{S}_1, \mathbf{S}_2) = \sum_{i=1}^{m} \cos^2\theta_i = \|\hat{\mathbf{S}}_1^{\mathrm{T}}\mathbf{A}\hat{\mathbf{S}}_2\|_F^2, \tag{3.2}$$

$$\begin{aligned} d_{\mathbf{A}}(\mathbf{S}_1, \mathbf{S}_2) &= \sum_{i=1}^{m} \sin^2\theta_i = \sum_{i=1}^{m}(1 - \cos^2\theta_i) \\ &= m - \|\hat{\mathbf{S}}_1^{\mathrm{T}}\mathbf{A}\hat{\mathbf{S}}_2\|_F^2 = m - f_{\mathbf{A}}(\mathbf{S}_1, \mathbf{S}_2), \end{aligned} \tag{3.3}$$

where the orthonormal basis $\mathbf{S}_1$, $\mathbf{S}_2$ of each subspace are converted to $\mathbf{A}$-orthonormal basis $\hat{\mathbf{S}}_1$, $\hat{\mathbf{S}}_2$, i.e., $\hat{\mathbf{S}}_1^{\mathrm{T}}\mathbf{A}\hat{\mathbf{S}}_1 = \hat{\mathbf{S}}_2^{\mathrm{T}}\mathbf{A}\hat{\mathbf{S}}_2 = \mathbf{I}$ to preserve the orthonormality of a subspace basis under the $\mathbf{A}$-based metric by the following equation:

$$\hat{\mathbf{S}}_i = \mathbf{S}_i\mathbf{U}_i\Sigma_i^{-1/2}, \tag{3.4}$$

where the columns of $\mathbf{U}_i$ are the eigenvectors, and the diagonal elements of Σ_i are the eigenvalues of the matrix $\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_i$. If $\mathbf{A}$ is the identity matrix, the above definition corresponds to the conventional canonical angles in the Euclidean scalar product.

Inspired by this formulation, we propose a metric learning method based on this new scalar product space (also called metric space). To demonstrate its capabilities, we build a high-performance classification method for image sets in the next section.

3.2.2 Problem formulation

Given N image sets $\{\mathbf{X}_i \in \mathbb{R}^{d \times n_i}\}_{i=1}^{N}$, where each image set $\mathbf{X}_i = [\mathbf{x}_1^i, \ldots, \mathbf{x}_{n_i}^i]$ has n_i images, and each image is represented by a $d-$dimensional feature vector $\mathbf{x}_j^i$, our objective is to classify $\mathbf{X}_i$. We first generate subspaces $\{\mathbf{S}_i\}_{i=1}^{N}$ corresponding to each image set. Then, we learn the metric $\mathbf{A}$ by minimizing a discriminative cost function.

To design the cost function, we utilize the idea of local Fisher discriminant analysis (LFDA) [62]. LFDA can work even if each class has a multimodal distribution by incorporating local relationships between data. First, we define two terms: the sum of the local similarity J_b between subspaces of different categories, and the sum of the local dissimilarity J_w between subspaces of the same category as follows:

$$J_b(\mathbf{A}) = \frac{1}{kN} \sum_{i=1}^{N} \sum_{j \in \mathscr{N}_k^i} f_{\mathbf{A}}(\mathbf{S}_i, \mathbf{S}_j), \tag{3.5}$$

$$\begin{aligned} J_w(\mathbf{A}) &= \frac{1}{kN} \sum_{i=1}^{N} \sum_{j \in \mathscr{P}_k^i} d_{\mathbf{A}}(\mathbf{S}_i, \mathbf{S}_i) \\ &= \frac{1}{kN} \sum_{i=1}^{N} \sum_{j \in \mathscr{P}_k^i} m - f_{\mathbf{A}}(\mathbf{S}_i, \mathbf{S}_i), \end{aligned} \tag{3.6}$$

where, $\mathscr{N}_k^i$ and $\mathscr{P}_k^i$ are index sets of k-neighbor subspaces with the i-th subspace in different categories and the same category, respectively. Finally, we define the discriminative cost function as follows:

$$J(\mathbf{A}) = J_b(\mathbf{A}) + J_w(\mathbf{A}) + \lambda(1 - \|\mathbf{A}\|_F^2)^2, \tag{3.7}$$

where, the last term is the regularization to prevent the norm of $\mathbf{A}$ from becoming too large, and $\lambda(\geq 0)$ is the weight parameter of the regularization. As the cost function decreases, we can obtain a more suitable metric space for classification. In the next subsection, we describe the optimization method for the cost function with respect to $\mathbf{A}$.

3.2.3 Optimization

To minimize the cost function, we utilize the Riemann conjugate gradient method (RCG), since the metric $\mathbf{A}$ is on the manifold of symmetric positive definite matrices. The RCG updates $\mathbf{A}$ by searching a suitable matrix along the geodesic of the direction to the Riemann gradient of the cost function. The Riemann gradient is, intuitively, the closest vector to the Euclidean gradient that is also tangent to the manifold, in this case of symmetric positive definite matrices. Therefore, we need to show the Euclidean gradient of the cost function. To this end, since the function has the terms of the similarity $f_{\mathbf{A}}$, in order to easily obtain the gradient, we first reformulate the similarity as follows:

$$\begin{aligned} f_{\mathbf{A}}(\mathbf{S}_i, \mathbf{S}_j) &= \|\hat{\mathbf{S}}_i^{\mathrm{T}} \mathbf{A} \hat{\mathbf{S}}_j\|_F^2 \\ &= tr(\hat{\mathbf{S}}_j^{\mathrm{T}} \mathbf{A} \hat{\mathbf{S}}_i \hat{\mathbf{S}}_i^{\mathrm{T}} \mathbf{A} \hat{\mathbf{S}}_j) = tr(\hat{\mathbf{S}}_j \hat{\mathbf{S}}_j^{\mathrm{T}} \mathbf{A} \hat{\mathbf{S}}_i \hat{\mathbf{S}}_i^{\mathrm{T}} \mathbf{A}) \\ &= tr(\mathbf{S}_j \mathbf{U}_j \Sigma_j^{-1/2} \Sigma_j^{-1/2} \mathbf{U}_j^{\mathrm{T}} \mathbf{S}_j^{\mathrm{T}} \mathbf{A} \mathbf{S}_i \mathbf{U}_i \Sigma_i^{-1/2} \Sigma_i^{-1/2} \mathbf{U}_i^{\mathrm{T}} \mathbf{S}_i^{\mathrm{T}} \mathbf{A}) \\ &= tr(\mathbf{S}_i^{\mathrm{T}} \mathbf{A} \mathbf{S}_j (\mathbf{S}_j^{\mathrm{T}} \mathbf{A} \mathbf{S}_j)^{-1} \mathbf{S}_j^{\mathrm{T}} \mathbf{A} \mathbf{S}_i (\mathbf{S}_i^{\mathrm{T}} \mathbf{A} \mathbf{S}_i)^{-1}). \end{aligned} \tag{3.8}$$

In the above equation, we can use the cyclic property of the matrix trace and $\mathbf{U}_i\Sigma_i^{-1}\mathbf{U}_i^{\mathrm{T}} = (\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_i)^{-1}$. This allows us to obtain the gradient $\nabla_{\mathbf{A}} f_{\mathbf{A}}(\mathbf{S}_i, \mathbf{S}_i)$ of the similarity with respect to $\mathbf{A}$ as follows:

$$\begin{aligned}
&\nabla_{\mathbf{A}} f_{\mathbf{A}}(\mathbf{S}_i, \mathbf{S}_i) = \nabla_{\mathbf{A}} tr(\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_j(\mathbf{S}_j^{\mathrm{T}}\mathbf{A}\mathbf{S}_j)^{-1}\mathbf{S}_j^{\mathrm{T}}\mathbf{A}\mathbf{S}_i(\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_i)^{-1}) \\
&\quad = \mathbf{S}_i(\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_i)^{-1}\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_j(\mathbf{S}_j^{\mathrm{T}}\mathbf{A}\mathbf{S}_j)^{-1}\mathbf{S}_j^{\mathrm{T}} \\
&\quad\quad - \mathbf{S}_j(\mathbf{S}_j^{\mathrm{T}}\mathbf{A}\mathbf{S}_j)^{-1}\mathbf{S}_j^{\mathrm{T}}\mathbf{A}\mathbf{S}_i(\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_i)^{-1}\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_j(\mathbf{S}_j^{\mathrm{T}}\mathbf{A}\mathbf{S}_j)^{-1}\mathbf{S}_j^{\mathrm{T}} \\
&\quad\quad + \mathbf{S}_j(\mathbf{S}_j^{\mathrm{T}}\mathbf{A}\mathbf{S}_j)^{-1}\mathbf{S}_j^{\mathrm{T}}\mathbf{A}\mathbf{S}_i(\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_i)^{-1}\mathbf{S}_i^{\mathrm{T}} \\
&\quad\quad - \mathbf{S}_i(\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_i)^{-1}\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_j(\mathbf{S}_j^{\mathrm{T}}\mathbf{A}\mathbf{S}_j)^{-1}\mathbf{S}_j^{\mathrm{T}}\mathbf{A}\mathbf{S}_i(\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_i)^{-1}\mathbf{S}_i^{\mathrm{T}}.
\end{aligned} \tag{3.9}$$

With this gradient $\nabla_{\mathbf{A}} f_{\mathbf{A}}$, we can obtain the Euclidean gradient of the cost function as follows:

$$\begin{aligned}
\nabla_{\mathbf{A}} J(\mathbf{A}) =& \frac{1}{kN}\sum_{i=1}^{N}\sum_{j\in\mathscr{N}_k^i} \nabla_A f_{\mathbf{A}}(\mathbf{S}_i, \mathbf{S}_j) \\
&- \frac{1}{kN}\sum_{i=1}^{N}\sum_{j\in\mathscr{P}_k^i} \nabla_A f_{\mathbf{A}}(\mathbf{S}_i, \mathbf{S}_j) \\
&- \lambda(1 - \|\mathbf{A}\|_F^2)\mathbf{A}.
\end{aligned} \tag{3.10}$$

$\mathbf{A}$ can be updated by the RCG with this Euclidean gradient, until the maximum number of iterations is reached.

The image-set classification can be executed by the nearest neighbor strategy by using the similarity by the optimized $\mathbf{A}$.

3.3 Dimension reduction of a metric space

In this section, we describe the details of the proposed method of sequentially reducing the dimension of the original metric space. The method consists of two steps, as mentioned in Sec. 3.1. First, we outline the algorithm of the optimization with a low-rank constraint. Next, we show how to apply principal component analysis (PCA) to our $\mathbf{A}$-based metric space to reduce its dimensionality.

3.3.1 Optimization with Low rank constraint

Our idea is based on the fact that the rank of metric $\mathbf{A}$ indicates the dimension of a metric space, as mentioned previously. This suggests that we can realize the dimension reduction effectively by decreasing the rank of a metric matrix $\mathbf{A}$. We impose a low-rank constraint on $\mathbf{A}$ by adding a term of trace norm regularization to our cost function $J(\mathbf{A})$. The cost function is then modified as

$$J_{lr}(\mathbf{A}) = J(\mathbf{A}) + \eta\|\mathbf{A}\|_*, \tag{3.11}$$

where $\|\mathbf{A}\|_*$ indicates the trace norm of $\mathbf{A}$, and $\eta(> 0)$ is the weight parameter of the regularization.

To minimize the above mentioned cost function, $J_{lr}(\mathbf{A})$, we utilize the proximal gradient method (PGM) [63, 64, 65]. This method is built as a combination of two processes: a regularization operation and the traditional gradient-based method without the trace norm regularization.

In this thesis, we consider a combination of an operation for the trace norm regularization and the RCG method for optimizing $J(\mathbf{A})$ described in the previous section. The first operation $\mathrm{prox}_{\eta}^{tr}$ is defined as follows:

$$\mathrm{prox}_{\eta}^{tr}(\mathbf{A}) = \mathbf{U}\max(\Sigma - \eta\mathbf{I},\ 0)\mathbf{U}^{\mathrm{T}}, \tag{3.12}$$

where the columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{A}$, the Σ is a diagonal matrix whose diagonal elements are the eigenvalues of $\mathbf{A}$, $\mathbf{I}$ is the identity matrix, and $\max$ is the element-wise max operation [66]. In PGM, the operation $\mathrm{prox}_{\eta}^{tr}$ is applied to $\mathbf{A}$, after each step of the RCG to minimize $J(\mathbf{A})$. PGM can be interpreted as it incorporates the effect of the regularization, after optimizing $\mathbf{A}$ under the original cost function $J(\mathbf{A})$. Therefore, the rank of $\mathbf{A}$ is reduced in the step of applying the operation $\mathrm{prox}_{\eta}^{tr}$. After the rank has decreased to d' by this optimization, we reduce the dimension by PCA with the $\mathbf{A}$-based scalar product. In the next subsection, we describe a method to make a transformation matrix $\mathbf{V} \in \mathbb{R}^{d \times d'}$ from the d-dimension metric space to the actual d'-dimensional metric space by PCA.

Algorithm 1: Algorithm of the proposed metric learning method.

```
Input: $N$ image sets $\{\mathbf{X}_i \in \mathbb{R}^{d \times n_i}\}_{i=1}^{N}$, class labels $\{l_i\}_{i=1}^{N}$ and weight parameters $\lambda, \eta$, subspace
    dimension $m$, the number of neighborhood subspaces $k$, and the number of maximum
    iterations $T$
Output: $\mathbf{A}_T$ and $\mathbf{V}_T$
    Initialization : $\mathbf{A}_0 \in \mathbb{R}^{d \times d}$
$\mathbf{V}_0 = \mathbf{I}$
Generate $m$-dimensional subspace $\mathbf{S}_i$ for each image set $\mathbf{X}_i$.
for $t = 1$ to $T$ do
    Compute the gradient $\nabla_{\mathbf{A}_{t-1}} J(\mathbf{A}_{t-1})$ by Eq. 3.10.
    $\mathbf{A}_t$ = RCG($\mathbf{A}_{t-1}$, $\nabla_{\mathbf{A}_{t-1}} J(\mathbf{A}_{t-1})$ )
    $\mathbf{A}_t$ = $\mathrm{p}rox_{\eta}^{tr}(\mathbf{A}_t)$, (Eq. 3.12)
    if Rank of $\mathbf{A}_t$ is reduced then
        Calculate $\mathbf{V}$ by Eq. 3.16
        $\mathbf{V}_t$ = $\mathbf{V}^{\mathrm{T}}\mathbf{V}_{t-1}$
        $\mathbf{A}_t = \mathbf{V}^{\mathrm{T}}\mathbf{A}_t\mathbf{V}$
        for $i = 1 \ldots N$ do
            $\hat{\mathbf{X}}_i = \mathbf{V}_t\mathbf{X}_i$
            Generate $m$-dimensional subspace $\mathbf{S}_i$ from $\hat{\mathbf{X}}_i$
        end for
    else
        $\mathbf{V}_t$ = $\mathbf{V}_{t-1}$
    end if
end for
```

3.3.2 Dimension reduction based on $\mathbf{A}$-based PCA

The dimension of the metric space still appears to remain d, after the rank of $\mathbf{A}$ is reduced to d' in the previous subsection. This means that we need to extract the actual d'-dimensional metric space. To this end, we apply PCA to a set of learning data, where PCA is required to be performed in the updated metric space based on the optimized metric $\mathbf{A}$ in the previous section.

In this subsection, we explain the technical details of how to obtain a transformation matrix $\mathbf{V} \in \mathbb{R}^{d \times d'}$ that maps data from the original d-dimensional metric space to the $d'(=$ the reduced rank of $\mathbf{A})$-dimensional metric space.

First of all, we describe how to perform PCA in the $\mathbf{A}$-based metric space. Given N image sets $\{\mathbf{X}_i \in \mathbb{R}^{d \times n_i}\}_{i=1}^{N}$, where each image set $\mathbf{X}_i = [\mathbf{x}_1^i, \dots, \mathbf{x}_{n_i}^i]$ has n_i images, and each image is represented by a d-dimensional vector $\mathbf{x}_j^i$. As well known, PCA is a method for maximizing the scalar product between image vectors $\{\mathbf{x}_j^i\}$ and a principal component vector $\mathbf{w}$. We define the objective function of PCA in the $\mathbf{A}$-based metric space as follows:

$$\arg\max_{\mathbf{w}} \sum_{i,j} (\mathbf{w}, \mathbf{x}_j^i)_{\mathbf{A}}^2 = \sum_{i,j} (\mathbf{w}^{\mathrm{T}} \mathbf{A} \mathbf{x}_j^i)^2 \quad s.t. \ \|\mathbf{w}\|_{\mathbf{A}}^2 = 1. \tag{3.13}$$

The above optimization problem can be rewritten by the Lagrange multipliers method as follows:

$$\arg\max_{\mathbf{w}} \sum_{i,j} (\mathbf{w}^{\mathrm{T}} \mathbf{A} \mathbf{x}_i^j)^2 + \alpha(\mathbf{w}^{\mathrm{T}} \mathbf{A} \mathbf{w} - 1)^2. \tag{3.14}$$

The differential of the Eq. (3.14) yields the following equation.

$$\mathbf{ARAw} - \beta \mathbf{Aw} = \mathbf{ARv} - \beta \mathbf{v}, \tag{3.15}$$

where $\mathbf{v} = \mathbf{A}w$ and $\mathbf{R} = \sum_{i,j} (\mathbf{x}_j^i)(\mathbf{x}_j^{i\mathrm{T}})$. By assuming that Eq. (3.15) equals 0, the transformation matrix $\mathbf{V} \in \mathbb{R}^{d \times d'}$ can be obtained as a matrix whose columns are the $d'(= rank(\mathbf{A}))$ eigenvectors corresponding to the d' largest eigenvalues of the following equation:

$$\mathbf{ARV} = \beta \mathbf{V}. \tag{3.16}$$

With this matrix $\mathbf{V}$, we extract a new metric $\mathbf{A}_1$ of the actual d'-dimensional metric space as $\mathbf{V}^{\mathrm{T}} \mathbf{A} \mathbf{V}$. Each image vector is projected onto this new metric space as $\mathbf{V}^{\mathrm{T}} \mathbf{x}_j^i$. After that, new subspaces $\{\mathbf{S}_i\}$ are generated from the projected vectors $\{\mathbf{V}^{\mathrm{T}} \mathbf{x}_j^i\}_j$.

AMLS with the Low-rank constraint is called AMLSL. This algorithm is summarized in Algorithm 1. If $\eta = 0$, the algorithm corresponds to AMLS.

3.4 Evaluation experiments

In this section, we demonstrate the effectiveness of the proposed methods through the extensive experiments on two tasks: multi-view image-based object recognition and video-based face recognition. For object recognition task, we used ETH-80 dataset [67] and RGB-D dataset [68]. For face recognition task, we used YouTube Celebrity dataset [69] and YouTube Face dataset [70].

3.4.1 Experimental settings

The **ETH-80 dataset** consists of eight different categories, captured from 41 viewpoints, and there are 10 objects for each category. Five objects were randomly sampled from each category and used as a training data, and the remaining five objects were used as a testing data. As an input image set, we used 41 multi-view images for each object. We resized each image to 32 × 32 pixels and used a 1024-dimensional feature vector whose element is a pixel intensity of the corresponding image. We evaluated the classification performance of each method in terms of the average accuracy of ten trials using randomly divided datasets.

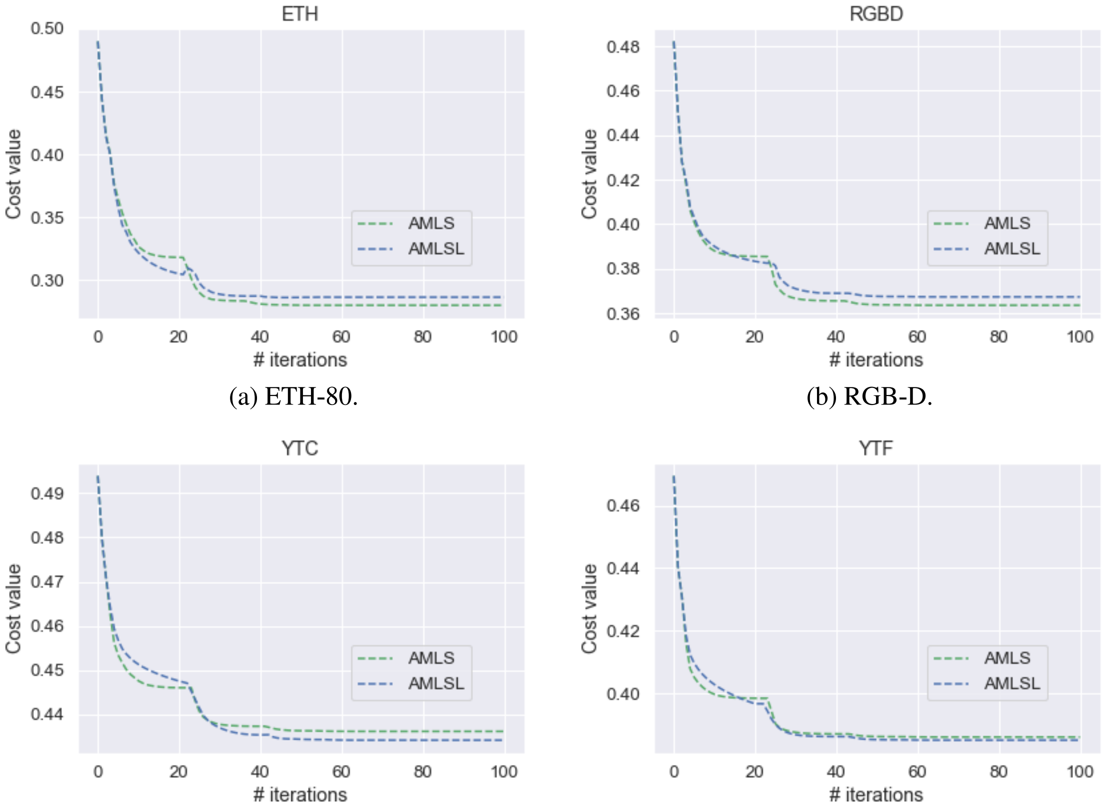
The **RGBD Object dataset** [68] consists of object images in 51 different classes. There are 3 to 14 objects in each class, and each object is captured from over 200 viewpoints. As an image set, we used a set of multi-view images of each object. The half image sets per each class were randomly selected as training data, and the remaining image sets were used as test data. We repeated the evaluation five times with different random selections. For this dataset, we used CNN features [71] extracted from the ResNet-18 [72] trained on ImageNet [73].

The **YTC dataset [69]** contains 1910 videos of 47 people. Face recognition using this dataset is still challenging since all of the face images are low-resolution and face images of the same person have extreme variations, such as change of face direction and emotion. We used a set of face images extracted from a video by the Viola and Jones detection algorithm [74], as an image set. All the extracted face images were resized to 20 × 20 pixels. After converting each image to grayscale, we applied a histogram equalization contrast adjustment method as a preprocessing. We used a 400-dimensional feature vector whose element is a pixel intensity of the corresponding image. We used six videos per each person randomly selected as training data, and nine videos per each person randomly selected as test data. We evaluated the classification performance of each method in terms of the average accuracy of five trials using randomly divided datasets.

The **YouTube Faces (YTF) database** [70] contains 3425 videos of 1595 people. We cropped face regions with the annotated data [70] and used the cropped face images of each video as an image set. As we removed the classes with only one or two videos, the number of classes used for this experiment is 226. As with the experiment for the RGBD dataset, the image sets of each class were randomly split in half into training and test data. We repeated the evaluation five times with different random splits. We used CNN features extracted from the ResNet-50 trained on the VGGFace2 dataset [75].

The proposed methods have five parameters: the maximum number of optimization steps, weight parameters λ and η for corresponding regularization, subspace dimension, and the number of neighbor subspaces k. The maximum number of optimization steps was fixed to 100 and λ was fixed to 1e-3. For the proposed method with the low-rank constraint, AMLSL, η was fixed at 1e-4. The subspace dimension was tuned at the range from 10 to 30 with the increments of 10 by the grid search algorithm on the training data. The number of neighbors k was tuned at the range from 2 to 10 with the increments of 2 by the same strategy.

To examine the effectiveness of the proposed methods, we compared them with various methods: classification methods based on models other than subspace: affine hull based method (AHISD) [41], sparse representation-based method (SANP) [43], Gaussian distribution-based method (GPL) [12] and covariance based metric learning methods, Log-Euclidean Metric Learning (LEML) [44] and Riemann Manifold Metric Learning on Symmetric Positive Definite manifold (RMML-SPD) [38].



(a) ETH-80.

(b) RGB-D.

(c) YTC.

(d) YTF.

Figure 3.2: The curves of the cost function $J(\mathbf{A})$ on four datasets. The horizontal axis means the number of iterations of the optimization method. The blue and green lines are values of the cost function on AMLS and AMLSL respectively.

For comparison with subspace based classification method, we compared with traditional methods: MSM [1], CMSM [25], OMSM [28], Discriminative Canonical Correlations (DCC) [24], Grassmann Discriminant Analysis (GDA) [31], Graph-Embedding GDA (GGDA)[55], Manifold-manifold distance (MMD) [76], Grassmann dictionary learning method (GRDICT) [35], and and state-of-the-art metric learning method: Projection Metric Learning (PML) [37], and RMML on Grassmann manifold (RMML-GM) [38].

Besides, to confirm the effectiveness of AMLSL, we show the result by the combination of AMLS and the naïve PCA, where the PCA is used to reduce the feature dimension into the same dimension obtained by AMLSL.

The parameters, other than the subspace dimension, of the above methods were tuned at the suggested range by the original papers under our experimental settings using the grid-search algorithm on the training data. The subspace dimension was tuned at the range from 10 to 30 with the increments 10 by the same strategy.

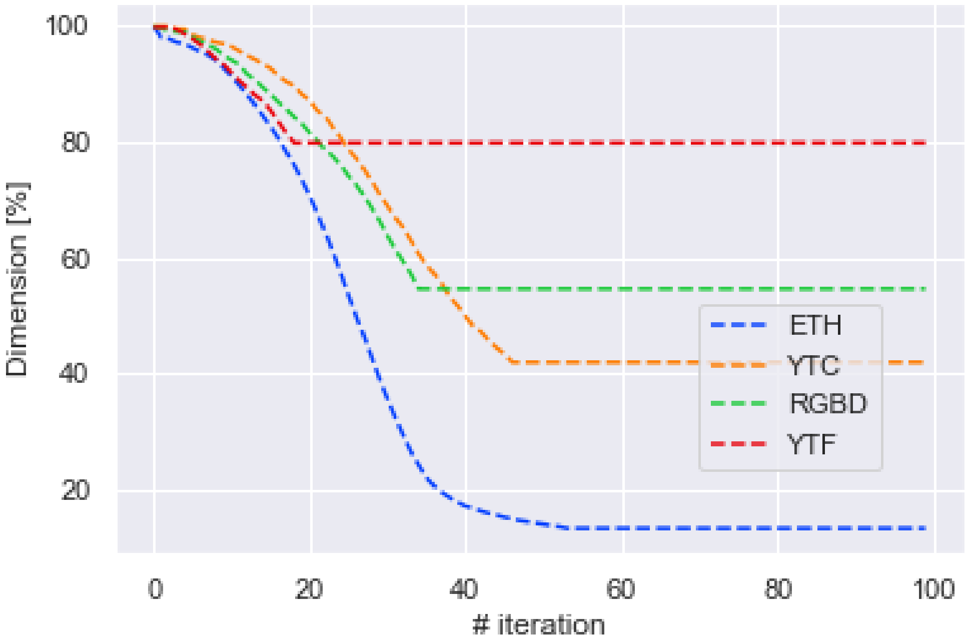


Figure 3.3: The curves of rank of $\mathbf{A}$ (the dimension of a metric space). The horizontal axis means the number of iterations of the optimization method. The vertical axis shows the ratio of the dimensions of the metric space and to the original vector space, i.e., $d'/d \times 100$.

3.4.2 Results and discussion

We first discuss the characteristic of the proposed methods by using the results shown in Fig. 3.2 and 3.3. Figure 3.2 and 3.3 show transitions of the cost function $J(\mathbf{A})$ (Eq. (3.7)) and the rank of $\mathbf{A}$ (the dimension of a metric space) in the optimization process. Since the cost values decrease as the number of iterations of the optimization method increases, the basic ability of the proposed optimization method using the differential (Eq. (3.10)) was validated. The low-rank constraint did not contribute to the improvement of the cost value unfortunately, as the cost values of AMLSL are almost the same as that of AMLS. However, it was confirmed that the optimization step automatically learns the dimension of the metric space. This may suppress the effect of overfitting in the testing phase. The more detailed discussion is in the later with the results of classification.

Table 3.1 shows the experimental results including those from various conventional methods. Overall, subspace-based methods show superior or the same results compared with the other models under our settings. The effectiveness of the proposed methods can be seen as it achieved better results compared with the subspace-based metric learning methods, such as PML and RMML-GM, which learn classification methods based on canonical angles in the standard metric space. This supports the effectiveness of our key idea; to learn efficient metric space for calculating canonical angles.

Furthermore, AMLSL showed competitive results compared with AMLS and better results than

Table 3.1: Experimental results (recognition rate (%), standard deviation) of the proposed metric learning methods on the four datasets.

	ETH-80	YTC	RGB-D	YTF
AHISD [41]	71.00±5.43	68.13±2.22	80.14±1.73	88.99±0.44
SANP	81.25±1.32	58.37±1.83	78.13±2.19	70.09±1.86
GPL	63.25±5.53	72.20±2.70	83.88±1.09	93.04±0.85
LEML [44]	86.00±5.43	62.13±2.04	88.06±3.12	78.68±0.42
RMML-SPD [38]	86.25±4.75	59.55±2.19	88.20±3.58	74.76±0.65
MSM	88.25±3.88	68.06±2.00	89.78±1.06	89.96±0.53
CMSM	94.50±4.15	75.15±1.93	90.22±1.85	91.85±0.57
OMSM	89.75±2.61	73.07±2.09	86.91±1.47	92.38±0.53
DCC [24]	72.75±8.20	73.52±1.96	89.78±2.12	91.63±0.93
GDA [31]	92.50±1.44	64.85±3.32	88.06±1.55	81.19±2.88
GGDA [55]	94.75±3.22	70.97±2.42	88.78±0.64	81.94±2.46
MMD	63.75±5.30	67.28±1.65	82.45±2.72	86.39±1.37
GRDICT	95.25±2.49	74.26±1.90	92.37±1.20	91.81±1.20
PML [37]	93.75±3.17	68.00±0.04	90.22±0.03	89.60±0.06
RMML-GM [38]	90.25±4.16	69.03±1.96	90.93±1.48	89.74±0.70
AMLS	**95.75±0.05**	**76.78±0.01**	92.66±0.00	92.86±0.00
AMLS with PCA	92.50±1.94	75.56±1.53	92.37±0.35	91.94±0.41
AMLSL	94.75±0.68	75.72±0.02	**92.81±0.00**	**92.91±0.00**

AMLS with PCA. This suggests that by applying dimension reduction automatically, AMLSL can efficiently extract essential information for the classification.

3.5 Summary

In this chapter, we discussed $\mathbf{A}$-based Metric Learning for Subspace representation (AMLS). We further extended the idea of AMLS by adding the Low-rank constraint. This enhanced AMLSL is a powerful method with a function of dimension reduction and high discriminative ability. The core ideas behind the proposed methods are 1) measuring the canonical angles in a metric space equipped with metric $\mathbf{A}$, a symmetric positive definite matrix, and 2) designing suitable metric space for calculating canonical angles by optimizing metric matrix $\mathbf{A}$ in terms of the local relationship among subspaces. We formulated the optimization method of $\mathbf{A}$ based on the Riemann conjugate gradient method. To this end, we have rewritten the subspace similarity on a metric space and then derived the gradient of it. We verified the effectiveness of the proposed methods through the extensive classification experiments.

Chapter 4

Convex cone-based method

In this chapter, we propose the three types of image set based classification methods using convex cone representations.

In Section 4.1, we describe the background of the proposed methods. In Section 4.2, we elaborate the fundamental convex cone-based method (Mutual Convex cone Method; MCM). In Section 4.3, we present an extension method of the MCM, Constrained MCM (CMCM), by introducing the projection onto a discriminant space. In Section 4.4, we further extend CMCM by incorporating weight on the generation of the discriminant space. In Section 4.5, we demonstrate the effectiveness of the proposed methods through various experiments. Finally, Section 4.6 concludes this chapter.

4.1 Background

In this chapter, we propose a method for image-set classification based on convex cone models, which can exactly represent the geometrical structure of an image set. As discussed in chapter 1, a convex cone can be a more accurate representation compared with a subspace. To incorporate the convex cone into the framework of image-set recognition, we need to consider how to calculate the structural similarity between two convex cones. To this end, we define multiple angles between two convex cones to capture exactly the geometrical relationship between them, like the canonical angles between two subspaces [22, 49]. We then propose a new method for obtaining the angles in turn from the smallest to the largest by applying the alternating least squares method (ALS) [77] to the convex cones sequentially. Finally, we define the geometric similarity between two convex cones based on the obtained angles. We call the classification method using this similarity the *mutual convex cone method* (MCM).

Furthermore, to improve the performance of the MCM, we introduce the projection of convex cones onto a discriminant space $\mathscr{D}$, which minimizes the within-class variance and enlarges the gaps (between-class variance) between convex cones. The gaps between convex cones precisely capture the difference component between the cones, i.e., difference information, such as shape difference, among various objects. Since such information is essential for classification, the projection onto $\mathscr{D}$ enhances the classification ability of MCM, similarly to that of the projection of class subspaces onto a generalized difference subspace (GDS) in CMSM [51]. Finally, we classify an input image set by using the cone similarity between the projected input and class convex cones $\hat{\mathscr{C}}_{in}, \{\hat{\mathscr{C}}_j\}$, as

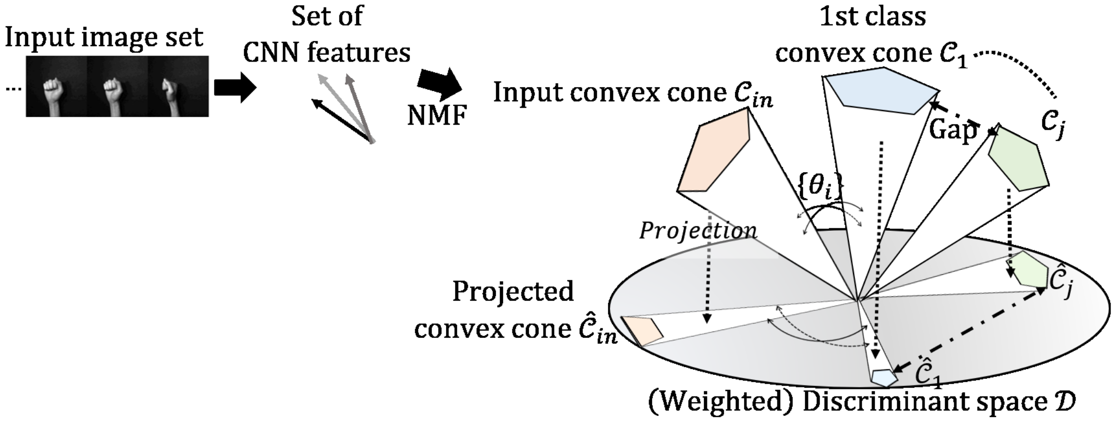

Input image set
Set of CNN features
NMF
1st class convex cone $\mathcal{C}_1$
Input convex cone $\mathcal{C}_{in}$
Gap
$\mathcal{C}_j$
$\{\theta_i\}$
Projection
Projected convex cone $\hat{\mathcal{C}}_{in}$
$\hat{\mathcal{C}}_j$
$\hat{\mathcal{C}}_1$
(Weighted) Discriminant space $\mathcal{D}$

Figure 4.1: Conceptual diagram of the basic idea of the proposed methods. First, feature vectors are extracted from an image set. Then, each set of features is represented by a convex cone. The classification is performed by calculating the similarity based on the angles $\{\theta_i\}$ between the convex cones $\mathscr{C}_{in}$ and $\{\mathscr{C}_j\}$. To enhance the classification ability of this approach, the projection of the convex cones onto the discriminant space $\mathscr{D}$ or the weighted discriminant space is introduced before calculating the angles.

shown in Fig. 4.1. We call this method the *constrained mutual convex cone method* (CMCM).

Then, we further extend the proposed MCM and CMCM, considering more practical cases. So far, MCM and CMCM assume that an image set of a class can be well represented by a single convex cone. However, it is not necessarily reasonable in many practical applications, e.g., in the case that there are multiple videos collected under different situations for a class. In such cases, a single cone is insufficient to represent the complicated structure information. To address this issue, we represent a set of images by multiple convex cones instead of a single convex cone.

Moreover, for the above representation, we redesign how to generate a discriminant space for CMCM. We re-define the between-class variance (gaps) to extract more complicated gaps between multiple convex cones. We calculate the gaps for every pair of convex cones from different classes and then generate a discriminant space from these gaps. In this process, we also introduce weights on gaps to enhance the discriminant ability. The basic idea is that the weight on a gap is set to be larger when the gap is small. From this strategy, we expect the effect that small gaps between two cones can be enlarged after the projection onto the discriminant space.

However, the extension method requires high computational cost in compensation for its high discriminant ability. To alleviate this high computational cost, we introduce a new strategy for fast implementation. The key idea is to divide the similarity calculation into two steps, where we first use the subspace-based method and then use the cone-based method.

In preparation for this fast implementation strategy, we generate the subspaces containing each convex cone by applying the Gram-Schmidt orthogonalization to the basis of the cone in advance. Then, in the first step, we calculate the similarities between the input and reference subspaces corresponding to their original cones, and select several neighbourhood cones of the input cone by using the similarities. The calculation using subspaces is much faster than that directly using cones.

In the second step, we precisely calculate the similarities between the input cone and the selected neighbourhood cones by the proposed method. We name the CMCM with this selection process "fast CMCM". As shown later, the fast CMCM can achieve about ten times speedup in comparison with the original method.

The main contributions of this chapter are summarized as follows.

1. To enhance the subspace-based methods, we introduce a convex cone representation to accurately and compactly represent a set of features.
2. We introduce two novel mechanisms in our cone-based classification: a) multiple angles between two convex cones to measure the similarity between the cones; b) projection of convex cones onto a discriminant space to enlarge the class separability.
3. To enhance the classification performance of the cone-based framework, we propose a weighted discriminant space to further enlarge the class separability by reflecting the local relationship between multiple convex cones of a class.
4. To reduce the computational cost induced by the convex cone representation, we develop a fast implementation of CMCM by switching the cone-based and subspace-based methods.
5. With the valid combination of the contributions 2) and 3), we build three types of novel image-set based classification methods, called MCM, CMCM and extended CMCM, based on the convex cone representation, the discriminant space, and the multiple cone representation.

4.2 Mutual convex cone method

In this section, we describe the algorithm of MCM, after establishing the definition of geometric similarity between two convex cones.

4.2.1 Basic idea

The canonical angles between subspaces can be analytically calculated from the projection matrices in closed form. In contrast, the calculation of the angles between cones is not trivial, as the projection onto a cone includes the process of NNLS (Eq. 2.4). Hence, we propose a new method for obtaining the angles in turn from the smallest to the largest by applying the alternating least squares method (ALS) [77] to the convex cones sequentially. The key idea here is to project convex cones onto the orthonormal complement space of the subspace spanned by two vectors forming the angle obtained in the previous step and then to apply ALS to the projected cones again. This sequential projection works effectively like the orthogonal decomposition of a convex cone in high dimensional vector space.

The following subsection describes the detailed definition of the multiple angles and the similarity between convex cones.

```
Algorithm 2: Algorithm to search for the pair $\mathbf{p}_1$ and $\mathbf{q}_1$.
  Input: Basis vectors $\{\mathbf{b}_i^1\}, \{\mathbf{b}_i^2\}$ of two convex cones, $\mathscr{C}_1$ and $\mathscr{C}_2$.
  Let $\mathscr{P}_j(\mathbf{y})$ be the projection operator of a vector $\mathbf{y}$ onto a convex cone $\mathscr{C}_j$, explained in
    Section 2.4.
  1. Randomly initialize $\mathbf{y} \in \mathbb{R}^d$.
  2. $\mathbf{p}_1 = \mathscr{P}_1(\mathbf{y})/\|\mathscr{P}_1(\mathbf{y})\|_2$.
  3. $\mathbf{q}_1 = \mathscr{P}_2(\mathbf{y})/\|\mathscr{P}_2(\mathbf{y})\|_2$.
  4. $\hat{\mathbf{y}} = (\mathbf{p}_1 + \mathbf{q}_1)/2$.
  5. If $\|\hat{\mathbf{y}} - \mathbf{y}\|_2$ is sufficiently small, the procedure is completed. Otherwise, return to step 2
    after setting $\mathbf{y} = \hat{\mathbf{y}}$.
  return $\cos^2\theta_1 = (\frac{\mathbf{p}_1^{\mathrm{T}}\mathbf{q}_1}{\|\mathbf{p}_1\|_2\|\mathbf{q}_1\|_2})^2$.
```

4.2.2 Multiple angles and geometric similarity between two convex cones

To define the geometric similarity between two convex cones, we consider how to define multiple angles between two convex cones like canonical angles. Let two convex cones $\mathscr{C}_1$, $\mathscr{C}_2$ be formed by basis vectors $\{\mathbf{b}_i^1 \in \mathbb{R}^d\}_{i=1}^{N_1}$ and $\{\mathbf{b}_i^2 \in \mathbb{R}^d\}_{i=1}^{N_2}$, respectively. Assume that $N_1 \leq N_2$ for convenience. As we need to consider the non-negative constraint, the angles between two convex cones cannot be obtained analytically, unlike the canonical angles. Instead, we find two vectors, $\mathbf{p} \in \mathscr{C}_1$ and $\mathbf{q} \in \mathscr{C}_2$, which form the smallest angles between the convex cones. In this way, we sequentially define multiple angles from the smallest to the largest, in order.

First, we search for a pair of d-dimensional vectors $\mathbf{p}_1 \in \mathscr{C}_1$ and $\mathbf{q}_1 \in \mathscr{C}_2$, which have the maximum correlation, by solving the following optimization problem:

$$\cos\theta_1 = \max_{\mathbf{p}_1 \in \mathscr{C}_1} \max_{\mathbf{q}_1 \in \mathscr{C}_2} \mathbf{p}_1^{\mathrm{T}}\mathbf{q}_1, \;\; s.t. \; \|\mathbf{p}_1\|_2 = \|\mathbf{q}_1\|_2 = 1. \tag{4.1}$$

This problem can be solved by the alternating least squares method (ALS) [77]. Thus, the first angle θ_1 can be obtained as the angle formed by $\mathbf{p}_1$ and $\mathbf{q}_1$. The algorithm of the ALS is summarized in Algorithm 2.

For the second angle θ_2, we find a pair of vectors $\mathbf{p}_2$ and $\mathbf{q}_2$ with the maximum correlation, but with the minimum correlation with $\mathbf{p}_1$ and $\mathbf{q}_1$. Such a pair can be found by applying ALS to the projected convex cones $\mathscr{C}_1$ and $\mathscr{C}_2$ on the orthogonal complement space $\mathscr{S}^{\perp}$ of the subspace $\mathscr{S}$ spanned by the vectors $\mathbf{p}_1$ and $\mathbf{q}_1$ as shown in Fig. 4.2. Then θ_2 is formed by $\mathbf{p}_2$ and $\mathbf{q}_2$. In this way, we can obtain all of the pairs of vectors $\mathbf{p}_i, \mathbf{q}_i$ forming the i-th angle θ_i, $i = 1, \ldots, N_1$.

With the resulting angles $\{\theta_i\}_{i=1}^{N_1}$, we define the geometric similarity sim between two convex cones $\mathscr{C}_1$ and $\mathscr{C}_2$ as

$$sim(\mathscr{C}_1, \mathscr{C}_2) = \frac{1}{N_1}\sum_{i=1}^{N_1}\cos^2\theta_i. \tag{4.2}$$

4.2.3 Algorithm of mutual convex cone method

The mutual convex cone method (MCM) classifies an input convex cone, using the similarities defined by Eq. (4.2) between the input and reference convex cones. MCM consists of two phases, a

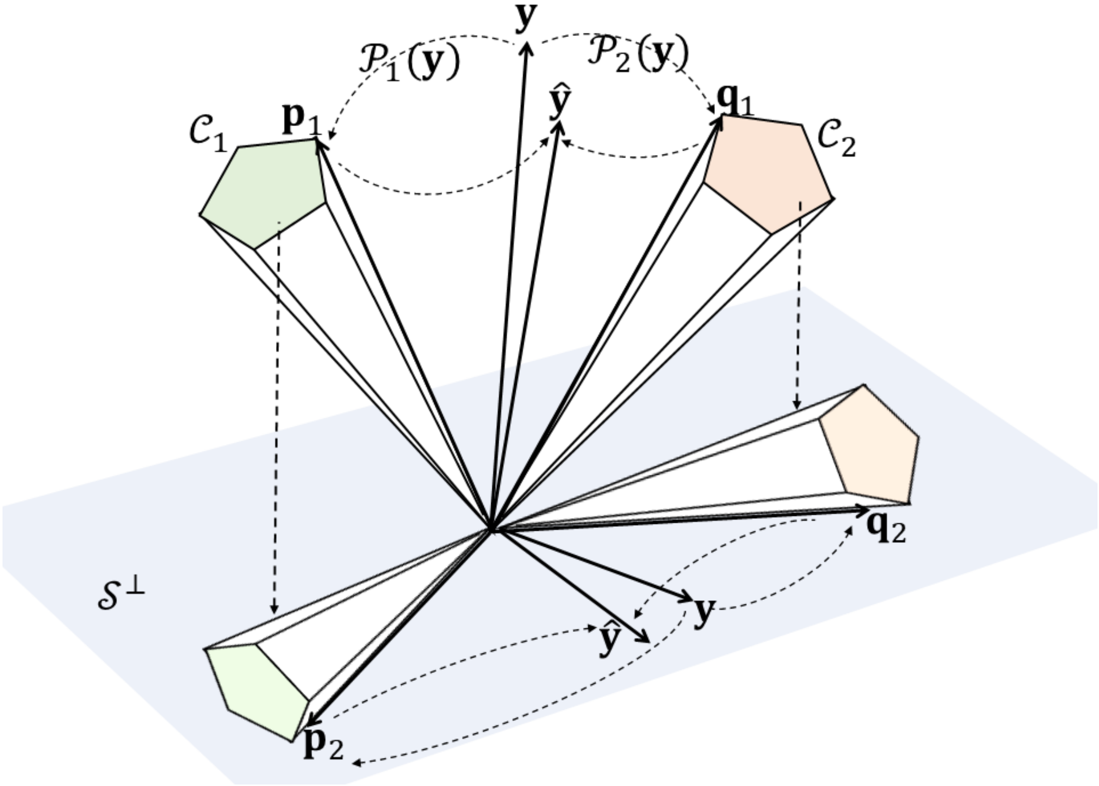


Figure 4.2: Conceptual diagram of the procedure searching for pairs of vectors $\{\mathbf{p}_i, \mathbf{q}_i\}$. The first pair of $\mathbf{p}_1$ and $\mathbf{q}_1$ can be found by the alternating least squares method. The second pair of $\mathbf{p}_2$ and $\mathbf{q}_2$ is obtained by searching the orthogonal complement space $\mathscr{S}^\perp$ of the two dimensional subspace $\mathscr{S} = \text{Span}\{\mathbf{p}_1, \mathbf{q}_1\}$.

training phase and a recognition phase.

Given C class sets with L images $\{\mathbf{x}_i^c\}_{i=1}^L$.

Training Phase

1. Feature vectors $\{\mathbf{f}_i^c\}$ are extracted from the images $\{\mathbf{x}_i^c\}$ of class c.
2. The basis vectors of class-c convex cone, $\{\mathbf{b}_j^c\}$, are generated by applying NMF to the set of feature vectors $\{\mathbf{f}_i^c\}$.
3. $\{\mathbf{b}_j^c\}$ are recorded as the class convex cone of class c.
4. The above process is conducted for all C classes.

Recognition Phase

1. A set of images $\{\mathbf{x}_i^{in}\}$ is input.
2. Feature vectors $\{\mathbf{f}_i^{in}\}$ are extracted from images $\{\mathbf{x}_i^{in}\}$.

3. The basis vectors of the input convex cone, $\{\mathbf{b}_j^{in}\}$, are generated by applying NMF to the input set of feature vectors $\{\mathbf{f}_i^{in}\}$.

4. The input image set $\{\mathbf{x}_i^{in}\}$ is classified based on the similarity (Eq. (4.2)) between the input convex cone $\{\mathbf{b}_j^{in}\}$ and the class-c convex cone $\{\mathbf{b}_j^{c}\}$.

4.3 Constrained mutual convex cone method

In this section, we extend MCM by introducing the projection onto a discriminant space. We first describe the basic idea of the introduction of a discriminant space, and then define the discriminant space based on the gaps among convex cones. After that, we detail the algorithm of the extended MCM.

4.3.1 Basic idea

As convex cones capture essential information of each image set, the gaps between them precisely capture the difference components between corresponding objects, such as the shape difference. The performance of MCM can be enhanced by extracting the gap information, since such information is essential for classification. To this end, we design a discriminant space based on the gaps.

4.3.2 Generation of discriminant space

To enhance the performance of MCM, we introduce a discriminant space $\mathscr{D}$, which enlarge the gaps (the between-class variance $\mathbf{S_b}$) and minimizes the within-class variance $\mathbf{S_w}$ for the convex cones projected on $\mathscr{D}$, similarly to the Fisher discriminant analysis (FDA). In our method, the within-class variance $\mathbf{S_w}$ is calculated from basis vectors of convex cones, and the between-class variance $\mathbf{S_b}$ is calculated from gaps among convex cones for effectively utilizing the information formed by convex cones.

We define these gaps as follows. Let $\mathscr{C}_c$ be the c-th class convex cone with N_c basis vectors $\{\mathbf{b}_i^c\}_{i=1}^{N_c}$, $\mathscr{P}_c$ be the projection operation of a vector onto $\mathscr{C}_c$ defined by Eq. (2.4), and C be the number of the classes. We consider C vectors $\{\mathbf{p}_1^c\}$, $c = 1, 2, \ldots, C$, such that the sum of the correlation $\sum_{i \neq j} (\mathbf{p}_1^i)^{\mathrm{T}} \mathbf{p}_1^j / (\|\mathbf{p}_1^i\|_2 \|\mathbf{p}_1^j\|_2)$ is maximum. Such vectors can be obtained by using the concept of generalized canonical correlation analysis [78, 79]. The detailed procedure is shown in Algorithm 3, which is almost the same as the original algorithm, except that the non-negative least squares (LS) method is used instead of the standard LS method.

Next, we search for a set of second vectors $\{\mathbf{p}_2^c\}$ with the maximum sum of the correlations under the constraint that they have the minimum correlation with the previously found $\{\mathbf{p}_1^c\}$. The second vectors $\{\mathbf{p}_2^c\}$ can be obtained by applying the above procedure to the convex cones projected onto the orthogonal complement space of the vector $\mathbf{y}_1$. In the same way, a set of the j-th vectors $\{\mathbf{p}_j^c\}$ can be computed by applying the same procedure to the convex cones projected onto the orthogonal complement space of $\{\mathbf{y}_k\}_{k=1}^{j-1}$. In this way, we finally obtain the sets of $\{\mathbf{p}_j^c\}$. With the sets of $\{\mathbf{p}_j^c\}$, we define a difference vector $\mathbf{d}_j^{c_1 c_2}$ as

$$\mathbf{d}_j^{c_1 c_2} = \mathbf{p}_j^{c_1} - \mathbf{p}_j^{c_2}. \tag{4.3}$$

Algorithm 3: Procedure to search for a set of first vectors $\{\mathbf{p}_1^c\}_{c=1}^C$

```
Input: Basis vectors $\{\mathbf{b}_i^c\}$ of convex cones $\{\mathscr{C}_c\}_c$
Let $\mathscr{P}_j(\mathbf{y})$ be the projection operator of a vector $\mathbf{y}$ onto a convex cone $\mathscr{C}_j$.
1. Randomly initialize $\mathbf{y}_1$.
2. Project $\mathbf{y}_1$ onto each convex cone, and then normalize the projection as
   $\mathbf{p}_1^c = \mathscr{P}_c(\mathbf{y}_1)/\|\mathscr{P}_c(\mathbf{y}_1)\|_2$.
3. $\hat{\mathbf{y}}_1 = \sum_{c=1}^C \mathbf{p}_1^c/C$.
4. If $\|\mathbf{y}_1 - \hat{\mathbf{y}}_1\|_2$ is sufficiently small, the procedure is completed. Otherwise, return to step
   2 after setting $\mathbf{y}_1 = \hat{\mathbf{y}}_1$.
return $\{\mathscr{P}_c(\mathbf{y})\}_c$
```

Considering that each difference vector represents the gap between the two convex cones, we use these vectors to define $\mathbf{S_b}$ as

$$\mathbf{S_b} = \sum_{j=1}^{N_g} \sum_{c_1=1}^{C-1} \sum_{c_2=c_1+1}^{C} \mathbf{d}_j^{c_1c_2}(\mathbf{d}_j^{c_1c_2})^{\mathrm{T}}, \tag{4.4}$$

where N_g is the minimum number of basis vectors of class convex cones, i.e., $\min(\{N_c\})$.

Next, we define the within-class variance $\mathbf{S_w}$ by using the basis vectors $\{\mathbf{b}_i^c\}$ for all classes of convex cones:

$$\mathbf{S_w} = \sum_{c=1}^{C} \sum_{i=1}^{N_c} (\mathbf{b}_i^c - \mu_c)(\mathbf{b}_i^c - \mu_c)^{\mathrm{T}}, \tag{4.5}$$

where $\mu_c = \sum_{i=1}^{N_c} \mathbf{b}_i^c/N_c$. Finally, the N_d-dimensional discriminant space $\mathscr{D}$ is spanned by N_d eigenvectors $\{\boldsymbol{\phi}_i\}_{i=1}^{N_d}$ corresponding to the N_d largest eigenvalues $\{\gamma_i\}_{i=1}^{N_d}$ of the following eigenvalue problem:

$$\mathbf{S_b}\boldsymbol{\phi}_i = \gamma_i \mathbf{S_w}\boldsymbol{\phi}_i. \tag{4.6}$$

4.3.3 Algorithm of constrained mutual convex cone method

We construct the constrained MCM (CMCM) by incorporating the projection onto the discriminant space $\mathscr{D}$ into the MCM. CMCM consists of a training phase and a recognition phase. In the following, we explain each phase for the case in which C classes have L images $\{\mathbf{x}_i^c\}_{i=1}^L$ each and the discriminant space $\mathscr{D}$ is utilized.

Training Phase

1. Feature vectors $\{\mathbf{f}_i^c\}$ are extracted from images $\{\mathbf{x}_i^c\}$.

2. The basis vectors of the c-th class convex cone, $\{\mathbf{b}_j^c\}$, are generated by applying NMF to each class set of feature vectors.

3. Difference vectors $\{\mathbf{d}_j^{c_1c_2}\}$ are computed according to the method explained in section 4.3.2.

4. The discriminant space $\mathscr{D}$ is generated by solving Eq. (4.6) with $\{\mathbf{b}_j^c\}$ and $\{\mathbf{d}_j^{c_1 c_2}\}$.

5. The basis vectors $\{\mathbf{b}_j^c\}$ are projected onto $\mathscr{D}$.

6. A convex cone formed by a set of the projected basis vectors $\{\hat{\mathbf{b}}_j^c\}_j$ is registered as the class convex cones of class c.

Recognition Phase

1. A set of images $\{\mathbf{x}_i^{in}\}$ is input.

2. Feature vectors $\{\mathbf{f}_i^{in}\}$ are extracted from images $\{\mathbf{x}_i^{in}\}$.

3. The basis vectors of a convex cone, $\{\mathbf{b}_j^{in}\}$, are generated by applying NMF to the set of feature vectors.

4. The basis vectors $\{\mathbf{b}_j^{in}\}$ are projected onto the discriminant space $\mathscr{D}$ and then the lengths of the projected basis vectors are normalized to 1. The normalized projections are represented by $\{\hat{\mathbf{b}}_j^{in}\}$.

5. The input set $\{\mathbf{x}_i^{in}\}$ is classified based on the similarity (Eq. (4.2)) between the input convex cone $\{\hat{\mathbf{b}}_j^{in}\}$ and each class convex cone $\{\hat{\mathbf{b}}_j^c\}$.

4.4 Extension of constrained mutual convex cone method

In this section, we further enhance the CMCM by incorporating the information of the fine local structure between different classes into the generation of an enhanced discriminant space, considering the case that an image set of a class has the complex structure. We further improve the ability of the enhanced discriminant space by introducing weights on gaps. Finally, we propose a fast implementation of the enhanced CMCM.

4.4.1 Basic idea

In practical applications, a class distribution are often complicated, for example, in the case that an image set of a class contains multiple videos collected under different situations. In these cases, it is reasonable to represent each class by multiple cones instead of a single cone. In our method using multiple reference cones, the classification of an input cone $\mathscr{C}_{in}$ is performed by the nearest neighbor classifier in a very similar procedure to the original CMCM using a single convex cone, except that this enhanced CMCM uses a newly designed discriminant space.

In the following sections, we redesign the method for generating an enhanced discriminant space in response to the multiple convex cone representation of a class. Then, we introduce weights on gaps to incorporate local structure into the generation of an enhanced discriminant space.

4.4.2 CMCM with enhanced discriminant space

Consider n^c training image sets $\{\mathbf{X}_i^c\}_{i=1}^{n^c}$ for the c-th class, where each image set $\mathbf{X}_i^c$ has n_i^c images $\{\mathbf{x}_{i,j}^c\}_{j=1}^{n_i^c}$. Let $\mathbf{F}_i^c \in \mathbb{R}^{d \times n_i^c}$ be a matrix whose j-th column vector is a feature vector such as pixel intensities and CNN feature extracted from the image $\mathbf{x}_{i,j}^c$, and $\{\mathbf{b}_{i,j}^c \in \mathbb{R}^d\}_{j=1}^{N_i^c}$ be the basis vectors of each reference convex cone $\mathscr{C}_i^c$ generated from each feature set $\mathbf{F}_i^c$.

We first reformulate the between-class variance $\mathbf{S_b}$ by using multiple gaps (difference vectors) $\{\mathbf{d}_j^{ik,hl}\}$ between every pair of convex cones $\mathscr{C}_i^k$ and $\mathscr{C}_h^l$ from different classes as follows:

$$\mathbf{S}'_\mathbf{b} = \sum_{j=1}^{N_g} \sum_{k=1}^{C-1} \sum_{l=k+1}^{C} \sum_{i=1}^{n^k} \sum_{h=1}^{n^l} \mathbf{d}_j^{ik,hl} (\mathbf{d}_j^{ik,hl})^{\mathrm{T}}, \tag{4.7}$$

where N_g is the minimum number of basis vectors of references convex cones, min($\{N_i^c\}$), and $\{\mathbf{d}_j^{ik,hl}\}$ are the difference vectors between vector pairs $\{\mathbf{p}_j^{ik}, \mathbf{p}_j^{hl}\}$ of two convex cones $\mathscr{C}_i^k$ and $\mathscr{C}_h^l$, which are obtained by applying the method described in Sec. 4.3.2, and C is the number of classes.

Subsequently, we reformulate the within-class variance $\mathbf{S_w}$ as follows:

$$\mathbf{S}'_\mathbf{w} = \sum_{c=1}^{C} \sum_{i=1}^{n^c} \sum_{j=1}^{N_i^c} (\mathbf{b}_{i,j}^c - \mu_i^c)(\mathbf{b}_{i,j}^c - \mu_i^c)^{\mathrm{T}}, \tag{4.8}$$

where μ_i^c is the mean vector of the basis vectors $\{\mathbf{b}_{i,j}^c\}$ of the i-th reference convex cone of the c-th class, $\{\mathscr{C}_i^c\}$, which is calculated by $\sum_{j=1}^{N_i^c} \mathbf{b}_{i,j}^c / N_i^c$.

We obtain an enhanced discriminant space $\mathscr{D}_e$ as the subspace spanned by N_d eigenvectors corresponding to the N_d largest eigenvalues of the following eigenvalue problem:

$$\mathbf{S}'_\mathbf{b} \phi_i = \gamma_i \mathbf{S}'_\mathbf{w} \phi_i. \tag{4.9}$$

With the multiple cone representation, the projection of cones onto $\mathscr{D}_e$ can more accurately and finely maximize between-class variance $\mathbf{S}'_\mathbf{b}$ while minimizing within-class variance $\mathbf{S}'_\mathbf{w}$, in comparison with a naive discriminant space generated based on a single cone representation.

4.4.3 Enhanced discriminant space with weights

So far, all the gaps (difference vectors) are treated with the same contribution to generating the enhanced discriminant space $\mathscr{D}_e$. However, the smallest gaps between convex cones are more important for discrimination than the largest gaps. In fact, several studies reported the validity of this idea [55, 80, 62]. Motivated by these studies, we expect that adding different weights can ensure that the smallest gaps between two cones are selectively enlarged after the projection onto the discriminant space, which leads to a better discriminative ability. Concretely, we enhance the discriminant space by introducing the weighted between-class variance with weights $\{w^{ikhl}\}$ as follows:

$$\mathbf{S}''_\mathbf{b} = \sum_{j=1}^{N_g} \sum_{k=1}^{C-1} \sum_{l=k+1}^{C} \sum_{i=1}^{n^k} \sum_{h=1}^{n^l} w^{ikhl} \mathbf{d}_j^{ik,hl} (\mathbf{d}_j^{ik,hl})^{\mathrm{T}}, \tag{4.10}$$

where the weights $\{w^{ikhl}\}$ are defined as

$$w^{ikhl} = \sum_{j=1}^{N_g} ((\mathbf{p}_j^{ik})^{\mathrm{T}} \mathbf{p}_j^{hl})^2 / N_g. \tag{4.11}$$

This formulation means that the value of w^{ikhl} increases as the corresponding gap between convex cones becomes smaller. The enhanced discriminant space with weights maximizes more effectively the between-class variance, while minimizing the within-class variance. We can obtain the enhanced discriminant space as the subspace spanned by N_d eigenvectors corresponding to the N_d largest eigenvalues of the following eigenvalue problem:

$$\mathbf{S}_{\mathbf{b}}'' \boldsymbol{\phi}_i = \gamma_i \mathbf{S}_{\mathbf{w}}' \boldsymbol{\phi}_i. \tag{4.12}$$

In the following, we refer this further enhanced discriminant space as the weighted discriminant space $\mathscr{D}_{ew}$.

4.4.4 Validity of enhanced discriminant space

To see the high discriminative ability of the enhanced discriminant spaces, we visualize the projections of convex cones onto $\mathscr{D}$, $\mathscr{D}_e$, and $\mathscr{D}_{ew}$ as 2D maps by using multi-dimensional scaling (MDS) [81]. For the visualization, we synthesized ten cones $\{\mathscr{C}_i^1\}_{i=1}^{10}$ for class-1 and five cones $\{\mathscr{C}_i^2\}_{i=1}^{5}$ for class-2. Each convex cone $\mathscr{C}_i^c$ is spanned by three 100-dimensional basis vectors $\{\mathbf{b}_{i,j}^c \in \mathbb{R}^{100}\}_{j=1}^{3}$. Figure 4.3a shows the 2D visualization map of the synthesized cones without any projection.

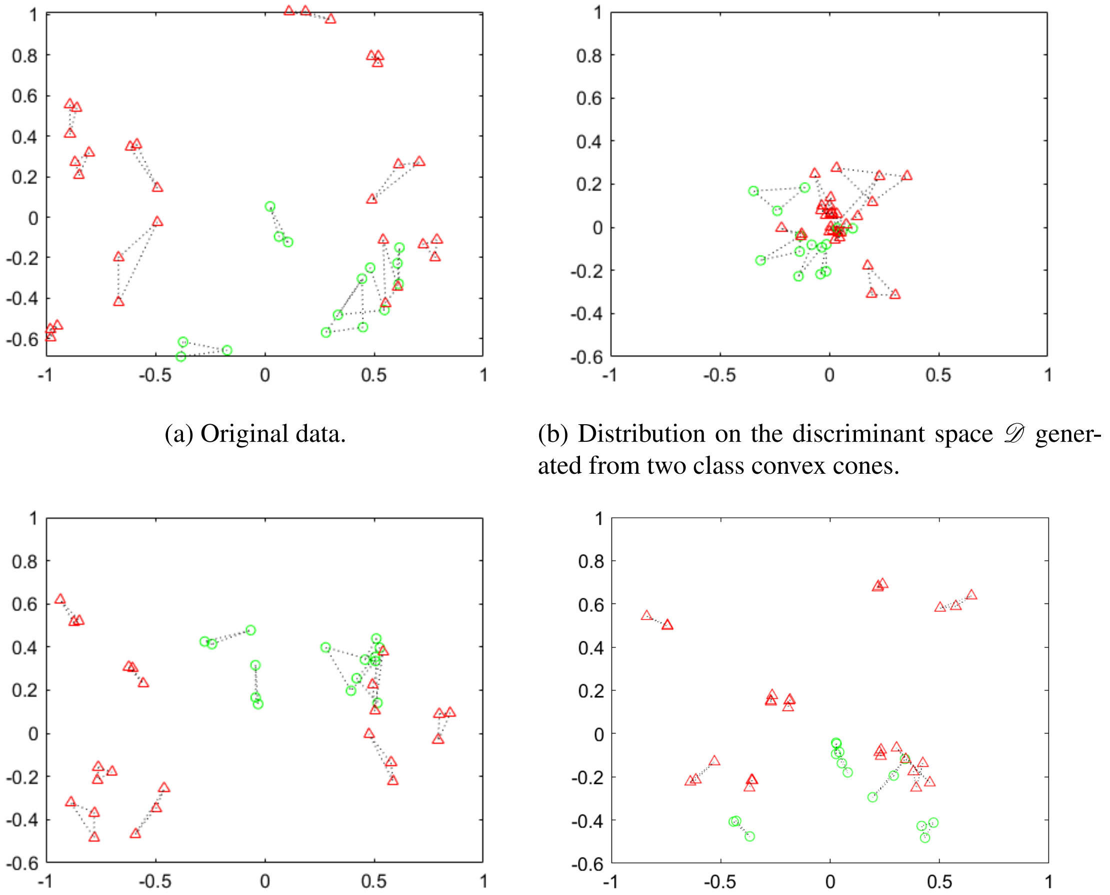
Figs. 4.3 b, c and d show the maps of the cones projected onto $\mathscr{D}$, $\mathscr{D}_e$ and $\mathscr{D}_{ew}$, respectively. Through the comparison between the two maps in b and c, we can see that the cones projected onto $\mathscr{D}_e$ are better separated than the cones projected onto $\mathscr{D}$. This indicates a clear advantage of the enhanced discriminant space $\mathscr{D}_e$ using multiple cones for each class over the naive discriminant space $\mathscr{D}$ using a single cone for each class.

Next, we evaluate the validity of introducing weights to the enhanced discriminant space $\mathscr{D}_{ew}$. We cannot see a large visual difference between the two projection maps of c and d, since a 2D projection map cannot capture completely high-dimensional structures in the 100-dimensional vector space. However, by comparing them carefully, we can observe that there is no overlap in d, while there are partial overlaps in c.

To further verify the advantage of $\mathscr{D}_{ew}$ over $\mathscr{D}_e$ quantitatively, we calculated the class separability of the projected cones in the original 100-dimensional vector space. The class separability is defined as $\mathrm{tr}(\mathbf{S_b}')/\mathrm{tr}(\mathbf{S}_{\mathbf{w}}')$. This index was used for both $\mathscr{D}_e$ and $\mathscr{D}_{ew}$ for fair comparison. The class separabilities of the projections on D_e and D_{ew} are 259.4 and 910.6, respectively. This large difference supports clearly the validity of the introduction of weights to the enhanced discriminant space.

4.4.5 Fast implementation of CMCM

CMCM is much more computationally costly than the subspace-based methods, MSM and CMSM. This is because the calculation of similarity between cones needs heavy computation due to ALS.



(a) Original data.

(b) Distribution on the discriminant space $\mathscr{D}$ generated from two class convex cones.

(c) Distribution on $\mathscr{D}_e$.

(d) Distribution on $\mathscr{D}_{ew}$.

Figure 4.3: Results of the projections onto discriminant spaces $\mathscr{D}, \mathscr{D}_e$, and $\mathscr{D}_{ew}$. Each plot is generated by MDS. The shapes of each point indicate the corresponding class. The dotted lines are plotted between basis vectors of a cone.

Moreover, the cost of the extended CMCM with N reference cones is N times higher than that of the original CMCM.

To alleviate this high cost, we divide the similarity calculation into two steps. The first step is based on the subspace similarity and the second step is based on the cone similarity in Eq.(4.2).

We generate in advance the subspaces containing each cone by applying the Gram-Schmidt orthogonalization to the bases of the cone. Then, in the first step, we generate the input subspace from an input cone and calculate the similarities between the input and reference subspaces. After that, we select several neighborhood reference subspaces according to the subspace similarities obtained above. In the second step, we calculate the similarities of the input cone and the reference cones, which correspond to the reference subspaces selected above. Finally, the input cone is classified into the class with the maximum cone similarity. This two-step process can reduce the computational cost largely, while maintaining the high discriminative ability of the extended CMCM, as clearly shown in experiments in sec.4.5.

4.5 Evaluation experiments

In this section, we conduct five experiments to evaluate the effectiveness of the proposed methods. In the first experiments, we mainly demonstrate the effectiveness of the convex cone representation by comparing the performances of the proposed methods (MCM and the CMCM with $\mathscr{D}$) with the fundamental subspace-based methods (MSM and CMSM). More concretely, the first experiment verifies the effectiveness of using multiple angles between convex cones to measure the structural similarity between them, using the multi-view objects dataset [67].

The second experiment analyzes the attribute of difference vectors between two convex cones by visualizing the difference vectors as heatmap images.

The third experiment reveals how efficiently a convex cone captures essential information of an image set for classification by observing the transitions of performances while varying the number of training data, using the multi-view hand shape dataset [82].

The forth experiment evaluates the validity of the convex cone model and the representation ability of multiple convex cone models for image-set classification on the YouTube Celebrities dataset [69], using four types of typical image features.

The fifth experiment thoroughly evaluates the classification performance of the proposed methods using thehree datasets: 1) YouTube Celebrities (YTC) [69], 2) RGBD Object dataset [68], and 3) YouTube Faces (YTF) [70].

4.5.1 Effectiveness of using multiple angles

In this experiment, we verify the effectiveness of using multiple angles for calculating the structural similarity between convex cones, through a classification experiment using the ETH-80 dataset [67].

Experimental protocol

The ETH-80 dataset consists of object images in eight different classes. Each class has ten types of objects. Thus, this dataset consists of images taken from 80 objects, and each object is captured from 41 viewpoints. One object randomly sampled from each class set was used for training, and the remaining nine objects were used for testing. As an input image set, we used 41 multi-view images for each object. Thus, we have eight image sets for training and 72 image sets for testing. We used images scaled to 32 × 32 pixels and converted to grayscale. Vectorized features of the grayscale images were used as input, i.e. the dimension of the feature vector is 1024.

We evaluated the classification performance of MCM and CMCM, while varying the number of angles used for calculating the similarity. As baselines, MSM and the CMSM were also evaluated. Dimensions of reference subspaces and convex cones were set to 20, and dimensions of input subspaces and convex cones were set to 10.

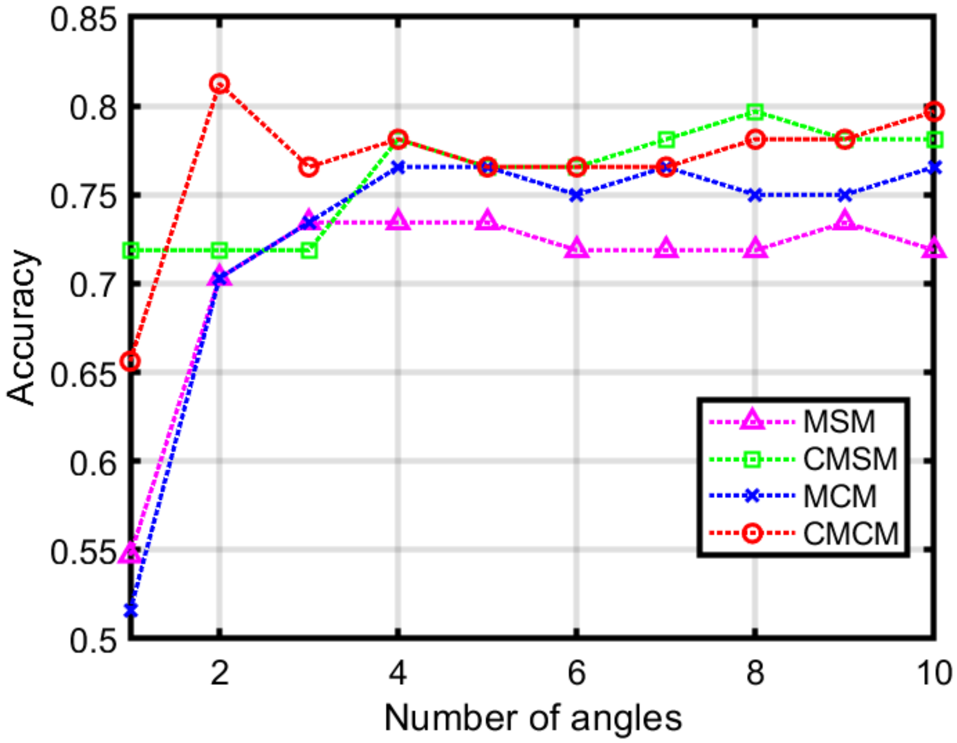


Figure 4.4: Results of classification experiment. The vertical axis is the accuracy, and the horizontal axis is the number of angles used for calculating the similarity.

Results and discussion

Figure 4.4 shows the accuracy changes of the different methods against the number of angles. The horizontal axis denotes the number of angles used for calculating the similarity. We can confirm that the accuracy of MCM and CMCM increases, as the number of angles increases. This result shows clearly the importance of comparing the whole structures of convex cones by using multiple angles rather than using only the minimum angle for accurate classification.

In the case of using one or two angles, the accuracy of CMCM is lower than CMSM. However, with an increase in the number of angles, CMCM outperforms the methods MSM and CMSM which are based on the subspace representation. This supports the effectiveness of the convex cone representation and indicates that using multiple angles is essential to compare the structures of two convex cones.

4.5.2 Visualization of difference vectors between convex cones

In this experiment, we analyze the attribute of difference vectors between convex cones through the visualization of them on two sets of facial expressions, neutral and smile. They were extracted from the CMU PIE dataset [83]. Each set has 20 front face images taken under various illumination conditions. Fig. 4.5 shows sample images of the face images.

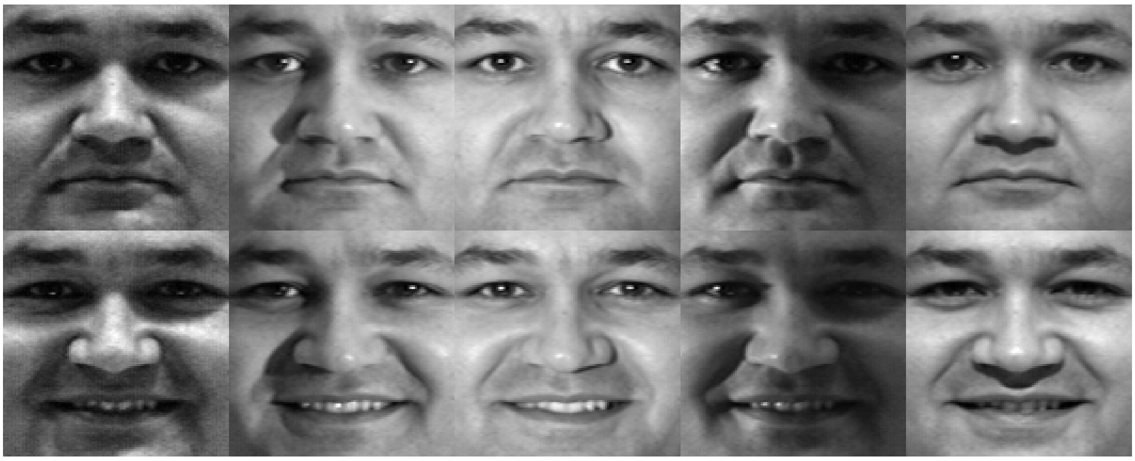


Figure 4.5: Sample images of the facial expression images used in the experiment. The top and bottom rows show the neutral and smile face images respectively.

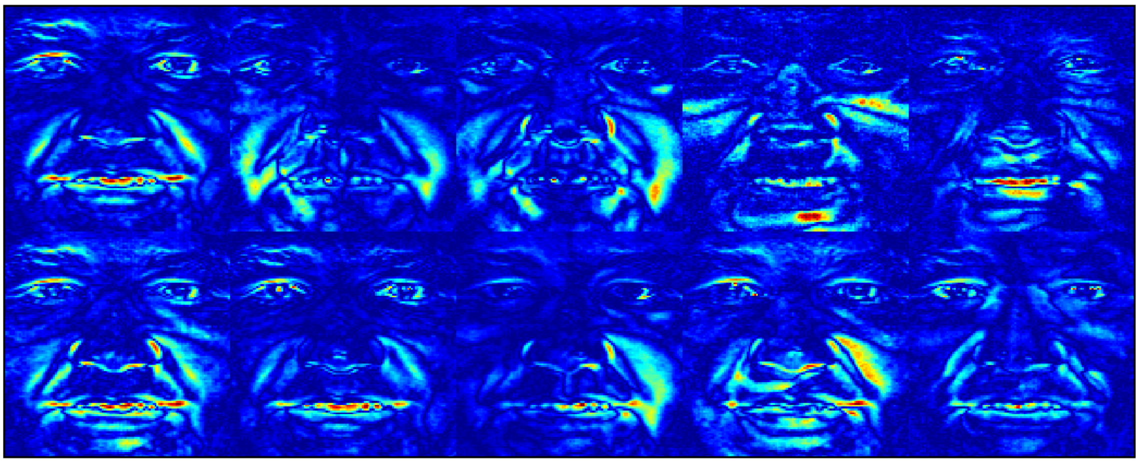
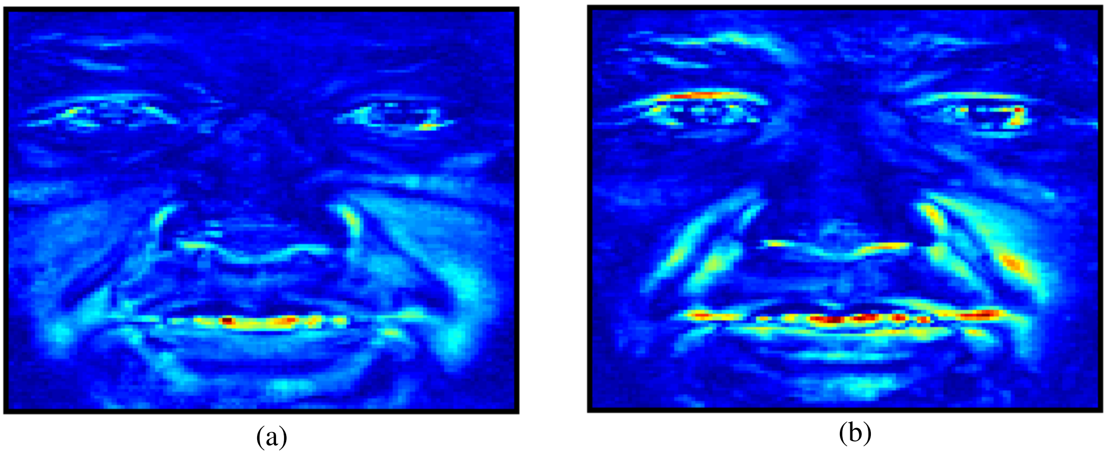


Figure 4.6: Results of visualizing the difference vectors. The top and bottom rows show the difference vector between two subspaces $\{\mathbf{z}_i\}$ and the difference vector between two convex cones $\{\mathbf{d}_i\}$, respectively.

According Facial Action Coding System (FACS) in psychology and anatomia [84, 85], an arbitrary facial expression can be represented as some combination of 44 facial muscle movements, where the i-th movement is called an action unit i (AU-i). For example, the change from neutral facial expression to smile can be represented as a combination of two AUs, raising cheek (AU-6) and pulling lip corner (AU-12). Thus, we evaluate the propriety of the proposed difference vectors by observing how accurately the visualization obtained in this experiment captures the AU-6 and AU-12.

After representing the two sets of raw images as convex cones, we generated the difference vectors $\{\mathbf{d}_i\}$ between the two convex cones according to Eq. (4.3). For comparison, we also calculated



(a) (b)

Figure 4.7: Mean images of absolute value images of the difference vectors $\{\mathbf{d}_i\}_{i=1}^5$ between convex cones and the difference vectors $\{\mathbf{z}_i\}_{i=1}^5$ between subspaces. (a) $\sum_{i=1}^5 |\mathbf{z}_i|/5$. (b) $\sum_{i=1}^5 |\mathbf{d}_i|/5$.

the difference vectors $\{\mathbf{z}_i\}$ between the canonical vectors of two subspaces of the two sets. We set the number of basis vectors of each convex cone to 5 and the dimension of each subspace to 5. Thus, we have five difference vectors for each model.

Figure 4.6 shows the visualization results of $\{\mathbf{z}_i\}_{i=1}^5$ and $\{\mathbf{d}_i\}_{i=1}^5$. We can see that both sets of the difference vectors can emphasize regions around the cheek and mouth. These regions can move largely in comparison with other regions when changing from neutral facial expression to smile. However, the resolutions in variation captured by them are a bit different. To take a closer look at this difference, we calculated mean images of the absolute values of the difference vectors, by $\sum_{i=1}^5 |\mathbf{z}_i|/5$ and $\sum_{i=1}^5 |\mathbf{d}_i|/5$, as shown in Fig. 4.7. The difference vectors, $\{\mathbf{z}_i\}$, between the subspaces capture roughly difference on the whole face. On the other hand, the difference vectors, $\{\mathbf{d}_i\}$, between convex cones capture clearly fine difference around the cheek and mouth, which correspond precisely to the AU-6 and AU-12.

4.5.3 Representation ability of a convex cone

This experiment aims to reveal how efficiently a convex cone captures essential structural information of an image set. To this end, we evaluate our methods while changing the number of training data, using the multi-view hand shape dataset [82]. As detailed later, we show the results from the soft voting with a CNN, in addition to our methods and MSM/CMSM, to verify the importance of considering structural information of image sets.

Experimental protocol

The multi-view hand shape dataset [82] consists of 30 classes of hand shapes. Each class data was collected from 100 subjects at a speed of 1 fps for 4 seconds using a multi-camera system equipped with seven synchronized cameras at intervals of 10 degrees. During data collection, the subjects were asked to rotate their hands at a constant speed to increase the number of viewpoints.

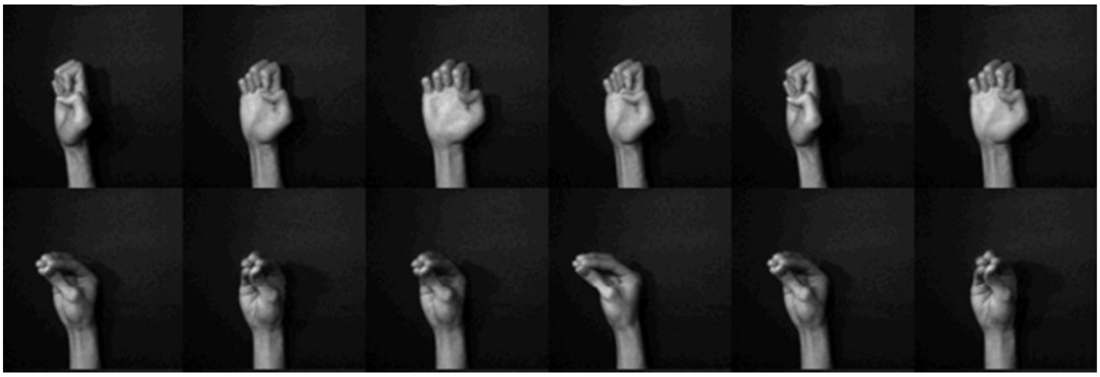


Figure 4.8: Sample images of the multi-view hand shape dataset used in the experiments. Each row shows a hand shape from various viewpoints.

Figure 4.8 shows sample images in the dataset. The total number of images collected was 84000 (= 30 classes×4 frames×7 cameras ×100 subjects).

We randomly divided the subjects into two sets. One set was used for training, and the other was used for testing. To evaluate the efficiency of convex cone representation, we conducted the experiment by setting the numbers of subjects used for training to 1, 2, 3, 4, 5, 10, and 15. Hence, the number of training images was $840N$ (= 30 classes×7 cameras×4 frames× N subjects). We set the number of subjects used for testing to 50. We treated 28 images of a subject as an input image set. Thus, the total number of convex cones for testing was 1500 (=30 classes×50 subjects).

In this experiment, we used CNN features. To extract effective CNN features, we fine-tuned the ResNet-50 [72] pre-trained on ImageNet [73]. To this end, we slightly modified the architecture of the ResNet-50 for our experimental setting. First, we replaced the final layer of the ResNet-50 with a 1024-way fully connected (FC) layer with the ReLU function. Next, we added a $class\ number$-way FC layer with softmax behind the replaced FC layer. Then, we trained the modified ResNet using the training images.

We extracted a CNN feature of each image from the replaced 1024-way FC layer. Thus, the dimensionality d of a CNN feature vector was 1024.

Besides, we utilized this fine-tuned network as a baseline of the methods, which do not consider the structure of an image set, with the following procedure; we classified an input image set based on the average value of the output conviction degrees class from the last FC layer with softmax. In the following, we call this method as "softmax".

For subspace and cone-based methods, the parameters were tuned by grid search on the training set with the following ranges: the dimension of class subspaces and convex cones varied from 10 to 50 in increments of 10; the dimension of input subspace and convex cone varied from 5 to 20 in increments of 5. For CMCM, the dimensions of the discriminant space $\mathscr{D}$ was set to the matrix rank of the between-class variance $\mathbf{S_b}$. For CMSM, the maximum dimension d_{max} of GDS is (the dimension of class subspace) × (the number of classes). The dimension of GDS was tuned by the same strategy while varying the reduction dimension d_{red} with the range from 5 to 30 in increments of 5. The dimension of GDS is set to $d_{max} - d_{red}$. This strategy was also conducted on the experiments in the later subsections.

Results and discussion

Table 4.1 shows the accuracies versus the number N of training subjects. The overall performances of the subspace and convex cone methods achieved competitive results compared with that of softmax. In particular, the improvements are significant when the number of training subjects N is small. From this result, we can see the importance of considering structural information of image sets.

Table 4.1: Change in the accuracies (%) against the number of training subjects.

N	1	2	3	4	5	10	15
softmax	36.07	71.41	83.87	86.60	91.60	95.73	96.53
MSM	62.27	73.47	85.27	87.60	91.13	95.27	96.20
CMSM	65.87	74.73	87.40	91.00	92.87	95.73	96.27
MCM	63.07	74.60	85.67	88.27	92.07	95.40	96.67
CMCM	**67.87**	**75.33**	**87.47**	**91.33**	**93.53**	**96.27**	**97.00**

Our methods outperformed the subspace-based methods, MSM and CMSM. This supports the effectiveness of our core ideas: the utilizing of convex cone representation and the cone similarity with multiple angles. Besides, the result suggests that a convex cone can extract meaningful information of image-sets efficiently and stably, even if the number of training data is small.

Moreover, CMCM showed superior performance to MCM in all cases. This improvement indicates the effectiveness of the projection onto $\mathscr{D}$, which is designed to extract discriminative features based on differences among the cones. This insight also means the efficient representation ability of a convex cone model, since the difference among them works well.

4.5.4 Representation ability of multiple convex cones

In this subsection, we evaluate the representation ability of multiple convex cones in addition to a single convex cone, on the video-based face recognition dataset, YouTube Celebrities (YTC) [69], using four representative image features: Local Binary Pattern (LBP) [86], Histogram of Gradient (HoG) [87] and two types of CNN features, which are extracted from ResNets trained on ImageNet [73] and VGGFace2 [75] datasets, respectively. We show the classification performances of our methods, including the CMCM with $\mathscr{D}_{ew}$ and its fast implementation. For convenience, we use wCMCM to denote the CMCM with $\mathscr{D}_{ew}$.

Experimental protocol

The YTC dataset contains 1910 videos of 47 people. We used a set of face images extracted from a video by the Incremental Learning Tracker (ILT) [88], as an image set. Six videos per each person were randomly selected as training data, and nine videos per each person were randomly selected as test data. We repeated the evaluation five times with different random selections.

For extracting LBP and HoG features, all the extracted face images were scaled to 30 $\times$ 30 pixels and converted to grayscale. For extracting CNN features, all face images were scaled to 224 $\times$ 224 pixels and then inputted to the networks.

Table 4.2: Parameter ranges used in the grid search algorithm. d_{is} and d_{cs} are the dimensions of input and class subspaces, respectively. N^c and N^{in} are the number of basis vectors of input and class cones, respectively. d_{red} is a parameter for the dimension of GDS, and N_i^c is the number of basis vectors of each cone used for wCMCM. Each element [x–y/z] in the table means that the corresponding parameter varied from x to y in increments of z.

	YTC	RGBD	YTF
d_{is},N^{in}	[3–15/3]	[4–20/4]	[3–15/3]
d_{cs},N^c	[6–30/6]	[8–40/8]	[6–30/6]
d_{red}	[3–15/3]	[4–20/4]	[3–15/3]
N_i^c	[3–15/3]	[4–20/4]	[3–15/3]

The parameters were tuned by the grid search algorithm on the training set, with the ranges shown in the left column of Table 4.2. The dimensions of the discriminant space $\mathscr{D}$ and the weighted discriminant space $\mathscr{D}_{ew}$ were set to the matrix rank of the between-class variance $\mathbf{S_b}$ and $\mathbf{S_b''}$, respectively. Besides, for the fast wCMCM, the number of nearest convex cones to be selected by subspace similarities is set to 5.

Table 4.3: Experimental results (recognition rate (%), standard deviation) for the YTC dataset.

	LBP	HoG	ImageNet	VGG face2
#Clusters[min,max/mode]	[1,3/2]	[1,5/2]	[1,3/2]	[1,3/1]
MSM	32.91±1.29	60.90±1.57	55.51±2.04	89.69±0.96
CMSM	48.75±3.26	72.39±1.76	71.02±1.50	91.49±1.00
MCM	37.40±2.81	62.55±1.84	56.12±1.71	90.54±1.16
CMCM	53.33±2.26	72.86±1.92	71.39±1.92	92.34±1.00
wCMCM	**69.27±2.17**	**77.21±2.52**	**81.47±1.32**	**92.96±0.72**
fast wCMCM	69.08±2.18	76.97±2.87	81.42±1.28	92.91±1.16

Results and discussion

Table 4.3 shows the classification results of the baselines and the proposed methods using the four kinds of features. The experimental results support the effectiveness of our main ideas as well as the previous two experiments since MCM and CMCM showed competitive results compared with the baselines regardless of the features we used.

Furthermore, the performance of CMCM increased by introducing multiple convex cones and the weighted discriminant space $\mathscr{D}_{ew}$ in wCMCM. This further enhancement shows that 1) the cones have superior representation ability, and 2) the weights work effectively to obtain local fine structural information between cones of different classes as we expected. Moreover, we notice that the fast version of wCMCM achieved almost the same recognition rate as the original wCMCM, while speeding up more than ten times compared with the original wCMCM as shown in Table 4.4. This result concludes that we can compare the similarity between a pair of cones, which are faraway to each other, by using the subspace similarity instead of the cone similarity.

Although wCMCM significantly outperforms CMCM in all features, amounts of improvements are different. To analyze this difference, we automatically estimated the number of clusters in each class by applying DBSCAN clustering [89]. The second row of Table 4.3 shows the minimum, maximum, and mode numbers of clusters. It can be seen that the more clusters there are in a class on average, the more significantly wCMCM improves the classification performance. This indicates that we can efficiently represent the complex structure of each class by using multiple convex cones and extract meaningful information for classification by using the weighted differences between them.

Table 4.4: Average classification times (millisecond). The numbers of angles for the similarity are 10. This experiment is conducted by Matlab 2018b on Intel CPU i7-7700.

wCMCM	fast wCMCM
507.2	30.0

4.5.5 Comparison of classification performance with conventional methods

In this subsection, we thoroughly evaluate the classification performance of the proposed methods using three public datasets, YTC, RGBD and YTF. As comparison methods, in addition to the baselines, we show the results of the various fundamental and recent subspace-based methods (DCC [24], GDA [31], GGDA [55], MMD [76], PML [37], RMML-GM [38]), as references. In particular, PML and RMML-GM have been known as powerful classification methods for image-set based recognition. Besides, we show the results of other types of methods: covariance-based methods (LEML [44], RMML-SPD [38]), an affine hull-based method (AHISD [41]) and a sparse-representation method (SANP [43]). In the following, details of each dataset and experimental protocols are described. After that, experiment results are shown.

Datasets and experimental protocols

For RGBD and YTF datasets, we utilize the same protocol described in section 3.4. For YTC dataset, we utilize CNN feature vectors extracted from the ResNet50 trained on the VGG Face2 dataset, although other settings are the same as that described in the section 3.4.

The parameters of the proposed methods were tuned by the grid search algorithm on the training set, with the ranges shown in Table 4.2.

Results and discussion

Table 4.5 shows the classification results of the proposed methods and various conventional methods. The proposed methods showed consistent results with the previous experiments for all datasets, i.e., the results support the effectiveness of our key ideas: the convex cone representation, the cone similarity with multiple angles, and the discriminant spaces. For instance, CMCM and wCMCM showed better results than CMSM by more than 1% on the RGBD and YTF datasets.

Furthermore, the proposed method achieved competitive results compared with more powerful subspace-based methods. This result also supports the validity of the proposed methods, and in-

Table 4.5: Experimental results (recognition rate (%), standard deviation) for the three public datasets.

		YTC	RGBD	YTF
Conventional methods	AHISD [41]	90.02±1.17	80.14±1.73	88.99±0.44
	SANP [43]	89.97±1.08	79.28±2.85	70.09±1.86
	LEML [44]	90.83±2.00	88.06±3.12	78.68±0.42
	RMML-SPD [38]	89.93±1.40	88.20±3.58	74.76±0.65
	DCC [24]	92.34±0.81	89.78±2.12	91.63±0.93
	GDA [31]	90.36±1.55	88.06±1.55	81.19±2.88
	GGDA [55]	92.48±1.45	88.78±0.64	81.94±2.46
	MMD [76]	90.30±1.18	82.45±2.72	86.39±1.37
	PML [37]	91.25±0.10	90.22±0.03	89.60±0.06
	RMML-GM [38]	91.30±0.80	90.93±1.48	89.74±0.70
Base lines	MSM	89.69±0.96	89.78±1.06	89.96±0.53
	CMSM	91.49±1.00	90.22±1.85	91.85±0.57
Proposed methods	MCM	90.54±1.16	91.74±0.84	92.60±0.92
	CMCM	92.34±1.00	91.94±0.94	92.82±0.92
	wCMCM	**92.96±0.72**	**92.23±0.94**	**93.17±0.41**
	fast wCMCM	92.91±1.16	91.94±0.60	**93.17±0.31**

dicates that our cone-based frameworks can be further enhanced by utilizing the progress of the subspace-methods in the future.

4.6 Summary

In this chapter, we established a novel framework for image set classification, which is based on the convex cone representation, referred to as the constrained mutual convex cone method (CMCM).

The key idea of our framework is to represent an image set by a convex cone and then measure the similarity between two image sets as the geometric similarity between two corresponding convex cones. The geometric similarity of two convex cones is measured with the angles between them, which we defined newly in this paper, by using the alternating least squares method. To derive higher performance from our cone representation, we designed a new type of discriminant space that increases the class separability between sets of cones from different classes. Moreover, we enhanced the ability of this discriminant space by introducing weights to enlarge the gaps between a pair of close convex cones. As CMCM has high computational cost, we constructed its fast implementation by combining our cone-based method with the conventional subspace-based method.

In the evaluation experiments, we first verified that using multiple angles is essential to compare two convex cones. Then, we demonstrated that the difference between convex cones could capture more useful information for image-set classification. The classification performance of the proposed frameworks was evaluated through extensive experiments, showing that it can achieve competitive results compared with various conventional methods.

Chapter 5

Convex cone kernels

This section proposes convex cone kernels which allows us to apply classical machine learning techniques to convex cone representations.

In Section 5.1, we describe the background of the proposed methods. In Section 5.2, we define convex cone kernels and show their positive semi definiteness. In Section 5.3, we construct two image set classification methods based on the cone kernesl. In Section 5.4, we demonstrate the effectiveness of the proposed methods through two classification experiments. Section 5.5 concludes this chapter.

5.1 Background

As discussed in the previous chapter, it is not a trivial task to explore the complicated geometrical relationship of multiple convex cones. To alleviate this difficulty, we propose several types of convex cone kernel functions based on the smallest angles between two convex cones. We show that our kernel functions can work as positive-definitive functions, both theoretically and empirically. As a result, we can examine the detailed relationship among multiple sets of convex cones by only calculating the kernel functions, as shown in Fig. 5.1.

To illustrate the effectiveness of our convex cone kernels, we visualize the relationship of image-sets through convex cone representation, by using the algorithms of Kernel PCA [90]. Moreover, we propose convex cone discriminant analysis (CDA) for classifying multiple sets of convex cones. The essence of this method is to extend linear discriminant analysis (LDA) [91] to the feature space $\mathscr{F}$ defined by the convex cone kernel. This idea is motivated by kernel discriminant analysis [92, 93, 94]. In addition, to further enhance the performance of CDA, we introduce the projection of convex cones onto a discriminant space [95] to achieve more powerful feature extraction, before kernel mapping the convex cones using the kernel from the original vector space $\mathscr{V}$ into the feature space $\mathscr{F}$. This projection can selectively extract the difference component between convex cone classes, such that the kernel function generated from a set of these emphasized convex cones can be more discriminative. As a result, the discriminant ability of CDA can increase significantly. This enhanced CDA is named enhanced CDA (eCDA).

The main contributions of this paper are as follows:

1. Proposal of several types of convex cone kernels for handling multiple convex cones.

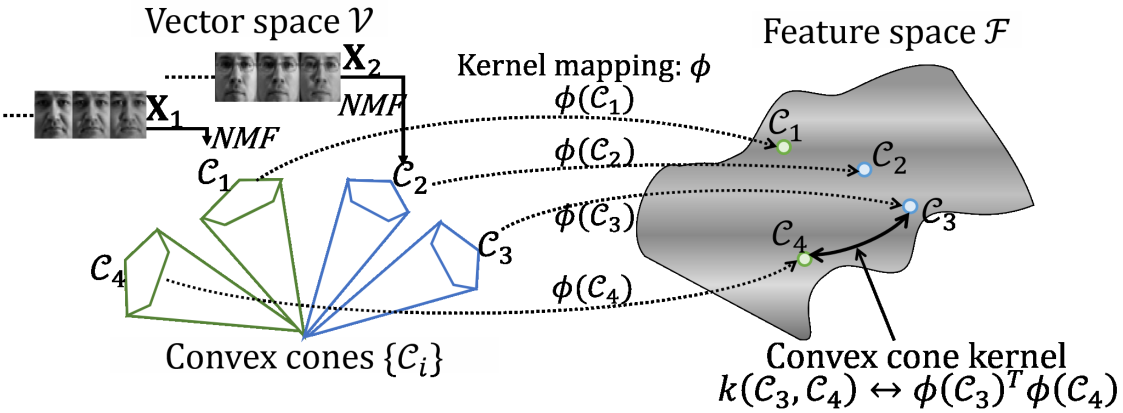


Figure 5.1: Conceptual diagram of the convex cone kernels. With the kernel trick, a set of convex cones is efficiently handled while preserving on the relationship between the convex cones. Since the kernel has positive definiteness, this operation can be regarded as the handling on the feature space $\mathscr{F}$ defined by the kernel function k, where ϕ is the mapping function defined by the kernel k.

2. Proposal of a visualization scheme for the relationship between convex cones, based on the proposed convex cone kernels, by using Kernel PCA.

3. Proposal of the cone discriminant analysis (CDA) and the enhanced CDA (eCDA).

4. Verification of the effectiveness of our convex cones representation framework.

5.2 Convex cone kernels

5.2.1 Definition of convex cone kernels

There are several options to measure the similarity between sets, such as using the maximum distance or the minimum distance [31, 1, 41]. The best choice highly depends on specific tasks or data distributions. To deal with various situations, we define three types of convex cone kernels based on the smallest angles $\{\theta_i\}_{i=1}^{r}$ between two convex cones $\mathscr{C}_1$ and $\mathscr{C}_2$ as follows:

· *Min angle kernel*

$$k_{min}(\mathscr{C}_1, \mathscr{C}_2) = \cos^2\theta_1. \tag{5.1}$$

· *Max angle kernel*

$$k_{max}(\mathscr{C}_1, \mathscr{C}_2) = \cos^2\theta_r. \tag{5.2}$$

· *Mean angle kernel*

$$k_{mean}(\mathscr{C}_1, \mathscr{C}_2) = \sum_{i=1}^{r}\cos^2\theta_i/r. \tag{5.3}$$

The above kernel functions approach 0 if the convex cones are orthogonal to each other, and approach 1 if the whole structure of the convex cones is the same. The k_{min} emphasizes the similar structure between $\mathscr{C}_1$ and $\mathscr{C}_2$, since the closest structure are analyzed in this kernel. On the other

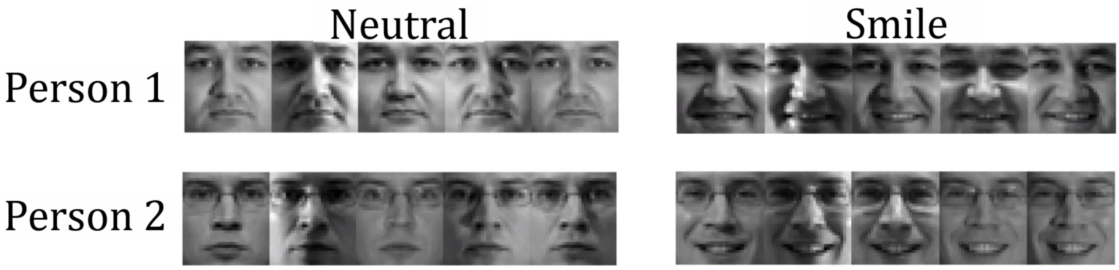


Figure 5.2: Examples of image-sets from CMU PIE. We used 5 images taken under various illumination conditions as an image-set.

hand k_{mean} exploits the centroids of $\mathscr{C}_1$ and $\mathscr{C}_2$, which provides information about the whole geometric structure of both $\mathscr{C}_1$ and $\mathscr{C}_2$. Finally, k_{max} take advantage of the most dissimilar structure contained in $\mathscr{C}_1$ and $\mathscr{C}_2$, which may be useful in many applications.

It is important to check whether a kernel function has positive definiteness. As the above kernels are clearly symmetric, to show that the proposed kernel functions have positive definiteness, we first prove that the mean angle kernel k_{mean} has positive definiteness as follows:

Proof. Let $\lambda_j, \lambda_k \in \mathbb{R}$, $\mathbf{p}_j^l, \mathbf{q}_k^l$ be a vector pair forming the l-th smallest angle $\theta_l^{j,k}$ between two convex cones $\mathscr{C}_j$, $\mathscr{C}_k$, and $\mathbf{Y}_j = [\mathbf{p}_j^1, \cdots, \mathbf{p}_j^r]$, $\mathbf{Y}_k = [\mathbf{q}_k^1, \cdots, \mathbf{q}_k^r]$, where $\|\mathbf{p}_j^l\|_2 = \|\mathbf{q}_k^l\|_2 = 1$.

$$\sum_{j,k}\lambda_j\lambda_k k_{mean}(\mathscr{C}_j,\mathscr{C}_k) = \sum_{j,k}\lambda_j\lambda_k\sum_{l}\cos^2\theta_l^{j,k}/r = \sum_{j,k}\lambda_j\lambda_k\|\mathbf{Y}_j^{\mathrm{T}}\mathbf{Y}_k\|_F^2/r$$

$$= \sum_{j,k}\lambda_j\lambda_k\mathbf{tr}(\mathbf{Y}_j\mathbf{Y}_j^{\mathrm{T}}\mathbf{Y}_k\mathbf{Y}_k^{\mathrm{T}})/r = \mathbf{tr}(\sum_{j}\lambda_j\mathbf{Y}_j\mathbf{Y}_j^{\mathrm{T}})^2/r = \|\sum_{j}\lambda_j\mathbf{Y}_j\mathbf{Y}_j^{\mathrm{T}}\|_F^2/r \geq 0. \tag{5.4}$$

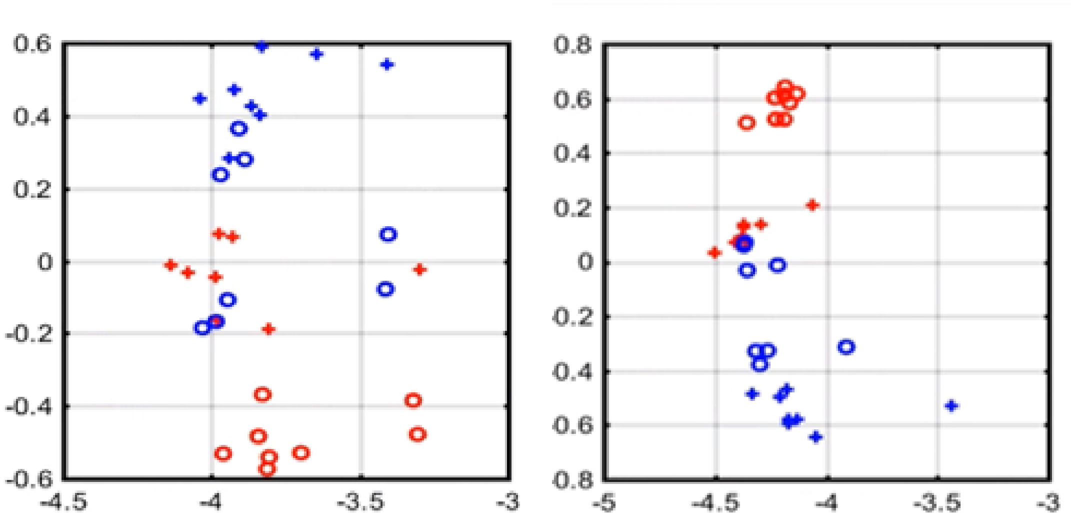
Therefore it is verified that the mean angle kernel is a positive-definite kernel function. □

The positive definiteness of the remaining two kernel functions can be proved by the above procedure, changing only $\mathbf{Y}_j$ and $\mathbf{Y}_k$. Thus, each kernel function can be considered as the inner product on the feature space that is defined by the kernel function, e.g., $k_{mean}(\mathscr{C}_1,\mathscr{C}_2) = \phi(\mathscr{C}_1)^{\mathrm{T}}\phi(\mathscr{C}_2)$, where $\phi(\mathscr{C}_i)$ is a mapping function to the feature space. We can apply kernel-based methods to a set of convex cones through our kernel functions. Besides, by assuming that the distance on the feature space is Euclidean like $\|\phi(\mathscr{C}_1) - \phi(\mathscr{C}_2)\|_2^2 = k_{mean}(\mathscr{C}_1,\mathscr{C}_1) + k_{mean}(\mathscr{C}_2,\mathscr{C}_2) - 2k_{mean}(\mathscr{C}_1,\mathscr{C}_2)$, we can apply distance-based manifold learning methods to a set of convex cones.

5.2.2 Visualization using convex cone kernels

To illustrate the applicability of the proposed kernels, we visualize the relationship between image-sets using the typical dimension reduction method, Kernel PCA (KPCA) [90], with the mean angle kernel.

In this experiment, we used front face images of 2 facial expressions of 2 people included in the CMU PIE dataset [83], i.e., we have four categories as shown in Fig.5.2. Each category has 40 images taken under various illumination conditions. Five images randomly selected from each set was used as an image-set. Thus, the number of image-sets is 32 (i.e. 4 (categories) × 40 (images)



(a) Subspace kernel.

(b) Mean angle kernel

Figure 5.3: Visualization results by Kernel PCA. Each point corresponds to an image-set. The color of each point indicates a person and the shape of each point indicates a facial expression.

/ 5). We generated a convex cone with 5 basis vectors from an image-set, and then applied KPCA to the set of convex cones. For the comparison, we also applied the methods to the set of subspaces that were generated from the image sets. For the subspaces, we used the projection kernel [31].

Figures 5.3 shows the visualization results. A point in each figure corresponds to one image set. The color of each point indicates a person, and the shape of each point indicates a facial expression. Although the subspace kernel k_{max} with KPCA causes overlapping between categories, k_{max} efficiently clusters the subjects. This suggests that our kernel can capture more discriminative information than the subspace-based method can, since the points referring to the two convex cone kernels make clusters according to the category. Compared with the subspace-based method, the proposed kernel function can be confirmed to be more effective.

5.3 Image-set based classification using convex cone kernels

In this section, we describe the details of the image-set based classification framework based on the convex cone discriminant analysis (CDA) and the enhanced CDA (eCDA).

5.3.1 Convex cone discriminant analysis

We construct a convex cone discriminant analysis based on the proposed kernels, by combining them with kernel discriminant analysis (KDA).

Let $\{\mathscr{C}_1, \ldots, \mathscr{C}_N\}$ be the convex cones and $\{y_1, \ldots, y_N\}$ be the class labels $y_i \in \{1, \ldots, C\}$, and N_c be the number of convex cones of the c-th class data. Without loss of generality, we assume that the data are ordered according to the class labels: $1 = y_1 \leq y_2 \leq \cdots \leq y_N$. CDA is formulated as

$$\max \frac{\boldsymbol{\alpha}^\top \boldsymbol{K}(\boldsymbol{V} - \boldsymbol{e}_N \boldsymbol{e}_N^\top / N) \boldsymbol{K} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top (\boldsymbol{K}(\boldsymbol{I}_N - \boldsymbol{V})\boldsymbol{K} + \sigma^2 \boldsymbol{I}_N)\boldsymbol{\alpha}}, \tag{5.5}$$

where $\boldsymbol{K}$ is the kernel matrix whose elements $(\boldsymbol{K})_{i,j}$ is calculated by the proposed kernel function, $\boldsymbol{e}_N$ is a vector of ones of length N, $\boldsymbol{V}$ is a block-diagonal matrix whose c-th block is the matrix $\boldsymbol{e}_{N_c}\boldsymbol{e}_{N_c}^\top / N_c$, and $\boldsymbol{\Sigma}_b = \boldsymbol{K}(\boldsymbol{V} - \boldsymbol{e}_N\boldsymbol{e}_N^\top/N)\boldsymbol{K}$. The term $\sigma^2 \boldsymbol{I}_N$ is used to regularize the covariance matrix $\boldsymbol{\Sigma}_w = \boldsymbol{K}(\boldsymbol{I}_N - \boldsymbol{V})\boldsymbol{K}$. It is composed of the covariance shrinkage factor $\sigma^2 > 0$, and the identity matrix $\boldsymbol{I}_N$ of size N. The $C-1$ optimal directions $\{\boldsymbol{\alpha}_i\}$ can be obtained as the eigenvectors of $(\boldsymbol{\Sigma}_w + \sigma^2 I_N)^{-1}\boldsymbol{\Sigma}_b$.

Based on these optimal directions, we construct the classification framework which consists of a training phase and a recognition phase as follows.

Training phase

Given training N image-sets $\{\mathbf{X}_i \in \mathbb{R}^{d \times N_i}\}_{i=1}^N$, where each set has N_i images, $\mathbf{X}_i = [\mathbf{x}_1^i, \ldots, \mathbf{x}_{N_i}^i]$, $\mathbf{x}_j^i \in \mathbb{R}^d$ is a feature vector extracted from an image, $\{y_1, \ldots, y_N\}$ is the class label, and $y_i \in \{1, \ldots, C\}$.

1. Generate convex cones $\mathscr{C}_i$ by applying NMF to the training data $\mathbf{X}_i$.

2. Generate optimal directions $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{C-1}]$ by solving optimization problem Eq.(5.5).

3. Obtain feature vectors $\{\mathbf{f}_i\}$ as $\mathbf{f}_i = \mathbf{A}^\mathrm{T}\mathbf{k}_i$, where $\mathbf{k}_i$ is a vector whose j-th element is calculated by a convex cone kernel using $\mathscr{C}_i$ and $\mathscr{C}_j$.

Recognition phase

Given input feature set $\mathbf{X}^{in} = [\mathbf{x}_1^{in}, \ldots, \mathbf{x}_{N_{in}}^{in}]$.

1. Generate input convex cone $\mathscr{C}^{in}$ by applying NMF to the input set $\mathbf{X}^{in}$

2. Obtain a feature vector $\mathbf{f}^{in}$ as $\mathbf{f}^{in} = \mathbf{A}^\mathrm{T}\mathbf{k}^{in}$, where $\mathbf{k}^{in}$ is a vector whose j-th element is calculated by a convex cone kernel using $\mathscr{C}_j$ and $\mathscr{C}^{in}$.

3. Classify the input based on the nearest neighbor method using $\mathbf{f}^{in}$ and $\{\mathbf{f}_i\}$.

5.3.2 Enhanced convex cone discriminant analysis

CDA can be considered as a linear discriminant analysis on the feature space defined by the kernel function. To further enhance the classification performance of the CDA-based framework, we introduce the projection onto a discriminant space. The projection is the operation on the original space to extract the difference between classes. Thus, the projection nearly orthogonalizes convex cones between classes. Consequently, projected convex cones can produce a set of more discriminant features, in comparison with the original convex cones.

The classification framework based on the enhanced CDA (eCDA) is as follows:

Training phase

Given training N image sets $\{\mathbf{X}_i \in \mathbb{R}^{d \times N_i}\}_{i=1}^N$, where each set has N_i images, $\mathbf{X}_i = [\mathbf{x}_1^i, \cdots, \mathbf{x}_{N_i}^i]$, $\mathbf{x}_j^i \in \mathbb{R}^d$ is a feature vector extracted from an image, $\{y_1, \cdots, y_N\}$ are the class labels and $y_i \in \{1, \cdots, C\}$.

1. Generate a convex cone $\mathscr{C}_i$ by applying NMF to the training data $\mathbf{X}_i$.

2. Generate class convex cones $\hat{\mathscr{C}_c}$ by applying NMF to convex cones $\{\mathscr{C}_i | y_i = c\}$ of a class.

3. Generate a discriminant $\mathscr{D}$ space by the method of [95] using $\hat{\mathscr{C}_c}$.

4. Project each convex cone $\mathscr{C}_i$ onto the discriminant space $\mathscr{D}$. We denote projected convex cones as $\mathscr{C}_i^*$.

5. Obtain optimal directions $\mathbf{A}$ and feature vectors $\{\mathbf{f}_i\}$ by steps 2 and 3 in the CDA's training phase with $\mathscr{C}_i^*$.

Recognition phase

Given input feature set $\mathbf{X}^{in} = [\mathbf{x}_1^{in}, \cdots, \mathbf{x}_{N_{in}}^{in}]$.

1. Generate input convex cone $\mathscr{C}^{in}$ by applying NMF to the input set $\mathbf{X}^{in}$

2. Project the input convex cone $\mathscr{C}^{in}$ onto the discriminant space $\mathscr{D}$. We denote projected input convex cone as $\mathscr{C}^{in*}$

3. Obtain a feature vector $\mathbf{f}^{in}$ as $\mathbf{f}^{in} = \mathbf{A}^{\mathrm{T}}\mathbf{k}_{in}$, where $\mathbf{k}_{in}$ is a vector whose j-th element is calculated by a convex cone kernel using $\mathscr{C}_j^*$ and $\mathscr{C}^{in*}$.

4. Classify the input based on nearest neighbor method using $\mathbf{f}^{in}$ and $\{\mathbf{f}_i\}$.

5.4 Evaluation experiments

In this section, we demonstrate the effectiveness of the proposed methods through experiments on action recognition using videos and object recognition using multi-view images. Through the action recognition experiments, we show the effectiveness of the proposed methods by comparing with Grassmann manifold based discriminant analysis (GDA)[31] and enhanced GDA (eGDA)[3] as baselines, since they have been widely known as one of the powerful classification methods based on subspace representation using a kernel function. Then, through the object recognition experiments, we evaluate the effectiveness by comparing with various conventional methods.

5.4.1 Action recognition

Experimental protocol

We conducted an action recognition experiments using the KTH action dataset [96]. The KTH dataset consists of 6 classes of actions. Each action was performed by 25 subjects in videos, filmed under 4 different shooting conditions: outdoors, outdoors with variations of zooming, outdoors with different clothes, and indoors. We conducted 10-fold cross-validation, with 10 subjects randomly selected for training and the remaining subjects for testing. For each sequence, we used the bounding box from [97] to do segmentation between actions. Each frame was resized to a 16×16 pixels and converted to grayscale. We used the raw pixel values with additional information of the height and width of the bounding box of the subject, resulting in a 258-dimensional vector for each frame.

For feature vectors $\mathbf{f}_i$, we used time elastic (TE) features [98], since it was confirmed that the TE feature is suitable for subspace-based method [99]. A TE feature can be obtained by concatenating multiple frames randomly selected from a sequence while retaining the original order in the sequence. By repeating the random sampling, we can obtain a set of TE features from a sequence. A subspace or convex cone of a sequence is generated from a set of TE features of a sequence.

For the experiment, 5 frames were randomly sampled from a sequence to generate the vector, i.e., the dimension of a feature vector is 1290 (i.e. 5×258), and 100 TE features extracted from a sequence. The number of basis vectors of a convex cone varied from 5 to 25 in increments of 5, and the number of basis vectors of each class convex cones varied from 30 to 90 in increments of 30.

As another baseline method, we conducted the nearest neighbor method using the mean angle kernel as the similarity. In the following, we refer to this method as "Cone-1NN".

Table 5.1: Experimental results (recognition rate (%), standard deviation) for the action recognition.

	KTH
GDA	67.56±1.70
eGDA	69.17±1.30
Cone-1NN	59.54±2.36
CDA with k_{min}	52.36±2.45
CDA with k_{max}	42.60±6.26
CDA with k_{mean}	69.81±1.51
eCDA with k_{min}	61.38±2.84
eCDA with k_{max}	48.22±2.04
eCDA with k_{mean}	**72.95±1.70**

Results and discussion

Table 5.1 lists the experimental results. The methods using the mean angle kernel k_{mean} showed superior results to the other proposed kernel functions. This result may suggest that it is important to compare the whole structure of convex cones in this experimental situation. Moreover, the k_{mean} based methods showed superior results to the subspace-based methods, GDA/eGDA. These results indicate that the convex cone representation works effectively for the problems, and the mean angle kernel k_{mean} is appealing for measuring the structure information of convex cones.

As CDA and eCDA with k_{mean} outperformed Cone-1NN, it was confirmed that these methods generate suitable directions for the classification based on convex cone representation. As can be seen in Table 5.1, eCDA outperformed CDA. This result supports the validity of our another core idea: projecting onto a discriminant space helps to improve the classification ability of CDA.

5.4.2 Object recognition

Experimental protocol

In this experiment, we used the ETH-80 dataset [67]. The ETH-80 dataset consists of eight different categories, captured from 41 viewpoints. Each category has ten types of objects. Five objects randomly sampled from each category were used for training, and the remaining objects were used for testing. As an input image set, we used 41 multi-view images for each object. To conduct a consistent experiment with previous works, we used images scaled to 32 × 32 pixels [100, 101]. For the feature vector, we used pixel intensities, i.e., vectorized images $\mathbf{f}_i \in \mathbb{R}^{1024}$ was used. We evaluated the classification performance of each method in terms of the average accuracy of ten trials using randomly-split datasets.

The number of basis vectors of a convex cone varied from 5 to 15 in increments of 5, and the number of basis vectors of each class convex cones varied from 20 to 80 in increments of 20.

Table 5.2: Experimental results (recognition rate (%), standard deviation) for object recognition.

	ETH-80
DCC[24]	91.75±3.74
MMD[76]	77.50±5.00
CHISD[41]	79.53±5.32
MMDML[102]	94.50±3.50
ADNT[100]	**98.12±1.69**
PLRC[103]	87.72±5.67
Reconst. Model [101]	94.75±4.32
GDA[31]	87.55±6.40
eGDA[99]	95.00±2.64
Cone-1NN	88.25±4.72
CDA with k_{min}	56.00±22.52
CDA with k_{max}	64.25±6.46
CDA with k_{mean}	93.00±3.87
eCDA with k_{min}	83.75±6.04
eCDA with k_{max}	70.50±8.32
eCDA with k_{mean}	**97.00**±2.58

Results and discussion

Table 5.2 shows the experiments results including those from various conventional methods. First of all, consistently with the action recognition experiment, we can see that the proposed methods, CDA and eCDA with k_{mean}, performed better than the subspace-based methods, GDA and eGDA. These results also suggest that the proposed methods based on the mean angle kernel significantly capture useful information for the classification using convex cone representation. In addition, eCDA achieved superior result to CDA. From this result, we could confirm the effectiveness of eCDA again.

The proposed methods also achieved the same or superior results compared with other conventional methods. In particular, since the essence of the proposed methods is very simple, it is impressive that we could achieve competitive results compared with deep neural network (DNN) based methods, such as MMDML, ADNT, and Reconst. model, although the essence of the proposed methods is much simpler. This also indicates the effectiveness of the proposed kernel function.

5.5 Summary

In this chapter, we proposed three types of convex cone kernels based on the smallest angles between convex cones for effectively handling image-set data. After verifying that each kernel is a positive definite function, we conducted visualization experiments using Kernel PCA with the proposed kernel. The visualization illustrated the effectiveness of the proposed kernel compared with the projection kernel, the kernel function for subspaces.

We also used our proposed kernels to construct an image-set based classification framework, which is termed convex cone discriminant analysis (CDA). Furthermore, to enhance the classification performance of CDA, we introduced the projection onto a discriminant space before mapping the feature space defined by a convex cone kernel, as the projection can emphasize the difference between classes. As a result, we constructed a more powerful classification framework based on the enhanced CDA (eCDA). We verified the effectiveness of CDA and eCDA through two experiments. Our results confirmed not only that the proposed methods outperform the subspace-based methods, but also that they can achieve competitive results with much more complex methods using deep neural networks.

Chapter 6

Concluding remarks

This chapter summarizes the work described in this thesis and the future directions.

6.1 Summary

In this study, we introduced image set based classification methods using two data representations, subspaces and convex cones. The motivations of the proposed methods are that while the subspace representation has remarkable representation ability of a set, there are two large problems in the representation ability for the classification: 1) canonical angles are calculated in Euclidean space, i.e., there is large room for the improvement of the representation ability in which an identity matrix is naively used as the metric matrix, and 2) subspace representation is a rough approximation of convex cone representation. The goal was to solve these problems and develop algorithms with high classification performance.

In chapter 3, we proposed the metric learning method for subspace representations. The key idea behind this method is to introduce a metric space in the calculation of the canonical angles and learn the metric space to produce discriminative subspace representations. To this end, we differentiated the calculation of the canonical angles and use it to minimize our cost function by the Riemannian conjugate gradient method. Besides, we introduced the low-rank constraint on the metric matrix to automatically extract essential information for the classification. We demonstrated the effectiveness of the proposed methods through the extensive experiments.

In chapter 4, we introduced two image set classification methods using novel representation, a convex cone. The reason for introducing convex cone representations is that the cone representation has a high probability to produce exact class boundaries due to its accurate representation ability compared with a subspace. To construct a convex cone based method, we defined multiple angles between convex cones and used it for calculating the similarity between two convex cones. Furthermore, we introduced two types of discriminant spaces, which enlarge the gaps among convex cones to increase the performance of the framework using convex cone representations.

In chapter 5, we defined three types of cone kernels and constructed two classification methods using the concept of Fisher discriminant analysis, to efficiently handle a set of convex cones. We showed the effectiveness of the proposed cone-based methods through object classification experiments using two datasets.

6.2 Future work

Although we validated the effectiveness of the proposed methods, there are still many challenges to be tackled in the future.

The metric learning methods proposed in chapter 3 is the first pursuit of designing a metric space for measuring canonical angles based on $\mathbf{A}$-based scalar product. As this approach can fully utilize the geometry of multiple subspaces and is fundamentally different from conventional methods, which are designed to utilize standard canonical angles efficiently, we think that our methods can offer various possible future directions, e.g., the combination of the proposed method with the kernel-based subspace learning method [31, 55], or to incorporate the success of the metric learning for vector data [104, 105]. Besides, we will consider embedding the proposed learning method in the learning process of neural networks by using the derived gradient of the subspace similarity.

In chapter 4 and 5, we proposed the fundamental convex cone-based frameworks by defining the similarity between convex cones. In the future, we will further explore the development of 1) a novel discriminant space for convex cones by incorporating recent advances in image feature extraction methods, such as [106, 107, 108], 2) a novel learning algorithms for convex cones by incorporating the development of the subspace-based methods, and 3) a fast calculation method of the smallest angles between convex cones. Besides, we will seek the theoretical relationship between subspace and cone-based methods by using the canonical angles in a metric space as an interface.

Subspace-based methods have a wide range of applications, such as text classification and acoustic classification [109, 110, 111]. Although all proposed methods were constructed for image set classification, the proposed methods can be applied to these applications. We will consider applying our method to new fields and developing further extensions.

Bibliography

[1] Osamu Yamaguchi, Kazuhiro Fukui, and Kenichi Maeda. Face recognition using temporal image sequence. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 318–323, 1998.

[2] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

[3] Lincon Souza, Hideitsu Hino, and Kazuhiro Fukui. 3d object recognition with enhanced grassmann discriminant analysis. In *ACCV 2016 Workshop (HIS 2016)*, 2016.

[4] Lincon S. Souza, Bernardo B. Gatto, Jing-Hao Xue, and Kazuhiro Fukui. Enhanced grassmann discriminant analysis with randomized time warping for motion recognition. *Pattern Recognition*, 97:107028, 2020.

[5] Zhong-Qiu Zhao, Shou-Tao Xu, Dian Liu, Wei-Dong Tian, and Zhi-Da Jiang. A review of image set classification. *Neurocomputing*, 335:251–260, 2019.

[6] Yulian Zhu and Jing Xue. Face recognition based on random subspace method and tensor subspace analysis. *Neural Computing and Applications*, 28(2):233–244, 2017.

[7] Dong Wei, Xiaobo Shen, Quansen Sun, Xizhan Gao, and Wenzhu Yan. Prototype learning and collaborative representation using grassmann manifolds for image set classification. *Pattern Recognition*, 100:107123, 2020.

[8] Guoqing Zhang, Junchuan Yang, Yuhui Zheng, Zhiyuan Luo, and Jinglin Zhang. Optimal discriminative feature and dictionary learning for image set classification. *Information Sciences*, 547:498–513, 2021.

[9] Ting-Yao Hu and Alexander G. Hauptmann. Statistical Distance Metric Learning for Image Set Retrieval. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1765–1769, 2021.

[10] Jun Xu, Wangpeng An, Lei Zhang, and David Zhang. Sparse, collaborative, or nonnegative representation: Which helps pattern classification? *Pattern Recognition*, 88:679–688, 2019.

[11] Bo Liu, Liping Jing, Jia Li, Jian Yu, Alex Gittens, and Michael W Mahoney. Group collaborative representation for image set classification. *International Journal of Computer Vision*, 127(2):181–206, 2019.

[12] W Wang, R Wang, Z Huang, S Shan, and X Chen. Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets. *IEEE Transactions on Image Processing*, 27(1):151–163, 2018.

[13] Rui Wang, Xiao-Jun Wu, Kai-Xuan Chen, and Josef Kittler. Multiple Riemannian Manifold-valued Descriptors based Image Set Classification with Multi-Kernel Metric Learning. *IEEE Transactions on Big Data*, pages 1–1, 2020.

[14] Lei Luo, Jie Xu, Cheng Deng, and Heng Huang. Robust Metric Learning on Grassmann Manifolds with Generalization Guarantees. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4480–4487, July 2019.

[15] Mehrtash T Harandi, Mathieu Salzmann, and Richard Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In *European conference on computer vision*, pages 17–32. Springer, 2014.

[16] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[17] A theoretical study of pattern recognition by matching method.

[18] Matthew Turk and Alex Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 01 1991.

[19] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[20] Peter N. Belhumeur and David J. Kriegman. What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision*, 28(3):245–260, 1998.

[21] Kuang-Chih Lee, Jeffrey Ho, and David J Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

[22] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[23] Ken-ichi Maeda, Osamu Yamaguchi, and Kazuhiro Fukui. Towards 3-dimensional pattern recognition. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 1061–1068. Springer, 2004.

[24] Tae-Kyun Kim, Josef Kittler, and Roberto Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1005–1018, 2007.

[25] Kazuhiro Fukui and Osamu Yamaguchi. Face recognition using multi-viewpoint patterns for robot vision. In *International Symposium of Robotics Research*, pages 192–201, 2005.

[26] Masashi Nishiyama, Osamu Yamaguchi, and Kazuhiro Fukui. Face recognition with the multiple constrained mutual subspace method. In Takeo Kanade, Anil Jain, and Nalini K. Ratha, editors, *Audio- and Video-Based Biometric Person Authentication*, pages 71–80, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[27] Tae-Kyun Kim, Josef Kittler, and Roberto Cipolla. Incremental learning of locally orthogonal subspaces for set-based object recognition. In *BMVC*, pages 559–568, 2006.

[28] Kazuhiro Fukui and Osamu Yamaguchi. The kernel orthogonal mutual subspace method and its application to 3D object recognition. In *Asian Conference on Computer Vision*, pages 467–476, 2007.

[29] Hitoshi Sakano and Naoki Mukawa. Kernel mutual subspace method for robust facial image recognition. In *International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, volume 1, pages 245–248, 2000.

[30] Kazuhiro Fukui, Björn Stenger, and Osamu Yamaguchi. A framework for 3D object recognition using the kernel constrained mutual subspace method. In *Asian Conference on Computer Vision*, pages 315–324. 2006.

[31] Jihun Hamm and Daniel D Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *International Conference on Machine Learning*, pages 376–383. ACM, 2008.

[32] Jihun Hamm and Daniel Lee. Extended grassmann kernels for subspace-based learning. *Advances in neural information processing systems*, 21:601–608, 2008.

[33] Mehrtash Tafazzoli Harandi, Mathieu Salzmann, Sadeep Jayasumana, Richard I. Hartley, and Hongdong Li. Expanding the Family of Grassmannian Kernels: An Embedding Perspective. In *European Conference on Computer Vision*, 2014.

[34] Hengliang Tan, Zhengming Ma, Sumin Zhang, Zengrong Zhan, Beibei Zhang, and Chenggong Zhang. Grassmann manifold for nearest points image set classification. *Pattern Recognition Letters*, 68:190–196, 2015.

[35] Mehrtash Harandi, Conrad Sanderson, Chunhua Shen, and Brian C. Lovell. Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution. In *International Conference on Computer Vision*, December 2013.

[36] Mehrtash Harandi, Richard Hartley, Chunhua Shen, Brian Lovell, and Conrad Sanderson. Extrinsic methods for coding and dictionary learning on grassmann manifolds. *International Journal of Computer Vision*, 114(2):113–136, 2015.

[37] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on Grassmann manifold with application to video based face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 140–149, 2015.

[38] Pengfei Zhu, Hao Cheng, Qinghua Hu, Qilong Wang, and Changqing Zhang. Towards generalized and efficient metric learning on Riemannian manifold. In *International Joint Conference on Artificial Intelligence*, pages 3235–3241, 2018.

[39] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on Grassmann manifolds. In *AAAI Conference on Artificial Intelligence*, 2018.

[40] Lincon S. Souza, Naoya Sogi, Bernardo B. Gatto, Takumi Kobayashi, and Kazuhiro Fukui. An interface between grassmann manifolds and vector spaces. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[41] Hakan Cevikalp and Bill Triggs. Face recognition based on image sets. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2567–2573. IEEE, 2010.

[42] Ruiping Wang, Huimin Guo, Larry S Davis, and Qionghai Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2496–2503. IEEE, 2012.

[43] Yiqun Hu, Ajmal S Mian, and Robyn Owens. Sparse approximated nearest points for image set classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 121–128. IEEE, 2011.

[44] Zhiwu Huang, Ruiping Wang, Shiguang Shan, Xianqiu Li, and Xilin Chen. Log-Euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *International Conference on Machine Learning*, pages 720–729, 2015.

[45] Yiqun Hu, Ajmal S Mian, and Robyn Owens. Face recognition using sparse approximated nearest points between image sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1992–2004, 2012.

[46] Hakan Cevikalp and Golara Ghorban Dordinejad. Discriminatively learned convex models for set based face recognition. In *International Conference on Computer Vision*, pages 10123–10132, 2019.

[47] Rui Wang, Xiao-Jun Wu, and Josef Kittler. Symnet: A simple symmetric positive definite manifold deep learning method for image set classification. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2021.

[48] Zhiwu Huang and Luc Van Gool. A riemannian network for spd matrix learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[49] Sydney N Afriat. Orthogonal and oblique projectors and the characteristics of pairs of vector spaces. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 53, pages 800–816, 1957.

[50] Takumi Kobayashi and Nobuyuki Otsu. Cone-restricted subspace methods. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[51] Kazuhiro Fukui and Atsuto Maki. Difference subspace and its generalization for subspace-based methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2164–2177, 2015.

[52] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

[53] Hyunsoo Kim and Haesun Park. Nonnegative matrix factorization based on alternating non-negativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730, 2008.

[54] Rasmus Bro and Sijmen De Jong. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11(5):393–401, 1997.

[55] Mehrtash T Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C Lovell. Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2705–2712. IEEE, 2011.

[56] Hengliang Tan, Ying Gao, and Zhengming Ma. Regularized constraint subspace based method for image set classification. *Pattern Recognition*, 76:434–448, 2018.

[57] Andrew V Knyazev and Merico E Argentati. Principal angles between subspaces in an A-based scalar product: algorithms and perturbation estimates. *SIAM Journal on Scientific Computing*, 23(6):2008–2040, 2002.

[58] Erkki Oja and Maija Kuusela. The ALSM algorithm—an improved subspace method of classification. *pattern Recognition*, 16(4):421–427, 1983.

[59] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[60] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[61] Hiroyuki Sato. A dai–yuan-type riemannian conjugate gradient method with the weak wolfe conditions. *Computational optimization and Applications*, 64(1):101–118, 2016.

[62] M. Sugiyama. Local Fisher discriminant analysis for supervised dimensionality reduction. In *International Conference on Machine Learning*, pages 905–912. ACM, 2006.

[63] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[64] Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[65] Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[66] Maryam Fazel, Haitham Hindi, Stephen P Boyd, et al. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, volume 6, pages 4734–4739. Citeseer, 2001.

[67] Bastian Leibe and Bernt Schiele. Analyzing appearance and contour based methods for object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 409–415, 2003.

[68] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *IEEE International Conference on Robotics and Automation*, pages 1817–1824. IEEE, 2011.

[69] Minyoung Kim, Sanjiv Kumar, Vladimir Pavlovic, and Henry Rowley. Face tracking and recognition with visual constraints in real-world videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[70] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 529–534. IEEE, 2011.

[71] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition workshops*, pages 806–813, 2014.

[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[73] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[74] Paul Viola and Michael J Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[75] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *IEEE International Conference on Automatic Face & Gesture recognition*, pages 67–74. IEEE, 2018.

[76] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based on image set. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[77] Michel Tenenhaus. Canonical analysis of two convex polyhedral cones and applications. *Psychometrika*, 53(4):503–524, 1988.

[78] Javier Vía, Ignacio Santamaría, and Jesús Pérez. Canonical correlation analysis (CCA) algorithms for multiple data sets: Application to blind SIMO equalization. In *European Signal Processing Conference*, pages 1–4, 2005.

[79] Javier Vía, Ignacio Santamaría, and Jesús Pérez. A learning algorithm for adaptive canonical correlation analysis of several data sets. *Neural Networks*, 20(1):139–152, 2007.

[80] Masashi Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, 8(May):1027–1061, 2007.

[81] Ingwer Borg and Patrick Groenen. Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement*, 40(3):277–280, 2003.

[82] Yasuhiro Ohkawa and Kazuhiro Fukui. Hand-shape recognition using the distributions of multi-viewpoint image sets. *IEICE Transactions on Information and Systems*, 95(6):1619–1627, 2012.

[83] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[84] P Ekman, W Friesen, and J Hager. Facial action coding system: Research nexus. *Network Research Information, Salt Lake City, UT*, 1, 2002.

[85] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended Cohn-Kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *IEEE Conference on Computer Vision and Pattern Recognition - Workshops*, pages 94–101. IEEE, 2010.

[86] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[87] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[88] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[89] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

[90] Sebastian Mika, Bernhard Schölkopf, Alex J Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel PCA and de-noising in feature spaces. In *Advances in Neural Information Processing Systems*, pages 536–542, 1999.

[91] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188, 1936.

[92] Gaston Baudat and Fatiha Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404, 2000.

[93] Yongmin Li, Shaogang Gong, and Heather Liddell. Recognising trajectories of facial identities using kernel discriminant analysis. *Image and Vision Computing*, 21(13-14):1077–1086, 2003.

[94] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX*, pages 41–48. IEEE, 1999.

[95] N. Sogi, T. Nakayama, and K. Fukui. A method based on convex cone model for image-set classification with CNN features. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2018.

[96] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local SVM approach. In *International Conference on Pattern Recognition*, volume 3, pages 32–36. IEEE, 2004.

[97] Zhuolin Jiang, Zhe Lin, and Larry Davis. Recognizing human actions by learning and matching shape-motion prototype trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):533–547, 2012.

[98] Chendra Hadi Suryanto, Jing-Hao Xue, and Kazuhiro Fukui. Randomized time warping for motion recognition. *Image and Vision Computing*, 54:1–11, 2016.

[99] Lincon S. Souza, Bernardo Bentes Gatto, and Kazuhiro Fukui. Enhancing discriminability of randomized time warping for motion recognition. In *International Conference on Machine Vision Applications (MVA)*, pages 77–80. IEEE, 2017.

[100] Munawar Hayat, Mohammed Bennamoun, and Senjian An. Deep reconstruction models for image set classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):713–727, 2015.

[101] Syed Afaq Ali Shah, Uzair Nadeem, Mohammed Bennamoun, Ferdous Ahmed Sohel, and Roberto Togneri. Efficient image set classification using linear regression based image reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition - Workshops*, pages 601–610, 2017.

[102] Jiwen Lu, Gang Wang, Weihong Deng, Pierre Moulin, and Jie Zhou. Multi-manifold deep metric learning for image set classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1137–1145, 2015.

[103] Qingxiang Feng, Yicong Zhou, and Rushi Lan. Pairwise linear regression classification for image set retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4865–4872, 2016.

[104] Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[105] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

[106] Zhenwen Ren, Quansen Sun, Bin Wu, Xiaoqian Zhang, and Wenzhu Yan. Learning Latent Low-Rank and Sparse Embedding for Robust Image Feature Extraction. *IEEE Transactions on Image Processing*, 29:2094–2107, 2020.

[107] Haishun Du, Yuxi Wang, Fan Zhang, and Yi Zhou. Low-Rank Discriminative Adaptive Graph Preserving Subspace Learning. *Neural Processing Letters*, 52(3):2127–2149, 2020.

[108] Zheng Wang, Feiping Nie, Lai Tian, Rong Wang, and Xuelong Li. Discriminative feature selection via a structured sparse subspace learning module. In *International Joint Conference on Artificial Intelligence*, pages 3009–3015, 2020.

[109] Erica K. Shimomoto, Lincon S. Souza, Bernardo B. Gatto, and Kazuhiro Fukui. Text classification based on word subspace with term-frequency. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.

[110] Bernardo B. Gatto, Juan G. Colonna, Eulanda M. dos Santos, and Eduardo F. Nakamura. Mutual singular spectrum analysis for bioacoustics classification. In *IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2017.

[111] Lincon S. Souza, Bernardo B. Gatto, and Kazuhiro Fukui. Grassmann singular spectrum analysis for bioacoustics classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 256–260, 2018.

List of publications

1. Naoya Sogi, Rui Zhu, Jinghao Xue, Kazuhiro Fukui, “Constrained mutual convex cone method for image set based recognition”, Pattern Recognition, Elsevier, vol.121, pp.108-190, 2022.

2. Naoya Sogi, Lincon Souza, Bernardo Gatto, Kazuhiro Fukui, “Metric Learning with A-based Scalar Product for Image-set Recognition”, IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp.1-8, 2020.

3. Naoya Sogi, Taku Nakayama, Kazuhiro Fukui, “A Method Based on Convex Cone Model for Image-Set Classification with CNN Features”, International Joint Conference on Neural Networks, pp1-8, 2018

4. Naoya Sogi, Kazuhiro Fukui, “Action Recognition Method Based on Sets of Time Warped ARMA Models”, International Conference on Pattern Recognition, pp1773-1778, 2018.

5. 枌 尚弥，Lincon Souza, Bernardo Gatto, 福井 和広，“部分空間表現に基づく画像セット識別のための計量学習法”, 画像の認識・理解シンポジウム, 2020.

6. 枌 尚弥，Lincon Souza, Bernardo Gatto, Rui Zhu, Jinghao Xue, 福井 和広，“凸錐判別分析に基づく画像セットベース識別”, 画像の認識・理解シンポジウム, 2019.

7. 枌 尚弥, 福井 和広, “CNN特徴の凸錐表現に基づく画像セットベース識別”, 画像の認識・理解シンポジウム, 2018.

8. 枌 尚弥, 福井 和広, “錐制約差分部分空間を用いた特徴抽出とその応用”, 画像の認識・理解シンポジウム, 2017.