Fast and Accurate Face Recognition with Image Sets

Hakan Cevikalp Hasan Serhan Yavuz
Eskisehir Osmangazi University
Electrical and Electronics Engineering, Meselik, Eskisehir, Turkey
hakan.cevikalp@gmail.com, hsyavuz@ogu.edu.tr

Abstract

For large-scale face recognition applications using image sets, the images of the query set typically lie in compact regions surrounded by a diffuse sea of images of the gallery set. In this study, we propose a fast and accurate method to approximate the distances from gallery images to the region spanned by the query set for large-scale applications. To this end, we propose a new polyhedral conic classifier that will enable us to compute those distances efficiently by using simple dot products. We also introduce one-class formulation of the proposed classifier that can use query set examples only. This makes the method ideal for real-time applications since testing time approximately becomes the independent of the size of the gallery set. One-class formulation is very important for large-scale face recognition problems in the sense that it can be used in a cascade system with more complex and time-consuming methods to return the most promising candidate gallery sets in the first stage of the cascade so that more complex methods can be run on those a few candidate sets. As a result, we strongly believe that the proposed method will impact future methods and it will enable to introduce face recognition methods working in real-time even for large-scale set based recognition problems. Experimental results on both small and moderate sized face recognition datasets support these claims and demonstrate the efficacy of the proposed method. More precisely, the proposed methods achieve the best accuracies on all tested datasets and we obtained improvements around 18% compared to the best performing rival methods on larger datasets.

1. Introduction

Face recognition using image sets has gained significant attention in recent years. For set based face recognition, the user supplies a set of images of the same unknown individual rather than supplying a single query image. In general, the gallery also contains a set of images for each known individual; therefore, the system must recover the individual whose gallery set is the best match for a given query set. Methods based on image sets are expected to give better performance than ones based on single individual images because they incorporate information about the variability of the individual's appearance.

Existing set based classification methods mainly differ in the ways in which they represent the image sets and compute the distances (or similarity) between them. Based on the set model representation types, methods can be divided into two categories: parametric and non-parametric methods. Parametric methods such as [1, 21] used parametric probability distributions to model image sets, and the Kullback-Leibler divergence is used to measure the similarity between these distributions. Nonparametric methods, on the other hand, use different models (e.g., linear or affine subspaces, Grassmannian manifolds, or some different combinations of these) to approximate image sets. Based upon the type of representation, different metrics have been proposed for computing set-to-set distances.

As non-parametric recognition methods, Yamaguchi et al. [29] used linear subspaces to model image sets, and canonical angles between subspaces were used to measure the similarity between the subspaces. Another way of dealing with image set based classification is to consider each sample as a point in a Grassmannian manifold. Hamm and Lee [9] used Grassmannian discriminant analysis on fixed dimensional linear subspaces. Wang and Shi [25] proposed kernel Grassmannian distances to compare image sets. Grasmannian manifolds have been used for multi-view embedding in the context of image classification in [26]. More recently, manifolds of Symmetric Positive Definite matrices are used to model images sets, and the similarities between these manifolds are computed by using different Riemannian metrics such as Affine-Invariant metric or Log-Euclidean metric [17, 16].

Cevikalp and Triggs [2] introduced affine/convex hull models to approximate image sets, and geometric distances between these models are used to measure the similarity. This methodology offers a number of attractive properties: affine/convex hull models can better localize the image set

regions compared to linear subspaces; the model for each individual can be fitted independently; computing distances between models is straightforward due to the convexity, and resistance to outliers can be incorporated by using robust fitting to estimate convex models. Yalcin et al. [28] introduced a new method to speed-up the face recognition methods using the kernelized convex hulls. After introduction of affine/convex hull models, different variants of these models have been proposed [14, 30]. For example, SANP (Sparse Approximated Nearest Points) [14] methodology extended the affine hull method by enforcing the sparsity of samples used for affine hull combination. In a similar manner, [30] used regularized affine hull models to represent image sets where L2-norms of affine hull combination coefficients are minimized when computing the smallest distances between affine hulls. More recently, new extensions [27, 31] of these methods used the so-called collaborative representations for affine/convex hull models. The basic difference is that they model all gallery sets as a single affine/convex hull and then query set is classified by using the reconstruction residuals computed from only individual gallery sets. Other methods using sparse models for image set based recognition can be found in [6, 5, 4]. Most of the aforementioned methods have kernelized versions that can be used to estimate nonlinear face models.

More recently, deep neural networks have demonstrated a great success in visual object classification and feature learning. So, they have been used for image set based recognition [12, 10, 19]. Hayat et al. [12, 10] proposed a deep learning framework to estimate the nonlinear geometric structure of the image sets. They trained an Adaptive Deep Network Template for each set to learn the class-specific models and then the query set is classified based on the minimum reconstruction error computed by using those pre-learned class-specific models. Lu et al. [19] also use deep networks to model nonlinear face manifolds using face image sets as in [12, 10], and then they apply a learning algorithm to maximize the margin between different manifolds approximated by deep networks.

Our Contributions: In this study, we propose a novel method that is both very fast and accurate for face recognition using image sets. To this end, we introduce a polyhedral conic classifier that enables us to approximate distances from gallery face images to convex hulls of the query set images. The class assignment is made based on the closest distances from gallery image examples to the convex region spanned by query set images. The proposed classifier can also be used as one-class classifier that uses query set samples only. To improve the speed, we introduced a cascade where the first stage returns the closest candidate gallery images by using one-class classifier formulation, and the second stage returns the closest face image set by using binary-classifier that is run on the returned candidate classes and the query set. As a result, the proposed method can be used in real-life face recognition systems. More precisely, to the best of our knowledge, the proposed method is currently the fastest algorithm in the literature after the methods using linear subspaces [29], linear affine hulls [2], and linear hypersphere models [28]; and it significantly outperforms these methods in terms of the accuracy. Also, it has the potential to run faster than the subspace and affine hulls methods for large-scale problems since using one-class formulation makes the testing time of the proposed method independent of the gallery set size. We strongly believe that our method will impact future methods working on large-scale face recognition problems in such a way that it returns the most promising candidate gallery sets and rejects the majority of the gallery sets quickly, and more complex algorithms can be run on the returned image sets instead of using entire gallery sets.

2. METHOD

2.1. Motivation

In the proposed method, we model query face image sets by using linear convex hull approximations. Convex hulls are tighter models compared to linear subspaces and affine hulls, and they provide better localization in large-scale applications [28]. It should be noted that convex hulls are largely used to approximate data classes in other classification applications, e.g., the linear SVM uses convex hull modeling and the margin between two classes is equivalent to the geometric distance between convex hulls of classes. Once we approximate a query image set with a linear convex hull, we need to compute the distances from gallery images to the convex hull of the query image set. The distances can be found by solving a convex quadratic programming (QP) problem for each gallery image sample. This will be very slow and impractical for large gallery sets. Therefore, Cevikalp and Triggs [2] also approximate each gallery set with a convex hull and find the distances between the convex hulls of test and gallery image sets. This only requires solving C QP problems where C is the number of classes in the gallery set, and it is more practical than solving a QP problem for each image in the gallery.

For large-scale face recognition problems, the images of the query (test) set will typically lie in specific regions surrounded by a diffuse sea of face images of gallery sets as shown in Fig. 1. As aforementioned, finding distances from each gallery image to the region spanned by the query set is computationally expensive even for the cases when this region is modeled by using a linear convex hull. Our goal is to find an efficient and quick way to approximate the distances from gallery images to the region spanned by the query set. Hayat et al. [11, 13] use a linear SVM classifier for approximating distances between query and gallery sets. But, it

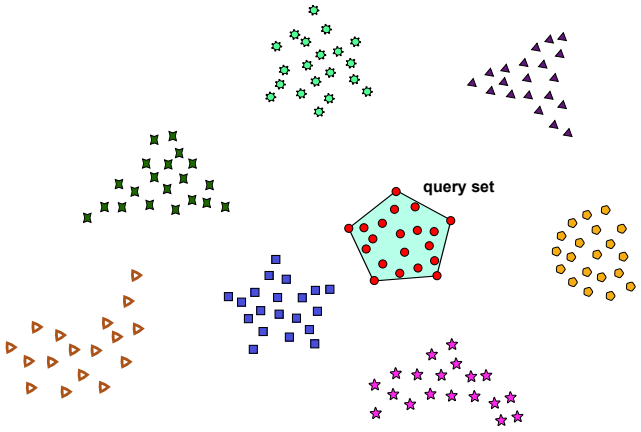

Figure 1. In large-scale applications, the query sets will be typically inside the convex hull formed by combining all image sets in the gallery. In such cases, joint or collaborative models [27, 31] and linear SVMs [11, 13] fail. The best strategy would be to approximate query set with a convex model and then find the distances from gallery images to this convex model. This can be efficiently done by using polyhedral conic classifiers.

should be noted a linear binary classifier using a hyperplane separation, e.g., a linear SVM classifier, cannot be used for this purpose when the query set images are surrounded by gallery images as visualized in Fig. 1 since the data are not separable by a hyperplane. Our experiments at the end support this fact. On the other hand, a nonlinear (kernelized) one-class or binary classifiers can be used to accomplish this task. For example, the region spanned by images of the query set can be approximated by using the kernelized one-class Support Vector Data Description (SVDD) method as described in [28]. But, this method is also not practical for real-time applications since one needs to evaluate kernel functions against all support vectors returned by the classifier. So, this will be also slow. Another problem would be related to training of the method with large-scale data since kernelized SVDD using SMO (Sequential Minimal Optimization) algorithm cannot scale well with the training set size. Our solution to this problem is to use a polyhedral conic classifier (PCC). As opposed to the linear SVMs, PCC can return polyhedral acceptance regions and finding distances in this setup can be easily accomplished by using a single dot product as described below as opposed to the computationally expensive kernel function evaluations.

2.2. Classification Based on Polyhedral Conic Functions

We build our methods based on polyhedral conic functions defined in [8]. [8] introduced polyhedral conic functions which are used to construct a separation function for the given two arbitrary finite point disjoint sets. These functions are formed as an augmented L1 norm with a linear part added. A graph of such a function is a polyhedral cone with a sub-level set including an intersection of at most 2^d half spaces. See Fig. 2 for visualization of different separation types.

Now, consider a binary classification problem with training data given in the form $\{\mathbf{x}_i, y_i\}$, $i = 1, \ldots, n$, and $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$. We first need following definition from [8].

Definition 1. A function $f(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$ is called polyhedral conic if its graph is a cone and all its level sets

$$S_\alpha = \left\{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) \leq \alpha\right\} \tag{1}$$

for $\alpha \in \mathbb{R}$, are polyhedrons.

We define a Polyhedral Conic function (PCF) $\underset{\mathbf{w},\mathbf{c},\gamma,b}{f(\mathbf{x})}$: $\mathbb{R}^d \rightarrow \mathbb{R}$ as

$$f(\mathbf{x}) = \mathbf{w}^\top(\mathbf{x} - \mathbf{c}) + \gamma||\mathbf{x} - \mathbf{c}||_1 - b, \quad \text{(PCF)} \tag{2}$$

where $\mathbf{w}, \mathbf{c} \in \mathbb{R}^d$, $\gamma, b \in \mathbb{R}$, and $||\mathbf{u}||_1 = |u_1| + \ldots + |u_d|$ is the l_1 norm of the vector $\mathbf{u} \in \mathbb{R}^d$. The fact that such a function defines a polyhedral cone follows from the following Lemma [8].

Lemma 2.1. *A graph of the function* $\underset{\mathbf{w},\mathbf{c},\gamma,b}{f(\mathbf{x})}$ *defined in (2) is a polyhedral cone with a vertex at* $(\mathbf{c}, -b)$.

Given these definitions, our goal is to find polyhedral conic functions whose level sets separate the positive samples from the negatives. To this end, we construct polyhedral regions such that each polyhedral region divides the input space into two parts such that most of the negative samples remain outside the polyhedral region whereas the majority of the positive class samples remain inside the region.

Instead of using Polyhedral Conic functions, we use our *Extended Polyhedral Conic Functions (EPCF)* [3] outperforming PCF in the form

$$f_{\mathbf{w},\boldsymbol{\gamma},\mathbf{c},b}(\mathbf{x}) = \mathbf{w}^\top(\mathbf{x} - \mathbf{c}) + \boldsymbol{\gamma}^\top|\mathbf{x} - \mathbf{c}| - b \quad \text{(EPCF)} \tag{3}$$

Here $\mathbf{x} \in \mathbb{R}^d$ is a test point, $\mathbf{c} \in \mathbb{R}^d$ is the cone vertex, $\mathbf{w} \in \mathbb{R}^d$ is a weight vector and b is an offset, and $|\mathbf{u}| = (|u_1|, ..., |u_d|)^\top$ denotes the component-wise modulus and $\boldsymbol{\gamma} \in \mathbb{R}^d$ is a corresponding weight vector.

Our polyhedral conic classifiers use EPCF, with decision regions $f(\mathbf{x}) < 0$ for positives and $f(\mathbf{x}) > 0$ for negatives. It should be noted that it is the opposite of popular SVM decision rule. Similarly, our margin based training methods enforce $f(\mathbf{x}) \leq -1$ for positives and $f(\mathbf{x}) \geq +1$ for negatives. For both PCF and EPCF, the positive region is essentially a hyperplane-section through an L_1 cone centered at $\mathbf{c}$, specifically the region $\mathbf{x} \in \mathbb{R}^d$ in which the

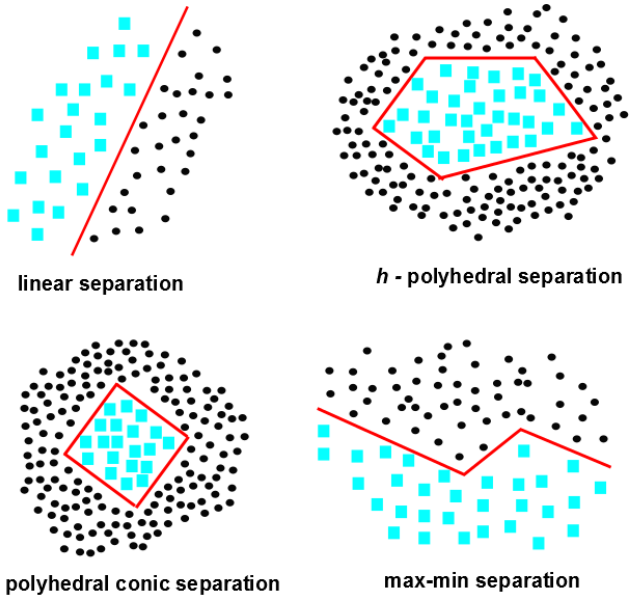

Figure 2. Geometric interpretation of four separation types: linear, h-polyhedral, polyhedral conic and max-min separation. In this study, we use polyhedral conic separation.

hyperplane $z = \mathbf{w}^\top(\mathbf{x}-\mathbf{c}) - b$ lies above the L_1 cone $z = \gamma\|\mathbf{x}-\mathbf{c}\|_1$ (PCF) or the diagonally-scaled L_1 cone $z = \boldsymbol{\gamma}^\top|\mathbf{x}-\mathbf{c}| = \|\mathrm{diag}(\boldsymbol{\gamma})\,(\mathbf{x}-\mathbf{c})\|_1$ (EPCF).

Note that for EPCF with $b > 0$, $\boldsymbol{\gamma} > 0$, $|w_i| < \gamma_i, i = 1, ..., d$, and any τ, the region $f(\mathbf{x}) < \tau$ is convex and compact, and it contains $\mathbf{c}$. It would be straightforward to enforce these inequalities during learning, but at present we simply leave the decision regions free to adapt to the training data; compact query sets surrounded by many gallery set images naturally tend to produce compact acceptance regions in any case.

To define margin-based classifiers over input feature vectors $\mathbf{x}$ from this, for EPCF we augment the feature vector to $\tilde{\mathbf{x}} \equiv \begin{pmatrix} \mathbf{x}-\mathbf{c} \\ |\mathbf{x}-\mathbf{c}| \end{pmatrix} \in \mathbb{R}^{2d}$ and the weight vector to $\tilde{\mathbf{w}} \equiv \begin{pmatrix} \mathbf{w} \\ -\boldsymbol{\gamma} \end{pmatrix} \in \mathbb{R}^{2d}$ and let $\tilde{b} = b$, giving the SVM form $\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}} + \tilde{b} > 0$ for positives, but now in $2d$ dimensions. The above ∓ 1 margins for EPCF translate to the familiar ± 1 SVM margins, allowing us to use standard SVM software for maximum margin training[1]. It thus suffices to run the familiar SVM quadratic program on the augmented feature vectors:

$$\begin{aligned} \arg\min_{\tilde{\mathbf{w}},\tilde{b}} \;& \tfrac{1}{2}\tilde{\mathbf{w}}^\top\tilde{\mathbf{w}} + C_+\textstyle\sum_i \xi_i + C_-\sum_j \xi_j \\ \text{s.t. } & \tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_i + \tilde{b} + \xi_i \geq +1, \;\; i \in I_+, \\ & \tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_j + \tilde{b} - \xi_j \leq -1, \;\; j \in I_-, \\ & \xi_i,\, \xi_j \geq 0, \end{aligned} \tag{4}$$

[1]This only holds if we agree to ignore the optional compact-convex-region constraints $\|\mathbf{w}\|_\infty < \gamma$ or $|w_i| < \gamma_i, i = 1, ..., d$.

where the $I_\pm$ are indexing sets for the positive and negative training samples, the ξ's are slack variables for the samples' margin constraint violations, and the $C_\pm$ are corresponding penalty weights.

Inserting the EPCF feature vectors into the above training procedure gives our *Extended Polyhedral Conic Classifier* (EPCC) method [3]. The above procedure does not attempt to optimize the position $\mathbf{c}$ of the cone vertex as that would lead to a non-convex problem. It would be possible to optimize for $\mathbf{c}$ at least locally, but here we simply set it to a pre-specified position in the positive training set. The mean, medoid, or coordinate-wise median of the training positives can all be used for this with good results. We used the mean in our experiments. Note that the classifier assigns its highest positive confidence to the samples near the cone vertex.

The above quadratic program based formulation has several advantages over the linear programming approach of [8], which forbids margin violation by negative samples – thus creating a tendency to overfit – and which needs to form a large $(n_+ + n_-) \times (n_+ + d + 2)$ constraint matrix where $n_\pm$ are the numbers of positive and negative training samples – thus making large-scale application difficult.

During training we treat query set as positive class and all gallery images are considered as negative class. For moderate gallery data set sizes, training can be easily accomplished using fast linear SVM solvers, such as Pegasos [22] or LIBOCAS [7]. However, as the gallery set size increases, the training will be slow which makes the method impractical for real-time applications. Therefore, we introduce one-class EPCC that can use positive samples (query set samples) only below. As a result, the training time will be independent of gallery set size which makes the method ideal for real-time application problems with large-scale data.

2.3. One-Class EPCC (OC-EPCC)

EPCC usually outperforms both linear SVM and PCC owing to its flexibility, but its positive acceptance regions are bounded and convex only when $|w_i| < \gamma_i$ for all i – *i.e.* when the hyperplane section has a shallower slope than every facet of the L_1 cone. This sometimes fails to hold for feature space dimensions along which the negatives do not surround the positives on all sides. Even though such EPCC acceptance regions are typically still much smaller than the corresponding linear SVM ones, to ensure tighter bounding we would like to enforce $|w_i| < \gamma_i$, $i = 1, \ldots, d$. Moreover, in EPCC the ∓ 1 margin is the only thing that fixes the overall weight scale and hence prevents a degenerate solution, and negative data is essential for this. (Moving every negative sample outwards to infinity causes $(\mathbf{w}, \boldsymbol{\gamma}, b)$ to progressively shrink to zero, even though the width of the positive class has not changed). To ensure that EPCC can

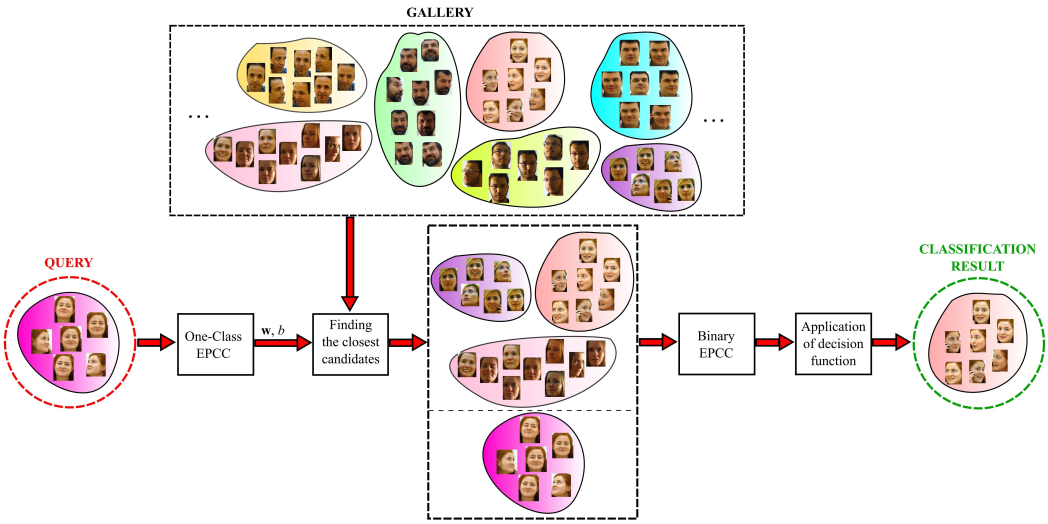

Figure 3. The proposed cascade classification system: One-class EPCC returns $\mathbf{w}, b$ parameters used for finding the closest candidate sets from the gallery. Then, binary EPCC classifier is run on query set and the returned gallery sets. Finally, decision is made based on the closest images to the query set at the last step.

work with only positive samples, we need to force its acceptance regions to stay bounded and compact. The acceptance region has width $O(b/\gamma_i)$ along axis i, so we need to ensure that the γ_i can not shrink to zero. The easiest way to achieve this is to replace the ± 1 margin scaling with a $b = 1$ offset scaling and include negative cost penalties on the γ_i so that these quantities will tend to increase and hence keep the acceptance region widths small and the sets well separated. This leads to the following "One-Class EPCC" formulation:

$$\begin{aligned} \underset{\mathbf{w},\boldsymbol{\gamma}}{\arg\min} \;\; & \tfrac{\lambda}{2}\mathbf{w}^\top\mathbf{w} + \tfrac{1}{n_+}\textstyle\sum_i \xi_i + \tfrac{1}{n_-}\sum_j \xi_j - \mathbf{s}^\top\boldsymbol{\gamma} \\ \text{s.t.} \;\; & \mathbf{w}^\top(\mathbf{x}_i - \mathbf{c}) + \boldsymbol{\gamma}^\top|\mathbf{x}_i - \mathbf{c}| - 1 \le \xi_i, \;\; i \in I_+, \\ & \mathbf{w}^\top(\mathbf{x}_i - \mathbf{c}) + \boldsymbol{\gamma}^\top|\mathbf{x}_i - \mathbf{c}| - 1 \ge 1 - \xi_j, \;\; j \in I_-, \\ & \xi_i,\, \xi_j \ge 0. \qquad\qquad \text{(OC-EPCC)} \end{aligned} \tag{5}$$

Here λ is a regularization weight for $\mathbf{w}$, $\mathbf{s} > 0$ is a user-supplied vector of cost penalties for increasing $\boldsymbol{\gamma}$. At present we use simple stochastic gradient (SG) method to solve this optimization problem, and the algorithm can be seen in the text given as Supplementary material.

2.4. Decision Function and Cascade EPCC Classification

Once we determine $\tilde{\mathbf{w}} \equiv \begin{pmatrix} \mathbf{w} \\ -\boldsymbol{\gamma} \end{pmatrix} \in \mathbb{R}^{2d}$ and $\tilde{b}$ by training one-class or binary EPCC classifiers for a particular query set, we augment each gallery image feature vector as $\tilde{\mathbf{x}} \equiv \begin{pmatrix} \mathbf{x}-\mathbf{c} \\ |\mathbf{x}-\mathbf{c}| \end{pmatrix} \in \mathbb{R}^{2d}$ where $\mathbf{c}$ is set to the mean of query set samples. Then, the following function can be used to approximate similarities for the gallery set samples

$$s(\mathbf{x}_i) \;\; = \;\; \tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_i + \tilde{b}, \;\; i = 1, \ldots, n, \tag{6}$$

where n is the number of all images in the gallery. To assign query set to a class, we choose K samples with the highest scores (these samples are the ones closest to the polyhedral region spanned by the query set), and assign the query set to the most frequent class label among the K samples. In case of ties, we assign the query set to the class yielding the highest mean score. The best value of K can be determined based on cross-validation.

It should be noted that computing distances using (6) is very fast since it requires a simple dot product. So, the speed of the classifier system largely depends on the learning algorithm returning $(\tilde{\mathbf{w}}, \tilde{b})$. Therefore, using one-class EPCC formulation yields to very fast recognition since it does not use gallery set images during training. But, its accuracy is lower since it ignores gallery images. On the other hand, binary EPCC classifier achieves very high accuracies, but it is slower since one needs to train the binary classifier using entire gallery sets. To take the best of both worlds, we use a cascade classifier for large-scale problems where the first stage uses the one-class EPCC formulation and returns the closest candidate gallery sets, and the second stage uses binary EPCC trained with the query set and the closest gallery sets returned by the first stage. This cascade classifier system is visualized in Fig. 3. The proposed cascade is much faster compared to the method using binary EPCC classifier trained with entire gallery sets and the accuracy will be higher than the method using one-class formulation. One does not need to use binary EPCC classifier in the second stage. More sophisticated and time-consuming methods that can better estimate the nonlinear structure of query sets can be used in this stage since we do not need to use all gallery sets.

3. Experiments

We tested the proposed methods[2] on four face recognition datasets: Honda/UCSD [18], YouTube Celebrities, ESOGU-285 Face Videos [28], and COX videos [15]. We tested one-class EPCC using the query set samples only, binary EPCC using the query and all gallery sets, and a cascade classifier system using one-class EPCC in the first stage and the binary EPCC in the second stage. The maximum number of iterations is set to $T = 600$ in SG algorithm used for one-class EPCC classifier. To allow comparison with the literature on various datasets, we followed the simple protocol of [2, 30, 24, 27]: the detected face images were histogram equalized, but no further pre-processing such as alignment or background removal was performed on them. In experiments, we used gray-level values or local binary pattern (LBP) features. For all kernelized methods, we used the Gaussian kernels and the Gaussian kernel width is determined based on experiments using randomly selected subsets of image sets. We compared the proposed method to the linear/kernelized affine hull method (AHISD) [2], linear/kernelized convex hull method (CHISD) [2], Mutual Subspace Method (MSM) [29], SANP [14], Regularized Nearest Points (RNP) [30], Collaboratively Regularized Nearest Points(CRNP) [27], Manifold-Manifold Distance (MMD) [24], Self-Regularized Nonnegative Adaptive Distance Metric Learning (SRN-ADML) [20], and methods of Yalcin et al. [28] using reduced kernelized convex hulls (Kernel RCHISD) and linear/kernelized bounding hypersphere (HS) models for image set classification. As aforementioned, Hayat et al. [11, 13] use a linear SVM classifier for computing distances between query and gallery set images and this may not be appropriate for large scale datasets. To show this, we also compared the proposed classifiers to linear SVMs in our classification setting. To this end, we used the same SVM solver software LIBOCAS [7] for both SVM and EPCC classifiers, and the query set is assigned to a class based on the nearest returned images as described earlier for both methods.

3.1. Experiments on Small Sized Data Sets

3.1.1 Experiments on Honda/UCSD Data Set

The Honda/UCSD data set was collected for video-based face recognition. It consists of 59 video sequences involving 20 individuals. Each sequence contains approximately 300-500 frames. Twenty sequences were set aside for training, leaving the remaining 39 for testing. The detected faces were resized to gray-scale images and histogram equalized, and the resulting pixel values were used as features. We did not extract LBP features for this dataset since pixel values already achieved very high accuracies. We set the number of the closest samples $K = 10$ for both one-class and binary EPCC classifiers. For the cascade EPCC classifiers, we trained the binary EPCC classifier with the closest 10 gallery sets returned by the one-class EPCC classifier.

Table 1 shows the accuracies and testing times for all tested methods. Testing time shows the amount of time spent to classify a single test set on the average. The proposed binary and cascade EPCC classifiers together with kernelized convex hull models, RNP, MMD and CRNP achieve the best accuracy. The linear hypersphere method is the worst performing method, but it is one of the fastest methods. Both one-class and binary EPCC classifiers are quite fast, thus using a cascade system does not bring any advantage in terms of speed here. Compared to linear SVM, EPCC achieves a better accuracy and it is much faster despite it uses longer feature vectors (2 times the original input dimension). Linear SVM is the slowest method since the data are not separable by an hyperlane so the algorithm takes much longer to return a solution.

Table 1. Classification Rates (%) and Testing Times on the Honda/UCSD Dataset.

Method	Accuracy	Testing Time (sec)
One-Class EPCC	94.9	4.2 sec
Binary EPCC	**100**	6.1 sec
Cascade EPCC	**100**	7.1 sec
Linear SVM	97.4	152.0 sec
Linear AHISD	97.4	1.6 sec
Linear CHISD	97.4	5.1 sec
Linear HS	59.0	0.6 sec
MSM	97.4	2.2 sec
SANP	97.4	16.7 sec
RNP	**100**	5.4 sec
CRNP	**100**	2.6 sec
SRN-ADML	97.4	6.2 sec
MMD	**100**	7.1 sec
Kernel AHISD	97.4	14.2 sec
Kernel CHISD	**100**	7.6 sec
Kernel HS	94.9	2.8 sec
Kernel RCHISD	**100**	3.7 sec

3.1.2 Experiments on YouTube Celebrities Data Set

The YouTube Celebrities data set contains 1910 videos of 47 celebrities that are collected from YouTube. Each sequence includes different number of frames that are mostly low resolution. The data set does not provide the cropped faces from videos. Therefore, we manually cropped faces using a semi-automatic annotation tool and resized them to 40×40 gray-scale images and extract LBP features. We conduct 10 runs of experiments by randomly selecting 9

[2]The codes are available at http://mlcv.ogu.edu.tr/softwares.html.

videos (3 for training, 6 for testing) for each experiment by following the same protocol of [31, 23]. We used the same settings for one-class (binary) EPCC and cascade classifiers as in Honda/UCSD experiments.

The averages of the classification rates and testing times are shown in Table 2. The videos in this data set mostly include frontal views of people; therefore, linear methods perform well here.The proposed binary EPCC classifier achieves the best accuracy with a slight edge in front of the CRNP. One-Class EPCC also achieves a good accuracy and it outperforms more complex methods such as MSM, Linear AHISD, SANP, SRN-ADML, etc. The proposed binary EPCC classifier again outperforms linear SVM classifier as before, but linear SVM is slightly faster than binary EPCC on this data set. Using a cascade system slightly improves the testing time over binary EPCC, but the accuracy drops to 68.3% from 72.1%.

Table 2. Classification Rates (%) and Testing Times on the YouTube Celebrities Dataset.

Method	Accuracy	Testing Time (sec)
One-Class EPCC	64.9 ± 2.5	3.5 sec
Binary EPCC	$\mathbf{72.1} \pm 2.4$	5.7 sec
Cascade EPCC	68.3 ± 2.5	4.9 sec
Linear SVM	69.7 ± 2.5	4.7 sec
Linear AHISD	62.0 ± 2.4	39.7 sec
Linear CHISD	65.7 ± 2.5	23.6 sec
Linear HS	50.9 ± 2.6	0.8 sec
MSM	63.8 ± 2.3	2.4 sec
SANP	58.6 ± 3.8	75.5 sec
RNP	68.4 ± 2.6	20.6 sec
CRNP	71.2 ± 2.6	76.6 sec
SRN-ADML	63.6 ± 2.9	76.1 sec
MMD	65.5 ± 2.4	34.0 sec
Kernel AHISD	63.2 ± 2.0	47.1 sec
Kernel CHISD	65.7 ± 2.4	25.4 sec
Kernel HS	53.9 ± 2.1	1.2 sec
Kernel RCHISD	64.9 ± 2.6	4.4 sec

3.2. Experiments on Larger Sized Data Sets

3.2.1 Experiments on ESOGU-285 Face Videos

ESOGU-285 Face Videos includes videos of 285 people captured in two sessions separated by at least three weeks [28]. In each session, four short videos were captured with four different scenarios for each person. The shortest video includes 100 frames and the average frame number is around 300. The total number of face image frames is 764 006; therefore, it is the largest data set in terms of frame number (the total number of face images) among all datasets used in this study. We used LBP features extracted from 120×90 gray-scale images. The number of the closest samples is set to $K = 20$ for one-class EPCC and $K = 50$ for binary EPCC classifier. For the cascade EPCC classifiers, we trained the binary EPCC classifier with the closest 50 gallery sets returned by the one-class EPCC.

Table 3. Classification Rates (%) and Testing Times on the ESOGU Dataset.

Method	Accuracy	Testing Time (sec)
One-Class EPCC	60.2	9.4 sec
Binary EPCC	**86.4**	250.5 sec
Cascade EPCC	84.3	55.6 sec
Linear SVM	81.9	325.6 sec
Linear AHISD	66.8	180.0 sec
Linear CHISD	76.6	390.1 sec
Linear HS	39.5	0.8 sec
MSM	69.6	5.1 sec
SANP	79.1	564.6 sec
RNP	51.9	2205.3 sec
CRNP	OOM	–
SRN-ADML	68.4	380.2 sec
MMD	77.6	150.4 sec
Kernel AHISD	76.1	4369.0 sec
Kernel CHISD	77.6	480.4 sec
Kernel HS	49.4	12.9 sec
Kernel RCHISD	75.4	46.1 sec

The image sets captured in the first session were used as the gallery set whereas the sets captured in the second session were used for testing as in [28]. The recognition accuracies and testing times are given in Table 3. We could not implement CRNP because of memory issues since it requires to operate on matrices with a large size of $n \times n$, and n is the number of frames in the gallery (OOM indicates the "out of memory" problem in the table). The proposed binary EPCC method significantly outperforms all other tested methods and yields an accuracy of 86.4%. It is followed by the proposed cascade classifier and binary SVM. The fourth best method SANP achieves an accuracy of 79.1% which is 7.3% lower than the proposed method. Classification accuracy of one-class EPCC is 60.2% which is low compared to binary EPCC and cascade classifiers. In terms of the speed, the proposed one-class EPCC method is the third fastest method after linear subspaces and hyperspheres. Similarly, both of our binary EPCC and cascade classifiers are faster than the majority of the kernelized methods as well as linear methods including SVM, SANP, CHISD, RNP, and SRN-ADML. In contrast to the experiments on small datasets, using cascade classifier significantly improves the testing time over the binary EPCC.

3.2.2 Experiments on COX Video to Video Data Set

The COX Faces dataset contains 3000 video sequences of 1000 walking individuals [15]. The videos are captured with three fixed camcorders when the subjects walk around the pre-designed S-shape route. The dataset has variations in illumination, pose and resolution through this S-shape route. For this database we used LBP features (LBP features are extracted from 32×40 face images since we do not have access to original video frames) as visual features. There are 3 image sets per person. We choose one set from each person for testing and the remaining two sets were used as gallery. For the second and the third trials, we have chosen the test set from the ones that were not used for testing earlier. We set the number of the closest samples to $K = 10$ for one-class EPCC and $K = 110$ for binary EPCC classifier. For the cascade EPCC classifiers, we trained the binary EPCC classifier with the closest 50 gallery sets returned by the one-class EPCC classifier as before.

Results are given in Table 4. As in ESOGU experiments, the proposed binary EPCC and cascade methods significantly outperform all other tested methods. The closest methods are linear SVM and kernelized CHISD methods and kernelized CHISD only achieves 45.6% accuracy which is 18.4% is lower than accuracy of binary EPCC. Using cascade classifier decreases the accuracy to 62.5%, but it is approximately 6 times faster than using EPCC. Linear SVM also works well for this dataset and it is faster than binary EPCC.

Table 4. Classification Rates (%) and Testing Times on the COX Dataset.

Method	Accuracy	Testing Time (sec)
One-Class EPCC	44.0 ± 8.2	15.7 sec
Binary EPCC	$\mathbf{64.0} \pm 11.5$	171.7 sec
Cascade EPCC	62.5 ± 10.8	27.9 sec
Linear SVM	61.9 ± 12.7	107.4 sec
Linear AHISD	44.3 ± 9.8	82.9 sec
Linear CHISD	44.8 ± 11.3	54.3 sec
Linear HS	25.1 ± 4.9	1.5 sec
MSM	41.6 ± 5.3	18.6 sec
SANP	43.6 ± 11.2	978.7 sec
RNP	45.4 ± 13.7	217.3 sec
CRNP	OOM	–
SRN-ADML	44.6 ± 7.9	351.7 sec
MMD	42.7 ± 10.5	60.3 sec
Kernel AHISD	45.4 ± 10.3	276.4 sec
Kernel CHISD	45.6 ± 10.9	250.2 sec
Kernel HS	42.4 ± 7.6	71.3 sec
Kernel RCHISD	44.3 ± 11.5	65.2 sec

4. Conclusion

In this study, we proposed fast and accurate polyhedral conic classifiers for face recognition to be used with image sets. As opposed to the other linear classifiers returning hyperplane separators, the proposed methods can return polyhedral acceptance regions, which makes them ideal for data sets in which query set is surrounded by gallery images. Computing distances from images of the gallery set to the polyhedral acceptance regions is straightforward in the sense that it requires simple dot product evaluations. The proposed classifier can also be used as one-class classifier which uses the query set samples only. Using one-class formulation of the classifier for face recognition makes the method extremely fast since the training time of the classifier becomes almost the independent of the gallery set size. Therefore, the proposed one-class classifier can be used with other time-consuming methods in a cascade structure (as given in the paper), where one-class EPCC classifier in the first stage returns the most promising candidate gallery sets for more complicated methods.

Experimental results clearly indicate that the proposed EPCC classifiers achieve both fast and accurate recognition and they significantly outperform linear SVMs. More precisely, we obtain the state-of-the-art results on all tested datasets. Our results on larger sized ESOGU-285 and COX datasets are particularly promising in the sense that we achieve accuracies which are at least 7% and 18.6% better than SANP and Kernel CHISD which achieve the best accuracy among other tested rival methods. In addition, the proposed methods are much faster compared to the majority of the tested methods during testing.

It is also worth to point out that EPCC learns the query set region by using both positive and negative classes in contrast to CHISD and one-class EPCC that learn the convex models independently from the gallery sets. EPCC significantly outperforms both methods and this shows a clear indisputable advantage of negative class-sensitive discriminative learning. For larger datasets, even linear SVMs significantly outperform other tested methods that learn each image set model independent of other models in the gallery. So, we definitely need algorithms that learn a class-specific model by taking other classes into consideration. Lastly, we believe that in order to make real significant contributions in the image set based recognition, we should work on large-scale datasets with many classes like the already available ones for other vision tasks such as Imagenet or classical face recognition using single face images. We strongly believe that the advantages of the proposed methods will be more notable in large-scale settings.

References

[1] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005. 1

[2] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010. 1, 2, 6

[3] H. Cevikalp and B. Triggs. Polyhedral conic classifiers for visual object detection and classification. In *CVPR*, 2017. 3, 4

[4] S. Chen, C. Sanderson, M. T. Harandi, and B. C. Lovell. Improved image set classification via joint sparse approximated nearest subspaces. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013. 2

[5] Y.-C. Chen, V. M. Patel, S. Shekhar, R. Chellappa, and P. J. Phillips. Video-based face recognition via joint sparse representation. In *IEEE Int. Conf. on Automat. Face Gesture Recognit.*, 2013. 2

[6] Z. Cui, H. Chang, S. Shan, B. Ma, and X. Chen. Joint sparse representation for video-based face recognition. *Neurocomputing*, 135:306–312, 2014. 2

[7] V. Franc and S. Sonnenburg. Optimized cutting plane algorithm for large-scale risk minimization. *Journal of Machine Learning Research*, 10:2157–2232, 2009. 4, 6

[8] R. N. Gasimov and G. Ozturk. Separation via polyhedral conic functions. *Optimization Methods and Software*, 21:527–540, 2006. 3, 4

[9] J. Hamm and D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Int. Conf. Mach. Learn.*, 2008. 1

[10] M. Hayat, M. Bennamoun, and S. An. Learning non-linear reconstruction models for image set classification. In *EEE Conf. Comput. Vis. Pattern Recognit.*, 2014. 2

[11] M. Hayat, M. Bennamoun, and S. An. Reverse training: an efficicient approach for image set classification. In *ECCV*, 2014. 2, 3, 6

[12] M. Hayat, M. Bennamoun, and S. An. Deep reconstruction models for image set classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37:713–727, 2015. 2

[13] M. Hayat, S. Khan, and M. Bennamoun. Empowering simple binary classifiers for image set based face recognition. *International Journal of Computer Vision*, pages 1–20, 2017. 2, 3, 6

[14] Y. Hu, A. S. Mian, , and R. Owens. Face recognition using sparse approximated nearest points between image sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):1992–2004, 2012. 2, 6

[15] Z. Huang, S. Shan, R. Wang, H. Zhang, S. Lao, A. Kuerban, and X. Chen. A benchmark and comparative study of video-based face recognition on cox face database. *IEEE Transactions on Image Processing*, 24:5967–5981, 2015. 6, 8

[16] Z. Huang, R. Wang, X. Li, W. Liu, S. Shan, L. V. Gool, and X. Chen. Geometry-aware similarity learning on SPD manifolds for visual recognition. *CoRR*, abs/1608.04914, 2016. 1

[17] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *ICML*, 2015. 1

[18] K. Lee, J. Ho, M. Yang, and D. Kriegman. Visual tracking and recognition using probabilistic appearance manifolds. *Computer Vision and Image Understanding*, 99(3):303–331, 2005. 6

[19] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou. Multi-manifold deep metric learning for image set classification. In *CVPR*, 2015. 2

[20] A. Mian, Y. Hu, R. Hartley, and R. Owens. Image set based face recognition using self-regularized non-negative coding and adaptive distance metric learning. *IEEE Transactions on Image Processing*, 22:5252–5262, 2013. 6

[21] G. Shakhnarovich, J. W. Fisher, and T. Darrell. Face recognition from long-term observations. In *European Conf. On Comp. Vis.*, pages 851–868, 2002. 1

[22] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-grdient solver for SVM. In *International Conference on Machine Learning*, 2007. 4

[23] R. Wang and X. Chen. Manifold discriminant analysis. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009. 7

[24] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image sets. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008. 6

[25] T. Wang and P. Shi. Kernel grassmannian distances and discriminant analysis for face recognition from image sets. *Pattern Recognition Letters*, 30:1161–1165, 2009. 1

[26] X. Wang, W. Bian, and D. Tao. Grassmannian regularized structured multi-view embedding for image classification. *IEEE Transactions on Image Processing*, pages 2646–2660, 2013. 1

[27] Y. Wu, M. Minoh, and M. Mukunoki. Collaboratively regularized nearest points for set based recognition. In *British Machine Vision Conference*, 2013. 2, 3, 6

[28] M. Yalcin, H. Cevikalp, and H. S. Yavuz. Towards large-scale face recognition based on videos. In *International Conference on Computer Vision Workshop on Video Summarization for Large-Scale Analytics*, 2015. 2, 3, 6, 7

[29] O. Yamaguchi, K. Fukui, and K.-I. Maeda. Face recognition using temporal image sequence. In *International Symposium of Robotics Research*, pages 318–323, 1998. 1, 2, 6

[30] M. Yang, P. Zhu, L. V. Gool, and L. Zhang. Face recognition based on regularized nearest points between image sets. In *IEEE Int. Conf. on Automat. Face Gesture Recognit.*, 2013. 2, 6

[31] P. Zhu, W. Zuo, L. Zhang, S. C.-K. Shiu, and D. Zhang. Image set-based collaborative representation for face recognition. *IEEE Transactions on Information Forensics and Security*, 9:1120–1132, 2014. 2, 3, 7