ABSTRACT

Title of dissertation: SPARSE REPRESENTATIONS AND FEATURE LEARNING FOR IMAGE SET CLASSIFICATION AND CORRESPONDENCE ESTIMATION

Mohammed E. Fathy Salem
Doctor of Philosophy, 2018

Dissertation directed by: Professor Rama Chellappa
Department of Computer Science

The use of effective features is a key component in solving many computer vision tasks including, but not limited to, image (set) classification and correspondence estimation. Many research directions have focused on finding good features for the task under consideration, traditionally by hand crafting and recently by machine learning. In our work, we present algorithms for feature extraction and sparse representation for the classification of image sets. In addition, we present an approach for deep metric learning for correspondence estimation.

We start by benchmarking various image set classification methods on a mobile video dataset that we have collected and made public. The videos were acquired under three different ambient conditions to capture the type of variations caused by the 'mobility' of the devices. An inspection of these videos reveals a combination of favorable and challenging properties unique to smartphone face videos. Besides mobility, the dataset has other challenges including partial faces, occasional pose changes, blur and fiducial point localization errors. Based on the evaluation, the recognition rates drop dramatically when

enrollment and test videos come from different sessions.

We then present Bayesian Representation-based Classification (BRC), an approach based on sparse Bayesian regression and subspace clustering for image set classification. A Bayesian statistical framework is used to compare BRC with similar existing approaches such as Collaborative Representation-based Classification (CRC) and Sparse Representation-based Classification (SRC), where it is shown that BRC employs precision hyperpriors that are more non-informative than those of CRC/SRC. Furthermore, we present a robust probe image set handling strategy that balances the trade-off between efficiency and accuracy. Experiments on three datasets illustrate the effectiveness of our algorithm compared to state-of-the-art set-based methods.

We then propose to represent image sets as a dictionaries of hand-crafted descriptors based on Symmetric Positive Definite (SPD) matrices that are more robust to local deformations and fiducial point location errors. We then learn a tangent map for transforming the SPD matrix logarithms into a lower-dimensional Log-Euclidean space such that the transformed gallery atoms adhere to a more discriminative subspace structure. A query image set is then classified by first mapping its SPD descriptors into the computed Log-Euclidean tangent space and then using the sparse representation over the tangent space to decide a label for the image set. Experiments on four public datasets show that representation-based classification based on the proposed features outperforms many state-of-the-art methods.

We then present Nonlinear Subspace Feature Enhancement (NSFE), an approach for nonlinearly embedding image sets into a space where they adhere to a more discriminative subspace structure. We describe how the structured loss function of NSFE can be optimized in a batch-by-batch fashion by a two-step alternating algorithm. The algorithm makes very

few assumptions about the form of the embedding to be learned and is compatible with stochastic gradient descent and back-propagation. We evaluate NSFE with different types of input features and nonlinear embeddings and show that NSFE compares favorably to state-of-the-art image set classification methods.

Finally, we propose a hierarchical approach for deep metric learning and descriptor matching for the task of point correspondence estimation. Our idea is motivated by the observation that existing metric learning approaches based on supervising and matching with only the deepest layer result in features that are suboptimal in some aspects to shallower features. Instead, the best matching performance, as we empirically show, is obtained by combining the high invariance of deeper features with the geometric sensitivity and higher precision of shallower features. We compare our method to state-of-the-art networks as well as fusion baselines inspired from existing semantic segmentation networks and empirically show that our method is more accurate and better suited to correspondence estimation.

SPARSE REPRESENTATIONS AND FEATURE LEARNING FOR IMAGE SET CLASSIFICATION AND CORRESPONDENCE ESTIMATION

by

Mohammed E. Fathy Salem

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Wojciech Czaja, Dean's Representative
Professor Furong Huang
Professor David Jacobs
Professor Min Wu

Dedication

To my mother and the loved memory of my father

Acknowledgments

I begin by thanking Allah, Most Gracious, Most Merciful who inspired me and blessed me. I also owe gratitude to those people who helped me in my journey through PhD.

Undeniable are the tremendous support and advice of Prof. Rama Challeppa. I benefited a lot from his intellect, wisdom, and exemplary hard work. Many times I would send him drafts, sometimes very close to the deadline, and he would return them with brilliant edits and feedback, in almost no time. And this holds for all students in our group despite the relatively big size of it. With that being said, I also remember how supportive Prof. Chellappa was when some nerve problems affected my hands. He had no hesitation to approve the installation of ergonomic equipment on my desk, which helped my hands recover. Of course this is just one of the many examples of how Prof. Chellappa has been very nice to me and the rest of the lab.

I wish also to thank Azadeh Alavi for the dedication, time and, effort she offered during our fruitful research collaboration. In addition, I would like to thank Quoc-Huy Tran (NEC Labs), Zeeshan Zia (MS HoloLens), and Manmohana Chandraker (NEC Labs, UCSD) for their mentorship during my NEC Labs internship and their advice afterwards.

Special thanks go to the great UMIACS computing staff for their support with hardware and software issues. I benefited a lot from their UNIX expertise.

I am very grateful to my wife for sharing with me the journey through PhD and life. I will always remember how patient and very supportive she was. I am also grateful to both of my daughters for their love.

Last but not least, I would like to thank my great parents who have encouraged me and devoted themselves to support me in my whole life. They selflessly spared no time, effort, money, or prayers to help me until I traveled to start my PhD. Whatever I do, I will never manage to return the favor.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Image Set Classification

In many practical applications such as surveillance-based face recognition and smartphone video-based face authentication, the test example contains a set of images that share the same, yet to be determined label. The interest in the use of image sets for visual recognition tasks, such as face recognition, has grown in line with the increasing prevalence of video-capable consumer devices and surveillance cameras [17, 20, 23, 24, 44, 46, 50, 51, 53, 55, 79, 80, 84, 85, 93, 120, 121, 122, 126]. A video is typically believed to have richer information than in a still image and so should lead to better classification performance. The improvement in performance is limited in practice because videos share many of the challenges present in still images (e.g. variations in pose, illumination and occlusion) in addition to video-specific challenges such as the low resolution at which videos are sometimes captured to reduce bandwidth and storage requirements. Additional challenges may exist in particular instances of image set classification problems. One such challenge

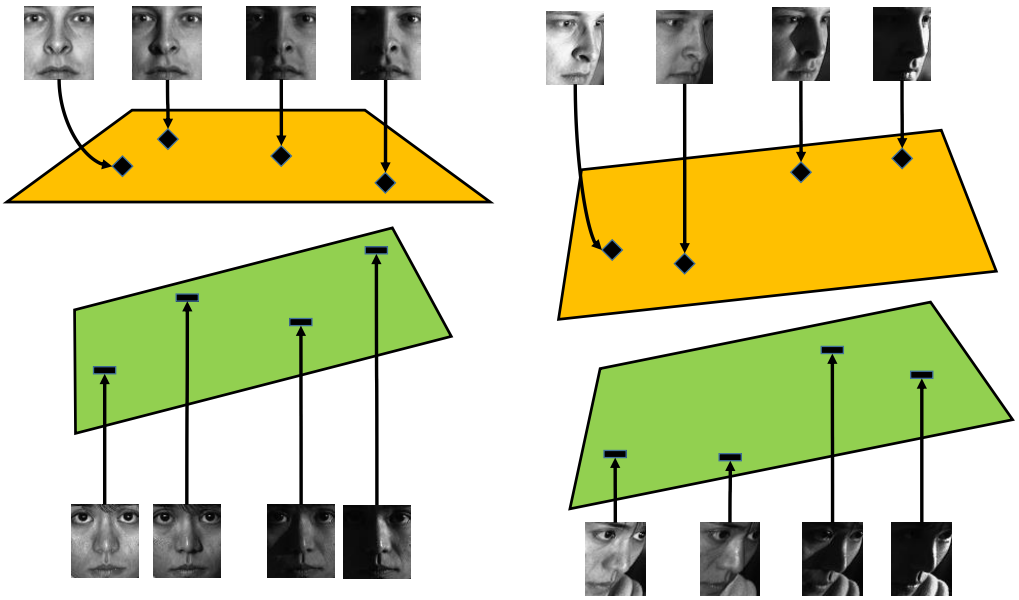


Figure 1.1: An illustration of the discriminative subspace structure that is naturally exhibited by the *controlled* images of a visual object (e.g. a person's face) [12, 133]. The example illustrates the property for face images of two different subjects, taken under two different poses and varying illumination. Images in which the visual object (i.e. face) has the same pose and identity lie close to a low-dimensional subspace regardless of the variations in Lambertian illumination.

is the presence of outlier samples either in the query set, gallery sets, or both. This becomes a problem when some of the detections produced from a video by object detection and tracking algorithms are wrong or improperly localized. Since image sets are constructed by such error-prone automatic algorithms, the classification algorithm should be designed to be robust to such outliers.

Among the many algorithms that have been successfully used for image set classification, Sparse Representation-based Classification (SRC) over dictionaries has been shown to be very effective [24, 93]. The standard SRC algorithm has become popular in visual identification tasks since the seminal work of Wright et al. [133]. The success of this method, as well as many other image set classification methods, is justified by the discriminative low-dimensional subspace structure that is naturally exposed in the

space of visual images of an object. More specifically, it has been mathematically proved that images of a fixed object taken under varying Lambertian illumination from a fixed viewpoint lie on a low-dimensional subspace [12]. As illustrated in Fig. 1.1, this suggests that the instances from a particular class lie on (or close to) a low-dimensional linear subspace (assuming static object with no change in pose across images) or a small number of such subspaces (to account for variations in pose and deformations).

This unique geometric layout of image vectors, which we refer to as the *subspace property*, has inspired many algorithms for visual recognition and image set classification. On the other hand, the subspace property was proved under a certain set of assumptions that may not always hold in practice. One such assumption is that images are represented by their raw intensities and are perfectly aligned. Raw intensities as features are very sensitive to noise and illumination changes. In addition, small deformations can lead to dramatic changes in the layout of intensity representation. Other feature representations that are less sensitive to such factors can be obtained by non-linear transformations of raw intensities. Inevitably, the nonlinearity involved in these transformations breaks the sparse linear dependence between the same-class same-pose image vectors, leading to the loss of the discriminative subspace structure that SRC and many other algorithms utilize to infer image classes.

In order to make use of the subspace property for classification, many classification algorithms based on the subspace property tend to make one additional assumption that may not also hold in practice. In particular, such algorithms require that the pose of a query object image be present in the object's image gallery/training set so that it can be related successfully with its true class. For example, the authors of SRC explicitly

assume that all gallery and probe images are frontal face images [133]. If the pose of the probe mismatches the poses of the gallery images of the true class, the subspace-based classification algorithm may end up preferring the pose over identity.

1.2 Feature Learning for Correspondence Estimation

The advent of repeatable high curvature point detectors [49, 75] heralded a revolution in computer vision that shifted the emphasis of the field from holistic models of objects and direct matching of image patches [147] to highly discriminative hand-crafted descriptors. This descriptor revolution that started in the late 1990s, had an impact on virtually every task in computer vision, and pipelines were designed around feature descriptors to solve tasks from optical flow to object detection, and from 3D reconstruction to action recognition.

The current decade is witnessing a wide-ranging revolution in our field, brought about by the reemergence of deep neural networks. Yet there exist computer vision pipelines that, thanks to extensive engineering efforts of the past decades, have proven impervious to end-to-end learned solutions. The best deep learning solutions have not succeeded in convincingly outperforming state-of-the-art methods on problems such as structure-from-motion (SfM) [124] and object instance detection [102]. We see a consensus emerging that some of the pipelines employing interest point detectors and descriptors are here to stay, but it might rather be advantageous to leverage deep learning for individual components of those pipelines.

Recently, a few convolutional neural network (CNN) architectures [26, 125, 137, 144] have been proposed with the aim of learning strong geometric feature descriptors for

matching images. While such 'deep descriptors' have been shown to be highly robust, we highlight in our work some of the limitations that result from the excessive invariance of such features and explore different ways to enrich the high invariance of deeper features with high sensitivity to fine image structure.

1.3 Summary of Contributions

In Chapter 3, we evaluate algorithms for image set classification and video-based face recognition under new practical conditions that have received less attention in the past. In particular, we consider the application of video-based face recognition to actively authenticating smartphone users based on videos of the user's face that are captured by the smartphone's front-facing camera during normal user interaction with the phone. For this aspect, we experiment with a dataset of 750 videos covering 50 users while doing various tasks, then we split these videos into small sub-videos that are used for user identification. We inspect these video for these challenges that are specific to the domain of mobile devices and we benchmark various state-of-the-art algorithms on different scenarios.

In Chapter 4, we analyze, from a Bayesian statistical perspective the two most commonly used approaches for representation-based classification, namely Collaborative Representation-based Classification (CRC) [145] and SRC. We show that the two approaches are identical up to a different implicit choice of precision hyperpiors. Based on that analysis, we also describe Bayesian Representation-based Classification (BRC), which is obtained by choosing a precision hyperprior that is more non-informative than those of CRC and SRC. Then, we describe extensions of BRC so as to handle image sets.

Extensive comparisons on different image set classification datasets show the superiority of BRC compared to both SRC and CRC.

State-of-the-art algorithms for describing an image set use descriptors that are either very high-dimensional and/or sensitive to outliers and image misalignment. Accordingly, we propose in Chapter 5 to represent image sets as dictionaries of Symmetric Positive Definite (SPD) matrices that are more robust to local deformations and outliers. In addition, we propose Subspace Feature Learning (SFL), which learns a tangent map for transforming the SPD matrix logarithms into a lower-dimensional Log-Euclidean space such that the transformed gallery atoms adhere to a more discriminative subspace structure. A query image set is then classified by first mapping its SPD descriptors into the computed Log-Euclidean tangent space and using the sparse representation over the tangent space to decide a label for the image set. We also consider the case of imbalanced as well as large gallery sets and show how dictionary learning can be integrated into SFL to increase its robustness and classification-time efficiency. Experiments on four challenging datasets show that the proposed method outperforms many state-of-the-art methods. We also show that SFL performs well on various types of deep feature inputs.

While several methods have been proposed for modeling and recognizing image sets, the success of these methods relies heavily on how well the image data follows the assumptions of the underlying models. Among the models that have been utilized by many image set classification methods, the physically inspired subspace model assumes that the images of an object lie on a union of low-dimensional subspaces. Despite their successful performance in controlled environments, the performance of such subspace-based classifiers suffers in practical unconstrained settings, where the data may not strictly

follow the assumptions necessary for the subspace model to hold. Accordingly, we propose in Chapter 6 Nonlinear Subspace Feature Enhancement (NSFE), an approach for nonlinearly embedding image sets into a space where they adhere to a more discriminative subspace structure. In turn, this improves the performance of subspace-based classifiers such as sparse representation-based classification. We describe how the structured loss function of NSFE can be optimized in a batch-by-batch fashion by a two-step alternating algorithm. The algorithm makes very few assumptions about the form of the embedding to be learned and is compatible with stochastic gradient descent and back-propagation. This makes NSFE usable with deep, feed-forward embeddings and trainable in an end-to-end fashion. We experiment with two different types of features and nonlinear embeddings over three image set datasets and show that our method compares favorably to state-of-the-art image set classification methods.

We then shift focus to local feature learning for correspondence estimation in Chapter 7 where we present a hierarchical approach for deep metric learning and descriptor matching. During training, our approach simultaneously supervises shallower as well as deeper layers of a Convolutional Neural Network (CNN) using Correspondence Contrastive Loss (CCL) coupled with active hard-negative mining. During matching, our approach uses the more geometrically sensitive shallower features to refine the rough matches established by the highly invariant deeper features. Since dense fusion of features from different layers has already been utilized for semantic segmentation, we compare the proposed bi-level hierarchical approach to local feature learning baselines based on hypercolumn fusion [48] and top-down refinement [99] architectures for semantic segmentation, in addition to state-of-the-art architectures for local feature learning.

Our idea is motivated by the observation that existing metric learning approaches that base supervision and matching on only the deepest layer result in features that are suboptimal in some aspects to shallower features. Instead, the best matching performance, as we empirically show, is obtained by combining the high invariance of deeper features with the geometric sensitivity and higher precision of shallower features. We compare our method to state-of-the-art networks as well as fusion baselines inspired from existing semantic segmentation networks and we empirically show that our method is more accurate and better suited to correspondence estimation.

As future work, we propose to extend our shallow feature learning method into a deep feature learning one with the same goal of enhancing the discriminative subspace arrangement of the visual data. In addition, we propose to use CNNs to learn local feature descriptors for the purpose of matching points lying on ground planes. This can allow more robust estimation of ground plane parameters while solving the structure from motion problems involved in monocular visual odometry.

Chapter 2

Image Set Classification Related Work

2.1 Overview

The image set classification problem has been formulated in various ways. One popular formulation is to compute the distance, either over a vector space or a manifold, between the probe set and each gallery set and then associate the probe with the class of its nearest gallery set. These include discriminative [43, 46, 54, 55, 120, 122, 126] and non-discriminative methods [17, 23, 24, 53, 121]. Other formulations that do not rely on nearest neighbor-based classification include the binary SVM reverse-training approach of [51], neural network-based methods [50, 80], linear representation/coding methods [93, 149] and clustering methods [85].

In this chapter, we review the relevant literature on image set classification and video-based face recognition. We give more emphasis on linear representation and manifold-based algorithms due to their relevance to the research we have conducted.

2.2 Vector Space Methods

Several methods treat the whole image set as a subspace and measure the distance between subspaces by finding the pair of closest points inside them. Such methods include Affine (or Convex) Hull Image Set Distance (AHISD/CHISD) [17] and Sparse-Approximated Nearest Points (SANP) [53]. The Sparse-Approximated Nearest Subspaces (SANS) [23] applies sparse coding to subspace-cluster each gallery image set and measures the distance from the gallery set to the probe set by finding the average distance of each cluster in the gallery set to its nearest subspace approximation from the probe set. Dictionary-based Face Recognition from Videos (DFRV) [24] learns a dictionary consisting of K subdictionaries for each gallery image set after clustering its images by appearance into K groups. The probe set is associated with the class whose gallery dictionaries result in the lowest reconstruction error for the majority of the images in the probe set. Simultaneous Feature and Dictionary Learning (SFDL) [79] discriminatively learns dictionaries for the different classes in addition to learning a linear projection $\mathbf{W}$ to improve the separation between the instances of the different classes. The classification algorithm is identical to DFRV except that the probe images are first transformed using $\mathbf{W}$. Hierarchical subspace clustering of the combined set of faces of the gallery and the probe has been proposed using Grassmann manifolds [85]. The probe set is associated with the class for which the distribution of its images over the clusters is most similar to the distribution of the images of the probe set. In effect, this approach can be too expensive as it needs to recompute the sparse representation of all instances in the gallery and run clustering every time a probe set is to be classified.

2.3 Manifold Methods

Another approach is to represent the image sets as manifolds (or points on a manifold) and use the distance $d(\mathcal{P}, \mathcal{G})$ between the probe $\mathcal{P}$ and each gallery set $\mathcal{G}$ to label $\mathcal{P}$. Methods based on this general idea differ on how they represent an image set as/on a manifold and the way the distance between the sets is measured. Examples of methods that represent each image set as a separate manifold include the Manifold-Manifold Distance (MMD) method Wang et al. [121] and the Manifold Discriminant Analysis (MDA) method [120]. Other manifold methods have represented the subspace approximately spanning an image set as a point on a Grassmann manifold (as opposed to representing each set as a separate manifold). Kernels for Grassmann manifolds are then utilized to perform Discriminant Analysis (DA) [43] or graph-based DA [46] and distances in the embedded space are used for classification. Kernel dictionary learning and sparse coding on Grassmann manifold have also been considered for image set classification [44]. Instead of using kernels, Projection Metric Learning (PML) [54] discriminatively learns a mapping into another, lower-dimensional Grassmann manifold where the projection distance between a pair of points is used for nearest neighbor classification. Covariance Discriminant Learning (CDL) [122] treats the covariance of the image set as a point on a Riemannian manifold that is mapped to a Euclidean space via the logarithmic map. Partial Least Squares (PLS) is then used to learn the mapping from the gallery points to their labels and the obtained mapping is used to classify the probe point. Another related method learns a discriminative, geometry-preserving Mahalanobis metric over the logarithm of the mean-modified covariance matrices and is shown to outperform CDL in [55]. Discriminant

Analysis on the Riemannian manifold of Gaussian distributions (DARG) models each image set as a Mixture of Gaussians (MoG) and then runs kernel discriminant analysis based on a combined kernel for Gaussians [126]. Harandi et al. [45] suggested to model each image set as a probability density function using kernel density estimation on the statistical manifold.

2.3.1 Log-Euclidean Feature Learning

Various approaches for learning features, metrics, and/or dimensionality reduction embeddings have been proposed within the Log-Euclidean (LE) framework [55, 71, 118, 122, 126, 134, 136]. The goal of these approaches is to boost the performance of nearest neighbor classification whereas the goal of our proposed work is to boost the performance of subspace-based classification. Qiu and Sapiro [101] proposed an approach for learning linear transformations that improve the performance of subspace-based classification in Euclidean space. This approach uses sub-gradient descent to minimize a non-convex cost function which can take too many iterations to converge and may fall into local minima. In addition, the approach requires performing Singular Value Decomposition (SVD) at each iteration, which makes it even more expensive. The proposed LE feature learning approach is significantly more robust as it is not subject to local minima. In addition, our approach is faster and more scalable as the optimal solution is obtained by solving a single generalized eigenvalue problem.

2.4 Linear Representation (Coding) Methods

An effective approach, proposed in [133], for utilizing the subspace assumption for recognizing a given face feature vector is to first compute its linear representation with respect to the gallery samples (i.e. projecting it on the gallery) then associate it with the class contributing the most to the representation. SRC [133], proposed for recognition of still face images, adopts this idea and casts the recognition problem as that of solving a convex Lasso optimization problem for the representation of the probe instance with respect to the gallery. While SRC was shown to be quite successful, the empirical results obtained in [145] show that replacing the Lasso's l_1-regularization term with an l_2-regularization term may *sometimes* perform as well as (or even better than) l_1-regularization in terms of classification performance, while leading to a computationally more efficient solution.

Methods utilizing SRC and CRC for image set classification have been developed such as the Mean Sequence SRC (MS-SRC) [93] and Image Set CRC (ISCRC) [150]. While SRC assumes Euclidean space, Harandi et al. [44] extended the sparse coding approach to Grassmann manifold where it has been applied to face recognition from image sets. Sparse coding over SPD manifolds was also considered but for non-image set classification tasks as in [42, 47, 140]. More specifically, SRC over the LE tangent space of SPD manifolds for the task of action recognition was considered in [42, 140]. Harandi et al. [47] and Li et al. [72] proposed kernel approaches for sparse coding over the SPD manifold. Kernel-based methods, however, run the risk of fast growth in running time and memory requirements with the increase of the number of gallery samples, since building the kernel matrix requires $\Theta(n^2)$ time and memory for n samples.

2.5 Neural Network Methods

With recent successes of deep networks in many vision tasks, different neural network architectures have been recently utilized for image set classification. Two such examples are the generative, per-class five-layer model proposed in [50] and the discriminative, per-class two-layer model proposed in [80]. In Chapter 6, we describe an algorithm for training the parameters of a nonlinear embedding for the purpose of subspace feature enhancement. Both of the nonlinear concrete embeddings we use therein are based on neural networks.

Chapter 3

Performance of Video-Based Face Recognition on Mobile Devices

3.1 Overview

Developments in sensing and communication technologies have led to an explosion in the use of mobile devices such as smartphones and tablets. Mobile devices make the management of personal information such as emails, bank accounts and profiles convenient and flexible. However, with the increasing use of mobile devices one has to constantly worry about the security and privacy as the loss of a mobile device would compromise personal information of the user.

Most mobile devices use passwords, pin numbers, or secret patterns for authenticating users. As long as the device remains active, there is no mechanism to verify that the user originally authenticated is still the user in control of the device. As a result, unauthorized individuals may improperly gain access to personal information of the user if

Figure 3.1: Sample video frames for 20 (out of 50) users. The head of the user is close always close to the camera. The bottom row shows some of the challenges present in the data including illumination, pose, expression, partial faces and blur.

the password is compromised. Active Authentication (AA) systems deal with this issue by continuously monitoring the user identity after the initial access has been granted. However, AA remains an unsolved problem specially for smartphones. Various efforts for authenticating smartphones have been proposed. Examples include systems based on screen touch gestures [37, 38], gait recognition [29], and device movement patterns (as measured by the accelerometer) [100]. As smartphones come equipped with a user-facing camera and multiple core processors/GPUs, it is becoming more feasible to utilize the existing body of research in face recognition for face-based AA on smartphones.

Over the years, many algorithms have been proposed for face recognition from still-images, image sets and videos. Examples include Eigenfaces [117], Fisherfaces [14, 34], SRC [133], AHISD/CHISD [17], SANP [53], DFRV [24], and MSSRC [93] just to name a few. While such algorithms have been tested on challenging benchmarks [66, 67, 94, 98] it is hard to predict if they will achieve the same performance on smartphone face videos as they may involve challenges different from those in surveillance-based face recognition datasets. Thus, it becomes necessary to (a) build a dataset that captures the challenges

of smartphone face videos and (b) provide a benchmark to quantify how well existing algorithms can solve the problem in addition to helping future research efforts. MOBIO is the only other benchmark that is based on smartphone face videos [88]. Unlike our study, the benchmark of MOBIO considered only still-image-based methods and only one frame per video is manually cropped, normalized and included in the evaluation [41]. So challenges such as partial faces and incorrect facial/fiducial point detections are not addressed in that work.

In this chapter, we present a benchmark for measuring (and comparing) the effectiveness of face recognition techniques when used for active authentication using face videos captured by the smartphone's front-facing camera. The benchmark dataset consists of 750 videos from 50 different users and two evaluation protocols that reflect some of the challenges a typical face-based active authentication system is likely to deal with in practical smartphone applications. We used the two protocols to evaluate several existing techniques for still-image-based and image set-based face recognition including state-of-the-art ones. Although some techniques perform better than others, the best performance obtained is still not adequate even when the features are extracted around face fiducial points. To encourage further research, we have made the dataset publicly available.

3.2 Mobile Face Dataset Description

The dataset was collected using a custom-written app on an iPhone 5s. The app collected data for five different tasks (See Figure 3.2). During each task, the app recorded each users' face video from the front camera as well as the touch data sensed by the screen. Each

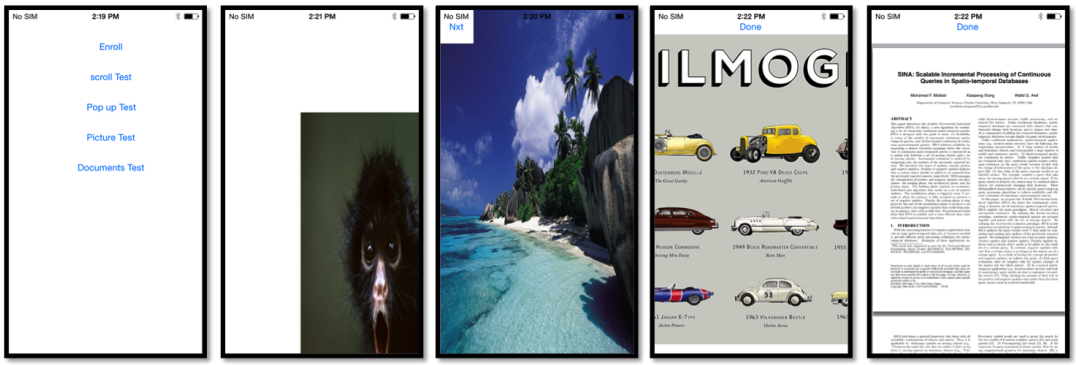


Figure 3.2: Screen shots of the application and tasks used to collect data on an iPhone 5s.

user performed five tasks in three settings (sessions) with very different environmental conditions. These setting were as follows: (a) in a well-lit room, (b) in the same room but with dim lighting, and (c) in a different room with natural daytime illumination. Although the three sessions of a given user were collected in the same day, the benchmark results indicate that the dataset is still challenging as state-of-the-art methods fail to achieve good performance in cross-session evaluations. The different tasks are described below.

- **Enrollment Task**: The user would enroll his/her face by turning his/her head to the left, then to the right, then up, and finally down while being recorded by the front-facing camera on the iPhone. Following the enrollment task, the user would perform four tasks with both face and screen touch data being recorded simultaneously. The four tasks are described as follows.

- **Document Task**: The user is presented with a 12-page long PDF research paper and is asked to count the number of items indicated by the test proctor such as figures, tables etc.

- **Picture Task**: A large poster-like image displayed 72 cars with different colors in a 12 by 6 table. The user was asked to count the number of cars of a particular color selected by the test proctor. Only a few cars could be seen at any given time on the screen and so scrolling was necessary to view all cars.

- **Popup Task**: Fifteen images were positioned off screen in such a way that only a little bit of the image was shown. The user was required to drag the image and position it at the center of the iPhone to the best of their ability.

- **Scrolling Task**: The app displayed a collection of images that were arranged horizontally and vertically. Each image would take up the whole screen and the user was required to swipe (using their finger) on the screen left and right or up and down in order to navigate through the images.

The new dataset consists of 750 video sequences from 50 different users. Before starting each task, the task description was verbally conveyed to the user. No further instructions were given to the users regarding their pose or the way they held and interacted with the device while doing the different tasks. The resolution of each video is 1280×720. The average video duration is 11 seconds for the Enrollment Task, 43 seconds for the Document Task, 40 seconds for the Picture Task, 51 seconds for the Popup Task, and 32 seconds for the Scrolling Task. Figure 3.1 shows some sample recorded images from this dataset.

An inspection of videos in this dataset reveals a combination of characteristics that is unique to front camera videos. Some of these are favorable characteristics that can be utilized to increase the robustness and efficiency of the authentication process. For

example, most users keep their heads close to the smartphone while using it. Most of the time, users keep their faces and eyes directed towards the phone (i.e. the camera) while they interact or read something off the phone although they may turn their heads occasionally, for example, to speak to someone or look around.

Other characteristics present in these videos are challenging for many state-of-the-art face authentication systems. The fact that the device (and so the camera) is held by the user during data acquisition phase contributes to many observed variations in face images. For example, the imaging device is subject to shakes and sudden movements which result in blurred frames in some of the videos (even normal head movements contribute to the blurring of faces). Users can also adjust the height and distance of the device relative to their heads in the middle of any interaction, which can change the background and the location, size and distortion of the face within the images. We also noticed that some users hold the device during some interactions such that only a part of the face remains fully within the field of view of the camera.

In addition to the aforementioned challenges, a major challenge with smartphone face videos is the variations in illumination and contextual conditions within the videos of the same subject resulting from the mobility of the device. This issue is practically inevitable as smartphones are designed to be carried and used everywhere and all the time. To capture this mobility challenge in our dataset, the data for each user has been collected under the three aforementioned sessions, each of which has different illumination condition.

Figure 3.3: Increasing the size of the smallest search window of VJ detector to 25% of the frame size eliminates all the false alarms within the 149 detections (shown in the left) made in a sample video file while keeping the 8 true positives (shown in the right). The figure is best viewed electronically.

3.3 Preprocessing

Face Detection The first step is to locate the user's face from each frame. While there are several algorithms for face detection [87], we used the Viola-Jones (VJ) detector [119] as it is relatively fast and has tuned open-source implementations available on popular smartphone platforms. We utilized the fact that the user's face is close to the camera during acquisition time by setting the size of the smallest search window to 25% of the frame resolution. This makes the detector run 46 times faster (28 fps on MATLAB using a single-core 2.2 GHz processor) while reducing the false positives drastically which usually have small dimensions (see Figure 3.3 for an example).

It is worth noting that some frames contribute no detections. In many cases, this is because of partial faces or the user looking away from the phone.

Figure 3.4: Top row: cropped facial detections (before histogram normalization). Bottow row: the fiducial points computed by the pre-trained model of [7]. The left three pairs are examples of good results while the right three pairs are examples of incorrectly placed fiducial points.

Fiducial Point Detection Given the face bounding box, we use the pre-trained landmark detector of [7] available from [1] to identify fiducial points at the eyes, nose and mouth. We use these to guide the feature extraction step in an effort to normalize appearance variation due to pose and expression. For robustness, we drop any detection if we find that any of the fiducial points on the eyes, nose or mouth is outside or too close to the boundary of the face detection rectangle. A fiducial point is considered too close if it lies less than 5 pixels away from any of the four sides of the detection rectangle. Since all preprocessing is fully automatic, the resulting detections may not always be perfect. Figure 3.4 shows examples of good and bad results obtained. We do not attempt to filter out these bad results manually and we rely on the robustness of the subsequent image set classifier to deal with such outliers.

The detected faces are then cropped out and rescaled to 256×256. We then apply histogram equalization to reduce the variations due to illumination. The resulting face images are then used for feature extraction.

Feature Extraction Given a detected face image $\mathbf{I}$, we extract a 400D feature vector $\boldsymbol{x} = \mathbf{F}(\mathbf{I})$. We consider two types of intensity features $\mathbf{F}_1$ and $\mathbf{F}_2$. The first type $\mathbf{F}_1$ is holistic in nature which works by rescaling $\mathbf{I}$ into 20×20 and arranging the intensity values into $\boldsymbol{x}$. The second type $\mathbf{F}_2$ utilizes the locations of fiducial points to improve the alignment of the intensity values in $\boldsymbol{x}$. It achieves this by computing four bounding boxes of the mouth, left eye, right eye, and nose fiducial points and then we extend each bounding box by including five more pixels in each direction to include more context. Subsequently, we resize the mouth box to 7×14, each eye box to 9×11, and the nose box to 8×13. This gives a total of 400 intensity values which are arranged into the feature vector x (see Figure 3.5 for illustration). We refer to such features as *MEEN* features because they are constructed from the Mouth, left Eye, right Eye, and the Nose. As expected, we obtain better accuracy using the MEEN features.

If there are n face images $\{\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_n\}$ in a given video $\mathbf{V}$, we obtain a set of n corresponding feature vectors $\mathbf{F}(\mathbf{V}) = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$. Image set-based face recognition techniques (or simple extensions of still-image-based ones) are then used for training and/or testing.

3.4 Evaluation Protocols

A typical practical scenario for using an active face authentication system on smartphones would involve an enrollment stage in which the user enrolls their face for at least one session. After enrollment, the system is set to query mode and is expected to receive a

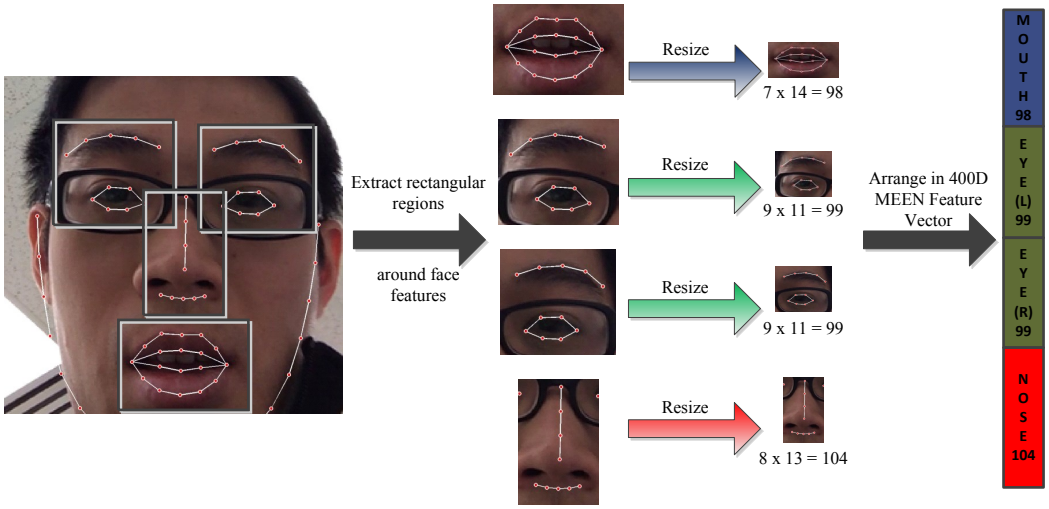


Figure 3.5: MEEN features. The regions surrounding the landmarks on the mouth, eyes and nose are extracted, rescaled and arranged into a 400D feature vector.

continuous sequence of image set queries where the overall amount of query data is much larger than the enrollment data. Since smartphones are designed to be used everywhere, the query sets may involve places and illumination settings different from those present during enrollment.

We consider in this benchmark two evaluation protocols that model this scenario. In both protocols, the overall amount of query data is bigger than that of enrollment data. In addition, the illumination settings are different from those of enrollment. In protocol 1, the system is trained on the enrollment videos from one session (e.g. session 1) and is tested on non-enrollment video clips from the other two sessions (e.g. sessions 2 and 3). In protocol 2, the system is trained on the data from two enrollment sessions (e.g. sessions 1 and 2) and is tested on non-enrollment video clips from the other session (e.g. session 3).

The test video clips are created from the non-enrollment task videos by splitting each task video into 10-second long video clips and keeping only those clips with at least one face detection. The rationale is that in practice, the system should authenticate the

user continuously and one way to achieve this is to run a query periodically. The query period we have adopted in this work is 10 seconds. Given a query video clip, the system should identify the subject present in that video clip. Accordingly, we cast the problem as a 50-class identification problem.

3.5 Experimental Results

We evaluated four still-image-based methods including Eigenfaces (EF) [117], Fisherfaces (FF) [14, 34], Large-Margin Nearest Neighbour (LMNN) [127], and Sparse Representation-based Classification (SRC) [133]. In addition, we included five image set-based methods based on Affine Hull-based Image Set Distance (AHISD) [17], Convex Hull-based Image Set Distance (CHISD) [17], Sparse-Approximated Nearest Points (SANP) [53], Dictionary-based Face Recognition from Videos (DFRV) [24], and Mean-Sequence SRC (MSSRC) [93]. We adjusted the computation of the data mean and scatter matrices in EF and FF by reweighting the contribution of the samples of each class so that all classes contribute equally regardless of the different class sizes. As in [14], we dropped the first three principal components in EF and use the subsequent 150 components to define the PCA projection matrix (adding more components does not improve the recognition rate in our experiments). Still-image-based methods process an image set query by independently classifying each vector in the query and declaring the most frequently occuring label as the winner.

Table 3.1 shows the recognition rates under protocol 1 and Table 3.2 shows the recognition rates under protocol 2. For the sake of completeness, we show in Table 3.3

Table 3.1: Recognition rates under protocol 1: The different models are trained using one session's enrollment videos and tested on video clips from another session. For each row, we show **in bold** the three highest recognition rates achieved for this experimental setting. ES = Enrollement Session, TS = Testing Session.

ES	TS	EF	FF	LMNN	SRC	AHISD	CHISD	SANP	DFRV	MSSRC
1	2	40.95	**54.48**	30.80	**52.79**	22.17	14.55	17.26	29.78	**47.21**
1	3	34.02	**45.27**	30.77	**51.18**	21.30	13.91	17.01	35.65	**46.15**
2	1	22.23	25.52	13.41	**44.18**	10.23	7.97	10.60	**32.55**	**43.06**
2	3	49.70	**56.80**	43.05	**58.58**	47.78	44.67	44.97	46.30	**60.36**
3	1	**28.05**	**24.77**	**22.05**	17.64	10.69	11.63	13.04	19.89	17.64
3	2	**55.50**	**56.01**	50.76	**51.95**	46.87	41.12	43.82	47.04	45.85

Table 3.2: Recognition rates under protocol 2: The different models are trained using the enrollment videos of two sessions and tested on video clips from the remaining session. For each row, we show **in bold** the three highest recognition rates achieved for this experimental setting. ES = Enrollement Session, TS = Testing Session.

ES	TS	EF	FF	LMNN	SRC	AHISD	CHISD	SANP	DFRV	MSSRC
$\{1, 2\}$	3	55.18	**74.85**	48.37	**72.93**	51.18	47.04	48.08	52.81	**72.19**
$\{2, 3\}$	1	**30.11**	**54.69**	25.33	24.20	14.35	16.51	16.79	**39.21**	22.14
$\{1, 3\}$	2	63.96	**71.91**	56.18	**72.93**	50.08	43.15	46.70	50.25	**69.71**

Table 3.3: Recognition rates when enrollment videos and non-enrollment test video clips come from the same session. The recognition rates for such setting are relatively good compared to those of protocol 1 and protocol 2. For each row, we show **in bold** the three highest recognition rates achieved for this experimental setting. ES = Enrollement Session, TS = Testing Session.

ES	TS	EF	FF	LMNN	SRC	AHISD	CHISD	SANP	DFRV	MSSRC
1	1	91.84	93.53	**94.65**	93.25	94.00	**95.50**	**94.84**	91.56	93.25
2	2	79.70	84.77	84.94	84.94	**86.46**	**85.45**	85.11	83.59	**85.45**
3	3	82.25	**86.98**	**83.58**	80.47	**85.06**	82.54	83.14	76.78	73.52
$\{1, 2\}$	1	92.31	93.34	**94.18**	92.96	93.90	**95.31**	**94.47**	92.03	93.06
$\{1, 2\}$	2	81.73	83.76	83.76	**85.11**	**85.96**	**85.11**	84.77	83.93	**85.79**
$\{2, 3\}$	2	81.90	**84.09**	82.57	79.86	**85.45**	**84.43**	83.59	82.57	68.02
$\{2, 3\}$	3	84.17	**91.72**	85.21	82.40	**88.46**	84.76	**85.50**	81.66	73.22
$\{1, 3\}$	1	93.25	93.25	93.71	92.78	**94.09**	**96.06**	**95.03**	92.50	92.68
$\{1, 3\}$	3	83.14	**87.43**	83.88	**85.36**	**84.62**	82.40	83.58	78.25	83.88

the recognition rates obtained by testing on the non-enrollment video clips from the same sessions used for training. These are the accuracies that would be obtained when the mobility challenge is excluded (although other challenges such as partial faces, blur,

expression, pose variations, and face/landmark localization errors are still present). All tables show results obtained using the fiducial point-based features. The less superior results obtained with holistic features are not shown due to page limitations.

Tables 3.1 and 3.2 indicate that the best performing methods are FF, SRC, and MSSRC. Yet, the recognition rates (in percentages) they achieve for protocol 1 range between 24.8 and 56.8 for FF, 17.6 and 58.6 for SRC, and 17.6 and 60.4 for MSSRC. For protocol 2, they range between 54.7 and 74.9 for FF, 24.2 and 73.9 for SRC, and 22.1 and 72.2 for MSSRC. Compared to the recognition rates obtained in 3.3, it can be seen that the evaluated methods (including state-of-the-art image set methods) have difficulty coping with the mobility challenge despite their relatively good performance when the mobility challenge is excluded while all the other challenges are kept.

3.6 Benchmark Conclusion

We have investigated how well contemporary image set-based methods combined with fiducial-point-based features can be used for active authentication on smartphones. A dataset of 750 videos was collected over three sessions with different illumination conditions to capture the kind of variations that are likely to be present with mobile devices. An examination of the videos in the dataset revealed a unique combination of properties and challenges that is specific to smartphone face videos. We utilized the fact that the user's head is always close to the phone to increase the efficiency and reduce the false positives of the face detection phase. Although the compared state-of-the-art techniques perform relatively well when the enrollment and evaluation data come from the same session, the

experiments indicate that they have difficulty addressing the variations in illumination and context that are likely to be present due to the mobility of the device.

One of the limitations of our study is that all the three sessions of any given user are collected on the same day. Therefore, the dataset misses appearance variations due to change in hair style, shaving, and/or introduction/removal of face-covering clothing such as scarves or hats. This does not limit the usefulness of the dataset since it captures a subset of the practically possible variations that has already been shown through experiments to be challenging to state-of-the-art algorithms included in the comparison.

The benchmark presented in this chapter motivates the development of better features and recognition algorithms that are invariant to the mobility challenge yet efficient to compute. Also the detection and classification of partial faces need further research to allow video clips with partial faces to be processed rather than incorrectly getting flagged them as face-empty.

Chapter 4

Bayesian Representation-Based Image Set Classification

4.1 introduction

This chapter describes Bayesian Representation-based Classification (BRC), an approach based on sparse Bayesian regression and subspace clustering for image set classification. Similar to existing representation-based approaches such as Sparse RC (SRC) and Collaborative RC (CRC), BRC assumes that a test image is approximated by a linear combination of the gallery images of the true class. Given a probe sample $\boldsymbol{y} \in \mathbb{R}^d$, we use the Relevance Vector Machine (RVM) [112, 113] to approximate the posterior distribution $\mathrm{P}(\boldsymbol{x}|\boldsymbol{y})$ of the linear representation $\boldsymbol{x} \in \mathbb{R}^N$ of the probe sample $\boldsymbol{y} \in \mathbb{R}^d$ with respect to the images in the gallery sets. The Maximum A Posterior (MAP) estimate of $\boldsymbol{x}$ is then used to classify $\boldsymbol{y}$ by finding the class contributing the most to the regression parameters $\boldsymbol{x}$. In addition, we use a Bayesian statistical framework to compare CRC, SRC, and CRC where we show that BRC

employs precision hyperpriors that are more non-informative than those of CRC/SRC. Given a probe set $\mathbf{Y}$, we analyze the assumptions of existing strategies for selecting the individual images to classify from $\mathbf{Y}$ (e.g. sequence mean [93]) and we show that these strategies can still work under milder assumptions. Finally, we present a more robust probe set handling strategy that balances the tradeoff between efficiency and accuracy. Experiments on three datasets illustrate the effectiveness of our algorithm compared to state-of-the-art set-based methods.

The contributions of this section are as follows:

1. While sparse Bayesian regression and RVM have been used for basis pursuit [131] and compressive sensing [59, 60], to the best of our knowledge, this work is the first to apply sparse Bayesian regression and automatic relevance determination to image set classification.

2. Although SRC and CRC are not formulated in a probabilistic fashion like BRC, we show that these three methods can be viewed using a common statistical framework by employing established results from Bayesian statistics literature [5, 96]. Within this framework, we conduct an intuitive comparison among these three approaches and show that each approach is the result of a different choice of a hyperprior with BRC's hyperprior being a more natural choice.

3. We provide an analysis of existing approaches for selecting images from the probe set for classification (all images versus mean image) where we show that the coding of the mean image is sufficient under milder assumptions than those required in [93]. Moreover, we present a flexible approach based on subspace clustering that achieves

a good balance between accuracy and efficiency.

4. We conduct extensive multi-fold experiments on three public face video datasets using different variants of BRC where we show that BRC achieves comparable performance with other existing methods for face recognition from image sets.

In the following sections, we describe the formulation of the BRC model and we give the corresponding algorithm. Then, we conduct the statistical comparison between CRC, SRC, and BRC. Finally, we extend BRC to image set classification and present the experimental results in the final section.

4.2 Probabilistic Model

Suppose that the images are represented by feature vectors in $\mathbb{R}^d$. We base our probabilistic model on the subspace assumption and seek to identify which of the low-dimensional subspaces spanned by the gallery samples lie closest to the probe images. In particular, we identify a test image $\boldsymbol{y} \in \mathbb{R}^d$ discriminatively by first computing the linear representation of $\boldsymbol{y}$ with respect to the images in the gallery set and finding the class that contributes the most to the representation. If we assume, following the notation in [133], that the matrix $\mathbf{A}_c \in \mathbb{R}^{d \times N_c}$ contains the N_c gallery images from all sets associated with class c and the matrix $\mathbf{A} = [\ \mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_C\ ] \in \mathbb{R}^{d \times N}$ denotes the gallery matrix of all the $N = \sum_{c=1}^{C} N_c$ gallery images, the relationship between the gallery matrix $\mathbf{A}$ and the test image $\boldsymbol{y}$ can be written as:

$$\boldsymbol{y} = \mathbf{A}\bar{\boldsymbol{x}} + \boldsymbol{\varepsilon} \tag{4.1}$$

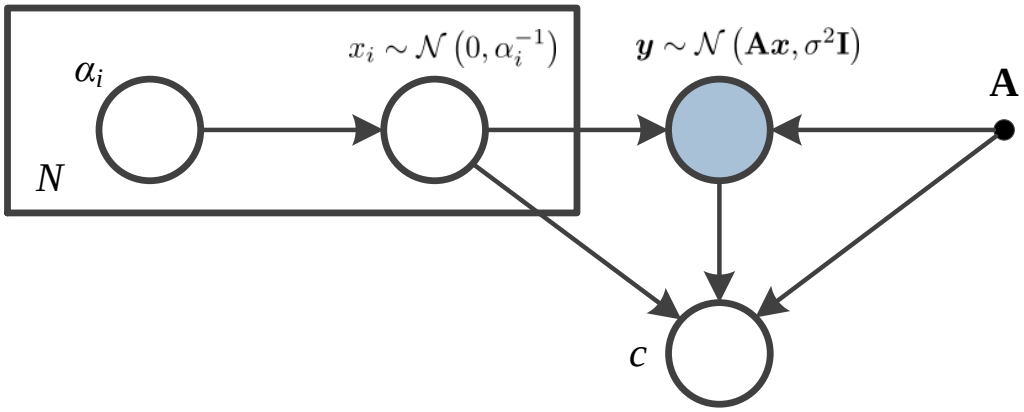


Figure 4.1: A graphical model of the sparse Bayesian regression model used in the chapter.

where $\bar{\boldsymbol{x}} \in \mathbb{R}^N$ is the true (yet unknown) representation of $\boldsymbol{y}$ with respect to the gallery and $\boldsymbol{\varepsilon} \in \mathbb{R}^d$ is a small isotropic Gaussian noise vector. If we know the true representation $\bar{\boldsymbol{x}}$, we can find the class contributing the most to the representation and with which $\boldsymbol{y}$ should be associated using the minimum residual rule of Wright et al. [133]. If we let δ_c be the $N \times N$ diagonal matrix with all zeros except at the N_c diagonal entries corresponding to the atoms of class c, the residual $r_c(\boldsymbol{y}; \bar{\boldsymbol{x}})$ corresponding to class c is given by:

$$r_c(\boldsymbol{y}; \bar{\boldsymbol{x}}) = \|\boldsymbol{y} - \mathbf{A}\delta_c\bar{\boldsymbol{x}}\|^2 \tag{4.2}$$

The class for which r_c is minimum is chosen as the label for $\boldsymbol{y}$.

The major component of classification algorithms based on such a model is how to obtain the estimate $\boldsymbol{x}$ of the true $\bar{\boldsymbol{x}}$. Although we do not know the exact values, we have prior information that the error $\boldsymbol{\varepsilon}$ is small and the parameter vector $\boldsymbol{x}$ is sparse. Our main contribution is in how to model and compute $\boldsymbol{x}$. In particular, we propose a Bayesian probabilistic model that captures the characteristics of $\boldsymbol{x}$ and (implicitly) $\boldsymbol{\varepsilon}$. Under this model, $\boldsymbol{y}$ is sampled from $\mathcal{N}(\boldsymbol{y}|\mathbf{A}\boldsymbol{x}, \sigma^2\mathbf{I})$ and $\boldsymbol{x}$ from the zero-mean normal prior

$\mathcal{N}(\boldsymbol{x}|\mathbf{0}, \boldsymbol{\Lambda}^{-1})$ where the precision hyperparameter $\boldsymbol{\Lambda} = \text{diag}(\alpha_1, \ldots, \alpha_N),\ \alpha_i > 0\ \forall\ i$. The corresponding graphical model is shown in Figure 4.1. The precision α_i is a measure of our (un)certainty about x_i. The sparsity of the true $\bar{\boldsymbol{x}}$ implies that the distribution of some of the coefficients x_i will be concentrated at zero (i.e. $\alpha_i \to \infty$) while others will have some variance (i.e. α_i is some finite positive value). Before observing the test image or knowing its associated class, we have no prior information on values the precision hyperparameter α_i may assume. Accordingly, we place a non-informative (i.e. flat) hyperprior on α_i [82, 112, 114].

An advantage of the flat precision hyperpriors is their sparsifying effect [113]. It takes place whenever the linear dependence between the target values (the test vector $\boldsymbol{y}$ in our case) and the basis vectors (the gallery vectors in $\mathbf{A}$ in our case) can be captured with a small subset of the basis vectors in $\mathbf{A}$. In that case, the flat hyperprior for $\boldsymbol{\alpha}$ allows the posterior probability mass $\text{P}(\alpha_i|\boldsymbol{y})$ for many of the coefficients in $\boldsymbol{\alpha}$ to concentrate at very large or infinite values. This in turn makes the corresponding $\text{P}(x_i|\boldsymbol{y})$ behave like a Dirac-Delta distribution focused at $x_i = 0$ and can result in a very sparse $\boldsymbol{x}$. When $\boldsymbol{y}$ is similar to one of the subjects encoded in $\mathbf{A}$, we expect the above model would utilize the low-dimensional subspace relationship and produce a sparse $\boldsymbol{x}$ that can easily discriminate the class of $\boldsymbol{y}$. The above model is akin to the well-established Bayesian models for sparse linear regression like those described in [16, 112, 113].

4.3 Classification Algorithm

Suppose that we want to classify a candidate test image $\boldsymbol{y}$ selected from the probe set. Our classification algorithm is based on the *empirical Bayes* approximation framework [16, 82, 83, 113] and has three main steps. (a) First, we compute the value of the precision hyperparameter $\boldsymbol{\alpha}^*$ that maximizes the marginal likelihood function $\mathrm{P}(\boldsymbol{y}|\boldsymbol{\alpha})$. (b) Next, we use the optimal precision estimate $\boldsymbol{\alpha}^*$ to construct the posterior distribution $\mathrm{P}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^*)$ of the regression coefficient vector $\boldsymbol{x}$. This distribution is a Gaussian and so the mean and the mode are coincident with a closed form. (c) Finally, we use the class-specific residuals $r_c(\boldsymbol{y}; \hat{\boldsymbol{x}})$ computed at the *Maximum A Posteriori (MAP)* estimate of the regression parameters $\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \mathrm{P}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^*)$ to find the class with which $\boldsymbol{y}$ should be associated. We next describe these steps in more details.

To perform the optimization in the first step, we use the probability sum rule to expand the marginal likelihood $\mathrm{P}(\boldsymbol{y}|\boldsymbol{\alpha}) = \int \mathrm{P}(\boldsymbol{y}|\boldsymbol{x})\mathrm{P}(\boldsymbol{x}|\boldsymbol{\alpha})d\boldsymbol{x}$:

$$\begin{aligned}
\mathrm{P}(\boldsymbol{y}|\boldsymbol{\alpha}) =& \int \mathcal{N}(\boldsymbol{y}|\mathbf{A}\boldsymbol{x}, \frac{1}{\sigma^2}\mathbf{I})\, \mathcal{N}(\boldsymbol{x}|\mathbf{0}, \boldsymbol{\Lambda}^{-1})d\boldsymbol{x} && (4.3)\\
=& \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \Sigma_{\boldsymbol{y}|\boldsymbol{\alpha}} = \sigma^2\mathbf{I} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^T) && (4.4)\\
=& \text{const} \left|\Sigma_{\boldsymbol{y}|\boldsymbol{\alpha}}\right|^{-0.5} \times \exp\left(-\frac{1}{2}\boldsymbol{y}^T\Sigma_{\boldsymbol{y}|\boldsymbol{\alpha}}\boldsymbol{y}\right) && (4.5)
\end{aligned}$$

Taking the log of the marginal likelihood (4.5) (and dropping constants) gives:

$$f(\boldsymbol{\alpha}) = \ln\mathrm{P}(\boldsymbol{y}|\boldsymbol{\alpha}) = -\frac{1}{2}\ln\left|\Sigma_{\boldsymbol{y}|\boldsymbol{\alpha}}\right| - \frac{1}{2}\boldsymbol{y}^T\Sigma^{-1}_{\boldsymbol{y}|\boldsymbol{\alpha}}\boldsymbol{y} \qquad (4.6)$$

Due to the flat precision hyperprior and the inherent sparse dependence between the

test and gallery vectors, the optimal precision $\boldsymbol{\alpha}^*$ will have many of its components infinite. The corresponding coefficients in $\boldsymbol{x}$ will have distributions concentrated at 0 because their posterior variances are zeros. This produces a sparse $\boldsymbol{x}$ [16, 113].

There are two types of iterative algorithms for maximizing the nonconcave function (4.6) [16]. One type starts the optimization with all the components in $\boldsymbol{\alpha}$ present and gradually discards those precisions that approach infinity in addition to their corresponding atoms. Earlier iterations made by such algorithms are quite expensive, taking $O(N^3)$ time and $O(N^2)$ space whereas, the last few iterations solve much smaller problems involving a small subset of the atoms in $\mathbf{A}$. In our work, we use the faster incremental algorithm proposed by Tipping and Faul [114] which starts with all components of $\boldsymbol{\alpha}$ absent (i.e. set to infinity) and in each iteration identifies one component to (a) activate (by changing from infinity to a finite optimal value), (b) update or (c) discard (by resetting to infinity). This algorithm runs very fast in our case as the number of active components is typically low. This significantly reduces the dimension of the linear subproblems to be solved in each iteration.

Once $\boldsymbol{\alpha}^*$ is computed, we discard the atoms from $\mathbf{A}$ and the regression coefficients from $\boldsymbol{x}$ corresponding to the infinite precision components that we exclude from $\boldsymbol{\alpha}^*$. The corresponding posterior parameter distribution $\mathrm{P}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^*)$ has the following form:

$$\mathrm{P}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^*) = \frac{\mathcal{N}(\boldsymbol{y}|\mathbf{A}\boldsymbol{x}, \sigma^2\mathbf{I})\, \mathcal{N}\left(\boldsymbol{x}|\mathbf{0}, \boldsymbol{\Lambda}^{*-1}\right)}{\mathrm{P}(\boldsymbol{y}|\boldsymbol{\alpha}^*)} \tag{4.7}$$

$$= \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) \tag{4.8}$$

where $\mathbf{\Lambda}^* = \text{diag}(\boldsymbol{\alpha}^*)$ and

$$\mathbf{\Sigma}^* = \left(\mathbf{\Lambda}^* + \sigma^{-2}\mathbf{A}^T\mathbf{A}\right)^{-1}, \boldsymbol{\mu}^* = \sigma^{-2}\mathbf{\Sigma}^*\mathbf{A}^T\boldsymbol{y} \tag{4.9}$$

We use the point $\hat{\boldsymbol{x}} = \boldsymbol{\mu}^*$ for calculating the class-specific residuals and therefore identifying $\boldsymbol{y}$. As $\text{P}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^*)$ is a Gaussian, the point $\boldsymbol{\mu}^*$ is the most probable (i.e. MAP) representation and also the posterior average of all possible representations which makes it the most intuitive choice.

4.3.1 Noise Variance σ^2

For performance and robustness reasons, we decided to keep the noise variance σ^2 constant. This corresponds to choosing a dirac-delta prior centered at the constant of choice for σ^2. Performance-wise, this allows the iterative optimization algorithm to use more efficient equations for updating $\boldsymbol{\alpha}$ [114]. In addition, this allows us to utilize our prior knowledge that the features in $\boldsymbol{y}$ have non-trivial measurement error as they are computed based on quantized input intensities. Specifically, this prevents the algorithm from trying representations that overfit $\boldsymbol{y}$ which can occur if the algorithm picks infinitesimal values of σ^2. Such values will force the algorithm to use denser $\boldsymbol{x}$ in an attempt to precisely fit $\boldsymbol{y}$ while in reality it is trying to fit the quantization error component in $\boldsymbol{y}$. Using dense $\boldsymbol{x}$ would mean bad discriminative performance as well as increased computational requirements in terms of time and space.

4.3.2 Choosing The Candidate Test Image

Given a probe set $\mathbf{Y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_{N_y}\}$, there are several ways to choose from $\mathbf{Y}$ a candidate image $\boldsymbol{y}$ to apply BRC on. One way is to choose every image in $\mathbf{Y}$ as a candidate for classification and then choose the most frequent classification as the label for the complete $\mathbf{Y}$. We call this variant Full Sequence-BRC (FS-BRC). The advantage of FS-BRC is the robustness to outliers in the probe set (e.g. wrong detections or badly-aligned faces). During majority voting, the few outliers could only result in as few invalid labels that will be dominated by the correct majority. Although FS-BRC is embarrassingly parallel and scales very well with parallel computing, FS-BRC can be infeasible without sufficient computing resources. Accordingly, we analyze two practical alternatives to FS-BRC.

Mean Sequence-BRC (MS-BRC): It is shown in [93] that coding the complete $\mathbf{Y}$ can be reduced to the coding of its sample mean $\bar{\boldsymbol{y}}$ under the strong assumption that all images in $\mathbf{Y}$ share the same true sparse representation $\bar{\boldsymbol{x}}$, i.e. $\boldsymbol{y}_i = \mathbf{A}\bar{\boldsymbol{x}}_i + \boldsymbol{\varepsilon}_i, \bar{\boldsymbol{x}}_i = \bar{\boldsymbol{x}} \forall i$. The assumption is too strong; it implies that all sample images in $\mathbf{Y}$ are identical in terms of pose, illumination, and all other conditions (except for the noise term) which is rarely the case in practice. Meanwhile, relatively good results were reported in [93] using this strategy and we also found that it worked relatively well in our experiments. To explain this observation, we perform an analysis of $\bar{\boldsymbol{y}}$ where we show that coding $\bar{\boldsymbol{y}}$ can be sufficient under milder conditions in addition to some other desirable properties of $\bar{\boldsymbol{y}}$ which justify the relatively good performance of MS-SRC/MS-BRC.

Without assuming a constant $\bar{\boldsymbol{x}}$ shared by $\boldsymbol{y}_i, \forall i$, the mean image $\bar{\boldsymbol{y}}$ is derived as

follows:

$$\bar{\boldsymbol{y}} = \frac{1}{N_y}\sum_{i=1}^{N_y}(\mathbf{A}\bar{\boldsymbol{x}}_i + \boldsymbol{\varepsilon}_i) = \mathbf{A}\underline{\bar{\boldsymbol{x}}} + \bar{\boldsymbol{\varepsilon}} \tag{4.10}$$

To recover the correct label from $\bar{\boldsymbol{y}}$, the mean vector $\underline{\bar{\boldsymbol{x}}}$ has to be sufficiently sparse, the nonzeros of $\underline{\bar{\boldsymbol{x}}}$ has to correspond to the gallery samples of the true class, and the mean error term $\bar{\boldsymbol{\varepsilon}}$ has to have relatively small norm. Since the L2-norm is convex, $\|\bar{\boldsymbol{\varepsilon}}\|_2$ will be small if $\|\boldsymbol{\varepsilon}_i\|$ is relatively small $\forall i$ because

$$\|\bar{\boldsymbol{\varepsilon}}\|_2 \leq \sum_i \|\boldsymbol{\varepsilon}_i\|_2/N_y \leq \max_i \|\boldsymbol{\varepsilon}_i\|_2 \tag{4.11}$$

In fact, a better bound can be proved if we assume that the errors are I.I.D. Gaussian noise, in which case $\bar{\boldsymbol{\varepsilon}}$ will also be Gaussain with the covariance scaled down by $\sqrt{N_y}$.

The other two conditions on $\underline{\bar{\boldsymbol{x}}}$ are satisfied if all images in $\mathbf{Y}$ correspond to the face of the true identity (i.e. no outliers) and the support of $\bar{\boldsymbol{x}}_i$ is the same $\forall i$ (i.e. the corresponding faces share the same pose/subspace) or their supports are not too different (i.e. the corresponding faces have small pose variations) in order to guarantee that the support of $\underline{\bar{\boldsymbol{x}}}$ remains sparse. This explains why MS-SRC was found to work well in more general settings than dictated by the constant, shared representation assumption. The summary of steps for the mean-sequence variant of BRC (MS-BRC) is given in Algorithm 1. The steps for the other variants are omitted as it is straightforward to modify MS-BRC to obtain either FS-BRC or C-BRC (described next).

Clustering-BRC (C-BRC): In practice, $\mathbf{Y}$ can have outliers and/or wild pose variations. Applying MS-BRC in such situations may lead to incorrect classifications (i.e.

Input: $d \times N$ Gallery Matrix $\mathbf{A}$, $d \times N_Y$ probe image set $\mathbf{Y} = \{\boldsymbol{y}_i\}_{i=1}^{i=N_Y}$, noise variance σ^2

Output: class label

1 Compute the candidate test image $\boldsymbol{y} = \frac{1}{|\mathbf{Y}|} \sum_{\boldsymbol{y}_i \in \mathbf{Y}} \boldsymbol{y}_i$;

2 l_2-normalize the atoms of $\mathbf{A}$ and $\boldsymbol{y}$;

3 Estimate $\boldsymbol{\alpha}^*$ by maximizing (4.6) using the incremental algorithm of Tipping and Faul [114];

4 Discard infinite components from $\boldsymbol{\alpha}^*$ and the corresponding atoms from $\mathbf{A}$ then compute $\hat{\boldsymbol{x}} = \boldsymbol{\mu}^*$ as in (4.9);

5 Use (4.2) to compute the residual $r_c(\boldsymbol{y}; \hat{\boldsymbol{x}})$ for each class c. Associate $\boldsymbol{y}$ with the class c for which r_c is minimum;

Algorithm 1: MS-BRC Algorithm Summary

outliers will corrupt $\bar{\boldsymbol{y}}$ which is used solely for classifying the whole set). An intuitive way to deal with this is to partition $\mathbf{Y}$ into k clusters where each cluster contains images sharing the same subspace (i.e. faces having the same pose). Then one can generate a label for each cluster with MS-BRC and use *weighted* majority voting to determine a label for the complete set. For subspace clustering, we modify the algorithm in [32] where we replace the L1 regularization term with L2 to simplify and speed up the algorithm.

C-BRC can be thought of as an approximation to FS-BRC which reduces the redundant computations FS-BRC carries out when it codes near-identical samples. As we increase the number of clusters k, outliers are assigned to even smaller clusters which helps limit their corruption effect to only these cluster. If $k = 1$, C-BRC reduces to MS-BRC. If $k = N_y$, C-BRC turns into FS-BRC.

4.4 Comparison of CRC, SRC, and BRC

The three algorithms CRC, SRC, and BRC use the solution of a regression problem to perform classification. SRC places an l_1-norm penalty on the regression coefficient $\boldsymbol{x}$ as a sort of regularization [133]. In effect, SRC solves the following LASSO optimization problem:

$$\text{SRC} : \hat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^N}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_1 \tag{4.12}$$

Similarly, CRC places an l_2-norm penalty on the regression coefficient $\boldsymbol{x}$ for regularization. In effect, CRC solves the ridge optimization problem:

$$\text{CRC} : \hat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^N}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|^2 + \frac{1}{2}\lambda\|\boldsymbol{x}\|^2 \tag{4.13}$$

It is possible to interpret the regularization terms used by CRC and SRC as prior distributions for $\boldsymbol{x}$. In the case of CRC, the regularization term is equivalent to requiring $\boldsymbol{x}$ to have a prior of the form $\mathcal{N}(\mathbf{0}, \frac{1}{\sqrt{\lambda}}\mathbf{I})$. The prior in the case of SRC comes in the form of a zero-mean Laplace distribution [16] with a rate parameter λ for each coefficient in $\boldsymbol{x}$:

$$\text{P}(x_i) = \mathcal{L}(x_i) = \frac{\lambda}{2}\exp\left(-\lambda|x_i|\right) \tag{4.14}$$

Assuming the observed test vector $\boldsymbol{y}$ has a Gaussian distribution $\text{P}(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\mathbf{A}\boldsymbol{x}, \mathbf{I})$, the cost function of either CRC or SRC is obtained when we attempt to find the MAP estimate $\hat{\boldsymbol{x}}$ that minimizes the negative log of the posterior distribution $\text{P}(\boldsymbol{x}|\boldsymbol{y})$ (constants

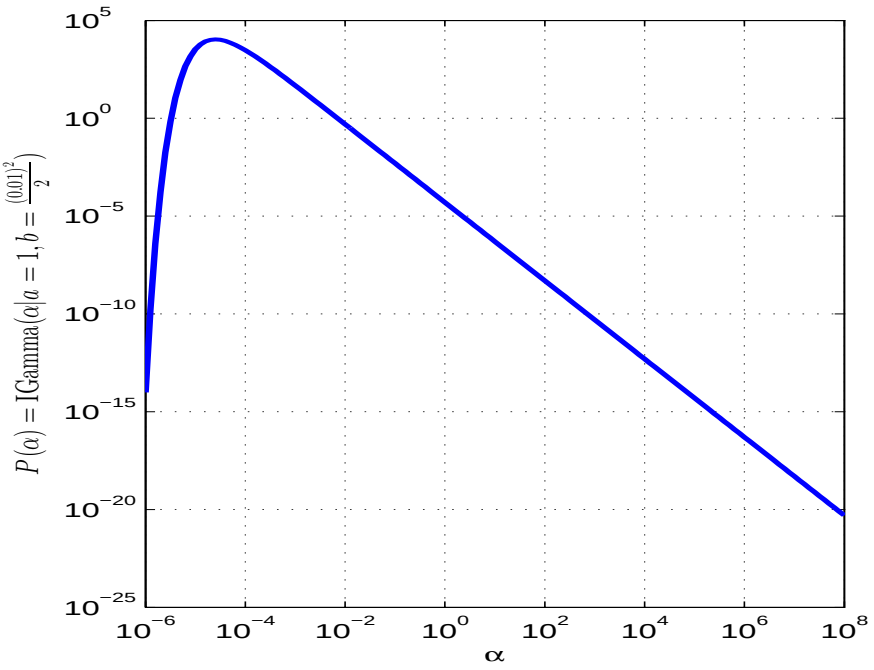


Figure 4.2: The inverse Gamma hyperprior with shape $a = 1$ and scale $b = \frac{\lambda^2}{2}$ that is (implicitly) imposed on α by SRC. The curve is drawn for $\lambda = 0.01$.

dropped):

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^N}{\operatorname{argmin}} \{-\log \mathrm{P}(\boldsymbol{x}|\boldsymbol{y})\} \tag{4.15}$$

$$= \underset{\boldsymbol{x}\in\mathbb{R}^N}{\operatorname{argmin}} \{-\log \mathrm{P}(\boldsymbol{y}|\boldsymbol{x}) - \log \mathrm{P}(\boldsymbol{x})\} \tag{4.16}$$

Both CRC and SRC can be interpreted as special cases of the proposed probabilistic model. This is shown by replacing the non-informative hyperprior $\pi(\alpha_i)$ with an informative one. For the case of CRC, the fact that α_i is the constant λ can be realized by choosing a dirac-delta hyperprior for α_i centered at λ:

$$\pi(\alpha_i) = \delta_\lambda(\alpha_i) \tag{4.17}$$

For SRC, we use the fact that the Laplace prior $\mathcal{L}(x_i)$ on x_i has been shown to be equivalent to a zero-mean Gaussian distribution with a marginalized precision α_i that follows an inverse Gamma hyperprior with shape $a = 1$ and scale $b = \frac{\lambda^2}{2}$ [5, 96] (see Figure 4.2):

$$\mathcal{L}(x_i) = \int_{\mathbb{R}^+} \mathcal{N}(x_i|0, \alpha_i^{-1}) \times \pi(\alpha_i)\, d\alpha_i \tag{4.18}$$

$$\pi(\alpha_i) = \frac{\lambda^2}{2\alpha_i^2} \exp\left(\frac{-\lambda^2}{2\alpha_i}\right), \alpha_i > 0 \tag{4.19}$$

These hyperpriors prefer certain ranges of possible precisions more than others. CRC sacrifices most of the flexibility by fixing the precision at a constant value a priori for the benefit of having a fast, closed-form solution. Although to a lesser extent, SRC does also sacrifice some flexibility by favoring, without prior evidence, certain ranges of precisions to others (see Figure 4.2) for the sake of getting a convex energy function that is easier to optimize for the MAP estimate of $\boldsymbol{x}$. Our method uses a non-informative hyperprior that treats all precisions uniformly. This is a better model of the fact that before looking at the data we do not have any information regarding the correct precision values to use, making BRC's flat hyperprior a more natural choice from a statistical viewpoint than either CRC's or SRC's.

4.5 Experimental Evaluation

We have conducted extensive experiments to compare the performance of the proposed algorithms (i.e. MS-BRC, C-BRC and FS-BRC) against several existing algorithms for image-set classification. The compared vector space-based methods include AHISD [17],

its convex variant (CHISD) [17], SANP [53], DFRV [24], and SSDML [149]. The compared representation-based methods include MS-SRC [93] and a variation of MS-SRC that uses CRC [145] for classifying the mean of the sequence (MS-CRC). The comparison also includes a neural network-based method which is DRM [50]. The manifold-based methods we have included are CDL [122] and LEML [55]. For existing methods, we have used the source code provided by the original authors and set the parameters according to the recommendations made in their respective papers. The only exception to this are MS-CRC and MS-SRC which we have implemented ourselves. To guarantee a fair comparison, the same features and dataset splits were used to compare all the methods. For the manifold-based methods, we generate an SPD descriptor from each image by dividing the image into a 6×6 grid and computing the covariance from each cell based on the per-pixel intensities, Gabor filter responses, and normalized spatial coordinates. For the BRC-based methods, we found that setting the noise variance $\sigma^2 = (0.02)^2$ gives reasonable performance. For C-BRC, we use $k = 10$ clusters. For probe sets with fewer than k images, C-BRC skips clustering and proceeds to classify each image in the probe set as done by FS-BRC. We have also compared with soft-margin Support Vector Machine (SVM) [18] as an example of a standard classification approach where majority voting is used to combine the labels of the set samples into a single label for the image set.

The datasets used in our comparison are described below and three example images from each dataset are shown in Figure 4.3.

4.5.1 YouTube Celebrities (YTC)

The YTC dataset contains 1,910 YouTube-downloaded videos of 47 subjects [62] (with total duration of 207.57 minutes). For a given subject, the videos are short segments clipped from three longer, parent videos downloaded from YouTube. YTC has been built to be very challenging for face tracking and recognition by choosing low resolution videos with wild variations in pose, scale, hair style, make-up, illumination, motion and number of people per frame.

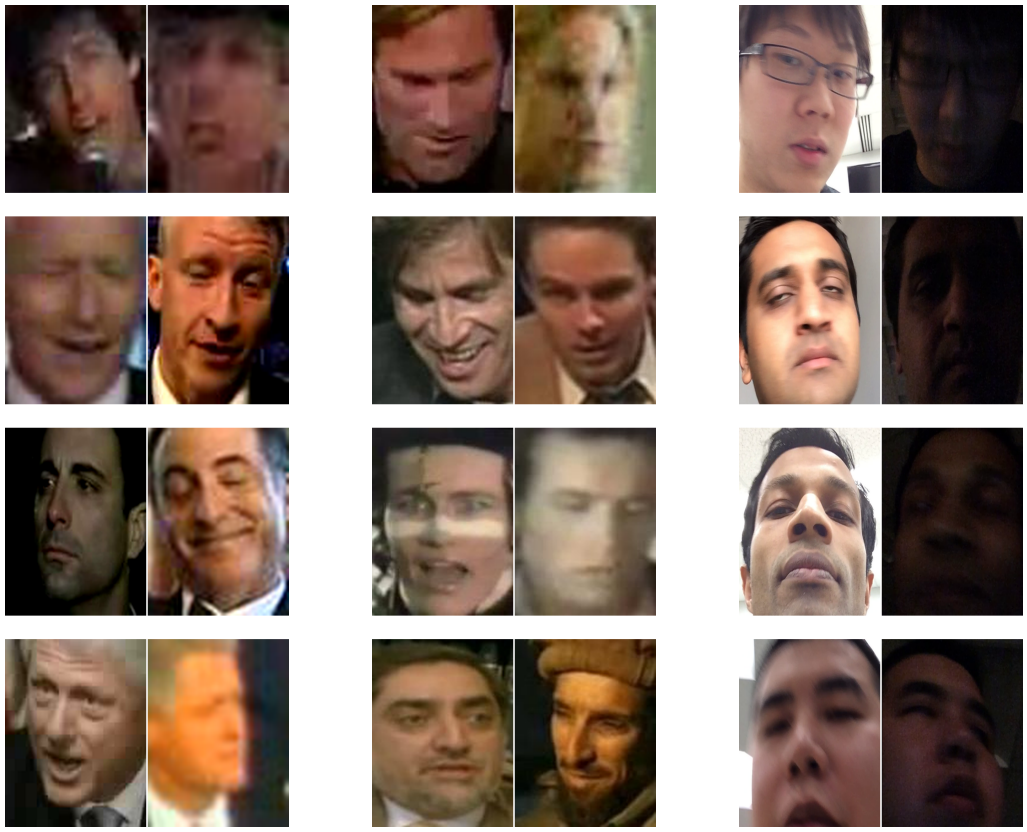
Experimental Protocol: We run ten-fold cross-validation experiment. The $9 \times 47 = 423$ videos in each fold are randomly selected from the complete dataset while minimizing the overlap between different folds as much as possible.

Feature Extraction: We use the Viola-Jones (VJ) detector [119] to locate the faces in each video. Then we use the eye locations detected using the method of Asthana et al. [7] to align the subject's face to a standard, 30×36 pixel frame. The intensities are histogram equalized and arranged in a 1080D feature vector. We use the feature vectors from a given video define a corresponding image set.[1]

4.5.2 YouTube Faces (YTF)

The YTF dataset contains 3,425 videos of 1,595 subjects with diverse ethnicities [132]. The total video duration of YTF is 431.22 minutes. Similar to YTC, YTF videos are downloaded from YouTube and are very challenging for face recognition. We conduct our experiments on those subjects with four or more videos available. This results in

[1] We have not cleaned any of the bad detections or misaligned faces in an effort to test the robustness of the compared methods to such outliers.



YTC YTF MobFaces

Figure 4.3: Sample face pairs from YTC (first column), YTF (second column) and MobFaces (third column). Each pair of faces in each column belong to the same subject. YTC and YTF photos reveal large intra-class appearance variations and low resolution. MobFaces photos are relatively frontal but they reveal some challenges such as blur and intra-class variations in illumination and context due to the change in sessions.

226 subjects. After randomly dropping one subject, we randomly split the remaining 225 subjects into five mutually exclusive groups, with 45 subjects each. We repeat the experiment for each group where we use the first three videos of each subject as gallery sets and the remaining videos for testing. Since the dataset provides aligned face images, we extract intensity features from each image by cropping the central 100×100 box from each image, resizing it to 30×36, and histogram-equalizing it.

4.5.3 Mobile Faces (MobFaces)

The MobFaces dataset contains 750 videos of 50 subjects taken by a smartphone's front camera during various user interactions with the phone [35]. The total video duration is 441.02 minutes. There are three sessions of five videos each (one enrollment + four tasks) per subject. Each session is taken under a different illumination and/or in a different place. The dataset includes some of the unique challenges of mobile-based continuous facial authentication such as the wild variations in illumination and context due to the mobility of the device. We compute the features using the same pipeline we developed for the YTC dataset. Although the features used in this experiment are different from [35], we adopt the two evaluation protocols suggested in [35] by dividing the task videos into ten-second long clips and treating each clip as a separate query. In the first protocol (MobFaces-I), training is done using only the 50 enrollment videos of one session and testing is performed on the clipped task video clips from the other two sessions. In the second protocol (MobFaces-II), training is done on the 100 enrollment videos of two sessions and testing is done on the task video clips of the remaining session. Clipping test videos leads to 1065 clips from

Table 4.1: The recognition rates of the compared methods on YTC, YTF, MobFaces-I and II. We have highlighted in bold the rates of the top three performing methods for each dataset. Although YTC and YTF have similar challenges, the rates obtained for YTC are higher because the test protocol for YTC guarantees that for each test video clip there is a corresponding gallery video clip such that both are segments from the same parent YouTube video. For DRM, we report the recognition rate obtained with histograms of LBP features as recommended in [50] except for MobFaces where we report the performance with the same intensity features used with other methods as DRM performed better with these features on MobFaces.

Methods	YTC	YTF	MobFaces-I	MobFaces-II
AHISD	57.27	17.18	26.12	39.39
CHISD	64.79	32.99	21.96	35.76
SANP	66.99	31.62	21.96	34.94
DFRV	66.70	36.77	28.30	42.62
SSDML	69.22	34.02	22.95	38.27
SVM	68.79	41.92	24.48	44.18
MS-CRC	66.88	43.64	50.09	72.52
FS-CRC	66.52	42.27	50.54	72.26
MS-SRC	74.68	45.02	41.29	59.79
FS-SRC	75.60	47.77	44.43	63.20
DRM	70.35	43.99	37.06	62.42
CDL	67.62	41.92	40.21	64.97
LEML	73.26	48.45	44.39	61.09
MS-BRC	**75.00**	**48.80**	**51.49**	**72.65**
C-BRC-30	**76.91**	**55.33**	**55.07**	**76.10**
FS-BRC	**76.91**	**56.36**	**55.16**	**76.62**

the first session, 587 from the second, and 666 from the third.

4.5.4 Results

Table 4.1 shows the mean of the recognition rates of the compared methods for YTC, YTF, and the two protocols of Mob-Faces. The results show the superiority of the proposed method in comparison with the other methods. The Table, as well as Figure 4.4, also show that there is a non-trivial gap of performance between MS-BRC and FS-BRC while C-BRC-30 is almost as accurate as FS-BRC over all datasets. This confirms the analysis presented earlier in Section 4.3.2 of the shortcomings of reducing an image set to its sample

Table 4.2: The train and test times for competing methods in seconds.

Method	Train (s)	Test (s)
AHISD	18.81	18.82
CHISD	N/A	10.85
SANP	N/A	82.99
DFRV	2870.99	45.63
SSDML	3447.62	56.92
SVM	3991.9	6.8
MS-CRC	13.38	0.25
FS-CRC	13.38	61.8
MS-SRC	N/A	3.88
FS-SRC	N/A	861.6
DRM	12511.44	4.67
CDL	275.28	2.94
LEML	2667.53	153.11

Table 4.3: The sequential and parallel test times (using 12 processors) for BRC-based methods in seconds.

Method	Sequential (s)	Parallel (s)
MS-BRC	4.1	N/A
C-BRC-5	27.2	11.8
C-BRC-10	77.3	18.4
C-BRC-15	98.2	19.7
C-BRC-20	132.5	22.8
C-BRC-25	175.9	28.5
C-BRC-30	195.9	32.5
FS-BRC	1464.8	159.1

mean (i.e. MS-BRC) as compared to the robustness of the clustering-based approach (i.e. C-BRC).

The results on MobFaces in Table 4.1 also indicate that l_1-regularization (i.e. MS-SRC) can sometimes lead to inferior results compared to those obtained with the l_2-regularization (i.e. MS-CRC) which was also observed in the experiments conducted in [145]. On the other hand, MS-BRC yields a slightly better performance than MS-CRC on MobFaces whereas C-BRC-30 and FS-BRC are the top performers.

Running Times: In Tables 4.2 and 4.3, we report the time taken to train different methods on the first fold of YTC as well as the time taken to classify a single test image set of size 165 from YTC. No training time is reported for our method as it is training free. All methods were tested on MATLAB using a system running at 2.2 GHz. The testing time of

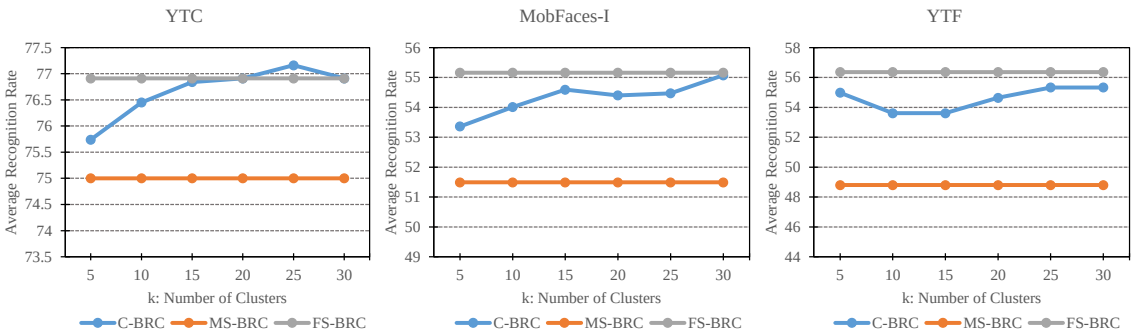


Figure 4.4: The average recognition rates of C-BRC obtained on YTC, MobFaces-I, and YTF as a function of the number of clusters k (The plot for MobFaces-II is in the supplemental material). C-BRC improves on the performance of MS-BRC and generally approaches the performance of FS-BRC as k is increased.

all methods was measured using a single-threaded MATLAB process. The timings reveal that C-BRC-30 runs almost 7.5 times faster than FS-BRC although C-BRC-30 provides very close accuracy. In addition, the timings also show the ability of BRC-based methods to utilize parallel processing to reduce the running time. The clustering step contributes 0.54 seconds to the reported test time of C-BRC-30. Its contribution is lower for smaller values of k.

4.6 Summary

We proposed BRC, an approach based on Bayesian regression and RVM for face recognition from image sets. After approximating the posterior distribution of the linear representation $\boldsymbol{x}$ of the the probe $\boldsymbol{y}$ with respect to the gallery, BRC applies the residual decision rule of SRC on the MAP estimate of $\boldsymbol{x}$ to classify $\boldsymbol{y}$. To prevent overfitting the non-trivial intensity quantization errors present in $\boldsymbol{y}$, BRC is made more robust (and efficient) by fixing the noise variance σ^2. A comparison of the models employed by BRC, CRC, and SRC was made in Section 4.4 where it was shown that BRC uses a more non-informative

hyperprior for precision than those of CRC and SRC. The extensive experiments conducted in this chapter indicate that BRC-based methods (FS-BRC/C-BRC in particular) outperform state-of-the-art image set methods on the YTC, YTF, and MobFaces. The speed-accuracy tradeoff of our algorithm can be effectively controlled by setting the k parameter of the C-BRC variant of our method. While this chapter introduced BRC in the context of image set-based face recognition, the extension of BRC to other classification tasks in computer vision is an interesting research direction to pursue in the future.

Chapter 5

Log-Euclidean Subspace Feature Learning for Image Set Classification

In this chapter, we present an approach that combines the extraction of robust SPD features, discriminative Subspace Feature Learning (SFL), and sparse coding for the purpose of image set classification. In this approach, we first describe each image by generating a Grid of Region Covariance Matrices (GRCMs) that are fused into a single compressed SPD descriptor; then we map the SPD descriptor from the SPD manifold $\mathbb{S}_+^Q$ to the Log-Euclidean tangent space $\mathrm{T}_{\mathbf{I}}\mathbb{S}_+^Q$ and use a dictionary of atoms from $\mathrm{T}_{\mathbf{I}}\mathbb{S}_+^Q$ to represent each gallery image set. While previous LE approaches for image set classification extract from each image set, a single or very few LE samples that have very high dimensionality, our approach extracts from each set many LE samples of a much lower dimensionality, reducing the possibility of over-fitting. Given the LE features, we then formulate an optimization problem for learning an embedding into a lower-dimensional LE tangent space $\mathrm{T}_{\mathbf{I}}\mathbb{S}_+^q$ in which the data has a more discriminative subspace structure. To classify

a probe image set, we use the LE feature transform computed during training to embed the dictionary of LE atoms extracted from the probe image set. Next, we apply the LE sparse coding approach [42, 140] to classify the embedded probe atoms with respect to the augmented gallery dictionary. We also consider the case of imbalanced and/or large gallery image sets and we propose Dictionary-Based SFL (DBSFL) which address these issues by integrating dictionary learning into SFL, coding and residual-based classification.

Extensive experiments on four public datasets show that our approach outperforms many state-of-the-art methods. When deep features are used as input, our experiments show that SFL as well as DBSFL lead to improved performance compared to other competing methods. We also run an empirical ablation analysis to understand how the different components of our approach contribute to the final performance. A preliminary version of this work appeared in [36]. In order of importance, the contributions that are included in that preliminary version can be summarized as follows:

- A feature learning and dimensionality reduction algorithm that leads to a more discriminative subspace structure, subsequently enhancing the performance of representation-based classifiers (like SRC) with nonlinear input features (such as GRCM-based LE features). Since it reduces the learning problem into that of solving a single generalized eigenvalue problem in a non-iterative fashion, the algorithm is also efficient.

- An image set feature extractor which models each image set as a dictionary of LE atoms that is more robust to local deformations and has significantly lower dimensions than other LE image set descriptors [55, 122, 126], making the proposed

LE features more robust to over-fitting. In our experiments, we show that the proposed features also improve the performance of another recent LE image set method proposed in [55].

- To the best of our knowledge, this work is the first to apply SRC for image set classification on LE tangent spaces. Note that SRC has been extended to image set classification on Euclidean space by Ortiz et al. [93] and to other classification problems on LE tangent spaces [42, 140]. Our experiments show that the proposed approach outperforms existing methods on three public video face datasets.

In addition to the above contributions, the current extended version of the work includes the following contributions:

- A dictionary-based version of the SFL algorithm that addresses imbalanced and/or large gallery image sets. The Dictionary-Based SFL (DBSFL) improves the classification-time efficiency of SFL and leads to improved classification accuracy especially when the gallery image sets are imbalanced.

- We extend the previous experiments by including experiments on a fourth, recently released dataset, namely the IARPA Janus Benchmark-B (IJB-B) and by experimenting with various deep features. We show that the proposed methods SFL/DBSFL lead to faster and more accurate performance compared to other competing methods.

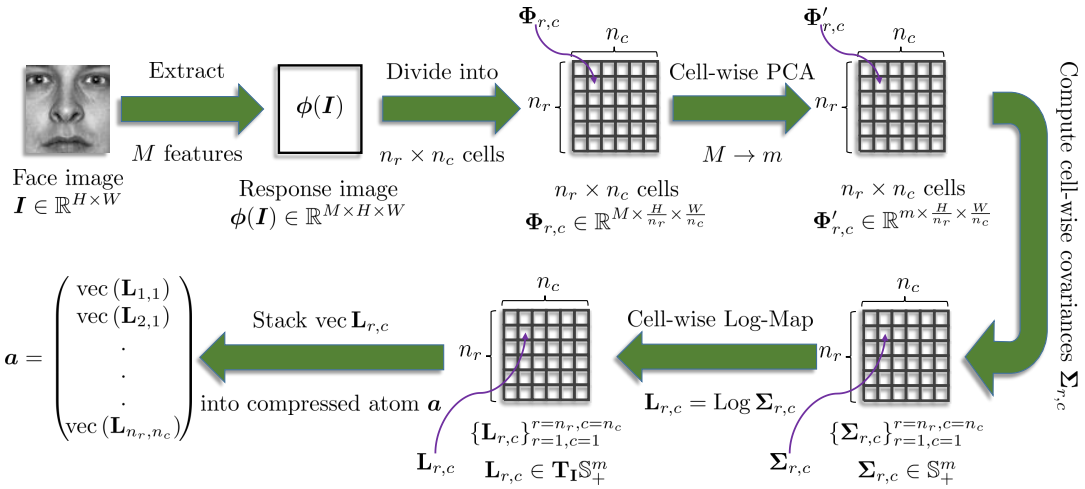


Figure 5.1: The steps for extracting the LE features from each image.

5.1 Log-Euclidean Grid of Covariance Matrices

We describe the three components of our approach (description, embedding, and coding) in the following subsections.

5.1.1 Image Set Descriptor: Dictionary of LE Atoms

Existing SPD image set descriptors, like those used in [55, 122, 126] compute a single or a small number of SPD matrices per image set. These descriptors suffer from some drawbacks. One such drawback is the curse of dimensionality as each SPD matrix descriptor has the dimensions $WH \times WH$ (or more) assuming the images have size $W \times H$ (the descriptor is 160,000-D for images as small as 20×20). The image set may also contain too few images to reliably compute such high-dimensional descriptors, leading to undersampling at the level of each descriptor. Undersampling at the gallery level (and subsequent overfitting) may also be a problem as the typical gallery contains few image sets per class, leading to correspondingly lower number of (high-dimensional) examples

per class that may not be enough to reliably train a machine learning model.

To avoid these problems, we extract a symmetric matrix feature $\mathbf{L} \in \mathbf{T}_{\mathbf{I}}\mathbb{S}_{+}^{Q}$ from each image $\boldsymbol{I}$, leading to more samples from each image set. Moreover, we limit the curse of dimensionality by compressing $\mathbf{L}$ into $\boldsymbol{a} = \text{comp}\left(\text{vec}\left(\mathbf{L}\right)\right) \in \mathbb{R}^{D}$ and using $\boldsymbol{a}$ as an atom in a dictionary corresponding to the image set. The steps for computing the image-level compressed features are summarized in Figure 5.1 and described below.

At the heart of our image-level descriptor is the use of Region Covariance Matrices (RCMs) [95]. This is justified by the ability of covariances to fuse various types of features and keep track of their statistics. In addition, the covariance of a set of samples is invariant to a rearrangement of these samples, giving RCMs more robustness to misalignment, a problem that face identification systems have to deal with even when automatic face alignment is applied, as the obtained detection and alignment may not be perfect.

To compute the covariance matrices, we first compute a feature image $\boldsymbol{\phi}(\boldsymbol{I})$ similar to [47, 95] which produces at each pixel the following $M = 43$ responses:

$$\boldsymbol{\phi}_{x,y}^{T} = \left[\; x, y, \boldsymbol{I}(x,y), |G_{0,0}(x,y)|, \quad \ldots, \quad |G_{4,7}(x,y)| \; \right]$$

where $G_{u,v}(x,y) = g_{u,v}(x,y) * \boldsymbol{I}(x,y)$ is the response of the image to the 2D Gabor wavelet $g_{u,v}(x,y)$ [47]:

$$\frac{k_v^2}{4\pi^2} e^{-\frac{k_v^2}{8\pi^2}\left(x^2+y^2\right)} \left(e^{ik_v(x\cos\theta_u + y\sin\theta_u)} - e^{-2\pi^2} \right)$$

where u is the orientation index, v is the scale index, $k_v = 1/\sqrt{2^{v-1}}$, and $\theta_u = \pi u/8$. To

balance the trade-off between robustness to misalignment and spatial encoding, we follow the tradition of breaking the image into $n_r \times n_c$ cells and computing a covariance matrix for each cell based on the pixel responses in that cell.

To avoid the curse of dimensionality in the extracted descriptor, we compress the M responses at each pixel in cell (r, c), prior to computing the cell-specific covariance matrix, by projecting the M-D response vector into a subspace of a lower dimensionality m using a cell-specific, $M \times m$ column-orthogonal projection matrix $\mathbf{U}_{r,c}$. Each matrix $\mathbf{U}_{r,c}$ is computed by performing PCA on the M-D response vectors at all pixels within the cell (r, c) from all the images in all gallery image sets. In our experiments, we set $m = 10$.

After compressing the responses, we calculate the $m \times m$ covariance matrix $\mathbf{\Sigma}_{r,c}$ from the m-D responses in cell (r, c). Next, we arrange the $n_r \times n_c$ covariances into the diagonal blocks of a $Q \times Q$ matrix $\mathbf{\Sigma}$, where $Q = n_r n_c m$. The matrix $\mathbf{\Sigma}$ can be easily shown to be SPD and so it lives in the non-Euclidean SPD manifold $\mathbb{S}_+^Q$. To measure the similarity in this non-Euclidean space, we endow $\mathbb{S}_+^Q$ with the LE Metric [6] which measures the distance between any pair of SPD matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ by first using the Log map: $\mathrm{Log} : \mathbb{S}_+^Q \to \mathbf{T_I}\mathbb{S}_+^Q$ to map them to the LE tangent space $\mathbf{T_I}\mathbb{S}_+^Q$ and then computing the Frobenius distance $\|\mathrm{Log}\,\mathbf{X}_2 - \mathrm{Log}\,\mathbf{X}_1\|_F$. If the Singular Value Decomposition (SVD) of an SPD matrix of dimensions $m \times m$ is $\mathbf{X} = \mathbf{U}\,\mathrm{diag}(s_1, ..., s_m)\mathbf{V}^T$, the Log map is defined as:

$$\mathrm{Log}\,\mathbf{X} = \mathbf{U}\,\mathrm{diag}(\log s_1, ..., \log s_m)\mathbf{V}^T \tag{5.1}$$

The LE tangent space $\mathbf{T_I}\mathbb{S}_+^Q$ is equivalent to the space of symmetric matrices $\mathbb{S}^Q$, which is a vector space. This allows us to apply familiar Euclidean machine learning algorithms to

SPD matrices once they are mapped to the LE tangent space. Accordingly, the final steps are (a) mapping the SPD matrix $\mathbf{\Sigma}$ to the LE tangent space by computing $\mathbf{L} = \operatorname{Log} \mathbf{\Sigma}$, (b) computing the uncompressed atom $\tilde{\boldsymbol{a}} = \operatorname{vec}(\mathbf{L}) \in \mathbb{R}^{Q^2}$, and then obtaining the compressed atom $\boldsymbol{a} = \operatorname{comp}(\tilde{\boldsymbol{a}}) \in \mathbb{R}^{D}$ where the operator comp() retains only the $D = n_r n_c m^2$ entries of $\tilde{\boldsymbol{a}}$ corresponding to the $n_r n_c$ diagonal blocks of $\mathbf{L}$ while discarding the rest (see the structure of $\boldsymbol{a}$ in Figure 5.1).

Arranging the cell covariances into the diagonal blocks of $\mathbf{\Sigma}$ and mapping $\mathbf{\Sigma}$ to the LE tangent space unnecessarily requires more memory and processing time. Instead, we apply the equivalent but more efficient process of separately mapping each cell covariance matrix $\mathbf{\Sigma}_{r,c}$ to the LE tangent space, which gives $\mathbf{L}_{r,c} = \operatorname{Log} \mathbf{\Sigma}_{r,c}$. In addition, we store only the $D = n_r n_c m^2$ nonzero entries of $\mathbf{L}$, which correspond to its diagonal blocks $\mathbf{L}_{r,c}$, into the compressed atom $\boldsymbol{a} \in \mathbb{R}^{D}$. All the remaining steps use the compressed atom $\boldsymbol{a}$ instead of the uncompressed, higher dimensional atom $\tilde{\boldsymbol{a}} \in \mathbb{R}^{Q^2}$.

The feature extraction step can be very efficiently implemented by making use of GPUs for performing convolutions and matrix multiplication. For each image, $n_r \times n_c$ small eigenvalue problems need to be computed for SPD matrices of size $m \times m$ in order to compute their matrix logarithms. Additional $n_r \times n_c$ eigenvalue problems of $M \times M$ matrices need to be solved for performing PCA during training but these are done only once for the complete gallery set rather than for each image.

5.1.2 Log-Euclidean Subspace Feature Learning (LE-SFL)

The goal of this step is to map the image descriptors from the LE tangent space $\mathbf{T_I}\mathbb{S}_+^Q$ into a lower-dimensional LE tangent space $\mathbf{T_I}\mathbb{S}_+^q$ in which they have a more discriminative subspace structure. In other words, we want the samples from one class to stay, in the new space, as far as possible from other-class subspaces while staying close to the same-class subspaces. In this new space, the sparse coding of a query sample $\boldsymbol{y}$ from class c over the dictionary $\mathbf{A}$ will more likely find that the subdictionary $\mathbf{A}_c$ provides better reconstruction of $\boldsymbol{y}$ compared to other subdictionaries. Consequently, the sparse coding will more likely associate $\boldsymbol{y}$ with its true class c.

Tangent Map Formulation: There are different ways to formulate the tangent map $\mathcal{W} : \mathbf{T_I}\mathbb{S}_+^Q \rightarrow \mathbf{T_I}\mathbb{S}_+^q$. One way is by the linear formulation given by:

$$\mathrm{vec}\,(\mathbf{L}') = \mathcal{W}_1(\mathbf{L}) = \mathbf{W}^T\,\mathrm{vec}\,(\mathbf{L}) \tag{5.2}$$

where $\mathbf{W} \in \mathbb{R}^{Q^2 \times q^2}$. To guarantee that $\mathbf{L}' \in \mathbb{S}^q$ for any $\mathbf{L} \in \mathbb{S}^Q$, the matrix $\mathbf{W}$ has to be constrained, such that it has $q(q+1)/2$ unique columns while the other $q(q-1)/2$ columns are permutations of other columns[1].

The second formulation $\mathcal{W}_2$ is a variation of $\mathcal{W}_1$ that avoids placing constraints on $\mathbf{W}$ by keeping only the unique $q(q+1)/2$ columns in $\mathbf{W}$ so that we just compute the (vectorized) lower triangular submatrix $\mathrm{tril}(\mathbf{L}') \in \mathbb{R}^{q(q+1)/2}$ instead of the complete matrix

[1]There are other algebraically equivalent ways to express the constraint on the columns of $\mathbf{W}$, all of them leading to the same measure of distance between symmetric matrices. Since we do not use the formulation $\mathcal{W}_1$ in this chapter, further elaboration on these ways is beyond the scope of this chapter.

$\mathbf{L}' \in \mathbb{R}^{q \times q}$:

$$\text{tril}(\mathbf{L}') = \mathcal{W}_2(\mathbf{L}) = \mathbf{W}^T \,\text{vec}\,(\mathbf{L}) \tag{5.3}$$

where $\mathbf{W} \in \mathbb{R}^{Q^2 \times q(q+1)/2}$. Since $\tilde{\boldsymbol{a}} = \text{vec}\,(\mathbf{L})$ has only D nonzero entries at known locations, the projection matrix $\mathbf{W}$ in both $\mathcal{W}_1$ and $\mathcal{W}_2$ needs only to contain the D rows corresponding to these nonzero entries. In this case, the dimensions of $\mathbf{W}$ in $\mathcal{W}_1$ can be reduced to $D \times q^2$ while in $\mathcal{W}_2$ it will be $D \times q(q+1)/2$. For simplicity, we use the second formulation, in which $\mathbf{W}$ is unconstrained and has dimensions $D \times q(q+1)/2$.

It is worth noting that a third formulation was used in [55, 134] which has the advantage of using much fewer parameters in the projection $\mathbf{W}$. However, the formulation is quadratic (nonlinear) in the projection parameters compared to linear formulations $\mathcal{W}_1$ and $\mathcal{W}_2$. Such a quadratic formulation is useful for applications in which the SPD descriptors are very high-dimensional such as the 400×400 image set covariance descriptor used by Wang et al. [122]. The SPD descriptors in this chapter have considerably fewer dimensions, and so we opt to use the simpler linear form $\mathcal{W}_2$. As we see later, our choice of a linear formulation for the embedding leads to an easier-to-solve optimization problem in which finding the globally optimal solution is straightforward and efficient.

Optimization Problem: Let $\mathbf{A}_c \in \mathbb{R}^{D \times N_c}$ be the dictionary containing all the N_c compressed atoms from all image sets associated with class c (after removing all identical atoms due to identical images):

$$\mathbf{A}_c = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \dots & \boldsymbol{a}_{N_c} \end{bmatrix}$$

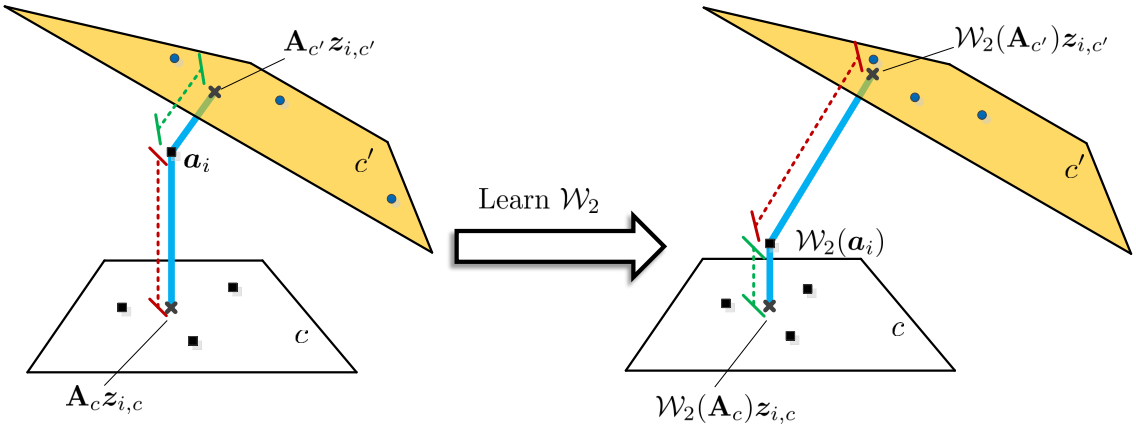


Figure 5.2: To improve the discriminative subspace arrangement of the data, the LE feature map $\mathcal{W}_2$ is learned such that it maximizes the distance between each atom $\boldsymbol{a}_i$ and its projection $\mathbf{A}_{c'}\boldsymbol{z}_{i,c}$ on every other-class dictionary $\mathbf{A}_{c'}$ while minimizing the distance between the sample and its projection $\mathbf{A}_c\boldsymbol{z}_{i,c}$ on the dictionary $\mathbf{A}_c$ of its own class c.

Furthermore, let N be the total number of atoms in the gallery, C be the number of classes, $c(i)$ be the class associated with atom $\boldsymbol{a}_i$, and let $\boldsymbol{z}_{i,c}$ be the dense representation of an atom $\boldsymbol{a}_i$ with respect to the dictionary $\mathbf{A}_c$ of a different class c:

$$\boldsymbol{z}_{i,c} = \underset{\boldsymbol{z}}{\operatorname{argmin}} \|\boldsymbol{a}_i - \mathbf{A}_c\boldsymbol{z}\|_2^2 + \lambda_1\|\boldsymbol{z}\|_2^2 \tag{5.4}$$

where we use $\lambda_1 = 0.001$. If we let $\mathbf{J}_c = \mathbf{A}_c^T\mathbf{A}_c + \lambda_1\mathbf{I}$, we obtain $\boldsymbol{z}_{i,c} = \mathbf{J}_c^{-1}\mathbf{A}_c^T\boldsymbol{a}_i$. The first goal the tangent map $\mathcal{W}_2$ should achieve is to maximize the distance between every atom $\boldsymbol{a}_i$, from a certain class $c(i)$, and its dense projection $\mathbf{A}_c\boldsymbol{z}_{i,c}$ on the dictionary of each

other class $c \neq c(i)$ (see Figure 5.2):

$$\frac{1}{C}\sum_{c=1}^{C}\sum_{i,c(i)\neq c}\frac{1}{N_{c(i)}(C-1)}\|\mathbf{W}^T(\boldsymbol{a}_i-\mathbf{A}_c\boldsymbol{z}_{i,c})\|_2^2 \tag{5.5}$$

$$= \frac{1}{C}\sum_{c=1}^{C}\sum_{i,c(i)\neq c}\frac{1}{N_{c(i)}(C-1)} \operatorname{tr}\mathbf{W}^T\left(\boldsymbol{a}_i-\mathbf{A}_c\boldsymbol{z}_{i,c}\right)\left(\boldsymbol{a}_i-\mathbf{A}_c\boldsymbol{z}_{i,c}\right)^T\mathbf{W} \tag{5.6}$$

$$= \operatorname{tr}\mathbf{W}^T\mathbf{S}_1\mathbf{W} \tag{5.7}$$

where

$$\mathbf{S}_1 = \tfrac{1}{C}\textstyle\sum_{c=1}^{C}\sum_{i,c(i)\neq c}\displaystyle\frac{1}{N_{c(i)}(C-1)} (\boldsymbol{a}_i-\mathbf{A}_c\boldsymbol{z}_{i,c})(\boldsymbol{a}_i-\mathbf{A}_c\boldsymbol{z}_{i,c})^T \tag{5.8}$$

In the above derivation (Eq. (5.6)), we use the matrix identity

$$\|\mathbf{B}\|_F = \mathbf{B}\mathbf{B}^T = \mathbf{B}^T\mathbf{B} \tag{5.9}$$

The reason we use dense, l_2-regularized representations is that it has a closed form solution that is more efficient to evaluate. Moreover (and more importantly), we want to maximize the distance between each atom $\boldsymbol{a}_i$ and the span of as many as possible of those different-class atoms that may contribute to reconstructing $\boldsymbol{a}_i$. This makes the dense representation a more appropriate choice.

Before describing the other goal, we redefine the dense representation $\boldsymbol{z}_{i,c}$ for the

case in which we project $\boldsymbol{a}_i$ on the dictionary of its own class (i.e. $c = c(i)$):

$$\boldsymbol{z}_{i,c} = \underset{\boldsymbol{z}}{\operatorname{argmin}} \|\boldsymbol{a}_i - \mathbf{A}_c \boldsymbol{z}\|_2^2 + \lambda_1 \|\boldsymbol{z}\|_2^2, \text{s.t. } z^{(i)} = 0. \tag{5.10}$$

The only difference between (5.4) and (5.10) is the constraint $z^{(i)} = 0$ which excludes any solution in which $\boldsymbol{a}_i$ contributes to its own representation. If we let $\boldsymbol{u}_{i,c} = \mathbf{J}_c^{-1} \mathbf{A}_c^T \boldsymbol{a}_i$, and $w_i = \boldsymbol{u}_{i,c}^{(i)} / \mathbf{J}_c^{-1(i,i)}$, we obtain

$$\boldsymbol{z}_{i,c} = \boldsymbol{u}_{i,c} - w_i \text{col}_i(\mathbf{J}_c^{-1}) \tag{5.11}$$

The other goal the tangent map has to meet is minimizing the distance between every atom $\boldsymbol{a}_i$ from a certain class c to its dense projection $\mathbf{A}_c \boldsymbol{z}_{i,c}$ on the dictionary of its own class:

$$\frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i,c(i)=c} \|\mathbf{W}^T (\boldsymbol{a}_i - \mathbf{A}_c \boldsymbol{z}_{i,c})\|_2^2 \tag{5.12}$$

$$= \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i,c(i)=c} \operatorname{tr} \mathbf{W}^T (\boldsymbol{a}_i - \mathbf{A}_c \boldsymbol{z}_{i,c})(\boldsymbol{a}_i - \mathbf{A}_c \boldsymbol{z}_{i,c})^T \mathbf{W} \tag{5.13}$$

$$= \operatorname{tr} \mathbf{W}^T \mathbf{S}_2 \mathbf{W} \tag{5.14}$$

where

$$\mathbf{S}_2 = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i,c(i)=c} (\boldsymbol{a}_i - \mathbf{A}_c \boldsymbol{z}_{i,c})(\boldsymbol{a}_i - \mathbf{A}_c \boldsymbol{z}_{i,c})^T \tag{5.15}$$

In addition, we add a regularization term $\|\mathbf{W}\|_F^2 = \operatorname{tr} \mathbf{W}^T\mathbf{W}$ to the quantity to be minimized. We then combine all goals into one criterion by maximizing the following ratio of quadratic forms:

$$\max_{\mathbf{W}} \frac{\operatorname{tr} \mathbf{W}^T\mathbf{S}_1\mathbf{W}}{\operatorname{tr} \mathbf{W}^T(\mathbf{S}_2 + \mathbf{I})\mathbf{W}} \tag{5.16}$$

The optimal solution to this problem is obtained by finding the $q(q+1)/2$ generalized eigenvectors with the largest eigenvalues of the following generalized eigenvalue problem:

$$\mathbf{S}_1\boldsymbol{w}_k = \lambda_k(\mathbf{S}_2 + \mathbf{I})\boldsymbol{w}_k \tag{5.17}$$

After finding $\mathbf{W}$, we use it to embed the dictionaries (i.e. gallery matrices) of all classes. If we assume all the C classes have the same number of images $N_c = N/C$, the computational complexity of subspace feature learning is:

$$O\left(D^3 + C \times \left(CN_c^3 + DN_c^2 + CD^2N_c\right)\right) \tag{5.18}$$

where it takes $O(D^3)$ for the solution of the $D \times D$ generalized eigenvalue problem in (5.16), $O(DN_c^2 + N_c^3)$ for computing $\mathbf{J}_c$ and inverting it for one class, $O(CN_c^3)$ for computing the representations of same-class samples and other-class samples with respect to the dictionary (i.e. gallery matrix) $\mathbf{A}_c$ of one class, and $O(CD^2N_c)$ for computing the contribution of one class to the two scatter matrices $\mathbf{S}_1$ and $\mathbf{S}_2$.

5.1.3 Coding and Classification

Given a probe image set $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N_y}]$, the method extracts the LE dictionary from the set $\mathbf{Y}$ as described in Section 5.1.1; then uses the tangent map $\mathbf{W}$ to project each atom in $\mathbf{Y}$'s dictionary to the LE feature space. Subsequently, we apply a representation-based classification algorithm like SRC or CRC to compute the label for $\mathbf{Y}$. More specifically, we solve for the sparse representation vector $\boldsymbol{x} \in \mathbb{R}^N$ corresponding to the mean $\bar{\boldsymbol{y}}$ of the embedded feature vectors (i.e. LE Frechet mean [6]):

$$\boldsymbol{x} = \underset{\boldsymbol{x} \in \mathbb{R}^N}{\operatorname{argmin}} \frac{1}{2} \|\bar{\boldsymbol{y}} - \mathbf{A}\boldsymbol{x}\|^2 + \lambda \|\boldsymbol{x}\|_1 \tag{5.19}$$

where $\mathbf{A}$ is the dictionary containing all the embedded LE atoms from all classes. A variant of the above formulation is the CRC scheme which uses the l_2-norm for regularization instead of l_1:

$$\boldsymbol{x} = \underset{\boldsymbol{x} \in \mathbb{R}^N}{\operatorname{argmin}} \frac{1}{2} \|\bar{\boldsymbol{y}} - \mathbf{A}\boldsymbol{x}\|^2 + \frac{\lambda}{2} \|\boldsymbol{x}\|_2 \tag{5.20}$$

Given the (sparse) representation $\bar{\boldsymbol{x}}$, we can find the class contributing the most to the representation, and with which $\bar{\boldsymbol{y}}$ should be associated, using the minimum residual rule of [133]. If we let δ_c be the $N \times N$ diagonal matrix with all zeros except at the N_c diagonal entries corresponding to the atoms of class c, the residual $r_c(\boldsymbol{y}; \bar{\boldsymbol{x}})$ corresponding to class c is given by:

$$r_c(\boldsymbol{y}; \bar{\boldsymbol{x}}) = \|\boldsymbol{y} - \mathbf{A}\delta_c \bar{\boldsymbol{x}}\|^2 \tag{5.21}$$

The class for which r_c is minimum is chosen as the label for the probe set.

5.1.4 More General Discriminative Subspace Feature Learning

While Section 5.1.2 described Subspace Feature Learning (SFL) for mapping between LE tangent spaces, the same algorithm can be used to learn an embedding $\mathcal{F} : \mathcal{A} \rightarrow \mathbb{R}^d$:

$$\mathcal{F}(\boldsymbol{a}) = \mathbf{W}^T \boldsymbol{a} \tag{5.22}$$

where $\mathcal{A}$ is any vector space of an appropriate dimensionality and $\mathbb{R}^d$ is a lower-dimensional vector space. Note that the above definition does not restrict $\mathcal{A}$ or $\mathbb{R}^d$ to LE tangent spaces. For example, $\mathcal{A}$ may be the space of raw intensity images of a particular width and height. Alternatively, $\mathcal{A}$ might be the deep features produced by a Convolutional Neural Network (CNN)-based embedding. As before, the embedding parameters $\mathbf{W}$ is obtained by solving the generalized eigenvalue problem in (5.17) resulting from the optimization problem in (5.16).

5.2 Dictionary-Based Subspace Feature Learning (DBSFL)

The algorithm presented in Section 5.1.2 uses the gallery class matrices $\{\mathbf{A}_c \in \mathbb{R}^{D \times N_c}\}_{c=1}^{C}$ to learn the embedding during training and to perform coding and classification during testing. There are two potential issues with this approach:

1. Different classes can have different numbers of gallery image sets and/or different image sets may have different numbers of atoms. The number N_c of atoms in a given class matrix $\mathbf{A}_c$ may not be the same as other classes. Such imbalance can negatively affect performance during both training and testing. More specifically,

a class with more atoms will contribute more terms to the objective functions in Eq. (5.5) and (5.12), which can bias the learning of the embedding. During testing, a class with more atoms would correspond to a subspace of a higher dimensionality which can bias the coding (and the subsequent classification) to prefer these classes that have more atoms. This becomes more of a problem when N_c is relatively small for each class c.

2. The overall number N_c of samples from each class can be too large. This can adversely affect the running time of the coding of test data during testing.

To mitigate these potential issues, we propose Dictionary-Based Subspace Feature Learning (DBSFL), a variant of SFL that uses dictionary learning to handle imbalanced gallery sets and/or classes with too many gallery samples. This speeds up coding at test time and can improve the quality of the embedding with unbalanced training sets. The rest of this section describes the steps of DBSFL.

5.2.1 Dictionary Learning

The first step in DBSFL is to learn a dictionary $\mathbf{D}_c \in \mathbb{R}^{D \times k}$ from the class gallery matrix $\mathbf{A}_c \in \mathbb{R}^{D \times N_c}$ for each class c, where k is the number of atoms in the learned dictionary $\mathbf{D}_c$. These dictionaries $\{\mathbf{D}_c\}_{c=1}^{C}$ are then used instead of the class gallery matrices $\{\mathbf{A}_c\}_{c=1}^{C}$ in the formulation of the objective function to be optimized by the embedding parameter $\mathbf{W}$.

To learn the dictionary $\mathbf{D}_c$ of class c, we solve the following optimization problem:

$$\underset{\mathbf{D}_c, \mathbf{Z}_c}{\operatorname{argmin}} \quad \frac{1}{N_c}\|\mathbf{A}_c - \mathbf{D}_c\mathbf{Z}_c\|_F^2 + \lambda_1\|\mathbf{Z}_c\|_1, \tag{5.23}$$

$$\text{subject to} \quad \mathbf{D}_c^T\mathbf{D}_c = \mathbf{I}_{N_c} \tag{5.24}$$

where we obtain as a by-product the matrix $\mathbf{Z}_c \in \mathbb{R}^{k \times N_c}$ containing the sparse representations of the atoms in the gallery matrix $\mathbf{A}_c$ with respect to the dictionary $\mathbf{D}_c$. These representations are utilized while formulating the objective function as we later illustrate.

While several algorithms have been proposed for dictionary learning [3, 33, 65, 69, 86, 92], we use the Online Dictionary Learning (ODL) algorithm of Mairal et al. [86] as it scales well with bigger gallery image sets, which are common in video datasets. ODL works by iteratively alternating between (a) updating the sparse code vector of one of the gallery matrix atoms (i.e. one sample from $\mathbf{A}_c$) and (b) updating the dictionary $\mathbf{D}_c$.

5.2.2 Objective Function

Similar to SFL, DBSFL seeks an embedding $\mathcal{F}$ that maximizes the distance between the span of every class c and the span of every other class $s \neq c$ for $c, s \in \{1, \ldots, C\}$. However, DBSFL uses the estimated dictionaries $\{\mathbf{D}_c \in \mathbb{R}^{D \times k}\}_{c=1}^{C}$ to characterize the spans of different classes rather than the original gallery matrices $\{\mathbf{A}_c \in \mathbb{R}^{D \times N_c}\}_{c=1}^{C}$ (which SFL uses). Since all the class dictionaries have the same number of atoms k, DBSFL avoids the potential imbalance in the sizes of the class gallery matrices $\{\mathbf{A}_c \in \mathbb{R}^{D \times N_c}\}_{c=1}^{C}$ that SFL uses for formulating its objective function in Eq.(5.5, 5.16). To formulate a measure of the distance between the spans of different classes, we first obtain the sparse

representation $\mathbf{Z}_{c,s} \in \mathbb{R}^{k \times k}$ of the dictionary atoms of $\mathbf{D}_c$ with respect to the dictionary $\mathbf{D}_s$ of class s for every $c, s \in \{1, \ldots, C\}, s \neq c$ by solving the following sparse representation problem

$$\underset{\mathbf{Z}_{c,s}}{\operatorname{argmin}} \|\mathbf{D}_c - \mathbf{D}_s \mathbf{Z}_{c,s}\|_F^2 + \lambda_1 \|\mathbf{Z}_{c,s}\|_1 \tag{5.25}$$

The first goal we need the embedding $\mathcal{F}$ to achieve is to maximize the distance between the atoms of every dictionary $\mathcal{F}(\mathbf{D}_c)$ and their projections $\mathcal{F}\left(\mathbf{D}_s \mathbf{Z}_{c,s}\right)$ on the spans of other classes, which is characterized by the following function:

$$\begin{aligned} & \frac{1}{C} \sum_{c=1}^{C} \frac{1}{k(C-1)} \sum_{s, s \neq c}^{C} \|\mathbf{W}^T (\mathbf{D}_c - \mathbf{D}_s \mathbf{Z}_{c,s})\|_F^2 && (5.26) \\ = \; & \frac{1}{kC(C-1)} \sum_{c=1}^{C} \sum_{s, s \neq c}^{C} \\ & \qquad \operatorname{tr} \mathbf{W}^T (\mathbf{D}_c - \mathbf{D}_s \mathbf{Z}_{c,s})(\mathbf{D}_c - \mathbf{D}_s \mathbf{Z}_{c,s})^T \mathbf{W} && (5.27) \\ = \; & \qquad \operatorname{tr} \mathbf{W}^T \mathbf{S}_1 \mathbf{W} && (5.28) \end{aligned}$$

where

$$\begin{aligned} \mathbf{S}_1 \; = \; & \frac{1}{kC(C-1)} \operatorname{tr} \sum_{c=1}^{C} \sum_{s, s \neq c}^{C} \\ & \qquad \operatorname{tr} (\mathbf{D}_c - \mathbf{D}_s \mathbf{Z}_{c,s})(\mathbf{D}_c - \mathbf{D}_s \mathbf{Z}_{c,s})^T && (5.29) \end{aligned}$$

In addition, the desired embedding $\mathcal{F}$ should minimize the distance between the samples $\mathbf{A}_c$ and the spans of their classes. Using the learned dictionaries to characterize

the spans of the different classes, we require $\mathbf{W}$ to minimize

$$\frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\|\mathbf{W}^T(\mathbf{A}_c-\mathbf{D}_c\mathbf{Z}_c)\|_F^2 \tag{5.30}$$

$$= \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\operatorname{tr}\mathbf{W}^T(\mathbf{A}_c-\mathbf{D}_c\mathbf{Z}_c)(\mathbf{A}_c-\mathbf{D}_c\mathbf{Z}_c)^T\mathbf{W} \tag{5.31}$$

$$= \operatorname{tr}\mathbf{W}^T\mathbf{S}_2\mathbf{W} \tag{5.32}$$

where $\mathbf{Z}_c$ is the sparse representation of the atoms in $\mathbf{A}_c$ with respect to $\mathbf{D}_c$, obtained as a by-product of solving the dictionary learning optimization problem in Eq. (5.23) and

$$\mathbf{S}_2 = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\operatorname{tr}(\mathbf{A}_c-\mathbf{D}_c\mathbf{Z}_c)(\mathbf{A}_c-\mathbf{D}_c\mathbf{Z}_c)^T \tag{5.33}$$

Similar to SFL, we then combine all goals into a ratio to be maximized:

$$\max_{\mathbf{W}}\frac{\operatorname{tr}\mathbf{W}^T\mathbf{S}_1\mathbf{W}}{\operatorname{tr}\mathbf{W}^T(\mathbf{S}_2+\mathbf{I})\mathbf{W}} \tag{5.34}$$

This results in a generalized eigenvalue problem similar to Eq. (5.17) and the d columns of the optimal solution $\mathbf{W}$ are the d generalized eigenvectors of Eq. (5.17) corresponding to the d largest generalized eigenvalues. We then use $\mathbf{W}$ to embed the learned dictionaries.

5.2.3 Coding and Classification

Given a probe image set $\mathbf{Y}=[\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{N_y}]$, we first compute its embedding $\mathbf{Y}'=\mathbf{W}^T\mathbf{Y}$ then we apply a representation-based classification algorithm like SRC or CRC to compute

the label for the mean of $\mathbf{Y}'$ as in Section 5.1.3. Unlike SFL, DBSFL uses the embedded dictionaries instead of the embedded gallery matrices to perform coding and residual-based classification. Since the number of atoms per dictionary k is chosen to be lower than the number of atoms in the class gallery matrix, the coding and residual-based classification in DBSFL are faster than in SFL.

5.3 Experimental Evaluation

We have conducted extensive experiments involving multiple methods, four challenging datasets and different features to evaluate the performance of the proposed algorithms. We describe these experiments below.

5.3.1 Shallow Features Experiments

The first set of experiments are based on raw intensity features. We compare the two Log-Euclidean variants of our proposed approach: Log-Euclidean Subspace Feature Learning (LE-SFL) and its dictionary-based extension (LE-DBSFL). In addition, we evaluate the performance of different representation-based algorithms (mean-sequence SRC and CRC) when they are combined with LE-SFL and LE-DBSFL. This results in four schemes: LE-SFL-SRC, LE-DBSFL-SRC, LE-SFL-CRC, and LE-DBSFL-CRC. To understand the contribution to performance made by the different components of our classifier, we also compare with two variants of our classifier: one without LE features but with subspace feature learning applied to intensity features (SFL-SRC), and another with the LE features but without the subspace feature learning (LE-SRC).

We also include in the comparison several existing methods for image-set classification. The competing methods include Affine Hull-based Image Set Distance (AHISD) [17], its convex variant (CHISD) [17], Sparse-Approximated Nearest Points (SANP) [53], Dictionary-based Face Recognition from Videos (DFRV) [24], Mean Sequence Sparse Representation-based Classification (MS-SRC) [93], a variation of MS-SRC that uses CRC instead of SRC [145] for classifying the mean of the sequence (MS-CRC), Set to Set Distance Metric Learning (SSDML) [149], Deep Reconstruction Models (DRM) [50], Projection Metric Learning (PML) [54], and Log-Euclidean Metric Learning (LEML) [55].

For existing methods, we have used the source code provided by the original authors and set the parameters according to the recommendations made in their respective papers. Exceptions to this are MS-CRC and MS-SRC which we have implemented ourselves. To guarantee a fair comparison, the same features and dataset splits were used to compare all the methods. We made an exception for the DRM approach where we report the performance using the 1475-D LBP features extracted from the intensity features used with the rest of the methods. The reason for this exception is that the original paper of DRM [50] and its publicly available source-code included the extraction of LBP features as one of the preprocessing steps of DRM. For LEML, we use the LE-GRCM descriptor proposed in this work and used with our LE-SFL-SRC method for a fair comparison between the two LE-based methods. For PML, we have modified the method to deal with the situation in which the number of images n_s in a given image set s is lower than the dimensionality d of the subspace PML computes from each image set. In that case, we synthesize additional images by small random translations and rotations of the original n_s images so that s has

$2d$ images. Since PML requires the gallery to have at least two image sets for each class whereas MobFaces-I provides a single gallery image set per class, we randomly split each gallery set in MobFaces-I into two subsets of nearly equal sizes (the difference in size is at most one).

Parameter Settings: We use the following parameters in our proposed method. For GRCM feature extraction, we resize each input image to $w = 120$ and $h = 144$. We then divide each image into a $n_r = 6 \times n_c = 6$ grid of non-overlapping cells for calculating the RCMs. As stated earlier, we use $m = 10$ as the dimension for the compressed per-pixel responses. Finally, we set the symmetric dimension of the lower-dimensional LE tangent space to $q = 28$ which corresponds to an LE tangent map $\mathbf{W}$ with $q(q+1)/2 = 406$ columns. It is worth noting that smaller grids (i.e. smaller n_r and n_c) lead to inferior recognition performance. Grids larger than $n_r = 6 \times n_c = 6$ could possibly lead to better performance, although this will be at the expense of increasing the memory footprint of the algorithm. For DBSFL, we set $k = 50$ atoms per class-specific dictionary for MobFaces-I and $k = 75$ for MobFaces-II since MobFaces-I has smaller gallery size for each class. Since the gallery sets of YTC and YTF contains enough many samples for each class, we use $k = 100$ for both datasets. For the representation-based classifiers (SRC, CRC, and their extensions), we set the regularization parameter λ equal to 0.01.

In this set of experiments, we use the YTC, YTF, and MobFaces datasets. Figure 4.3 shows examples from each dataset. A brief description of the datasets YTC, YTF, and MobFaces and the corresponding experimental protocols is covered in Sections 4.5.1, 4.5.2, and 4.5.3, respectively.

Table 5.1: The multi-fold sample mean and standard deviation of the recognition rates obtained with the compared methods on YTC and YTF. We have highlighted in bold the rates of the top two performing methods for each dataset. Although YTC and YTF have similar challenges, the rates obtained for YTC are higher because the test protocol for YTC guarantees that for each test video clip there is a corresponding gallery video clip such that both are segments from the same parent YouTube video.

Methods	YTC	YTF
AHISD (CVPR, 2010)	57.27 ± 3.44	17.18 ± 8.93
CHISD (CVPR, 2010)	64.79 ± 1.72	32.99 ± 7.97
SANP (CVPR, 2011)	66.99 ± 0.69	31.62 ± 8.56
DFRV (ECCV, 2012)	66.70 ± 1.52	36.77 ± 10.19
MS-CRC (ICCV, 2011)	66.88 ± 2.21	43.64 ± 8.27
MS-SRC (CVPR, 2013)	74.68 ± 1.96	45.02 ± 5.82
SSDML (ICCV, 2013)	69.22 ± 1.64	34.02 ± 10.03
DRM (CVPR, 2014)	70.35 ± 2.52	43.99 ± 5.23
PML (CVPR, 2015)	68.55 ± 1.76	40.21 ± 11.98
LEML (ICML, 2015)	73.26 ± 1.50	48.45 ± 5.66
SFL-SRC (ours)	75.71 ± 1.57	45.36 ± 3.45
LE-SRC (ours)	75.11 ± 1.49	49.83 ± 7.51
LE-SFL-SRC (ours)	76.28 ± 2.22	53.26 ± 8.10
LE-DBSFL-SRC (ours)	$\mathbf{76.74 \pm 2.12}$	$\mathbf{58.42 \pm 9.78}$
LE-SFL-CRC (ours)	72.84 ± 2.55	46.05 ± 6.73
LE-DBSFL-CRC (ours)	$\mathbf{76.91 \pm 1.89}$	$\mathbf{61.86 \pm 8.18}$

5.3.1.1 Results

Table 5.1 shows the mean and standard-deviation of the recognition rates of the compared methods for the YTC and YTF datasets while Tables 5.2 and 5.3 show the recognition rates for the six different evaluation scenarios for the MobFaces dataset. The tables clearly show the superiority of the proposed methods (LE-SFL-SRC, LE-DBSFL-SRC, LE-DBSFL-CRC) in comparison with other methods.

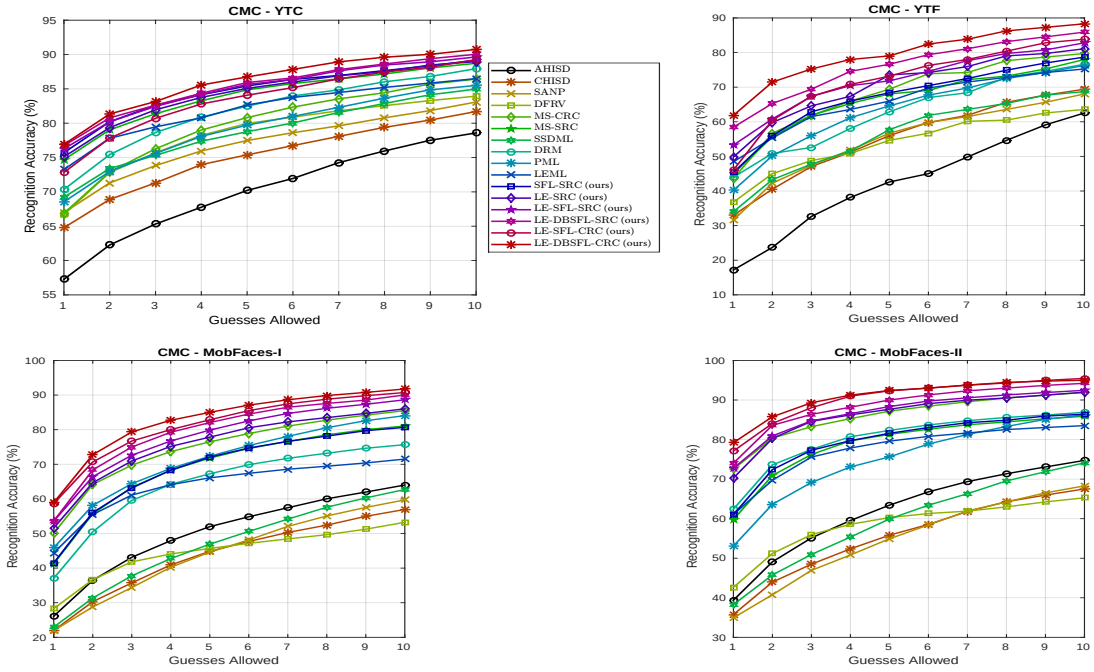
Table 5.2: The recognition rates obtained on the MobFaces dataset the MobFaces-I protocol. The setting $(1 \rightarrow \{2, 3\})$ involves training on session 1 (i.e. the lit session) and testing on sessions 2 and 3 (i.e. the unlit and day-lit sessions). The other two settings are defined in a similar manner. The 'avg' column contains the average of the rates obtained under the three settings to its left. Since each session has a different number of test video clips, the average column weighs the rate of each setting by its number of test sets. We have highlighted in bold the rates of the top two performing methods for each setting.

	MobFaces-I			
Methods	$\{2, 3\} \rightarrow 1$	$\{1, 3\} \rightarrow 2$	$\{1, 2\} \rightarrow 3$	avg
AHISD (CVPR, 2010)	15.00	31.14	29.30	26.12
CHISD (CVPR, 2010)	10.61	26.57	25.73	21.96
SANP (CVPR, 2011)	9.34	27.09	26.15	21.96
DFRV (ECCV, 2012)	19.39	32.29	30.87	28.30
MS-CRC (ICCV, 2011)	48.20	51.30	50.24	50.09
MS-SRC (CVPR, 2013)	32.40	46.56	42.49	41.29
SSDML (ICCV, 2013)	10.53	28.89	26.15	22.95
DRM-LBP (CVPR, 2014)	23.46	32.41	36.38	31.41
DRM-GRAY	33.28	38.94	37.95	37.06
PML (CVPR, 2015)	51.16	45.98	41.77	45.88
LEML (ICML, 2015)	42.70	45.93	44.07	44.39
SFL-SRC (ours)	32.88	46.97	42.25	41.48
LE-SRC (ours)	47.01	52.63	54.00	51.60
LE-SFL-SRC (ours)	48.20	56.21	54.90	53.58
LE-DBSFL-SRC (ours)	48.44	56.21	54.72	53.58
LE-SFL-CRC (ours)	**57.06**	**58.69**	**59.81**	**58.65**
LE-DBSFL-CRC (ours)	**53.71**	**59.79**	**61.99**	**58.93**

Table 5.3: The recognition rates obtained on the MobFaces dataset under the MobFaces-II protocol. The setting $(\{2,3\} \rightarrow 1)$ involves training on sessions 2 and 3 (i.e. the unlit and day-lit sessions) while testing on session 1 (i.e. the lit session). The other two settings are defined in a similar manner. The 'avg' column contains the average of the rates obtained under the three settings to its left. Since each session has a different number of test video clips, the average column weighs the rate of each setting by its number of test sets. We have highlighted in bold the rates of the top two performing methods for each setting.

	MobFaces-II			
Methods	$\{2,3\} \rightarrow 1$	$\{1,3\} \rightarrow 2$	$\{1,2\} \rightarrow 3$	avg
AHISD (CVPR, 2010)	24.41	51.28	52.85	39.39
CHISD (CVPR, 2010)	23.29	44.97	47.60	35.76
SANP (CVPR, 2011)	20.38	48.89	45.95	34.94
DFRV (ECCV, 2012)	32.11	50.60	52.40	42.62
MS-CRC (ICCV, 2011)	69.01	73.59	77.18	72.52
MS-SRC (CVPR, 2013)	43.29	71.89	75.53	59.79
SSDML (ICCV, 2013)	21.31	50.09	54.95	38.27
DRM-LBP (CVPR, 2014)	38.97	62.86	65.77	52.72
DRM-GRAY	53.62	70.53	69.37	62.42
PML (CVPR, 2015)	45.92	56.56	61.41	53.06
LEML (ICML, 2015)	49.39	66.95	74.62	61.09
SFL-SRC (ours)	44.98	72.40	76.58	61.00
LE-SRC (ours)	59.25	73.59	84.98	70.28
LE-SFL-SRC (ours)	62.72	75.64	86.19	72.74
LE-DBSFL-SRC (ours)	64.23	76.15	**87.84**	74.03
LE-SFL-CRC (ours)	**71.55**	**76.49**	86.48	**77.09**
LE-DBSFL-CRC (ours)	**73.24**	**78.71**	**89.19**	**79.21**

Figure 5.3: The mean Cumulative Matching Characteristic (CMC) curves for YTC (top-left), YTF (top-right), MobFaces-I (bottom-left), and MobFaces-II (bottom-right). DBSFL-CRC achieves the highest CMC curve on all the benchmarks.



The improvement in performance by SFL-SRC over MS-SRC is less significant except on YTC. This is because subspace feature learning does not help much with intensity features, which inherently have a subspace structure. Accordingly, the only significant advantage SFL-SRC provides over MS-SRC is the reduction of dimensionality without loss in identification accuracy. On the other hand, the results show that the improvement due to subspace feature learning is significant when we compare LE-SRC with LE-SFL-SRC. Despite being more robust, the LE-GRCM features have nonlinear dependence on raw intensities, and so they do not preserve the sparse linear dependencies between samples under the raw representation. To address this, the subspace feature learning algorithms (SFL and DBSFL) not only reduces the dimensionality of the LE features but also improves their discriminative subspace structure which in turn boosts the performance of SRC. It is worth noting that although LE-SFL-SRC outperforms MS-SRC and LE-SRC, LE-SFL-SRC uses only 406 features per atom which is fewer than the $30 \times 36 = 1080$ features used by MS-SRC and the $D = 3600$ features used by LE-SRC.

SFL vs DBSFL: The classification performance of the dictionary-based variant LE-DBSFL-SRC is at least as accurate as LE-SFL-SRC (and sometimes significantly better as in YTF) while using fewer atoms at test time (see Table 5.4) and taking, consequently, shorter classification time (see Table 5.7). However, the robustness of DBSFL against gallery class size imbalance significantly results in more significant accuracy improvement in the case of CRC. In particular, DBSFL improves the accuracy of LE-DBSFL-CRC over LE-SFL-CRC, especially on YTC and YTF where the gallery imbalance is significant. It is worth noting that the lack of dictionary balancing makes LE-SFL-CRC quite inferior, on YTC and YTF, to the SRC variants LE-SFL-SRC and LE-DBSFL-SRC, whereas DBSFL

makes LE-DBSFL-CRC the best performing on YTC and YTF. On MobFaces-I and II, LE-DBSFL-CRC improves the performance of LE-SFL-CRC but since the gallery in this dataset is relatively balanced, LE-SFL-CRC already does relatively well. This shows that the proposed dictionary-based variant allows the faster (but more imbalance-sensitive) representation-based classification approach based on CRC to be the best performing method in terms of both accuracy and efficiency in our experiments (see the test running times at Table 5.7).

Figure 5.3 shows the mean Cumulative Matching Characteristic (CMC) curves for the different methods on YTC, YTF, MobFaces-I and MobFaces-II. A CMC curve plots for each $r \in \{1, 2, 3, \dots\}$ the recognition rate or the fraction of the samples that are correctly classified if we consider for each sample the top r identity guesses predicted by the method for that sample rather than just the top guess. The curves in the Figure show that methods based on SFL and DBSFL still achieve the highest recognition rates as we consider more identity predictions.

Table 5.4: The average number of atoms used by SRC/CRC in LE-SFL-SRC/CRC and LE-DBSFL-SRC/CRC on each dataset.

Datasets	LE-SFL-SRC	LE-DBSFL-SRC
YTC	21398	4700
YTF	27011	4500
MobFaces-I	10526	2500
MobFaces-II	21052	3750
IJB-B	3349	923

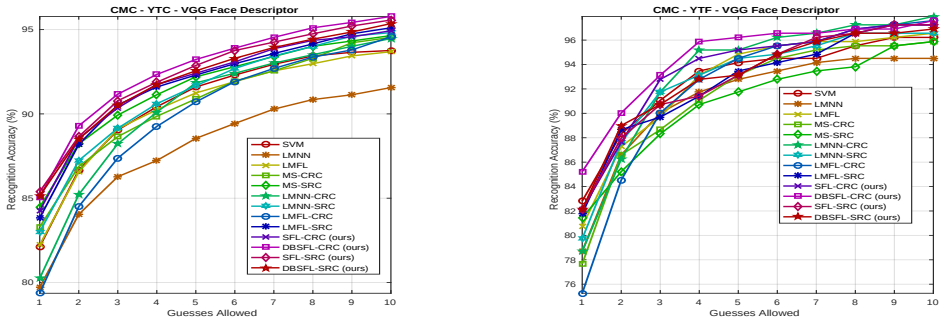
Table 5.5: The multi-fold sample mean and standard deviation of the recognition rates obtained with our methods using the VGG deep face descriptor [97]. We have highlighted in bold the rates of the top two performing methods for each dataset. As expected, the use of deep features leads to significant performance improvement in both datasets.

Methods	YTC	YTF
SVM	82.13 ± 1.96	$\mathbf{82.82 \pm 7.16}$
LMNN (JMLR, 2009)	79.72 ± 2.24	78.69 ± 9.31
LMFL (BMVC, 2015)	82.23 ± 1.57	80.76 ± 5.97
MS-CRC (ICCV, 2011)	83.30 ± 1.77	77.66 ± 5.57
MS-SRC (CVPR, 2013)	84.47 ± 1.59	81.44 ± 6.43
LMNN-CRC	80.25 ± 1.87	78.69 ± 7.52
LMNN-SRC	83.01 ± 1.32	79.73 ± 4.75
LMFL-CRC	79.36 ± 2.12	75.26 ± 4.81
LMFL-SRC	83.83 ± 1.33	81.79 ± 5.60
SFL-CRC (ours)	84.26 ± 1.57	82.13 ± 4.36
DBSFL-CRC (ours)	85.07 ± 1.89	$\mathbf{85.22 \pm 5.50}$
SFL-SRC (ours)	$\mathbf{85.39 \pm 1.53}$	82.13 ± 3.27
DBSFL-SRC (ours)	$\mathbf{85.14 \pm 1.48}$	82.13 ± 4.09

5.3.2 VGG Deep Face Descriptor Experiments

We also ran experiments over YTC and YTF using the 4096-dimensional Visual Geometry Group (VGG) deep face descriptor. In these experiments, we compare methods based on SFL, DBSFL, dimensionality reduction based on Large-Margin Metric Learning (LMNN) [116, 127] and the online iterative version of LMNN proposed in [97] (which we refer to as Large-Margin Feature Learning (LMFL)). Like SFL and DBSFL, we use LMNN and LMFL as means of discriminative dimensionality reduction prior to applying MS-SRC and MS-CRC. In other words, we present results for X-SRC and X-CRC for X in $\{$LMNN, LMFL, SFL, DBSFL$\}$. In the four methods, we reduce dimensions from 4096 to 1024. Moreover, we also present results for applying Nearest-Neighbor (NN) classification after obtaining the LMNN and LMFL features. Nearest Neighbor (NN) classification over a query image set works by applying NN over each sample in the query image set followed

Figure 5.4: The mean Cumulative Matching Characteristic (CMC) curves for the YTC (left) and YTF (right) datasets, based on the VGG deep face descriptor of Parkhi et al. [97]. DBSFL-CRC achieves the highest CMC curve on both datasets (only up to 7 guesses for YTF).



by majority voting to obtain an overall classification for the image set. The table also shows the performance of MS-SRC and MS-CRC without prior dimensionality reduction in addition to set-based linear Support Vector Machine (SVM) [18] which trains a one-versus-all SVM for each class and classifies an image set by classifying individual samples and using majority voting to determine an overall set label. Cross-validation is used to tune the hyper-parameters of SVM during training.

We use the same folds of YTC and YTF as in Section 5.3.1 and we extract and normalize the VGG deep face descriptor [97] from each aligned facial detection after resizing it to 227×227 to match the VGG CNN input image size. As in [97], the VGG descriptor is taken as the pre-activation (i.e. pre-ReLU) output of the fc7 layer. We then apply SFL and DBSFL directly on the normalized VGG descriptors without first extracting LE-GRCM descriptors as in the previous experiment.

The recognition rates of the different methods are shown in Table 5.5 and the average CMC curves are shown in Figure 5.4. The use of deep features clearly results in superior overall classification performance compared to the shallow features used in Section 5.3.1.

In absolute terms, the best rank-1 classification accuracy obtained is now 85.39% on YTC compared to 76.91% obtained earlier, and 85.22% on YTF compared to 61.86% obtained earlier.

Considering one identity prediction only on YTC, SFL-SRC achieves the highest recognition rate, followed by DBSFL-SRC and DBSFL-CRC where the gap in performance is quite small. For $r > 1$ identity guesses, the CMC curves show that DBSFL-CRC achieves the highest rank-r recognition rate, followed by SFL-SRC and DBSFL-SRC. On YTF, DBSFL-CRC achieves, with a significant margin, the highest rank-1 recognition rate followed by the set-based SVM baseline. As we consider more identity guesses, DBSFL-CRC maintains the highest rank-r recognition rate up to $r = 6$ guesses. The results also supports our previous observation the dictionary-based DBSFL-CRC always outperforms the imbalance-sensitive SFL-CRC version.

Other discriminative dimensionality reduction approaches considered in this experiment are not as effective as SFL and DBSFL at improving the performance of MS-CRC and MS-SRC. In particular, Table 5.5 shows that LMNN-SRC and LMFL-SRC are inferior to MS-SRC on YTC and LMNN-SRC is inferior to MS-SRC on YTF. Only LMFL-SRC insignificantly outperforms MS-SRC on YTF. Similarly, LMFL-CRC and LMNN-CRC are inferior to MS-CRC on YTC and LMFL-CRC is inferior to to MS-CRC on YTF. Only LMFL-CRC slightly outperforms MS-CRC on YTF. Both LMNN and LMFL are designed to minimize intra-class sample to sample distances while maximizing inter-class sample-to-sample distances. Instead, SFL and DBSFL consider intra/inter-class sample-to-subspace distances, which makes them more appropriate for subspace-based classifiers like SRC and CRC.

Figure 5.5: Sample face images from the IJB-B dataset [130]. The pair of images in each column show the same subject. The IJB-B dataset has more challenging pose variations and more geographically diverse subject pool.

5.3.3 Other Challenges: More Classes and Smaller Gallery

While YTC and YTF involve challenges with low resolution frames, we now consider a dataset with other kinds of challenges, namely hundreds of classes in the gallery and very few images per class (sometimes just one image) (see Table 5.4). The dataset has other challenges as described below.

5.3.3.1 IARPA Janus Benchmark-B (IJB-B)

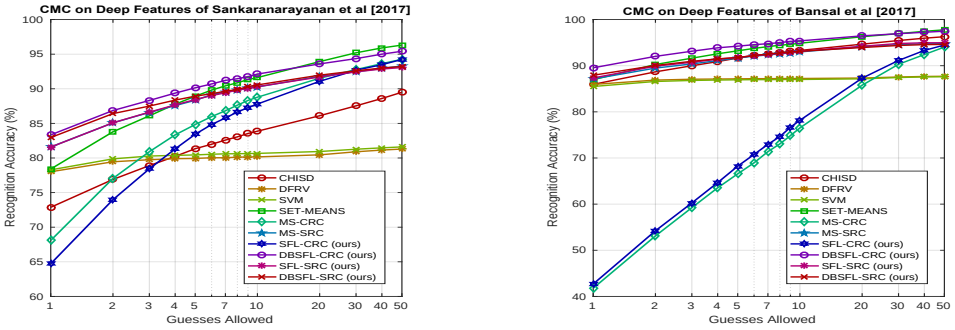
The IJB-B dataset contains 11,754 still images and 7,011 videos of 1,845 different subjects, with an average of 6 still face images/subject, 30 video frames/subject, and 4 videos/subject [130]. Compared to other datasets like YTF, IJB-B tends to have larger pose variations (see Figure 5.5) and more geographically diverse subject pool [130]. The dataset offers multiple evaluation protocols. In this chapter, we consider the closed-set 1-to-N identification protocol based on sets of still images. In that protocol, the set of still face images are divided into three disjoint subsets: two gallery sets and one probe set. The first gallery set contains 931 image sets for 931 subjects whereas the second contains

Table 5.6: The two-fold sample mean and standard deviation of the recognition rates obtained with our methods using the deep descriptors of Sankaranarayanan et al. [106] and Bansal et al. [11] on the IJB-B benchmark based on still image sets. Since some classes have only one single-image image set in the gallery, many image set classification methods cannot be applied under such setting and thus we have excluded them from comparison. This also makes it hard for discriminative feature learning/dimensionality reduction methods that use triplet loss where it is assumed that each class has at least two gallery images (which we have also excluded). We have highlighted in bold the rates of the top two performing methods for each dataset.

Methods	Sankaranarayanan et al. [106]	Bansal et al. [11]
CHISD (CVPR, 2010)	72.88 ± 2.78	85.96 ± 0.64
DFRV (ECCV, 2012)	78.01 ± 0.90	85.95 ± 0.66
SVM	78.28 ± 1.78	85.50 ± 0.97
SET-MEANS	78.41 ± 3.56	86.97 ± 1.56
MS-CRC (ICCV, 2011)	68.09 ± 3.03	41.78 ± 9.40
MS-SRC (CVPR, 2013)	81.59 ± 0.58	87.13 ± 0.35
SFL-SRC (ours)	81.65 ± 0.75	87.44 ± 0.28
DBSFL-SRC (ours)	$\mathbf{82.96 \pm 0.77}$	$\mathbf{87.93 \pm 1.08}$
SFL-CRC (ours)	64.70 ± 3.57	42.68 ± 8.82
DBSFL-CRC (ours)	$\mathbf{83.34 \pm 1.09}$	$\mathbf{89.55 \pm 1.44}$

914 image sets for the remaining 914 subjects. The probe set contains 8,104 probe sets covering all the 1,845 subjects. Accordingly, we run a two-fold experiment where fold i trains on the gallery set i and tests on the probes corresponding to the subjects in gallery i, for $i = 1, 2$. In both gallery sets, there are very few images per subject and different subjects have different numbers of images (i.e. the gallery is imbalanced) with some subjects having only one image in their gallery image sets. This makes IJB-B challenging for many image set classification methods which often assume each class has one or more gallery image sets containing more than just one image. In fact, the IJB-B benchmark breaks the basic assumption of some of the competing methods which require a minimum image set cardinality > 1 in the gallery for each class and so we we report results for only the competing methods that can operate under these limitation.

Figure 5.6: The mean Cumulative Matching Characteristic (CMC) curves for the IJB-B dataset, based on the deep features of Sankaranarayanan et al. [106] (left) and Bansal et al. [11] (right). DBSFL-CRC is the top-performing method achieves the highest CMC curve on the IJB-B dataset up to 14 guesses using Sankaranarayanan et al. [106] features and 29 guesses using [11] features. The CMC curve of SET-MEANS becomes the highest afterwards. The methods based on majority voting like set SVM and DFRV fail to improve recognition accuracy when additional identity guesses are allowed. This is because the query image sets in IJB-B contain much fewer samples than the number of available classes. This in turn reduces the probability that the correct class labels is randomly chosen within the top r identity guesses generated by the classifier.



5.3.3.2 Experimental Settings

To extract features, we use the ground truth pose information that comes with every face image to align the face in that image. Then we run two separate experiments using two different kinds of deep features extracted from the aligned images using the deep networks of Sankaranarayanan et al. [106] and Bansal et al. [11].

We compare in this experiment the set-based SVM [18], CHISD [17], MS-CRC [145], MS-SRC [93], our methods SFL-CRC, DBSFL-CRC, SFL-SRC, and DBSFL-CRC. We also include a popular baseline approach [130], SET-MEANS, which computes the mean of the normalized descriptors in each gallery image set and classifies a query image set by matching the mean of its normalized descriptors against the mean of each class in the gallery. We set the parameters of SFL and DBSFL as in the earlier experiments except for the number k of atoms per class in DBSFL which we set to $k = 1$. This is because some

of the classes in the IJB-B dataset have just one image sample in gallery in addition to the fact that other classes have very few samples in the gallery. For the same reasons, we adjust the parameters of DFRV so that we use in one cluster, one dictionary containing one atom for every class.

5.3.3.3 Results

The mean recognition rates of the different methods and different features are shown in Table 5.6. The corresponding mean CMC curves are shown in Figure 5.6. Similar to previous experiments, DBSFL-CRC achieves the highest rank-1 accuracy, followed by DBSFL-SRC and SFL-SRC. However, the gap in performance between DBSFL-CRC and SFL-CRC is huge in this experiment. This is caused by the generally smaller gallery and the relatively high imbalance among classes in terms of the number of labeled images available for each class. Due to its inherent balancing mechnism, DBSFL allows the computatonally efficient but imbalance-sensitive CRC to competitively perform on this datasets, as well as the previous datasets.

Table 5.7: Training and average test times (per image set) for the different methods in seconds. These times were measured on an identical setup over the first fold of the YTC dataset. The table clearly shows the time SFL takes once to train results in significant speedup in classification time for SFL-CRC compared to MS-CRC. The classification speedup is even more significant with the dictionary-based variant DBSFL-CRC, which has the fastest classification performance among all competing methods in addition to achieving the highest accuracy on all the benchmarks (except for YTC-VGG benchmark where its rank-1 classification accuracy is very close to the highest performance).

Method	Train Time (s)	Test Time (s)
AHISD (CVPR, 2010)	18.81	18.82
CHISD (CVPR, 2010)	N/A	10.85
SANP (CVPR, 2011)	N/A	82.99
DFRV (ECCV, 2012)	2870.99	45.63
MS-CRC (ICCV, 2011)	13.38	0.25
MS-SRC (CVPR, 2013)	N/A	3.88
SSDML (ICCV, 2013)	3447.62	56.92
DRM (CVPR, 2014)	12511.44	4.67
PML (CVPR, 2015)	2433.36	2.09
LEML (ICML, 2015)	2667.53	153.11
SFL-SRC (ours)	401.95	2.29
DBSFL-SRC (ours)	1107.88	0.52
SFL-CRC (ours)	404.03	0.07
DBSFL-CRC (ours)	1109.10	0.02

5.3.4 Running Times

Table 5.7, we report the time taken to train different methods on the first fold of YTC as well as the time taken to classify a single test image set of size 165 from YTC. All methods were tested on MATLAB using a system running at 2.2 GHz. The testing time of all methods was measured using a single-threaded MATLAB process. The test times reveal that SFL-SRC has 1.7 times faster classification compared to MS-SRC while DBSFL-SRC is 7.5 times faster than MS-SRC. Similarly, SFL-CRC is 3.6 times faster than MS-CRC while DBSFL-CRC is 12.5 times faster than MS-CRC. Overall, DBSFL-CRC has the

fastest classification among the compared methods.

5.4 Summary

We proposed an approach for image set classification using GRCM feature extraction, Subspace Feature Learning (SFL) on Log-Euclidean (LE) tangent space tangent of the SPD manifold $\mathbb{S}_+^Q$, and representation-based classification. We also proposed a dictionary-based variant of the method (DBSFL) for dealing with imbalanced galleries and empirically showed that the proposed approach outperforms other image-set classification methods in terms of accuracy and speed. Extensive experiments on four challenging datasets and different kinds of shallow and deep features showed the superiority of the proposed approach.

Chapter 6

Nonlinear Subspace Feature Enhancement for Image Set Classification

6.1 Introduction

Despite the theoretical foundations of the subspace model, the success of the associated algorithms relies on how well these assumptions are satisfied in practice (i.e. the convexity of the imaged object, the fixing of viewpoint, the Lambertian illumination, and the use of raw intensities to represent images). In practical unconstrained settings, these requirements may not be met and so the data may not strictly follow the subspace model in such scenarios (e.g. varying pose and/or use of image features nonlinearly derived from intensities).

To mitigate this, we propose an algorithm to learn a nonlinear embedding that enhances the low-dimensional discriminative subspace structure of the image sets. Under

such an embedding, an instance from one class is more likely to be closer to the subspace spanned by the samples of the same class than to the subspaces spanned by the samples from other classes. This can enhance the performance of subspace-based classifiers, such as Sparse-Representation-based Classification (SRC) [133], which essentially finds a low-dimensional subspace that is closest to the test sample and uses the labels in that subspace to decide a label for the particular test sample. Given a batch of samples, we formulate a novel structured loss function that encourages the distance between each sample and the subspace spanned by the same-class samples (within the batch) to be lower than the distances between the sample and the subspaces spanned by other classes (present in the batch). We then present a two-step alternating optimization algorithm to minimize the loss function in a way that is compatible with back-propagation. This allows the function to be minimized with Stochastic Gradient Descent (SGD)-based algorithms that are typically used to train deep networks [31, 110]. At the end of training, the learned embedding is used to project the image sets and the Mean-Sequence SRC (MS-SRC) [93] is used to classify the test image sets.

The rest of this chapter is organized as follows. We describe in Section 6.2 the structured loss function and the optimization procedure of the NSFE algorithm. We experimentally evaluate NSFE in Section 6.3 where the results show the superiority of NSFE compared to several existing image set classification methods. We conclude the chapter in Section 6.4.

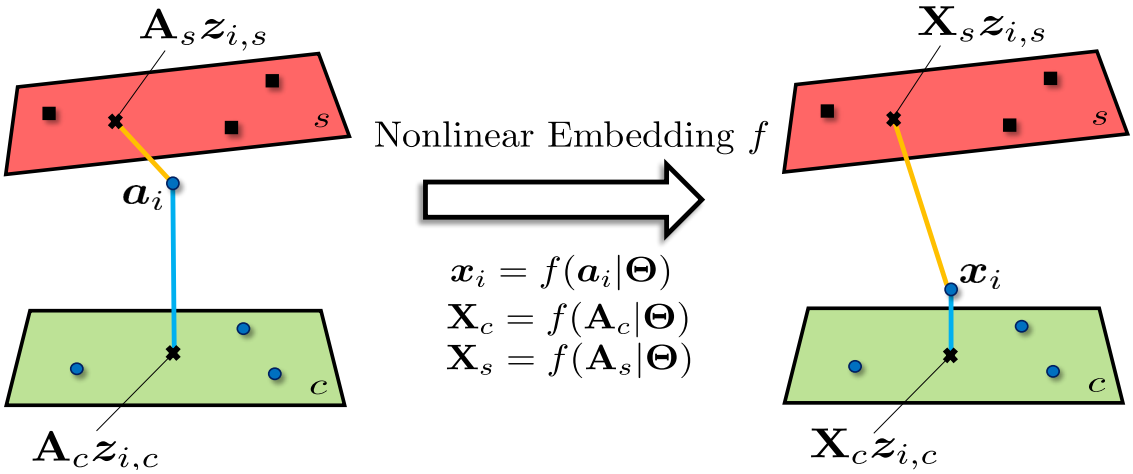


Figure 6.1: An illustration of images and class-specific subspaces before and after the embedding. NSFE aims to improve the discriminative subspace arrangement of the data such that the images of a particular class c lie closer to the subspace $\mathbf{X}_c = f(\mathbf{A}_c)$ spanned by that class than any subspace $\mathbf{X}_s = f(\mathbf{A}_s)$ spanned by any other class s.

6.2 Nonlinear Subspace Feature Enhancement (NSFE)

We assume that there is a mapping $f : \mathcal{A} \rightarrow \mathbb{R}^m$ that maps every input image $\boldsymbol{a}$ from the vector space $\mathcal{A}$ (i.e. the space of raw intensity images) to $\boldsymbol{x} = f(\boldsymbol{a})$ in some feature space $\mathbb{R}^m$. We further assume that the mapping f is parameterized by a real tensor $\boldsymbol{\Theta}$ and that the parameter subgradients of f : $\partial f / \partial \boldsymbol{\Theta}$ are defined. For example, the mapping f could be a neural network and $\boldsymbol{\Theta}$ could be the network weights. Assuming that during training labeled samples arrive in batches, our goal is to learn a value of the parameter tensor $\boldsymbol{\Theta}$ that would make an embedded sample $\boldsymbol{x}$ from a particular class c closer to the subspace spanned by batch samples from c than to any subspaces spanned by batch samples from any other class $s \neq c$.

Definitions and Notations: In the following discussion, we use $\mathbf{B}$ to denote the current batch of samples and $|\mathbf{B}|$ to denote the number of samples in the batch. Furthermore, we use n_c to denote the number of samples from class c present in batch $\mathbf{B}$ while $\mathbf{X}_c =$

$\left[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_c} \right] \in \mathbb{R}^{m \times n_c}$ is the matrix (dictionary) containing these samples along its columns. We use $C(\mathbf{B})$ to denote the set of class indices present in $\mathbf{B}$. In all our experiments, we sample each batch to contain nearly the same number of samples n_c from each class (the maximum difference between n_c and n_s is 1 for $c, s \in C(\mathbf{B})$). The sampling procedure ignores the boundaries between sets belonging to the same class and thus the subset drawn from a given class can contain samples from different sets within that class. In subsequent derivations, we assume $n_c > 1$ for all $c \in C(\mathbf{B})$ although in our experiments we have $6 \leq n_c \leq 20$. We also assume that the ith coordinate of a vector $\boldsymbol{z}$ is given by $\boldsymbol{z}^{(i)}$, the (i, j)th entry of a matrix $\mathbf{J}$ is given by $\mathbf{J}^{(i,j)}$, and the ith column is given by $\text{col}_i(\mathbf{J})$.

6.2.1 Structured Loss Function

Before describing the loss function to be minimized, we need to formulate some measures of distance between a sample and different subspaces. Assuming the ith sample $\boldsymbol{x}_i$ is associated with class c (i.e. $c(i) = c$), we let $\boldsymbol{z}_{i,c}$ denote the linear representation of $\boldsymbol{x}_i$ with respect to the dictionary $\mathbf{X}_c$ (which is formed by the batch samples of class c present in $\mathbf{B}$). The representation $\boldsymbol{z}_{i,c}$ is obtained by solving the optimization problem

$$\boldsymbol{z}_{i,c} = \underset{\boldsymbol{z} \in \mathbb{R}^{n_c}}{\text{argmin}} \|\boldsymbol{x}_i - \mathbf{X}_c \boldsymbol{z}\|_2^2 + \lambda \|\boldsymbol{z}\|_2^2, \text{s.t. } z^{(i)} = 0 \tag{6.1}$$

where we use l_2-norm instead of the sparsity inducing l_1-norm for efficiency purposes and also because n_c is typically small. It can be shown that

$$\boldsymbol{z}_{i,c} = \boldsymbol{u}_{i,c} - w_i \text{col}_i(\mathbf{J}_c^{-1}) \tag{6.2}$$

where $\mathbf{J}_c = \mathbf{X}_c^T\mathbf{X}_c + \lambda\mathbf{I}$, $\boldsymbol{u}_{i,c} = \mathbf{J}_c^{-1}\mathbf{X}_c^T\boldsymbol{z}_{i,c}$, and $w_i = \boldsymbol{u}_{i,c}^{(i)}/\mathbf{J}_c^{-1(i,i)}$. Similarly, we define the linear representation $\boldsymbol{z}_{i,s}$ of the sample $\boldsymbol{x}_i$ with respect to the dictionary $\mathbf{X}_s$ formed by the batch samples of a different class $s \neq c = c(i)$ as a solution to the following optimization problem

$$\boldsymbol{z}_{i,s} = \underset{\boldsymbol{z}\in\mathbb{R}^{n_s}}{\text{argmin}} \|\boldsymbol{x}_i - \mathbf{X}_s\boldsymbol{z}\|_2^2 + \lambda\|\boldsymbol{z}\|_2^2 \tag{6.3}$$

which has the closed form

$$\boldsymbol{z}_{i,s} = \mathbf{J}_s^{-1}\mathbf{X}_s^T\boldsymbol{x}_i \tag{6.4}$$

Our goal is to learn the embedding f such that we have

$$\|\boldsymbol{x}_i - \mathbf{X}_c\boldsymbol{z}_{i,c}\|_2^2 < \|\boldsymbol{x}_i - \mathbf{X}_s\boldsymbol{z}_{i,s}\|_2^2 \tag{6.5}$$

for all valid choices i, c, and s. If such a discriminative subspace property is achieved for all choices of $c, s, \mathbf{X}_c$, and $\mathbf{X}_s$, a test sample $f(\boldsymbol{q})$ can be reconstructed using the samples of the true class more accurately compared to the samples of other classes. Applying a subspace classifier (like SRC) is thus more likely to associate $f(\boldsymbol{q})$ with its true class.

The proposed structured loss function, which we call Large-Margin Subspace Loss

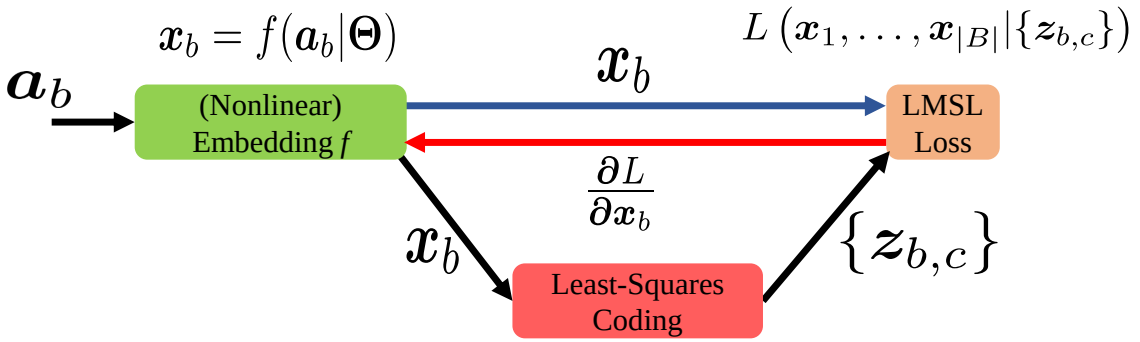


Figure 6.2: An illustration of the alternating learning algorithm. After embedding the samples in the forward pass, the sparse codes $\boldsymbol{z}_{b,c}$ are computed $\forall(b, c)$ and substituted into the loss function. The sparse codes are held constant, the loss function is evaluated, and the derivatives of loss function with respect to $\boldsymbol{x}_b, \forall b$ are back-propagated. The chain rule (6.7) is then applied to evaluate the parameter subgradients $\boldsymbol{\partial} L/\boldsymbol{\partial}\theta_k$ of the loss function, which can then be used to update the parameters by an SGD-like algorithm.

(LMSL), considers for every valid sample-to-subspaces-based triplet within the batch how well (6.5) is met. More specifically, LMSL is defined as

$$L = \frac{1}{T} \sum_{c \in C(\mathbf{B})} \sum_{\substack{i=1, \\ c(i)=c}}^{|\mathbf{B}|} \sum_{\substack{s \in C(\mathbf{B}), \\ s \neq c}} [\|\boldsymbol{x}_i - \mathbf{X}_c \boldsymbol{z}_{i,c}\|_2^2 + m - \|\boldsymbol{x}_i - \mathbf{X}_s \boldsymbol{z}_{i,s}\|_2^2]_+ \tag{6.6}$$

where m is the margin and the above sum is normalized by the number of terms/triplets T included the sum, which is $T = |\mathbf{B}|\,(|C(\mathbf{B})| - 1)$. It should be noted that the actual objective function being minimized is the sum of L and any other parameter regularization on $\boldsymbol{\Theta}$. LMSL can be thought of a kind of sample-to-subspace triplet loss [108, 128]. The loss function treats as an anchor every sample $\boldsymbol{x}_i$ in the batch $\mathbf{B}$. For each anchor $\boldsymbol{x}_i$, LMSL considers as its corresponding positive point the class projection $\mathbf{X}_c \boldsymbol{z}_{i,c}$ and as a negative point its projection on one of the other-class subspaces $\mathbf{X}_s \boldsymbol{z}_{i,s}$. Thus, we have a total of $|C(B)| - 1$ triplets that have the sample $\boldsymbol{x}_i$ as the anchor.

6.2.2 Learning Algorithm

The LMSL function L can be difficult to optimize jointly with respect to both the sparse codes and $\boldsymbol{\Theta}$. Accordingly, we follow an alternating optimization approach. In this approach, we evaluate the sparse codes of all batch anchors using (6.2, 6.4). Then, we treat the sparse codes as constants and use the chain rule and back-progation to compute the parameter gradients of the loss function $\frac{\partial L}{\partial \theta_k}$, which are necessary for updating $\boldsymbol{\Theta}$ (see Figure 6.2):

$$\frac{\partial L}{\partial \theta_k} = \sum_{b=1}^{|\mathbf{B}|} (\frac{\partial L}{\partial \boldsymbol{x}_b})^T \frac{\partial \boldsymbol{x}_b}{\partial \theta_k} \tag{6.7}$$

If we assume $\boldsymbol{x}_b$ is associated with class s, b is its index within the batch, and r is its column index within $\mathbf{X}_s$, the left factor $\frac{\partial L}{\partial \boldsymbol{x}_b}$ in the above inner product is given by:

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{x}_b} &= \frac{2}{T} \sum_{c \in C(\mathbf{B}), c \neq s} \{\Delta_{b,c} (\boldsymbol{x}_b - \mathbf{X}_s \boldsymbol{z}_{b,s}) \\ &- \Delta_{j,c} \sum_{j=1, c(j)=s, j \neq b}^{|\mathbf{B}|} \boldsymbol{z}_{j,s}^{(r)} (\boldsymbol{x}_j - \mathbf{X}_s \boldsymbol{z}_{j,s})\} \\ &+ \frac{2}{T} \sum_{i=1, c(i) \neq s}^{|\mathbf{B}|} \Delta_{i,s} \boldsymbol{z}_{i,s}^{(r)} (\boldsymbol{x}_i - \mathbf{X}_s \boldsymbol{z}_{i,s}) \end{aligned} \tag{6.8}$$

where $\Delta_{i,s}$ is a binary variable that is 1 iff the loss term corresponding to anchor sample i and negative class s is non-zero. The loss gradient in (6.8) is computed for each sample $\boldsymbol{x}_b$ in the batch and back-propagated for computing parameter updates. A summary of the learning algorithm of NSFE is given in Algorithm 2.

Input: A batch of samples $\left[\boldsymbol{a}_1, ..., \boldsymbol{a}_{|\mathbf{B}|}\right]$ and their labels.

1 Group batch samples by class.

2 Embed and l_2-normalize each sample in the batch: $\boldsymbol{x}_b = f(\boldsymbol{a}_b|\boldsymbol{\Theta}_t)$.

3 For each class $c \in C(\mathbf{B})$, use Cholesky-Factorization to invert $\mathbf{J}_c = \mathbf{X}_c^T\mathbf{X}_c + \lambda\mathbf{I}$.

4 For each class $c \in C(\mathbf{B})$, use Eq. (6.2) to compute the code vector of its batch samples with respect to $\mathbf{X}_c$.

5 For each class $c \in C(\mathbf{B})$, use Eq. (6.4) to compute the code vector of other-class samples in the batch with respect to $\mathbf{X}_c$.

6 Compute the LMSL loss L using Eq. (6.6). Compute and back-propagate the LMSL gradient $\frac{\partial L}{\partial \boldsymbol{x}_b}$, for $b = 1, \ldots, |\mathbf{B}|$.

7 Use the chain rule and Eq. (6.7) to compute the loss gradients $\frac{\partial L}{\partial \theta_k}$ of the parameters which can then be used to update these parameters.

Algorithm 2: NSFE Learning Algorithm Summary

6.2.3 Concrete Embeddings

Our method can work with any vector-space inputs and can easily utilize any nonlinear embeddings for which the parameter subgradients of $f : \boldsymbol{\partial} f / \boldsymbol{\partial \Theta}$ are defined, including feed-forward neural networks. We test the proposed method with two types of vector-space inputs: raw intensity images and the hand-crafted Log-Euclidean Grid of Region Covariance Matrices (LE-GRCM) features proposed in [36]. With intensity images as inputs, we use the 32-layer deep fully convolutional residual network proposed in [52] for the CIFAR-10 dataset. The network has the following configuration:

(a) An initial $3 \times 3 \times 16$ convolutional layer. The notation specifies 16 filters, each has a weight kernel of dimensions 3×3. The stride is always 1 in both directions.

(b) A first block of ten $3 \times 3 \times 16$ convolutional layers, with residual connections made every two layers. The last layer is followed by a 2×2 average pooling with a stride of 2 in both directions.

(c) A second block of ten $3 \times 3 \times 32$ convolutional layers. Residual connections and a final average pooling are defined for this block.

(d) A third block of ten $3 \times 3 \times 64$ convolutional layers. It uses residual connections in a similar fashion but does not have a subsequent pooling layer.

(e) A final $1 \times 1 \times 10$ convolutional layer that is not followed by any nonlinearities or batch normalization [57]. The output of that layer is reshaped as a vector and l_2-normalized to produce the final embedded feature vector.

The final layer replaces the global average pooling operation used in [52] in an attempt to retain spatial information in the computed features. Unless otherwise stated, we add batch normalization and ReLU nonlinearities according to the architecture in [52]. The total number of parameters in this architecture is 463,856, which is less than 0.5 million.

Since an LE-GRCM vector input is not a 2D image, we cannot use a conventional CNN for the embedding to process such hand-crafted features. Instead, we use a very basic, fully-connected 2-layer network with the following architecture: FC-3600 $\rightarrow$ ReLU $\rightarrow$ FC-406 $\rightarrow$ l_2-normalization, where FC-k is a linear fully-connected layer with k units.

6.2.4 Classification

After training, the learned embedding f is used to map the training data then we use the Online Dictionary Learning (ODL) algorithm described in [86] to compute a dictionary $\mathbf{D}_c$ for each class c. Given a test set, we use the learned embedding f to map it and we follow the MS-SRC approach [93] by computing the mean vector $\bar{\boldsymbol{y}}$ of the embedded test set and then using SRC to find a label for $\bar{\boldsymbol{y}}$. The details of ODL and MS-SRC algorithms

can be found in [86] and [93], respectively.

It is worth noting that ODL is an unsupervised algorithm and enhanced performance can be further achieved by using any of the discriminative dictionary learning algorithms instead of ODL. However, we only use ODL in the next section to objectively and more precisely evaluate the effect of NSFE on accuracy.

6.3 Experiments

We experimentally compare the performance of NSFE against several existing algorithms for image-set classification. The compared methods include Affine Hull-based Image Set Distance (AHISD) [17], its convex variant (CHISD) [17], Sparse-Approximated Nearest Points (SANP) [53], Dictionary-based Face Recognition from Videos (DFRV) [24], Mean Sequence Sparse Representation-based Classification (MS-SRC) [93], Set to Set Distance Metric Learning (SSDML) [149], Deep Reconstruction Models (DRM) [50], Covariance Discriminative Learning (CDL) [122], Log-Euclidean Metric Learning (LEML) [55], and the shallow subspace Feature Learning+SRC (FL+SRC) approach of [36] both with intensity images as inputs (FL+SRC) as well as LE-GRCM features (LE-FL+SRC). We show the results of our method with both intensity features as inputs (NSFE) and LE-GRCM features (LE-NSFE). For comparability, the results of other Log-Euclidean methods (i.e. CDL and LEML) are obtained using LE-GRCM features.

In the experiments, each method is given a set of labeled image sets for training and is required to classify (or more specifically identify) a number of test image sets. For performance comparison, we use the classification accuracy (i.e. recognition rate) as a

metric by measuring the percentage of test image sets that are correctly classified.

For existing methods, we have used the source code provided by the original authors and set the parameters according to the recommendations made in their respective papers.

NSFE Parameter Settings: In the experiments, we use SGD with momentum to update the weights of the embedding network in each iteration for a total of 50K iterations. The momentum is set to 0.9 and we use a learning rate schedule of 0.1 for the first 20K iterations then we divide it by 10 for each subsequent 10K iterations. For the 2-layer fully-connected network, we train for 20K iterations with a learning rate of 0.01 that we decrease to 0.001 after 10K iterations. We use a batch of size 128. We also set the representation regularization parameter λ of NSFE to 0.01, the margin $m = 0.5$, and the desired number of atoms in each class-specific dictionary computed by ODL to 50.

To guarantee a fair comparison with other methods and to accurately measure the ability of our method to learn effective features, we do not perform any pre-training on any external data and we initialize the weights of our embeddings randomly.

We run experiments over the YTC, YTF, and MobFaces datasets. Figure 4.3 shows examples from each dataset. A brief description of the datasets YTC, YTF, and MobFaces and the corresponding experimental protocols is covered in Sections 4.5.1, 4.5.2, and 4.5.3, respectively.

6.3.1 Results

Table 6.1 shows the mean recognition rate of the compared methods for the YTC and YTF datasets where we group the methods by the type of input features (raw images vs

Table 6.1: The mean recognition rates obtained with the compared methods on YTC and YTF.

Methods	**YTC**	**YTF**
AHISD	57.27	17.18
CHISD	64.79	32.99
SANP	66.99	31.62
DFRV	66.70	36.77
SSDML	69.22	34.02
DRM	70.35	43.99
MS-SRC	74.68	45.02
FL+SRC	75.71	45.36
NSFE (ours)	$*$**78.23**	**54.91**
Methods with LE-GRM Input		
CDL	67.62	41.92
LEML	73.26	48.45
LE-FL+SRC	76.28	53.26
LE-NSFE (ours)	**76.42**	$*$**56.66**

LE-GRCM). For each group, we highlight in **bold** the highest performance under each setting and we place an asterisk $*$ next to the single highest overall performance for that setting. For both datasets and types of input features, the proposed method, NSFE/LE-NSFE, achieves the highest mean recognition rate. Tables 6.2 and 6.3 show the recognition rates for the six different splits for the MobFaces dataset where we use the same grouping and highlighting adopted by Table 6.1. The training image sets of this dataset contain very limited visual variations (namely once short video per subject for each setting in MobFaces-I and two such videos in MobFaces-II) while the test image sets are captured under ambient conditions different from those of training. Meanwhile, our method (NSFE/LE-NSFE) ranks among the top two best performing methods under each individual setting and it is the best performing on average for each of the two protocols, with a significant margin on MobFaces-II due to the availability of more images and visual variations during training in

Table 6.2: The recognition rates obtained on the MobFaces dataset under the MobFaces-I protocol. The setting $(1 \rightarrow \{2, 3\})$ involves training on session 1 (i.e. the lit session) and testing on sessions 2 and 3 (i.e. the unlit and day-lit sessions). The other two settings are defined in a similar manner. The 'avg' column contains the average of the rates obtained under the three settings to its left. Since each session has a different number of test video clips, the average column weighs the rate of each setting by its number of test sets.

	MobFaces-I			
Methods	$1 \rightarrow \{2, 3\}$	$2 \rightarrow \{1, 3\}$	$3 \rightarrow \{1, 2\}$	avg
AHISD	15.00	31.14	29.30	26.12
CHISD	10.61	26.57	25.73	21.96
SANP	9.34	27.09	26.15	21.96
DFRV	19.39	32.29	30.87	28.30
SSDML	10.53	28.89	26.15	22.95
DRM	33.28	38.94	37.95	37.06
MS-SRC	32.40	46.56	42.49	41.29
FL+SRC	32.88	**46.97**	42.25	41.48
NSFE (ours)	**47.01**	46.27	**46.49**	**46.55**
Methods with LE-GRM Input				
CDL	41.66	36.78	42.68	40.21
LEML	42.70	45.93	44.07	44.39
LE-FL+SRC	48.20	***56.21**	54.90	53.58
LE-NSFE (ours)	***49.08**	49.68	***61.38**	***53.69**

that protocol. This shows that our method achieves relatively higher gain in performance as more deta and variations become available for training.

Table 6.3: The recognition rates obtained on the MobFaces dataset under the MobFaces-II protocol. The setting ($\{1,2\} \rightarrow 3$) involves training on sessions $\{1,2\}$ (i.e. the enrollment samples of the lit and un-lit sessions) and testing on session 3 (i.e. the task samples of the day-lit session). The other two settings are defined in a similar manner. The 'avg' column contains the average of the rates obtained under the three settings to its left. Since each session has a different number of test video clips, the average column weighs the rate of each setting by its number of test sets.

	MobFaces-II			
Methods	$\{2,3\} \rightarrow 1$	$\{1,3\} \rightarrow 2$	$\{1,2\} \rightarrow 3$	avg
AHISD	24.41	51.28	52.85	39.39
CHISD	23.29	44.97	47.60	35.76
SANP	20.38	48.89	45.95	34.94
DFRV	32.11	50.60	52.40	42.62
SSDML	21.31	50.09	54.95	38.27
DRM	**53.62**	70.53	69.37	62.42
MS-SRC	43.29	71.89	75.53	59.79
FL+SRC	44.98	72.40	76.58	61.00
NSFE (ours)	52.11	***81.43**	**83.63**	**68.59**
Methods with LE-GRM Input				
CDL	63.57	67.12	65.32	64.97
LEML	49.39	66.95	74.62	61.09
LE-FL+SRC	62.72	75.64	86.19	72.74
LE-NSFE (ours)	***68.92**	**76.15**	***87.84**	***76.19**

6.4 Summary

We presented NSFE, an approach for discriminatively learning a nonlinear embedding that can improve the subspace structured representation of image sets, and thus improve the performance of subspace-based classifiers such as MS-SRC. Since the proposed structured loss function LMSL is minimized in an online fashion, the proposed approach can be used to train existing feed-forward architectures via back-propagation. The minimization algorithm can also utilize the capabilities of modern GPUs, which provide APIs for solving batches of small linear systems of equations. In fact, all the linear systems solved in our batch processing algorithm are small, ranging from 6×6 to 22×22 systems of

equations, depending on the number of samples from a certain class available in the batch. Consequently, we were able to train and test many copies of our model for the different experiments described above without facing any unusual delays.

Chapter 7

Hierarchical Metric Learning and Matching for Correspondence Estimation

7.1 Introduction

Recent deep metric learning approaches for correspondence estimation extract feature descriptors from the deepest layers [26, 125, 137, 144], with the expectation that the minimization of the loss would encourage the deep layer to produce good features. On the contrary, several studies [143, 148] suggest that deeper layers respond to high-level abstract concepts and are by design invariant to fine variations in the input image. The same studies also reveal that shallower layers are receptive to local regions in image space and are thus more sensitive to local structures and geometric details.

In this chapter, we empirically show that deep metric learning approaches that rely

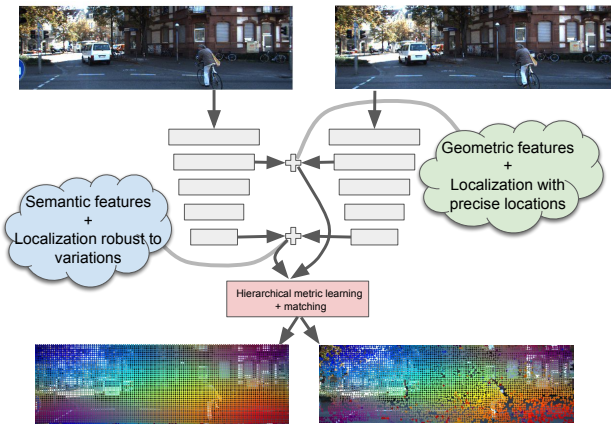


Figure 7.1: Our hierarchical approach to metric learning retains the best properties of various levels of abstraction in CNN feature representations. For geometric matching, we combine the robustness of deeper features that imbibe greater invariance, with the localization sensitivity of shallower features. This allows learning better feature representations, as well as an improved better correspondence search strategy that progressively exploits feature representations from higher recall (robustness) to higher precision (spatial sensitivity).

solely on the deepest features to be sub-optimal, and that superior matching performance can be achieved by utilizing both shallower and deeper features. Furthermore, we leverage recent studies that highlight the importance of carefully marshaling the training process: (i) by deeply supervising [64, 70] intermediate feature layers to learn task-relevant features, and (ii) on-the-fly hard negative mining [26] that forces each iteration of neural network training to achieve more. Finally, we exploit the intermediate activation maps generated within the CNN itself as a proxy [27] for image pyramids traditionally used to enable coarse-to-fine matching. Thus, at test time, we employ a hierarchical matching framework, using deeper features to perform coarse matching, benefiting from greater context and higher level visual concepts, followed by a fine grained matching step that involves searching for shallower features. Figure 7.1 illustrates our proposed approach.

The list of our contributions in this chapter is given below:

- We demonstrate that applying a metric learning correspondence loss on deeper layers

does not always result in more robust features. Instead, the more geometrically sensitive features produced by shallower layers allow for more precise matching.

- We empirically show that applying correspondence loss along with hard-negative mining at both shallower and deeper layers result in better matching performance than the supervision of one layer at a time or the fusion of features through hyper-columns [48] or top-down refinement [99].

- To combine the high recall of deeper features with the high-precision of shallower ones, we propose a CNN-based scheme for coarse-to-fine bi-level hierarchical matching.

- We experimentally validate our ideas by comparing against state-of-the-art CNN architectures for local description approaches and feature fusion baselines. In addition, we perform an ablative analysis of our proposed solution. We evaluate for the tasks of interest point matching as well as optical flow.

The remainder of the chapter is organized as follows. We review the relevant literature in Section 7.2 and introduce our framework including details of our network architecture and matching approach in Section 7.3. We discuss experimental results in Section 7.4, concluding the chapter in Section 7.5.

7.2 Related Work

The SIFT [78] descriptor proposed at the end of the 1990s, spurred a revolution in computer vision, and remains to this day the most highly cited work in the entire field. Over

the next decade, hand-crafted descriptors were applied to every known problem in computer vision. Further, several variants of SIFT were proposed targeting specific problems: SURF [13] and FPFH [105] for low-latency applications (the latter for 3D point clouds), and ShapeContext [15] and HOG [28] for object class recognition. With the revival of deep neural networks, many new ideas have emerged both pertaining to learned feature descriptors and directly learning networks for low-level vision tasks in an end-to-end fashion. We review these lines of work in the following.

Hand-crafted Descriptors: Precise registration tasks have benefited the most from matching sparse interest points, as is the standard in almost all Structure-from-Motion (SfM) pipelines [2, 90, 91]. SIFT [78], SURF [13], BRISK [68] were thus designed to complement high curvature point detectors, with [78] even proposing its own algorithm for such a detector. These methods provide varying degrees of invariance to imaging variations such as lighting changes or local affine transformations. In fact, despite the interest in learned methods, they are still the state-of-the-art for precision [10, 107], even if they are less effective in achieving high recall rates.

Learned Descriptors: While early work [74, 77, 129] leveraged intermediate activation maps of a CNN trained with an arbitrary loss for the task of keypoint matching, most recent methods rely on an explicit metric loss [26, 39, 135, 137, 142, 144, 146] to learn descriptors. The hidden assumption behind the use of contrastive or triplet loss at the final layer of a CNN is that this explicit loss will cause the relevant features to emerge at the top of the feature hierarchy; despite the conventional wisdom [143] that it is the early layers of the CNN that learn local geometric features. Consequently, many of these works

demonstrate superior performance to handcrafted descriptors on semantic matching tasks but often lag far behind on geometric matching.

Universal Correspondence Network (UCN) [26] combines a fully convolutional network in a Siamese setup, with a spatial transformer module [58] and contrastive loss [25]. A GPU implementation is employed to speed up k-nearest neighbour search which allows performing on-the-fly hard negative mining. We employ a similar approach to perform negative mining, albeit across multiple feature learning layers. UCN beat the state-of-the-art on different semantic matching tasks, but only demonstrates geometric feature matching as a side task with limited accuracy.

Recently, AutoScaler [125] explicitly applies a learned feature extractor on multiple scales of the input image. While this takes care of the issue that a deep layer may have an unnecessarily large receptive field when learning on the basis of contrastive loss. Our approach avoids reprocessing the image at different scales by utilizing the feature pyramid by CNN that is typically generated while processing the input image.

Learned Optical Flow: Recent works have also shown state-of-the-art results on optical flow by training CNNs in an end-to-end fashion [30, 56], followed by Conditional Random Field (CRF) inference [103] to capture crisp object boundaries. These methods rely on either concatenating the image pair as separate channels or employ two separate input branches (Siamese architecture) to feed the input image, and a fully convolutional architecture to generate an output image. Synthetically generated motion [30] has been found useful to mitigate the problem of not having sufficient real image training data, and luckily transfers well to real test pairs. Unfortunately, most problems in computer vision are not amenable to such end-to-end learning since the long-tailed distribution of

real world scenes makes synthesizing data for most tasks impossible. Thus, we believe in leveraging deep learning to strengthen individual modules of otherwise extensively engineered vision pipelines.

Deep Supervision: Recent evidence [64, 70, 141] suggests that providing explicit supervision to intermediate layers of a CNN has the potential to yield higher performance on unseen data, by regularizing the training process. However, to the best of our knowledge, the idea has neither been tested on the task of keypoint matching nor have the intermediate features thus learned been evaluated. We do both in our work.

Image Pyramids: Downsampling pyramids have been a fixture of computer vision algorithms since early days [81]. As recently demonstrated in [27] for image alignment, we argue that the growing receptive field in deep CNN layers [143] provide a natural way to parse multiple an image at various scales. In our hierarchical matching scheme, we thus employ features extracted from a deeper layer with greater receptive field that capture higher-level semantic notions [148] for coarsely locating the interest point, followed by shallower features for precise registration.

7.3 Method

We introduce new ideas into recent metric learning frameworks that learn interest point descriptors. In the following, we first identify the general principles behind our framework, and then propose concrete neural network architectures that realizes these principles.

7.3.1 Hierarchical Metric Learning

We follow the standard CNN-based metric learning setup proposed as the Siamese architecture [25]. This involves two Fully Convolutional Networks [76] with tied weights, parsing two images of the same scene. We extract sub-blocks out of intermediate convolutional layer activation maps around the locations corresponding to the interest points in the input images, and after normalization obtain their Euclidean distance. At training time, separate contrastive losses applied to multiple levels in the feature hierarchy encourage the network to learn embedding functions that place descriptors for matching interest points close together in Euclidean space, whereas unmatched interest points are moved far apart.

Correspondence Contrastive Loss (CCL): We borrow the correspondence contrastive loss formulation introduced in [26], and adapted from [25]. Here, $\phi_I^l(x)$ represents the feature extracted from the lth feature level in the source image (versus I' for destination image) at pixel location $x \in \mathbf{Z}^2$. Let $\mathcal{D}$ represent a dataset of triples (x, x', y), where x is a location in the source image, x' is a location in the destination image, and $y \in \{0, 1\}$ is 1 iff. (x, x') are a match. Letting m be a margin parameter, we define the notation

$$\hat{\phi}_I^l(x) := \frac{\phi_I^l(x)}{\|\phi_I^l(x)\|_2} \tag{7.1}$$

$$d_{x,x'}^l := \|\hat{\phi}_I^l(x) - \hat{\phi}_{I'}^l(x')\|_2 \tag{7.2}$$

we define our training loss $\mathcal{L}$ in the following way, which sums CCL losses over multiple levels l:

$$\mathcal{L} := \sum_{l=1}^{L} \sum_{(x,x',y)\in\mathcal{D}} y(d_{x,x'}^l)^2 + (1-y)\max(0, m - d_{x,x'}^l)^2 \tag{7.3}$$

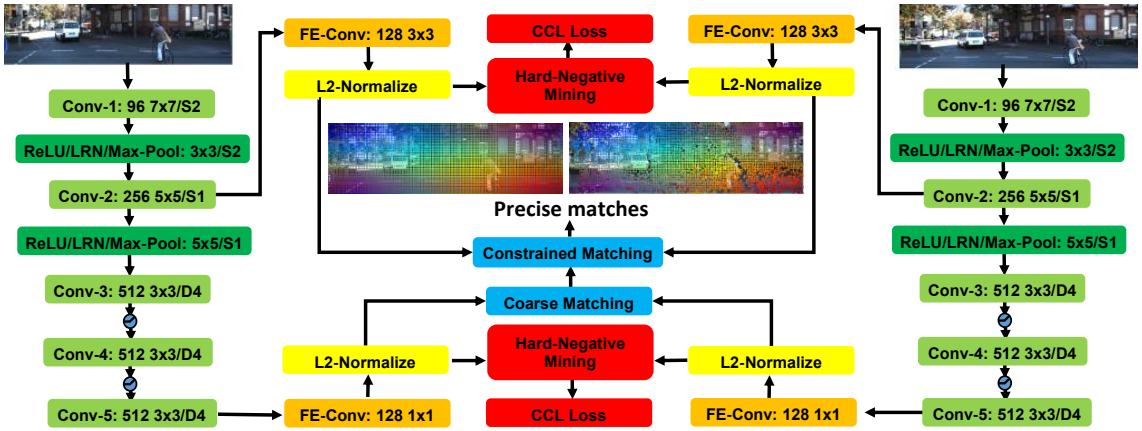


Figure 7.2: One instantiation of our proposed ideas. Note that the hard-negative mining and CCL losses (red blocks) are relevant for training, and matching (blue blocks) for testing. Convolutional blocks (green) on the left and right Siamese branches share weights. S and D denote striding and dilation offsets.

Deep Supervision: Our rationale behind applying the loss at multiple levels of the feature hierarchy is twofold. First, recent studies [64, 70] indicate that deep supervision of a CNN architecture contributes to improved regularization, by encouraging the network early on to learn task-relevant features and avoid overfitting to accidental regularities in the middle layers. Secondly, we know from literature that [143] shallow CNN layers learn lower-level geometric features, whereas deeper layers [148] not only have larger receptive fields but also tend to learn higher level semantic concepts. Our experiments demonstrate that this causes shallower features to be better at matching for small pixel thresholds, and deeper features to be better for large pixel thresholds.

Hard Negative Mining: Since in practice our training data only explicitly includes positive correspondences, we actively search for hard negative matches in the set of training correspondences i.e. false interest point pairs whose distances in the learned space fall below the matching threshold. During the forward pass, we use the feature $\phi_I(x)$ obtained at the source point x of a positive correspondence (x, x') to find the point $\|x'' - x'\|_2^2 > \sigma$ in

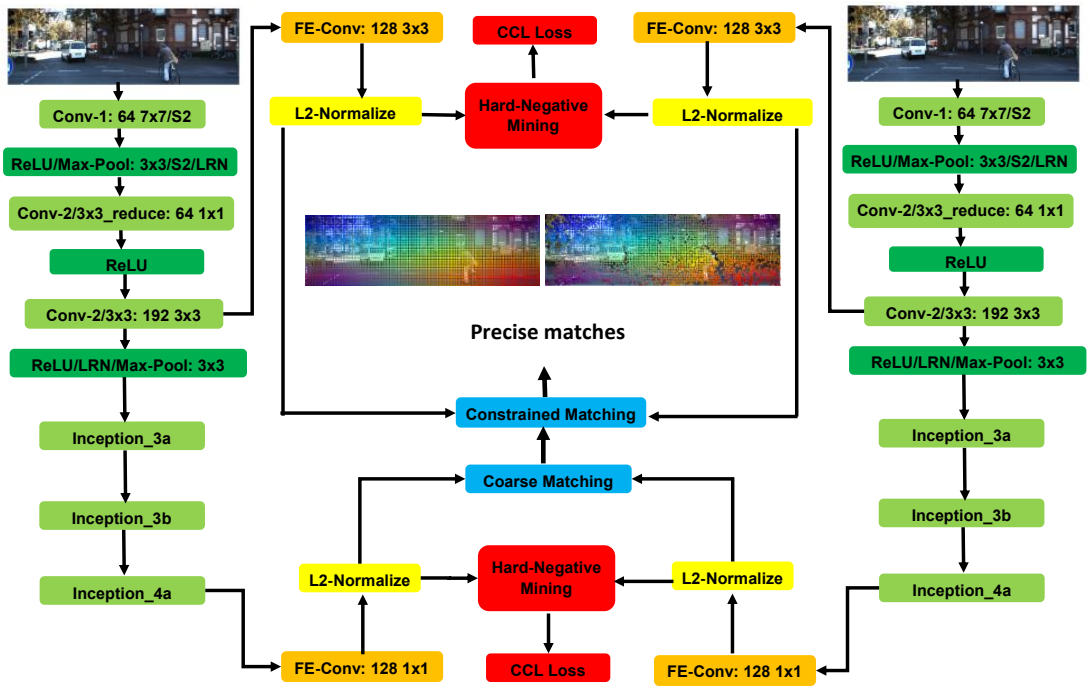


Figure 7.3: An instantiation of our proposed approach using a GoogLeNet baseline truncated after the *inception_4a* layer [111]. We use *conv2/3x3* as the source for shallow features and *inception_4* as the source for deep features (which UCN [26] uses as the sole source of features).

the destination image whose descriptor $\phi_{I'}(x'')$ is most similar to $\phi_I(x)$. Unlike UCN, our hard-negative mining happens independently for each of the feature levels being supervised. We perform this mining "on-the-fly" to leverage the latest instance of network weights, as [26].

Implementation Specifics: We visualize one specific instantiation of the above ideas in Figure 7.2, adapting the VGG-M [19] architecture for the task. We retain the first 5 convolutional layers, initializing them with weights pre-trained for ImageNet classification [104]. We use ideas from semantic segmentation literature [21, 138] to increase the resolution of the intermediate activation maps by (a) eliminating down-sampling in the second convolutional and pooling layers (setting their stride value to 1 down from

2) (b) increasing the pooling window size for the second layer from 3x3 to 5x5 and (c) dilating [138] the subsequent convolutional layers (*conv3*, *conv4* and *conv5*) to retain the receptive fields they were pretrained on.

At training, the network is provided a pair of images and a set of point correspondences. The network is replicated in a Siamese scheme [25] during training (with shared weights) where each sub-network processes one image from the pair; and thus after each feed-forward pass, we have 4 feature maps: 2 shallow ones and 2 deep ones, respectively from the second and fifth convolutional layers (*conv2*, *conv5*). We apply explicit supervision after these same layers (*conv2*, *conv5*).

We also experiment with a GoogLeNet [111] baseline as employed in UCN [26]. Specifically, we augment, as shown in Figure 7.3, the network with a 1x1 convolutional layer and L2-normalization following the fourth convolutional block (*inception_*4a/output) as in UCN [26]. In addition, we augment the network with a 3x3 convolutional layer right after the second convolutional layer (*conv2*/3x3, followed by l2-normalization, but before the corresponding non-linear ReLU squashing function. We extract the shallow and deep feature maps based on the normalized outputs after the second convolutional layer *conv2*/3x3 and the *inception_*4a/output layers respectively.

7.3.2 Hierarchical Matching

We adapt and train our networks as described in the previous section, optimizing network weights for matching using the features extracted from different layers. Yet, we find that features from different depths offer complementary capabilities as predicted by

earlier investigation [143, 148] and confirmed by our empirical evaluation in Section 6.3. Specifically, features extracted from shallower layers obtain superior matching accuracies for smaller distance thresholds (precision), where as those from deeper layers provide better accuracies for larger distance thresholds (recall). Such coarse-to-fine matching has been used in computer vision for almost four decades [81], however recent work highlights how employing CNN feature hierarchy for the task (at least in the context of image alignment [27]) is more robust.

2D correspondences: We compare a point p in I against a point p' in I', utilizing the corresponding feature maps (ϕ^l_I, $\phi^l_{I'}$, ϕ^l_I, $\phi^h_{I'}$) (where l and h refer to the indices of the shallower and deeper feature maps respectively). Specifically, we use the deep feature maps $\phi^h_I(p)$ and $\phi^h_{I'}(p')$ to find an initial estimate of the nearest neighbor of p, which is p' in I'. Next, we use ϕ^l_I and $\phi^l_{I'}$ to refine this estimate to get a precise match.

7.4 Experiments

In this section, we first provide our implementation details and parameter settings that are used in our following experiments. Next, we benchmark our proposed method for correspondence estimation against single-level based metric learning and matching approaches, feature fusion-based approaches, and state-of-the-art learned and hand-crafted methods for extracting correspondences. Further, we show how our feature-based method for correspondence estimation can be applied for optical flows and compare it against state-of-the-art feature-based methods and multi-cue methods for optical flow estimation. In the following, we denote our method as *BiL+BiM*, which is short for Bi-level metric

Learning and Bi-level Matching.

7.4.1 Implementation Details and Parameter Settings

We implement our system and other systems in our comparisons using Caffe [61] and we train each on a single P6000 GPU. The proposed constrained bi-level matching is implemented using CUDA and is run on the GPU. It takes our system an average of 8.41 seconds to densely extract features and compute per-pixel correspondences for a pair of input images of size 1242x376 each.

We use the ADAM stochastic minimization algorithm [63] to train our network for $50K$ iterations using a base learning rate of 10^{-3}. Pre-trained layers are fine-tuned with a learning rate multiplier of 0.1 whereas the weights of the newly-added feature-extraction layers are randomly initialized using Xavier's method [40]. We use a weight decay parameter of 10^{-4} and L2 weight regularization. During training, each batch consists of three randomly chosen image pairs and we randomly choose 1K positive correspondences from each pair. It takes the VGG-M variant of our system around 43 hours to train whereas it takes 30 hours to train our GoogLeNet-based variant. During testing, we set fix the search radius of the low level location refinement of BiM matching to 32 pixels (measured with respect to the input image space).

7.4.2 Correspondence Estimation Experiments

We empirically evaluate the performance of our approach against different approaches for obtaining feature descriptors for correspondence estimation. We first consider single-level

Figure 7.4: One siamese branch of two of the baseline architectures considered in our evaluation, namely *conv3* (left) and *hypercolumn-fusion* (right). The *conv3* is obtained by truncating all layers after *conv3* in the VGG-M architecture in Figure 7.2. Other *convi* baselines are obtained similarly. The 1x1 max-pooling after *conv1* in the *hypercolumn-fusion* baseline as added to down-sample the *conv1* feature map for valid concatenation with other feature maps.

based metric learning and matching approaches where we separately train five networks, based on the VGG-M baseline in Figure 7.2. Each one of the five networks has a different depth and we refer to the ith network by *convi* to indicate the network is truncated at the *convi* layer, for $i \in 1, 2, ..., 5$. The left side of Figure 7.4 shows one branch of the *conv3* baseline as an example. We train the *convi* network by adding a feature extraction convolution, L2-normalization and CCL loss to the output of the last layer (which is *convi*).

In addition, we also compare our method against two approaches for fusing features from different layers that are inspired by corresponding methods for semantic segmentation [48, 99]. One is *hypercolumn-fusion* [48] (right side of Figure 7.4) where features from

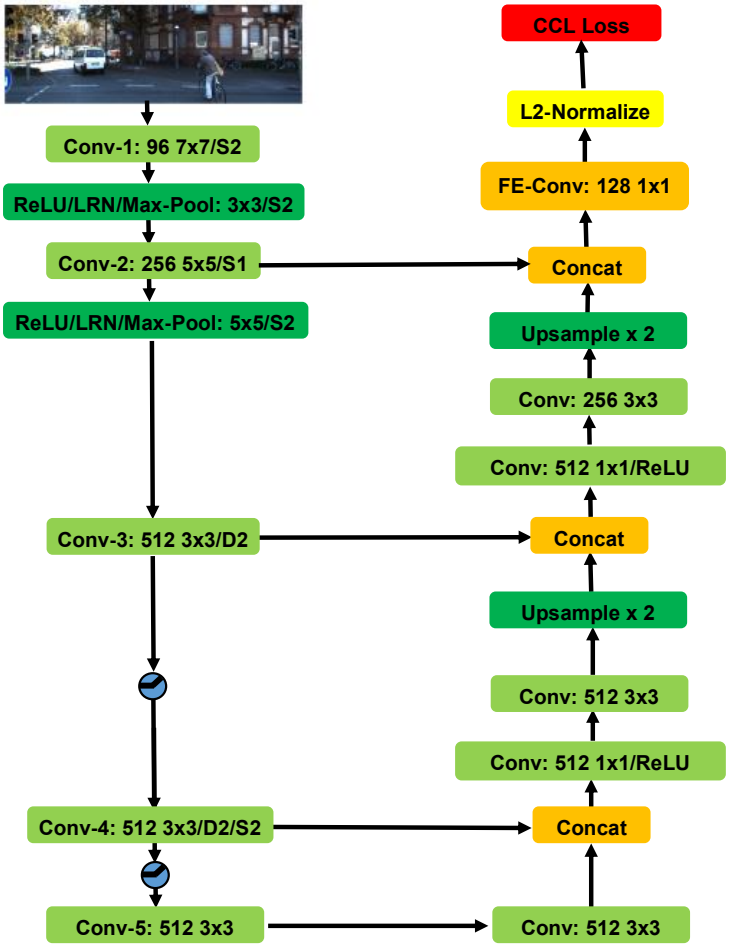
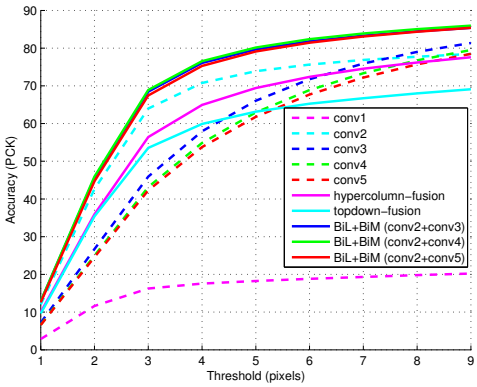


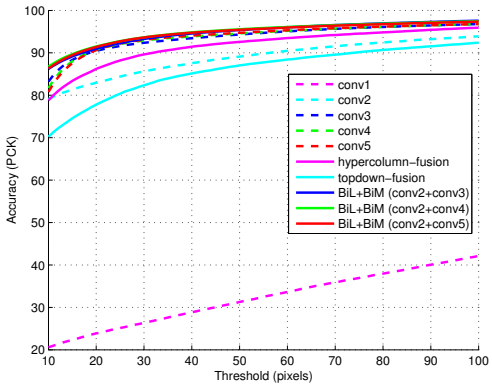
Figure 7.5: One siamese branch of the *topdown-fusion* baseline used in our evaluation.

all layers (*conv1* through *conv5*) are concatenated at each point and the 1x1 convolution is used to extract features to be used for training the CCL loss and for matching. The other approach we considered is the top-down refinement method [99] (namely, *topdown-fusion* shown in Figure 7.5) where refinement modules similar to the ones introduced in [99] are used to refine the top-level *conv5* features gradually down the network by combining with lower level features till the *conv2* layer.

We do the evaluation on the KITTI Flow 2015 dataset where all networks are trained on 80% of the image pairs and the remaining 20% are used for evaluation. During training,

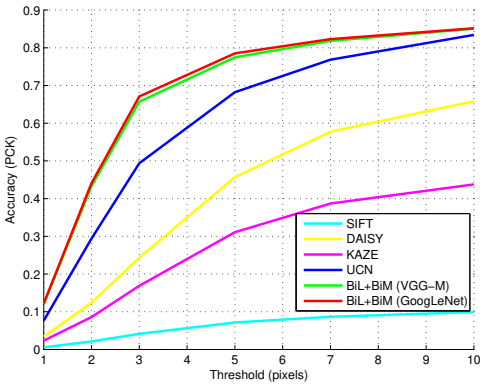


(a) Accuracy over small pixel thresholds

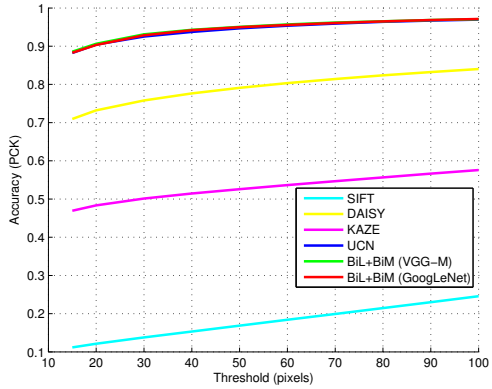


(b) Accuracy over large pixel thresholds

Figure 7.6: PCK performance of the various CNN feature-based methods for correspondence estimation over KITTI Flow 2015.



(a) Accuracy over small thresholds on KITTI Flow 2015



(b) Accuracy over large thresholds on KITTI Flow 2015

Figure 7.7: Comparison results on KITTI Flow 2015.

we randomly choose 1000 positive correspondences from each image pair for training. For a fair comparison, we use the same train-test split for all methods and we train all of them for 50K iterations using the ADAM optimizer. During testing, we use the correspondences $\{(x_i, x_i')\}$ in each image pair (obtained using all non-occluded ground truth flows) for evaluation. Specifically, each method predicts a point $\hat{x}_i'$ in the right image that matches the input point x_i from the left image $\forall i$.

Evaluation Metric: We use the Percentage of Correct Keypoints (PCK) [26, 77] metric in this evaluation. Given a pixel threshold θ, the PCK measures the percentage of predicted points $\hat{x}'_i$ that are within θ pixels or less from the ground truth corresponding point x'_i (and so are considered correct up to T pixels).

Discussion: We plot the PCK curves obtained for all methods under consideration in Figure 7.6 where we split the graph into sub-graphs based on the pixel threshold range. The graphs reveal that, for smaller thresholds, shallower features (*e.g. conv2*) provide higher PCK than deeper ones (*e.g. conv5*), with the exception of *conv1* which performs worst. On the other hand, deeper features provide have better performance for higher thresholds. This suggests that, for best performance, one would need to utilize shallower as well as deeper features produced by the network rather than using just the output of the last layer for correspondence estimation.

The graph also indicates that while baseline approaches for fusing features improve the PCK for smaller thresholds, they do not perform on par with the simple *conv*2-based features. Their performance tends to be bounded by the performances of the component features.

Different variants of our method achieve the highest PCK for smaller thresholds without losing accuracy for higher thresholds. In fact, our method is able to outperform the *conv*2 features although it uses them for refining the rough correspondences estimated by the deeper layers. This is justified by the invariance of the deeper features that are used to establish initial rough estimates of the correspondences which helps to avoid matching patches that have similar appearance but rather belong to different objects.

Hand-crafted Features: We also compare the performance of (a) our BiL+BiM

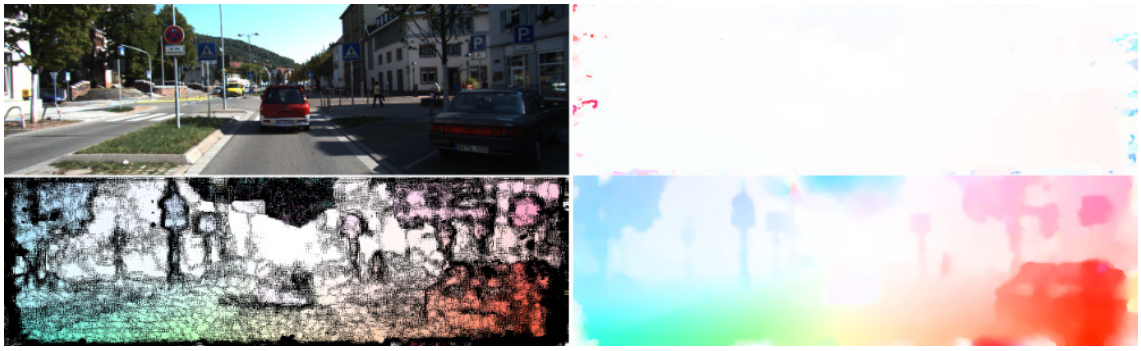


Figure 7.8: Optical flow pipeline. Top left: input image. Top right: initial noisy matches from BiL+BiM. Bottom left: filtered matches after consistency checks and motion constraints. Bottom right: final optical flows after interpolation using EpicFlow [103].

(*conv2+conv5*, VGG-M), (b) the variant of our method based on GoogLeNet/UCN (described in Section 7.3), (c) the original UCN architecture of [26], and (d) the following hand-crafted approaches: SIFT [78], KAZE [4], DAISY [115]. We use the same KITTI Flow 2015 evaluation set utilized in the previous experiment. To evaluate hand-crafted approaches, we use them to compute the descriptors at the input points in the left image and we match the resulting descriptors against the descriptors computed on the right image over a grid of 4 pixel spacing in both directions.

Figure 7.7 compares the resulting PCKs and shows that our method outperforms UCN for smaller thresholds. The difference in performance is not the result of baseline shift since the GoogLeNet variant of our method (which shares the same baseline network as UCN) has similar (or slightly better) performance compared to the VGG-M variant. The graph also indicates the relatively higher invariance of CNN-based descriptors that allow them to obtain a higher percentage of roughly-localized correspondences.

Method	Fl-bg	Fl-fg	Fl-all
FlowNet2 [56]	10.75%	8.75%	10.41%
SDF [8]	8.61%	26.69%	11.62%
SOF [109]	14.63%	27.73%	16.81%
CNN-HPM [9]	18.33%	24.96%	19.44%
SPM-BP [73]	24.06%	24.97%	24.21%
FullFlow [22]	23.09%	30.11%	24.26%
AutoScaler [125]	21.85%	31.62%	25.64%
EpicFlow [103]	25.81%	33.56%	27.10%
DeepFlow2 [129]	27.96%	35.28%	29.18%
PatchCollider [123]	30.60%	33.09%	31.01%
BiL+BiM	23.73%	21.79%	23.41%

Table 7.1: Quantitative results on KITTI Flow 2015 [89]. As per KITTI benchmark convention: 'Fl-bl', 'Fl-fg', and 'Fl-all' represent the outlier percentage on background pixels, foreground pixels, and all pixels respectively.

7.4.3 Optical Flow Experiments

In this section, we demonstrate the application of our geometric correrspondences for obtaining dense optical flows using the KITTI Flow 2015 benchmark [89]. We emphasize that the objective here is not to outperform approaches that have been extensively engineered [8, 9, 56, 109] for optical flows. Rather, we wish to further confirm that our approach is in fact able to compete with the state-of-the-art in low-level matching without employing extensive post-processing operations.

Architecture: For dense optical flow estimation, we leverage GoogLeNet [111] as our backbone architecture. However, at test time, we modify the trained network to obtain dense per-pixel correspondences. To this end: (i) we set the stride to 1 in the first convolutional and pooling layers (*conv1* and *pool1*), (ii) we set the kernel size of the first pooling layer (*pool1*) to 5 instead of 3, (iii) we set the dilation offset of the second convolutional layer (*conv2*) to 4, and (iv) we set the stride of the second pooling layer (*pool2*) to 4. These changes allow us to obtain our shallow feature maps at the same

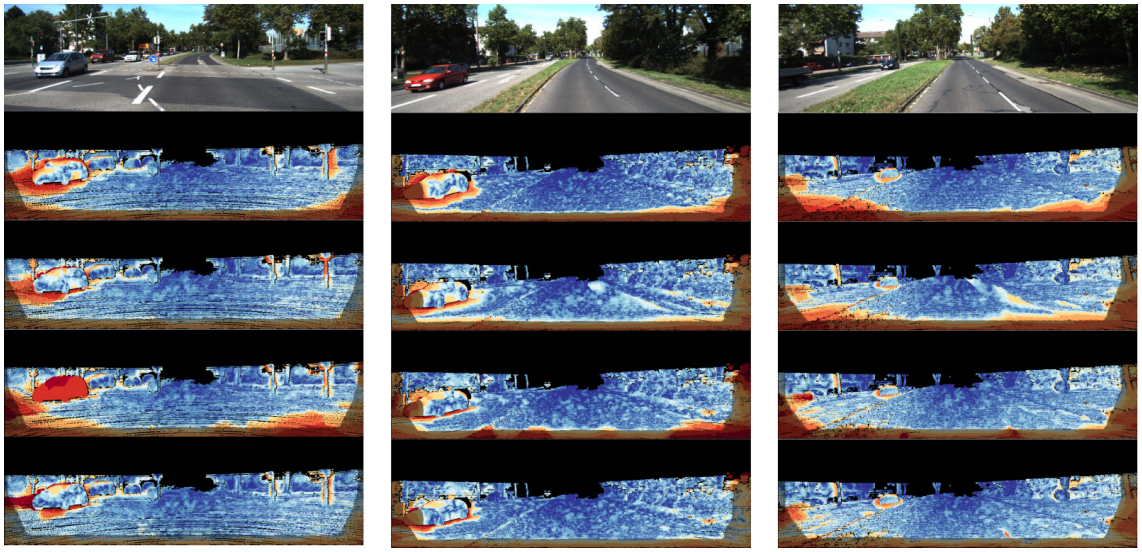


Figure 7.9: Qualitative results on KITTI Flow 2015. First row: input images. Second row: DeepFlow2 [129]. Third row: EpicFlow [103]. Forth row: SPM-BP [73]. Fifth row: BiL+BiM. Red colors mean high errors while blue colors mean low errors.

resolution as the input images (W x H) and the deep feature maps at $W/4$ x $H/4$, and to obtain dense per-pixel correspondences faster and with significantly fewer requirements on the GPU memory as compared to an approach that would process the feature maps at full resolution through all layers of the network.

Procedure: We first extract feature descriptors for every pixel in the source images using the proposed method. These initial matches are usually contaminated by outliers or incorrect matches. Therefore, we follow the protocol of [125] for outlier removal. In particular, we enforce local motion constraints using a window of $[-240, 240]$x$[-240, 240]$ and perform forward-backward consistency checks with a threshold of 0 pixel. These filter matches are then fed to EpicFlow [103] interpolation for producing the final optical flow output. Figure 7.8 illustrates an example of this procedure.

Quantitative evaluation: We tabulate the results of our quantitative evaluation on the KITTI Flow 2015 benchmark in Table 7.1, reporting errors separately for background

pixels (*Fl-bg*) and foreground pixels (*Fl-fg*) as well as for the entire image (*Fl-all*), following [89]. As mentioned earlier, our objective is not necessarily to obtain the best optical flow performance; rather we wish to emphasize that we are able to provide high-quality interest point matches. In fact, many recent works [8, 56, 109] focus on embedding rich domain priors at the level of explicit object classes into their models, which allows them to make good guesses when data is missing, e.g. due to occlusions and truncations and on homogenous surfaces. Yet we are able to outperform all the methods in our comparisons except [56] for foreground pixels (*Fl-fg*, 21.79% errors vs. 24.96%–35.25% excluding 8.75% for [56]). As expected, we do not get as good matches in regions of the image where relatively less structure is present (background, *Fl-bg*), and for such regions methods that employ strong prior models significantly outperform our method. However, even on background regions, we are able to either beat or perform on par with most of our competitors (23.73% vs 21.85%–30.60%) including machinery proposed specifically for optical flows such as [103, 129]). Overall, we obtain better error rates than 6 out of 10 of the state-of-the-art methods evaluated (*Fl-all*).

Qualitative evaluation: We visualize qualitative results over several test images in Figure 7.9, to contrast DeepFlow2 [129], EpicFlow [103], and SPM-BP [73] against our method. As expected from the earlier discussion, we observe superior results for our method on the image regions belonging to the vehicles, because of strong local structures, whereas for instance in first column (fourth row) SPM-BP [73] entirely fails on the blue car. We observe errors in the estimates of our method largely in regions which are occluded (surroundings of other cars) or truncated (lower portion of the images), where the competing methods visualized here also have high errors.

7.5 Summary

We draw inspiration from recent studies [143, 148] as well as conventional wisdom about CNN architectures to enhance learned representations for geometric matching. Convolutional network architectures naturally learn hierarchies of features, thus, a contrastive loss applied at a deep layer will return features that are less sensitive to local image structure. We propose to remedy this by employing features at multiple levels of the feature hierarchy for interest point description. Further, we leverage recent ideas in deep supervision to explicitly obtain task-relevant features at intermediate layers. Finally, we exploit the receptive field growth for increasing layer depths as a proxy to replace conventional coarse-to-fine image pyramid approaches for matching. We thoroughly evaluate these ideas realized as concrete network architectures, on challenging benchmark datasets. Our evaluation on the task of explicit keypoint matching outperforms hand-crafted descriptors, a state-of-the-art descriptor learning approach [26], as well as various ablative baselines including hypercolumns and top-down-fusion. Further, a preliminary evaluation for optical flow computation outperforms several competing methods even without extensive engineering or leveraging higher-level semantic scene understanding.

Chapter 8

Directions for Future Work

8.1 Overview

Convolutional Neural Networks have achieved, over the past few years, unprecedented performance in many computer vision tasks, redefining the state-of-the-art for these tasks. While we have explored different ways of integrating CNNs for feature learning in image set classification and local feature description, there are other practical issues that need to be addressed for more effective utilization of CNNs. In this chapter, we explore a few such issues which may need further research to handle them.

8.2 CNNs for Local Feature Description And Semantic Segmentation

While local feature description for correspondence estimation and semantic segmentation are two different visual tasks, the two problems share the need to densely process images

and the networks the CNN approaches to both problems share some structural properties. For example, fully convolutional CNNs are popular for the two problems [26, 48, 99, 137] and dilated convolution [26, 139]. They do also share the relatively high computational burden, in terms of speed and memory, during training and testing, as they need to densely process the input images.

Some practical scenarios, like autonomous driving, require the solution of both problems simultaneously and in real time. It is thus important to find cheaper and more computationally efficient means to solve these problems. One direction would be to find a common network architecture to solve both problems simultaneously and efficiently, utilizing the fact that the networks for both problems have many similarities. An interesting problem would be finding a technique to supervise the network for both tasks and can still perform as well as or even better in each task than a separately trained network for that particular task.

8.3 Supervising Metric Learning for Correspondence Estimation

Existing approaches for supervising correspondence estimation CNNs rely on siamese architectures and consider for each example a pair of positive (corresponding) points from the two input images in addition to a number of negatives [125] or one hard negative [26]. However, in many scenarios, the same point might be observed in more than two images and so, in principle, we can form examples that contain more than two positive points.

The use of such track-based examples in addition to or instead of pair-based examples could potentially enhance the robustness of the supervision and improve the quality of learned features. It would be interesting to find loss functions that could effectively utilize such track-based examples. It is also interesting to find more efficient ways to process such examples without increasing the number of images that have to be processed by the network to produce descriptors for all the points in the example.

Bibliography

[1] http://ibug.doc.ic.ac.uk/resources/drmf-matlab-code-cvpr-2013/. URL http://ibug.doc.ic.ac.uk/resources/drmf-matlab-code-cvpr-2013/. 22

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Commun. ACM*, 54 (10):105–112, 2011. 106

[3] Michal Aharon, Michael Elad, and Alfred Bruckstein. k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Processing*, 54(11):4311–4322, 2006. 67

[4] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *Proc. European Conf. on Comput. Vision (ECCV)*, 2012. 119

[5] David F Andrews and Colin L Mallows. Scale mixtures of normal distributions. *J. of Royal Stat. Soc. B*, pages 99–102, 1974. 30, 42

[6] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM j. on matrix analysis and applications*, 29(1):328–347, 2007. 56, 64

[7] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with constrained local models. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 3444–3451, 2013. viii, 22, 44

[8] M. Bai, W. Luo, K. Kundu, and R. Urtasun. Exploiting Semantic Information and Deep Matching for Optical Flow. In *Proc. European Conf. on Comput. Vision (ECCV)*, 2016. 120, 122

[9] C. Bailer, K. Varanasi, and D. Stricker. CNN-based Patch Matching for Optical Flow with Thresholded Hinge Embedding Loss. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2017. 120

[10] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2017. 106

[11] Ankan Bansal, Carlos Castillo, Rajeev Ranjan, and Rama Chellappa. The do's and don'ts for cnn-based face verification. *arXiv preprint arXiv:1705.07426*, 2017. ix, xii, 83, 84

[12] Ronen Basri and David W Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(2):218–233, 2003. viii, 2, 3

[13] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. In *Proc. European Conf. on Comput. Vision (ECCV)*, 2006. 106

[14] Peter N. Belhumeur, João P Hespanha, and David Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(7):711–720, 1997. 16, 25

[15] S. Belongie, J. Malik, and J. Puzicha. Shape Context: A new descriptor for shape matching and object recognition. In *Advances in Neural Info. Processing Sys. (NIPS)*, 2001. 106

[16] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006. 33, 34, 35, 40

[17] Hakan Cevikalp and Bill Triggs. Face recognition based on image sets. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 2567–2573, 2010. 1, 9, 10, 16, 25, 42, 43, 71, 84, 97

[18] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. 43, 80, 84

[19] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. British Machine Vision Conf. (BMVC)*, 2014. 111

[20] Liang Chen. Dual linear regression based classification for face cluster recognition. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 2673–2680, 2014. 1

[21] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Machine Intell.*, 2017. 111

[22] Q. Chen and V. Koltun. Full Flow: Optical Flow Estimation by Global Optimization over Regular Grids. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2016. 120

[23] Shaokang Chen, Conrad Sanderson, Mehrtash T Harandi, and Brian C Lovell. Improved image set classification via joint sparse approximated nearest subspaces. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 452–459, 2013. 1, 9, 10

[24] Yi-Chen Chen, Vishal M Patel, P Jonathon Phillips, and Rama Chellappa. Dictionary-based face recognition from video. In *Proc. European Conf. on Comput. Vision (ECCV)*, pages 766–779. 2012. 1, 2, 9, 10, 16, 25, 43, 71, 97

[25] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*. 2005. 107, 109, 112

[26] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal Correspondence Network. In *Advances in Neural Info. Processing Sys. (NIPS)*. 2016. x, 4, 103, 104, 106, 107, 109, 111, 112, 118, 119, 123, 125

[27] J. Czarnowski, S. Leutenegger, and A. J. Davison. Semantic texture for robust dense tracking. In *Int'l Conf. on Comput. Vision Workshops (ICCVW)*, 2017. 104, 108, 113

[28] N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2005. 106

[29] M.O. Derawi, C. Nickel, P. Bours, and C. Busch. Unobtrusive user-authentication on mobile phones using biometric gait recognition. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 306–311, Oct 2010. 16

[30] A. Dosovitskiy, P. Fischer, E. Ilg, P. Husser, C. Hazirbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *Int'l Conf. on Comput. Vision (ICCV)*, 2015. 107

[31] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The J. of Machine Learning Research (JMLR)*, 12(Jul):2121–2159, 2011. 89

[32] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Machine Intell.*, 35(11):2765–2781, 2013. 39

[33] Kjersti Engan, Sven Ole Aase, and JH Husoy. Frame based signal compression using method of optimal directions (mod). In *Proc. IEEE Int'l Symposium on Circuits and Systems*, volume 4, pages 1–4, 1999. 67

[34] Kamran Etemad and Rama Chellappa. Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A*, 14:1724–1733, 1997. 16, 25

[35] Mohammed E Fathy, Vishal M Patel, and Rama Chellappa. Face-based active authentication on mobile devices. In *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1687–1691, 2015. 46

[36] Mohammed E Fathy, Azadeh Alavi, and Rama Chellappa. Discriminative log-euclidean feature learning for sparse representation-based recognition of faces from videos. In *Proc. Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, pages 3359–3367, 2016. 52, 95, 97

[37] Tao Feng, Ziyi Liu, Kyeong-An Kwon, Weidong Shi, Bogdan Carbunar, Yifei Jiang, and Nhung Nguyen. Continuous mobile authentication using touchscreen gestures. In *Proc. IEEE Int'l Symposium on Technologies for Homeland Security (HST)*, pages 451–456, 2012. 16

[38] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Trans. on information forensics and security*, 8(1): 136–148, 2013. 16

[39] David Gadot and Lior Wolf. PatchBatch: A Batch Augmented Loss for Optical Flow. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2016. 106

[40] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. Int'l Workshop on Artificial Intelligence and Stat. (AISTATS)*, 2010. 114

[41] M Gunther, Artur Costa-Pazo, Changxing Ding, Elhocine Boutellaa, Giovani Chiachia, Honglei Zhang, Marcus de Assis Angeloni, Vitomir Struc, Elie Khoury, Esteban Vazquez-Fernandez, et al. The 2013 face recognition evaluation in mobile environment. In *International Conference on Biometrics (ICB)*, pages 1–7, 2013. 17

[42] Kai Guo, Prakash Ishwar, and Janusz Konrad. Action recognition using sparse representation on covariance manifolds of optical flow. In *Int'l Conf. Advanced Video and Signal Based Surveillance (AVSS)*, pages 188–195, 2010. 13, 52, 53

[43] Jihun Hamm and Daniel D Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proc. Int'l Conf. on Machine Learning (ICML)*, pages 376–383, 2008. 9, 11

[44] Mehrtash Harandi, Conrad Sanderson, Chunhua Shen, and Brian C Lovell. Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution. In *Int'l Conf. on Comput. Vision (ICCV)*, pages 3120–3127, 2013. 1, 11, 13

[45] Mehrtash Harandi, Mathieu Salzmann, and Mahsa Baktashmotlagh. Beyond gauss: Image-set matching on the riemannian manifold of pdfs. In *Int'l Conf. on Comput. Vision (ICCV)*, pages 4112–4120, 2015. 12

[46] Mehrtash T Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C Lovell. Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 2705–2712, 2011. 1, 9, 11

[47] Mehrtash T Harandi, Conrad Sanderson, Richard Hartley, and Brian C Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In *Proc. European Conf. on Comput. Vision (ECCV)*, pages 216–229, 2012. 13, 55

[48] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2015. 7, 105, 115, 125

[49] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference (AVC)*, 1988. 4

[50] Munawar Hayat, Mohammed Bennamoun, and Senjian An. Learning non-linear reconstruction models for image set classification. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 1915–1922, 2014. xi, 1, 9, 14, 43, 47, 71, 97

[51] Munawar Hayat, Mohammed Bennamoun, and Senjian An. Reverse training: An efficient approach for image set classification. In *Proc. European Conf. on Comput. Vision (ECCV)*, pages 784–799. 2014. 1, 9

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 95, 96

[53] Yiqun Hu, Ajmal S Mian, and Robyn Owens. Sparse approximated nearest points for image set classification. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 121–128, 2011. 1, 9, 10, 16, 25, 43, 71, 97

[54] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 140–149, 2015. 9, 11, 71

[55] 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 *Proc. Int'l Conf. on Machine Learning (ICML)*, pages 720–729, 2015. 1, 9, 11, 12, 43, 52, 53, 54, 59, 71, 97

[56] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2017. 107, 120, 122

[57] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int'l Conf. on Machine Learning (ICML)*, pages 448–456, 2015. 96

[58] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Info. Processing Sys. (NIPS)*. 2015. 107

[59] Shihao Ji and Lawrence Carin. Bayesian compressive sensing and projection optimization. In *Proc. Int'l Conf. on Machine Learning (ICML)*, pages 377–384, 2007. 30

[60] Shihao Ji, Ya Xue, and Lawrence Carin. Bayesian compressive sensing. *IEEE Trans. Signal Processing*, 56(6):2346–2356, 2008. 30

[61] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM-Multimedia*, 2014. 114

[62] Minyoung Kim, Sanjiv Kumar, Vladimir Pavlovic, and Henry Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 44

[63] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proc. Int'l Conf. on Learning Representations (ICLR)*, 2014. 114

[64] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-Supervised Nets. *Proc. Int'l Workshop on Artificial Intelligence and Stat. (AISTATS)*, 2015. 104, 108, 110

[65] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in Neural Info. Processing Sys. (NIPS)*, pages 801–808, 2007. 67

[66] Kuang-Chih Lee and David Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, volume 1, pages 852–859, 2005. 16

[67] Kuang-Chih Lee, Jeffrey Ho, Ming-Hsuan Yang, and David Kriegman. Visual tracking and recognition using probabilistic appearance manifolds. *Comput. Vision and Image Understanding*, 99(3):303–331, 2005. 16

[68] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *Int'l Conf. on Comput. Vision (ICCV)*, 2011. 106

[69] Michael S Lewicki and Terrence J Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000. 67

[70] Chi Li, M Zeeshan Zia, Quoc-Huy Tran, Xiang Yu, Gregory D Hager, and Manmohan Chandraker. Deep supervision with shape concepts for occlusion-aware 3d object parsing. 2017. 104, 108, 110

[71] Peihua Li, Qilong Wang, and Lei Zhang. A novel earth mover's distance methodology for image matching with gaussian mixture models. In *Int'l Conf. on Comput. Vision (ICCV)*, pages 1689–1696, 2013. 12

[72] Peihua Li, Qilong Wang, Wangmeng Zuo, and Lei Zhang. Log-euclidean kernels for sparse representation and dictionary learning. In *Int'l Conf. on Comput. Vision (ICCV)*, pages 1601–1608, 2013. 13

[73] Y. Li, D. Min, M. S. Brown, M. N. Do, and J. Lu. SPM-BP: Sped-up PatchMatch Belief Propagation for Continuous MRFs. In *Int'l Conf. on Comput. Vision (ICCV)*, 2015. x, 120, 121, 122

[74] K. Lin, J. Lu, C. S. Chen, and J. Zhou. Learning Compact Binary Descriptors with Unsupervised Deep Neural Networks. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2016. 106

[75] T. Lindeberg. Feature detection with automatic scale selection. *Int'l J. Comput. Vision*, 30(2):79–116, 1998. 4

[76] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2015. 109

[77] J. L. Long, N. Zhang, and T. Darrel. Do Convnets Learn Correspondence? In *Advances in Neural Info. Processing Sys. (NIPS)*, 2014. 106, 118

[78] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. Comput. Vision*, 60(2):91–110, 2004. 105, 106, 119

[79] Jiwen Lu, Gang Wang, Weihong Deng, and Pierre Moulin. Simultaneous feature and dictionary learning for image set based face recognition. In *Proc. European Conf. on Comput. Vision (ECCV)*, pages 265–280. 2014. 1, 10

[80] Jiwen Lu, Gang Wang, Weihong Deng, Pierre Moulin, and Jie Zhou. Multi-manifold deep metric learning for image set classification. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 1137–1145, 2015. 1, 9, 14

[81] B. D. Lucas and T. Kanade. Optical navigation by the method of differences. In *Proc. Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, 1985. 108, 113

[82] David J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1991. 33, 34

[83] David J.C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11(5):1035–1068, 1999. 34

[84] Arif Mahmood and Ajmal Mian. Hierarchical sparse spectral clustering for image set classification. In *Proc. British Machine Vision Conf. (BMVC)*, pages 1–11, 2012. 1

[85] Arif Mahmood, Ajmal Mian, and Robyn Owens. Semi-supervised spectral clustering for image set classification. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 121–128, 2014. 1, 9, 10

[86] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *The J. of Machine Learning Research (JMLR)*, 11(Jan):19–60, 2010. 67, 96, 97

[87] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. Face detection without bells and whistles. In *Proc. European Conf. on Comput. Vision (ECCV)*, volume 4, pages 720–735, 2014. 21

[88] Chris McCool and Sébastien Marcel. Mobio database for the icpr 2010 face and speech competition. Technical report, Idiap, 2009. 17

[89] M. Menze and A. Geiger. Object Scene Flow for Autonomous Vehicles. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2015. xiii, 120, 122

[90] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31:1147–1163, 2015. 106

[91] D. Nister, O. Naroditsky, and J. Bergen. Visual Odometry. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2004. 106

[92] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997. 67

[93] Enrique G Ortiz, Alan Wright, and Mubarak Shah. Face recognition in movie trailers via mean sequence sparse representation-based classification. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 3531–3538, 2013. 1, 2, 9, 13, 16, 25, 30, 37, 43, 53, 71, 84, 89, 96, 97

[94] Alice J OToole, P Jonathon Phillips, Samuel Weimer, Dana A Roark, Julianne Ayyad, Robert Barwick, and Joseph Dunlop. Recognizing people from dynamic and static faces and bodies: Dissecting identity with a fusion approach. *Vision Research*, 51(1):74–83, 2011. 16

[95] Yanwei Pang, Yuan Yuan, and Xuelong Li. Gabor-based region covariance matrices for face recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, 18 (7):989–993, 2008. 55

[96] Trevor Park and George Casella. The bayesian lasso. *J. of American Stat. Assoc.*, 103(482):681–686, 2008. 30, 42

[97] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *Proc. British Machine Vision Conf. (BMVC)*, 2015. ix, xii, 79, 80

[98] P Jonathon Phillips, Patrick J Flynn, J Ross Beveridge, W Todd Scruggs, Alice J O'Toole, David Bolme, Kevin W Bowyer, Bruce A Draper, Geof H Givens, Yui Man Lui, et al. Overview of the multiple biometrics grand challenge. In *International Conference on Advances in Biometrics*, pages 705–714, 2009. 16

[99] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *Proc. European Conf. on Comput. Vision (ECCV)*, 2016. 7, 105, 115, 116, 125

[100] Abena Primo, Vir V Phoha, Rajesh Kumar, and Abdul Serwadda. Context-aware active authentication using smartphone accelerometer measurements. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 98–105, 2014. 16

[101] Qiang Qiu and Guillermo Sapiro. Learning transformations for clustering and classification. *The J. of Machine Learning Research (JMLR)*, 16(1):187–225, 2015. 12

[102] M. Rad and V. Lepetit. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth. In *Int'l Conf. on Comput. Vision (ICCV)*, 2017. 4

[103] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2015. x, 107, 119, 120, 121, 122

[104] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int'l J. Comput. Vision*, 115(3):211–252, 2015. 111

[105] R. B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D registration. In *Proc. Int'l Conf. on Robotics and Automation (ICRA)*, 2009. 106

[106] Swami Sankaranarayanan, Azadeh Alavi, Carlos D Castillo, and Rama Chellappa. Triplet probabilistic embedding for face verification and clustering. In *IEEE Int'l Conf. on Biometrics Theory, Applications and Systems (BTAS)*, 2016. ix, xii, 83, 84

[107] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative Evaluation of Hand-Crafted and Learned Local Features. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2017. 106

[108] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 93

[109] L. Sevilla-Lara, D. Sun, V. Jampani, and M. J. Black. Optical Flow with Semantic Segmentation and Localized Layers. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2016. 120, 122

[110] Ilya Sutskever, James Martens, George E Dahl, and Geoffrey E Hinton. On the importance of initialization and momentum in deep learning. *Proc. Int'l Conf. on Machine Learning (ICML)*, pages 1139–1147, 2013. 89

[111] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2015. x, 111, 112, 120

[112] Michael E Tipping. The relevance vector machine. In *Advances in Neural Info. Processing Sys. (NIPS)*, pages 652–658, 2000. 29, 33

[113] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *The J. of Machine Learning Research (JMLR)*, 1:211–244, 2001. 29, 33, 34, 35

[114] Michael E Tipping and Anita C Faul. Fast marginal likelihood maximisation for sparse bayesian models. In *Proc. Int'l Workshop on Artificial Intelligence and Stat. (AISTATS)*, 2003. 33, 35, 36, 39

[115] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal. Machine Intell.*, 32(5): 815–830, 2010. 119

[116] Lorenzo Torresani and Kuang-chih Lee. Large margin component analysis. In *Advances in Neural Info. Processing Sys. (NIPS)*, pages 1385–1392, 2007. 79

[117] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 586–591, 1991. 16, 25

[118] Raviteja Vemulapalli and David W Jacobs. Riemannian metric learning for symmetric positive definite matrices. *arXiv preprint arXiv:1501.02393*, 2015. 12

[119] Paul Viola and Michael J Jones. Robust real-time face detection. *Int'l J. Comput. Vision*, 57(2):137–154, 2004. 21, 44

[120] Ruiping Wang and Xilin Chen. Manifold discriminant analysis. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 429–436, 2009. 1, 9, 11

[121] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based on image set. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 1, 9, 11

[122] Ruiping Wang, Huimin Guo, Larry S Davis, and Qionghai Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 2496–2503, 2012. 1, 9, 11, 12, 43, 52, 54, 59, 97

[123] S. Wang, S. Fanello, C. Rhemann, S. Izadi, and P. Kohli. The Global Patch Collider. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2016. 120

[124] S. Wang, R. Clark, H. Wen, and N. Trigoni. DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. In *Proc. Int'l Conf. on Robotics and Automation (ICRA)*, 2017. 4

[125] Shenlong Wang, Linjie Luo, Ning Zhang, and Jia Li. AutoScaler: Scale-Attention Networks for Visual Correspondence. In *Proc. British Machine Vision Conf. (BMVC)*, 2017. 4, 103, 107, 120, 121, 125

[126] Wen Wang, Ruiping Wang, Zhiwu Huang, Shiguang Shan, and Xilin Chen. Discriminant analysis on riemannian manifold of gaussian distributions for face recognition with image sets. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 2048–2057, 2015. 1, 9, 12, 52, 54

[127] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *The J. of Machine Learning Research (JMLR)*, 10:207–244, 2009. 25, 79

[128] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Info. Processing Sys. (NIPS)*, pages 1473–1480, 2005. 93

[129] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Int'l Conf. on Comput. Vision (ICCV)*, 2013. x, 106, 120, 121, 122

[130] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K. Jain, James A. Duncan, Kristen Allen, Jordan Cheney, and Patrick Grother. Iarpa janus benchmark-b face dataset. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition Workshops (CVPRW)*, 2017. ix, 82, 84

[131] David P Wipf and Bhaskar D Rao. Sparse bayesian learning for basis selection. *Signal Processing, IEEE Transactions on*, 52(8):2153–2164, 2004. 30

[132] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, pages 529–534, 2011. 44

[133] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Machine Intell.*, 31(2):210–227, 2009. viii, 2, 4, 13, 16, 25, 31, 32, 40, 64, 89

[134] Chunyan Xu, Canyi Lu, Junbin Gao, Wei Zheng, Tianjiang Wang, and Shuicheng Yan. Discriminative analysis for symmetric positive definite matrices on lie groups. *IEEE Trans. on Circuits and Systems for Video Technology*, 25(10):1576–1585, 2015. 12, 59

[135] T. Y. Yang, J. H. Hsu, Y. Y. Lin, and Y. Y. Chuang. DeepCD: Learning Deep Complementary Descriptors for Patch Representations. In *Int'l Conf. on Comput. Vision (ICCV)*, 2017. 106

[136] Florian Yger and Masashi Sugiyama. Supervised logeuclidean metric learning for symmetric positive definite matrices. *arXiv preprint arXiv:1502.03505*, 2015. 12

[137] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. In *Proc. European Conf. on Comput. Vision (ECCV)*, 2016. 4, 103, 106, 125

[138] F. Yu and V. Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *Proc. Int'l Conf. on Learning Representations (ICLR)*. 2016. 111, 112

[139] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proc. Int'l Conf. on Learning Representations (ICLR)*, 2016. 125

[140] Chunfeng Yuan, Weiming Hu, Xi Li, Stephen Maybank, and Guan Luo. Human action recognition under log-euclidean riemannian metric. In *Proc. Asian Conf. on Comput. Vision (ACCV)*, pages 343–353. 2010. 13, 52, 53

[141] A. R. Zamir, T.-L. Wu, L. Sun, W. Shen, B. E. Shi, J. Malik, and S. Savarese. Feedback Networks. 2017. 108

[142] J. Zbontar and Y. LeCun. Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. *The J. of Machine Learning Research (JMLR)*, 17:1–32, 2016. 106

[143] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. European Conf. on Comput. Vision (ECCV)*, 2014. 103, 106, 108, 110, 113, 123

[144] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2017. 4, 103, 106

[145] Lei Zhang, Meng Yang, and Xiangchu Feng. Sparse representation or collaborative representation: Which helps face recognition? In *Int'l Conf. on Comput. Vision (ICCV)*, pages 471–478, 2011. 5, 13, 43, 48, 71, 84

[146] Xu Zhang, Felix X. Yu, Sanjiv Kumar, and Shih-Fu Chang. Learning Spread-out Local Feature Descriptors. In *Int'l Conf. on Comput. Vision (ICCV)*, 2017. 106

[147] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995. 4

[148] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object Detectors Emerge in Deep Scene CNNs. In *Proc. Int'l Conf. on Learning Representations (ICLR)*, 2015. 103, 108, 110, 113, 123

[149] Pengfei Zhu, Lei Zhang, Wangmeng Zuo, and Dejing Zhang. From point to set: Extend the learning of distance metrics. In *Int'l Conf. on Comput. Vision (ICCV)*, pages 2664–2671, 2013. 9, 43, 71, 97

[150] Pengfei Zhu, Wangmeng Zuo, Lei Zhang, Simon Chi-Keung Shiu, and Dejing Zhang. Image set-based collaborative representation for face recognition. *IEEE Trans. on information forensics and security*, 9(7):1120–1132, 2014. 13