SIP (2021), vol. 10, e1, page 1 of 22 © The Author(s), 2021. Published by Cambridge University Press.. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution, and reproduction in any medium, provided the original work is properly cited.
doi:10.1017/ATSIP.2020.27

OVERVIEW PAPER

Subspace learning for facial expression recognition: an overview and a new perspective

CIGDEM TURAN,[1,2] RUI ZHAO,[1] KIN-MAN LAM[1] AND XIANGJIAN HE[3]

For image recognition, an extensive number of subspace-learning methods have been proposed to overcome the high-dimensionality problem of the features being used. In this paper, we first give an overview of the most popular and state-of-the-art subspace-learning methods, and then, a novel manifold-learning method, named soft locality preserving map (SLPM), is presented. SLPM aims to control the level of spread of the different classes, which is closely connected to the generalizability of the learned subspace. We also do an overview of the extension of manifold learning methods to deep learning by formulating the loss functions for training, and further reformulate SLPM into a soft locality preserving (SLP) loss. These loss functions are applied as an additional regularization to the learning of deep neural networks. We evaluate these subspace-learning methods, as well as their deep-learning extensions, on facial expression recognition. Experiments on four commonly used databases show that SLPM effectively reduces the dimensionality of the feature vectors and enhances the discriminative power of the extracted features. Moreover, experimental results also demonstrate that the learned deep features regularized by SLP acquire a better discriminability and generalizability for facial expression recognition.

Keywords: Subspace learning, Facial expression recognition, Deep learning

Received 3 August 2020; Revised 13 December 2020

I. INTRODUCTION

Dimensionality reduction, which aims to find the distinctive features to represent high-dimensional data in a low-dimensional subspace, is a fundamental problem in classification. Many real-world computer-vision and pattern-recognition applications, e.g. facial expression recognition, involve large volumes of high-dimensional data. Subspace analysis is an effective method to handle the high-dimensional data, and serves two important tasks. The first one is the dimensionality reduction, which makes the original data easier to visualize and analyze. The second task is for manifold learning, with the high-dimensional data being projected into a lower-dimensional manifold representation. According to Boufounos *et al.* [1], dimensionality reduction techniques contribute significantly to various industrial applications, by reducing the time complexity of the algorithms and improving the semantic intensity of the visual features. As an effective approach for dimensionality reduction, subspace learning has been widely studied in the literature for learning a low-dimensional space to describe the high-dimensional data, while preserving their structure. Principal component analysis (PCA) [2, 3] and linear discriminant analysis (LDA) [3, 4] are two notable linear methods for subspace learning. PCA aims to find principal projection vectors, which are those eigenvectors associated with the largest eigenvalues of the covariance matrix of training samples, to project the high-dimensional data to a low-dimensional subspace. Unlike PCA, which is an unsupervised method that considers common features of training samples, LDA employs the Fisher criteria to maximize the between-class scattering and to minimize the within-class scattering, so as to increase the discriminative power of the learned low-dimensional features. Although LDA is superior to PCA for pattern recognition, it suffers from the small-sample-size (SSS) problem [5] because the number of training samples available is much smaller than the dimension of the feature vectors (FVs) in most of the real-world applications. To overcome the SSS problem, Li *et al.* [6] proposed the maximum margin criterion (MMC) method, which utilizes the difference between the within-class and the between-class scatter matrices as the objective function. In [7], it is shown that intra-class scattering has an important effect when dealing with overfitting in training a model. Unlike the conventional wisdom, too much compactness within each class decreases the generalizability of the manifolds. Since LDA and MMC are too "harsh," they need to be softened. Liu *et al.* [7] proposed

[1]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[2]The Department of Computer Science, Technical University of Darmstadt, Darmstadt, Germany
[3]Computer Science, School of Electrical and Data Engineering, University of Technology, Sydney, Australia

Corresponding author:
C. Turan
Email: cigdem.turan@connect.polyu.hk

the soft discriminant map (SDM), which tries to control the spread of the different classes. MMC can be considered as a special case of SDM, where the softening parameter $\beta = 1$.

Linear methods, such as PCA, LDA, and SDM, may fail to find the underlying nonlinear structure of the data under consideration, and they may lose some discriminant information of the manifolds during the linear projection. To overcome this problem, some nonlinear dimensionality reduction techniques have been proposed. In general, the techniques can be divided into two categories: kernel-based and manifold-learning-based approaches. Kernel-based methods, as well as the linear methods mentioned above, only employ the global structure while ignoring the local geometry of the data. However, manifold-learning-based methods can explore the intrinsic geometry of the data. Popular nonlinear manifold-learning methods include ISOMAP [8], locally linear embedding (LLE) [9], and Laplacian eigenmaps [10], which can be considered as special cases of the general framework for dimensionality reduction named "graph embedding" [11]. Although these methods can represent the local structure of the data, they suffer from the out-of-sample problem. Locality preserving projection (LPP) [12] was proposed as a linear approximation of the nonlinear Laplacian eigenmaps [10] to overcome the out-of-sample problem. LPP considers the manifold structure via the adjacency graph. The manifold-learning methods presented so far are based on unsupervised learning, i.e. they do not consider the class information. Several supervised-based methods [13–15] have been proposed, which utilize the discriminant structure of the manifolds. Marginal Fisher analysis (MFA) [11] uses the Fisher criterion and constructs two adjacency graphs to represent the within-class and the between-class geometry of the training data. Several other methods have been proposed with similar ideas, such as locality-preserved maximum information projection (LPMIP) [16], constrained maximum variance mapping (CMVM) [17], and locality sensitive discriminant analysis (LSDA) [18]. Quite recently, more effective graph-construction methods [19, 20] have been investigated, which show great potential in manifold learning. Jia *et al.* [19] presented a joint learning framework to construct clustering-aware graphs. They further enhanced their framework in [20]. In real-life applications, unlabeled data exist because of various reasons. To deal with this problem, various semi-supervised learning algorithms have also been proposed [21–23]. Jia *et al.* [24] presented the graph-Laplacian principal component analysis (GL-PCA), which uses weak supervision to capture both local and global data structures.

Although the above-mentioned subspace-learning methods have demonstrated promising performance by increasing the discriminative power of the learned features, they fail to penalize the between-class distance in local data structure, when learning manifolds. Those methods also cannot exhibit a similar performance on testing data, which leads to the poor generalization ability of the learned manifolds in real-world applications. In this paper, we will first give an overview of subspace-learning methods, and then, propose a new graph-based method to solve the generalization problem of the existing subspace-learning methods by extending their merits to form a better method. The major novelties of the proposed method, named "soft locality-preserving map (SLPM)," can be outlined as follows:

(i) SLPM constructs a within-class graph matrix and a between-class graph matrix using the k-nearest neighborhood and the class information to discover the local geometry of the data.
(ii) To overcome the SSS problem and to decrease the computational cost of computing the inverse of a matrix, SLPM defines its objective function as the difference between the between-class and the within-class Laplacian matrices.
(iii) Inspired by the idea of SDM on the importance of the intra-class spread, a parameter β is added to control the penalty on the within-class Laplacian matrix so as to avoid the overfitting problem and to increase the generalizability of the underlying manifold.

To improve the generalizability of the manifolds generated by the subspace-analysis methods, more training samples, which are located near the boundaries of the respective classes, are desirable. In this paper, we apply our proposed SLPM method to facial expression recognition, and propose an efficient way to enhance the generalizability of the manifolds of the different expression classes by feature augmentation or generation. An expression video sequence, which ranges from a neutral-expression face to the highest intensity of an expression, allows us to select appropriate samples for learning a better and more representative manifold for the expression classes. For the optimal manifold of an expression class, its center should represent those samples that best represent the facial expression concerned, i.e. those expression face images with the highest intensities. When moving away from the manifold center, the corresponding expression intensity should be reducing. Those samples near the boundary of a manifold are important for describing the expression, which also defines the shape of the manifold. To describe a manifold boundary, images with low-intensity expressions should be considered. Since the FVs used to represent facial expressions usually have high dimensionality, many training samples near the manifold boundary are required, so as to represent it completely. However, we usually have a limited number of weak-intensity expression images, so feature generation is necessary to learn more complete manifolds.

With the rapid development of deep learning technologies, more and more attention has been paid to the generation of the learned deep features. The convolutional neutral network (CNN) is one of the most powerful deep learning techniques to learn discriminative representations for facial expressions [25]. However, CNN-based methods, learned via logistic regression, generally suffer from poor discriminability and generalization in real-world scenarios. To address these issues, deep subspace learning methods

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

have been widely studied in recent years to enhance the discriminative and generalization ability of the learned deep features from CNNs [26–28]. In this paper, we describe the extension of LPP to deep learning, and then formulate the proposed SLPM algorithm for deep learning as well, so more discriminative deep features of facial expressions can be learned. Specifically, we employ LPP or SLPM as an additional regularization term in the objective function for training the deep models. Extensive experiments show that the regularization effect from SLPM contributes to a more robust deep model with better generalization for the real-world facial-expression recognition task.

In other applications, additional samples have also been generated for manifold learning. In [29], faces are morphed between two people with different percentages so as to generate face images near the manifold boundaries. By generating more face images and extracting their FVs, the manifold for each face subject can be learned more accurately. Therefore, the decision region for each subject can be determined for watch-list surveillance. In our algorithm, rather than morphing faces and extracting features from the synthesized face, we propose generating features for low-intensity expressions directly in the feature domain. Generating features in this way should be more accurate than extracting features from distorted faces generated by morphing. Several fields of research, such as text categorization [30], handwritten digit recognition [31], facial expression recognition [32, 33], etc., have also employed feature generation to achieve better learning. Unlike these methods, which generate features in the image domain, the proposed method generates features in the feature domain.

The structure of this paper is as follows. In Section II, we first explain the graph-embedding techniques, and give a detailed comparison of those existing subspace-learning approaches similar to our proposed method. After that, we present the extension of subspace analysis to deep learning. In Section III, the proposed SLPM, is formulated. Its relations to SDM is further explored, and its extension to deep soft locality preserving learning is presented. In Section IV, we explain the local descriptors used in our experiments and the feature-generation algorithm, and describe how to enhance the manifold learning with low-intensity images. In Section V, we present the databases used in our experiments, and the preprocessing of the face images. Then, experimental results are presented, with a discussion. We conclude this paper in Section VI.

II. AN OVERVIEW OF SUBSPACE LEARNING

In this section, an overview of the graph-embedding techniques is presented in detail, with the different variants. Then, graph-based subspace-learning methods are described in two parts: (1) how the adjacency matrices are constructed, and (2) how their objective functions are defined. In addition, we also review the subspace learning methods, extended for deep learning, for enhancing the feature discriminative power in facial image analysis. Table 1 summarizes the notations used in this section.

A) Graph embedding

Given m data points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\} \in \mathbb{R}^D$, the graph-based subspace-learning methods aim to find a transformation matrix $\mathbf{A}$ that maps the training data points to a new set of points $\{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_m\} \in \mathbb{R}^d (d \ll D)$, where $\boldsymbol{y}_i = \mathbf{A}^T \boldsymbol{x}_i$ and $\mathbf{A}$ is the projection matrix. After the transformation, the data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, which are close to each other, will have their projections in the manifold space $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ close to each other. This goal can be achieved by minimizing the following objective function:

$$\sum_{ij} (\boldsymbol{y}_i - \boldsymbol{y}_j)^2 w_{ij}, \tag{1}$$

where w_{ij} represents the similarity between the training data $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. If w_{ij} is non-zero, $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ must be close to each other, in order to minimize (1). Taking the data points in the feature space as nodes of a graph, an edge between nodes i and j has a weight of w_{ij}, which is not zero, if they are close to each other. In the literature, we have found three different ways to determine the local geometry of a data point:

1 ***ε*-neighborhood:** this uses the distance to determine the closeness. Given $\varepsilon \in \mathbb{R}$, ε-neighborhood chooses the data points that fall within the circle around $\boldsymbol{x}_i$ with a radius ε. Those data points fall within the ε-neighborhood of $\boldsymbol{x}_i$ can be defined as

$$O(\boldsymbol{x}_i, \varepsilon) = \{\boldsymbol{x} | \ \|\boldsymbol{x} - \boldsymbol{x}_i\|^2 < \varepsilon\}. \tag{2}$$

2 ***k*-nearest neighborhood:** another way of determining the local structure is to use the nearest neighborhood information. Presuming that the closest k points of $\boldsymbol{x}_i$ would still be the closest data points of $\boldsymbol{y}_i$ in the projected manifold space, we can define a function $N(\boldsymbol{x}_i, k)$, which outputs the set of k-nearest neighbors of $\boldsymbol{x}_i$. Two types of neighborhood, with label information incorporated, are considered: $N(\boldsymbol{x}_i, k^+)$ and $N(\boldsymbol{x}_i, k^-)$, which represent the sets of k-nearest neighbors of $\boldsymbol{x}_i$ of the same label and of different labels, respectively.

3 **The class information:** the class or label information is often used in supervised subspace methods. In a desired manifold subspace, the data points belonging to the class of $\boldsymbol{x}_i$ are to be projected such that they are close to each other, so as to increase the intra-class compactness. The data points belonging to other classes are projected, such that they will become farther apart and have larger inter-class separability. The class label information is often combined with either the ε-neighborhood or the k-nearest neighborhood.

The similarity graph is constructed by setting up edges between the nodes. There are different ways of determining the weights of the edges, considering the fact that the distance between two neighboring points can also provide useful information about the manifold. Given a sparse

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

Table 1. List of mathematical notations, acronym and their corresponding descriptions

Symbol	Description	Acronym	Description
$\boldsymbol{x}_i \in \mathbb{R}^D$	i-th data point with the length of D	PCA	Principal component analysis
$\boldsymbol{y}_i \in \mathbb{R}^d$	i-th data point in the learned subspace with the length of d	LDA	Linear discriminant analysis
$\mathbf{A}$	Transformation matrix	MMC	Maximum margin criterion
$\mathbf{A}^T$	Transpose of the transformation matrix $\mathbf{A}$	SDM	Soft discriminant map
$\mathbf{W}$	Similarity matrix	LLE	Locally linear embedding
$\mathbf{D}$	Diagonal matrix	LPP	Locality preserving projection
$\mathbf{L}$	Laplacian matrix	LPMIP	Locality-preserved maximum information projection
$N(\boldsymbol{x}_i, k)$	The set of k-nearest neighbors of $\boldsymbol{x}_i$	CMVM	Constrained maximum variance mapping
$l(\boldsymbol{x}_i)$	The class label of $\boldsymbol{x}_i$	LSDA	Locality-sensitive discriminant analysis
w_{ij}	Similarity between data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$	GL-PCA	Graph Laplacian principal component analysis
w_{ij}^w	Within-class similarity between data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$	OLLP	Orthogonal locality preserving projection
w_{ij}^b	Between-class similarity between data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$	SOLLP	Supervised orthogonal locality preserving projection
k^+	The number of nearest neighbors of the same class label	MFA	Marginal Fisher analysis
k^-	The number of nearest neighbors of the different class label	MMDA	Multi-manifolds discriminant analysis
		SLPM	Soft locality preserving map

symmetric similarity matrix $\mathbf{W}$, two variations have been proposed in the literature:

1 Binary weights: $w_{ij} = 1$ if, and only if, the nodes i and j are connected by an edge, otherwise $w_{ij} = 0$.
2 Heat kernel ($t \in \mathbb{R}$): if the nodes i and j are connected by an edge, the weight of the edge is defined as

$$w_{ij} = \exp\left(\frac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}\right). \tag{3}$$

After constructing the similarity matrix with the weights, the minimization problem defined in (1) can be solved by using the spectral graph theory. Defining the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the diagonal matrix whose entries are the column sum of $\mathbf{W}$, i.e. $d_{ii} = \sum_j w_{ij}$, the objective function is reduced to

$$\begin{aligned} \min \sum_{ij} (\boldsymbol{y}_i - \boldsymbol{y}_j)^2 w_{ij} &= \min \sum_{ij} (\mathbf{A}^T \boldsymbol{x}_i - \mathbf{A}^T \boldsymbol{x}_j)^2 w_{ij} \\ &= \min \mathbf{A}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{A}. \end{aligned} \tag{4}$$

where $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m]$, and $\mathbf{A}$ is the projection matrix whose columns are the projection vectors. To avoid the trivial solution of the objective function, the constraint $\mathbf{A}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{A} = 1$ is often added. After specifying the objective function, the optimal projection matrix $\mathbf{A}$ can be computed by solving the standard eigenvalue decomposition or generalized eigenvalue problem. Equation (4) can be solved by using Lagrange multiplier, as follows:

$$\frac{\partial}{\partial \mathbf{A}} (\mathbf{A}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{A} - \lambda (\mathbf{A}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{A} - 1)) = 0. \tag{5}$$

The solution is $(\mathbf{X}\mathbf{D}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A} = \lambda\mathbf{A}$. The columns of $\mathbf{A}$ should be the eigenvectors of the matrix $(\mathbf{X}\mathbf{D}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{L}\mathbf{X}^T$, corresponding to the $d (d \ll D)$ smallest non-zero eigenvalues.

1) Constructing the within-class and the between-class graph matrices

As mentioned in the previous section, one of the most popular graph-based subspace-learning methods is LPP [12], which uses an intrinsic graph to represent the locality information of the data points, i.e. the neighborhood information. The idea behind LPP is that if the data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are close to each other in the feature space, then they should also be close to each other in the manifold subspace. The similarity matrix $\mathbf{W}$ for LPP can be defined as follows:

$$w_{ij} = \begin{cases} 1, \text{ if } \boldsymbol{x}_i \in N(\boldsymbol{x}_j, k) \text{ or } \boldsymbol{x}_j \in N(\boldsymbol{x}_i, k), \\ 0, \text{ otherwise,} \end{cases} \tag{6}$$

where $N(\boldsymbol{x}_j, k)$ represents the set of k-nearest neighbors of $\boldsymbol{x}_i$. One shortfall of the above formulation for w_{ij} is that it is an unsupervised method, i.e. not using any class-label information. Thinking that the label information can help to find a better separation between different class manifolds, supervised locality preserving projections (SLPP) was introduced in [13]. Denote $l(\boldsymbol{x}_i)$ as the corresponding class label of the data point $\boldsymbol{x}_i$. SLPP uses either one of the following formulations:

$$w_{ij} = \begin{cases} 1, \text{ if } l(\boldsymbol{x}_i) = l(\boldsymbol{x}_j), \\ 0, \text{ otherwise,} \end{cases} \tag{7}$$

$$w_{ij} = \begin{cases} 1, \text{ if } (\boldsymbol{x}_i \in N(\boldsymbol{x}_j, k) \text{ or } \boldsymbol{x}_j \in N(\boldsymbol{x}_i, k)) \\ \quad \text{and } l(\boldsymbol{x}_i) = l(\boldsymbol{x}_j), \\ 0, \text{ otherwise.} \end{cases} \tag{8}$$

Note that equation (7) does not include the neighborhood information to the adjacency graph, and the similarity matrices defined above can be constructed using the heat kernel. This strategy is also adopted in [19, 20] for graph construction. Orthogonal locality preserving projection (OLPP) [34] whose eigenvectors are orthogonal to each other is an extension of LPP. It is worth noting that, in our experiments we applied supervised orthogonal locality preserving projections (SOLPP), which is OLPP with its adjacency matrix including class information.

Yan *et al.* [11] proposed a general framework for dimensionality reduction, named marginal Fisher analysis (MFA). MFA, which is based on graph embedding as LPP, uses

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

two graphs, the intrinsic and penalty graphs, to characterize the intra-class compactness and the interclass separability, respectively. In MFA, the intrinsic graph w_{ij}^{w}, i.e. the within-class graph, is constructed using the neighborhood and class information as follows:

$$w_{ij}^{w} = \begin{cases} 1, \text{ if } \boldsymbol{x}_i \in N(\boldsymbol{x}_j, k_1^+) \text{ or } \boldsymbol{x}_j \in N(\boldsymbol{x}_i, k_1^+), \\ 0, \text{ otherwise,} \end{cases} \tag{9}$$

where k_1^+ is the number of nearest neighbors of the same class of $\boldsymbol{x}_i$. Similarly, the penalty graph w_{ij}^{b}, i.e. the between-class graph, is constructed as follows:

$$w_{ij}^{b} = \begin{cases} 1, \text{ if } \boldsymbol{x}_i \in N(\boldsymbol{x}_j, k_2^-) \text{ or } \boldsymbol{x}_j \in N(\boldsymbol{x}_i, k_2^-), \\ 0, \text{ otherwise,} \end{cases} \tag{10}$$

where k_2^- is the number of nearest neighbors whose class is different from $\boldsymbol{x}_i$.

LSDA [18] and improved locality-sensitive discriminant analysis (ILSDA) [35] are subspace-learning methods proposed in 2007 and 2015, respectively. They construct the similarity matrices in the same way, but LSDA uses binary weights, while ILSDA sets the weight of the edges using the heat kernel. The similarity matrices of LSDA are defined as follows:

$$w_{ij}^{w} = \begin{cases} 1, \text{ if } \boldsymbol{x}_i \in N(\boldsymbol{x}_j, k) \text{ and } l(\boldsymbol{x}_i) = l(\boldsymbol{x}_j), \\ 0, \text{ otherwise.} \end{cases} \tag{11}$$

$$w_{ij}^{b} = \begin{cases} 1, \text{ if } \boldsymbol{x}_i \in N(\boldsymbol{x}_j, k) \text{ and } l(\boldsymbol{x}_i) \neq l(\boldsymbol{x}_j), \\ 0, \text{ otherwise.} \end{cases} \tag{12}$$

It can be observed that the intrinsic and the penalty graphs of MFA, LSDA, and ILSDA are similar to each other. In MFA, the numbers of neighboring points for both the similarity matrices are known, i.e. k_1 and k_2. In LSDA and ILSDA, the k neighbors of $\boldsymbol{x}_i$ are selected, which are then divided for constructing the within-class (k^+ samples the same class as $\boldsymbol{x}_i$) and the between-class matrices (k^- samples of other classes), i.e. $k = k^+ + k^-$. Let k_1 and k_2 be the numbers of samples belonging to the same class and different classes, respectively, for MFA. It is worth noting that the equation $N(\boldsymbol{x}_i, k_1) \cap N(\boldsymbol{x}_i, k_2) = N(\boldsymbol{x}_1, k)$ is not always true. This is because it is not necessarily true that $k^+ = k_1$ and $k^- = k_2$. Therefore, the neighboring points of $\boldsymbol{x}_i$ in LSDA and ILSDA are not the same as MFA, even if $k = k_t = k_1 + k_2$. However, the adjacency matrices constructed in the manifold learning methods are similar to each other. The main difference between the existing methods in the literature is in their definitions of the objective functions. We will elaborate on the differences in the objective functions in the next section.

LPMIP [16], proposed in 2008, uses the ε-neighborhood condition, i.e. $O(\boldsymbol{x}_i, \varepsilon)$. Although it was originally applied as an unsupervised learning method, the class labels were used to construct the locality and non-locality information for facial expression recognition. In 2008, Li *et al.* [17] proposed CMVM, which aims to keep the local structure of the data, while separating the manifold of the different classes farther apart. The local-structure graphs, i.e. the between-class graph and the dissimilarities graph, are defined as follows:

$$w_{ij}^{w} = \begin{cases} 1 \text{ or } \exp\left(\dfrac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}\right), \text{ if } \boldsymbol{x}_i \in O(\boldsymbol{x}_j, \varepsilon), \\ 0, \text{ otherwise.} \end{cases} \tag{13}$$

$$w_{ij}^{b} = \begin{cases} 1, \text{ if } l(\boldsymbol{x}_i) \neq l(\boldsymbol{x}_j), \\ 0, \text{ otherwise.} \end{cases} \tag{14}$$

As (13) and (14) show, the within-class matrix of CMVM only preserves the local structure of the whole data, while the between-class matrix only uses the class label to increase the separability of different class manifolds. In 2015, an extension of CMVM, namely CMVM+ [36], was proposed to overcome the obstacles of CMVM. CMVM+ adds the class information and neighborhood information to the similarity matrices. The updated version of the graphs can be written as follows:

$$w_{ij}^{w} = \begin{cases} 1, \text{ if } \boldsymbol{x}_i \in N(\boldsymbol{x}_j, k) \text{ and } l(\boldsymbol{x}_i) = l(\boldsymbol{x}_j) \\ 0, \text{ otherwise.} \end{cases} \tag{15}$$

$$w_{ij}^{b} = \begin{cases} 1, \text{ if } l(\boldsymbol{x}_j) \in C_{inc}(\boldsymbol{x}_i), \\ 0, \text{ otherwise.} \end{cases} \tag{16}$$

where $C_{inc}(\boldsymbol{x}_i)$ is a set of neighboring points belonging to different classes, i.e. $l(\boldsymbol{x}_i) \neq l(\boldsymbol{x}_j)$. More details of the function $C_{inc}(\boldsymbol{x}_i)$ can be found in [36].

In 2011, multi-manifolds discriminant analysis (MMDA) [37] was proposed for image feature extraction, and applied to face recognition. The idea behind MMDA is to keep the points from the same class as close as possible in the manifold space, with the within-class matrix defined as follows:

$$w_{ij}^{w} = \begin{cases} \exp\left(\frac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}\right), & \text{if } l(\boldsymbol{x}_i) = l(\boldsymbol{x}_j) \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

MMDA also constructs a between-class matrix in order to separate the different classes from each other. The difference between the between-class matrix of MMDA and the other subspace methods is that its graph matrix is constructed by not taking all the data points as nodes, but rather calculating the weighted centers of different classes by averaging all the data points belonging to the classes under consideration. Let $\mathbf{M} = [\tilde{\boldsymbol{m}}_1, \tilde{\boldsymbol{m}}_2, \ldots, \tilde{\boldsymbol{m}}_c]$ be the class-weighted centers, where c is the number of classes. Then, the between-class matrix of MMDA can be written as:

$$w_{ij}^{b} = \exp\left(\frac{-\|\tilde{\boldsymbol{m}}_i - \tilde{\boldsymbol{m}}_j\|^2}{t}\right). \tag{18}$$

In Table 2, a summary is given of the within-class graph and between-class graph for the subspace-learning methods, reviewed in this paper. In this table, the determination of the nearest neighbors; whether or not the class

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

information being used, i.e. supervised or unsupervised learning; and the formulation of the weights are provided for the within-class graph and the between-class graph. For LPP/OLPP/SLPP/SOLPP, only the within-class graph is considered, so the neighborhood, class information, and weight for the between-class graph are listed as n/a (not available).

2) Defining the objective functions

Table 3 summarizes the objective functions of the approaches reviewed in the previous section, as well as the constraints used. We can see that SLPP has only one Laplacian matrix defined in its objective function, because it constructs one similarity matrix only, while all the other methods have two matrices: one is based on the intrinsic graph, and the other on the penalty graph.

In general, there are two ways of defining the objective functions with the intrinsic and the penalty matrices. The first one utilizes the Fisher criterion to maximize the ratio between the scattering of the between-class and that of the within-class Laplacian matrices. MFA, MMDA, and CMVM+ employ the Fisher criterion. Although the application of the Fisher criterion shows its robustness, it involves taking the inverse of a high-dimensional matrix to solve a generalized eigenvalue problem. To solve this problem, LSDA, LPMIP, and our proposed SLMP define their objective functions as the difference between the intrinsic and the penalty-graph matrices, while MMC and SDM use the difference between the inter-class and the intra-class scatter matrices.

As shown in Table 3, ILSDA adopts a similar objective function to LSDA, but with a difference that the within-class scatter matrix is included in the objective function. The within-class scatter matrix $\mathbf{S}_w$ – as used in LDA – indicates the compactness of the data point in each class. ILSDA uses the scatter matrix to project outliers closer to the class centers under consideration. The objective function of ILSDA is defined as follows:

$$\max \mathbf{A}^T(\mathbf{P} - \alpha\mathbf{S}_w)\mathbf{A}, \tag{19}$$

where $\mathbf{P} = \mathbf{X}(\mathbf{L}_b - \mathbf{L}_w)\mathbf{X}^T$, as defined in the objective function of LSDA. CMVM, unlike other methods which aim to minimize the within-class spread, intends to maintain the within-class structure for each class by defining a constraint, i.e. $\mathbf{A}^T\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{A} = \mathbf{X}^T\mathbf{L}_w\mathbf{X}$, while increasing the inter-class separability with the following objective function:

$$\max \mathbf{A}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{A}, \tag{20}$$

where $\mathbf{L}_w$ and $\mathbf{L}_b$ are the within-class and the between-class Laplacian matrices, respectively.

B) Deep subspace learning

Deep subspace learning generally employs multi-level subspace mapping to extract abstract features from an image. One of the most commonly used deep subspace learning frameworks is PCANet [38], proposed by Chan *et al.*, which iteratively utilizes the convolutional layers with the PCA filters to learn image representations. They further proposed LDANet [38] to enhance the feature representations, based on the Fisher criterion. To tackle the efficiency problem caused by cascading the PCA or LDA filters, binary hashing, and block-wise histograms, the pooling operation is introduced into the deep frameworks to reduce the dimensionality of the extracted features, such as the general pooling [39], the rank-based average pooling [40], and the spatial pyramid pooling [41].

The above-mentioned methods aim to learn linear mappings to form the subspace for describing facial images. However, these methods lack the capacity to describe the complexity of facial expressions in real-world scenarios. Deep convolutional neural networks (DCNNs) provide an alternative to learning the subspace by using more complicated nonlinear mappings. Recently, DCNNs have shown their superiority in various computer vision tasks, including image classification [42, 43], object detection [44, 45], and image restoration [46, 47]. A DCNN model is generally trained by minimizing an empirical risk as follows:

$$\theta^* = \arg\min_{\theta} \sum_i \mathcal{L}(f_\theta(\boldsymbol{x}_i), l(\boldsymbol{x}_i)), \tag{21}$$

where f_θ is the nonlinear mapping function with its trainable parameters θ under an objective function $\mathcal{L}$, and $\boldsymbol{x}_i$ represents an input signal, with its corresponding label $l(\boldsymbol{x}_i)$. In terms of the facial expression recognition task, the objective function is generally defined as the softmax loss $\mathcal{L}_{\text{sfm}}$ as follows:

$$\mathcal{L}_{\text{sfm}} = -\sum_i \log \frac{e^{\boldsymbol{W}_{l(\boldsymbol{x}_i)}^T \boldsymbol{y}_i + \boldsymbol{b}_{l(\boldsymbol{x}_i)}}}{\sum_{j=1}^{n} e^{\boldsymbol{W}_j^T \boldsymbol{y}_i + \boldsymbol{b}_j}}, \tag{22}$$

where $\boldsymbol{y}_i = f_\theta(\boldsymbol{x}_i)$ is the learned deep feature for sample $\boldsymbol{x}_i$, and $\boldsymbol{W}$ and $\boldsymbol{b}$ are the trainable kernels and bias of the output layer, respectively. However, the features learned under the softmax loss can achieve class separability only, but there is no guarantee for discriminability. Therefore, deep subspace regularizers are proposed to introduce the within-class and the between-class variances as the additional penalty into the objective function. By this means, the learned deep features become more discriminative and generalized for recognizing new unseen query faces. Similar to the traditional subspace learning algorithms introduced in Section A), deep subspace learning aims to learn a subspace that characterizes face features by widening the between-class differences and compacting the within-class variations. In this paper, we mainly focus on the regularization-based deep subspace learning methods.

1) Regularization for deep subspace learning

Wen *et al.* [28] proposed the Center loss for face recognition, which aims to minimize the within-class variations while keeping the features of different classes separable. To this end, the Center loss minimizes the distance between each sample and its corresponding class center in the latent space,

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

Table 2. Comparison of the within-class graph and the between-class graph for different subspace-learning methods

	The within-class graph			The between-class graph		
Methods	Neighborhood	Class info	Weight	Neighborhood	Class info	Weight
LPP [12]/OLPP [34]	Optional	No	Optional	n/a	n/a	n/a
SLPP [13]/SOLPP	Optional	Yes	Optional	n/a	n/a	n/a
LSDA [18]	knn	Yes	bn	knn	Yes	bn
MFA [11]	knn	Yes	bn	knn	Yes	bn
CMVM [17]	ε-ball	No	bn/hk	n/a	Yes	bn
LPMIP [16]	ε-ball	No	hk	ε-ball	No	hk
MMDA [37]	n/a	Yes	hk	Class centers	Yes	hk
CMVM+ [36]	knn	Yes	bn	knn	Yes	bn
ILSDA [35]	knn	Yes	hk	knn	Yes	hk
SLPM (proposed)	knn	Yes	bn/hk	knn	Yes	hk

bn: binary weights, hk: heat kernel, knn: k-nearest neighbor.

Table 3. Comparison of the objective functions used by different subspace methods

Methods	Objective functions	Constraints (s.t.)
LPP [12]/SLPP [13]	$\max_{\mathbf{A}} \mathbf{A}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A}$	$\mathbf{A}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{A} = \mathbf{I}$
LSDA [18]	$\max_{\mathbf{A}} \mathbf{A}^T\mathbf{X}(\alpha\mathbf{L}_b + (1-\alpha)\mathbf{W}_w)\mathbf{X}^T\mathbf{A}$	$\mathbf{A}^T\mathbf{X}\mathbf{D}_w\mathbf{X}^T\mathbf{A} = \mathbf{I}$
MFA [11]	$\min_{\mathbf{A}} \frac{\mathbf{A}^T\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{A}}{\mathbf{A}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{A}}$	n/a
CMVM [17]	$\max_{\mathbf{A}} \mathbf{A}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{A}$	$\mathbf{A}^T\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{A} = \mathbf{X}\mathbf{L}_w\mathbf{X}^T$
LPMIP [16]	$\max_{\mathbf{A}} \mathbf{A}^T\mathbf{X}(\alpha\mathbf{L}_b - (1-\alpha)\mathbf{W}_w)\mathbf{X}^T\mathbf{A}$	$\mathbf{A}^T\mathbf{A} - \mathbf{I} = 0$
MMDA [37]	$\max_{\mathbf{A}} \frac{\mathbf{A}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{A}}{\mathbf{A}^T\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{A}}$	n/a
CMVM+ [36]	$\max_{\mathbf{A}} \frac{\mathbf{A}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{A}}{\mathbf{A}^T\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{A}}$	n/a
ILSDA [35]	$\max \mathbf{A}^T(\mathbf{P} - \alpha\mathbf{S}_w)\mathbf{A}$ where $\mathbf{P} = \mathbf{X}(\mathbf{L}_b - \mathbf{L}_w)\mathbf{X}^T$	$\mathbf{A}^T\mathbf{A} - \mathbf{I} = 0$
SDM [7]	$\max \mathbf{S}_b - \alpha\mathbf{S}_w$	n/a
SLPM	$\max \mathbf{A}^T(\mathbf{X}\mathbf{L}_b\mathbf{X}^T - \beta\mathbf{X}\mathbf{L}_w\mathbf{X}^T)\mathbf{A}$ or $\max \mathbf{A}^T\mathbf{X}(\mathbf{L}_b - \beta\mathbf{L}_w)\mathbf{X}^T\mathbf{A}$	$\mathbf{A}^T\mathbf{A} - \mathbf{I} = 0$

as follows:

$$\mathcal{L}_{\text{Center}} = \frac{1}{2}\sum_{i=1}^{m} ||\boldsymbol{y}_i - \boldsymbol{c}_{l(\boldsymbol{x}_i)}||_2^2, \tag{23}$$

where $\boldsymbol{c}_{l(\boldsymbol{x}_i)}$ denotes the center or mean deep feature for the class $l(\boldsymbol{x}_i)$, and m is the batch size. This Center loss can effectively describe the within-class variations. However, the between-class variations are not sufficiently considered, which may result in the overlaps between the clusters of different classes in the latent space. Therefore, in order to enlarge the between-class distance, Cai *et al.* [26] proposed the Island loss, which further introduces a regularization term to penalize the pairwise distance between the centers of different classes, as follows:

$$\begin{aligned}\mathcal{L}_{\text{Island}} = &\frac{1}{2}\sum_{i=1}^{m} ||\boldsymbol{y}_i - \boldsymbol{c}_{l(\boldsymbol{x}_i)}||_2^2 \\ &+ \lambda \sum_{\boldsymbol{c}_j \in N} \sum_{\substack{\boldsymbol{c}_k \in N \\ k \neq j}} \left(\frac{\boldsymbol{c}_k \cdot \boldsymbol{c}_j}{||\boldsymbol{c}_k||_2 ||\boldsymbol{c}_j||_2} + 1 \right),\end{aligned} \tag{24}$$

where λ is a hyperparameter controlling the trade-off between the intra-class and the inter-class variations, and N denotes the set of class centers in the learned subspace. The second term in this loss function is normalized to the range $[0, 2]$. Compared to the Center loss, the Island loss further maximizes the distance between the centers of the different classes. This effectively addresses the overlap issue and significantly improves the feature discriminative ability.

On the contrary, local information is essential for the formation of a feature space with better generalization [27]. Inspired by SLPP [13], Li *et al.* [27] proposed the locality preserving (LP) loss, and established the deep locality preserving-CNN (DLP-CNN), which aims to guarantee the local consistency in the learned subspace. The locality preserving loss is formulated as follows:

$$\mathcal{L}_{LP} = \sum_{i,j} S_{i,j} ||\boldsymbol{y}_i - \boldsymbol{y}_j||, \tag{25}$$

where the similarity matrix $S_{i,j}$ is defined, based on SLPP [13], as follows:

$$S_{i,j} = \begin{cases} 1, \text{ if } (\boldsymbol{x}_i \in N(\boldsymbol{x}_j, k) \text{ or } \boldsymbol{x}_j \in N(\boldsymbol{x}_i, k)) \\ \quad \text{and } l(\boldsymbol{x}_i) = l(\boldsymbol{x}_j), \\ 0, \text{ otherwise.} \end{cases} \tag{26}$$

However, to calculate the sum of the pairwise distances, the entire training set is required to be fed to the network for training in each iteration, which is computationally intensive. Therefore, Li *et al.* [27] further proposed to make an approximation by only searching the k nearest neighbors of $\boldsymbol{y}_i$ in each iteration. Thus, the locality preserving loss is

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

reformulated as follows:

$$\mathcal{L}_{\mathrm{LP}} = \frac{1}{2}\sum_{i=1}^{m}\left\|\boldsymbol{y}_i - \frac{1}{k}\sum_{\boldsymbol{y}\in N(\boldsymbol{y}_i,k)}\boldsymbol{y}\right\|_2^2, \tag{27}$$

where $N(\boldsymbol{y}_i, k)$ denotes the ensemble of the k-nearest neighbors of the feature point $\boldsymbol{y}_i$ with the same label. Equation (27) effectively characterizes the "local" within-class scatters, because the samples from the same class are forced to be close to each other in the latent subspace. Moreover, if we set the number of neighbors $k = N_j$, where N_j is the number of samples from the j-th class in the entire training set, the locality preserving loss becomes the Center loss. Thus, the Center loss can be regarded as a special case of the locality preserving loss.

However, the locality preserving loss, similar to the Center loss, has not sufficiently considered the local inter-class variations, and only adopts the softmax loss to make the features separable. Therefore, in this paper, we reformulate the proposed SLPM as an additional regularization, i.e. soft locality preserving (SLP) loss, for training the CNN models, in order to learn more discriminative representations for facial expressions. We summarize the above-mentioned deep subspace-learning methods in Table 4.

2) Learning scheme

Both the center-based [26, 28] and the locality-based [27] methods need to take all the training samples, which are fed to the network, to calculate the respective class centers in training. This is not only time-consuming, but also impractical in real-world applications. To address this problem, a mini-batch-based training scheme is necessary to reliably update or compute the centers for those additional regularization terms.

For the center-based approaches, Wen *et al.* [26] proposed to compute the gradients of the class centers based on the samples in a mini-batch only. Specifically, the class centers are first randomly initialized at the beginning of the training process. In each iteration, the gradient of $\mathcal{L}_{\mathrm{Center}}$ with respect to the FV $\boldsymbol{y}_i$ is calculated as follows:

$$\frac{\partial \mathcal{L}_{Center}}{\partial \boldsymbol{y}_i} = \boldsymbol{y}_i - \boldsymbol{c}_{l(\boldsymbol{x}_i)}, \tag{28}$$

and then the centers are updated in each iteration as follows:

$$\Delta \boldsymbol{c}_j = \frac{\sum_{i=1}^{m}\delta(l(\boldsymbol{x}_i)=j)(\boldsymbol{c}_j - \boldsymbol{y}_i)}{1+\sum_{i=1}^{m}\delta(l(\boldsymbol{x}_i)=j)}, \tag{29}$$

where $\delta(\text{condition}) = 1$ if the condition is satisfied, and $\delta(\text{condition}) = 0$, otherwise. By this means, the randomly initialized class centers can be optimized during the mini-batch training. Similarly, the center gradients in the Island loss can be computed as follows:

$$\Delta \boldsymbol{c}_j = \frac{\sum_{i=1}^{m}\delta(l(\boldsymbol{x}_i)=j)(\boldsymbol{c}_j - \boldsymbol{y}_i)}{1+\sum_{i=1}^{m}\delta(l(\boldsymbol{x}_i)=j)} + \frac{\lambda}{|N|-1}\sum_{\substack{\boldsymbol{c}_k\in N\\ k\neq j}}\frac{\boldsymbol{c}_k}{||\boldsymbol{c}_k||_2||\boldsymbol{c}_j||_2} - \left(\frac{\boldsymbol{c}_k\cdot\boldsymbol{c}_j}{||\boldsymbol{c}_k||_2||\boldsymbol{c}_j||_2^3}\right)\boldsymbol{c}_j, \tag{30}$$

where $|N|$ denotes the total number of expression classes.

As the center-based method is just a special case of the locality-based method, the center updating scheme in DLP-CNN [27] can be defined similarly, as follows:

$$\frac{\partial \mathcal{L}_{LP}}{\partial \boldsymbol{y}_i} = \boldsymbol{y}_i - \frac{1}{k}\sum_{\boldsymbol{y}\in N(\boldsymbol{y}_i,k)}\boldsymbol{y}. \tag{31}$$

It is worth noting that all the above-mentioned regularizers are cooperating with the softmax loss defined in equation (22) to jointly supervise the subspace learning process.

III. SOFT LOCALITY PRESERVING MAP

In this section, we introduce the proposed method, SLPM, with its formulation and connection to the previous studies. Then, we will also describe the local descriptors used for facial expression recognition in our experiments. Finally, we extend SLPM to deep learning, and describe the deep network architecture for learning discriminative features supervised by the soft locality preserving loss.

A) Formulation of the SLPM

Similar to other manifold-learning algorithms, two graph-matrices, i.e. the between-class matrix $\mathbf{W}_b$ and the within-class matrix $\mathbf{W}_w$, are constructed to characterize the discriminative information, based on the locality and class-label information. Given m data points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\} \in \mathbb{R}^D$ and their corresponding class labels $\{l(\boldsymbol{x}_1), l(\boldsymbol{x}_2), \ldots, l(\boldsymbol{x}_m)\}$, we denote $N_w(\boldsymbol{x}_i, k_w) = \{\boldsymbol{x}_i^{w_1}, \boldsymbol{x}_i^{w_2}, \ldots, x_i^{w_{k_w}}\}$ as the set of k_w-nearest neighbors with the same class label as $\boldsymbol{x}_i$, i.e. $l(\boldsymbol{x}_i) = l(\boldsymbol{x}_i^{w_1}) = l(\boldsymbol{x}_i^{w_2}) = \cdots = l(\boldsymbol{x}_i^{w_{k_w}})$, and $N_b(\boldsymbol{x}_i, k_b) = \{\boldsymbol{x}_i^{b_1}, \boldsymbol{x}_i^{b_2}, \ldots, \boldsymbol{x}_i^{b_{k_b}}\}$ as the set of its k_b nearest neighbors with different class labels from $\boldsymbol{x}_i$, i.e. $l(\boldsymbol{x}_i) \neq l(\boldsymbol{x}_i^{w_j})$, where $j = 1, 2, \ldots, k_b$. Then, the inter-class weight matrix $\mathbf{W}_b$ and the intra-class weight matrix $\mathbf{W}_w$ can be defined as below:

$$w_{ij}^b = \begin{cases} \exp\left(\dfrac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}\right), \boldsymbol{x}_j \in N_b(\boldsymbol{x}_i, k_b), \\ 0, \text{ otherwise.} \end{cases} \tag{32}$$

$$w_{ij}^w = \begin{cases} \exp\left(\dfrac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}\right), \boldsymbol{x}_j \in N_w(\boldsymbol{x}_i, k_w), \\ 0, \text{ otherwise.} \end{cases} \tag{33}$$

SLPM is a supervised manifold-learning algorithm, which aims to maximize the between-class separability, while controlling the within-class spread with a control parameter β

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

Table 4. Comparison of the different deep subspace-learning regularizers

	The within-class variation			The between-class variation		
Methods	Region	Class info	Weight	Region	Class info	Weight
Center loss [28]	Global	Yes	bn	n/a	n/a	n/a
Island loss [26]	Global	Yes	bn	Global	Yes	bn
LP loss [27]	Local	Yes	bn	n/a	n/a	n/a
SLP loss	Local	Yes	bn/hk	Local	Yes	hk

bn: binary weights, hk: heat kernel.

used in the objective function. Consider the problem of creating a subspace, such that data points from different classes, i.e. represented as edges in $\mathbf{W}_b$, stay as distant as possible, while data points from the same class, i.e. represented as edges in $\mathbf{W}_w$, stay close to each other. To achieve this, two objective functions are defined as follows:

$$\max \frac{1}{2}\sum_{i,j}(\boldsymbol{y}_i - \boldsymbol{y}_j)^2 w^b_{i,j}, \tag{34}$$

$$\min \frac{1}{2}\sum_{i,j}(\boldsymbol{y}_i - \boldsymbol{y}_j)^2 w^w_{i,j}. \tag{35}$$

Equation (34) ensures that the samples from different classes will stay as far as possible from each other, while equation (35) is to make samples from the same class stay close to each other after the projection. However, as shown in [48] and [7], small variations in the manifold subspace can lead to overfitting in training. To overcome this problem, we add the parameter β to control the intra-class spread. Note that, the method SDM in [7] uses the within-class scatter matrix $\mathbf{S}_w$ – as defined for LDA – to control the intra-class spread. In our proposed method, we adopt the graph-embedding method, which uses the locality information about each class, in addition to the class information. Hence, the two objective functions equations (34) and (35) can be combined as follows:

$$\begin{aligned}\max \frac{1}{2}&\left(\sum_{i,j}(\boldsymbol{y}_i - \boldsymbol{y}_j)^2 w^b_{i,j} - \beta\sum_{i,j}(\boldsymbol{y}_i - \boldsymbol{y}_j)^2 w^w_{i,j}\right),\\ &= \max\left(J_b(\mathbf{A}) - \beta J_w(\mathbf{A})\right),\end{aligned} \tag{36}$$

where $\mathbf{A}$ is a projection matrix, i.e. $\mathbf{Y} = \mathbf{A}^T\mathbf{X}$ and $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]$. Then, the between-class objective function $J_b(\mathbf{A})$ can be reduced to

$$\begin{aligned}J_b(\mathbf{A}) &= \frac{1}{2}\sum_{i,j}(\boldsymbol{y}_i - \boldsymbol{y}_j)^2 w^b_{i,j}\\ &= \mathbf{A}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{A},\end{aligned} \tag{37}$$

where $\mathbf{L}_b = \mathbf{D}_b - \mathbf{W}_b$ is the Laplacian matrix of $\mathbf{W}_b$ and $d_{b_{ii}} = \sum_j w^b_{ij}$ is a diagonal matrix. Similarly, the within-class objective function $J_w(\mathbf{A})$ can be written as

$$\begin{aligned}J_w(\mathbf{A}) &= \frac{1}{2}\sum_{i,j}(\boldsymbol{y}_i - \boldsymbol{y}_j)^2 w^w_{i,j}\\ &= \mathbf{A}^T\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{A},\end{aligned} \tag{38}$$

where $\mathbf{L}_w = \mathbf{D}_w - \mathbf{W}_w$ and $d_{w_{ii}} = \sum_j w^w_{ij}$. If J_b and J_w are substituted into equation (36), the objective function becomes as follows:

$$\begin{aligned}\max J_T(\mathbf{A}) &= \max\left(J_b(\mathbf{A}) - \beta J_w(\mathbf{A})\right)\\ &= \max\left(\mathbf{A}^T\mathbf{X}\mathbf{L}_b\mathbf{X}^T\mathbf{A} - \beta\mathbf{A}^T\mathbf{X}\mathbf{L}_w\mathbf{X}^T\mathbf{A}\right)\\ &= \max \mathbf{A}^T\mathbf{X}(\mathbf{L}_b - \beta\mathbf{L}_w)\mathbf{X}^T\mathbf{A},\end{aligned} \tag{39}$$

which is subject to $\mathbf{A}^T\mathbf{A} - \mathbf{I} = 0$, so as to guarantee orthogonality. By using Lagrange multiplier, we obtain

$$\mathrm{L}(\mathbf{A}) = \mathbf{A}^T\mathbf{X}(\mathbf{L}_b - \beta\mathbf{L}_w)\mathbf{X}^T\mathbf{A} - \lambda(\mathbf{A}^T\mathbf{A} - \mathbf{I}). \tag{40}$$

By computing the partial derivative of $\mathrm{L}(\mathbf{A})$, the optimal projection matrix $\mathbf{A}$ can be obtained, as follows:

$$\frac{\partial \mathrm{L}(\mathbf{A})}{\partial \mathbf{A}} = \mathbf{X}(\mathbf{L}_b - \beta\mathbf{L}_w)\mathbf{X}^T\mathbf{A} - \lambda\mathbf{A}, \tag{41}$$

i.e. $\mathbf{X}(\mathbf{L}_b - \beta\mathbf{L}_w)\mathbf{X}^T\mathbf{A} = \lambda\mathbf{A}$. The projection matrix $\mathbf{A}$ can be obtained by computing the eigenvectors of $\mathbf{X}(\mathbf{L}_b - \beta\mathbf{L}_w)\mathbf{X}^T$. The columns of $\mathbf{A}$ are the d leading eigenvectors, where d is the dimension of the subspace. The proposed SLPM algorithm requires computing the pairwise distance between the samples for construction of the within-class and between-class matrices. The complexity of this is $\mathcal{O}(m^2)$, where m is the number of training samples. In addition, the complexity of calculating the eigenvalue decomposition is $\mathcal{O}(m^3)$. LDA, LPP, MFA, and other manifold-learning algorithms, whose objective functions have a similar structure, lead to a generalized eigenvalue problem. Such methods suffer from the matrix-singularity problem, because the solution involves computing the inverse of a singular matrix. Although computing the inverse of a matrix also involves a time complexity of $\mathcal{O}(m^3)$, the proposed objective function is designed in such a way as to overcome this singularity problem. However, in our algorithm, PCA is still applied to data, so as to reduce its dimensionality and to reduce noise.

B) Intra-class spread

As we have mentioned before, the manifold spread of the different classes can affect the generalizability of the learned classifier. To control the spread of the classes, the parameter β is adjusted in our proposed method, like SDM. Figure 1 shows the change in the spread of the classes when β increases. We can see that increasing β will also increase the separability of the data, e.g. the training data is located at almost the same position in the subspace when $\beta = 1000$.

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

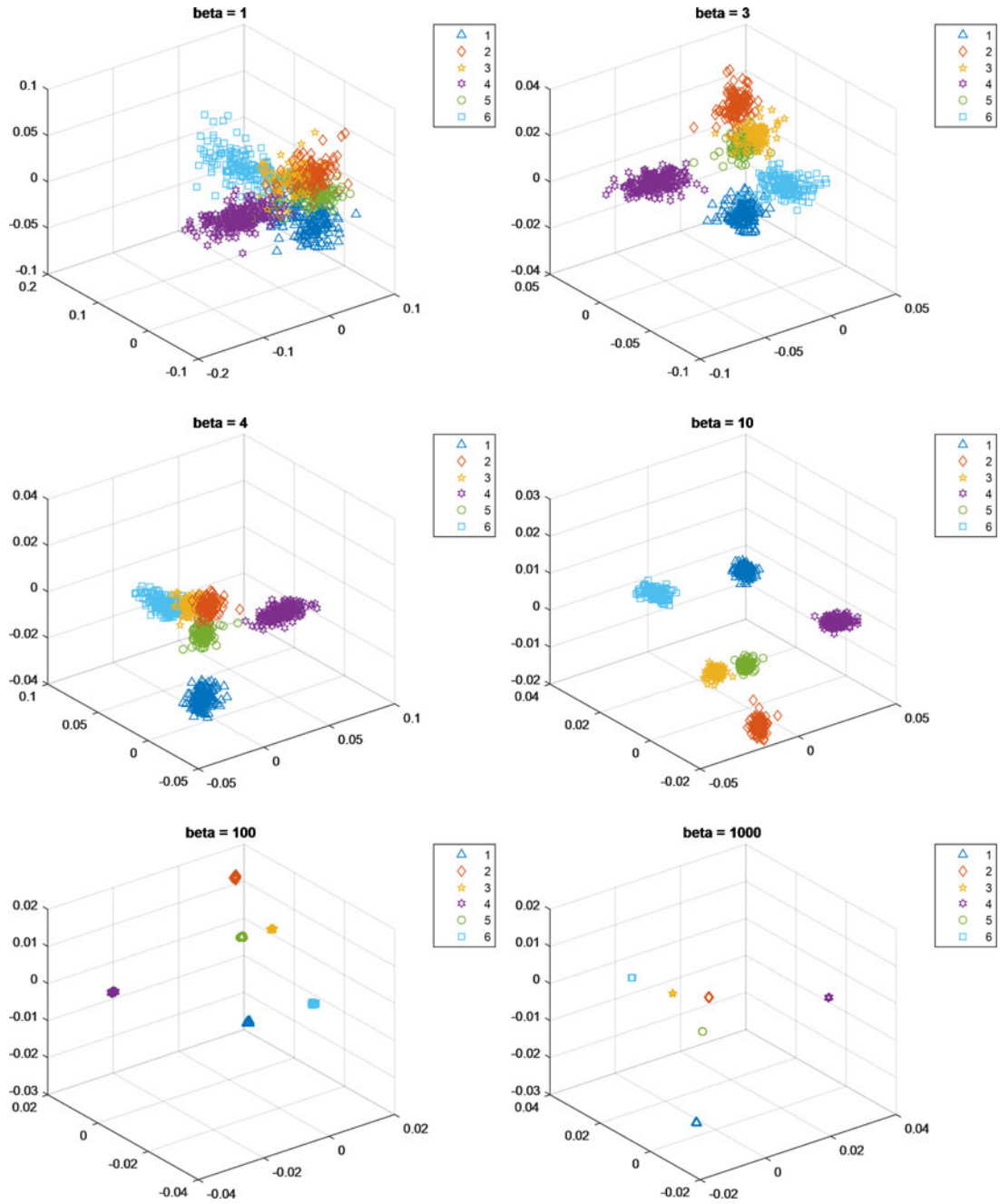


Fig. 1. Spread of the respective expression manifolds when the value of β increases from 1 to 1000: (1) anger, (2) disgust, (3) fear, (4) happiness, (5) sadness, and (6) surprise.

C) Relations to other subspace-learning methods

As discussed in Section 2, there have been extensive studies on manifold-learning methods. They share the same core idea, i.e. using locality and/or label information to define an objective function, so that the data can be represented in a specific way after projection.

There are two main differences between SLPM and LSDA. First, LSDA defines their objective function as a subtraction of two objective functions like SLPM. However, LSDA imposes the constraint $\mathbf{A}^T\mathbf{X}\mathbf{D}_w\mathbf{X}^T\mathbf{A} = \mathbf{I}$, which results in a generalized eigenvalue problem. As we mentioned in Section 2, the generalized eigenvalue problem suffers from the computational cost of calculating an inverse matrix. SLPM only determines the orthogonal projections, with the constraint $\mathbf{A}^T\mathbf{A} - \mathbf{I} = 0$. Therefore, SLPM can still be computed by eigenvalue decomposition, without requiring computing any inverse matrix. Second, LSDA finds the neighboring points followed by determining whether the considered neighboring points are of the same class or of different classes. This may lead to an unbalanced and unwanted division of neighboring points, simply because of the fact that a sample point may be surrounded by more

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

Table 5. Network architecture for learning discriminative features supervised by the soft locality preserving loss

Type	Conv	ReLU	MPool	Conv	ReLU	MPool	Conv	ReLU	Conv	ReLU	MPool	Conv	ReLU	Conv	ReLU	FC	ReLU	FC
KS	3	–	2	3	–	2	3	–	3	–	2	3	–	3	–	–	–	–
OC	64	–	–	96	–	–	128	–	128	–	–	256	–	256	–	2000	–	7
P	1	0	0	1	0	0	1	0	1	0	0	1	0	1	0	–	0	–
S	1	1	2	1	1	2	1	1	1	1	2	1	1	1	1	–	1	–

KS, OC, P, and S refer to the kernel size, output channel, padding, and stride, respectively.

samples belonging to the same class than samples with different class labels. In order not to lose locality information in such a case, SLPM defines two parameters k_1 and k_2, which are the numbers of neighboring points belonging to the same class and different classes, respectively. In other words, the numbers of neighboring points belonging to the same class and different classes can be controlled.

Both SDM and ILSDA also consider the intra-class spread when defining the objective function. SDM controls the level of spread by applying a parameter to the within-class scatter matrix $\mathbf{S}_w$. However, it only uses the label information about the training data – its scatter matrices do not consider the local structure of the data. Our proposed SLPM aims to include the locality information by employing graph embedding in our objective functions. Therefore, SLPM is a graph-based version of SDM. ILSDA uses both the label and neighborhood information represented in the adjacent matrices, and also aims to control the spread of the classes. However, ILSDA achieves this by adding the scatter matrix $\mathbf{S}_w$ to its objective function. In our algorithm, we propose controlling the spread with the within-class Laplacian matrix $\mathbf{L}_w$, without adding a separate element to the objective function.

D) Deep soft locality preserving learning

The objective function of SLPM, as described in equation (36), is extended for deep learning for learning more discriminant deep features. The objective function is formulated as a loss function, which aims to minimize the within-class variation, while maximizing the between-class difference. In equation (36), $\boldsymbol{y}$ is the feature in the learned subspace formed by a linear projection matrix $\mathbf{A}$. w^b and w^w represent the similarity between the between-class samples and the within-class samples, respectively, in a local neighborhood. If we employ a deep neural network to extract the FV from each sample, denoted as $\boldsymbol{y}_i = f_\theta(\boldsymbol{x}_i)$, the learning objective becomes as follows:

$$\begin{aligned}\theta^* = \underset{\theta}{\operatorname{argmin}}\, \beta \sum_{i,j} w^w_{i,j} ||\boldsymbol{y}_i - \boldsymbol{y}_j||_2^2 \\ - \sum_{i,j} w^b_{i,j} ||\boldsymbol{y}_i - \boldsymbol{y}_j||_2^2, \\ \text{with } \boldsymbol{y}_i = f_\theta(\boldsymbol{x}_i),\end{aligned} \tag{42}$$

where the similarity of the inter-class and intra-class samples, i.e. $w^b_{i,j}$ and $w^w_{i,j}$, are computed based on equations (32) and (33), respectively.

Similar to DLP-CNN [27], the learned feature $\boldsymbol{y}_i$ should be updated iteratively during the mini-batch training. Therefore, we adopt the same approximation in [27] to only consider the k nearest neighbors of each feature $\boldsymbol{y}_i$, and reformulate the objective function as follows:

$$\begin{aligned}\mathcal{L}_{SLP} = \beta \sum_{i=1}^{m} \left|\left| \boldsymbol{y}_i - \frac{1}{k_w} \sum_{\boldsymbol{y} \in N_w(\boldsymbol{y}_i, k_w)} \boldsymbol{y} \right|\right|_2^2 \\ - \sum_{i=1}^{m} \left|\left| \boldsymbol{y}_i - \frac{1}{k_b} \sum_{\boldsymbol{y} \in N_b(\boldsymbol{y}_i, k_b)} \boldsymbol{y} \right|\right|_2^2,\end{aligned} \tag{43}$$

where β also controls the intra-class spread, which affects the generalization of the resultant feature extractor. Equation (43) represents the proposed soft locality-preserving (SLP) loss, which effectively characterizes the within-class and the between-class variations in a local region, and consequently enhances the discriminability and the generalization of the model.

We follow the learning strategy in [26–28], and adopt the joint supervision of the softmax and the SLP loss to train up the CNN model, named SLP-CNN, for subspace learning. Thus, the overall loss function is defined as follows:

$$\mathcal{L} = \mathcal{L}_{sfm} + \lambda \mathcal{L}_{SLP}, \tag{44}$$

where λ balances the trade-off between the two loss terms. The softmax loss guarantees the separability of the global scatter, while the SLP loss enhances the discriminative power based on local scatters.

To learn the deep discriminative features with respect to facial expressions, we establish a convolutional neural network with the same architecture as DLP-CNN [27], whose structure is shown in Table 5. We adopt an 18-layer CNN with the ReLU [49] activation function. The last fully connected layer in Table 5 is the softmax layer for introducing the softmax supervision. It can be seen from the table that we extract a 2000-dimensional FV from each facial sample. The SLP loss is computed based on these 2000-dimensional FVs, produced by the second last layer. We summarize our proposed learning algorithm, SLP-CNN, as shown in Algorithm 1. In terms of the time complexity, the proposed algorithm only affects that in the training stage, and thus, brings no burden to the inference stage. The proposed SLP loss requires computing the pairwise distance between the samples doing training. Therefore, the time complexity in a mini-batch is described as $\mathcal{O}(m^2)$. As a comparison, the LP loss requires computing the distance between samples in the same cluster, whose computation should be $\mathcal{O}(m'^2)$,

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

where m' denotes the number of samples belonging to the same class in a mini-batch.

Algorithm 1: Learning algorithm for SLP-CNN

Input: Training samples $\{\boldsymbol{x}_i\}_{i=1}^{N}$.

1: **Initialize**: Network parameters $\theta^{(0)}$, learning rate μ, hyperparameters λ, softmax layer parameters $\phi^{(0)}$, neighboring nodes k_w and k_b, number of epochs T, mini-batch size m.

2: **for** $t = 0 : T$ **do**

3: extract feature with CNN:

$$\boldsymbol{y}_i = f_{\theta^{(t)}}(\boldsymbol{x}_i)$$

4: compute the within-class and the between-class centers from the k-nearest neighbors of $\boldsymbol{y}_i$:

$$\boldsymbol{c}_w = \frac{1}{k_w}\sum_{j=1}^{m}\boldsymbol{y}_j w_{i,j}^{w}$$
$$\boldsymbol{c}_b = \frac{1}{k_b}\sum_{j=1}^{m}\boldsymbol{y}_j w_{i,j}^{b}$$

5: update the softmax layer parameters:

$$\phi^{(t+1)} = \phi^{(t)} - \mu^{(t)}\frac{\partial \mathcal{L}_{sfm}^{(t)}}{\partial \phi^{(t)}},$$

6: update backpropagation error:

$$\frac{\partial \mathcal{L}^{(t)}}{\boldsymbol{y}_i} = \frac{\partial \mathcal{L}_{sfm}^{(t)}}{\boldsymbol{y}_i} + \lambda\frac{\partial \mathcal{L}_{SLP}^{(t)}}{\boldsymbol{y}_i},$$

7: update network parameters:

$$\theta^{(t+1)} = \theta^{(t)} - \mu^{(t)}\frac{\partial \mathcal{L}^{(t)}}{\partial \theta^{(t)}}$$
$$= \theta^{(t)} - \mu^{(t)}\sum_{i=1}^{m}\frac{\partial \mathcal{L}^{(t)}}{\partial \boldsymbol{y}_i}\frac{\partial \boldsymbol{y}_i}{\partial \theta^{(t)}},$$

8: **end for**

Output: Trained network parameter $\theta^{(T)}$.

IV. FEATURE DESCRIPTORS AND GENERATION

In this section, we will first present the descriptors used for representing facial images for expression recognition, then investigate the use of face images with low-intensity and high-intensity expressions for manifold learning, which represent the corresponding samples at the core and boundary of the manifold for an expression. After that, we will introduce our proposed feature-generation algorithm.

A) Descriptors

Recent research has shown that local features can achieve higher and more robust recognition performance than by using global features, such as eigenfaces and Fisherfaces, and intensity values. Therefore, in order to show the robustness of our proposed method, four different commonly used local descriptors for facial expression recognition, local binary pattern (LBP) [50, 51], local phase quantization (LPQ) [52], pyramid of histogram of oriented gradients [53], and Weber local descriptor (WLD) [54], are considered in our experiments. These descriptors can represent face images, in terms of different aspects such as intensity, phase, shape, etc., so that they are complementary to each other [55]. As shown in Algorithm 2, features are extracted using one of the above-mentioned local descriptors, followed by the subspace learning with SLPM and a feature-generation method.

Algorithm 2: The overall flow of our proposed SLPM

1. Extract features from face images: $\mathbf{X}_{desc}$.
2. Learn the projection matrix $\mathbf{W}_{pca}$ via PCA.
3. Construct the within-class graph matrix $\mathbf{W}_w$ and the between-class similarity matrices $\mathbf{W}_b$.
4. Calculate the Laplacian matrices $\mathbf{L}_w$ and $\mathbf{L}_b$.
5. Solve the eigenvalue decomposition of $\mathbf{X}(\mathbf{L}_b - \beta\mathbf{L}_w)\mathbf{X}^T$.
6. Choose the eigenvectors corresponding to the d largest eigenvalues, $\mathbf{W}_{mL}$.
7. $\mathbf{Y}_{desc} = \mathbf{W}_{mL}^T\mathbf{W}_{pca}^T\mathbf{X}_{desc}$.
8. Add features obtained with either low-intensity images ($\mathbf{Y}^l$) or feature generation ($\overline{\mathbf{Y}}^l$) to form the training data $\mathbf{T}^l$ or $\overline{\mathbf{T}}^l$, respectively.
9. Learn the nearest neighbor classifier.

B) Feature generation

Features in a projected subspace still have a high dimension. A large number of samples for each expression is necessary in order to accurately represent its corresponding manifold. This is similar to deep learning in that an extremely large amount of training samples are necessary to solve the overfitting problem. For deep learning, data augmentation is carried out to generate more samples from a single training image. However, for conventional subspace analysis methods, data augmentation does not work properly. To achieve effective learning, it is necessary to generate more features located near the manifold boundaries. Then, more accurate decision boundaries can be determined for accurate facial expression. In other words, feature augmentation should be performed, rather than data augmentation.

Video sequences with face images, changing from neutral expression to a particular expression, are used for learning. Let $f_{i,\theta}$ denote the frame index of the face image of expression intensity θ ($0 \le \theta \le 1$, 0 = neutral expression and 1 = the highest intensity of an expression, i.e. the peak expression) of the sequence S_i in a dataset of m video sequences. Let $x_i^{\theta} \in \mathbb{R}^D$ be the FV extracted from the $f_{i,\theta}$-th frame of the sequence S_i. The frame index $f_{i,\theta}$ can be calculated as follows:

$$f_{i,\theta} = n_i \times \theta, \tag{45}$$

where n_i is the number of frames in the sequence S_i. Therefore, $\{\boldsymbol{x}_1^1, \boldsymbol{x}_2^1, \ldots, \boldsymbol{x}_m^1\} \in \mathbb{R}^D$are the FVs extracted from the face images with high-intensity expressions, i.e. the last frames of the m video sequences. Suppose that $\{\boldsymbol{x}_1^{\xi}, \boldsymbol{x}_2^{\xi}, \ldots, \boldsymbol{x}_m^{\xi}\}$are the FVs extracted from the corresponding low-intensity images, and the corresponding frame number in the respective video sequences is $f_{i,\xi}$. In our

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

algorithm, we use a different set of ξ values, where $0.6 \leq \xi \leq 0.9$, to learn the different expression manifolds.

1) Manifold learning with high- and low-intensity training samples

A projection matrix $\mathbf{A}$ that maps the FVs $\mathbf{X}^1 = [\boldsymbol{x}_1^1, \boldsymbol{x}_2^1, \ldots, \boldsymbol{x}_m^1]$ to a new subspace is first calculated using SLPM. The corresponding projected samples are denoted as $\mathbf{Y}^1 = [\boldsymbol{y}_1^1, \boldsymbol{y}_2^1, \ldots, \boldsymbol{y}_m^1]$, i.e. $\boldsymbol{y}_i^1 = \mathbf{A}^T \boldsymbol{x}_i^1$. Then, the same projection matrix $\mathbf{A}$ is used to map the low-intensity FVs $\mathbf{X}^\xi = [\boldsymbol{x}_1^\xi, \boldsymbol{x}_2^\xi, \ldots, \boldsymbol{x}_m^\xi]$, i.e. $\boldsymbol{y}_i^\xi = \mathbf{A}^T \boldsymbol{x}_i^\xi$, which should lie on the boundary of the corresponding expression manifold. The high-intensity and low-intensity samples in the subspace form a training matrix, denoted as $\mathbf{T}_\xi$, as follows:

$$\mathbf{T}_\xi = [\mathbf{Y}^1 \quad \mathbf{Y}^\xi] = [\mathbf{A}^T\mathbf{X}^1 \quad \mathbf{A}^T\mathbf{X}^\xi], \tag{46}$$

where $\xi (0 \leq \xi \leq 1)$ represents the intensity of the low-intensity images. Figures 2(b) and (c) demonstrate the training data $\mathbf{T}_\xi$ with two different values of ξ on the CK+ database.

Conventional manifold-learning methods map training samples, irrespective of how strong the expressing images are, as close as possible after transformation. This results in limited performance in terms of generalization. In our feature-generation algorithm, the subspace learning method, SLPM, is first applied to features extracted from high-intensity expressions. Then, features extracted from low-intensity expressions are mapped to the learned subspace. As observed in Fig. 3, features extracted from low-intensity expressions are located farther from the core samples (formed by high-intensity expressions) and near the boundary of the manifolds after the mapping.

Let $\{\boldsymbol{x}_{s_1}^0, \boldsymbol{x}_{s_2}^0, \ldots, \boldsymbol{x}_{s_p}^0\} \in \mathbb{R}^D$ be the set of FVs extracted from neutral face images, where $\boldsymbol{x}_{s_i}^0$ is the FV of the neutral face image belonging to the subject s_i, and p is the number of the subjects in the dataset. The expression images of the subject s_i are denoted as

$$\mathbf{X}_{s_i}^1 = [\boldsymbol{x}_{s_{i,1}}^1, \boldsymbol{x}_{s_{i,2}}^1, \ldots, \boldsymbol{x}_{s_{i,r}}^1], \tag{47}$$

where r is the number of expression images belonging to s_i and $x_{s_{i,j}}^1$ is the FV extracted from the j-th expression image of s_i. Then, the feature matrix for all the expressions is formed as follows:

$$\mathbf{X}_s^1 = [\mathbf{X}_{s_1}^1, \mathbf{X}_{s_2}^1, \ldots, \mathbf{X}_{s_p}^1]. \tag{48}$$

The proposed sample-generation method operates in the learned subspace. Thus, the FVs extracted from the neutral face images and the expression images are all mapped to the learned subspace using the projection matrix $\mathbf{A}$ learned from $\mathbf{X}_s^1$, as follows:

$$\mathbf{Y}^1 = \mathbf{A}^T\mathbf{X}_s^1 = [\mathbf{Y}_{s_1}^1, \mathbf{Y}_{s_2}^1, \ldots, \mathbf{Y}_{s_p}^1], \text{ and} \tag{49}$$

$$\mathbf{Y}^0 = \mathbf{A}^T\mathbf{X}_s^0 = [\boldsymbol{y}_{s_1}^0, \boldsymbol{y}_{s_2}^0, \ldots, \boldsymbol{y}_{s_p}^0]. \tag{50}$$

Equations (49) and (50) represent the set of FVs of high-intensity expressions and neutral expressions of all subjects, respectively, in the subspace.

The proposed feature-generation method generates low-intensity FVs based on vector-pairs selected from two different sets: (1) vector-pairs from $\mathbf{Y}_{s_i}^1$ and (2) vector-pairs from $\mathbf{Y}_{s_i}^1$ and $\boldsymbol{y}_{s_i}^0$. In the following sections, we will describe the feature-generation method with respect to two different vector-pairs.

2) Vector-pairs from $\mathbf{Y}_{s_i}^1$ and $\mathbf{y}_{s_i}^0$

Let $\overline{\mathbf{Y}}_{s_i}^{\theta_{ne}} = [\boldsymbol{y}_{s_{i,1\to0}}^{\theta_{ne}}, \boldsymbol{y}_{s_{i,2\to0}}^{\theta_{ne}}, \ldots, \boldsymbol{y}_{s_{i,r\to0}}^{\theta_{ne}}]$ be the feature matrix of possible low-intensity expressions with an intensity of $\theta_{ne} (0 < \theta_{ne} < 1)$ belonging to the subject s_i, where $\boldsymbol{y}_{s_{i,j\to0}}^{\theta_{ne}}$ is the corresponding low-intensity FV generated using $\boldsymbol{y}_{s_{i,j\to0}}^1$ and $\boldsymbol{y}_{s_i}^0$. In the rest of the paper, the arrow "$\to$" indicates the direction of the FVs to be generated, with 0 and 1 being a neutral face image and a face image with the highest intensity, respectively. $\boldsymbol{y}_{s_{i,j\to0}}^{\theta_{ne}}$ means that the FV is generated in the direction from $\boldsymbol{y}_{s_{i,j}}^1$ to $\boldsymbol{y}_{s_i}^0$ where $\boldsymbol{y}_{s_{i,j}}^1$ is the mapped FV extracted from the j-th expression image of s_i.

A set of FVs extracted from an expression video sequence, which starts from a neutral-expression face to the highest intensity of an expression, can be perceived as a path from the reference center, i.e. the neutral manifold, to a particular expression manifold wherein the distance of an expression manifold from the center is directly proportional to the intensity of the expression [56]. Therefore, for databases consisting of only static expression images, the feature matrix of possible low-intensity expressions can be obtained by assuming that the relation between the distance from $\boldsymbol{y}_{s_{i,j\to0}}^{\theta_{ne}}$ to $\boldsymbol{y}_{s_i}^0$ and the expression intensity is linear. As illustrated in Fig. 4(a), the low-intensity FV $\boldsymbol{y}_{s_{i,j\to0}}^{\theta_{ne}}$, belonging to s_i, can be computed as follows:

$$\boldsymbol{y}_{s_{i,j\to0}}^{\theta_{ne}} = \theta_{ne} \cdot \boldsymbol{y}_{s_{i,j}}^1 + (1 - \theta_{ne}) \cdot \boldsymbol{y}_{s_i}^0, \tag{51}$$

Figures 2(d) and (e) outline the training data with the feature generation using neutral images when $\theta_{ne} = 0.9$ and $\theta_{ne} = 0.7$, respectively. As seen in Fig. 2, both the absolute low-intensity FVs and the possible low-intensity FVs generated by linear interpolation have a similar structure.

3) Vector-pairs from $\mathbf{Y}_{s_i}^1$

The respective expression manifolds can be far from each other in the learned subspace. For this reason, more features between expression manifolds are also needed. In the previous section, we proposed the idea that the FVs extracted from low-intensity expression images should be distant from the corresponding manifold center, thus, this can enhance the generalizability of the learned manifold. Using a similar idea, more features that are distant from the manifold centers can be generated using vector-pairs from the feature matrix of high-intensity expressions of the same subject, $\mathbf{Y}_{s_i}^1$, as illustrated in Fig. 4(b). A FV $\boldsymbol{y}_{s_{i,j\to k}}^{\theta_{exp}}$, which lies on the line from the j-th expression-vector of s_i, $\boldsymbol{y}_{s_{i,j}}^1$, to the k-th expression-vector of s_i, $\boldsymbol{y}_{s_{i,k}}^1$, with a weight $\theta_{exp} (0 < \theta_{exp} < 1)$ can be computed as follows:

$$\boldsymbol{y}_{s_{i,j\to k}}^{\theta_{exp}} = \theta_{exp} \cdot \boldsymbol{y}_{s_{i,j}}^1 + (1 - \theta_{exp}) \cdot \boldsymbol{y}_{s_{i,j}}^1, \tag{52}$$

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

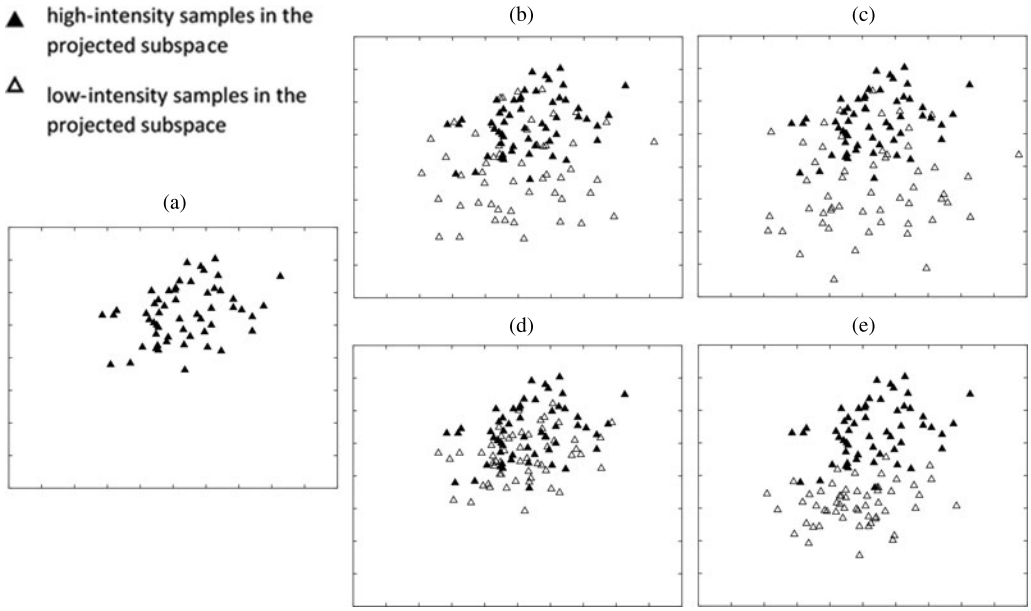


Fig. 2. Representation of the FVs of happiness (HA) on the CK+ database, after SLPM: (a) HA, i.e. high-intensity expression samples are applied to SLPM, (b) HA+ low intensity FV with $\xi = 0.9$, (c) HA+ low intensity FV with $\xi = 0.7$, (d) HA+ generated FV with $\theta_{ne} = 0.9$, and (e) HA+ generated FV with $\theta_{ne} = 0.7$.

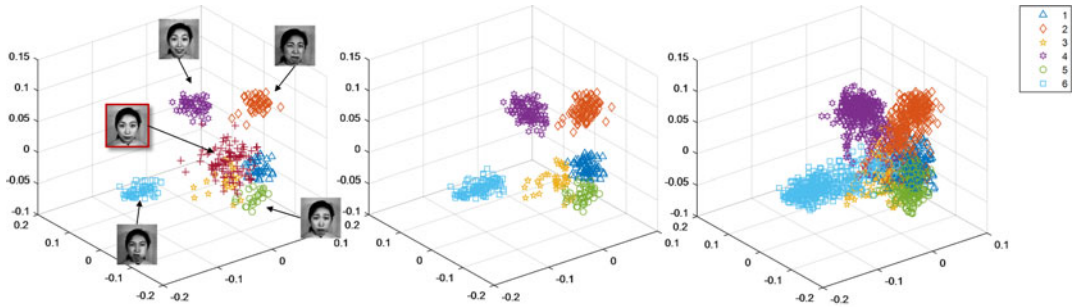


Fig. 3. Subspace learned using SLPM, with local descriptors "LP," based on the dataset named CK+: (a) the mapped features extracted from high-intensity expression images and neutral face images, (b) the mapped features extracted from high-intensity and low-intensity ($\xi = 0.7$) images, and (c) the mapped features extracted from high-intensity and low-intensity ($\xi = \{0.9, 0.8, 0.7, 0.6, 0.5, 0.4\}$) images.

Suppose that $c_j = l(\boldsymbol{y}^1_{s_{i,j}})$ and $c_k = l(\boldsymbol{y}^1_{s_{i,k}})$ are the expression classes of the j-th and the k-th expression vectors, respectively, and n_{i,c_j} and n_{i,c_k} are the number of expression-vectors of expression classes c_j and c_k, respectively, belonging to subject s_i. Then, a total of n_{i,c_j} n_{i,c_k} FVs can be generated. The feature matrix consisting of the generated features using the pairs from $\mathbf{Y}^1_{s_i}$ can be denoted as follows:

$$\begin{aligned}\overline{\mathbf{Y}}^{\theta_{exp}}_{s_{i,exp}} =& [\boldsymbol{y}^{\theta_{exp}}_{s_{i,1\to 2}}, \boldsymbol{y}^{\theta_{exp}}_{s_{i,2\to 3}}, \ldots, \boldsymbol{y}^{\theta_{exp}}_{s_{i,1\to r}}, \ldots, \\ & \boldsymbol{y}^{\theta_{exp}}_{s_{i,r\to 1}}, \boldsymbol{y}^{\theta_{exp}}_{s_{i,r\to 2}}, \ldots, \boldsymbol{y}^{\theta_{exp}}_{s_{i,r-1\to r}}].\end{aligned} \tag{53}$$

The training matrix, $\mathbf{T}_\theta$, is updated to $\overline{\mathbf{T}}_\theta$, which is used as a static database, as follows:

$$\overline{\mathbf{T}}_\theta = [\mathbf{Y}^1 \quad \overline{\mathbf{Y}}^{\theta_{ne}}_{ne} \quad \overline{\mathbf{Y}}^{\theta_{exp}}_{exp}], \tag{54}$$

where $\overline{\mathbf{Y}}^{\theta_{ne}}_{ne} = [\overline{\mathbf{Y}}^{\theta_{ne}}_{s_{1,ne}}, \overline{\mathbf{Y}}^{\theta_{ne}}_{s_{2,ne}}, \ldots, \overline{\mathbf{Y}}^{\theta_{ne}}_{s_{p,ne}}]$ and $\overline{\mathbf{Y}}^{\theta_{exp}}_{exp} = [\overline{\mathbf{Y}}^{\theta_{exp}}_{s_{1,exp}}, \overline{\mathbf{Y}}^{\theta_{exp}}_{s_{2,exp}}, \ldots, \overline{\mathbf{Y}}^{\theta_{exp}}_{s_{p,exp}}]$. In our experiments, we vary the θ_{ne} and the θ_{exp} values from 0.7 to 0.9. Algorithm 1 lists the overall flow of the proposed algorithm.

When an FV is generated, it is checked whether or not it is closest to its corresponding manifold class. Furthermore, the FVs are generated solely for the pairs of clusters that are in close proximity to each other in the learned subspace.

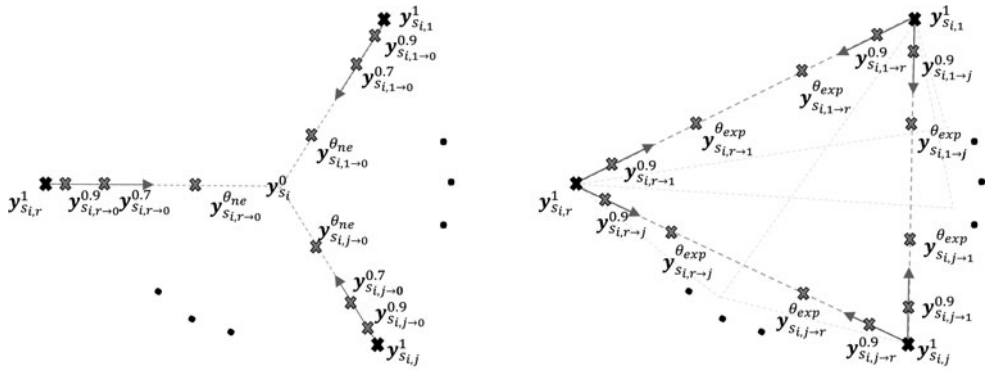
V. EXPERIMENTAL SET-UP AND RESULTS

A) Experimental setup

In our experiments, four facial-expression databases, (1) Bahcesehir University Multilingual Affective Face Database (BAUM-2) [57], (2) Extended Cohn-Kanade (CK+) [58] database, (3) Japanese Female Facial Expression (JAFFE) [59] database, and (4) Taiwanese Facial Expression Image Database (TFEID) [60], were used to evaluate the robustness and performances of the different subspace-learning methods. Following is a brief description of each of the four databases.

The *BAUM-2 multilingual* database [57] consists of short videos extracted from movies. In our experiments, an image

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press



(a) (b)

Fig. 4. Representation of the sample-generation process based on (a) FVs extracted from high-intensity images and neutral-face images, and (b) FVs extracted from high-intensity images.

dataset, namely BAUM-2i, consisting of images with peak expressions from the videos in BAUM-2, is considered. There are 829 face images from 250 subjects in the BAUM-2i static expression dataset, which express six basic emotions. However, only 536 of them, which have their facial-feature points provided, are considered in our experiments. Since the BAUM-2 database was created by extracting images from movies, the images are close to real-life conditions (i.e. under pose, age, illumination variations, etc.), and are more challenging than those in acted databases.

The *Extended Cohn-Kanade* (CK+) [58] database contains a total of 593 posed sequences across 123 subjects, of which 304 of the sequences have been labeled with one of the six discrete emotions, which are anger, disgust, fear, happiness, sadness, and surprise. Each sequence starts with a neutral face and ends with a frame of peak expression. The last frame of each sequence, and the first frames of the sequences that have unique subject labels, as well as their landmarks provided, are used for recognition. There are a total of 414 face images. Note that some of the first frames are also discarded, because the expressed emotions are of low intensity. We further split the selected images into five folds, based on the identity information. The five-fold identity-independent cross-validation strategy is adopted to evaluate SLPM on CK+. Specifically, 80% of the images are used for training, while the remaining 20% of images are further split into two equal sub-sets for validation and testing, respectively. The final performance is evaluated by averaging the recognition accuracy over the five runs.

JAFFE [59] consists of 213 images from 10 Japanese females, which express six basic emotions – anger, disgust, fear, happiness, sadness, and surprise – and neutral. JAFFE is also a widely used acted database, which means that it was recorded in a controlled environment. Similar to CK+, we adopt the five-fold identity-independent cross-validation strategy to evaluate the proposed algorithm on JAFFE.

The *TFEID* database [60] contains 268 images, with the six basic expressions and the neutral expression, from 40 Taiwanese subjects. Similar to CK+ and JAFFE, TFEID is also an acted database.

Each of the above-mentioned databases has its own characteristics. Table 6 shows the number of images for each expression class for the different databases. Although some of the databases also have the contempt expression, only the six basic prototypical facial expressions (i.e. anger, disgust, fear, happiness, sadness, and surprise), as well as the neutral facial expression, are considered in our experiments. Note that neutral facial expression has been used only for creating FVs of low-intensity expressions.

Table 6. Comparison of the number of images for different expression classes in the databases used in our experiments

Emotion	BAUM-2	CK+	JAFFE	TFEID
Anger	80	45	30	34
Disgust	32	59	29	40
Fear	35	25	32	40
Happiness	139	69	31	40
Sadness	68	28	31	39
Surprise	83	82	30	36
Neutral	99	106	30	39
Total	536	414	213	268

Subspace-learning methods are often applied to FVs formed by the pixel intensities of face images. In our method, features are first extracted using the state-of-the-art local descriptors, and then a subspace-learning method is applied for manifold learning and dimensionality reduction. The usual way of using local descriptors is to divide a face image into a number of overlapping or non-overlapping regions, then extract features from these regions, and finally concatenate them to form a single FV. In this way, local information, as well as spatial information, can be obtained. Another way of using local descriptors is to consider only the regions that have more salient information about the considered expression classes. Following this idea, features extracted from the eye and mouth regions are used in [61], which showed that features extracted from these regions only can achieve higher recognition rates than those extracted by dividing face images into sub-regions.

In the experiments for evaluating SLPM, the face images from the different databases are all scaled to the size of 126×189 pixels, with a distance of 64 pixels between the two eyes. To determine the eye and mouth windows, the facial landmarks, i.e. the eyes and mouth corners, are used.

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

If facial landmarks are not provided for a database, the required facial-feature points are detected by the face alignment method [62]. The eye window and the mouth window are further divided into 12 and 8 sub-regions, respectively. The nearest neighbor classifier and SVM with linear kernel are used in the experiments.

B) Implementation details of SLP-CNN

Different from of SLMP, the facial images, used to train and evaluate SLP-CNN, are aligned by the method in [62]. Each aligned face is cropped and resized to 100 × 100. The training samples are generated by randomly cropping a 90 × 90 region from each aligned facial image. Random rotation at $\{5^\circ, 10^\circ, 15^\circ, -5^\circ, -10^\circ, -15^\circ\}$ and random mirroring are applied to the cropped images for data augmentation.

We implement SLP-CNN with PyTorch [63], based on the network architecture presented in Table 5. We adopt the stochastic gradient descent [64] optimizer to minimize the objective function defined in equation (44), with the hyperparameters λ, β, and $k_w = k_b = k$ empirically set to 0.1, 0.5, and 20, respectively. The weight decay and momentum are set to 0.005 and 0.9, respectively. We train SLP-CNN for 600 epochs on a Nvidia GEFORCE GTX 2080 Ti GPU with the batch size of 128. The learning rate is initialized to 10^{-2}, and is decreased by a factor of 10 at the 200-th and the 400-th epoch. We detach the last softmax layer in the trained network, and employ the remaining parts as a feature extractor for the facial samples. Finally, linear SVMs with a one-against-one strategy, implemented by LibSVM [65], are applied for classification. Since the selected databases are relatively small, a trained deep neural network is easy to overfit on the training samples. Thus, we first pretrain the base network on a large-scale database, i.e. the Real-world Affective Faces database (RAF-DB) [27]. The RAF-DB database consists of over 30k greatly diverse facial expression images collected from the Internet, with the emotion labels provided. In the pretraining phase, we only utilize the images with one of the seven basic expressions from the training set of RAF-DB. In addition, we adopt the same pre-processing, augmentation, and training settings mentioned above to obtain the pretrained models.

To make fair comparisons, all the above-mentioned settings, including the network architecture, the optimizer, the learning rate, etc., are uniformly used in all the comparison methods. The hyperparameters of the comparison methods are set to the default values, according to the implementation provided by the original authors.

C) Experimental results

1) Evaluation on SLPM

In this section, we evaluate the performances of our proposed SLPM, using four different descriptors, on the four different databases. We also compare our method with four subspace-learning methods, as well as without using any subspace-learning method.

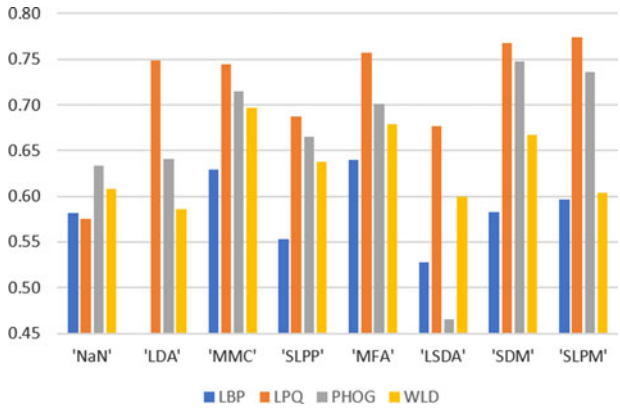


Fig. 5. Recognition rates of the different subspace methods, with different local descriptors, based on a combined dataset of BAUM-2, CK+, JAFFE, and TFEID.

First, the four acted databases, i.e. BAUM-2, CK+, JAFFE, and TFEID, are combined to form a single dataset, called COMB4, so that we can better measure the general performances of the different subspace-learning methods and the descriptors.

Figure 5 shows that MFA, SDM, and SLPM are the three best subspace-learning methods, which outperform the other subspace-learning methods. The LPQ local descriptor achieves the highest recognition rates, for the different subspace-learning methods, on COMB4. Therefore, the subspace-learning methods, MFA and SDM, with the local descriptor, LPQ, are chosen to further compare to the performance of the proposed method on each of the individual datasets. In Fig. 5, we can observe that SDM outperforms most of the subspace-learning methods, except SLPM, because the intra-class spread is adjustable. Furthermore, SDM is also computationally simpler than the other compared methods, but it does not incorporate the local geometry of the data. In our proposed method, information about local structure is incorporated into the objective function. Thus, SLPM can achieve higher recognition rates than SDM.

Figure 6 shows the recognition rates of SLPM on COMB4, with the dimensionality of the subspace varied. The results show that SLPM has converged to its highest recognition rate, when the dimensionality is lower than 10. In other words, our method is still very effective even at a low dimensionality. Based on these results, we set the subspace dimensionality at 11 in the rest of the experiments.

To investigate the effect of the use of images of expression with low intensities, several experiments have been conducted on the CK+ database. As shown in Table 7, the recognition rate is the highest when $\xi = 0.9$. Tables 8 and 9 show the recognition rates of the three subspace-learning methods, MFA, SDM, and SLPM, as well as SLPM, using feature generation with different θ_{exp} and θ_{ne} values, with the LPQ descriptor, on the four different databases using the nearest neighbor classifier and SVM classifier, respectively. It can be found that SLPM achieves the best classification performance again, when compared to the other methods. The classification performance is further improved by

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

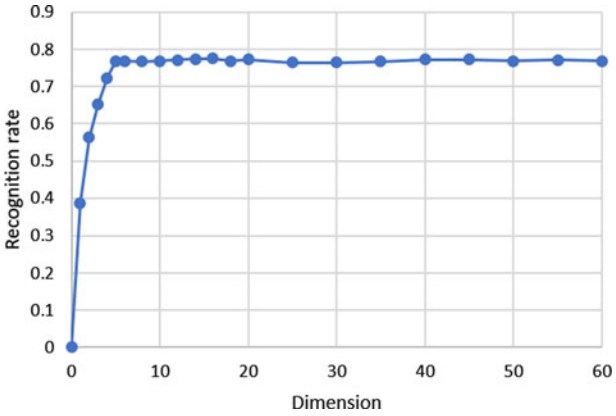


Fig. 6. Recognition rates of our proposed method in terms of different dimensions.

Table 7. Comparison (%) of recognition rates obtained by using low-intensity images with different ξ values on the CK+ database, using the LPQ feature

Methods	CK+
SLPM	94.81
SLPM with $\xi = 0.9$	94.81
SLPM with $\xi = 0.8$	95.45
SLPM with $\xi = 0.7$	94.16
SLPM with $\xi = 0.6$	91.88

Table 8. Comparison (%) of subspace learning methods on different datasets, with the LPQ descriptor being used with the nearest neighbor classifier

	BAUM-2	CK+	JAFFE	TFEID
MFA [11]	62.01	93.83	89.07	91.70
SDM [7]	62.01	93.51	89.07	92.58
SLPM	62.93	94.81	90.71	93.45
SLPM with $\theta_{exp} = 0.9, \theta_{ne} = 0.9$	**63.62**	94.81	91.26	93.45
SLPM with $\theta_{exp} = 0.8, \theta_{ne} = 0.8$	62.93	**96.10**	91.26	**94.32**
SLPM with $\theta_{exp} = 0.7, \theta_{ne} = 0.7$	63.13	95.13	**91.80**	93.89
SLPM with $\theta_{exp} = 0.6, \theta_{ne} = 0.6$	62.01	94.16	90.71	93.45

The best results are highlighted in **bold**.

up to 2%, when feature generation is employed. Furthermore, as observed in Tables 8 and 9, the nearest neighbor classifier outperforms the SVM classifier in most of the databases. Finally, additional experiments were conducted to validate the efficiency of the proposed subspace learning methods. Table 10 tabulates the runtimes in milliseconds for each of the subspace learning methods. We can see that SLPM is twice as fast as MFA, which solves the generalized eigenvalue problem instead of calculating eigenvalue decomposition like SLPM. SDM is much slower than MFA and SLPM.

2) Evaluation on SLP-CNN

Comparison with state-of-the-art methods:

We first evaluate SLP-CNN by comparing it with other state-of-the-art deep subspace-learning methods. Specifically, we compare the proposed SLP loss with the other regularizers, including the Center loss [28], the Island loss

Table 9. Comparison (%) of subspace learning methods on different datasets, with the LPQ descriptor being used with the SVM classifier

	BAUM-2	CK+	JAFFE	TFEID
MFA [11]	61.10	92.21	**91.26**	91.70
SDM [7]	60.18	92.21	89.62	92.58
SLPM	63.16	92.53	89.62	93.01
SLPM with $\theta_{exp} = 0.9, \theta_{ne} = 0.9$	**63.84**	92.86	**91.26**	94.76
SLPM with $\theta_{exp} = 0.8, \theta_{ne} = 0.8$	62.47	93.83	**91.26**	**95.20**
SLPM with $\theta_{exp} = 0.7, \theta_{ne} = 0.7$	62.24	**94.48**	89.07	94.32
SLPM with $\theta_{exp} = 0.6, \theta_{ne} = 0.6$	61.56	**94.48**	88.52	94.32

The best results are highlighted in **bold**.

Table 10. Comparison of the runtimes (in ms) required by the different subspace learning methods (MFA, SDM, and SLPM) on different datasets, with the LPQ descriptor used

	BAUM-2	CK+	JAFFE	TFEID
MFA [11]	96	69	45	51
SDM [7]	151	133	120	118
SLPM	**65**	**37**	**23**	**25**

The best results are highlighted in **bold**.

[26], and the LP loss [27]. The results are summarized in Table 11, in which the reference model, i.e. Base-CNN, is trained under the standard softmax loss. It can be observed that SLP-CNN consistently outperforms Base-CNN on the four selected databases, which demonstrates that the SLP loss can effectively enhance the discriminative power of the learned features. Compared with the other deep subspace learning regularizers, SLP-CNN surpasses the Center loss and the LP loss by about 2 and 0.5% on the four databases, respectively. This is because the Center and the LP regularizers only penalize the intra-class distance, while the SLP loss considers both the intra-class and the inter-class variations. However, SLP-CNN performs slightly worse than the Island regularizer. The Island loss characterizes the class spread across the whole dataset, while the SLP loss mainly focuses on the local neighborhoods. Therefore, as introduced in Section II.B), the Island loss learns more discriminative features, while the SLP loss can enhance the model generalization. We will further validate this point in the following sections. To better illustrate the performance of the deep subspace regularizers, we visualize the learned deep features of the CK+ testing images. The results are presented in Fig. 7. We adopt t-SNE [66] to show the 2000-dimensional deep features. It is obvious from the figure that the subspace-learning methods can force the samples with the same expression closer to each other in the latent space, which shows the effectiveness of subspace learning in enhancing the feature discriminative power.

Generalization test:

To show the generalization ability of the proposed algorithm, we further conduct the generalization test on SLP-CNN. Specifically, we pretrain the CNN model on the RAF-DB database, and directly employ it to extract features from the images in the four selected databases without fine-tuning. Thus, this is a cross-database evaluation. We also

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

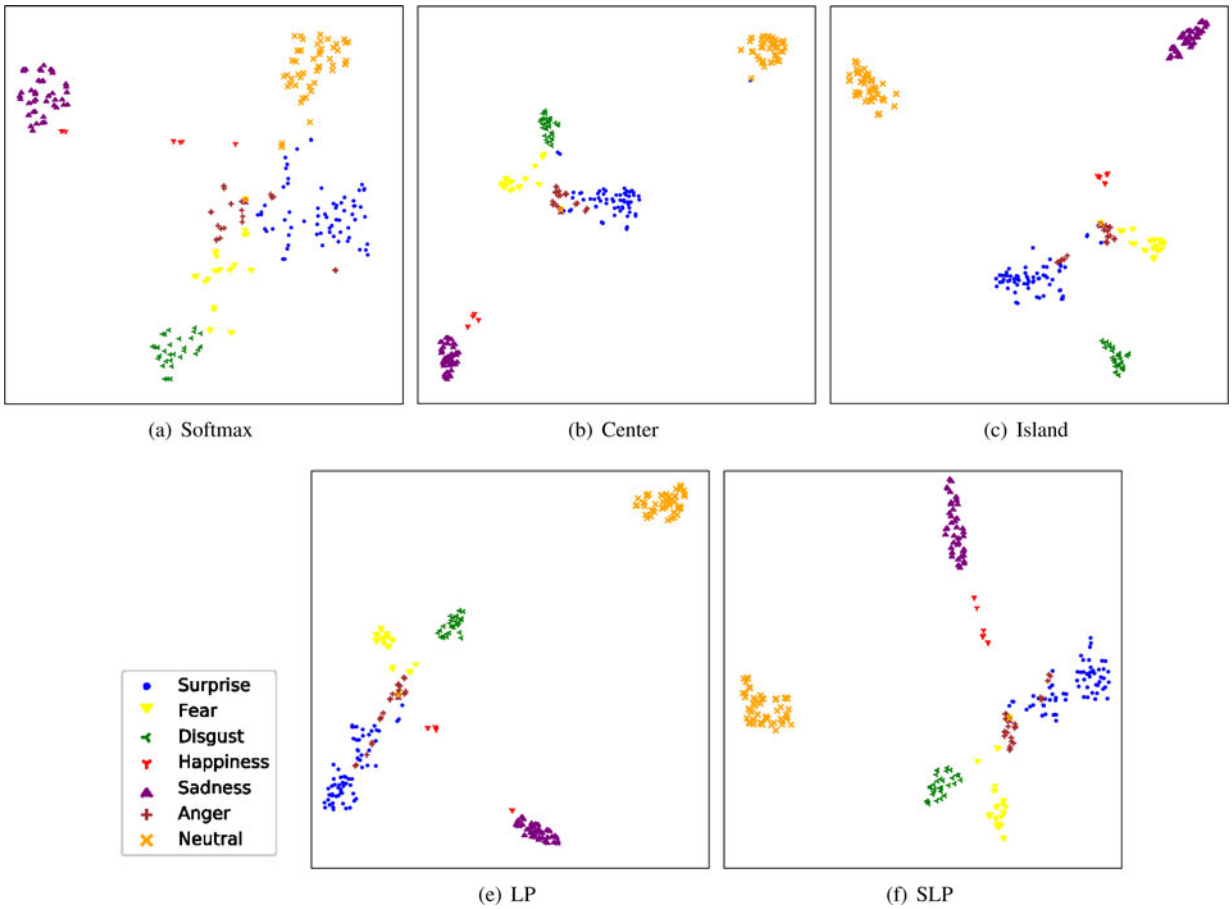


Fig. 7. Visualization of the deeply learned features extracted from the samples of one testing fold in CK+, based on the different methods. (a) Softmax. (b) Center. (c) Island. (d) LP. (e) SLP.

Table 11. Comparison in terms of the recognition rates (%) of the deep subspace-learning methods on different datasets

	BAUM-2	CK+	JAFFE	TFEID	Ave.
Base-CNN	63.84	92.53	90.71	94.32	85.35
Center loss [28]	65.25	94.48	92.35	95.20	86.82
Island loss [26]	**67.03**	**96.73**	**94.52**	**96.07**	**88.58**
LP loss [27]	66.37	96.10	92.35	94.76	87.39
SLP loss	66.61	96.42	93.44	95.64	88.03

The best results are highlighted in **bold**.

compare the deep subspace-learning methods with other FER algorithms that train and test on the same databases. In other words, we evaluate and compare the deep subspace-learning methods in a more challenging condition. The results are tabulated in Table 12.

It can be seen that those deep subspace models, without fine-tuning their feature extraction CNNs, still generalize well on the target datasets. More importantly, the locality-preserving strategy, i.e. the LP and SLP losses, can obtain better generalization ability than the center-based methods, i.e. the Center loss and the Island loss, as the LP and SLP CNNs consistently outperform the Center and Island CNNs on the four target databases. Compared with the other FER methods that train and test on the same domain, the proposed SLP-CNN can produce comparable or even higher recognition accuracy, which demonstrates its effectiveness when facing cross-domain samples in real-world scenarios.

Hyperparameters' sensitivity analysis: We investigate the sensitivity of the hyperparameters k, λ, and β in equations (43) and (44), which significantly affect the trade-off performance. We first fix $\lambda = 0.1$ and $\beta = 0.5$, and explore the effect of k, which controls the number of nearest-neighbor samples used to compute the SLP loss in a mini batch. Figure 8 reports the results of the recognition accuracy on the four databases, as well as the averaged accuracy over the four databases, in terms of different values of the different hyperparameters. We can observe from Fig. 8(a) that the optimal k appears at around 20. Similarly, we explore the model sensitivity to λ and β, and present the results in Figs 8(b) and 8(c), respectively. Based on the results, we obtain the settings for the hyperparameters as we introduced in Section V.B), i.e. $k = 20$, $\lambda = 0.1$, and $\beta = 0.5$.

VI. CONCLUSION

In this paper, we have given an overview of subspace analysis methods, and extended them to deep learning. We have proposed a subspace-learning method, named SLPM, which uses the neighborhood and class information to construct a projection matrix for mapping high-dimensional

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

Table 12. Generalization test (%) on the different deep subspace regularizers

Methods	BAUM-2	Methods	CK+	Methods	JAFFE	Methods	TFEID
WLD [67]	54.51	3DCNN [70]	92.4	Sobel [73]	92.60	REC [76]	85.45
LBP [68]	58.32	DTGN [71]	92.35	SAE [74]	94.10	LMBP [77]	90.49
LDP [69]	58.99	IACNN [72]	95.37	DCNN [75]	**97.71**	MPC [78]	92.54
Base CNN	57.65	Base CNN	89.37	Base CNN	87.43	Base CNN	88.62
Center loss	59.47	Center loss	92.86	Center loss	89.62	Center loss	89.94
Island loss	60.63	Island loss	94.48	Island loss	90.17	Island loss	91.25
LP loss	61.18	LP loss	**95.78**	LP loss	91.26	LP loss	93.01
SLP loss	**62.24**	SLP loss	**95.78**	SLP loss	91.26	SLP loss	**93.89**

The deep subspace-learning methods are pretrained on RAF-DB [27] without further fine-tuning the CNN feature extractors on the target databases. The best results are highlighted in **bold**.

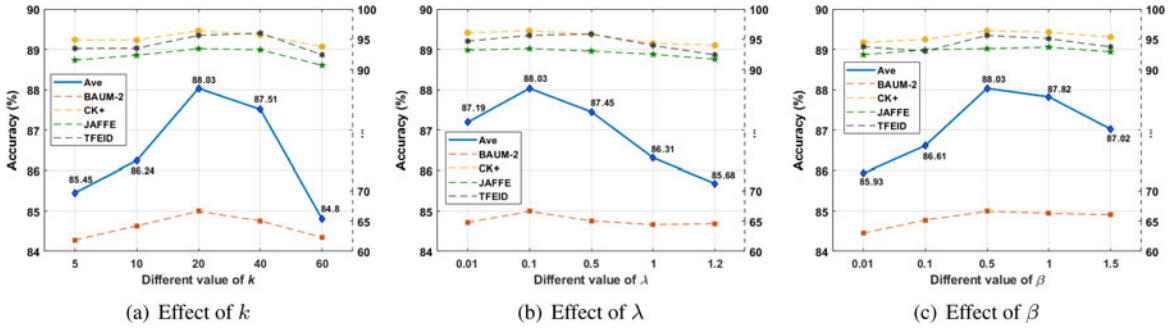


Fig. 8. Sensitivity analysis on the hyperparameters in the proposed SLP loss. (a) Effect of k. (b) Effect of λ. (c) Effect of β.

data to a meaningful low-dimensional subspace. We further reformulate the SLPM algorithm, and employ it as an additional regularization term for training a DCNN, named SLP-CNN, to enhance the discriminative power and generalization of the learned deep features. The difference between the within-class and between-class matrices is used to define the objective function, rather than the Fisher criteria, in order to avoid the singularity problem. Furthermore, a parameter β is added to control the within-class spread, so that the overfitting problem can be solved. The robustness and the generalizability of SLPM and SLP-CNN have been analyzed on four different databases, using four different state-of-the-art descriptors and two different classifiers, and SLPM has been compared with other subspace-learning methods. Moreover, we have proposed using the features of low-intensity expression images to learn a better manifold for each expression class. By taking advantage of domain-specific knowledge, we have proposed two methods of generating new low-intensity features in the subspace. Our experiment results have shown that SLPM outperforms the other subspace-learning methods, and is a good alternative to performing dimensionality reduction on high-dimensional datasets. Our experiment results have also shown that the proposed feature-generation method can further increase the recognition rates.

ACKNOWLEDGEMENT

The research described in this paper was supported by the GRF Grant PolyU 15217719 (project code: Q73V) of the Hong Kong Research Grants Council.

REFERENCES

[1] Mansour, H.; Rane, S.; Vetro, A.: Dimensionality reduction of visual features for efficient retrieval and classification. *APSIPA Trans. Signal Inf. Process.*, **5** (2016), e14.

[2] Hotelling, H.: Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.*, **24** (6) (1933), 417.

[3] Kak, A.C.; Martinez, A.: PCA versus LDA. *IEEE Trans. Pattern Anal. Mach. Intell.*, **23** (2) (2001), 228–233.

[4] Fisher, R.A.: The use of multiple measurements in taxonomic problems. *Ann. Eugen.*, **7** (2) (1936), 179–188.

[5] Raudys, S.J.; Jain, A.K.: Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Trans. Pattern Anal. Mach. Intell.*, **13** (3) (1991), 252–264.

[6] Li, H.; Jiang, T.; Zhang, K.: Efficient and robust feature extraction by maximum margin criterion. *IEEE Trans. Neural. Netw.*, **17** (1) (2006), 157.

[7] Liu, R.; Gillies, D.F.: Overfitting in linear feature extraction for classification of high-dimensional image data. *Pattern Recognit.*, **53** (2016), 73–86.

[8] Tenenbaum, J.B.; De Silva, V.; Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science*, **290** (5500) (2000), 2319–2323.

[9] Roweis, S.T.; Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *science*, **290** (5500) (2000), 2323–2326.

[10] Belkin, M.; Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering, in *Advances in Neural Information Processing Systems*, 2002, 585–591.

[11] Yan, S.; Xu, D.; Zhang, B.; Zhang, H.-J.; Yang, Q.; Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, **29** (1) (2006), 40–51.

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

[12] He, X.; Niyogi, P.: Locality preserving projections, in *Advances in Neural Information Processing Systems*, 2004, 153–160.

[13] Shan, C.; Gong, S.; McOwan, P.W.: Appearance manifold of facial expression, in *International Workshop on Human-Computer Interaction*, Springer, 2005, 221–230.

[14] Wong, W.K.; Zhao, H.: Supervised optimal locality preserving projection. *Pattern Recognit.*, **45** (1) (2012), 186–197.

[15] Chao, W.-L.; Ding, J.-J.; Liu, J.-Z.: Facial expression recognition based on improved local binary pattern and class-regularized locality preserving projection. *Signal Process.*, **117** (2015), 1–10.

[16] Wang, H.; Chen, S.; Hu, Z.; Zheng, W.: Locality-preserved maximum information projection. *IEEE Trans. Neural. Netw.*, **19** (4) (2008), 571–585.

[17] Li, B.; Huang, D.-S.; Wang, C.; Liu, K.-H.: Feature extraction using constrained maximum variance mapping. *Pattern Recognit.*, **41** (11) (2008), 3287–3294.

[18] Cai, D.; He, X.; Zhou, K.; Han, J.; Bao, H.: Locality sensitive discriminant analysis. *IJCAI*, **2007** (2007), 1713–1726.

[19] Jia, Y.; Liu, H.; Hou, J.; Kwong, S.: Clustering-aware graph construction: A joint learning perspective. *IEEE Trans. Signal Inf. Process. Netw.*, **6** (2020), 357–370.

[20] Jia, Y.; Liu, H.; Hou, J.; Kwong, S.: Pairwise constraint propagation with dual adversarial manifold regularization. *IEEE Trans. Neural Netw. Learn. Syst.*, **31** (12) (2020), 5575–5578.

[21] Wang, S.; Lu, J.; Gu, X.; Du, H.; Yang, J.: Semi-supervised linear discriminant analysis for dimension reduction and classification. *Pattern Recognit.*, **57** (2016), 179–189.

[22] Zhang, J.; Yu, J.; You, J.; Tao, D.; Li, N.; Cheng, J.: Data-driven facial animation via semi-supervised local patch alignment. *Pattern Recognit.*, **57** (2016), 1–20.

[23] Ptucha, R.; Savakis, A.: Manifold based sparse representation for facial understanding in natural images. *Image Vis. Comput.*, **31** (5) (2013), 365–378.

[24] Jia, Y.; Hou, J.; Kwong, S.: Constrained clustering with dissimilarity propagation-guided graph-Laplacian PCA. *IEEE Trans. Neural Netw. Learn. Syst.*, (2020), 1–13.

[25] Zhao, R.; Liu, T.; Xiao, J.; Lun, D.P.; Lam, K.-M.: Deep multi-task learning for facial expression recognition and synthesis based on selective feature sharing, in *2019 IEEE International Conference on Pattern Recognition (ICPR)*, 2020.

[26] Cai, J.; Meng, Z.; Khan, A.S.; Li, Z.; O'Reilly, J.; Tong, Y.: Island loss for learning discriminative features in facial expression recognition, in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, IEEE, 2018, 302–309.

[27] Li, S.; Deng, W.: Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. *IEEE Trans. Image Process.*, **28** (1) (2018), 356–370.

[28] Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y.: A discriminative feature learning approach for deep face recognition, in *European Conference on Computer Vision*, Springer, 2016, 499–515.

[29] Kamgar-Parsi, B.; Lawson, W.; Kamgar-Parsi, B.: Toward development of a face recognition system for watchlist surveillance. *IEEE Trans. Pattern Anal. Mach. Intell.*, **33** (10) (2011), 1925–1937.

[30] Gabrilovich, E.; Markovitch, S.: Feature generation for text categorization using world knowledge. *IJCAI*, **5** (2005), 1048–1053.

[31] Gader, P.D.; Khabou, M.A.: Automatic feature generation for handwritten digit recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18** (12) (1996), 1256–1261.

[32] Abboud, B.; Davoine, F.; Dang, M.: Facial expression recognition and synthesis based on an appearance model. *Signal Process. Image Commun.*, **19** (8) (2004), 723–740.

[33] Yu, J.; Bhanu, B.: Evolutionary feature synthesis for facial expression recognition. *Pattern Recognit. Lett.*, **27** (11) (2006), 1289–1298.

[34] Cai, D.; He, X.: Orthogonal locality preserving indexing, in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005, 3–10.

[35] Yi, Y.; Zhang, B.; Kong, J.; Wang, J.: An improved locality sensitive discriminant analysis approach for feature extraction. *Multimed. Tools. Appl.*, **74** (1) (2015), 85–104.

[36] Zhao, Z.; Han, J.; Zhang, Y., Bai, L.-f.: A new supervised manifold learning algorithm, in *International Conference on Image and Graphics*, Springer, 2015, 240–251.

[37] Yang, W.; Sun, C.; Zhang, L.: A multi-manifold discriminant analysis method for image feature extraction. *Pattern Recognit.*, **44** (8) (2011), 1649–1657.

[38] Chan, T.-H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y.: PCAnet: A simple deep learning baseline for image classification?. *IEEE Trans. Image Process.*, **24** (12) (2015), 5017–5032.

[39] Hu, G.; Yang, Y.; Yi, D.; Kittler, J.; Christmas, W.; Li, S.Z.; Hospedales, T.: When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition, in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, 142–150.

[40] Shi, Z.; Ye, Y.; Wu, Y.: Rank-based pooling for deep convolutional neural networks. *Neural Netw.*, **83** (2016), 21–31.

[41] He, K.; Zhang, X.; Ren, S.; Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, **37** (9) (2015), 1904–1916.

[42] Krizhevsky, A.; Sutskever, I.; Hinton, G.E.: Imagenet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems*, 2012, 1097–1105.

[43] He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 770–778.

[44] Ren, S.; He, K.; Girshick, R.; Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks, in *Advances in Neural Information Processing Systems*, 2015, 91–99.

[45] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A.: You only look once: Unified, real-time object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 779–788.

[46] Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L.: Beyond a Gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.*, **26** (7) (2017), 3142–3155.

[47] Zhao, R.; Lam, K.-M.; Lun, D.P.: Enhancement of a cnn-based denoiser based on spatial and spectral analysis, in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, 1124–1128.

[48] Kung, H.-W.; Tu, Y.-H.; Hsu, C.-T.: Dual subspace nonnegative graph embedding for identity-independent expression recognition. *IEEE Trans. Inf. Forensics. Sec.*, **10** (3) (2015), 626–639.

[49] Maas, A.L.; Hannun, A.Y.; Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. *Proc. ICML*, **30** (2013), 3.

[50] Ojala, T.; Pietikäinen, M.; Harwood, D.: A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.*, **29** (1) (1996), 51–59.

[51] Shan, C.; Gong, S.; McOwan, P.W.: Facial expression recognition based on local binary patterns: A comprehensive study. *Image Vis. Comput.*, **27** (6) (2009), 803–816.

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

[52] Ojansivu, V.; Heikkilä, J.: Blur insensitive texture classification using local phase quantization, in *International Conference on Image and Signal Processing*, Springer, 2008, 236–243.

[53] Bosch, A.; Zisserman, A.; Munoz, X.: Representing shape with a spatial pyramid kernel, in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, 2007, 401–408.

[54] Chen, J. *et al.*: WLD: A robust local image descriptor. *IEEE Trans. Pattern Anal. Mach. Intell.*, **32** (9) (2009), 1705–1720.

[55] Turan, C.; Lam, K.-M.: Histogram-based local descriptors for facial expression recognition (FER): A comprehensive study. *J. Vis. Commun. Image Represent.*, **55** (2018), 331–341.

[56] Chang, Y.; Hu, C.; Turk, M.: Manifold of facial expression. *AMFG*, (2003), 28–35.

[57] Erdem, C.E.; Turan, C.; Aydin, Z.: Baum-2: a multilingual audio-visual affective face database. *Multimed. Tools Appl.*, **74** (18) (2015), 7429–7459.

[58] Kanade, T.; Cohn, J.F.; Tian, Y.: Comprehensive database for facial expression analysis, in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, IEEE, 2000, 46–53.

[59] Lyons, M.; Akamatsu, S.; Kamachi, M.; Gyoba, J.: Coding facial expressions with Gabor wavelets, in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE, 1998, 200–205.

[60] Chen, L.-F.; Yen, Y.-S.: Taiwanese facial expression image database, *Brain Mapping Laboratory, Institute of Brain Science, National Yang-Ming University*, 2007.

[61] Turan, C.; Lam, K.-M.: Region-based feature fusion for facial-expression recognition, in *2014 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2014, 5966–5970.

[62] Bulat, A.; Tzimiropoulos, G.: How far are we from solving the 2D & 3D face alignment problem?(and a dataset of 230,000 3D facial landmarks), in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 1021–1030.

[63] Paszke, A. *et al.*: Automatic differentiation in Pytorch, in Neural Information Processing Systems (NIPS) Workshop Autodiff, 2017.

[64] Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G.: On the importance of initialization and momentum in deep learning, in *International Conference on Machine Learning*, 2013, 1139–1147.

[65] Chang, C.-C.; Lin, C.-J.: Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. (TIST)*, **2** (3) (2011), 1–27.

[66] Maaten, L. v. d.; Hinton, G.: Visualizing data using t-sne. *J. Mach. Learn. Res.*, **9** (2008), 2579–2605.

[67] Chen, J. *et al.*: WLD: A robust local image descriptor. *IEEE Trans. Pattern Anal. Mach. Intell.*, **32** (9) (2009), 1705–1720.

[68] Ojala, T.; Pietikäinen, M.; Harwood, D.: A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.*, **29** (1) (1996), 51–59.

[69] Jabid, T.; Kabir, M.H.; Chae, O.: Robust facial expression recognition based on local directional pattern. *ETRI. J.*, **32** (5) (2010), 784–794.

[70] Liu, M.; Li, S.; Shan, S.; Wang, R.; Chen, X.: Deeply learning deformable facial action parts model for dynamic expression analysis, in *Asian Conference on Computer Vision*, Springer, 2014, 143–157.

[71] Jung, H.; Lee, S.; Yim, J.; Park, S.; Kim, J.: Joint fine-tuning in deep neural networks for facial expression recognition, in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, 2983–2991.

[72] Meng, Z.; Liu, P.; Cai, J.; Han, S.; Tong, Y.: Identity-aware convolutional neural network for facial expression recognition, in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, IEEE, 2017, 558–565.

[73] Hamester, D.; Barros, P.; Wermter, S.: Face expression recognition with a 2-channel convolutional neural network, in *2015 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2015, 1–8.

[74] Huang, B.; Ying, Z.: Sparse autoencoder for facial expression recognition, in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, IEEE, 2015, 1529–1532.

[75] Nwosu, L.; Wang, H.; Lu, J.; Unwala, I.; Yang, X.; Zhang, T.: Deep convolutional neural network for facial expression recognition using facial parts, in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, IEEE, 2017, 1318–1321.

[76] Gu, W.; Venkatesh, Y.; Xiang, C.: A novel application of self-organizing network for facial expression recognition from radial encoded contours. *Soft comput.*, **14** (2) (2010), 113–122.

[77] Goyani, M.M.; Patel, N.: Recognition of facial expressions using local mean binary pattern. *ELCVIA: Electron Lett. Comput Vision Image Anal.*, **16** (1) (2017), 54–67.

[78] Farajzadeh, N.; Pan, G.; Wu, Z.: Facial expression recognition based on meta probability codes. *Pattern Anal. Appl.*, **17** (4) (2014), 763–781.

Cigdem Turan received her Ph.D. degree from the Department of Electronic and Information Engineering from the Hong Kong Polytechnic University in 2018. Her research interests include affective computing, facial behavior understanding, and responsible artificial intelligence. She is currently a postdoctoral researcher at TU Darmstadt, Germany.

Rui Zhao received his B.Sc. degree in electrical engineering from Xi'an Jiaotong University, China, in 2015, and his M.Sc. degree in electrical and electronic engineering from Imperial College London, UK, in 2017. He is currently a Ph.D. candidate in the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University. His research interests include image processing, computer vision, and facial expression recognition.

Kin-Man Lam received the Associateship in electronic engineering from The Hong Kong Polytechnic University (formerly called Hong Kong Polytechnic) in 1986, his M.Sc. degree in communication engineering from the Department of Electrical Engineering, Imperial College, in 1987, and his Ph.D. degree from the Department of Electrical Engineering, The University of Sydney, in 1996. From 1990 to 1993, he was a Lecturer with the Department of Electronic Engineering, The Hong Kong Polytechnic University, where he joined the Department of Electronic and Information Engineering, as an Assistant Professor in October 1996, became an Associate Professor in 1999, and has been a Professor since 2010.He was the Director-Student Services and the Director-Membership

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press

Services of the IEEE Signal Processing Society between 2012 and 2014, and between 2015 and 2017, respectively. He was an Associate Editor of *IEEE Transactions on Image Processing* between 2009 and 2014, and *Digital Signal Processing* between 2014 and 2018. He was also an Editor of *HKIE Transactions* between 2013 and 2018, and an Area Editor of the *IEEE Signal Processing Magazine* between 2015 and 2017. Currently, he is the VP-Publications of the Asia-Pacific Signal and Information Processing Association (APSIPA). Prof. Lam serves as an Associate Editor of *APSIPA Transactions on Signal and Information Processing* and *EURASIP International Journal on Image and Video Processing*. His current research interests include human face recognition, image and video processing, and computer vision.

Xiangjian He received his Ph.D. degree in computer science from the University of Technology Sydney (UTS), Australia, in 1999. He is currently a Full Professor and the Director of the Computer Vision and Pattern Recognition Laboratory, Global Big Data Technologies Centre, UTS.

https://doi.org/10.1017/ATSIP.2020.27 Published online by Cambridge University Press