EURASIP Journal on Image
and Video Processing

RESEARCH Open Access

A semi-supervised convolutional neural network based on subspace representation for image classification

Bernardo B. Gatto[1,2*], Lincon S. Souza[3], Eulanda M. dos Santos[2], Kazuhiro Fukui[1,3], Waldir S. S. Júnior[2] and Kenny V. dos Santos[2]

*Correspondence:
bernardo@cvlab.cs.tsukuba.ac.jp
[1]Center for Artificial Intelligence Research (C-AIR), Tsukuba, Japan
[2]Federal University of Amazonas, Manaus, Brazil
Full list of author information is available at the end of the article

Abstract

This work presents a shallow network based on subspaces with applications in image classification. Recently, shallow networks based on PCA filter banks have been employed to solve many computer vision-related problems including texture classification, face recognition, and scene understanding. These approaches are robust, with a straightforward implementation that enables fast prototyping of practical applications. However, these architectures employ either unsupervised or supervised learning. As a result, they may not achieve highly discriminative features in more complicated computer vision problems containing variations in camera motion, object's appearance, pose, scale, and texture, due to drawbacks related to each learning paradigm. To cope with this disadvantage, we propose a semi-supervised shallow network equipped with both unsupervised and supervised filter banks, presenting representative and discriminative abilities. Besides, the introduced architecture is flexible, performing favorably on different applications whose amount of supervised data is an issue, making it an attractive choice in practice. The proposed network is evaluated on five datasets. The results show improvement in terms of prediction rate, comparing to current shallow networks.

Keywords: Subspace method, Shallow networks, Semi-supervised learning

1 Introduction

In supervised machine learning, classifiers employ labeled data to create models. However, in many practical situations, labeled data is often challenging and expensive to obtain, for example, real-world remote sensing [1], medical image analysis [2], and facial expression recognition [3]. Besides, the difficulty in finding specialists to label data in certain areas may lead projects to be unfeasible [4, 5]. In contrast, models based on unsupervised learning are generated from unlabeled data which, in some scenarios, is readily available, and can be obtained at low cost [6, 7]. For example, meteorological weather data, such as temperature and pressure, can be obtained inexpensively in environmental

Springer Open

preservation projects [8, 9]. In addition, unlabeled images and videos can be obtained in social networks and employed to train unsupervised machine learning models [10, 11].

There is often no consensus on how to employ labeled and unlabeled data in conjunction to improve machine learning models due to the large imbalance between labeled and unlabeled data [12, 13]. Therefore, most classification methods produce models based only on labeled datasets, neglecting unlabeled data. In order to solve this problem, there is in the literature a class of learning techniques called semi-supervised learning. This class may be categorized as supervised learning, though it also makes use of unlabeled data for training. In general, these techniques employ a large amount of unlabeled data with a small amount of labeled ones. Many studies show that this kind of combination can provide significant enhancement in learning accuracy over unsupervised learning [14, 15].

In the context of image classification, however, the supervised approach is dominant. Image classification is one of the central problems, covering a diverse range of applications including human-computer interaction [16, 17], image and video retrieval [18, 19], video surveillance [20, 21], biometrics [22, 23], and analysis of social media networks [24, 25]. Considering this context, deep learning methods, such as convolutional neural network (CNN), are currently the state-of-the-art in several applications [26–28].

The literature shows that deep learning [29, 30] has been employed as an alternative to handcrafted features for image classification, like Gabor features [31] and local binary patterns (LBP) [32, 33] for texture and face classification and scale-invariant feature transform (SIFT) [34] and histogram of oriented gradients (HOG) features [35] for object recognition [36, 37]. The central concept of deep learning is that all relevant information required for recognizing image patterns can be structured in an hierarchical model, which can be obtained through iterative learning of the training image patterns. When the amount of available data is large enough (e.g., ImageNet dataset [38]) and there are no computational resources restrictions, deep learning models outperform handcrafted features-based methods [39, 40].

Despite its success, the number of parameters to be trained in a typical deep learning model is huge, consequently requiring a large amount of data to be employed for training, which can lead to a high computational cost, even when computational resources equipped with GPU are available. As a result, the computational complexity required from most of the deep learning architectures prevents some computer vision applications to fully employ the capabilities of deep CNN.

As an alternative, shallow networks have been proposed to exploit the advantageous characteristics of deep learning models, while lightening the computational cost associated with its training. Although these networks hold hierarchical structures, their weights are obtained through non derivative methods, giving them a processing time advantage over the traditional deep network models by several orders of magnitude. For instance, in [41], a convolutional neural network with no pooling layers nor active functions and without end-to-end learning is proposed. Instead, PCA or LDA are employed to replace the convolutional kernels of a CNN. While presenting a simple architecture, this strategy exhibited performance comparable to the state-of-the-art for several image classification tasks. Other examples of similar solutions include LDA [42], Gabor and ICA [43].

Even though shallow networks have been successfully applied in various recognition tasks, such methods can only describe either supervised or unsupervised data and are not able to efficiently exploit both. This paper proposes a convolutional shallow network to solve this issue. In contrast to the conventional networks [41, 43], the filter banks employed by the proposed network are produced by both PCA and generalized difference subspaces (GDS) [44, 45], which preserve the discriminative information among different classes, generating more efficient representations.

Accordingly, the proposed network can operate on both labeled and unlabeled data, improving the performance when only small volumes of labeled data are available. This network is called dual flow subspace network (DFSNet), due to its flexibility in handling both learning paradigms. In addition to its advantages, semi-supervised learning is of theoretical interest, since it makes it possible to understand the mechanisms of human learning [46, 47].

Therefore, our work provides the following contributions:

1. We introduce a new type of filter bank based on GDS. Different from PCA, the filter banks produced by GDS can efficiently handle labeled data.
2. We introduce a semi-supervised shallow network based on PCA and GDS, presenting a flexible framework.

In summary, the organization of this work is as follows: Section 2 gives a brief review on shallow networks. Then, in Section 3, we develop the proposed semi-supervised neural network for image classification. Section 4 shows the advantages of DFSNet over current shallow networks by experimental results using CIFAR-10 and ETH-80 databases for object recognition, LFW and FERET databases for face recognition, and NYU Depth V1 database for scene recognition. Finally, conclusions and future work are discussed in the last section.

2 Related work

In this section, we provide a brief review on CNN-like shallow networks. This analysis is important in order to clarify the differences between DFSNet and current methods. In all these examples, the employed techniques can be conducted as CNN-like architectures based on local multistage filter banks [48]. The typical framework of these approaches is shown in Fig. 1. In this framework, the input images are processed by multiple layers, ranging from 2 to 4 layers, followed by a feature mapping and classification. In this section, we will discuss both supervised and unsupervised shallow networks.

PCANet [41] is an unsupervised shallow network based on CNN, where multistage filter banks are learned from the data as principal components at the local image patch level. In PCANet, the eigenvectors of the local patch covariance matrix are employed as filter banks for convolution and feature extraction, followed by binarization and blockwise histogramming. This straightforward shallow network works well in a variety of

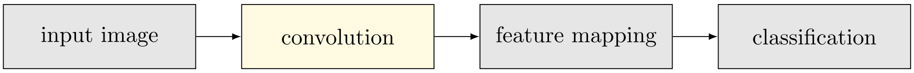


Fig. 1 Conceptual framework of the shallow networks investigated in this work. First, the input image is preprocessed by mean-removal or z-normalization. Then, the normalized image is processed by convolutional layers obtained by the reshaping of PCA or LDA basis vectors. The convolutional layers are obtained from either unsupervised or supervised approach. After that, a feature mapping strategy is applied, which consists of binarization and block-wise histogramming. Finally, classification is performed by KNN or SVM

image classification benchmarks, including handwritten and face recognition, achieving performance comparable to the state-of-the-art.

PCANet has been chosen as the main framework for several applications, including personal identification from ECG signal [49], traffic light recognition [50], remote sensing [51], medical image analysis [52], and automatic ship detection [53]. LDANet follows the same strategy used by PCANet and employs a similar architecture, with the difference that the filter banks used for convolution are obtained through the LDA basis vectors.

DCTNet [48] is an alternative to PCANet, which employs discrete cosine transform (DCT) as filter banks instead of PCA. DCTNet creates its filter banks by DCT, achieving a data-independent network, hence increasing the performance of the network. To reduce the computational complexity of the learning stages of this network, 2D DCT is also employed. Besides the low computational complexity, 2D DCT filter banks are independent of data, therefore generating a learning-free framework. DCTNet has been widely applied to several benchmarks of face databases and has shown performance equivalent or superior to PCANet.

Canonical correlation analysis network (CCANet) is introduced in [54], inspired by the flexibility and accuracy rate of wavelet scattering network (ScatNet) [55–57] and PCANet. It is also an unsupervised shallow network. On the other hand, different from ScatNet and PCANet, CCANet can handle images that are represented by two-view features, introducing more flexibility to the framework. Besides, CCANet produces the convolutional kernels by maximizing the correlation of the projected two-view variables. Therefore, the weights can reflect more discriminative information of the same object compared to PCANet and LDANet. The advantages of CCANet are as follows. First, CCANet can concurrently extract two-view features of a single image, which is assumed to minimize intra-class variance. Second is the reduced number of convolutional stages, in comparison to similar shallow networks. Also, as in PCANet and LDANet, CCANet does not require backpropagation algorithm to fine-tune its parameters. To demonstrate its effectiveness, CCANet was evaluated on several computer vision-related tasks in [54]. The results showed that CCANet outperformed PCANet and LDANet, for object, face, and handwritten digit recognition problems.

Although PCANet and similar networks achieve high recognition rates in several datasets, these networks may not extract discriminative features in more complicated computer vision problems, since PCA does not preserve the relationship between different classes, which can be useful in pattern classification problems. To lighten this issue, the discriminative canonical correlation network (DCCNet) [58] is introduced, where discriminative canonical correlations analysis (DCC) [59, 60] is employed as filter banks. Learning filters from DCC ensure that the network will provide discriminative features, generating more representative information by using supervised data. DCCNet was evaluated in four datasets, including objects and images of house numbers classification, outperforming PCANet, and LDANet in these tasks.

Despite its versatility, PCANet only works with unsupervised convolution filters, not making use of supervised information, when available. To solve this problem, orthogonal subspace network (OSNet) [61] is proposed to make use of supervised data. The central concept of OSNet is to express images as subspaces. In this scenario, the subspace representation is more compact than the traditional image set representation, since it selects the most relevant set of eigenvectors of an image set. To produce discriminative information, a space is computed to decorrelate the between class covariance matrix. Convolutional kernels of OSNet can be efficiently learned from class subspaces and directly employed to produce high discriminant features in a CNN-like architecture. Another benefit of subspace representation is that it requires less memory for storage and less processing time. The effectiveness of OSNet is shown in [61] by experiments using four databases, where OSNet outperformed PCANet.

In order to alleviate the high demand for storage space and computation required to learn deep features representation, a shallow network named compact feature representation (CFR-ELM) was proposed [62]. By using the extreme learning machine (ELM) under a shallow network design, this framework requires less storage space and computational resources, likewise the PCANet. This solution consists of the following steps: first, patch-based mean removal is employed, followed by an ELM auto-encoder (ELM-AE) feature extraction. Then, max pooling is used to compact the features. Finally, hashing and block-wise histogramming provide the post-processed features. The CFR-ELM was evaluated on MNIST, Coil-20/100, ETH-80, and CIFAR-10, demonstrating competitive results to the existing supervised shallow networks.

More recently, cosine convolutional kernel network (Cosine-CKN) [63] was proposed as an unsupervised convolutional network architecture that employs a kernel function designed by a convex combination of a (possibly uncountably infinite) number of cosine kernels. In contrast to the standard CKN, the introduced approximation is more related to CNN, where the inner product operator measures the similarity between filters and image patches. Different from the traditional CNN, Cosine-CKN has fewer hyperparameters, which makes its prototyping and training much faster. Cosine-CKN was evaluated on several datasets, including MNIST, CIFAR-10, C-Cube, and FERET. The experimental results demonstrated that this network reached better recognition accuracy and training time than PCANet and LDANet.

It is important to note that supervised shallow networks are dependent on the availability of labeled data and that unsupervised shallow networks do not have mechanisms to use labeled data, when available. In this case, a shallow network whose architecture allows the use of both labeled and unlabeled data may exhibit a significant advantage, since the network will be able to employ all types of data available, regardless of whether they are labeled or not. Besides, such flexibility also reflects the efficiency of the network, which is expected to provide competitive results concerning accuracy.

Finally, we should point out that PCA and LDA can be regarded as subspace-based methods, which is a class of learning techniques that employs subspaces to represent the data. Accordingly, we can introduce more sophisticated subspace methods such as GDS, where the discriminability of features is enhanced with the orthogonalization process of the different class subspaces. GDS has been employed in image set classification problems, achieving robustness to illuminations conditions. Due to its low computational cost, GDS is preferred compared to other supervised methods such as DCC or LDA. Another merit of using GDS is that it is robust to small sample size, which is a persistent problem in computer vision related problems [64].

By using supervised and unsupervised subspaces, we can introduce a shallow network capable of efficiently exploiting both learning paradigms, providing a very flexible architecture. After a thorough search of the relevant literature, we believe that this is the first work that introduces a semi-supervised shallow network based on subspaces for image classification. In Fig. 2, we show a conceptual schema of a semi-supervised shallow network for image classification. In the next section, we give details on the proposed architecture.

3 Proposed method

Inspired by shallow networks architectures, this section presents a semi-supervised network for image classification. The content of this section is organized as follows. First, we provide notations for the main concepts. Next, we explain the representation of the training images by patches. Then, we define the procedure of learning convolution filters through subspaces to generate supervised and unsupervised filter banks. After that, we describe the process of creating the final feature mapping.

3.1 Notations

In the context of this work, we will use the following notations. Scalars are denoted by upper case letters (e.g., N_u, M_u, N, M, K), vectors are denoted by lowercase letters, and matrices are denoted by boldface uppercase letters (e.g., v, $\mathbf{A}$, $\mathbf{X}_u$, $\mathbf{X}_s$). Calligraphic letters will be assigned to orthogonal basis vectors (e.g., $\mathcal{S}$, $\mathcal{M}$) as well as to filter banks $\mathcal{F}$. The set of filters $\{\phi_i\}_{i=1}^{D}$ contains D elements, e.g., $\{\phi_1, \ldots, \phi_D\}$. Given a matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$, $\mathbf{A}^T \in \mathbb{R}^{N\times M}$ denotes its transpose.

3.2 Problem setting

Let us consider a learning problem with two training sets $\mathbf{X}_u$ and $\mathbf{X}_s$, where $\mathbf{X}_u$ contains N_u unlabeled and $\mathbf{X}_s$ contains N_s labeled images of size $M \times N$.

The objective of DFSNet is to extract discriminative and representative structures in a way to maximize the classification result subject to its training data resources. Precisely, subspaces should be obtained from unsupervised and supervised training sets hierarchically, such that the features of different abstractions can be efficiently represented.

Then, given $\mathbf{X}_u$ and $\mathbf{X}_s$, we should implement a mechanism that produces $2Z$ filter banks, where Z denotes the number of convolutional layers in the network, in such manner that each layer will be equipped with an unsupervised $\mathcal{F}_u$ and a supervised $\mathcal{F}_s$ filter bank.

3.3 Representation by patches

We extract patches of size $K = K_1 \times K_2$ from $\mathbf{X}_u$ and $\mathbf{X}_s$. This procedure is performed by taking a patch around each pixel from each one of the $N_u + N_s$ training images. Here, we denote the set of unsupervised and supervised patches as $\mathbf{P}_u$ and $\mathbf{P}_s$, respectively. Given that each image patch will have size $K(= K_1 \times K_2)$, the sets $\mathbf{P}_u$ and $\mathbf{P}_s$ will then contain $M_u = N_uMN$ and $M_s = N_sMN$ patches, respectively.

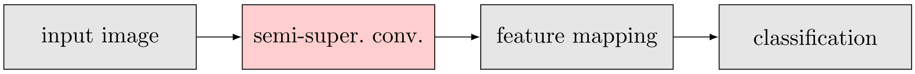


Fig. 2 Conceptual illustration of the proposed shallow network. DFSNet employs two distinct filters banks which work in complementary directions. In order to reduce the high dimensionality of the features and increase rotation invariance, the proposed method is followed by a feature mapping, as is done in most of the shallow networks. Then, the classification is performed by linear support vector machine

3.4 Producing unsupervised filter banks

The procedure for building unsupervised filters can be implemented in several ways. The literature points out that data-dependent filters (e.g. PCA, CCA) and data-independent filters (e.g. FFT, DCT, Wavelet transform) can be used to generate unsupervised filters. In our proposal, we will use PCA filter banks due to its flexibility in handling different applications [65, 66] and its fast training and test processing times.

The procedure to calculate PCA filters is carried as follows: we use the unsupervised patch set $\mathbf{P}_u = \{p_i \in \mathbb{R}^K\}_{i=1}^{M_u}$; the empirical mean vector is computed as $\overline{p} = \frac{1}{M_u}\sum_{i=1}^{M_u} p_i \in \mathbb{R}^K$ of $\mathbf{P}_u$. After that, we subtract the mean vector of each vector p_i to form the data centered set $\overline{\mathbf{P}_u}$. Once we obtain $\overline{\mathbf{P}_u}$, we can now build the feature matrix $\mathbf{A} \in \mathbb{R}^{M_u \times K}$ containing in its rows each element of $\overline{\mathbf{P}_u}$.

Once the feature matrix $\mathbf{A}$ is obtained, we can compute the autocorrelation matrix $\mathbf{C}_u = \mathbf{A}^T\mathbf{A} \in \mathbb{R}^{K\times K}$. Now that we are equipped with the autocorrelation matrix $\mathbf{C}_u$, we can move forward to calculate the matrix $\mathbf{U}_u$ of eigenvectors which diagonalizes the autocorrelation matrix $\mathbf{C}_u$:

$$\mathbf{D}_u = \mathbf{U}_u^{-1}\mathbf{C}_u\mathbf{U}_u, \tag{1}$$

In Eq. 1, $\mathbf{U}_u$ is an $K \times K$ orthogonal matrix, i.e., $\mathbf{U}_u\mathbf{U}_u^T = \mathbf{U}_u^T\mathbf{U}_u = \mathbf{I}$, where $\mathbf{I}$ is an $K \times K$ identity matrix. The columns of $\mathbf{U}_u$ that correspond to nonzero singular values compound a set of orthonormal basis vectors for the range of $\mathbf{C}_u$. $\mathbf{D}_u$ is the diagonal matrix of eigenvalues of $\mathbf{C}_u$.

The unsupervised filter bank $\mathcal{F}_u$ is defined by the first D_u vectors of $\mathbf{U}_u$ in descending order according to the eigenvalues of the matrix $\mathbf{D}_u$. Therefore, we define the unsupervised filter bank $\mathcal{F}_u$ as follows:

$$\mathcal{F}_u = \mathbf{U}_u\mathbf{R}_u, \tag{2}$$

where $\mathbf{R}_u$ is a $K \times K$ matrix containing 1 on its first D_u principal diagonal entries and 0 elsewhere. After this procedure, we should have an unsupervised filter bank $\mathcal{F}_u \in \mathbb{R}^{D_u \times K}$.

3.5 Producing supervised filter banks

There are also many types of supervised methods that can be employed to implement efficient supervised filters for DFSNet, such as LDA and DCC. In this work, we use GDS, which is suitable for the semi-supervised problem setting since it can work well with even a small quantity of supervised data. This problem setting, well known as small sample size problem, is very challenging for LDA and DCC due to its inability to estimate the within-class scatter matrix adequately in such circumstances. In contrast, GDS avoids this issue by introducing the subspace representation, which can be stably estimated from even few samples [64]. Practical examples exist in literature, for instance, illumination subspace can be generated from a set of at most 9 frontal face images. In this example, the subspace produced by GDS represents the explicit information about the object shape [44, 67], which is not achievable by LDA or DCC. Besides, the computational cost of GDS is relatively low for a supervised subspace-based method [68, 69].

To create the supervised filter banks, we will use the supervised patch set $\mathbf{P}_s = \{p_i \in \mathbb{R}^K\}_{i=1}^{M_s}$. For a C class classification problem, it is required to compute a set of C feature matrices $\{\mathbf{A}_j\}_{j=1}^{C}$. For each feature matrix $\mathbf{A}_j$, we need to compute the autocorrelation matrix $\mathbf{C}_j = \mathbf{A}_j{}^T\mathbf{A}_j$.

Equipped with all C autocorrelation matrices, we can move forward to calculate the matrix $\mathbf{U}_j$ of eigenvectors which diagonalizes the autocorrelation matrix $\mathbf{C}_j$:

$$\mathbf{D}_j = \mathbf{U}_j{}^{-1}\mathbf{C}_j\mathbf{U}_j, \quad j = \{1, \ldots, C\}. \tag{3}$$

In Eq. 3, each $\mathbf{U}_j$ is a $K \times K$ matrix satisfying $\mathbf{U}_j\mathbf{U}_j{}^T = \mathbf{U}_j{}^T\mathbf{U}_j = \mathbf{I}$. The columns of $\mathbf{U}_j$ that correspond to nonzero singular values compound a set of orthonormal basis vectors for the range of $\mathbf{C}_j$. $\mathbf{D}_j$ is the diagonal matrix of eigenvalues of $\mathbf{C}_j$. It is important to note that GDS does not center the data at the mean [44, 70], contrasting to the feature matrix created using PCA. In addition, unlike PCA, GDS produces a subspace for each class independently, in order to exploit the correlations among the different classes. Once all the basis vectors $\mathbf{U}_j$ have been obtained, we can then calculate the total projection matrix $\mathbf{G}$ as follows:

$$\mathbf{G} = \sum_{j=1}^{C} \mathbf{U}_j{}^T\mathbf{U}_j. \tag{4}$$

The eigen-decomposition of the total projection matrix $\mathbf{G}$ produces a $K \times K$ orthogonal matrix $\mathbf{U}_s$. The sum subspace $\mathcal{S}$, spanned by $\mathbf{U}_s$, can be decomposed into the sum of the following subspaces:

$$\mathcal{S} = \mathcal{M} \oplus \mathcal{D}, \tag{5}$$

where $\mathcal{D}$ is the generalized difference subspace. By using this decomposition, we can formulate the subspace that represents the differences among all the subspaces just excluding the subspace $\mathcal{M}$ from the sum subspace $\mathcal{S}$. In practical terms, the filter bank $\mathcal{F}_s$ is defined by the remaining D_s vectors of $\mathcal{S}$ after excluding the $D_{\mathcal{M}}$ first vectors. This procedure can be implemented by the following expression:

$$\mathcal{F}_s = \mathbf{U}_s\mathbf{R}_s, \tag{6}$$

where $\mathbf{R}_s$ is a $K \times K$ matrix containing 0 on its first $D_{\mathcal{M}}$ principal diagonal entries, 1 on the remaining D_s principal diagonal entries, and 0 elsewhere. After this procedure, we should have a supervised filter bank $\mathcal{F}_s \in \mathbb{R}^{D_s \times K}$.

3.6 Filtering an input image

Here, we describe how to filter an input image using the unsupervised and supervised filter banks developed previously. Since the filter banks are D_u and D_s-dimensional subspaces, we can use each eigenvector of $\mathcal{F}_u = \{\phi_r\}_{r=1}^{D_u}$ and $\mathcal{F}_s = \{\psi_t\}_{t=1}^{D_s}$ as convolutional filters. Therefore, given an input image $\mathbf{P}_{in} \in \mathbb{R}^{N \times M}$, the goal here is to filter $\mathbf{P}_{in}$ as follows:

$$\mathbf{V}_r = \mathrm{map}_K(\phi_r) * \mathbf{P}_{in}, \quad r = \{1, \ldots, D_u\}. \tag{7}$$

$$\mathbf{W}_t = \mathrm{map}_K(\psi_t) * \mathbf{P}_{in}, \quad t = \{1, \ldots, D_s\}. \tag{8}$$

In Eqs. 7 and 8, the operator $\mathrm{map}_K(\cdot)$ maps an input vector $y \in \mathbb{R}^{K_1K_2}$ onto a matrix $\mathbf{Y} \in \mathbb{R}^{K_1 \times K_2}$. The symbol $*$ refers to a convolution with zero-padding in the boundary of the image patch.

It is important to note that the output of the first layer of our proposed network will produce $D_s + D_u$ images. By using the unsupervised and supervised filtered images $\mathbf{V}_r$ and $\mathbf{W}_t$, more subspaces can be learned to create more layers. Usually, more than one layer is employed in shallow networks, so more features can be extracted from $\mathbf{P}_{in}$. For instance, for a $Z = 2$ layers network, we should learn 4 filter banks, where $\mathcal{F}_u^1$, $\mathcal{F}_s^1$ may be learned from $\mathbf{X}_u$ and $\mathbf{X}_s$, and $\mathcal{F}_u^2$ and $\mathcal{F}_s^2$ can be learned from $\mathbf{V}_r$ and $\mathbf{W}_t$. Figure 3 shows the convolution processes using two basis vectors.

3.7 Feature mapping

The feature vectors generated by the convolutional layers of shallow networks are usually very large, since there are no pooling layers. As the model becomes deeper (i.e., the number of layers increases), the number of feature maps grows exponentially. The fast growth of the feature vector severely limits feature extraction performance and processing efficiency. To solve this weakness, it is required to employ a specific layer to reduce the dimensionality of the feature vector generated by convolutional layers.

After filtering the input image $\mathbf{P}_{in}$, the produced filtered images are concatenated to achieve a high dimensional vector, for example, given a feature vector generated from a network with the following set of parameters: $K_1 = K_2 = 8$, input image size of $M = N = 28$, $D_u = D_s = 5$, and $Z = 1$. Then, the final feature vector will be a $(D_u + D_s)(MN) = 7840-$dimensional vector. In this simple simulation, it is clear that a dimensionality reduction technique is required.

For the Zth layer, $N_u^Z + N_s^Z$ images will be generated as a result of successive Z convolutions. The number of images in the final convolutional layer depends on the dimension of the unsupervised and supervised subspaces of each layer and can be obtained as follows:

$$N_u^Z = \prod_{z=1}^{Z} D_u^z. \tag{9}$$

$$N_s^Z = \prod_{z=1}^{Z} D_s^z. \tag{10}$$

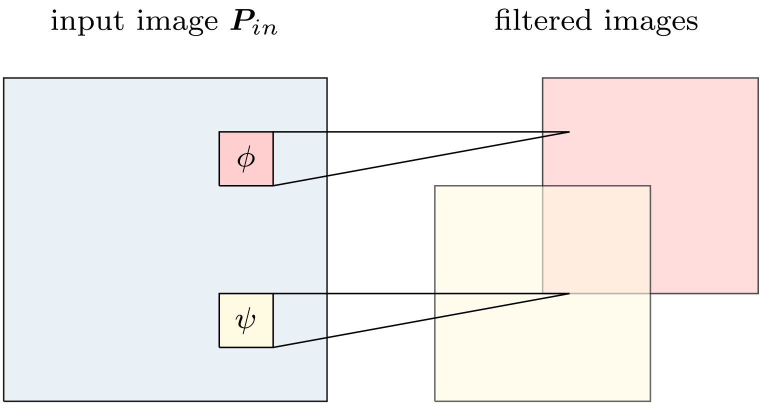


Fig. 3 Illustration of a single convolutional layer. The input image $\mathbf{P}_{in}$ or a feature map of the previous layer is convolved by unsupervised and supervised filters ϕ and ψ, respectively, to yield the output feature maps

Following the procedure of PCANet, we can convert the filtered images to a set of $N_u^{Z-1} + N_s^{Z-1}$ images as follows:

$$\mathbf{T}_u^m = \sum_{z=1}^{N_u^Z} 2^{(z-1)} \mathrm{H}(\mathbf{V}_m), \quad m = \{1, \ldots, N_u^{Z-1}\}. \tag{11}$$

$$\mathbf{T}_s^n = \sum_{z=1}^{N_s^Z} 2^{(z-1)} \mathrm{H}(\mathbf{W}_n), \quad n = \{1, \ldots, N_s^{Z-1}\}. \tag{12}$$

In Eqs. 11 and 12, the filtered images $\mathbf{V}_m$ and $\mathbf{W}_n$ are binarized using a Heaviside step-like function $\mathrm{H}(\cdot)$, whose value is 1 for positive entries and 0 otherwise. After this procedure, we achieve $N_u^{Z-1} + N_s^{Z-1}$ integer-valued $\mathbf{T}_u^m$ and $\mathbf{T}_s^n$ images with pixel value in the range $[\,0, 2^{N_u^Z} - 1]$ and $[\,0, 2^{N_s^Z} - 1]$, respectively. It is worth noting that this dimensionality reduction is also employed in shallow networks-based transfer learning [71]. Then, each $\mathbf{T}_u^m$ and $\mathbf{T}_s^n$ images are partitioned into B blocks, where block-wise histogram is applied. At last, the feature $f\ =[f_u, f_s]$ of the input image $\mathbf{P}_{in}$ is defined as the set of block-wise histograms $\mathbf{b}_h$:

$$f_u = [\,\mathbf{b}_h(\mathbf{T}_u^1), \mathbf{b}_h(\mathbf{T}_u^2), \ldots, \mathbf{b}_h(\mathbf{T}_u^{N_u^{Z-1}})]^T\ . \tag{13}$$

$$f_s = [\,\mathbf{b}_h(\mathbf{T}_s^1), \mathbf{b}_h(\mathbf{T}_s^2), \ldots, \mathbf{b}_h(\mathbf{T}_s^{N_s^{Z-1}})]^T\ . \tag{14}$$

Most modern networks [72] make use of features of each layer, creating a huge vector. Although the idea is appealing, we chose to use the strategy employed by PCANet, since it is more similar to the procedure used by CNN. In the investigated shallow networks, SVM is applied for the classification. The same classifier is then used with DFSNet.

One of the advantages of our proposed shallow network is its reduced number of parameters compared to deep learning networks. The hyper-parameters of DFSNet are as follows: the filter size K, the number of layers Z, the number of filters in each layer $D_u^1, D_u^2, \ldots, D_u^Z$ and $D_s^1, D_s^2, \ldots, D_s^Z$, and the block size B for the histogram. Figure 4 presents the proposed shallow network equipped with two convolutional layers and a feature mapping layer.

4 Experimental results and discussion

In this section, the effectiveness of the proposed network is evaluated using five datasets: CIFAR-10 [73], LFW [74], NYU Depth V1 [75], ETH-80 [76], and FERET [77], which

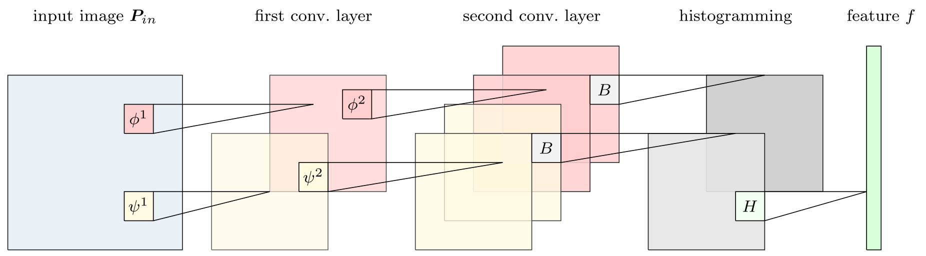


Fig. 4 Conceptual figure of DFSNet. The network employs two distinct filters banks based on PCA and GDS. To reduce the high dimensionality of the feature vectors and increase rotation invariance, the proposed method is followed by a feature mapping that includes binarization and block-wise histogram. Similar to most shallow networks, the classification is performed by linear SVM

include varied classification tasks such as face recognition, indoor scene recognition, and object classification. Our experiments are broken down into three main series. First, the visualization of the filters produced by the proposed network using the ALOI [78] dataset is provided to verify the similarities among them. Then, feature separability of DFSNet in different scenarios is analyzed, including when only unsupervised data is available and when just supervised data is employed. Finally, a comparison with current shallow networks is presented.

4.1 Visualization of the filters produced by the proposed method

In this experiment, the unsupervised and supervised filters are presented and analyzed. DFSNet is trained using the ALOI database with 50% of unsupervised data and 50% of supervised data in order to make a clear comparison.

ALOI is a database containing 72000 images and 1000 classes. These images were obtained from several points of view and with variations in the illumination. The ALOI dataset version that contains only changes of point of view was utilized. For sake of simplicity, DFSNet was trained with 1 layer, where $K_1 = K_2 = 8$. ALOI database provides good examples of high similar classes, which may expose the difficulties in extracting discriminative patterns. For visualization purposes, filters employed RGB data. Figure 5 shows samples of the ALOI dataset employed in this experiment.

Figure 6 presents the filters and the filtered images produced by the proposed network. Figure 6a shows the unsupervised filters produced by PCA, which are distributed in each row according to their eigenvalue in decreasing order, from left to right. Thus, the leftmost filter of each row is the most representative filter. Regarding the filters produced by PCA, it is possible to observe that the first filters are very similar to edge and contour detectors and that the following filters are very similar to texture and color detectors. Although these filters provide an interpretable view, they are not discriminative, since PCA does not account for the relation between patterns of different image classes.

Figure 6b presents the supervised filters generated by GDS. Again, the leftmost filter of each row is the most discriminative one. In this experiment, we set $D_{\mathcal{M}} = 2$, since this value reduces information loss. From the filtered images, we can notice that the ones produced by GDS exhibit higher variability than the filtered images produced by the PCA filter banks. For example, images filtered by PCA are very similar in terms of color aspects, while images filtered by GDS present more color variability. This phenomenon is directly related to the GDS approach, which acts by exposing discriminatory characteristics (that is, features that are not present in other classes of images), while images filtered by PCA focus on common patterns (i.e., the principal components). According to this observation, we can confirm that images filtered by GDS produce more distinctive features than features provided by PCA.

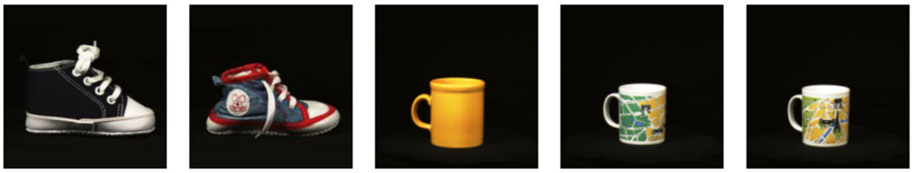
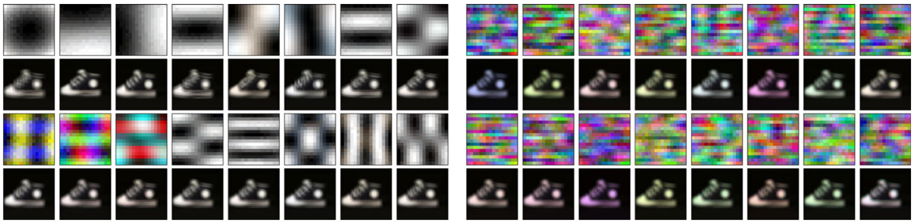


Fig. 5 Image samples of ALOI dataset



(a) Filters produced by PCA. (b) Filters produced by GDS.

Fig. 6 Filters produced by PCA and GDS on the ALOI dataset

Moreover, in filters produced by GDS, it may be observed that it is difficult to find visually interpretable patterns, such as those found in filters created by PCA. This behavior is specially due to the fact that GDS evaluates the differences between edges, contours, color, and textures generated by all classes. As a result, GDS filters provide less visual interpretability, since they represent the differences between all subspaces combinations.

4.2 Analyzing feature separability in different scenarios

The objective of this experiment is to determine whether supervised information improves the discriminability ability of DFSNet. To perform this experiment, the proposed method is trained using only 1 layer in 4 different scenarios: (1) when no supervised data is available, (2) when unsupervised data is abundant (80% of unsupervised and 20% of supervised data), (3) when unsupervised and supervised data are balanced (50% of each), and (4) when supervised data is abundant (20% of unsupervised and 80% of supervised data).

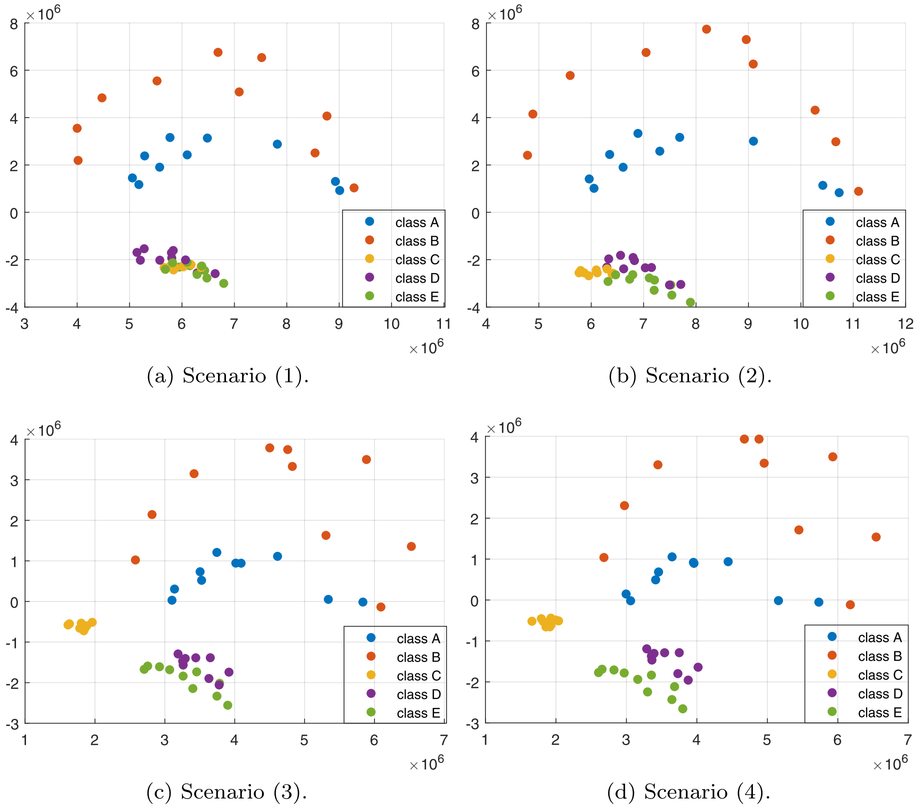
The multidimensional scaling (MDS) [79] is used to visualize features obtained from 5 classes of ALOI dataset. These classes, whose images are shown in Fig. 5, were selected due to their high similarity regarding shape and color. For example, first and second classes, called here classes A and B respectively, present a similar shape, whereas the three remaining classes (C, D, and E) exhibit identical texture and color.

Figure 7a shows the scatter when only unsupervised data is available. In this scenario, the proposed network is reduced to PCANet, where the filter banks are produced using only unsupervised data. This plot suggests that patterns of the classes C, D, and E present a high rate of overlap, where it is challenging for a classifier to generate appropriate separation hyperplanes.

In Fig. 7b, where unsupervised data is still abundant, but a few amount of labeled data is also used, patterns of the classes C, D, and E present lower overlap when compared to the previous scenario. In this case, a classifier trained with an appropriate kernel may learn a feasible solution. The situation where unsupervised data is abundant is the most realistic among all scenarios investigated in this section.

Figure 7c shows the illustration where unsupervised and supervised data are balanced. In this scheme, as expected, Fig. 7c suggests that the overlap between patterns is lower than in the previous scenario and may reflect the influence of supervised data. Here, GDS has sufficient supervised data to reduce overlap between the classes considerably and, visually, class C is well separated from classes D and E.

Finally, as it was also expected, Fig. 7d exhibits the best scenario, when supervised data is abundant. In this illustration, the extracted features are mostly supervised and



(a) Scenario (1).

(b) Scenario (2).

(c) Scenario (3).

(d) Scenario (4).

Fig. 7 Scatter plots using the first two MDS dimensions showing distances between five classes of the ALOI dataset on four different scenarios: (1) when no supervised data is available, (2) when unsupervised data is abundant, (3) when unsupervised and supervised data are balanced, and (4) when supervised data is abundant

reveal the discriminative ability of GDS to remove overlap between classes. Among all the investigated scenarios, this is less realistic regarding the semi-supervised learning paradigm.

4.3 Comparison with related shallow networks

In this section, we compare DFSNet to the following unsupervised shallow networks: PCANet, DCTNet, CCANet, and CFR-ELM, as well as to the supervised shallow networks: LDANet, DCCNet, OSNet, and CKNet. In the following, we describe the employed datasets and, after that, we show the experimental results.

4.3.1 Datasets and experimental settings

For face recognition evaluation, the FERET dataset [77] is employed. FERET comprises 1196 images from 429 subjects. Images were taken under varying lighting conditions, with diverse expressions and throughout 3 years. The dataset is divided into gallery and probe. The probe set is subdivided into 4 sections, as follows: Fb containing different expressions, Fc including varying lighting conditions, dup-I obtained within the period of 3 to 4 months, and finally, dup-II obtained after 1 and a half year apart from the initial dataset development. We employed 150 × 90 grayscale images with $K_1 = K_2 = 5, L_1 = L_2 = 8$ and the size of non-overlapping blocks was set to 15×15. The dimension of the produced features was reduced to 1000 by whitening PCA in order to facilitate the comparison with the other shallow networks. These parameter values were chosen experimentally

We employ ETH-80 dataset for object recognition. ETH-80 contains images of 8 object categories, where each category includes 10 object subcategories in 41 different image orientations, resulting in 410 images per category. In total, ETH-80 database contains 3280 images. We resized the images to 64 pixels. ETH-80 provides images with and without background. To analyze the behavior of the learning methods, we used the object images with background. In this experiment, we set $L_1 = L_2 = 8$, $K_1 = K_2 = 7$, block size 7×7, and block overlapping ratio 0.5. Since ETH-80 does not explicitly provide a training set, we conduct 10 experimental runs with 2000 training images, which were randomly selected for each run.

We use LFW dataset [74] for a more challenging face recognition evaluation. It consists of images of faces collected from the web. The faces were detected using Viola-Jones face detector and cropped into 150×80 pixels. LFW dataset is specially challenging because it was designed for studying the problem of unconstrained face recognition. Following the standard evaluation protocol, we perform 10-fold cross validation using the provided 10 subsets, where each subset contains 300 intra-class pairs and 300 inter-class pairs. In this experiment, we set $K_1 = K_2 = 7$, $L_1 = L_2 = 8$, and 15×13 for the non-overlapping block size. We report the average result of the 10 folds. For the final feature, we employ WPCA with a size 3000. Contrasting to the experimental setup reported in [41], we do not employ the square-root operation on the final feature to maintain consistency with the other experiments provided in this work.

For object recognition, we use CIFAR-10 [73] dataset that consists of 50,000 training and 10,000 test images. The large variability in scale, viewpoint, illumination, and background clutter of images in CIFAR-10 poses a significant challenge for classification. In this experiment, we set $K_1 = K_2 = 5$, $L_1 = 40$, $L_2 = 10$, and 8×8 for the overlapping block size with overlapping ratio of 0.5. Different from the experimental setup reported in [41], we do not employ spatial pyramid pooling in order to evaluate only the convolution method. Instead, we employ WPCA to produce a final feature vector of size 1000.

We also use NYU Depth V1 dataset [75] that was collected by the New York University. The dataset includes depth information, which contains both geometric information and distance of objects. NYU Depth V1 dataset consists of 2347 pairs of images grouped into 7 categories, including bathroom, bedroom, bookstore, cafe, kitchen, living room, and office. In this experiment, we employ $K_1 = K_2 = 7$ and $L_1 = L_2 = 8$. Exceptionally for LDANet, the number of filters is set to 6, since the reduced dimensionality must be less than the number of classes. For fair comparison, we adopt the same parameter setting for all the evaluated networks and we report results for the RGB data.

4.3.2 Results

Since the amount of unsupervised and supervised data may vary according to different applications, four versions of DFSNet are provided as follows: (1) when unsupervised data is abundant (80% and 20% of unsupervised and supervised data, respectively), (2) when unsupervised data is slightly more than the supervised one (60% and 40% of unsupervised and supervised data, respectively), (3) when there is slightly more supervised data than unsupervised one (40% and 60% of unsupervised and supervised data, respectively), and

(4) when supervised data is abundant (20% and 80% of unsupervised and supervised data, respectively).

For an adequate comparison, the Coiflets and Daubechies orthogonal wavelet transform are used to extract the low-frequency sub-images of the original images to generate two view features for the CCANet [54]. Besides, the TR normalization introduced in [48] is not employed so that we can evaluate the surface networks only in relation to their convolutional filters. As in PCANet, LDANet, and DCTNet, linear SVM is adopted for the classification step due to be relatively less prone to overfitting than its non-linear version.

Surprisingly, the investigated shallow networks obtained comparable recognition rates, regardless of the learning paradigm used. Although the difference is small, in some scenarios, it is evident that one learning paradigm presents an advantage over the other. More precisely, when the amount of training data is not enough to learn a robust model, unsupervised methods offer an advantage. This observation is visible in the FERET database, where DCTNet has shown superior results compared to the other methods. When the amount of training data is sufficient to learn a robust model, supervised methods have an advantage, as in the example of the CIFAR-10 database, where DCCNet produced a very competitive recognition rate. This observation suggests that applications may benefit from models that employ both learning paradigms, thus exploiting training data efficiently. More precisely, the required amount of labeled data to improve the accuracy of the method is relatively low, establishing a better compromise between the advantages of both supervised and unsupervised paradigm.

According to Table 1, PCANet and LDANet consistently produce high recognition rates. PCANet is very competitive, even compared to supervised methods, such as LDANet and OSNet. This is an indication of the advantages that the multistage model employed by shallow networks can provide, even in the absence of labeled data. Among the unsupervised methods, PCANet presented the highest recognition results.

Despite being built using only random Fourier features, CKNet is extremely competitive on FERET, ETH-80, and CIFAR-10 datasets. This method is very similar to DCTNet, with the difference that in DCTNet, filters are selected deterministically. CKNet presents the ability to decode textures, which is inherited from the Fourier descriptors. Besides, Fourier transform introduces translation, scalable, and rotation invariance to the features.

Table 1 Recognition rates of the proposed and the related shallow networks

	CIFAR-10 [73]	LFW [74]	NYU Depth V1 [75]	ETH-80 [76]	FERET [77]	Learning paradigm
PCANet [41]	78.67	85.20	79.58	93.96	97.25	Unsupervised
DCTNet [48]	73.29	85.33	75.17	89.35	97.32	Unsupervised
CCANet [54]	79.11	84.27	77.05	94.33	94.83	Unsupervised
CFR-ELM [62]	80.24	N.A.	N.A.	95.63	N.A.	Unsupervised
LDANet [41]	78.33	84.89	76.85	93.87	97.18	Supervised
DCCNet [58]	80.68	84.91	77.33	91.21	94.73	Supervised
OSNet [61]	78.81	83.69	76.59	94.06	93.07	Supervised
CKNet [63]	80.60	83.67	77.21	94.22	93.56	Supervised
DFSNet-1	80.77	84.96	80.31	94.23	96.96	Semi- supervised
DFSNet-2	80.97	85.29	80.53	94.43	97.28	Semi- supervised
DFSNet-3	81.06	85.45	80.61	94.52	97.47	Semi- supervised
DFSNet-4	81.20	85.55	80.68	94.66	97.54	Semi- supervised

N.A. not available

DCCNet and OSNet are subspace-based methods that exploit the concept of constraint subspace to create more discriminative features. The fundamental difference between these methods is that DCCNet employs an iterative process to create its constraint subspace, while OSNet produces it through the decomposition of the principal subspace $\mathcal{M}$. As a result, DCCNet is good on CIFAR-10, where the number of classes is low, and the number of training samples is high, due to the iterative method of calculating the constraint subspace. Also, DCCNet can represent nonlinear structures, which may be found in the CIFAR-10 database. OSNet is competitive on ETH-80, overcoming DCCNet. In this dataset, the restricted number of training examples benefits subspace methods based on decompositions, also suggesting that the iterative method employed by DCCNet requires more samples to obtain a more efficient constraint subspace.

Compared to PCANet and LDANet, CCANet presents competitive results on CIFAR-10and ETH-80, while performing not so well on the remaining datasets. This observation suggests that CCANet is recommended in problems involving object recognition. When applied to the face recognition datasets, PCANet and LDANet perform efficiently compared to CCANet. In comparison to PCANet and LDANet, CCANet has the disadvantage of easily overfit to noise correlations between datasets, weakening its discriminative capability.

DCTNet presents particularly good results in face recognition, achieving high accuracy on LFW and FERET, which are competitive results compared to PCANet and LDANet. DCTNet benefits from the ability of DCT to concentrate energy in a few first coefficients. The filter banks employed by DCTNet make use of the first coefficients and discard the high frequencies that generally represent noise. As a result, the feature vector produced by DCTNet can be viewed as denoised data, which shows good results on face recognition datasets.

The CFR-ELM provided impressive results on CIFAR-10 and ETH-80. The method achieved competitive results on CIFAR-10, outperforming the unsupervised methods in addition to producing competitive results to DCCNet and CKNet. These results suggest that the nonlinear adaptive processing capacity of CFR-ELM inherited from the ELM can learn a rich representation for CIFAR-10. The CFR-ELM attained the highest results on the ETH-80, suggesting that object classification tasks can benefit from the auto-encoder mechanism employed by CFR-ELM.

The proposed network demonstrated superior classification rate when compared to the other evaluated shallow networks, confirming the efficiency of employing the unsupervised and supervised subspaces as convolutional layers. When 20% of the information is supervised, the proposed method performs competitively. These results confirm that the supervised subspace provided by GDS produces discriminative features that improve the classification rate. CFR-EML performed slightly better on ETH-80. This result may be somewhat predictable from that the nonlinear adaptive processing of CFR-EML works effectively on the other datasets. This point suggests that by adding some nonlinear processing in the generation of the filters, we may improve our method further.

Here, we highlight that the proposed network attained superior recognition rate compared to the other shallow networks in the CIFAR-10 database. This observation may have been influenced by the amount of training data that the database presents, as well as the reduced number of classes. Once a database presents a large amount of training data, DFSNet can learn discriminative structures efficiently.

Given a small set of labeled data and abundant unlabeled data, GDS attempts to select the most discriminative subspace from the image classes, providing complementary information. Feature fusion in neural networks by concatenation or by addition have demonstrated to be a powerful strategy to provide deeper representations [80–82]. In this approach, features from adjacent layers are concatenated to produce a more representative feature. In DFSNet, we can observe that PCA and GDS work in a similar aspect, since GDS is based on the SVD of the PCA basis vectors.

Another justification for the proposed architecture is the benefits of using networks in parallel, such as the Siamese [83, 84] and Two-Stream [85, 86] networks. These networks have the purpose of extracting more information from data, using an architecture where there are two networks in parallel.

5 Conclusions and future work

In this paper, a new shallow network is proposed and tested on face recognition, object recognition, and scene understanding. Unlike conventional shallow networks, the proposed network is capable of manipulating both supervised and unsupervised data. This ability makes the proposed network efficient even when a small amount of supervised data is available. Another advantage of the proposed method is its independence from automatic differentiation algorithms. Because their convolution filters are formed by a decomposition performed by SVD per layer, this method has advantage when employed in contexts where time is a limiting factor. The results obtained in datasets CIFAR-10, LFW, NYU Depth V1, ETH-80, and FERET show that the proposed network is capable of producing highly discriminative features compared to networks of similar architectures.

The number of layers is a limitation directly associated with the network capacity. Modern neural networks that produce competitive results, in general, have a very large number of layers. We understand that the nature of the subspace method causes such a limitation. Since the basis vectors that span the subspaces are a subset of the basis vectors produced by PCA, an amount of information, even though small, is lost. The subspace used as the first convolution filter bank represents a total of 90% of the variation found in the database. As the second subspace is produced through the images processed by the first subspace and also has a cutoff margin, the information obtained by the second subspace is of the order of 81%, following the same threshold factor. This value becomes even lower if we add a third layer. Using the same threshold factor, this layer will represent only about 72% of the dataset. Without an optimization method that can adjust the subspaces to a more suitable direction, adding more layers makes the method slower and, worse, weakening the network representation.

The second limitation of our method is the absence of pooling. Although the results produced by shallow networks in general (PCANet, LDANet, and CCANet) are very competitive, the feature vector provided by such networks are very large. Since there is no dimensionality reduction mechanisms between the layers, the produced features have exponential growth according to the number of layers. This problem restricts these networks to no more than four layers. A pooling method would add robustness to pattern rotations and dimensionality reduction, which would make feature size independent of the number of layers.

Usually, the training algorithms for neural networks are iterative and, consequently, require some initial set of parameters from which to start the iterations. Also, training neural networks is a challenging task that most methods are significantly affected by selection of the initialization parameters. Motivated by this challenge, the proposed method can be an alternative to the random initialization process. In this direction, the filter banks of the proposed network can be employed as the filter banks of a deep neural network during its initialization stage. Since the proposed networks produce better results than RandNet [41], it is expected that employing the basis vectors of a subspace may provide better accuracy in fewer iterations.

An important research direction is to extend the proposed network to handle tensor data, which is recommended for video analysis, like gesture and action recognition. Tensor subspaces exist in literature and may provide convolutional filters for such networks. In addition, it is possible to employ CFR-ELM instead of PCANet in the semi-supervised framework. The learning paradigm employed in this work can be extended to deeper architectures, which can exhibit the same advantages (e.g., computational cost). In the same research line, the proposed network can be employed as an initialization method for deeper networks.

Abbreviations

CNN: Convolutional neural network; PCA: Principal component analysis; LDA: Linear discriminant analysis; DCC: Discriminative canonical correlations analysis; LBP: Local binary patterns; SIFT: Scale-invariant feature transform; HOG: Histogram of oriented gradients; GDS: Generalized difference subspaces; DCT: Discrete cosine transform; CCANet: Canonical correlation analysis network; PCANet: Principal component analysis network; DCTNet: Discrete cosine transform network; DCCNet: Discriminative canonical correlation network; LDANet: Linear discriminant analysis network; OSNet: Orthogonal subspace network; Cosine-CKN: Cosine convolutional kernel network; CFR-ELM: Compact feature representation; MDS: Multidimensional scaling

Acknowledgments

We are grateful to the anonymous reviewers for their constructive comments that contributed to improving the revised version of the manuscript.

Authors' contributions

BG, LS, and ES conceived the idea, developed the method, and conducted the experiment. KF, WJ, and KS were involved in the extensive discussions and evaluations and read and approved the final manuscript.

Funding

This work was supported by JSPS KAKENHI grant number 19K20335 and the Foundation for Research Support of the State of Amazonas (FAPEAM).

Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Center for Artificial Intelligence Research (C-AIR), Tsukuba, Japan. [2]Federal University of Amazonas, Manaus, Brazil.
[3]University of Tsukuba, Tsukuba, Japan.

Received: 1 April 2019 Accepted: 20 April 2020
Published online: 16 June 2020

References

1. Z. Gong, P. Zhong, Y. Yu, W. Hu, Diversity-promoting deep structural metric learning for remote sensing scene classification. IEEE Trans. Geosci. Remote Sens. **56**(1), 371–390 (2018)
2. N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, J. Liang, Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE Trans. Med. Imaging. **35**(5), 1299–1312 (2016)
3. A. T. Lopes, E. de Aguiar, A. F. De Souza, T. Oliveira-Santos, Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. Pattern Recog. **61**, 610–628 (2017)
4. X. Gao, T. Zhang, Unsupervised learning to detect loops using deep neural networks for visual slam system. Auton. Robot. **41**(1), 1–18 (2017)

5. X. Xie, H. Liu, M. Edmonds, F. Gaol, S. Qi, Y. Zhu, B. Rothrock, S. C. Zhu, in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Unsupervised learning of hierarchical models for hand-object interactions (IEEE, 2018), pp. 1–9
6. A. M. Dai, Q. V. Le, in *Advances in neural information processing systems*, Semi-supervised sequence learning, (2015), pp. 3079–3087
7. A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, T. Brox, in *Advances in Neural Information Processing Systems*, Discriminative unsupervised feature learning with convolutional neural networks, (2014), pp. 766–774
8. I. Bougoudis, K. Demertzis, L. Iliadis, Fast and low cost prediction of extreme air pollution values with hybrid unsupervised learning. Integr. Comput. Aided Eng. **23**(2), 115–127 (2016)
9. M. C. Thomas, W. Zhu, J. A. Romagnoli, Data mining and clustering in chemical process databases for monitoring and knowledge discovery. J. Process Control. **67**, 160–175 (2018)
10. M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, E. Muharemagic, Deep learning applications and challenges in big data analytics. J. Big Data. **2**(1), 1 (2015)
11. Q. Zhang, L. T. Yang, Z. Chen, Deep computation model for unsupervised feature learning on big data. IEEE Trans. Serv. Comput. **9**(1), 161–171 (2016)
12. A. M. Dai, Q. V. Le, in *Advances in neural information processing systems*, Semi-supervised sequence learning, (2015), pp. 3079–3087
13. M. I. Jordan, T. M. Mitchell, Machine learning: trends, perspectives, and prospects. Science. **349**, 255–260 (2015)
14. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, in *International conference on machine learning*, Decaf: a deep convolutional activation feature for generic visual recognition, (2014), pp. 647–655
15. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, in *Advances in Neural Information Processing Systems*, Improved techniques for training gans, (2016), pp. 2234–2242
16. A. Holzinger, Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Inf. **3**(2), 119–131 (2016)
17. S. S. Rautaray, A. Agrawal, Vision based hand gesture recognition for human computer interaction: a survey. Artif. Intell. Rev. **43**(1), 1–54 (2015)
18. J. Song, L. Gao, L. Liu, X. Zhu, N. Sebe, Quantization-based hashing: a general framework for scalable image and video retrieval. Pattern Recog. **75**, 175–187 (2018)
19. R. Xia, Y. Pan, H. Lai, C. Liu, S. Yan, in *AAAI*, Supervised hashing for image retrieval via image representation learning, (2014), p. 2
20. T. Bouwmans, E. H. Zahzah, Robust PCA via principal component pursuit: a review for a comparative evaluation in video surveillance. Comput Vision Image Underst. **122**, 22–34 (2014)
21. S. Ojha, S. Sakhare, in *Pervasive Computing (ICPC), 2015 International Conference on*, Image processing techniques for object tracking in video surveillance-a survey (IEEE, 2015), pp. 1–6
22. K. Jaseena, B. C. Kovoor, A survey on deep learning techniques for big data in biometrics. Int. J. Adv. Res. Comput. Sci. **9**(1) (2018)
23. K. Sundararajan, D. L. Woodard, Deep learning for biometrics: a survey. ACM Comput. Surv. (CSUR). **51**(3), 65 (2018)
24. X. Geng, H. Zhang, J. Bian, T. S. Chua, in *Proceedings of the IEEE International Conference on Computer Vision*, Learning image and user features for recommendation in social networks, (2015), pp. 4274–4282
25. J. Wang, M. Korayem, S. Blanco, D. J. Crandall, in *Proceedings of the 2016 ACM on Multimedia Conference*, Tracking natural events through social media and computer vision (ACM, 2016), pp. 1097–1101
26. D. Ciregan, U. Meier, J. Schmidhuber, in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, Multi-column deep neural networks for image classification (IEEE, 2012), pp. 3642–3649
27. C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1915–1929 (2013)
28. Y. Sun, Y. Chen, X. Wang, X. Tang, in *Advances in Neural Information Processing Systems*, Deep learning face representation by joint identification-verification, (2014), pp. 1988–1996
29. L. Nanni, S. Ghidoni, S. Brahnam, Handcrafted vs. non-handcrafted features for computer vision classification. Pattern Recogn. **71**, 158–172 (2017)
30. F. Zhu, L. Shao, J. Xie, Y. Fang, From handcrafted to learned representations for human action recognition: a survey. Image Vision Comput. **55**, 42–52 (2016)
31. M. R. Turner, Texture discrimination by gabor functions. Biol. Cybern. **55**(2-3), 71–82 (1986)
32. T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions. Pattern Recog. **29**(1), 51–59 (1996)
33. T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
34. D. G. Lowe, Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
35. N. Dalal, B. Triggs, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1*, Histograms of oriented gradients for human detection (IEEE, 2005), pp. 886–893
36. K. Lai, L. Bo, X. Ren, D. Fox, in *Robotics and Automation (ICRA) 2011 IEEE International Conference on*, A large-scale hierarchical multi-view RGB-D object dataset (IEEE, 2011), pp. 1817–1824
37. Q. Zhu, M. C. Yeh, K. T. Cheng, S. Avidan, in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2*, Fast human detection using a cascade of histograms of oriented gradients (IEEE, 2006), pp. 1491–1498
38. A. Krizhevsky, I. Sutskever, G. E. Hinton, in *Advances in neural information processing systems*, Imagenet classification with deep convolutional neural networks, (2012), pp. 1097–1105
39. M. A. Alsheikh, D. Niyato, S. Lin, H. P. Tan, Z. Han, Mobile big data analytics using deep learning and Apache Spark. IEEE Netw. **30**(3), 22–29 (2016)
40. Y. Qian, J. Dong, W. Wang, T. Tan, in *Media Watermarking, Security, and Forensics 2015, vol. 9409*, Deep learning for steganalysis via convolutional neural networks, (2015), p. International Society for Optics and Photonics
41. T. H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, Y. Ma, PCANet: a simple deep learning baseline for image classification? IEEE Trans. Image Process. **24**(12), 5017–5032 (2015)

42. M. Dorfer, R. Kelz, G. Widmer, Deep linear discriminant analysis. arXiv preprint arXiv:1511.04707 (2015)
43. C. Y. Low, A. B. J. Teoh, C. J. Ng, in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Multi-fold Gabor filter convolution descriptor for face recognition (IEEE, 2016), pp. 2094–2098
44. K. Fukui, A. Maki, Difference subspace and its generalization for subspace-based methods. IEEE transactions on pattern analysis and machine intelligence. **37**(11), 2164–2177 (2015)
45. M. Nishiyama, O. Yamaguchi, K. Fukui, in *International Conference on Audio-and Video-Based Biometric Person Authentication*, Face recognition with the multiple constrained mutual subspace method (Springer, 2005), pp. 71–80
46. S. Ding, X. Xi, Z. Liu, H. Qiao, B. Zhang, A novel manifold regularized online semi-supervised learning model. Cogn. Comput. **10**(1), 49–61 (2018)
47. T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, et al., Never-ending learning. Communications of the ACM. **61**(5), 103–115 (2018)
48. C. J. Ng, A. B. J. Teoh, in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Dctnet: a simple learning-free approach for face recognition (IEEE, 2015), pp. 761–768
49. J. N. Lee, Y. H. Byeon, S. B. Pan, K. C. Kwak, An EigenECG network approach based on PCANet for personal identification from ECG signal. Sensors. **18**(11), 4024 (2018)
50. T. Almeida, H. Macedo, L. Matos, N. Vasconcelos, Prototyping a traffic light recognition device with expert knowledge. Information. **9**(11), 278 (2018)
51. Y. Zi, F. Xie, Z. Jiang, A cloud detection method for Landsat 8 images based on PCANet. Remote Sens. **10**(6), 877 (2018)
52. X. Zhu, M. Ding, T. Huang, X. Jin, X. Zhang, PCANet-based structural representation for nonrigid multimodal medical image registration. Sensors. **18**(5), 1477 (2018)
53. N. Wang, B. Li, Q. Xu, Y. Wang, Automatic ship detection in optical remote sensing images based on anomaly detection and SPP-PCANet. Remote Sens. **11**(1), 47 (2018). https://doi.org/10.3390/rs11010047
54. X. Yang, W. Liu, D. Tao, J. Cheng, Canonical correlation analysis networks for two-view image recognition. Inf. Sci. **385**, 338–352 (2017)
55. J. Bruna, S. Mallat, Invariant scattering convolution networks. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1872–1886 (2013)
56. E. Oyallon, S. Mallat, L. Sifre, Generic deep networks with wavelet scattering. arXiv preprint arXiv:1312.5940 (2013)
57. L. Sifre, S. Mallat, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Rotation, scaling and deformation invariant scattering for texture discrimination, (2013), pp. 1233–1240
58. B. B. Gatto, E. M. dos Santos, in *Image Processing (ICIP) 2017 IEEE International Conference on*, Discriminative canonical correlation analysis network for image classification (IEEE, 2017), pp. 4487–4491
59. T. K. Kim, J. Kittler, R. Cipolla, Discriminative learning and recognition of image set classes using canonical correlations. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1005–1018 (2007)
60. T. K. Kim, B. Stenger, J. Kittler, R. Cipolla, Incremental linear discriminant analysis using sufficient spanning sets and its applications. Int. J. Comput. Vis. **91**(2), 216–232 (2011)
61. B. B. Gatto, E. M. dos Santos, K. Fukui, in *Document Analysis and Recognition (ICDAR) 2017 14th IAPR International Conference on, vol. 1*, Subspace-based convolutional network for handwritten character recognition (IEEE, 2017), pp. 1044–1049
62. D. Cui, G. Zhang, W. Han, L. Lekamalage Chamara Kasun, K. Hu Huang, in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, Compact feature representation for image classification using ELMs, (2017), pp. 1015–1022
63. M. R. Mohammadnia-Qaraei, R. Monsefi, K. Ghiasi-Shirazi, Convolutional kernel networks based on a convex combination of cosine kernels. Pattern Recogn. Lett. (2018)
64. K. Fukui, N. Sogi, T. Kobayashi, J. H. Xue, A. Maki, Discriminant analysis based on projection onto generalized difference subspace. arXiv preprint arXiv:1910.13113 (2019)
65. Y. Sun, L. Zheng, W. Deng, S. Wang, in *Computer Vision (ICCV) 2017 IEEE International Conference on*, SVDNet for pedestrian retrieval (IEEE, 2017), pp. 3820–3828
66. Z. Zou, Z. Shi, Ship detection in spaceborne optical image with SVD networks. IEEE Trans. Geosci. Remote Sens. **54**(10), 5832–5845 (2016)
67. K. C. Lee, J. Ho, D. J. Kriegman, Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Anal. Mach. Intell., 684–698 (2005)
68. Z. Q. Zhao, S. T. Xu, D. Liu, W. D. Tian, Z. D. Jiang, A review of image set classification. Neurocomputing (2018)
69. L. Chen, N. Hassanpour, Survey: How good are the current advances in image set based face identification?–Experiments on three popular benchmarks with a naïve approach. Comput. Vis. Image Underst. **160**, 1–23 (2017)
70. H. Tan, Y. Gao, Z. Ma, Regularized constraint subspace based method for image set classification. Pattern Recogn. **76**, 434–448 (2018)
71. L. Nanni, S. Ghidoni, S. Brahnam, Handcrafted vs. non-handcrafted features for computer vision classification. Pattern Recogn. **71**, 158–172 (2017)
72. S. Wazarkar, B. N. Keshavamurthy, A survey on image data analysis through clustering techniques for real world applications. J. Visual Commun. Image Represent. **55**, 596–626 (2018)
73. A. Krizhevsky, *Learning multiple layers of features from tiny images. Master's thesis*. (University of Tront, 2009)
74. G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, *Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Tech. rep., Technical Report 07-49*. (University of Massachusetts, Amherst, 2007)
75. N. Silberman, R. Fergus, in *Computer Vision Workshops (ICCV Workshops) 2011 IEEE International Conference on*, Indoor scene segmentation using a structured light sensor (IEEE, 2011), pp. 601–608
76. B. Leibe, B. Schiele, in *Computer Vision and Pattern Recognition, 2003. Proceedings 2003 IEEE Computer Society Conference on, vol. 2*, Analyzing appearance and contour based methods for object categorization (IEEE, 2003), pp. II–409
77. P. J. Phillips, H. Moon, S. A. Rizvi, P. J. Rauss, The FERET evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **22**(10), 1090–1104 (2000)
78. J. M. Geusebroek, G. J. Burghouts, A. W. Smeulders, The Amsterdam library of object images. Int. J. Comput. Vis. **61**(1), 103–112 (2005)

79. I. Borg, P. J. Groenen, P. Mair, *Applied multidimensional scaling and unfolding*. (Springer, 2017)
80. G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, in *CVPR*, Densely connected convolutional networks, (2017)
81. C. T. Chung, C. Y. Tsai, C. H. Liu, L. S. Lee, Unsupervised iterative deep learning of speech features and acoustic tokens with applications to spoken term detection. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(10), 1914–1928 (2017)
82. K. He, X. Zhang, S. Ren, J. Sun, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Deep residual learning for image recognition, (2016), pp. 770–778
83. L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, P. H. Torr, in *European conference on computer vision*, Fully-convolutional siamese networks for object tracking (Springer, 2016), pp. 850–865
84. R. R. Varior, M. Haloi, G. Wang, in *European Conference on Computer Vision*, Gated Siamese convolutional neural network architecture for human re-identification (Springer, 2016), pp. 791–808
85. C. Feichtenhofer, A. Pinz, A. Zisserman, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Convolutional two-stream network fusion for video action recognition, (2016), pp. 1933–1941
86. X. Peng, C. Schmid, in *European Conference on Computer Vision*, Multi-region two-stream R-CNN for action detection (Springer, 2016), pp. 744–759

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.