A novel multi-branch wavelet neural network for sparse representation based object classification

Tan-Sy Nguyen, Marie Luong, Mounir Kaaniche, Long Ngo, Azeddine Beghdadi

A novel multi-branch wavelet neural network for sparse representation based object classification

Tan-Sy Nguyen[1,2,*], Marie Luong[2], Mounir Kaaniche[2], Long H. Ngo[2], Azeddine Beghdadi[2]

Abstract

Recent advances in acquisition and display technologies have led to an enormous amount of visual data, which requires appropriate storage and management tools. One of the fundamental needs is the design of efficient image classification and recognition solutions. In this paper, we propose a wavelet neural network approach for sparse representation-based object classification. The proposed approach aims to exploit the advantages of sparse coding, multi-scale wavelet representation as well as neural networks. More precisely, a wavelet transform is firstly applied to the image datasets. The generated approximation and detail wavelet subbands are then fed into a multi-branch neural network architecture. This architecture produces multiple sparse codes that are efficiently combined during the classification stage. Extensive experiments, carried out on various types of standard object datasets, have shown the efficiency of the proposed method compared to the existing sparse coding and deep learning-based methods.

Keywords: Object classification, sparse coding, wavelet transform, neural networks, multi-branch architecture.

*Corresponding author

[1]T.-S. Nguyen is with Université Sorbonne Paris Nord, Laboratoire Analyse, Géométrie et Applications (LAGA), CNRS UMPR 7539, Villetaneuse, France. E-mail: tansy.nguyen@math.univ-paris13.fr.

[2]M. Luong, M. Kaaniche, L. H. Ngo and A. Beghdadi are with Université Sorbonne Paris Nord, Laboratoire de Traitement et de Transport de l'Information (L2TI), UR 3043, F-93430, Villetaneuse, France. E-mail: {marie.luong, mounir.kaaniche, hailong.ngo, azeddine.beghdadi}@univ-paris13.fr.

1. Introduction

Object classification is one of the most common problems in computer vision and pattern recognition fields. It plays a crucial role in applications such as face recognition, event detection [1], visual tracking [2], and more. Its main goal consists of finding a compact representation or set of relevant features that capture the intrinsic properties of the image contents. Object/image classification remains a challenging problem, especially in the case of large scale and complex datasets [3]. In this respect, many efforts have been made in recent years to design new efficient classification methods. Among these methods, sparse representation-based classification (SRC) has attracted much attention due to its several advantages. Namely, SR ensures high robustness to noise and other kinds of degradation while producing a compact representation of the data through only a small number of meaningful features [4]. Recently, crucial attention has been paid to deep learning-based techniques due to the benefits of using neural networks for image analysis and processing [5]. Motivated by the success of such powerful techniques, this work aims to combine sparse representation with neural networks to further enhance their classification performance. In the following, we will provide an overview of the state-of-the-art classification techniques and then summarize the contributions of the current work.

1.1. Conventional classification techniques

The main idea behind most of the conventional (i.e. non-deep learning) classification techniques relies on the sparse representation tool [6, 7]. For instance, the original SRC method [6] aims to estimate the sparsest representation of a test sample using an over-complete dictionary composed of training samples. The resulting sparse code is then used as a feature descriptor for classification purposes. The latter is often made according to the minimum reconstruction error criterion. Inspired by this original SRC method, different variants have also been developed. Indeed, a kernel sparse representation for image classification and face recognition has been proposed in [8]. Moreover, to reduce the

computational time of SRC, local and block-based SRC schemes have been designed in [9] and [10], respectively. Generally, the performance of these methods is limited in the case of large image datasets due to the fact that the dictionary is formed by all the training samples of each class. To cope with this drawback, several methods based on compact dictionary learning have been developed. In this context, an interesting approach is discriminative dictionary learning (DDL), which allows the generation of a dictionary with small size from a selective dataset. This can be achieved using an objective function with reconstructive and discriminative terms. Moreover, a Fisher Discrimination Dictionary Learning (FDDL) method is proposed in [11]. This method uses a Fisher discrimination criterion to learn a structured dictionary whose sub-dictionaries have specific class labels while producing discriminative sparse coding coefficients. In [12], the Label Consistent K-SVD (LC-KSVD) method is proposed. This method consists of using only a single small dictionary for jointly learning a discriminative dictionary as well as a linear classifier. To this end, the authors explicitly incorporated a discriminative sparse-code (also called label consistency) constraint term and a classification error term into the objective function which is solved by applying the K-SVD algorithm [13]. While the above methods as well as most of the existing DDL methods rely on a single layer dictionary learning process, a multi-layer DDL approach has recently been proposed in [14]. However, dictionary learning methods are still challenging in the case of large datasets. To tackle this problem, a projection step is often performed to transform high dimensional data into a low dimensional space. For this reason, a new approach for joint dimension reduction and dictionary learning has been developed in [15]. Another interesting solution will consist of applying a nonlinear projection using neural networks.

1.2. Neural networks-based classification techniques

Recent deep learning-based classification methods can be classified into three main categories: (i) spatial-based neural networks, (ii) wavelet-based neural networks, and (iii) sparse coding neural networks. Methods in the first category

employ various Neural Network (NN) models to extract deep features for image/object classification. These models include VGG19 [16], ResNet50 [17], Wide ResNet [18] and auto-encoder [19]. To further reduce the computational complexity and exploit other compact representations of the original images, the second category of developed methods have focused on the design of NN operating in the wavelet transform domain [20]. For instance, in [21], the input data is transformed into wavelet subbands that are fed into a Convolutional Neural Network (CNN). In [22], the authors proposed a wavelet-like auto-encoder model that decomposes an input image into two channels corresponding to low and high frequency information. Then, the low frequency channel is fed into a standard classification network (VGG16) to extract deep features. Finally, these features are fused with those extracted from the high frequency channel using a lightweight network. However, by only considering two channels, the edge information is not efficiently taken into account during the classification stage. Moreover, the output scores of these two channels are combined using an average operation. In [23], the authors propose to enhance CNNs by replacing max-pooling, strided-convolution, and average-pooling with Discrete Wavelet Transform. Similarly, a U-Net architecture is deployed in [24] with the intention of embedding wavelet decomposition into the CNN blocks to reduce the resolution of the feature maps. However, this method is mainly designed for image denoising applications. Finally, methods in the third category aim to combine neural networks with sparse representation/sparse coding models. More precisely, in [25], a deep sparse coding network is developed to combine the advantages of CNN and sparse coding-based classification techniques. A sparse auto-encoder (AE)-based model using an $\ell_{1/2}$ sparsity regularization as a constraint on the hidden representation is proposed in [26]. In [27], the conventional sparse coding scheme is extended to a multi-layer architecture yielding a deep sparse coding network. Another extension of SRC, referred to as Deep Sparse Representation Classification (DSRC), is developed in [28]. It is based on a convolutional AE to learn sparse representation and find the sparse codes for classification. More precisely, an encoder is firstly used to extract embed-

ding features which are fed into the sparse coding layer. Then, the sparse codes of the features are estimated by solving a sparse coding optimization problem. Finally, the recovered embedding features are fed into the decoder for image reconstruction purposes. The obtained sparse codes are exploited during the classification stage based on the minimum reconstruction residual criterion as performed in the standard SRC approach.

1.3. Contributions

While previous deep learning-based classification methods aim to extract deep features from original images, or combine neural networks with sparse coding or embed wavelet transform into neural networks, the main goal of this work is to take advantage of neural networks, sparse coding as well as wavelets. More precisely, we propose a hybrid approach using a neural network-based sparse representation technique that operates in the wavelet transform domain. Note that a preliminary version of this work, inspired by the Deep Sparse Representation Classification (DSRC) model [28], has recently been presented in [29]. However, unlike our previous work using only the low frequency subband, we propose here a more general framework, exploiting both the approximation as well as the detail subbands. To this end, a novel multi-branch wavelet neural network (MB-WNN) architecture is designed in this paper. This architecture produces multiple sparse codes at different orientations and resolution levels. During the classification stage, and in order to handle multiple sparse codes, we have extended the standard residual-based probabilistic approach [30] which is developed for single sparse code.

The remainder of this paper is organized as follows. In Section 2, we recall some background on Sparse Representation Wavelet-based Classification (SRWC). The proposed multi-branch wavelet neural network architecture is then described in Section 3. The experimental results are shown in Section 4. Finally, some conclusions and perspectives are drawn in Section 5.

2. Background on Sparse Representation Wavelet-based Classification

Before describing the proposed architecture, we first recall the main idea behind Sparse Representation Wavelet-based Classification (SRWC) method [31]. In fact, unlike the classical SRC methods, SRWC operates in the wavelet transform domain and consists of using the approximation subband instead of the entire image.

More precisely, by considering k-classes with their labeled training samples, a wavelet decomposition is first applied to all the samples. Then, the generated approximation wavelet coefficients are processed using the Principal Component Analysis (PCA) technique [32]. The resulting vectors of the different samples are referred to as atoms. The sparse coding step consists of constructing sub-dictionaries $\mathbf{D}_{tr}^{c}$ derived from the training atoms $\mathbf{d}_{i}^{c}$, where $i \in \{1, \ldots, n_c\}$, n_c is the number of atoms in class c, $c \in \{1, \ldots, k\}$ and k is the total number of classes. Thus, the sub-dictionary $\mathbf{D}_{tr}^{c}$ for a class c is given by

$$\mathbf{D}_{tr}^{c} = [\mathbf{d}_{1}^{c}, \mathbf{d}_{2}^{c}, ..., \mathbf{d}_{n_c}^{c}] \in \mathbb{R}^{m \times n_c}, \tag{1}$$

where m is the dimension of each atom.

Then, for k classes with total number of atoms n, with $n = \sum_{c=1}^{k} n_c$, a dictionary $\mathbf{D}_{tr}$ is constructed from the above sub-dictionaries as follows:

$$\mathbf{D}_{tr} = [\mathbf{D}_{tr}^{1}, \mathbf{D}_{tr}^{2}, ..., \mathbf{D}_{tr}^{k}] \in \mathbb{R}^{m \times n} \ . \tag{2}$$

Based on the sparse representation model, any new test atom $\mathbf{x}_{st} \in \mathbb{R}^{m}$ from a given class can be approximated by a linear combination of the atoms of the same class, yielding:

$$\mathbf{x}_{st} = s_{1}^{c}\mathbf{d}_{1}^{c} + \ldots + s_{n_c}^{c}\mathbf{d}_{n_c}^{c}, \tag{3}$$

where $[s_{1}^{c}, s_{2}^{c}, ..., s_{n_c}^{c}]^{\top} \in \mathbb{R}^{n_c}$ is the corresponding sparse coefficients vector associated with class c.

The class of a given test atom $\mathbf{x}_{st}$ is deduced from the dictionary $\mathbf{D}_{tr}$ by selecting a set of samples of the training atoms which can better approximate $\mathbf{x}_{st}$ by

the associated sparse code $\mathbf{s}$ whose non-zero entries correspond to the c^{th} class. Thus, the sparse representation vector can be expressed as

$$\mathbf{s} = [0, ...0, s_1^c, ..., s_{n_c}^c, ..., 0, 0]^\top \in \mathbb{R}^n \ . \tag{4}$$

Given the dictionary $\mathbf{D}_{tr}$, the approximated sparse vector $\hat{\mathbf{s}}$ of $\mathbf{x}_{st}$ is obtained by solving the Lasso optimization problem:

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \|\mathbf{x}_{st} - \mathbf{D}_{tr}\mathbf{s}\|_2^2 + \lambda_0 \|\mathbf{s}\|_1 \ , \tag{5}$$

where λ_0 is a parameter that controls the balance between the reconstruction error and sparsity.

Finally, the non-zero coefficients in $\hat{\mathbf{s}}$ allow to identify the class of the unlabeled test atom. However, in practice, the non-zero coefficients may be associated with different classes due to the noise or the correlation that may exist within multiple classes. For this reason, the residual between the original test sample $\mathbf{x}_{st}$ and the estimated one is computed for the possible candidate classes, and the predicted class is chosen according to the minimum residual value:

$$class(\mathbf{x}_{st}) = \arg\min_c \|\mathbf{x}_{st} - \mathbf{D}_{tr}\delta_c(\hat{\mathbf{s}})\|_2^2, \tag{6}$$

where $\hat{\mathbf{s}} \in \mathbb{R}^n$ and $\delta_c(\hat{\mathbf{s}})$ is the characteristic function that selects elements in $\hat{\mathbf{s}}$ that are only associated with class c.

3. Proposed multi-branch wavelet neural network architecture and classification scheme

Motivated by the advantages of wavelets in producing multi-scale representation with good space-frequency localization, a wavelet transform is applied to the image dataset. More precisely, the luminance component $\mathbf{X}$ is firstly extracted from each input color image. Then, the Haar transform is performed on the resulting images to generate four wavelet subbands: One approximation subband $\mathbf{X}^{(LL)}$ and three detail subbands oriented horizontally $\mathbf{X}^{(LH)}$, vertically $\mathbf{X}^{(HL)}$ and diagonally $\mathbf{X}^{(HH)}$. Thus, for notation concision, $\mathbf{X}^{(o)}$ denotes

a given wavelet subband at orientation $o \in \mathbb{O}$, where $\mathbb{O} = \{LL, HL, LH, HH\}$ is the set of the generated directional wavelet subbands. By repeating the same decomposition process on the approximation subband, a multi-resolution representation of the input image can be obtained.

Before describing the proposed architecture, let us introduce the following notations. For a given dataset, n, n_{tr}, n_v, and n_{st} represent the respective numbers of all, training, validation and testing samples. Thus, following the wavelet decomposition stage, the resulting subbands, with orientation $o \in \mathbb{O}$, of the training, validation and testing samples are designated by $\mathbf{X}_{tr}^{(o)} \in \mathbb{R}^{m \times n_{tr}}$, $\mathbf{X}_{v}^{(o)} \in \mathbb{R}^{m \times n_{v}}$, and $\mathbf{X}_{st}^{(o)} \in \mathbb{R}^{m \times n_{st}}$, respectively.

3.1. Architecture description

Once the wavelet decomposition is performed and the approximation $\mathbf{X}^{(LL)}$ as well as the horizontal $\mathbf{X}^{(LH)}$, vertical $\mathbf{X}^{(HL)}$ and diagonal $\mathbf{X}^{(HH)}$ detail subbands are obtained, the latter are fed into different neural networks. Thus, a multi-branch wavelet neural network architecture is obtained.

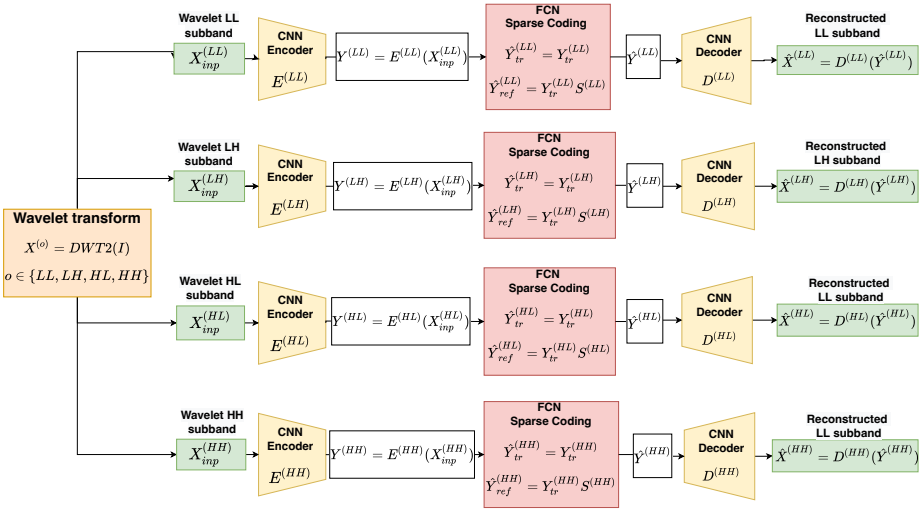


Figure 1: Proposed multi-branch wavelet neural network (MB-WNN) architecture.

It can be observed from Fig. 1 that each wavelet subband is processed through the following three main blocks:

- An encoder $E^{(o)}$ that generates the encoding features $\mathbf{Y}^{(o)}$ in the latent space. Thus, from the input vector $\mathbf{X}_{inp}^{(o)} = [\mathbf{X}_{tr}^{(o)}, \mathbf{X}_{ref}^{(o)}]$, we obtain:

$$\begin{aligned}\mathbf{Y}^{(o)} &= [E^{(o)}(\mathbf{X}_{tr}^{(o)}), E^{(o)}(\mathbf{X}_{ref}^{(o)})] \\ &= [\mathbf{Y}_{tr}^{(o)}, \mathbf{Y}_{ref}^{(o)}]\end{aligned} \tag{7}$$

  where the index ref refers either to the validation or test data. For instance, when the MB-WNN architecture is used in the training (resp. test) phase, $\mathbf{X}_{ref}^{(o)}$ and $\mathbf{Y}_{ref}^{(o)}$ terms correspond to $\mathbf{X}_{v}^{(o)}$ and $\mathbf{Y}_{v}^{(o)}$ (resp. $\mathbf{X}_{st}^{(o)}$ and $\mathbf{Y}_{st}^{(o)}$).

- A sparse coding layer, parameterized by $\mathbf{S}^{(o)}$, which aims to find the sparse encoding features $\hat{\mathbf{Y}}^{(o)}$ from $\mathbf{Y}^{(o)}$.

- A decoder $D^{(o)}$ that consists in reconstructing an output $\hat{\mathbf{X}}^{(o)} = D^{(o)}(\hat{\mathbf{Y}}^{(o)})$ close to the encoder input $\mathbf{X}_{inp}^{(o)}$.

It is important to note that the sparse coding layer plays a crucial role in this architecture. For instance, the estimation of the sparse representation of the encoding features $\mathbf{Y}^{(o)}$ is achieved by minimizing the approximation error under the sparsity constraint. This leads to the following Lasso optimization problem:

$$\min_{\mathbf{S}^{(o)}} \sum_{o \in \mathbb{O}} \left(\lambda_1^{(o)} \left\| \mathbf{Y}_{ref}^{(o)} - \mathbf{Y}_{tr}^{(o)} \mathbf{S}^{(o)} \right\|_F^2 + \lambda_2^{(o)} \left\| \mathbf{S}^{(o)} \right\|_1 \right), \tag{8}$$

where $\mathbf{S}^{(o)} \in \mathbb{R}^{n_{tr} \times n_{ref}}$ is the sparse coefficient matrix, $\lambda_1^{(o)}$ are positive constants weighting the different approximation errors across the different subbands, $\lambda_2^{(o)}$ are positive regularization parameters that control the sparsity penalty and the fidelity between the input and output of the sparse coder, and $\|\cdot\|_F$ is the Frobenius norm.

According to (8), the estimated sparse encoding features $\hat{\mathbf{Y}}_{ref}^{(o)}$ can be seen as the output of a Fully Connected Network (FCN) whose input layer is the encoded feature vector $\mathbf{Y}_{tr}^{(o)}$. Thus, we have:

$$\hat{\mathbf{Y}}_{ref}^{(o)} = \mathbf{Y}_{tr}^{(o)} \mathbf{S}^{(o)} \ . \tag{9}$$

Knowing that the sparse coder's output is given by $\hat{\mathbf{Y}}^{(o)} = [\hat{\mathbf{Y}}_{tr}^{(o)}, \hat{\mathbf{Y}}_{ref}^{(o)}]$ while $\hat{\mathbf{Y}}_{tr}^{(o)} = \mathbf{Y}_{tr}^{(o)}$, problem (8) can be rewritten as

$$\min_{\mathbf{S}^{(o)}} \sum_{o \in \mathbb{O}} \left(\lambda_1^{(o)} \|\mathbf{Y}^{(o)} - \hat{\mathbf{Y}}^{(o)}\|_F^2 + \lambda_2^{(o)} \|\mathbf{S}^{(o)}\|_1 \right). \tag{10}$$

3.2. Learning approach

Once the different blocks are described, we focus on the model learning approach. The proposed MB-WNN model will be trained using an appropriate loss function which aims to achieve the following two objectives for each branch of the architecture:

- Learning a meaningful representation of the input wavelet subbands $\mathbf{X}_{inp}^{(o)}$ by a nonlinear reduction method, using the encoder block of the AE, instead of the linear PCA as performed in SRWC [31]. The decoder block of the AE ensures that the encoded features allow to recover a reconstructed subband $\hat{\mathbf{X}}^{(o)}$ close to $\mathbf{X}_{inp}^{(o)}$. Hence, the loss related to the reconstruction error of the AE, $\mathcal{L}_{AE}^{(o)}$, is given by

$$\mathcal{L}_{AE}^{(o)} = \left\| \mathbf{X}_{inp}^{(o)} - \hat{\mathbf{X}}^{(o)} \right\|_F^2 . \tag{11}$$

- Estimating the sparse representation of the encoding features in the latent space by minimizing both the approximation error (i.e. first term in (10)) and the sparsity penalty constraint (i.e. second term in (10)). Thus, the loss related to the sparse coding layer, $\mathcal{L}_{SC}^{(o)}$, is defined by

$$\mathcal{L}_{SC}^{(o)} = \lambda_1^{(o)} \|\mathbf{Y}^{(o)} - \hat{\mathbf{Y}}^{(o)}\|_F^2 + \lambda_2^{(o)} \|\mathbf{S}^{(o)}\|_1 . \tag{12}$$

Since we are dealing with a multi-branch architecture, the global loss function is defined by considering the weighted sum of the two loss terms $\mathcal{L}_{SC}^{(o)}$ and $\mathcal{L}_{AE}^{(o)}$ associated to each branch. Therefore, the retained loss function of our MB-WNN architecture becomes:

$$\begin{aligned} \mathcal{L}_t &= \sum_{o \in \mathbb{O}} \left(\mathcal{L}_{SC}^{(o)} + \lambda_3^{(o)} \mathcal{L}_{AE}^{(o)} \right) \\ &= \sum_{o \in \mathbb{O}} \left(\lambda_1^{(o)} \|\mathbf{Y}^{(o)} - \hat{\mathbf{Y}}^{(o)}\|_F^2 + \lambda_2^{(o)} \|\mathbf{S}^{(o)}\|_1 + \lambda_3^{(o)} \left\| \mathbf{X}_{inp}^{(o)} - \hat{\mathbf{X}}^{(o)} \right\|_F^2 \right), \end{aligned} \tag{13}$$

where $\lambda_3^{(o)}$ are positive parameters used to weight the reconstruction errors resulting from the auto-encoders applied to the different wavelet subbands (while $\lambda_1^{(o)}$ and $\lambda_2^{(o)}$ have already been defined in (8)). Note that the choice of these weights will be discussed in the next section to better analyze the impact of the different subbands on the classification performance.

Finally, the loss function $\mathcal{L}_t$ is minimized using a standard stochastic gradient-based optimization method to get the optimal parameters of the overall architecture.

3.3. Multiple sparse codes-based classification stage

Once the model is trained, the obtained sparse codes are used in the test phase to proceed with the classification stage. More precisely, the class labels of the test samples will be identified using a residual-based probabilistic rule [30]. In the case of highly inter-correlated data, the probabilistic rule revealed good classification performance compared to the classical approach based on the truncated residual scheme where the samples from different classes may yield the same residual [6]. While the residual-based probabilistic approach has been developed for a single sparse code [29, 30], we propose to generalize it to deal with a multiple sparse code matrix $\mathbf{S}^{(o)}$.

In this respect, for each class $c \in \{1, \ldots, k\}$ and orientation $o \in \mathbb{O}$, the residual term $r_c^{(o)}$ is firstly computed as follows:

$$r_c^{(o)}\left(\mathbf{x}_{st}^{(o)}\right) = \frac{\left\|\mathbf{y}_{st}^{(o)} - \mathbf{Y}_{tr}^{(o)}\delta_c\left(\mathbf{s}^{(o)}\right)\right\|_2^2}{\left\|\delta_c\left(\mathbf{s}^{(o)}\right)\right\|_2^2}, \tag{14}$$

where $\mathbf{x}_{st}^{(o)}$ is the subband at orientation o of the observed sample in the test data $\mathbf{X}_{st}^{(o)}$, $\mathbf{y}_{st}^{(o)}$ is its embedding feature vector and $\mathbf{s}^{(o)}$ is its corresponding sparse code column in the sparse code matrix $\mathbf{S}^{(o)}$.

Then, a probability value $p_c^{(o)}$ is mapped to each residual term $r_c^{(o)}$ using the softmax function:

$$p_c^{(o)} = \frac{e^{-r_c^{(o)}}}{\sum_{c=1}^{k} e^{-r_c^{(o)}}}, \tag{15}$$

where k denotes the number of classes.

Finally, based on a linear combination of the probability values $p_c^{(o)}$ associated with the different subbands at orientation $o \in \mathbb{O}$, the class of the test sample $\mathbf{x}_{st}$ is identified as follows

$$\begin{aligned} class\ (\mathbf{x}_{st}) &= \arg\max_c (p_c) \\ &= \arg\max_c \Big(\sum_{o \in \mathbb{O}} \alpha^{(o)} p_c^{(o)} \Big), \end{aligned} \tag{16}$$

where p_c denotes the probability that the sample $\mathbf{x}_{st}$ belongs to class c, and $\alpha^{(o)}$ is a positive constant (with $\sum_{o \in \mathbb{O}} \alpha^{(o)} = 1$) representing the weight associated to each probability value $p_c^{(o)}$. In other words, $\alpha^{(o)}$ reflects the importance of subband o in the classification stage.

Particular case: single branch architecture

A particular case of the proposed MB-WNN architecture consists in considering *only one* wavelet subband, yielding a single branch architecture. Since wavelet transforms allow to concentrate the main information of a given input image in the low frequency subband, the single branch architecture could be easily deduced by selecting the branch dealing with the approximation subband (i.e. the upper branch of the MB-WNN shown in Fig. 1). While such a single branch model presents the advantage of reducing the complexity of the architecture, it may be less efficient as it does not take into account the detail wavelet coefficients (i.e. edge information) of the input images as addressed in the proposed MB-WNN-based framework.

4. Experimental results

To evaluate the performance of the proposed methods, extensive experiments have been conducted. The first round of experiments is devoted to the analysis of the influence of the different parameters involved in the proposed MB-WNN architecture. The latter is then compared to recent state-of-the-art classification methods.

4.1. Experimental settings

As mentioned in the previous section, each branch of the proposed architecture relies on three main blocks referred to as encoder, sparse coding and decoder. It is important to note here that the retained architecture, applied to each branch, is quite similar to that investigated recently in [28]. However, few modifications related to the kernel sizes are made to reduce the amount of the involved parameters. For instance, a description of the employed architecture is provided in Table 1. To train our models, a two-stage approach is adopted

Table 1: Description of the neural network model used in each branch of the overall architecture.

	Layer	Feature maps		Kernel size	# Param
		Number	Size		
Encoder	Conv2D-1	10	48×48	3×3	208
	Max-pool-1	10	24×24	∅	0
	Conv2D-2	20	24×24	3×3	1168
	Max-pool-2	20	12×12	∅	0
	Conv2D-3	30	12×12	1×1	4640
	Max-pool-3	30	6×6	∅	0
Sparse Coding	FCN	30	6×6	∅	540225
Decoder	Conv2D-4	30	6×6	1×1	9248
	Upsampling-1	30	12×12	∅	0
	Conv2D-5	20	12×12	3×3	4640
	Upsampling-2	20	24×24	∅	0
	Conv2D-6	10	24×24	3×3	1168
	Upsampling-3	10	48×48	∅	0
	Conv2D-7	1	48×48	3×3	208
Param	Total parameters: 561,474				

in our experiments as investigated in [28]. The first one corresponds to a pre-training stage where the model is trained without considering the sparse coding layer (i.e. like a traditional AE model). This step is achieved using 100 epochs. Then, in the second stage, we continue the training of the overall model, while including the sparse coding layer, by using 900 epochs. The models are trained using the ADAM optimizer [33] with the learning rate 10^{-3} while applying a decay of 0.9. These implementations were carried out using Pytorch 2.0 and NVIDIA Quadro RTX 6000 GPU. Note that the source code of the proposed approach is available on github[3].

4.2. Experimental datasets

To validate the effectiveness of the proposed methods, our simulations have been carried out on different types of standard balanced datasets often used for object classification and recognition purposes. More precisely, we have used two *digits* datasets (USPS [34] and SVHN [35]), three *face* datasets (AR face [36], AR gender [36] and UMDAA-01 [37]), and four *natural object* dataset (COIL-100 [38], ETH-80 [39], ARID [40], Tiny ImageNet [41]). Samples of some datasets are shown in Fig. 2.

(a) USPS [34] (b) SVHN [35] (c) AR face [36]

(d) UMDAA-01 [37] (e) COIL-100 [38] (f) ARID [40]

Figure 2: Samples of some employed datasets.

[3]https://github.com/tansyab1/MB-WNN-SRC

Note that 80%, 10% and 10% of each dataset are the proportions of samples which have been randomly selected for training, validation and test phases, respectively.

4.3. Comparison methods and performance evaluation

The proposed single and multi-branch wavelet neural network architectures for sparse representation classification (SRC) will be designated SB-WNN-SRC and MB-WNN-SRC, respectively. It should be noted here that our methods are tested using three resolution levels of the Haar transform while selecting the wavelet subbands obtained at the third level. Our approaches will be compared to different state-of-the-art methods. They include the following conventional sparse representation-based methods:

- SRC [6]: It represents the original sparse representation-based classification method.
- LC-KSVD [12]: Unlike the previous method where the dictionary is formed by all training samples of each class, this method aims to learn a dictionary of small size from a selective dataset. The learning is achieved by using an objective function with label consistency constraint. The optimization problem is solved by applying the K-SVD algorithm [13].
- FDDL [11]: It consists of learning a structured dictionary whose sub-dictionaries have specific class labels. In this respect, a Fisher discrimination criterion is included in the objective function to produce sparse codes.
- SRWC [31]: It is an enhanced version of the original SRC method which is performed in the wavelet transform domain as described in Section 2.
- SCCRC [42]: It is a recent improved version of SRC which consists in combining sparse and collaborative representations.

Moreover, we have considered the following deep learning-based methods:

- Wide ResNet-SRC [18]: It corresponds to sparse representation-based classification method using deep features extracted from a Wide ResNet architecture applied to the original dataset images. Note that other standard deep architectures have also been tested as discussed later.

- WAE-VGG16-SRC [22]: It uses a wavelet-like auto-encoder which firstly consists of applying an encoder to decompose the image in two channels containing low and high frequency information. Then, the low frequency information is fed into a standard network (VGG16) to extract its features. Finally, these features are fused with those extracted from the high frequency information.

- Kernel-SARL [43]: It is based on a kernel formulation of spectral adversarial representation learning framework.

- DSRC [28]: It consists of using an encoder to extract embedding features which are fed into the sparse coding layer. Then, the sparse codes of the features are estimated and used for classification.

It is worth noting that, in addition to Wide ResNet, other existing architectures, developed for deep feature extraction, have also been tested like VGG19 [16], ResNet50 [17] and AE [19] as it will be shown in Fig. 3. It should also be noted that DSRC [28] is the recent state-of-the-art method which is more related to the current work.

The comparison is performed using different criteria. First, the standard classification accuracy criterion is used. Then, the different methods are evaluated in terms of robustness against the proportion of employed training samples. Finally, the size of different neural networks (i.e. the number of parameters) will be provided.

4.4. Results and discussion

Before evaluating the different methods on the test datasets, we should first determine the optimal values of the parameters introduced by the proposed MB-

WNN-SRC approach. Thus, an analysis of the influence of these parameters on the accuracy performance is conducted.

4.4.1. Influence of the parameters introduced by the proposed MB-WNN-SRC approach

With the proposed multi-branch architecture, and according to the loss function defined in (13), the different parameters $\lambda_p^{(o)}$ with $p \in \{1,2,3\}$ and $o \in \mathbb{O}$ should be appropriately selected since their choice may impact the classification performance. On the one hand, for a given subband o, our analysis aims to find the relationship between the different values $\lambda_1^{(o)}$, $\lambda_2^{(o)}$ and $\lambda_3^{(o)}$ which are used to weight respectively the error due to the sparse coding layer, the sparsity of the estimated codes and the reconstruction error. On the other hand, for a given p value, the contributions of the different wavelet subbands o should be established.

Therefore, based on the previous study of the single branch architecture [29], we first propose to set $\lambda_2^{(LL)}$ to 1 and $\lambda_3^{(LL)}$ to 10. Then, different tests are performed to find the best relation between $\lambda_1^{(o)}$ and $\lambda_2^{(o)}$ while assuming

$$\forall o \in \{LL, LH, HL, HH\}, \quad \lambda_1^{(o)} = \eta \lambda_2^{(o)}, \tag{17}$$

$$\forall o \in \{LH, HL, HH\}, \quad \lambda_p^{(o)} = \frac{1}{\omega} \lambda_p^{(LL)}, \quad p \in \{1,2,3\}, \tag{18}$$

where η and ω are positive constants.

While η will range from a small value (set to 0.25) to a high value (set to 10), the tested ω values are chosen by considering three cases:

(i) In the first case, we set ω to 3, which means that the three detail subbands as well as the approximation one have the same level of importance during training, and so they contribute in equal manner to the learned model.

(ii) In the second case, we set ω to 4, which means that less importance is given to the three detail subbands compared to the approximation one.

(iii) In the third case, we set ω to 2, which means that more importance is given to the three detail subbands compared to the approximation one.

Once the training is achieved, the probability-based classification rule, defined in (16), is applied. In this respect, let us define the following weight vector:

$$\boldsymbol{\alpha} = \left(\alpha^{(LL)}, \alpha^{(LH)}, \alpha^{(HL)}, \alpha^{(HH)}\right)^{\top} . \tag{19}$$

The classification accuracy results, obtained with different η values, are provided in Tables 2, 3 and 4 for cases (i), (ii) and (iii), respectively.

Table 2: Influence of the relationship between $\lambda_p^{(o)}$ values on the classification accuracy (%), while setting $\lambda_2^{(LL)} = 1$, $\lambda_3^{(LL)} = 10$, $\omega = 3$ (i.e. case (i)), $\boldsymbol{\alpha} = \left(0.5, 0.2, 0.2, 0.1\right)^{\top}$ and using Eqs. (17)-(18).

Datasets	$\eta = 10$	$\eta = 8$	$\eta = 4$	$\eta = 2$	$\eta = 1$	$\eta = 0.5$	$\eta = 0.25$
USPS [34]	97.00	**97.18**	97.05	97.05	97.00	96.82	96.35
SVHN [35]	69.35	**69.35**	**69.35**	68.55	67.75	67.75	66.05
AR face [36]	97.65	**98.37**	**98.37**	97.70	97.70	97.55	95.55
AR gender [36]	96.55	**97.05**	97.00	96.80	96.48	96.48	95.40
UMDAA-01 [37]	95.00	**95.45**	95.40	94.80	94.40	92.10	92.10
COIL-100 [38]	93.00	**93.20**	92.20	91.65	91.32	90.35	90.35

Table 3: Influence of the relationship between $\lambda_p^{(o)}$ values on the classification accuracy (%), while setting $\lambda_2^{(LL)} = 1$, $\lambda_3^{(LL)} = 10$, $\omega = 4$ (i.e. case (ii)), $\boldsymbol{\alpha} = \left(0.5, 0.2, 0.2, 0.1\right)^{\top}$ and using Eqs. (17)-(18).

Datasets	$\eta = 10$	$\eta = 8$	$\eta = 4$	$\eta = 2$	$\eta = 1$	$\eta = 0.5$	$\eta = 0.25$
USPS [34]	**96.60**	**96.60**	**96.60**	**96.60**	96.00	96.00	96.00
SVHN [35]	**68.90**	**68.90**	68.30	68.30	68.30	67.50	67.50
AR face [36]	97.50	98.30	**98.32**	97.50	97.50	97.50	95.00
AR gender [36]	96.60	**96.80**	**96.80**	**96.80**	96.48	96.48	95.40
UMDAA-01 [37]	**95.00**	**95.00**	**95.00**	94.40	94.40	92.10	92.10
COIL-100 [38]	92.50	**92.80**	**92.80**	91.50	91.50	91.50	90.20

Thus, two main observations can be made from these experiments: First, whatever the importance given to the approximation and detail wavelet subbands

Table 4: Influence of the relationship between $\lambda_p^{(o)}$ values on the classification accuracy (%), while setting $\lambda_2^{(LL)} = 1$, $\lambda_3^{(LL)} = 10$, $\omega = 2$ (i.e. case (iii)), $\boldsymbol{\alpha} = \left(0.5, 0.2, 0.2, 0.1\right)^\top$ and using Eqs. (17)-(18).

Datasets	$\eta = 10$	$\eta = 8$	$\eta = 4$	$\eta = 2$	$\eta = 1$	$\eta = 0.5$	$\eta = 0.25$
USPS [34]	**97.15**	**97.15**	97.00	97.00	97.00	96.50	96.50
SVHN [35]	**69.35**	**69.35**	**69.35**	68.55	67.75	67.75	67.00
AR face [36]	97.50	**98.32**	**98.32**	**98.32**	97.70	97.55	95.55
AR gender [36]	96.55	**97.05**	**97.05**	96.50	96.50	96.50	95.50
UMDAA-01 [37]	**95.45**	**95.45**	**95.45**	94.00	94.00	93.50	93.50
COIL-100 [38]	93.00	**93.17**	92.20	91.50	91.50	90.20	90.20

(controlled by ω), the best accuracy results are achieved with η equal to 8 for all datasets. Second, the highest accuracy values are obtained with ω equal to 3 (Table 2). It is important to note here that setting ω to 2 (Table 4) leads to similar performance to the case where ω is equal to 3 (Table 2). However, a drop in classification performance occurs where ω was set to 4 (Table 3) (i.e. when less importance is given to the three detail subbands compared to the approximation one). This confirms the interest in exploiting both the approximation and detail subbands in the multi-branch architecture. Based on this study, the next simulations will be performed using $\omega = 3$ and $\eta = 8$.

Moreover, regarding the multiple sparse codes-based classification stage, we have also studied the impact of the choice of the weighting terms $\alpha^{(o)}$ associated to the probabilities $p^{(o)}$ as shown in (16). To this end, three cases are considered. The first case is $\boldsymbol{\alpha} = \left(0.5, 0.2, 0.2, 0.1\right)^\top$ which means that the same importance is given to the approximation subband ($\alpha^{(LL)} = 0.5$) and the three detail subbands ($\alpha^{(HL)}+\alpha^{(LH)}+\alpha^{(HH)} = 0.5$). The second case is $\boldsymbol{\alpha} = \left(0.6, 0.15, 0.15, 0.1\right)^\top$ which means that more importance is given to the approximation subband. The third case is $\boldsymbol{\alpha} = \left(0.4, 0.2, 0.2, 0.2\right)^\top$ which means that more importance is given to the three detail subbands.

Table 5 provides the accuracy results for the above considered three weight

Table 5: Influence of the choice of the weight vector $\boldsymbol{\alpha} = \left(\alpha^{(LL)}, \alpha^{(LH)}, \alpha^{(HL)}, \alpha^{(HH)}\right)^\top$ on the classification accuracy (%) of MB-WNN-SRC approach, while setting $\eta = 8$ and $\omega = 3$.

Datasets	$\boldsymbol{\alpha} = \left(0.6, 0.15, 0.15, 0.1\right)^\top$	$\boldsymbol{\alpha} = \left(0.5, 0.2, 0.2, 0.1\right)^\top$	$\boldsymbol{\alpha} = \left(0.4, 0.2, 0.2, 0.2\right)^\top$
USPS [34]	97.05	**97.18**	97.05
SVHN [35]	**69.35**	**69.35**	**69.35**
AR face [36]	**98.37**	**98.37**	97.70
AR gender [36]	**97.05**	**97.05**	**97.05**
UMDAA-01 [37]	**95.45**	**95.45**	95.40
COIL-100 [38]	**93.20**	**93.20**	**93.20**

vectors. Thus, similarly to the previous analysis, giving the same level of importance to the approximation and the three detail subbands yields the highest accuracy values. For this reason, the weight vector $\boldsymbol{\alpha}$ will be set to $\left(0.5, 0.2, 0.2, 0.1\right)^\top$ in the following experiments.

Finally, it should be noted that the performance of the proposed MB-WNN-SRC approach has also been evaluated by considering different neural network structures. This has been achieved by modifying the number of convolution and pooling layers as well as the kernel sizes in each branch. While the kernel size slightly affects the accuracy performance, it has been observed that the number of layers has more impact on the results. For instance, it can be seen from Table 6 that a significant improvement can be achieved by increasing the number of layers up to three. However, adding more layers yields similar performance. For these reasons, we have retained the neural network structure described in Table 1.

4.4.2. Comparison with the state-of-the-art classification methods

Once the proposed MB-WNN-SRC method is analyzed and the optimal parameter values are found (i.e. $\eta = 8$, $\omega = 3$ and $\boldsymbol{\alpha} = (0.5, 0.2, 0.2, 0.1)^\top$), we now focus on the comparison of its performance with the aforementioned state-of-the-art classification methods. The accuracy results of the different methods are reported in Table 7 where the two best values are highlighted in bold.

Table 6: Influence of the number of layers in the proposed MB-WNN architecture on the classification accuracy.

	Number of convolution and pooling layers			
	2	3	4	5
SVHN [35]	68.2640	69.3525	69.3525	69.3540
UMDAA-01 [37]	93.2550	95.4489	95.4489	95.4450
COIL-100 [38]	89.3685	93.2050	93.2050	93.2050
ARID [40]	80.5500	81.3680	81.3680	81.3680
Tiny ImageNet [41]	82.3250	84.7750	84.7750	84.5030

Table 7: Classification accuracy (%) of the proposed approach as well as the state-of-the-art methods.

Methods/ Datasets	USPS [34]	SVHN [35]	AR face [36]	AR gender [36]	UMDAA-01 [37]	COIL-100 [38]	ETH-80 [39]	ARID [40]	Tiny ImageNet [41]
SRC [6]	0.8778	0.1571	0.9761	0.9300	0.7900	0.9116	0.9177	0.6980	0.7115
FDDL [11]	0.9134	0.2254	0.9616	0.9400	0.8122	0.8822	0.9320	0.7115	0.7330
LC-KSVD [12]	0.8745	0.3531	0.9770	0.8680	0.8482	0.9142	0.9425	0.7320	0.7750
SRWC [31]	0.9545	0.2821	0.9839	0.9420	0.8529	0.9229	0.9315	0.7535	0.7660
SCCRC [42]	0.9465	0.6850	0.9363	0.9580	0.9450	0.8910	0.9420	0.7050	0.7840
Wide ResNet-SRC [18]	0.9523	0.5050	0.9833	0.9580	0.8850	0.9221	0.9267	0.7082	0.7840
WAE-VGG16-SRC [22]	0.9625	0.6850	0.9750	0.9650	0.9320	0.9100	0.9720	0.6520	0.7530
Kernel-SARL [43]	0.9680	0.6820	0.9810	**0.9708**	0.9482	0.9235	**0.9775**	**0.8220**	**0.8335**
DSRC [28]	0.9625	0.6775	0.9812	0.9648	0.9339	0.9112	0.9573	0.7890	0.8078
SB-WNN-SRC	**0.9682**	**0.6824**	**0.9837**	0.9654	**0.9510**	**0.9235**	0.9625	0.7930	0.8120
MB-WNN-SRC	**0.9718**	**0.6935**	**0.9837**	**0.9705**	**0.9545**	**0.9320**	**0.9877**	**0.8137**	**0.8478**

It should be noted that, in addition to the deep learning based-methods kernel-SARL [43] and DSRC [28], we only reported in Table 7 the results of Wide ResNet-SRC [18] and WAE-VGG16-SRC [22]. The latter have been selected since they have shown better performance than the remaining neural networks-based feature extraction techniques as illustrated in Fig. 3. Thus, it can be firstly seen from Table 7 that the particular SB-WNN-SRC method outperforms the existing ones for most of the employed datasets. Further improvements are

achieved thanks to the extended multi-branch architecture (MB-WNN-SRC). For instance, the multi-branch architecture achieves higher gain compared to the single branch variant, especially for the SVHN [35], COIL-100 [38], ETH-80 [39], ARID [40] and Tiny ImageNet [41] datasets. This can be explained by the fact that the multi-branch architecture combines various sparse codes produced from both approximation and detail wavelet subbands of the image dataset.

Moreover, and since neural networks are well known to be very efficient when they are trained on large scale datasets, we propose to evaluate the performance of different deep learning-based methods with respect to the number of labeled training samples. For this reason, for each employed dataset, four subsets are created by randomly selecting 20%, 40%, 60% and 80% of the samples of the whole dataset. Then, after separating the training/validation/testing samples in each subset, we tested the different classification methods. Fig. 3 shows the performance of these methods for the different employed datasets while varying their sizes. As previously stated, the plots confirm the high efficiency of the recent DSRC [28] method as well as the WAE-VGG16-SRC [22] one compared to the previous deep learning-based methods (i.e VGG19-SRC [16], ResNet50-SRC [17], Wide ResNet-SRC [18] and AE-SRC [19]) aiming at extracting deep features from original images. Regarding the effect of the dataset size, it can be observed that the aforementioned neural networks-based methods fail and result in a significant drop of performance when the number of samples is relatively small. However, the recent methods WAE-VGG16-SRC [22] and DSRC [28] as well as the proposed method are less sensitive to the training image dataset size. Most importantly, our MB-WNN-SRC method outperforms the state-of-the-art approaches and appears more robust to the size of the employed training dataset.

Finally, the different deep learning-based methods have been compared in terms of number of the network parameters and execution time as shown in Table 8 and Table 9, respectively. Indeed, one can see that the proposed multi-branch architecture has more parameters compared to the DSRC and SB-WNN models, since the latter are based on a single branch architecture. However, the number

of parameters in our MB-WNN architecture are much smaller than that of the other neural networks (VGG19, ResNet50, etc) based methods. It should be noted here that the reduced number of parameters explains the good behavior of the proposed methods, whatever the size of the training dataset. Another advantage of designing an efficient model with a small size is it allows to achieve a gain in storage memory.

Similarly to the complexity of the different methods in terms of number of parameters, the execution times obtained with ARID image dataset (of size 256× 256) show that our proposed multi-branch architecture requires an additional time of about 0.4 seconds compared to the single branch and DSRC approaches. However, it is much faster than the remaining neural networks-based methods.

Table 8: Evaluation of different deep neural networks in terms of number of parameters.

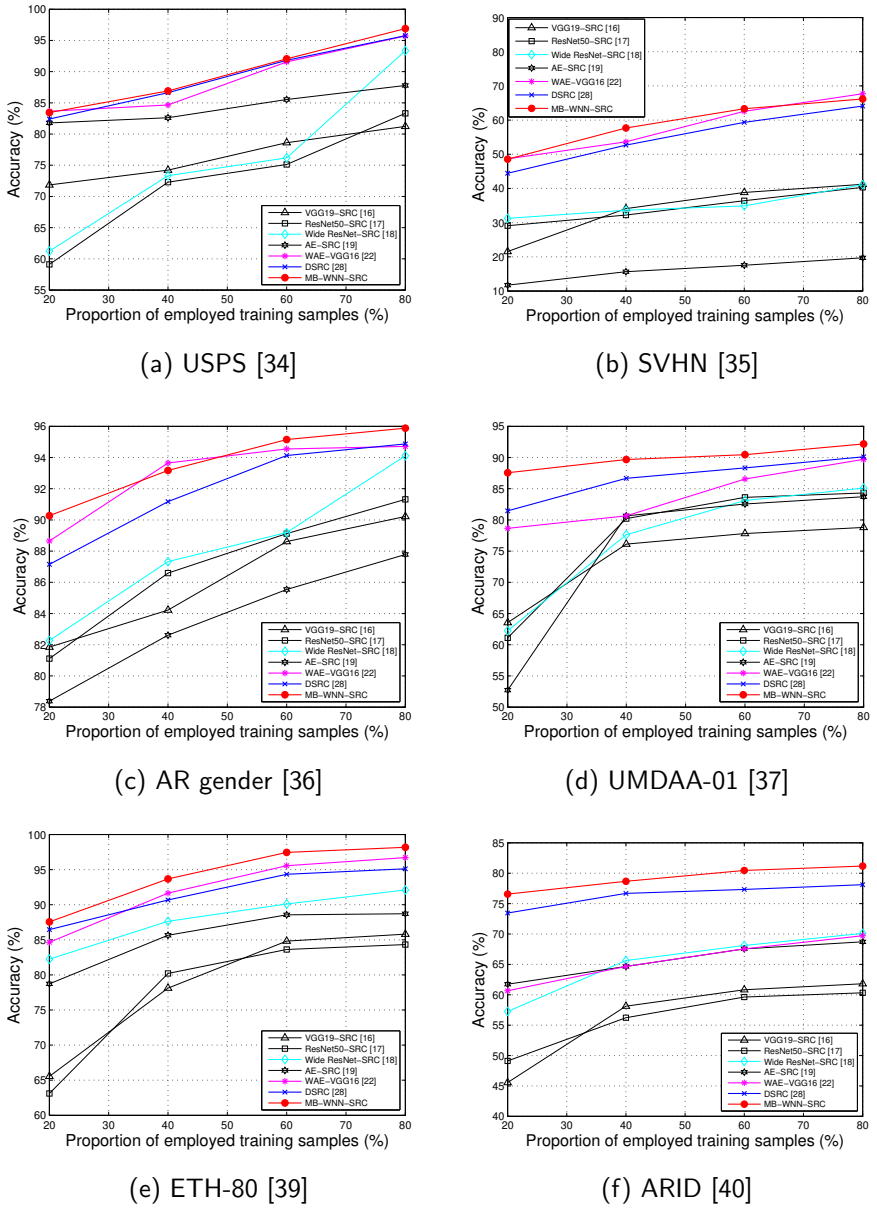
	VGG19-SRC [16]	ResNet50-SRC [17]	Wide ResNet-SRC [18]	WAE-VGG16-SRC [22]	Kernel-SARL [43]	DSRC [28]	**SB-WNN-SRC**	**MB-WNN-SRC**
#param	138M	25.6M	8.8M	57.4M	2.83M	24.5K	**12.7K**	2.25M

Table 9: Execution times (in seconds) for different deep learning-based classification methods.

Method	VGG19-SRC [16]	ResNet50-SRC [17]	Wide ResNet-SRC [18]	WAE-VGG16-SRC [22]	Kernel-SARL [43]	DSRC [28]	**SB-WNN-SRC**	**MB-WNN-SRC**
Time	4.91	2.36	1.36	2.87	1.02	0.19	0.15	0.57

5. Conclusion and perspective

An object classification method, exploiting simultaneously the advantages of neural networks as well as sparse coding techniques and multi-scale wavelet decompositions, is proposed. More precisely, a set of auto-encoders combined with sparse coding layers are applied to different wavelet subbands yielding a new multi-branch neural network architecture. Unlike existing sparse representation classification methods, the proposed architecture presents two main



(a) USPS [34] (b) SVHN [35]

(c) AR gender [36] (d) UMDAA-01 [37]

(e) ETH-80 [39] (f) ARID [40]

Figure 3: Effect of the training dataset size on the accuracy for different deep learning-based methods.

advantages: First, it exploits both low and high frequency information located in the approximation and detail wavelet subbands. Secondly, it allows to pro-

duce various sparse codes resulting in better discrimination ability. However, the main limitation of the proposed approach is the complexity of the architecture, since the number of involved branches is directly related to the retained wavelet subbands. Overall, the simulations carried out on various types of standard datasets have shown the benefits that can be drawn from the proposed architecture. Despite its good performance, the proposed architecture could be improved with further investigation. It is important to recall that the training of our architecture aims only to find the optimal sparse codes that are then exploited in the test phase for classification purposes. Therefore, a more efficient architecture could be designed by integrating the classification stage and resorting to an end-to-end learning approach. Moreover, as the proposed architecture operates in the wavelet transform domain, it would be interesting to investigate other decompositions like the recent neural network-based multiscale transforms [44, 45].

Funding

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 722068.

References

[1] Z. Yang, Q. Li, L. Wenyin, J. Lv, Shared multi-view data representation for multi-domain event detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (5) (2020) 1243–1256.

[2] P. Li, B. Chen, D. Wang, H. Lu, Visual tracking by dynamic matching-classification network switching, Pattern Recognition 107 (2020) 107419.

[3] Z. A. Khan, A. Beghdadi, M. Kaaniche, F. A. Cheikh, Residual networks based distortion classification and ranking for laparoscopic image quality assessment, in: IEEE International Conference on Image Processing (ICIP), 2020, pp. 176–180.

[4] M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, Springer Science and Business Media, USA, 2010.

[5] J. Lee, S.-U. Cheon, J. Yang, Connectivity-based convolutional neural network for classifying point clouds, Pattern Recognition 112 (2021) 107708.

[6] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.

[7] C. Zhang, S. Wang, Q. Huang, J. Liu, Image classification using spatial pyramid robust sparse coding, Pattern Recognition Letters 34 (9) (2013) 1046–1052.

[8] S. H. Gao, I. W.-H. Tsang, L.-T. Chia, Kernel sparse representation for image classification and face recognition, in: European Conference on Computer Vision, 2010, pp. 1–14.

[9] C.-G. Li, J. Guo, H.-G. Zhang, Local sparse representation based classification, in: International Conference on Pattern Recognition, 2010, pp. 649–652.

[10] Y. Wang, Y. Y. Tang, L. Li, X. Zheng, Block sparse representation for pattern classification: Theory, extensions and applications, Pattern Recognition 88 (2019) 198–209.

[11] M. Yang, L. Zhang, X. Feng, D. Zhang, Fisher discrimination dictionary learning for sparse representation, in: 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 543–550.

[12] Z. Jiang, Z. Lin, L. S. Davis, Label consistent K-SVD: Learning a discriminative dictionary for recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11) (2013) 2651–2664.

[13] Q. Zhang, B. Li, Discriminative K-SVD for dictionary learning in face recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, CA, USA, 2010, pp. 2691–2698.

[14] J. Song, X. Xie, G. Shi, W. Dong, Multi-layer discriminative dictionary learning with locality constraint for image classification, Pattern Recognition 91 (2019) 135–146.

[15] Y. Li, Y. Chai, H. Zhou, H. Yin, A novel dimension reduction and dictionary learning framework for high-dimensional data classification, Pattern Recognition 112 (2021) 107793.

[16] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, CA, USA, 2015.

[17] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nevada, USA, 2016, pp. 770–778.

[18] S. Zagoruyko, N. Komodakis, Wide residual networks, in: Proceedings of the British Machine Vision Conference (BMVC), no. 87, York, UK, 2016, pp. 1–12.

[19] A. Gogna, A. Majumdar, Discriminative autoencoder for feature extraction: Application to character recognition, Neural Processing Letters 49 (4) (2018) 1723–1735.

[20] S. Said, O. Jemai, S. Hassairi, R. Ejbali, M. Zaied, C. Ben Amar, Deep wavelet network for image classification, in: IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 2016, pp. 922–927.

[21] T. Williams, R. Li, Advanced image classification using wavelets and convolutional neural networks, in: IEEE International Conference on Machine Learning and Applications (ICMLA), CA, USA, 2016, pp. 233–239.

[22] T. Chen, L. Lin, W. Zuo, X. Luo, L. Zhang, Learning a wavelet-like auto-encoder to accelerate deep neural networks, in: AAAI Conference on Artificial Intelligence, Louisiana, USA, 2018, pp. 6722–6729.

[23] Q. Li, L. Shen, S. Guo, Z. Lai, Wavelet integrated CNNs for noise-robust image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 7243–7252.

[24] P. Liu, H. Zhang, W. Lian, W. Zuo, Multi-level wavelet convolutional neural networks, IEEE Access 7 (2019) 74973–74985.

[25] S. Zhang, J. Wang, X. Tao, Y. Gong, N. Zheng, Constructing deep sparse coding network for image classification, Pattern Recognition 64 (2017) 130–140.

[26] L. Feng, W. Wei, J. Zurada, Sparse representation learning of data by autoencoders with $\ell_{1/2}$ regularization, Neural Network World 28 (2018) 133–147.

[27] X. Sun, N. M. Nasrabadi, T. D. Tran, Supervised deep sparse coding networks for image classification, IEEE Transactions on Image Processing 29 (2019) 405 – 418.

[28] M. Abavisani, V. M. Patel, Deep sparse representation-based classification, IEEE Signal Processing Letters 26 (6) (2019) 948–952.

[29] T.-S. Nguyen, L. H. Ngo, M. Luong, M. Kaaniche, A. Beghdadi, Convolution autoencoder-based sparse representation wavelet for image classification, in: IEEE International Workshop on Multimedia Signal Processing (MMSP), 2020, pp. 1–6.

[30] J. Wei, J. Lv, C. Xie, A new sparse representation classifier (SRC) based on probability judgement rule, in: International Conference on Information System and Artificial Intelligence (ISAI), Hong Kong, China, 2016, pp. 338–342.

[31] L. H. Ngo, M. Luong, N. M. Sirakov, T. Le-Tien, S. Guerif, E. Viennet, Sparse representation wavelet based classification, in: IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 2974–2978.

[32] E. Mooi, M. Sarstedt, I. Mooi-Reci, Principal component analysis and factor analysis, in: Principal Component Analysis, Springer Series in Statistics, New York, NY, 1986.

[33] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, San Siego, USA, 2015, pp. 1–15.

[34] J. Hull, Database for handwritten text recognition research, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 550 – 554.

[35] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 2011.

[36] A. Martinez, R. Benavente, The AR face database, Tech. rep., Ohio State University, Barcelona, Spain (01 1998).

[37] H. Zhang, V. M. Patel, S. Shekhar, R. Chellappa, Domain adaptive sparse representation-based classification, in: IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, Vol. 1, Ljubljana, Slovenia, 2015, pp. 1–8.

[38] S. A. Nene, S. K. Nayar, H. Murase, Columbia Object Image Library (COIL-20), Tech. rep., Department of Computer Science, Columbia University (Feb 1996).

[39] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, 2003, pp. II–409.

[40] M. R. Loghmani, B. Caputo, M. Vincze, Recognizing objects in-the-wild: Where do we stand?, in: IEEE International Conference on Robotics and Automation (ICRA), 2018.

[41] Y. Le, X. Yang, Tiny imagenet visual recognition challenge, CS 231N 7 7 (2015) 3.

[42] Z.-Q. Li, J. Sun, X.-J. Wu, H.-F. Yin, Multiplication fusion of sparse and collaborative-competitive representation for image classification, International Journal of Machine Learning and Cybernetics 11 (2020) 2357–2359.

[43] B. Sadeghi, R. Yu, V. N. Boddeti, On the global optima of kernelized adversarial representation learning, International Conference on Computer Vision (ICCV) (2019) 7970–7978.

[44] L. Li, L.-J. Ma, L. Jiao, F. Liu, Q. Sun, J. Zhao, Complex contourlet-CNN for polarimetric SAR image classification, Pattern Recognition 100 (2020) 107110.

[45] T. Dardouri, M. Kaaniche, A. Benazza-Benyahia, J.-C. Pesquet, Dynamic neural network for lossy to lossless image coding, IEEE Transactions on Image Processing 31 (2021) 569–584.