Deep Discriminative Model for Video Classification

Mohammad Tavakolian(✉) and Abdenour Hadid

Center for Machine Vision and Signal Analysis (CMVS),
University of Oulu, Oulu, Finland
{mohammad.tavakolian,abdenour.hadid}@oulu.fi

Abstract. This paper presents a new deep learning approach for video-based scene classification. We design a Heterogeneous Deep Discriminative Model (HDDM) whose parameters are initialized by performing an unsupervised pre-training in a layer-wise fashion using Gaussian Restricted Boltzmann Machines (GRBM). In order to avoid the redundancy of adjacent frames, we extract spatiotemporal variation patterns within frames and represent them sparsely using Sparse Cubic Symmetrical Pattern (SCSP). Then, a pre-initialized HDDM is separately trained using the videos of each class to learn class-specific models. According to the minimum reconstruction error from the learnt class-specific models, a weighted voting strategy is employed for the classification. The performance of the proposed method is extensively evaluated on two action recognition datasets; UCF101 and Hollywood II, and three dynamic texture and dynamic scene datasets; DynTex, YUPENN, and Maryland. The experimental results and comparisons against state-of-the-art methods demonstrate that the proposed method consistently achieves superior performance on all datasets.

1 Introduction

Through the recent surge in digital content, video data has become an indisputable part of today's life. This has stimulated the evolution of advanced approaches for a wide range of video understanding applications. In this context, the understanding and classification of video content have gained a substantial research interest among the computer vision community. However, the automatic classification of scene in videos is subject to a number of challenges, including a range of natural variations in short videos such as illumination variations, viewpoint changes, and camera motions. Moreover, scene classification differs from the conventional object detection or classification, because a scene is composed of several entities which are often organized in a random layout. Therefore, devising an accurate, efficient and robust representation of videos is essential to deal with these challenges.

To achieve an effective representation of a scene in videos, we can model videos' spatiotemporal motion patterns using the concept of dynamic textures. Videos comprise dynamic textures which inherently exhibit spatial and temporal

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11208, pp. 401–418, 2018.
https://doi.org/10.1007/978-3-030-01225-0_24

regularities of a scene or an object. Dynamic textures widely exist in real-world video data, e.g. regular rigid motion like windmill, chaotic motion such as smoke and water turbulences, and sophisticated motion caused by camera panning and zooming. The modeling of dynamic textures in videos is challenging but very important for computer vision applications such as video classification, dynamic texture synthesis, and motion segmentation.

Despite all challenges, great efforts have been devoted to find a robust and powerful solution for video-based scene classification tasks. Furthermore, it has been commonly substantiated that an effective representation of the video content is a crucial step towards resolving the problem of dynamic texture classification. In previous years, a substantial number of approaches for video representation have been proposed, e.g. Linear Dynamic System (LDS) based methods [1], Local Binary Pattern (LBP) based methods [2], and Wavelet-based methods [3]. Unfortunately, the current methods are sensitive to a wide range of variations such as viewpoint changes, object deformations, and illumination variations. Coupled with these drawbacks, other methods frequently model the video information within consecutive frames on a geometric surface represented by a subspace [4], a combination of subspaces [5], a point on the Grassmann manifold [6], or Lie Group of Riemannian manifold [7]. These require prior assumptions regarding specific category of the geometric surface on which samples of the video are assumed to lie.

On the other hand, deep learning has recently achieved significant success in a number of areas [8–10], including video scene classification [11–14]. Unlike the conventional methods, which fail to model discontinuous rigid motions, deep learning based approaches have a great modeling capacity and can learn discriminative representations in videos. However, the current techniques have mostly been devised to deal with fixed-length video sequences. They fail to deal with long sequences due to their limited temporal coverage. This paper presents a novel deep learning approach, which does not assume any biased knowledge about the concept of data and it automatically explores the structure of the complex non-linear surface on which the samples of the video are present. According to the block diagram in Fig. 1, our proposed method defines a Heterogeneous Deep Discriminative Model (HDDM) whose weights are initialized by an unsupervised layer-wise pre-training stage using Gaussian Restricted Boltzmann Machines (GRBM) [15]. The initialized HDDM is then separately trained for each class using all videos of that class in order to learn a Deep Discriminative Model (DDM) for every class. The training is done so that the DDM learns to specifically represent videos of that class. Therefore, a class specific model is made to learn the structure and the geometry of the complex non-linear surface on which video sequences of that class exist. Also, we represent the raw video data using Sparse Cubic Symmetrical Pattern (SCSP) to capture long-range spatiotemporal patterns and reduce the redundancy between adjacent frames. For the classification of a given query video, we first represent the video based on the learnt class-specific DDMs. The representation errors from the respective DDMs are then computed and a weighted voting strategy is used to assign a class label to the query video.

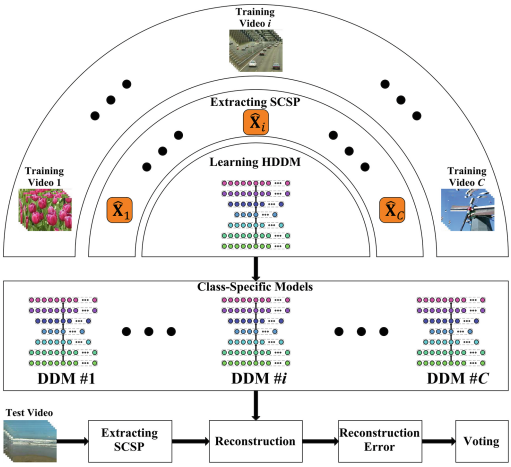

Fig. 1. The block diagram of the proposed DDM framework for video classification.

The main contributions of our proposed Deep Discriminative Model (DDM) are the followings. First, a novel deep learning based framework is introduced for video classification (Sect. 3). Moreover, we develop a Sparse Cubic Symmetrical Pattern (SCSP) to avoid the redundancy in video sequences and reduce the computational cost, and a weighted voting strategies is utilized for classification (Sect. 4). Finally, extensive experiments are conducted along with comparisons against state-of-the-art methods for video classification. The experimental results demonstrate that the proposed method achieves superior performance compared to the state-of-the-art methods (Sect. 5).

2 Related Work

Several approaches have been proposed for video classification [3,16,17]. A popular approach is Linear Dynamic System (LDS) [1,16], which is known as a probabilistic generative model defined over space and time. LDS approximates hidden states using Principal Component Analysis (PCA), and describes their trajectory as time evolves. LDS has obvious drawbacks due to its sensitivity to external variations. In order to overcome this limitation, Closed-Loop LDS (CLDS) [18] was proposed. However, CLDS tends to fail to capture some discontinuous rigid motions due to its simplistic linearity. Local Binary Pattern (LBP) based methods [2] have been widely used in texture analysis. Zhao *et al.* [19] extended LBP to the space and the time domains and proposed two LBP variants: (1) Volume Local Binary Pattern (VLBP) [19] which combines both the spatial and the temporal variations of the video; (2) Local Binary Pattern on Three Orthogonal Planes (LBP-TOP) [19], which computes LBP in three individual $x - y$, $x - t$,

and $y-t$ planes to describe the video. Likewise, other versions of LBP-TOP, such as Local Ternary Pattern on Three Orthogonal Planes (LTP-TOP) [20] and Local Phase Quantization on Three Orthogonal Planes (LPQ-TOP) [20], have been proposed. Although they are all effective in capturing the spatiotemporal information, they rarely achieve a satisfactory performance in the presence of camera motions.

Recently, there is a huge growing research interest in deep learning methods in various areas of computer vision, beating the state-of-the-art techniques [9,11–14]. Deep learning methods set up numerous recognition records in image classification [21], object detection [22], face recognition and verification [23], and image set classification [10]. Deep models, such as Deep Belief Networks and stacked autoencoders, have much more expressive power than traditional shallow models and can be effectively trained with layer-wise pre-training and fine-tuning [24]. Stacked autoencoders have been successfully used for feature extraction [25]. Also, they can be used to model complex relationships between variables due to the composition of several levels of non-linearity [25]. Xie *et al.* [26] modeled relationships between noisy and clean images using stacked denoising autoencoders. Although, deep autoencoders are rarely used to model time series data, there are researches on using variants of Restricted Boltzmann Machine (RBM) [27] for specific time series data such as human motion [28]. On the other hand, some convolutional architectures have been used to learn spatiotemporal features from video data [29]. Kaparthy *et al.* [11] used a deep structure of Convolutional Neural Networks (CNN) and tested it on a large scale video dataset. By learning long range motion features via training a hierarchy of multiple convolutional layers, they showed that their framework is just marginally better than single frame-based methods. Simonyan *et al.* [12] designed Two-Stream CNN which includes the spatial and the temporal networks. They took advantage of ImageNet dataset for pre-training and calculated the optical flow to explicitly capture the motion information. Tran *et al.* [13] investigated 3D CNNs [30] on realistic (captured in the wild) and large-scale video datasets. They tried to learn both the spatial and temporal features with 3D convolution operations. Sun *et al.* [14] proposed a factorized spatiotemporal CNN and exploited different ways to decompose 3D convolutional kernels.

The long-range temporal structure plays an important role in understanding the dynamics of events in videos. However, mainstream CNN frameworks usually focus on appearances and short-term motions. Thus, they lack the capacity to incorporate the long-range temporal structure. Recently, few other attempts (mostly relying on dense temporal sampling with a pre-defined sampling interval) have been proposed to deal with this problem [31,32]. This approach would incur excessive computational cost and is not applicable to real-world long video sequences. It also poses the risk of missing important information for videos that are longer than the maximal sequence length. Our proposed method deals with this problem by extracting Sparse Cubic Symmetrical Patterns (SCSP) from video sequences to feed its autoencoder structure (Sect. 4.1). In terms of spatiotemporal structure modeling, a key observation is that consecutive frames are

highly redundant. Therefore, dense temporal sampling, which results in highly similar sampled frames, is unnecessary. Instead, a sparse spatiotemporal representation will be more favorable in this case. Also, autoencoders reduce the dimension and keep as much important information as possible, and remove noise. Furthermore, combining them with RBMs helps the model to learn more complicated video structures based on their non-linearity.

3 The Proposed Deep Discriminative Model

We first define a Heterogeneous Deep Discriminative Model (HDDM) which will be used to learn the underlying structure of the data in Sect. 4.2. The architecture of the HDDM is shown in Fig. 2. Generally, an appropriate parameter initialization is inevitable for deep neural networks to have a satisfactory performance. Therefore, we initialize the parameters of HDDM by performing pre-training in a greedy layer-wise framework using Gaussian Restricted Boltzmann Machines. The initialized HDDM is separately fine-tuned for each of the C classes of the training videos. Therefore, we end up with a total of C fine-tuned Deep Discriminative Models (DDMs). Then, the fined-tuned models are used for video classification.

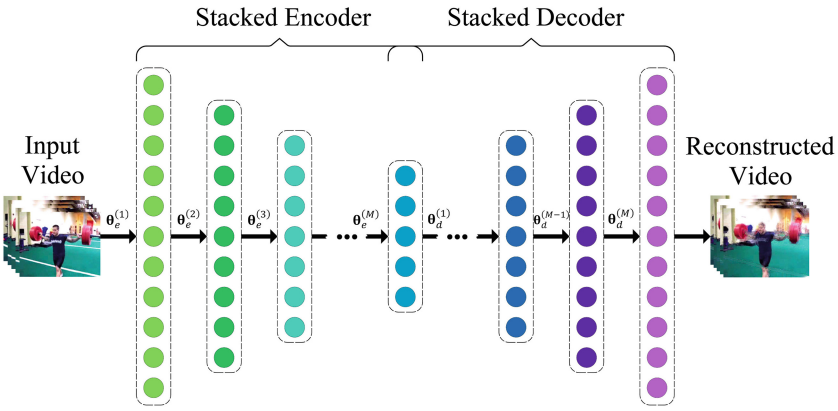

Fig. 2. The configuration of the proposed Heterogeneous Deep Discriminative Model.

3.1 The Heterogeneous Deep Discriminative Model

As can be seen in Fig. 2, the proposed HDDM is based on an autoencoder which comprises multiple encoder and decoder layers. In the proposed autoencoder structure, both the encoder and the decoder have M hidden layers each such that the M-th layer of the encoder is considered as the first layer of the decoder. The encoder section represents the input data in a lower dimension. The encoder consists of a combination of non-linear functions $s(\cdot)$ used to map the input data $\mathbf{x}$ to a representation $\mathbf{h}$ given by

$$\mathbf{h} = s\left(\mathbf{x}\middle|\boldsymbol{\theta}_e^{(1)}, \boldsymbol{\theta}_e^{(2)}, \ldots, \boldsymbol{\theta}_e^{(M)}\right) \tag{1}$$

where $\boldsymbol{\theta}_e^{(i)} = \left\{\mathbf{W}_e^{(i)}, \mathbf{b}_e^{(i)}\right\}$ denotes the parameters of the i-the encoder layer. So, $\mathbf{W}_e^{(i)} \in \mathbb{R}^{n_i \times n_{i-1}}$ is the encoder weight matrix for layer i having n_i nodes, $\mathbf{b}_e^{(i)} \in \mathbb{R}^{n_i}$ is the bias vector and $s(\cdot)$ is a non-linear sigmoid activation function. The encoder parameters are learnt by combining the encoder with the decoder and jointly train the encoder-decoder structure to represent the input data by optimizing a cost function. Hence, the decoder can be defined as series of non-linear functions, which calculate an approximation of the input $\mathbf{x}$ from the encoder output $\mathbf{h}$. The approximated output $\tilde{\mathbf{x}}$ of the decoder is obtained by

$$\tilde{\mathbf{x}} = s\left(\mathbf{h}\middle|\boldsymbol{\theta}_d^{(1)}, \boldsymbol{\theta}_d^{(2)}, \ldots, \boldsymbol{\theta}_d^{(M)}\right) \tag{2}$$

where $\boldsymbol{\theta}_d^{(j)} = \left\{\mathbf{W}_d^{(j)}, \mathbf{b}_d^{(j)}\right\}$ are the parameters of the j-the decoder layer. Consequently, we represent the complete encoder-decoder structure by its parameters $\boldsymbol{\theta}_{HDDM} = \{\boldsymbol{\theta}_{\mathbf{W}}, \boldsymbol{\theta}_{\mathbf{b}}\}$, where $\boldsymbol{\theta}_{\mathbf{W}} = \left\{\mathbf{W}_e^{(i)}, \mathbf{W}_d^{(i)}\right\}_{i=1}^{M}$ and $\boldsymbol{\theta}_{\mathbf{b}} = \left\{\mathbf{b}_e^{(i)}, \mathbf{b}_d^{(i)}\right\}_{i=1}^{M}$.

3.2 Parameter Initialization

We train the defined HDDM with videos of each class individually, which results in class-specific models. The training is performed through stochastic gradient descent with back propagation [33]. The training may not yield to desirable results if the HDDM is initialized with inappropriate weights. Thus, the parameters of the model are first initialized through an unsupervised pre-training phase. For this purpose, a greedy layer-wise approach is adopted and Gaussian RBMs [15] are used.

Basically, a standard RBM [27] is used for binary stochastic data. We therefore use an extension of RBM to process real valued data by appropriate modifications in its energy function. Gaussian RBM (GRBM) [15] is one such popular extension whose energy function is defined by changing the bias term of the visible layer.

$$E_{GRBM}(v, h) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j c_j h_j - \sum_{ij} w_{ij} \frac{v_i}{\sigma_i} h_j \tag{3}$$

where $\mathbf{W}$ is the weight matrix, and $\mathbf{b}$ and $\mathbf{c}$ are the biases of the visible and the hidden layer, respectively. We use a numerical technique called Contrastive Divergence (CD) [34] to learn the model parameter $\{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ of the GRBM in the training phase. v_i and h_j denote the visible layer and the hidden layer's nodes, respectively. Also, σ_i is the standard deviation of the real valued Gaussian distributed inputs to the visible node v_i. It is possible to learn σ_i for each visible unit but it becomes arduous when using CD for GRBM parameter learning. We therefore use another approach and set σ_i to a constant value.

Since there are no intra-layer node connections, result derivation becomes easily manageable for the RBM to the contrary of most directed graphical models. The probability distributions for GRBM are given by

$$\rho\left(h_j\,|\mathbf{v}\right) = s\left(\sum\nolimits_i w_{ij}v_i + c_j\right) \tag{4}$$

$$\rho\left(v_i\,|\mathbf{h}\right) = \frac{1}{\sigma_i\sqrt{2\pi}}\exp\left(\frac{-\left(v_i - u_i\right)^2}{2\sigma_i^2}\right) \tag{5}$$

where

$$u_i = b_i + \sigma_i^2 \sum_i w_{ij}h_j \tag{6}$$

Since our data are real-valued, we use GRBMs to initialize the parameters of the proposed HDDM. In this case, we consider two stacked layers at a time to obtain the GRBM parameters during the learning process. First, the input layer nodes and the first hidden layer nodes are considered as the visible units v and the hidden unit h of the first GRBM, respectively, and their respective parameters are obtained. The activations of the first GRBM's hidden units are then used as an input to train the second GRBM. We repeat this process for all four encoder hidden layers. The weights learnt for the encoder layers are then tied to the corresponding decoder layers.

4 Video Classification Procedure

In this section, we describe how to classify query videos using the representation error. Assume that there are C training videos $\{\mathbf{X}_c\}_{c=1}^{C}$ with the corresponding class labels $y_c \in \{1, 2, \cdots, C\}$. A video sequence is denoted by $\mathbf{X}_c = \left\{\mathbf{x}^{(t)}\right\}_{t=1}^{T}$, where $\mathbf{x}^{(t)}$ contains raw pixel values of the frame at time t. The problem is to assign class y_q to the query video $\mathbf{X}_q$.

4.1 Sparse Cubic Symmetrical Patterns

We represent dynamic textures by video blocks, video volumes spanning over both the spatial and temporal domains, to jointly model the spatial and temporal information. Since there are strong correlations between adjacent regions of scenes (which cause redundancy), we devise an approach to extract a sparse representation of the spatiotemporal encoded features. As a result, the less important information is discarded which makes the deep discriminative model representation more efficacious. For this purpose, we design a volumetric based descriptor to capture the spatiotemporal variations in the scene. Given a video, we first decompose it into a batch of small cubic spatiotemporal volumes. We only consider the video cubes of small size ($w \times h \times d$ pixels), which consists of relatively simple content that can be generated with few components.

Figure 3 illustrates the feature extraction process. We divide each series of frames $\mathbf{X} = \left\{\mathbf{x}^{(t)}\right\}_{t=1}^{T}$ into $w \times h \times d$ distinct non-overlapping uniformly spaced

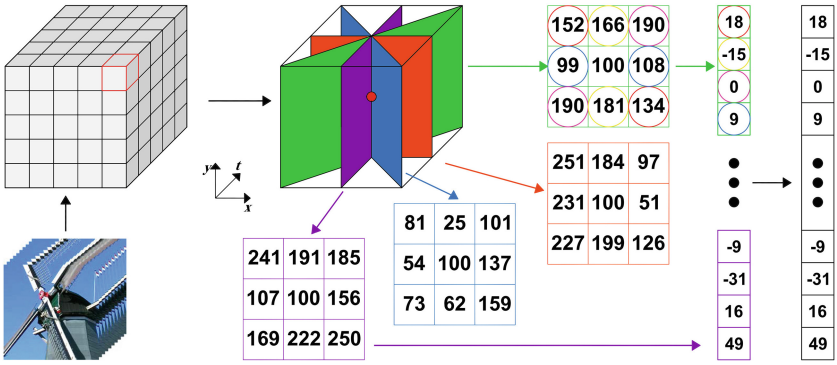

Fig. 3. An example of extracting symmetric signed magnitude variation pattern in a volumetric block of the video sequence.

cubic blocks and extract symmetric spatiotemporal variation pattern for every block which results in a feature vector of the corresponding block. Consequently, each video sequence $\mathbf{X}$ is encoded in terms of the symmetric signed magnitude variation patterns, denoted as $\mathbf{x}_E \in \mathbb{R}^d$, obtained by concatenating the feature vectors of all cubic blocks spanning over the entire video sequence. We do not consider the last one or two frames of video sequence if they do not fit in the cubic block structure. This does not affect the algorithm's performance due to the correlation between consecutive frames.

Given any pixel $x_o^{(t)}$, we represent the neighboring pixels by $x_1^{(t)}, \cdots, x_P^{(t)}$. The symmetric spatiotemporal variations for j-th plane is calculated as

$$F_j\left(x_o^{(t)}\right) = \biguplus_{p=1}^{\frac{P}{2}} \left(x_p^{(t)} - x_{p+\frac{P}{2}}^{(t)}\right) \tag{7}$$

where $x_p^{(t)}$ and $x_{p+\frac{P}{2}}^{(t)}$ are two symmetric neighbors of pixel $x_o^{(t)}$. Also, $\biguplus$ denotes the concatenation operator.

The aforementioned feature vectors are organized into the columns of a matrix $\mathbf{D} \in \mathbb{R}^{d \times N}$, where d is the dimension of the feature vectors and N is the total number of videos. In variable-length videos, we temporally partition the video sequences into non-overlapping segments with a fixed length k and extract features from the cubic blocks within each segment, separately. Then, we place the extracted features of each section into a column of the matrix $\mathbf{D}$. Here, we call matrix $\mathbf{D}$ the dictionary and aim to find a sparse representation $\hat{\mathbf{x}}$ of the encoded video $\mathbf{x}_E$, $\mathbf{x}_E = \mathbf{D}\hat{\mathbf{x}}$ by basis matching pursuit [35] such that

$$\min_{\widehat{\mathbf{x}}} \frac{1}{2} \|\mathbf{x}_E - \mathbf{D}\hat{\mathbf{x}}\|_2^2 + \lambda \|\hat{\mathbf{x}}\|_1 \tag{8}$$

where λ is a slack variable and $\|\cdot\|_1$ is the sparsity including ℓ_1 norm. The slack variable balances the trade-off between fitting data perfectly and employing a

sparse solution. For further improvements, we represent each color channel, individually. Also, we reshape the sparsely represented vector $\hat{\mathbf{x}}$ into a 3D structure of $\hat{\mathbf{X}} = \left\{\hat{\mathbf{x}}^{(l)}\right\}_{l=1}^{L}$, where L is the length of the structure. We feed the proposed deep model with Sparse Cubic Symmetrical Patterns (SCSP) instead of raw videos.

For notational simplicity, we will consider the sparsely represented $\hat{\mathbf{X}} = \left\{\hat{\mathbf{x}}^{(l)}\right\}_{l=1}^{L}$ as a sequence of frames with length L and denote it by $\mathbf{X} = \left\{\mathbf{x}^{(l)}\right\}_{l=1}^{L}$ hereafter.

4.2 Learning DDMs of the Training Classes

In order to initialize the parameters of the HDDM using GRBMs, we randomly shuffle a small subset, containing video sequences from all classes (of the training video sequences). We use this subset for layer-wise GRBM training of all encoder layers. The parameters of the decoder layers are then configured with their corresponding tied parameters of the encoder layers. This process assures us that rarely does the proposed network gets stuck in a local minimum.

We define a cost function based on the representation error over all frames of the video for learning class-specific parameters. In order to avoid over-fitting and enhance the generalization of the learnt model to unseen test data, the regularization terms are added to the cost function of the HDDM. A weight decay penalty term J_{wd} and a sparsity constraint J_{sp} are added.

$$J_{reg}\left(\theta_{HDDM}\left|x^{(l)} \in X_c\right.\right) = \sum \left\|x^{(l)} - \tilde{x}^{(l)}\right\|^2 + \lambda_{wd} J_{wd} + \lambda_{sp} J_{sp} \tag{9}$$

where λ_{wd} and λ_{sp} are regularization parameters. J_{wd} guarantees small values of weights for all hidden units and ensures that dropping out will not happen for hidden layers' units. It is defined as the summation of the Frobenius norm of all weight matrices:

$$J_{wd} = \sum_{i=1}^{M} \left\|\mathbf{W}_e^{(i)}\right\|_F^2 + \sum_{i=1}^{M} \left\|\mathbf{W}_d^{(i)}\right\|_F^2 \tag{10}$$

Moreover, J_{sp} enforces that the mean activation $\bar{\rho}_j^{(i)}$ (over all training samples) of the j-th unit of the i-th hidden layer is as close as possible to the sparsity target ρ which is a very small constant. J_{sp} is further defined based on the KL divergence as

$$J_{sp} = \sum_{i=1}^{2M-1} \sum_{j} \rho \log \frac{\rho}{\bar{\rho}_j^{(i)}} + (1-\rho) \log \frac{1-\rho}{1-\bar{\rho}_j^{(i)}} \tag{11}$$

Therefore, a class specific model $\boldsymbol{\theta}_c$ is obtained by optimizing the regularized cost function J_{reg} over all frames of the class $\mathbf{X}_c$.

$$\boldsymbol{\theta}_c = \arg \min_{\boldsymbol{\theta}_{HDDM}} J_{reg}\left(\boldsymbol{\theta}_{HDDM}\left|\mathbf{x}^{(l)} \in \mathbf{X}_C\right.\right) \tag{12}$$

We note that our proposed model is easily scalable. Enrolling new classes would not require re-training on the complete database. Instead, the class-specific models for the added classes can be learnt independently of the existing classes.

4.3 Classification

Given a query video sequence $\mathbf{X}_q = \left\{\mathbf{x}^{(t)}\right\}_{t=1}^{T_q}$, we first extract SCSPs and then separately reconstruct them using all class-specific DDMs $\boldsymbol{\theta}_c$, $c = 1, \cdots, C$, using Eqs. (1) and (2). Suppose $\tilde{\mathbf{x}}_c^{(l)}$ is the l-th frame of the reconstructed query video sequence $\tilde{\mathbf{X}}_{q_c}$ based on the c-th class model $\boldsymbol{\theta}_c$. We obtain the reconstruction errors, i.e. $\left\|x^{(l)} - \tilde{x}_c^{(l)}\right\|_2$, from all class specific models; then, a weighted voting strategy is employed to determine the class label of the given query video sequence $\mathbf{X}_q$. Each query video sequence's frame $\mathbf{x}^{(l)}$ casts a vote to all classes. Using the reconstruction error of each class's model, we assign a weight to the casted vote to each class. The weight $\mu_c\left(x^{(l)}\right)$ of the vote casted by a frame $\mathbf{x}^{(l)}$ to class c is defined as

$$\mu_c\left(\mathbf{x}^{(l)}\right) = \exp\left(-\left\|\mathbf{x}^{(l)} - \tilde{\mathbf{x}}_c^{(l)}\right\|_2\right) \tag{13}$$

The candidate class which achieves the highest accumulated weight from all frames of $\mathbf{X}_q$ is declared as the class y_q of the query video sequence:

$$y_q = \arg\max_c \sum_{\mathbf{X}_q} \mu_c\left(\mathbf{x}^{(l)}\right) \tag{14}$$

5 Experimental Analysis

We extensively evaluate the performance of the proposed method on five benchmarking datasets including UCF101 [36] and Hollywood II [37] datasets for action recognition, DynTex dataset [17] for dynamic texture recognition, and YUPENN [38] and Maryland [39] datasets for dynamic scene classification task.

5.1 Parameter Setting

We performed a grid search to obtain the optimal parameters and conducted experiments on a validation set. To be specific, the initial weights for layer-wise GRBM training are drawn from a uniform random distribution in the range of $[-0.005, 0.005]$. Contrastive Divergence (CD) was used to train GRBMs on 200 randomly selected videos from the training data. Mini-batches of 32 videos were used and the training was done for 50 epochs. A fixed learning rate of 10^{-3} was used. To train the pre-initialized HDDM to learn class-specific models, we used an annealed learning rate that is started with 2×10^{-3} and multiplied by a factor of 0.6 per epoch. We chose ℓ_2 weight decay (λ_{wd}) to be 0.01, a sparsity target (ρ) of 0.001, and non-sparsity penalty term (λ_{sp}) of 0.5. The training was performed by considering a mini-batch of 10 videos for 20 epochs.

Table 1. Comparison of the proposed method's accuracy (%) on the UCF101 database [36] with different block sizes for SCSP.

Block size	$1 \times 1 \times 3$	$1 \times 1 \times 5$	$3 \times 3 \times 3$	$3 \times 3 \times 5$	$5 \times 5 \times 3$	$5 \times 5 \times 5$	$7 \times 7 \times 5$
Accuracy	87.3	91.2	94.3	92.5	89.4	84.3	79.5

The size of volumetric blocks in SCSP also affects the performance of the algorithm. Therefore, we conducted an empirical study on different sizes of video blocks in Table 1. It is observed from Table 1 that the best result is achieved with the block size of $3 \times 3 \times 3$. With very small blocks (e.g. $1 \times 1 \times 3$), few spatiotemporal regions are captured and the model will have problems on dealing with the scene variations. Moreover, the blocks of large sizes (e.g. $7 \times 7 \times 5$) carry too much information that does not improve the model's performance.

In order to determine the number of layers and the number of units in each layer, we employed a multi-resolution search strategy. The ideas is to test some values from a large parameter range, choose a few best configurations, and then test again with smaller steps around these values. We tested the model with the escalating number of layers [40] and stopped where the performance reaches the highest rate on the validation set. The hidden layers sizes varies in the range of [250, 1000].

5.2 Human Action Recognition

We conducted experiments on two benchmark action recognition datasets, i.e. UCF101 [36] and Hollywood II [37] datasets, and compared the performance of the proposed method against state-of-the-art approaches.

The UCF101 dataset [36] is composed of realistic web videos which are typically captured with large variations in camera motion, object appearance/scale, viewpoint, cluttered background, and illumination variations. It has 101 categories of human actions ranging from daily life to sports. The UCF101 contains 13,320 videos with an average length of 180 frames. It has three splits setting to separate the dataset into training and testing videos. We report the average classification accuracy over these three splits.

We compare the average accuracy performance of our proposed DDM with both the traditional and deep learning-based benchmark methods for human action recognition in Table 2. Our model obtains an average accuracy of 91.5%. However, the accuracy of DDM on the UCF101 is less than that of KVMF [41] by 1.6%. We argue that the performance of DDM degrades since it only captures short range spatiotemporal information in the video sequence. The videos in UCF101 exhibit significant temporal variations. Moreover, the severe camera movements increase the complexity of video's dynamics and make data reconstruction challenging. These issues bring up difficulties for the algorithm to focus on the action happening at each time instance.

To tackle this problem, we feed the extracted SCSP features to our DDM. The proposed SCSP extracts detailed spatiotemporal information by capturing the

Table 2. Comparison of the average classification accuracy of DDM against state-of-the-art methods on the UCF101 dataset [36].

Method	Average accuracy (%)
iDT+HSV [42]	87.9
MoFAP [43]	88.3
Two-Stream CNN [12]	88.0
C3D (3 nets) [13]	85.2
C3D (3 nets)+iDT [13]	90.4
F_{ST}CN (SCI Fusion) [14]	88.1
TDD+FV [44]	90.3
KVMF [41]	93.1
DDM	**91.5**
DDM+SCSP	**94.3**

spatiotemporal variations of the video sequence within small volumetric blocks. By representing this information sparsely, it not only covers the whole length of the video sequence, but also decreases the redundancy of data. In this way, SCSP increases the discriminability of samples in the feature space in which the similar samples are mapped close to each other and dissimilar ones are mapped far apart. Therefore, the DDM can readily learn the underlying structure of each class. As can be seen from Table 2, the performance of our DDM improves by using SCSP features.

The Hollywood II dataset [37] has been constructed from 69 different Hollywood movies and includes 12 activity classes. It contains a total of 1,707 videos with 823 training videos and 884 testing videos. The length of videos varies from hundreds to several thousand frames. According to the test protocol, the performance is measured by the mean average precession over all classes [37].

To compare our approach with the benchmark, we obtain the average precession performance for each class and take the mean average precession (mAP) as indicated in Table 3. The best result is obtained using DDM with a 0.4 mAP improvement in the overall accuracy. The superior performance of the proposed method in action recognition task demonstrates the effectiveness of our long-term spatiotemporal modeling of videos approach.

5.3 Dynamic Texture and Dynamic Scene Recognition

We evaluated the capability of our proposed method in the case of dynamic texture and dynamic scene classification using DynTex [17] dataset, and YUPENN [38] and Maryland [39] datasets, respectively. In order to follow the standard comparison protocol, we use Leave-One-Out (LOO) cross validation. Note that the results are drawn from the related papers.

Table 3. Comparison of the mean average precession (mAP) of DDM with the state-of-the-art methods on the Hollywood II dataset [37].

Method	mAP (%)
DL-SFA [45]	48.1
iDT [46]	64.3
Actons [47]	64.3
MIFS [48]	68.0
NL-RFDRP+CNN [49]	70.1
HRP [14]	76.7
DDM	**75.3**
DDM+SCSP	**77.1**

The DynTex dataset [17] is a standard database for dynamic texture analysis containing high-quality dynamic texture videos such as windmill, waterfall, and sea waves. It includes over 650 videos recorded in PAL format in various conditions. Each video has 250 frames length with a 25 frames per second frame rate. Table 4 compares the rank-1 recognition rates of DDM with the benchmark approaches. It can be clearly seen that our proposed approach yields in the best results compared to all other methods.

The YUPENN dataset [38] is a stabilized dynamic scene dataset. This dataset was created with an emphasis on scene-specific temporal information. YUPENN contains 14 dynamic scene categories with 30 videos per category. There are significant variations in this dataset's video sequences such as frame rate, scene appearance, scaling, illumination, and camera viewpoint. We report the experimental results on this dataset in Table 5. It can be observed from Table 5 that DDM outperforms the existing state-of-the-art methods in the case of dynamic scene classification. The results confirm that the proposed DDM is effective for dynamic scene data classification in a stabilized setting.

Table 4. Comparison of the rank-1 recognition rates on the DynTex dataset [17].

Method	Recognition rate (%)
VLBP [19]	95.71
LBP-TOP [19]	97.14
DFS [50]	97.63
BoS Tree [51]	98.86
MBSIF-TOP [52]	98.61
st-TCoF [9]	98.20
DDM	**98.05**
DDM+SCSP	**99.27**

Table 5. Comparison of the classification results (%) on the YUPENN [38] and Maryland [39] dynamic scene datasets.

Method	YUPENN	Maryland
CSO [53]	85.95	67.69
SFA [54]	85.48	60.00
SOE [3]	80.71	43.10
BoSE [3]	96.19	77.69
LBP-TOP [19]	84.29	39.23
C3D [13]	98.10	N/A
st-TCoF [9]	99.05	88.46
DDM	**97.52**	**86.33**
DDM+SCSP	**99.18**	**90.27**

The Maryland dataset [39] is a dynamic scene database which consist of 13 natural scene categories containing 10 videos each with 617 frames on average. The dataset has videos showing a wide range of variations in natural dynamic scenes, e.g. avalanches, traffic, and forest fire. One notable difference between the Maryland dataset and the YUPENN dataset is that the former includes camera motions, while the latter does not.

We present the comparison between our proposed method and the state-of-the-art methods in Table 5. Since most of the videos in the Maryland dataset show significant temporal variations, the experimental results suggest that, for highly dynamic data, DDM is able to outperform its strongest rival st-TCoF by a margin of 1.81%. The promising performance of st-TCoF in the dynamic scene classification (Table 5) is due to incorporating the spatial and the temporal information of the video sequence. However, the results on Maryland dataset suggests that st-TCoF is sensitive to the significant camera motions. On the other hand, our DDM is strongly robust when the structure of the images drastically changes their position with time. Therefore, the DDM can effectively learn the complex underlying structure of the dynamic scene in the presence of severe camera movements.

5.4 Discriminability Analysis

In order to illustrate the disciminability power of the SCSP, Fig. 4 shows the distribution of the sampled data from different classes of UCF101 database before and after applying SCSP in a 3D space. Thanks to the existing redundancy, the samples are correlated before applying SCSP, which makes the data reconstruction a non-trivial task. However, the samples become scattered in the feature space after applying SCSP, i.e. the similar samples are closer to each other and dissimilar samples are far apart. This strategy makes the process of learning class-specific models easier for DDM by learning the underlying structure of each class from the feature space instead of raw video data.

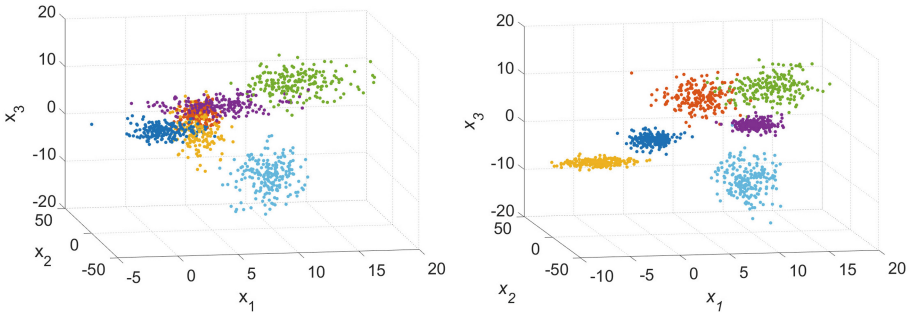

Fig. 4. An example of the distribution of the learnt classes from UCF101 dataset [36] before (**Left**) and after (**Right**) applying the proposed SCSP. The SCSP reduces the correlation between similar classes by condensing and scattering their samples in the feature space.

By enlarging the inter-class similarity of data, the proposed DDM reconstructs the videos of each class more effectively by learning the class-specif models. According to the distances between samples in the new feature space, the DDM can easily learn the pattern and the structure of each class, since the correlation and the redundancy are lessened by applying SCSP.

6 Conclusion

We proposed a novel deep learning approach for video-based scene classification. Specifically, a multi-layer deep autoencoder structure was presented which is first pre-trained for appropriate parameter initialization and then fine-tuned for learning class-specific Deep Discriminative Models (DDMs). Capturing the underlying non-linear complex geometric surfaces, the DDMs can effectively model the spatiotemporal variations within video sequences. In order to discard the redundant information in video sequences and avoid the strong correlations between adjacent frames, we captured the spatiotemporal variations in the video sequences and represented them sparsely using Sparse Cubic Symmetrical Pattern (SCSP). The learnt DDMs are used for a minimum reconstruction error-based classification technique during the testing phase. The proposed method has been extensively evaluated on a number of benchmark video datasets for action recognition, dynamic texture and dynamic scene classification tasks. Comparisons against state-of-the-art approaches showed that our proposed method achieves very interesting performance.

Acknowledgement. The financial support of the Academy of Finland and Infotech Oulu is acknowledged.

References

1. Ravichandran, A., Chaudhry, R., Vidal, R.: Categorizing dynamic textures using a bag of dynamical systems. IEEE Trans. PAMI **35**(2), 342–353 (2013)
2. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. PAMI **24**(7), 971–987 (2002)
3. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Bags of spacetime energies for dynamic scene recognition. In: IEEE CVPR, pp. 2681–2688 (2014)
4. Kim, T.K., Kittler, J., Cipolla, R.: Discriminative learning and recognition of image set classes using canonical correlations. IEEE Trans. PAMI **29**(6), 1005–1018 (2007)
5. Wang, R., Shan, S., Chen, X., Dai, Q., Gao, W.: Manifold-manifold distance and its application to face recognition with image sets. IEEE Trans. Image Process. **21**(10), 4466–4479 (2012)
6. Harandi, M., Sanderson, C., Shirazi, S., Lovell, B.C.: Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In: IEEE CVPR, pp. 2705–2712 (2011)
7. Wang, R., Guo, H., Davis, L.S., Dai, Q.: Covariance discriminative learning: a natural and efficient approach to image set classification. In: IEEE CVPR, pp. 2496–2503 (2012)
8. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. PAMI **35**(8), 1798–1828 (2013)
9. Qi, X., Li, C.G., Zhao, G., Hong, X., Pietikäinen, M.: Dynamic texture and scene classification by transferring deep image features. Neurocomputing **171**, 1230–1241 (2016)
10. Hayat, M., Bennamoun, M., An, S.: Deep reconstruction models for image set classification. IEEE Trans. PAMI **37**(7), 713–727 (2015)
11. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: IEEE CVPR, pp. 1725–1732 (2014)
12. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS, pp. 568–576 (2014)
13. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: IEEE ICCV, pp. 4489–4497 (2015)
14. Sun, L., Jia, K., Yeung, D.Y., Shi, B.E.: Human action recognition using factorized spatio-temporal convolutional networks. In: IEEE ICCV, pp. 4597–4605 (2015)
15. Welling, M., Rosen-Zvi, M., Hinton, G.: Exponential family harmoniums with an application to information retrieval. In: NIPS, pp. 1481–1488 (2004)
16. Chaudhry, R., Hager, G., Vidal, R.: Dynamic template tracking and recognition. IJCV **105**(1), 19–48 (2013)
17. Péteri, R., Fazekas, S., Huiskes, M.J.: DynTex: a comprehensive database of dynamic textures. Patt. Recogn. Lett. **31**(12), 1627–1632 (2010)
18. Yuan, L., Wen, F., Liu, C., Shum, H.-Y.: Synthesizing dynamic texture with closed-loop linear dynamic system. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3022, pp. 603–616. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24671-8_48
19. Zhao, G., Pietikäinen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. PAMI **29**(6), 915–928 (2007)

20. Rahtu, E., Heikkilä, J., Ojansivu, V., Ahonen, T.: Local phase quantization for blur-insensitive image analysis. Image Vis. Comput. **30**(8), 501–512 (2012)
21. Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: From generic to specific deep representations for visual recognition. In: IEEE CVPR, pp. 36–45 (2015)
22. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: OverFeat: integrated recognition, localization and detection using convolutional networks. CoRR abs/1312.6229 (2013)
23. Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: IEEE CVPR, pp. 1891–1898 (2014)
24. Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**(1), 1–127 (2013)
25. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
26. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Advances in Neural Information Processing Systems, pp. 350–358 (2012)
27. Smolensky, P.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1. MIT Press, Cambridge (1986). 194–281
28. Taylor, G.W., Hinton, G.E., Roweis, S.: Modeling human motion using binary latent variables. In: Advances in Neural Information Processing Systems, pp. 1345–1352 (2007)
29. Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatio-temporal features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6316, pp. 140–153. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15567-3_11
30. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. IEEE Trans. PAMI **35**(1), 221–231 (2013)
31. Ng, J.Y.H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: deep networks for video classification. In: IEEE CVPR, pp. 4694–4702 (2015)
32. Varol, G., Laptev, I., Schmid, C.: Long-term temporal convolutions for action recognition. CoRR abs/1604.04494 (2016)
33. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
34. Hinton, G.E., Osindero, S., Welling, M., Teh, Y.W.: Unsupervised discovery of nonlinear structure using contrastive backpropagation. Cogn. Sci. **30**(4), 725–731 (2006)
35. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Rev. **43**(1), 129–159 (2001)
36. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR abs/1212.0402 (2012)
37. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: IEEE CVPR, pp. 2929–2936 (2009)
38. Derpanis, K.G., Lecce, M., Daniilidis, K., Wildes, R.P.: Dynamic scene understanding: the role of orientation features in space and time in scene classification. In: IEEE CVPR, pp. 1306–1313 (2012)
39. Shroff, N., Turaga, P., Chellappa, R.: Moving vistas: exploiting motion for describing scenes. In: IEEE CVPR, pp. 1911–1918 (2010)

40. Larochelle, H., Erhan, D., Courville, A., Bergstra, J., Bengio, Y.: An empirical evaluation of deep architectures on problems with many factors of variation. In: ICML, pp. 473–480 (2007)
41. Zhu, W., Hu, J., Sun, G., Cao, X., Qiao, Y.: A key volume mining deep framework for action recognition. In: IEEE CVPR, 1991–1999 (2016)
42. Peng, X., Wang, L., Wang, X., Qiao, Y.: Bag of visual words and fusion methods for action recognition: comprehensive study and good practice. Comput. Vis. Image Underst. **150**, 109–125 (2016)
43. Wang, L., Qiao, Y., Tang, X.: MoFAP: a multi-level representation for action recognition. IJCV **119**(3), 254–271 (2016)
44. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: IEEE CVPR, pp. 4305–4314 (2015)
45. Sun, L., Jia, K., Chan, T.H., Fang, Y., Wang, G., Yan, S.: DL-SFA: deeply-learned slow feature analysis for action recognition. In: IEEE CVPR, pp. 2625–2632 (2014)
46. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: IEEE ICCV, pp. 3551–3558 (2013)
47. Zhu, J., Wang, B., Yang, X., Zhang, W., Tu, Z.: Action recognition with actons. In: IEEE ICCV, pp. 3559–3566 (2013)
48. Zhenzhong, L., Ming, L., Xuanchong, L., Hauptmann, A.G., Raj, B.: Beyond Gaussian pyramid: multi-skip feature stacking for action recognition. In: IEEE CVPR, pp. 204–212 (2015)
49. Fernando, B., Gavves, E., Oramas, J., Ghodrati, A., Tuytelaars, T.: Rank pooling for action recognition. IEEE Trans. PAMI **39**(4), 773–787 (2017)
50. Yong, X., Yuhui, Q., Haibin, L., Hui, J.: Dynamic texture classification using dynamic fractal analysis. In: IEEE ICCV, pp. 1219–1226 (2011)
51. Coviello, E., Mumtaz, A., Chan, A.B., Lanckriet, G.R.G.: Growing a bag of systems tree for fast and accurate classification. In: IEEE CVPR, pp. 1979–1986 (2012)
52. Arashloo, S.R., Kittler, J.: Dynamic texture recognition using multiscale binarized statistical image features. IEEE Trans. Multimedia **16**(8), 2099–2109 (2014)
53. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Spacetime forests with complementary features for dynamic scene recognition. In: BMVC, pp. 1–12 (2013)
54. Thériault, C., Thome, N., Cord, M.: Dynamic scene classification: learning motion descriptors with slow features analysis. In: IEEE CVPR, pp. 603–2610 (2013)