Video Classification Using Deep Autoencoder Network

Farshid Hajati[1,2] and Mohammad Tavakolian[3]
[1] School of Information Technology and Engineering, MIT Sydney, Sydney, Australia
[2] College of Engineering and Science,Victoria University Sydney, Sydney, Australia
Email: fhajati@mit.edu.au, farshid.hajati@vu.edu.au
[3] Center for Machine Vision and Signal Analysis (CMVS), University of Oulu, Oulu, Finland
Email: mohammad.tavakolian@oulu.fi

Abstract. We present a deep learning framework for video classification applicable to face recognition and dynamic texture recognition. A Deep Autoencoder Network Template (DANT) is designed whose weights are initialized by conducting unsupervised pre-training in a layer-wise fashion using Gaussian Restricted Boltzmann Machines. In order to obtain a class specific network and fine tune the weights for each class, the pre-initialized DANT is trained for each class of video sequences, separately. A majority voting technique based on the reconstruction error is employed for the classification task. The extensive evaluation and comparisons with state-of-the-art approaches on Honda/UCSD, DynTex, and YUPPEN databases demonstrate that the proposed method significantly improves the performance of dynamic texture classification.

1 Introduction

Classification and recognition tasks in different applications have been one of the interesting topics in recent years [1-8]. Videos contain dynamic textures which can be described as a visual process including a group of elements with random motions. Video dynamics 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 video dynamics is challenging but very important for subsequent vision tasks such as video classification, dynamic texture synthesis, motion segmentation, and so on.

Despite all challenges, great efforts have been devoted to find a robust and powerful solution for video-based recognition. Furthermore, it is commonly substantiated that effective representation of the video content is a crucial step towards resolving the problem of dynamic texture classification. In the past decade, a large number of approaches for video representation have been proposed, e.g. Linear Dynamic System (LDS) based methods [9], Local Binary Pattern (LBP) based methods [10], and Wavelet based methods [11]. Unfortunately, the current methods are sensitive to undesirable external phenomena. Coupled with these drawbacks, other

methods frequently model the video information within consecutive frames on a geometric surface so that it is represented by a subspace [12], a combination of subspaces [13], a point on the Grassmann manifold [14], or Lie Group of Riemannian manifold [15]. This requires prior assumptions regarding specific category of the geometric surface on which the samples of the video are believed to lie on.

A growing body of literature has investigated numerous approaches for video classification [11, 16]. Among them, Local Binary Pattern (LBP) based methods [10] have been widely used in texture analysis. Zhao et al. [17] extended the LBP to both space and time domains and proposed two new variants. Volume Local Binary Pattern (VLBP) [17] is an extension which combines both spatial and temporal variations of the video. Also, Local Binary Pattern on Three Orthogonal Planes (LBP-TOP) [17] computes LBP of three individual $x-y$, $x-t$, and $y-t$ planes to describe the video.

Recently, there is a huge growing research interest in deep learning methods in different areas of computer vision [18, 19]. Deep learning methods set up numerous recognition records in image classification [20], object detection [21], face recognition and verification [22]. Deep models have much more expressive power than traditional shallow models and can be effectively trained with layer-wise pre-training and fine-tuning [18]. Xie et al. [23] represented relationship between noisy and clean images using stacked denoising autoencoders. However, deep autoencoders are rarely used to model time series data. Despite this, there have been some works on using variants of Restricted Boltzmann Machine (RBM) [24] for specific time series data such as human motion [25]. Some other deep models have been put forward to address video data with convolutional learning of spatiotemporal features [26]. Needless to say that learning deep frameworks require huge amount of training data and is quite costly in computational demand. As a result, the deprivation of training data is indeed an obstacle to deploy a deep model for video classification tasks.

This paper presents a novel deep learning framework which makes no prior assumptions with respect to the underlying geometry and explore automatically the structure of the complex non-linear surface on which samples of video are present. The proposed method (see the block diagram in Figure 1) first defines a Deep Autoencoder Network Template (DANT) whose weights are initialized with an unsupervised layer-wise pre-training using Gaussian Restricted Boltzmann Machines (GRBM). In order to learn class specific Deep Autoencoder Network (DAN), the initialized DANT is then separately trained for each class using all videos of that class. Therefore, DANs can represent videos of each class based on the learnt structure of the corresponding class. A query video is represented using the learnt class specific DANs for classification purpose. The representation errors from the respective DANs are then computed and a voting technique is used to decide which class the query video shall belong to.

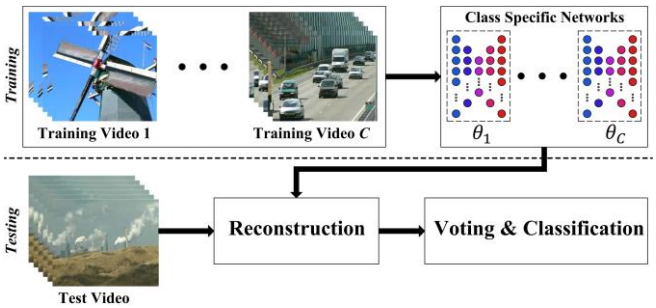

Figure 1: The block diagram of the proposed method.

2 Deep Autoencoder Network

We first define a Deep Autoencoder Network Template (DANT) which will be used to learn the underlying structure of the data. The architecture of the DANT is shown in Figure 2. For such a deep network, an appropriate parameter initialization is mandatory to achieve a good performance. Therefore, we initialize the parameters of DANT by performing a pre-training in a greedy layer-wise fashion using Gaussian Restricted Boltzmann Machines. The DANT with initialized parameters is then separately fine-tuned for each of the C classes of the training videos. We therefore end up with a total of C fine-tuned class-specific Deep Autoencoder Networks (DANs). The fined-tuned models are then used for video classification.

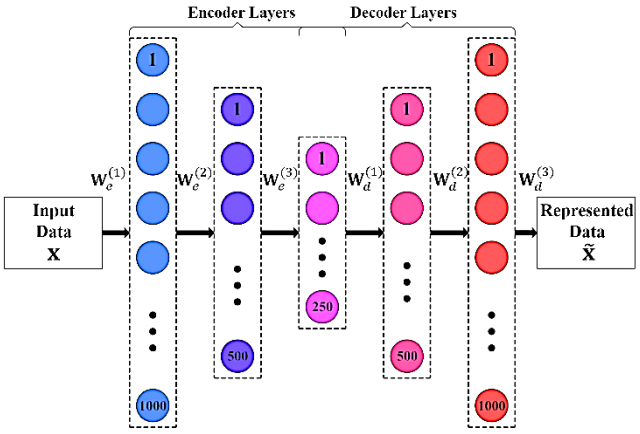

Figure 2: The structure of the proposed Deep Autoencoder Network (DAN).

2.1. The Deep Autoencoder Network Template

As can be seen in Figure 2, the proposed DANT is based on an autoencoder which comprises an encoder and a decoder. Both the encoder and the decoder have three hidden layers, with a shared third layer. The encoder calculates a compact low dimensional representation of the input data. We can formulate the encoder as a combination of non-linear functions $s(\cdot)$ to map the input data $\mathbf{x}$ to a representation $\mathbf{h}$ given by

$$\begin{aligned} \mathbf{h}_1 &= s\left(\mathbf{W}_e^{(1)}\mathbf{x} + \mathbf{b}_e^{(1)}\right) \\ \mathbf{h}_2 &= s\left(\mathbf{W}_e^{(2)}\mathbf{h}_1 + \mathbf{b}_e^{(2)}\right) \\ \mathbf{h} &= s\left(\mathbf{W}_e^{(3)}\mathbf{h}_2 + \mathbf{b}_e^{(3)}\right) \end{aligned} \tag{1}$$

where $\mathbf{W}_e^{(i)} \in \mathbb{R}^{n_i \times n_{i-1}}$ 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 training the encoder-decoder structure to represent the input data by optimizing a cost function. So, the decoder can be defined as a 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

$$\begin{aligned} \mathbf{x}_1 &= s\left(\mathbf{W}_d^{(1)}\mathbf{h} + \mathbf{b}_d^{(1)}\right) \\ \mathbf{x}_2 &= s\left(\mathbf{W}_d^{(2)}\mathbf{x}_1 + \mathbf{b}_d^{(2)}\right) \\ \tilde{\mathbf{x}} &= s\left(\mathbf{W}_d^{(3)}\mathbf{x}_2 + \mathbf{b}_d^{(3)}\right) \end{aligned} \tag{2}$$

We represent the complete encoder-decoder structure by its parameters $\theta_{DANT} = \{\theta_{\mathbf{W}}, \theta_{\mathbf{b}}\}$, where $\theta_{\mathbf{W}} = \left\{\mathbf{W}_e^{(i)}, \mathbf{W}_d^{(i)}\right\}_{i=1}^{3}$ and $\theta_{\mathbf{b}} = \left\{\mathbf{b}_e^{(i)}, \mathbf{b}_d^{(i)}\right\}_{i=1}^{3}$.

2.2. DANT Parameter Initialization

The above defined DANT is used to learn class specific networks. This is done by individual training of the DANT for videos of each class in the training set. The training is performed through stochastic gradient descent with back propagation [27]. The training may fail if the DANT is initialized by inappropriate weights. Thus, the parameters of the template are firstly set up by performing unsupervised pre-training. For this purpose, a greedy layer-wise approach is adopted and Gaussian RBMs are used.

RBMs [24] are generative undirected graphical models with a bipartite structure of two sets of binary stochastic nodes called the visible ($\{v_i\}_{i=1}^{N_v}, v_i \in \{0,1\}$) and the hidden layer nodes ($\{h_j\}_{j=1}^{N_h}, h_j \in \{0,1\}$). The visible layer's nodes are symmetrically connected with the nodes of the hidden layer through a weight matrix $\mathbf{W} \in \mathbb{R}^{N_h \times N_v}$, but there are no intra-layer node connections. The joint probability $p(\mathbf{v}, \mathbf{h})$ of the RBM is given by

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z}\exp\left(-E(\mathbf{v}, \mathbf{h})\right) \tag{3}$$

where Z is the partition function used as a normalization constant. $E(\mathbf{v}, \mathbf{h})$ is the energy function of the model defined as

$$E(\mathbf{v}, \mathbf{h}) = \sum_i b_i v_i - \sum_j c_j h_j - \sum_{ij} w_{ij} v_i h_j \tag{4}$$

where $\mathbf{b}$ and $\mathbf{c}$ are the biases of the visible and the hidden layers, respectively. In order to learn the model parameter $\{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ of the RBM, the training is performed by the numerical technique of Contrastive Divergence (CD) [28].

We can extend the standard RBM, which is used for binary stochastic data, to the real value data by appropriate modifications in its energy function. Gaussian RBM (GRBM) is one of such popular extensions whose energy function is defined by changing the bias term of the visible layer.

$$E_{GRBM}(\mathbf{v}, \mathbf{h}) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j c_j h_j - \sum_{ij} w_{ij} \frac{v_i}{\sigma_i} h_j \quad (5)$$

where 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 become staggering when using CD for GRBM parameter learning. Alternatively, we use another approach and fix σ_i to a constant value in the data pre-processing phase.

Inasmuch as there are no intra-layer node connections, inference becomes readily tractable for the RBM to the contrary of most directed graphical models. The probability distributions for GRBM are given by

$$p(h_j|v) = s\left(\sum_i w_{ij} v_i + c_j\right) \quad (6)$$

$$p(v_i|h) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(\frac{-(v_i - u_i)^2}{2\sigma_i^2}\right) \quad (7)$$

where

$$u_i = b_i + \sigma_i^2 \sum_j w_{ij} h_j \quad (8)$$

Since our data are represented by real values, we use GRBMs to initialize the parameters of the proposed DANT. Two layers are considered at a time and the GRBM parameters are leant. At first, we assume the nodes of the input layer as the visible units $\mathbf{v}$. Thus, the nodes of the first hidden layer are considered as the hidden unit $\mathbf{h}$ of the first GRBM and the parameters are tuned. The activations of the first GRBM's hidden units are then used as an input to train the second GRBM. The process is repeated for all three hidden layers of the encoder. The weights learnt for the encoder layers are then tied to the corresponding decoder layers.

3. Video Classification

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

3.1. Learning Class Specific Network

In order to initialize the parameters of the DANT using GRBMs, we randomly shuffle a small fraction of the training video sequences such that there are video sequences from all classes in this subset. We use this subset for layer-wise GRBM

training of all encoder's layers. The parameters of the decoder layers are then configured with their corresponding tied parameters of the encoder layers. This process assures that the proposed network rarely gets stuck in a local minimum point.

At this point, the DANT structure with the initialized weights is trained to learn class specific DANs. Here, the training of a DAN, θ_c, is carried out by minimization of the representation error over all frames $\mathbf{x}^{(t)}$ of the video $\mathbf{X}_c$.

$$J(\theta_{DANT}|\mathbf{x}^{(t)} \in \mathbf{X}_c) = \sum_t \left\|\mathbf{x}^{(t)} - \tilde{\mathbf{x}}^{(t)}\right\|^2 \tag{9}$$

where $\tilde{\mathbf{x}}^{(t)}$ is the t-th reconstructed frame of the video $\mathbf{X}_c$.

In order to avoid over-fitting and enhance generalization of the learnt model to unknown test data, the regularization terms are added to the cost function of DANT. A weight decay penalty term J_{wd} and a sparsity constraint J_{sp} are added.

$$J_{reg}(\theta_{DANT}; \mathbf{x}^{(t)} \in \mathbf{X}_c) = \sum_t \left\|\mathbf{x}^{(t)} - \tilde{\mathbf{x}}^{(t)}\right\|^2 + \lambda_{wd} J_{wd} + \lambda_{sp} J_{sp} \tag{10}$$

where λ_{wd} and λ_{sp} are regularization parameters. J_{wd} ensures small values of weights for all hidden units. It is defined as the summation of the Frobenius norm of all weight matrices.

$$J_{wd} = \sum_{i=1}^{3} \left\|\mathbf{W}_e^{(i)}\right\|_F^2 + \sum_{i=1}^{3} \left\|\mathbf{W}_d^{(i)}\right\|_F^2 \tag{11}$$

where $\mathbf{W}_e^{(i)}$ and $\mathbf{W}_d^{(i)}$ are the weight matrices of the i-th layer of the encoder and the decoder, respectively.

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 a spartisty target ρ which is a very small value constant. J_{sp} is further defined regarding the KL divergence.

$$J_{sp} = \sum_{i=1}^{5} \sum_j \rho \log \frac{\rho}{\bar{\rho}_j^{(i)}} + (1-\rho) \log \frac{1-\rho}{1-\bar{\rho}_j^{(i)}} \tag{12}$$

So, a class specific model θ_c is obtained by optimizing the regularized cost function J_{reg} over all frames of the class X_c. Since the sigmoid activation functions are non-linear and a number of layers are joined together, the autoencoder structure is capable of learning very intricate non-linear structures.

$$\theta_c = \arg\min_{\theta_{DANT}} J_{reg}(\theta_{DANT}|\mathbf{x}^{(t)} \in X_c) \tag{13}$$

3.2. Classification

Given a query video sequence $\mathbf{X}_q = \{\mathbf{x}^{(t)}\}_{t=1}^{T_q}$, we separately reconstruct it from all class specific DANs θ_c, $c = 1, \cdots, C$, using Equations (1) and (2). Suppose $\tilde{\mathbf{x}}_c^{(t)}$ is the t-th frame of the reconstructed query video sequence $\tilde{\mathbf{X}}_{q_c}$ based on the c-th class model θ_c. After computing the reconstruction errors for all C classes, the vote $v^{(t)}$ is assigned to the class whose network has reconstructed the frame $\mathbf{x}^{(t)}$ with the minimum reconstruction error.

$$v^{(t)} = \arg\min_c \left\|\mathbf{x}^{(t)} - \tilde{\mathbf{x}}_c^{(t)}\right\|_2 \tag{14}$$

The votes casted by all frames of $\mathbf{X}_q$ are then counted and the candidate class which achieves the maximum number of votes is declared as the class y_q of the query video sequence $\mathbf{X}_q$.

4. Experimental Results

The performance of the proposed method is evaluated on three databases for the tasks of video-based face recognition and dynamic texture classifications. For video-based face recognition, the performance evaluation is conducted on the Honda/UCSD [29] database. The DynTex [16] and the YUPPEN [30] databases are used for dynamic texture classification. For the Honda/UCSD the faces are detected, cropped, and resized using the same procedure as in [31].

4.1. Results on the Honda/UCSD Database

The Honda/UCSD database [29] contains 59 video sequences of 20 different subjects. The video sequences are recorded in an indoor environment for at least 15 seconds at 15 frame per second. Similar to [29], we use 20 video sequences for training and the remaining 39 for testing. We repeat our experiments ten times with different random selections of the training and testing sets.

The experimental results in terms of average recognition rates and the standard deviations of DAN and the state-of-the-art benchmark methods are summarized in Table 1. The results demonstrate that the proposed method achieves 100% classification on the Honda/UCSD dataset.

Table 1: Comparison of average recognition rates ± standard deviations (%) and equal error rates (%) on the Honad/UCSD dataset [29].

Method	Avg. Recognition Rate (%) ± Standard Deviation	EER (%)
VLBP [17]	77.94±1.00	23.15
VLBP+AdaBoost [32]	88.80±0.60	4.02
MMD [13]	95.55±1.84	3.51
MDA [33]	96.44±1.37	2.74
CDL [15]	98.97±1.32	0.63
SANP [31]	99.36±0.10	0.22
RNP [34]	95.90±2.16	3.08
DAN	**100.00±0.00**	**0.00**

4.2. Results on the DynTex Database

The DynTex database [16] is a standard database for dynamic texture analysis containing high-quality dynamic texture videos such as windmill, waterfall, smoke, etc. It contains over 650 videos recorded in PAL in different conditions. Each video has 250 frames length with the frame rate of 25 frame/sec. Table 2 compares the rank-1 recognition rates of the proposed DAN and the benchmark approaches. Following the standard protocol, we use Leave-One-Out (LOO) cross validation in the next experiments on the dynamic texture data.

Table 2: Comparison of the rank-1 recognition rate (%) of the proposed method with benchmark approaches on the DynTex database [16].

Method	Recognition Rate (%)
VLBP [17]	95.71

LBP-TOP [17]	97.14
DFS [35]	97.63
BoS Tree [36]	98.86
MBSIF-TOP [37]	98.61
st-TCoF [19]	98.20
DAN	**99.07**

4.3. Results on the YUPPEN Database

The YUPPEN database [30] is a stabilized dynamic scene dataset. This dataset was introduced to emphasize scene-specific temporal information. YUPPEN consists of 14 dynamic scene categories with 30 videos per category. The sequences in the YUPPEN dataset have significant variations, such as frame rate, scene appearance, scaling, illumination, and camera viewpoint. We present the experimental results on this database in Table 3. It can be observed that the DAN outperforms the state-of-the-art benchmark methods. The results confirm that the proposed DAN is effective for dynamic scene data in a stabilized setting.

Table 3: Comparison of the rank-1 recognition rate (%) of the proposed method to benchmark approaches on the YUPPEN database [30].

Method	**Recognition Rate (%)**
LBP-TOP [17]	84.29
BoSE [11]	96.19
SOE [11]	80.71
SFA [38]	85.48
CSO [39]	85.95
st-TCoF [19]	99.05
DAN	**99.16**

5. Conclusion

In this paper, we presented a novel deep learning framework for video classification. A multi-layer deep autoencoder network was designed which was first pre-trained for appropriate weight initialization and then used for learning class specific networks. The class specific network is capable to capture the underlying non-linear complex structure of videos. In order to classify a given query video, we adopt a voting strategy based on the minimum reconstruction error. The proposed Deep Autoencoder Network (DAN) are evaluated on three standard video datasets and achieved the best performance among the competing methods.

References

1. Hajati, F., M. Tavakolian, S. Gheisari, Y. Gao, and A.S. Mian, *Dynamic Texture Comparison Using Derivative Sparse Representation: Application to Video-Based Face Recognition.* IEEE Transactions on Human-Machine Systems, 2017. **47**(6): p. 970-982.

2. Hajati, F., K. Faez, and S.K. Pakazad. *An Efficient Method for Face Localization and Recognition in Color Images*. in *2006 IEEE International Conference on Systems, Man and Cybernetics*. 2006.
3. Hajati, F., A. Cheraghian, S. Gheisari, Y. Gao, and A.S. Mian, *Surface geodesic pattern for 3D deformable texture matching.* Pattern Recognition, 2017. **62**: p. 21-32.
4. R. Barzamini, F. Hajati, S. Gheisari, and M.B. Motamadinejad, *Short Term Load Forecasting using Multi-layer Perception and Fuzzy Inference Systems for Islamic Countries.* Journal of Applied Sciences, 2012. **12**(1): p. 40-47.
5. Shojaiee, F. and F. Hajati. *Local composition derivative pattern for palmprint recognition*. in *2014 22nd Iranian Conference on Electrical Engineering (ICEE)*. 2014.
6. Pakazad, S.K., K. Faez, and F. Hajati. *Face Detection Based on Central Geometrical Moments of Face Components*. in *2006 IEEE International Conference on Systems, Man and Cybernetics*. 2006.
7. Ayatollahi, F., A.A. Raie, and F. Hajati, *Expression-invariant face recognition using depth and intensity dual-tree complex wavelet transform features*. Vol. 24. 2015: SPIE. 1-13, 13.
8. Abdoli, S. and F. Hajati. *Offline signature verification using geodesic derivative pattern*. in *2014 22nd Iranian Conference on Electrical Engineering (ICEE)*. 2014.
9. Ravichandran, A., R. Chaudhry, and R. Vidal, *Categorizing Dynamic Textures Using a Bag of Dynamical Systems.* IEEE Trans. PAMI, 2013. **35**(2): p. 342-353.
10. Ojala, T., M. Pietikainen, and T. Maenpaa, *Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns.* IEEE Trans. PAMI, 2002. **24**(7): p. 971-987.
11. Feichtenhofer, C., A. Pinz, and R.P. Wildes, *Bags of Spacetime Energies for Dynamic Scene Recognition*, in *Proc. IEEE CVPR*. 2014. p. 2681-2688.
12. Kim, T.K., J. Kittler, and R. Cipolla, *Discriminative Learning and Recognition of Image Set Classes Using Canonical Correlations.* IEEE Trans. PAMI, 2007. **29**(6): p. 1005-1018.
13. Wang, R., S. Shan, X. Chen, Q. Dai, and W. Gao, *Manifold-Manifold Distance and its Application to Face Recognition with Image Sets.* IEEE Trans. Image Processing, 2012. **21**(10): p. 4466-4479.
14. Harandi, M., C. Sanderson, S. Shirazi, and B.C. Lovell, *Graph Embedding Discriminant Analysis on Grassmannian Manifolds for Improved Image Set Matching*, in *Proc. IEEE CVPR*. 2011. p. 2705-2712.
15. Wang, R., H. Guo, L.S. Davis, and Q. Dai, *Covariance Discriminative Learning: A Natural and Efficient Approach to Image Set Classification*, in *Proc. IEEE CVPR*. 2012. p. 2496-2503.
16. Péteri, R., S. Fazekas, and M.J. Huiskes, *DynTex: A Comprehensive Database of Dynamic Textures.* Pattern Recognition Letters, 2010. **31**(12): p. 1627-1632.
17. Zhao, G. and M. Pietikainen, *Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions.* IEEE Trans. PAMI, 2007. **29**(6): p. 915-928.
18. Bengio, Y., *Learning Deep Architectures for AI.* Foundations and Trends in Machine Learning, 2009. **2**(1): p. 1-127.
19. Qi, X., C.G. Li, G. Zhao, X. Hong, and M. Pietikäinen, *Dynamic Texture and Scene Classification by Transferring Deep Image Features.* Neurocomputing, 2016. **171**: p. 1230-1241.
20. Azizpour, H., A.S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, *From Generic to Specific Deep Representations for Visual Recognition*, in *Proc. IEEE CVPR*. 2015. p. 36-45.
21. Sermanet, P., D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, *Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks.* arXiv preprint arXiv:1312.6229, 2013.
22. Sun, Y., X. Wang, and X. Tang, *Deep Learning Face Representation from Predicting 10,000 Classes*, in *Proc. IEEE CVPR*. 2014. p. 1891-1898.
23. Xie, J., L. Xu, and E. Chen, *Image Denoising and Inpainting with Deep Neural Networks*, in *Advances in Neural Information Processing Systems*. 2012. p. 350-358.

24. Smolensky, P., *Information Processing in Dynamical Systems: Foundations of Harmony Theory*, in *Parallel Distributed Processing: Explorations in The Microstructure of Cognition*. 1986, MIT Press. p. 194-281.
25. Taylor, G.W., G.E. Hinton, and S. Roweis, *Modeling Human Motion Using Binary Latent Variables*, in *Advances in Neural Information Processing Systems*. 2007. p. 1345-1352.
26. Taylor, G.W., R. Fergus, Y. LeCun, and C. Bregler, *Convolutional Learning of Spatio-Temporal Features*, in *Proc. ECCV*. 2010. p. 140-153.
27. Hinton, G.E., S. Osindero, and Y.W. Teh, *A Fast Learning Algorithm for Deep Belief Nets.* Neural Computation, 2006. **18**(7): p. 1527-1554.
28. G. Hinton, S. Osindero, M. Welling, and Y.W. Teh, *Unsupervised Discovery of Nonlinear Structure Using Contrastive Backpropagation.* Cognitive Science, 2006. **30**(4): p. 725–731.
29. Lee, K.C., J. Ho, M.H. Yang, and D. Kriegman, *Video-Based Face Recognition Using Probabilistic Appearance Manifolds*, in *Proc. IEEE CVPR*. 2003. p. 313-320.
30. Derpanis, K.G., M. Lecce, K. Daniilidis, and R.P. Wildes, *Dynamic Scene Understanding: The Role of Orientation Features in Space and Time in Scene Classification*, in *Proc. IEEE CVPR*. 2012. p. 1306-1313.
31. Hu, Y., A.S. Mian, and R. Owens, *Face Recognition Using Sparse Approximated Nearest Points between Image Sets.* IEEE Trans. PAMI, 2012. **34**(10): p. 1992-2004.
32. Hadid, A. and M. Pietikainen, *Combining Appearance and Motion for Face and Gender Recognition from Videos.* Pattern Recognition, 2009. **42**(11): p. 2818-2827.
33. R. Wang and X. Chen, *Manifold Discriminant Analysis*, in *Proc. IEEE CVPR*. 2009. p. 429-436.
34. Yang, M., Z. Pengfei, L.V. Gool, and L. Zhang, *Face Recognition Based on Regularized Nearest Points between Image Sets*, in *Proc. IEEE Int. Conf. and Workshops on Automatic Face and Gesture Recognition (FG)*. 2013. p. 1-7.
35. Yong, X., Q. Yuhui, L. Haibin, and J. Hui, *Dynamic Texture Classification using Dynamic Fractal Analysis*, in *Proc. IEEE ICCV*. 2011. p. 1219-1226.
36. Coviello, E., A. Mumtaz, A.B. Chan, and G.R.G. Lanckriet, *Growing A Bag of Systems Tree for Fast and Accurate Classification*, in *Proc. IEEE CVPR*. 2012. p. 1979-1986.
37. Arashloo, S.R. and J. Kittler, *Dynamic Texture Recognition Using Multiscale Binarized Statistical Image Features.* IEEE Trans. Multimedia, 2014. **16**(8): p. 2099-2109.
38. Thériault, C., N. Thome, and M. Cord, *Dynamic Scene Classification: Learning Motion Descriptors with Slow Features Analysis*, in *Proc. IEEE CVPR*. 2013. p. 2603-2610.
39. C. Feichtenhofer, A. Pinz, and R.P. Wildes, *Spacetime Forests with Complementary Features for Dynamic Scene Recognition*, in *in Proc. BMVC*. 2013. p. 1-12.