Discriminative Feature Learning for Action Recognition Using a Stacked Denoising Autoencoder

Ruoxin Sang, Peiquan Jin, and Shouhong Wan

School of Computer Science and Technology,
Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences,
University of Science and Technology of China, Hefei, China
srx2007@mail.ustc.edu.cn, {jpq,wansh}@ustc.edu.cn

Abstract. In this paper, we propose a novel method to recognize human actions based on the depth information acquired by depth-based cameras. Representations of depth maps are learned and reconstructed using a stacked denoising autoencoder. By adding the category constraint, the learned features are more discriminative and able to capture the small but significant differences between actions. Greedy layer-wise training strategy is used to train the deep neural network. Then we use temporal pyramid matching on the feature representation to generate temporal representation. Finally a linear SVM is trained to classify each sequence into actions. Our method is evaluated on MSR Action3D dataset and show superiority over other popular methods. Experimental results also indicate the great power of our model to restore highly noisy input data.

Keywords: Action Recognition, Feature Learning, Stacked Denoising Autoencoders.

1 Introduction

Human action recognition has been an active field of research in computer vision. The goal of action recognition is to recognize people's behavior from videos in a given scenario automatically. It has many potential applications including content-based video search, human computer interaction, video surveillance, sports video [1, 2]. Most of these applications require high level understanding of spatial and temporal information from videos that are usually composed of multiple simple actions of persons.

Inferring high-level knowledge from a color video especially in a complex and unconstrained scene is very difficult and costful. However, the recent availability of depth cameras such as Kinect [3] has tremendously improved the abilities to understand human activities. Depth maps have several advantages over traditional intensity sensors. First, depth sensors can obtain the holistic 3D structure of the human body, which is invariant to color and texture. Second, color and texture methods perform worse in the dim lighter and the shadows may bring

J.-S. Pan et al. (eds.), *Intelligent Data Analysis and Its Applications, Volume 1*,
Advances in Intelligent Systems and Computing 297,
DOI: 10.1007/978-3-319-07776-5_54,

ambiguity. But the depth cameras can work in total darkness. Third, depth sensors greatly simplify the process of foreground extraction, removing plenty of noise and disturbance in the background [4,5].

Furthermore, the 3D skeleton joint positions can be estimated from the depth map accurately following the work of Shotton *et al.* [3]. The extracted skeleton joints have strong representation power, which is more discerning and compact than depth or color sequences. Although with these benefits, depth-based action recognition using joint features is still not an easy task [6]. Some of the estimated joints are not reliable when the human body is partly in view. The overlapping of human parts in some interactive actions can lead to the missing of some joint as well. Due to the noisy joint positions, extracting robust features from skeleton information is necessary.

Motivated by the satisfactory performance of previous work on exploring relative 3D joint features [2,7,8], we propose a novel method to learn robust and discriminative features from joint 3D features to recognize human actions. We build a deep neural network and employ denoising autoencoders, which has proved their strong abilities to reconstruct and denoise data, as the basic unit of our architecture. In order to seize very subtle spatio-temporal details between similar actions, we add the category constraint on denoising autoencoders to fuse intra-and inter-class information into features. We stack the denoising autoencoders with category constraint and greedy layer-wise training strategy is used to train the model. Then we use temporal pyramid matching on the feature representation to generate temporal representation. Finally a linear SVM is trained to classify each sequence into actions. Experiments show that this algorithm achieves superior results on a benchmark dataset.

The main contributions of this paper are three-fold. First, a new discriminative feature learning algorithm is proposed to recognize depth-based videos. Second, a novel category constraint is added into denoising autoencoders to preserve intra-and inter class information. Third, our extensive experiments show that our model has a strong capacity to reconstruct and denoise corrupted data.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes the entire flow of our methodology to recognize actions. Section 4 discusses the experimental results. Section 8 concludes the paper.

2 Related Work

Recently, low-level hand-crafted features have been designed to recognize human actions. Spatio-temporal salient points like STIP [9] or some local features, like Cuboids [10] and HOG\HOF [11] have been widely used. However, directly employ these original methods for color sequences on depth data is infeasible. Therefore, recent methods for action recognition in depth sequences explore alternative features particularly for depth-based videos. Li *et al.* [12] projected the depth map into three orthogonal planes and sampled representative 3D points to obtain a bag of 3D points. An action graph was deployed to model the dynamics

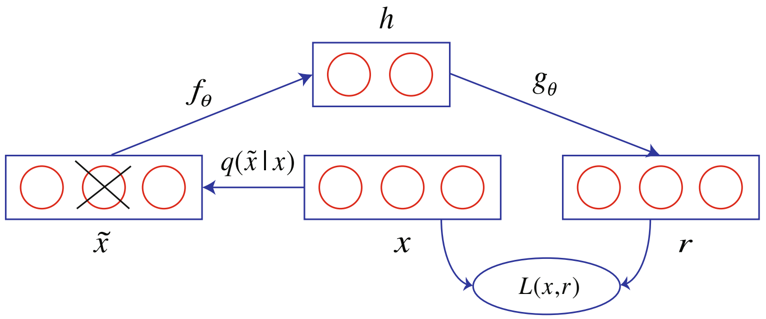

Fig. 1. The architecture of the denoising autoencoder. The input data x is stochastic corrupted into $\widetilde{x}$ by mapping function $q(\widetilde{x}|x)$. The autoencoder then maps $\widetilde{x}$ to h and maps back h to r, the reconstruction result. $L(x, r)$ is the reconstruction error measurement function.

of the salient postures. Lu *et al.* [4] extracted spatio-temporal interest points from depth videos and built a cuboid similarity feature. Similarly, in [5], Omar and Zicheng quantized the 4D space and represented the possible directions for the 4D normal in order to build a histogram in the 4D space.

As mentioned before, skeletal information has strong representation power. Lu *et al.* [7] computed histograms of 3D joint locations, reprojected the extracted features using LDA [13], and clustered them into visual words. The temporal evolutions of these words were modeled by HMMs [14]. Jiang *et al.* [2] combined skeleton and depth information to obtain Local Occupancy Patterns (LOP) at each joint and built a Fourier Temporal Pyramid, an actionlet ensemble was learn to represent the actions. Jiajia [6] proposed a dictionary learning algorithm adding the group sparsity and geometry constrains, obtain an overcomplete set of the input skeletal features. The Temporal Pyramid Matching was used for keeping the temporal information.

Deep Learning [15–18] is a set of algorithms that attempt to learn a hierarchy of features by building high-level features from low-level ones. Some models such as CNN [18], DBN [16] and Autoencoders [15] have achieved surprising result in areas like computer vision, natural language processing and speech recognition. One reason for the success of deep learning methods is that they usually learn to capture the posterior distribution of the underlying explanatory factors for the data [19]. Therefore, rather than elaborately designing the hand-crafted features as in [5], we choose to learn high level features from data. The experimental results further prove the feasibility and validity of deep learning methods.

3 Proposed Method

In this section, we will first describe the basic Denoising Autoencoders. Next, we will extend the model by adding the category constraint, to make the learned features more discriminative and obtain better accuracies for recognizing actions.

Then we introduce the stacking techniques to build a deep architecture. Finally, we employ temporal pyramid matching to generate the temporal representation and do classification.

3.1 Denoising Autoencoders

Autoencoders were proposed by Hinton [15] to recognize handwritten digits, which achieved the state of the art at that time. An autoencoder is a special kind of neural networks whose target values are equal to the input ones. A single-layer Autoencoder comprises two parts: **encoder** and **decoder**.

Encoder: The transformation function maps an input vector x into a hidden layer feature vector h. Its typical form is a non-linearity function. For each example $x^{(i)}$ from a data set $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$, we define:

$$f_\theta(x^{(i)}) = s(Wx^{(i)} + b) \tag{1}$$

Decoder: The parameterized function maps the hidden layer feature vector h back to the input space, producing a reconstruction vector:

$$g_\theta(h^{(i)}) = s(W'h^{(i)} + c) \tag{2}$$

The set of parameters of this model is $\theta = \{W, W', b, c\}$, where W and W' are the encoder and decoder weight matrices and b and c are the encoder and decoder bias vectors. It is worth mentioning the input vector $x^{(i)}$ and the reconstruction vector $r^{(i)}$ have the same dimension d_x, the hidden layer $h^{(i)}$ has the dimension d_h, thus the size of W is the same as the size of transpose of W', which is $d_h \times d_x$.

The basic autoencoders aim to minimize the reconstruction error of all samples:

$$L_{AE}(\theta) = \sum_i L(x^{(i)}, g_\theta(f_\theta(x^{(i)}))) \tag{3}$$

In practice, the choice of function s is usually a sigmoid function $s(x) = \frac{1}{1+e^{-x}}$ or a tanh function $s(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and the loss function L is usually a square loss function $L(x, r) = \|x - r\|^2$.

Vincent [20] proposed Stacked Denoising Autoencoders (SDA), exploring a strategy to denoise corrupted version of input data. The input x is first corrupted into $\widetilde{x}$ using stochastic mapping $\widetilde{x} \sim q(\widetilde{x}|x)$. This is like randomly selecting some nodes of the input and blinding them, that is, every node in the input layer has a possibility q to be switched to zero. The stochastic corrupted data is regarded as the input of next layer, see Fig. 1. This yields the following objective function:

$$L_{DAE}(\theta) = \sum_i \mathbb{E}_{q(\widetilde{x}|x^{(i)})} \left[L(x^{(i)}, g_\theta(f_\theta(x^{(i)}))) \right] \tag{4}$$

where $\mathbb{E}_{q(\widetilde{x}|x)}[\cdot]$ is the expectation over corrupted examples $\widetilde{x}$ drawn from the corruption process $q(\widetilde{x}|x)$.

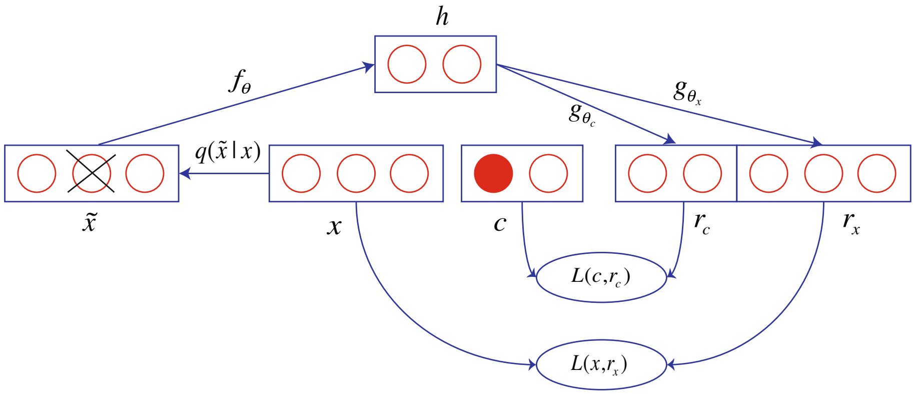

Fig. 2. The architecture of the denoising autoencoder after adding category constraint. c is a standard unit vector, indicating the category of the video where the frame belongs. The hidden layer h attempts to reconstruct x and c together, producing the reconstruction vector r_x and r_c. The objective error function is $L(c, r_c) + \lambda L(x, r_x)$.

The reason why DAE can denoise corrupted data is that the training data usually concentrate near a lower-dimensional manifold, yet most of the time the corruption vector is orthogonal to the manifold. The model learns to project the corrupted data back onto the manifold, thus denoising the input.

3.2 Adding the Category Constraint

Though the features learned by the denoised autoencoders can be highly expressive, as we use the frame-level joint features as the input, all the temporal and category information are discarded. Merely using the model mentioned above, the unsupervised learned features cannot distinguish the significant small differences between similar actions. We modify the denoising autoencoders, adding the category constraint, to make the model capable of emphasizing the imparities in different actions.

Fig. 2 demonstrates our modified autoencoder. Based on the structure of denoising autoencoders, we add an extra target c to the network where c is a vector whose length equals to the action class number d_c. The vector c has only one nonzero element whose index indicates the action type of the video where the example frame belongs. In consequence, a category vector r_c has to be reconstructed by the hidden layer h using a new mapping function g_{θ_c}. Similarly, r_x is the reconstruction vector of x by the mapping function g_{θ_x}. The new training objective of the denoised autoencoder with category constraint (DAE_CC) is:

$$L_{DAE\_CC}(\theta) = \sum_i \mathbb{E}_{q(\widetilde{x}|x^{(i)})} \left[L(x^{(i)}, g_{\theta_x}(f_\theta(x^{(i)}))) + \lambda L(c^{(i)}, g_{\theta_c}(f_\theta(x^{(i)}))) \right] \tag{5}$$

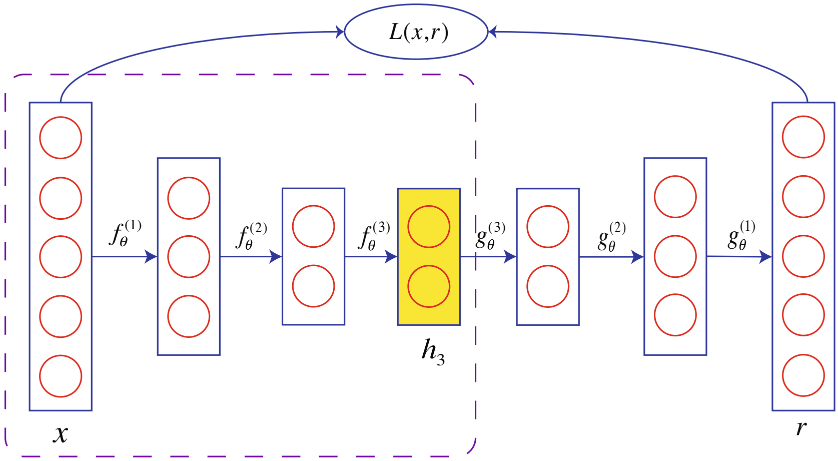

Fig. 3. Fine tuning of the stacking architecture. Each layer autoencoder is trained successively to obtain the encoding and decoding functions, which are used to initialize the parameters of the stacking architecture. All parameters are fine tuning to minimize the reconstruction error $L(x, r)$, by performing gradient descent. The structure inside the dotted box is the model to extract features and the deepest hidden layer h_3 is the final representation we seek.

where λ is a hyper-parameter controlling the strength of the category regularization. It can be optimized by stochastic gradient descent, analogous to the process of optimizing traditional autoencoders.

The reason why we use a regularization term rather than directly learn the class labels as targets is that the input is the joint vector for one frame, yet the class labels are for the whole video. Apparently there are some similar postures among actions. For example, the *stand and put the hands down* posture appears at the beginning of almost all actions. Training the same posture for different labels will lead to trivial results. The regularization term establishes a trade-off between preserving category information and reconstructing the input data.

3.3 Stacked Architecture

By stacking several layers of denoising autoencoders with the category constraint, we build a deep neural network with great expressive power. Greedy layer-wise training is employed: we first train the first layer to get the encoding function f_{θ_1}, then apply it on the clean input to compute the output value, which is used to train the second layer autoencoder to learn f_{θ_2}. The process is repeated from there. At last we fine-tune the deep neural network as in Fig. 3. We use the output of the last autoencoder as the output for the stacked architecture.

3.4 Temporal Representation and Classification

To add temporal information, a temporal pyramid matching (TPM) [6] is used to represent the temporal dynamics of these features. Motived by Spatial Pyramid Matching (SPM) [21], a max pooling function is used to generate the multi-scale structure. We recursively partition the video sequence into increasingly finer segments along the temporal direction and use max pooling to generate histograms from each sub-region. Typically, 4 levels with each containing 1, 2, 4 and 8 segments are used. The final feature is the concatenation of histograms from all segments.

After the final representation for each video is obtained, a multi-class linear SVM [22] is used to speed up the training and testing, results will be discussed in the next section.

4 Experimental Results

We evaluate our algorithm on a depth-based action recognition dataset, MSR Action3D dataset [12]. We compare our algorithm with several state-of-the-art methods on this dataset, the experimental result shows that our algorithm outperforms these methods. We also reveal the strong denoising capability of our method to reconstruct noisy 3D joint sequences. In all experiments, we train a deep architecture stacking by two autoencoders, where the first one contains 200 nodes in the hidden layer and the second one contains 400 nodes in the hidden layer. We penalize the average output $\bar{h_j}$ of the second autoencoder and pushing it to 0.1, in order to add some sparsity to the model and learn an over-completed representation of joint features. The parameter λ is set to 1.5.

4.1 MSR Action3D Dataset

MSR Action3D dataset [12] is an action dataset of depth sequences captured by a depth camera. The dataset contains 20 actions: *high arm wave*, *horizontal arm wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *hand clap*, *two hand wave*, *sideboxing*, *bend*, *forward kick*, *side kick*, *jogging*, *tennis swing*, *tennis serve*, *golf swing*, *pick up & throw*. Each action is performed by 10 subjects for three times. There are 567 depth map sequences in total. The provided skeleton data is used to train and test our model. We use the same experimental setting as in [2], half of the subjects are used for training and the rest half for testing. We compare our algorithm with several recent methods and report the results on Table 1. We obtain a recognition accuracy of 87.4%. Fig. 4 shows the confusion matrix of the proposed method. Fig. 5 compares the recognition accuracy for each action of our stacked denosing autoencoders with and without the category constraint. The recognition rate improve from 83.3% to 87.4% after adding the category constraint.

Table 1. Comparison of recognition rate on MSR Action3D Dataset

Method	Accuracy
Recurrent Neural Network [23]	0.425
Dynamic Temporal Warping [24]	0.54
Hidden Markov Model [14]	0.63
STIP [9] + BOW	0.696
Action Graph on Bag of 3D Points [12]	0.747
Eigenjoints [8]	0.823
Random Occupy Pattern [25]	0.865
HON4D [5]	0.859
Proposed Method	**0.874**

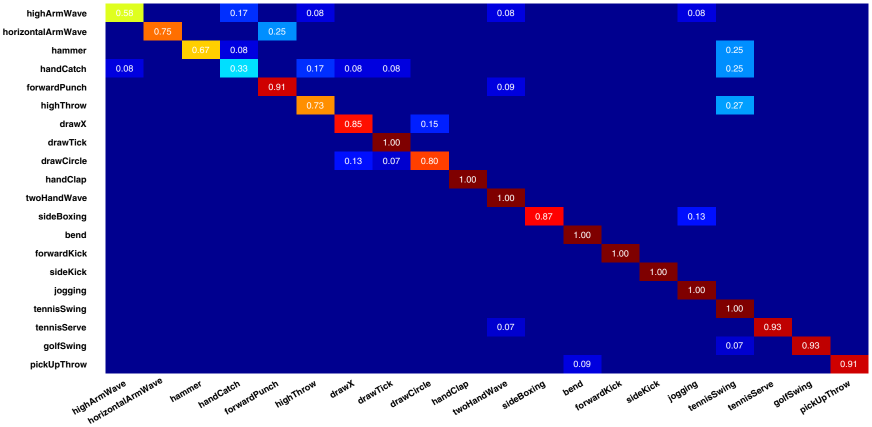

Fig. 4. Confusion matrix of the proposed method on MSR Action3D dataset

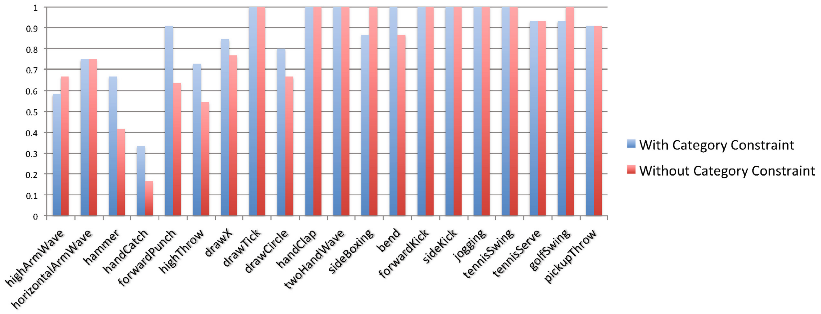

Fig. 5. Comparison of the recognition accuracy for each action before and after adding the category constraint

4.2 Capability to Denoise Corrupted Data

Our model has strong capability to reconstruct realistic data from corrupted input. The top row of Fig. 6 is an action sequence *high arm wave* selected from MSR Action3D dataset. In order to better demonstrate our algorithm efficiency, we add some Gaussian noise to the joint positions and leave out joints stochasticly. The bottom row is the reconstruction action sequence, where we can observe that the missing joints are all restored via our model and the motions are more natural and fluent than before.

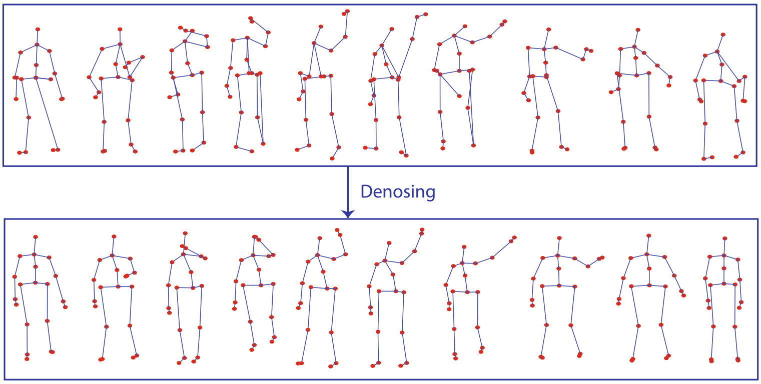

Fig. 6. Examples showing the capability of our model to denoise corrupted data. Top: the corrupted input 3D joint sequence *high arm wave* from MSR Action3D dataset. Bottom: the reconstructed 3D joint sequence.

5 Conclusion

This paper presented a novel feature learning methodology for human action recognition with depth cameras. To better represent the 3D joint features, a deep stacked denoising autoencoder that incorporated with the category constraint was proposed. The proposed model is capable of capturing subtle spatio-temporal details between actions and robust to the noises and errors in the joint positions. The experiments demonstrated the effectiveness and robustness of the proposed approach. In the future, we aim to integrate the temporal information into our feature learning architecture.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Grant No. 61272317) and the General Program of Natural Science Foundation of Anhui of China (Grant No. 1208085MF90).

References

1. Yang, X., Zhang, C., Tian, Y.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM International Conference on Multimedia, pp. 1057–1060. ACM (2012)
2. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: Computer Vision and Pattern Recognition, CVPR (2012)
3. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM 56(1), 116–124 (2013)
4. Xia, L., Aggarwal, J.: Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera (2013)
5. Oreifej, O., Liu, Z., Redmond, W.: Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: Computer Vision and Pattern Recognition, CVPR (2013)
6. Luo, J., Wang, W., Qi, H.: Group sparsity and geometry constrained dictionary learning for action recognition from depth maps (2013)
7. Xia, L., Chen, C.C., Aggarwal, J.: View invariant human action recognition using histograms of 3d joints. In: Computer Vision and Pattern Recognition Workshops, CVPRW (2012)
8. Yang, X., Tian, Y.: Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In: Computer Vision and Pattern Recognition Workshops, CVPRW (2012)
9. Laptev, I.: On space-time interest points. International Journal of Computer Vision 64(2-3), 107–123 (2005)
10. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance 2005, pp. 65–72. IEEE (2005)
11. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Computer Vision and Pattern Recognition, CVPR (2008)
12. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: Computer Vision and Pattern Recognition Workshops, CVPRW (2010)
13. Scholkopft, B., Mullert, K.R.: Fisher discriminant analysis with kernels. Neural Networks for Signal Processing IX
14. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
15. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
16. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation 18(7), 1527–1554 (2006)
17. Bengio, Y.: Learning deep architectures for ai. Foundations and Trends® in Machine Learning 2(1), 1–127 (2009)
18. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
19. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives (2013)

20. 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. The Journal of Machine Learning Research 9999, 3371–3408 (2010)
21. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Computer Vision and Pattern Recognition, CVPR (2006)
22. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 27 (2011)
23. Martens, J., Sutskever, I.: Learning recurrent neural networks with hessian-free optimization. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 1033–1040 (2011)
24. Müller, M., Röder, T.: Motion templates for automatic classification and retrieval of motion capture data. In: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 137–146. Eurographics Association (2006)
25. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3d action recognition with random occupancy patterns. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 872–885. Springer, Heidelberg (2012)