Action Recognition Using Hierarchical Independent Subspace Analysis with Trajectory

Vinh D. Luong, Lipo Wang, and Gaoxi Xiao

School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
ducvinh001@e.ntu.edu.sg, {ELPWang,EGXXiao}@ntu.edu.sg

Abstract. Action recognition in videos is an important and challenging problem in computer vision. One of the most crucial aspects of a successful action recognition system is its feature extraction component. Stacked, convolutional Independent Subspace Analysis (SC-ISA), has the best result among unsupervised learning algorithms for action recognition in Hollywood 2 (53.3%) and Youtube (75.8%). However, its performance still lags behind the current state-of-the-art, which uses computer vision-based feature engineering extraction techniques, by about 10%. In this paper, we improve SC-ISA's results by incorporating motion information into SC-ISA. By extracting blocks following motion trajectories in videos, we are able to reduce noise and increase the number of training samples without degrading the network's performance when training and testing SC-ISA. We increase SC-ISA's result by about 1%.

1 Introduction

Researchers in the field of action recognition in videos have made remarkable progress recently. As observed from the dataset aspect of the problem, the field has advanced rapidly from the limited, constrained datasets like the KTH dataset [1] to the more realistic and more challenging ones, e.g. Hollywood 2 [2], to large-scale, "in the wild" datasets such as HMDB 51 [3], UCF 101 [4], Sports-1M dataset [5]. Conventional computer vision-based techniques are currently the best methods [6], [7] to extract local, low level features for action recognition systems.

Deep learning has been a great success in object detection, localization and classification in images. In the supervised learning front, convolutional neural networks (CNN) [8] are currently the state-of-the-art in these tasks [9], [10]. As for unsupervised learning, large-scale networks have also made remarkable results such as the automatic emergence of human face and cat face detectors in the famous Google network [11]. However, in the problem of action recognition in videos, deep networks have not enjoyed such stunning progress. Unsupervised deep networks [12], [13] currently lag behind the current state-of-the-art [7] in

H. Handa et al. (eds.), *Proc. of the 18th Asia Pacific Symp. on Intell. & Evol. Systems – Vol. 1*,
Proceedings in Adaptation, Learning and Optimization 1, DOI: 10.1007/978-3-319-13359-1_42

relatively small but challenging datasets such as Hollywood 2. Furthermore, unsupervised networks have not been scaled up to tackle bigger datasets such HMDB 51 and UCF 101. Until recently, supervised networks [5] were also far behind the state-of-the-art [7]. This is really puzzling because deep networks with all the sophisticated learning algorithms have not only succeeded in the image domain but also they have made big improvements in speech recognition, a temporal domain. One possible answer is that, deep networks have not been able to incorporate motion information effectively into their network. In this paper, we will explicitly include motion information in training and testing SC-ISA network [12], which is also the current state-of-the-art for unsupervised learning algorithms in action recognition.

2 Review of Related Works

2.1 A Common Action Recognition Framework

In this paper, we limit our scope to action recognition systems that deal with local features because the local methods are the most dominant and the most accurate algorithms in the field at the moment. For global features and more comprehensive surveys of action recognition, readers should refer to [14], [15] and [16].

A common framework in action recognition for local features is as followed. First, features are extracted from training videos. Then, these features are quantized in some dictionaries using clustering and feature encoding methods such as k-means, Fisher vector [17] and VLAD [18]. After that, each video is encoded into vectors using the resulted dictionaries. Finally, a classifier, e.g. SVM, is used to train and test the videos.

2.2 Improved Trajectories

Wang et al. [6], [7], [19] explicitly reduce camera motion and use a dense optical flow algorithm [20] to track densely sampled points. They then compute Histograms of Oriented Gradients (HOG), Histograms of Optical Flow (HOF) [21], HOG/HOF [22] and Motion Boundary Histogram (MBH) [23] descriptors following the computed trajectories. They also introduce trajectory descriptor as normalized trajectory displacement. Using these newly computed descriptors, they achieve the best results in almost all datasets they test with, including challenging ones such as Hollywood 2 [2] (64.3%), Youtube [24] (85.4%) UCF-101 [4] (85.9%), HMDB51 [3] (57.2%).

2.3 Independent Subspace Analysis

Independent Subspace Analysis: Independent Subspace Analysis (ISA) is an unsupervised learning algorithm that models complex cells in V1 [25], [26]. A complex cell fires almost the same response to a grating regardless of the grating's phase. By putting linear features into groups (subspaces), features learned by ISA

are able to display limited phase invariance as well as selectivity to frequency and orientation. The combination of these features is what distinguishes ISA from linear methods such as Independent Component Analysis (ICA) [27], [26] in modeling cells in V1.

ISA originates from ICA. In ICA, an input vector $\mathbf{z}$[1], which, in our case, is resulted from the whitening preprocessing of a 3-D video block as linear combination of basis vectors (features) $\mathbf{b_i}$:

$$\mathbf{z} = \sum_{i=1}^{n} s_i \mathbf{b_i} \tag{1}$$

or:

$$\mathbf{z} = \mathbf{B}\mathbf{s} \tag{2}$$

where $\mathbf{B}$ is the matrix consisting of vectors $\mathbf{b_i}$ as columns and s_i are the coefficients, which are random variables. ICA learns $\mathbf{b_i}$ and s_i such that s_i are nongaussian and independent.

An interesting extension of ICA is multidimensional ICA [2] [28], [25], where s_i are not all mutually independent. In fact, the model assumes s_i are uncorrelated and have unit variance. The coefficients s_i are put into groups or subspaces as followed:

$$\mathbf{z} = \sum_{k=1}^{m} \sum_{i \in S(k)} s_i \mathbf{b_i} \tag{3}$$

Here, input $\mathbf{z}$ is decomposed into the sum of m subspaces $S(k)$, each of which contains a number of the components $\mathbf{b_i}$. We assume that the total number of features $\mathbf{b_i}$ are equal to the dimension of the input vector $\mathbf{z}$ and the matrix $\mathbf{B}$ is invertible. Therefore, given an input $\mathbf{z}$, s_i can be computed as:

$$s_i = \mathbf{v_i^T z} \tag{4}$$

where $\mathbf{v_i}$ are the column vectors of the inverse matrix $\mathbf{V}$of matrix $\mathbf{B}$. Vectors $\mathbf{v_i}$ are also called feature detectors.

Note that multidimensional ICA is still a linear model, thus it can not learn invariance feature. In order to transform multidimensional ICA into a nonlinear model that learns invariance feature, Hyvärinen et al. [25] uses the principle of invariant feature subspace [29], which utilizes a linear subspace as an invariant feature within the feature space. Given an input, the value of the invariant feature is the norm projection of that input to the corresponding linear subspace:

$$e_k = \sqrt{\sum_{i \in S(k)} s_i^2} = \sqrt{\sum_{i \in S(k)} (\mathbf{v_i^T z})^2} \tag{5}$$

[1] In this paper, we assume inputs are preprocessed by the same whitening preprocessing step as in Le et al. [12]

[2] Apparently, multidimensional ICA is called ISA or general ISA in Signal Processing community nowadays

Note that, the right hand side of the equation 5 is called L2-pooling. Here, e_k is the value of the invariant feature when the input is projected into subspace S_k. e_k is also called energy detector.

Given an input $\mathbf{z}$, the sparseness of the square energy detectors e_k^2 is:

$$\sum_{k=1}^{m} h(e_k^2) = \sum_{k=1}^{m} h(\sum_{i \in S(k)} s_i^2) = \sum_{k=1}^{m} h(\sum_{i \in S(k)} (\mathbf{v_i^T z})^2) \tag{6}$$

where h is a nonlinear function, which is suitable to measure the sparseness of the distribution of e_k^2 and m is the number of subspaces. In [26], h is chosen as $h(x) = -\sqrt{x}$.

For T inputs $\mathbf{z_j}$ $(j = 1, ..., T)$, the sparseness is measured as:

$$S_{sparse} = \sum_{j=1}^{T} \sum_{k=1}^{m} h(e_k^2) = \sum_{j=1}^{T} \sum_{k=1}^{m} h(\sum_{i \in S(k)} (\mathbf{v_i^T z_j})^2) \tag{7}$$

Here, feature detectors $\mathbf{v_i}$ can be learned by maximizing the sparseness S_{sparse} with regards to $\mathbf{v_i}$ subject to:

$$\mathbf{V V^T} = \mathbf{I} \tag{8}$$

where $\mathbf{V}$ is the matrix with columns as $\mathbf{v_i}$. The reason for $\mathbf{V}$ to be an orthogonal matrix is as followed. As the result of whitening, $E\{\mathbf{z z^T}\} = \mathbf{I}$. Therefore,

$$E\{\mathbf{z z^T}\} = E\{\mathbf{B s s^T B^T}\} = \mathbf{B} E\{\mathbf{s s^T}\} \mathbf{B^T} = \mathbf{I} \tag{9}$$

As we assume s_i are uncorrelated and have unit variance, so $\mathbf{B B^T} = \mathbf{I}$. Because $\mathbf{V}$ is the inverse of $\mathbf{B}$, thus $\mathbf{V}$ is also an orthogonal matrix.

Stacked, Convolutional ISA (SC-ISA): Le et al. [12] uses convolution and stacked layer idea [30] to scale up ISA into a hierarchical ISA network. In this network, each layer implements the ISA algorithm. In the first layer, they train ISA with inputs that have small spatial size. As for training of subsequent layers, in order to compute input for the next layer, the learned weights of the previous layer are copied over and convolved with input of larger spatial size. The outputs of these convolution operations will be combined and then reduced in dimension using PCA before they become the input for the next layer. The authors train the first layer to converge first before training the second layer and so on. After that, they concatenate the features learned from all layers, as previously done in [30], to create local features for further processing stages.

With this hierarchical network, they obtained the state-of-the-art results of unsupervised learning algorithms for challenging datasets such as Hollywood 2 [2] (53.5%), Youtube [24] (75.8%).

3 Proposed Improvements to Increase Performance of ISA

3.1 Trajectories

Le et al. [12] use dense sampling approach to extract blocks from videos for training and testing. We note that the way Le et al. [12] extracts blocks from input videos for training and testing might not take advantage of the dynamics of motion in videos. They randomly extract sequences of patches that have the same spatial coordinates in each frame. Such straight blocks usually do not capture many types of motion correctly. Thus, the training inputs might contain a lot of noise, which reduces the accuracy of the network learned by ISA. Inspired by the trajectory approach [6], [7], we extract blocks following motion trajectories, instead of following straight paths as in Le et al. [12].

3.2 Increase the Number of Training Inputs

We note that Le et al. [12] uses only 200 blocks per video to train, which we think is probably too small to be representative of each training video. We hypothesize that, because of too much noise from the way they extract training inputs, more training samples will only degrade the performance of the system. Blocks following motion trajectories might have less noise than straight blocks. Thus, we might be able to use more trajectory blocks per video to train the system.

4 Experiments

4.1 Baseline Code and Dataset

Le et al.'s [12] release two versions of their source code. One version can be used with low end systems where users do not have powerful NVIDIA graphics cards. This version only makes use of one resolution version of video datasets. The other version utilizes multiple resolution versions of video datasets to extract more features, thus has better classification results. Due to the limited computational power available to us, we use the former version as the baseline code in our experiments. Because of the same reason, we have to make some further reduction on the video dataset that we use by assuming that running on a smaller but challenging subset of the Hollywood 2 dataset would reflect the performance of our algorithm when running with the full dataset. The Hollywood 2 dataset has 12 actions in total and we select a challenging subset of 5 actions, in which even the baseline code has difficulties with. Working with a smaller, challenging subset would help us to save time for running more experiments and at the same time it is not likely to compromise the performance of our proposed changes.

In training and testing the baseline code and our modifications, we use the same procedure used by Le et al.'s [12], which is listed in the subsection 2.1. In vector quantization step, we run k-means 8 times and select the best results.

Table 1. Baseline code's result with the chosen subset of Hollywood 2 dataset using half-resolution version

Action name	Average Precision (AP)
HugPerson	43.0024%
Kiss	68.6265%
SitDown	72.2845%
SitUp	29.8177%
StandUp	76.3940%
Mean AP	**58.0250%**

The result for our chosen subset when running the baseline code is shown in table 1. In comparison with the mean Average Precision (mAP) of about 50.5%[3] when running the baseline code with the full Hollywood 2 dataset, it is obvious that our chosen subset is quite challenging. Even though, the number of actions is reduced by more than half, the mAP of the subset is only about 8% higher than that of the full dataset.

4.2 Trajectories

Currently, we use the dense trajectory extraction approach in [6] to extract dense trajectories from input videos. We modify the original source code to relax the constraints of a valid trajectory so that we can have enough trajectories for each input video and the trajectories are able to capture the motion dynamics of some difficult actions. For each video, once we extract dense trajectories, we extract blocks following the trajectories for training and testing. Figure 1 shows the trajectories in two videos in the subset that we experiment with. Each trajectory shown here, as a small green line or curve ended with a red point, is an optical flow of one pixel tracked across a number of frames[4].

Fig. 1. Motion trajectory

Figure 2 shows the performance of the baseline code and our trajectory modification running with various training samples (blocks) per video (BpV) using

[3] As posted in the authors' website: http://ai.stanford.edu/~wzou/. We also obtain a similar result (about 50.4%) when running the baseline code with the full Hollywood 2 dataset (half-resolution version).

[4] (The default is 15 frames

the half resolution version of the dataset. Note that, the trajectory modification's results outperforms the baseline code's best results in all settings. The baseline code's mAP increases as the number of blocks per video increases up to 600 BpV, then falls off rapidly as we adds more training blocks up to 1000 BpV. After that, the baseline code's performance goes up slightly as the number of blocks increases. In contrast, our trajectory modification's mAP decreases as we use more training blocks up to 600 BpV. After that, the performance of our trajectory modification increases as more training blocks per video are employed.

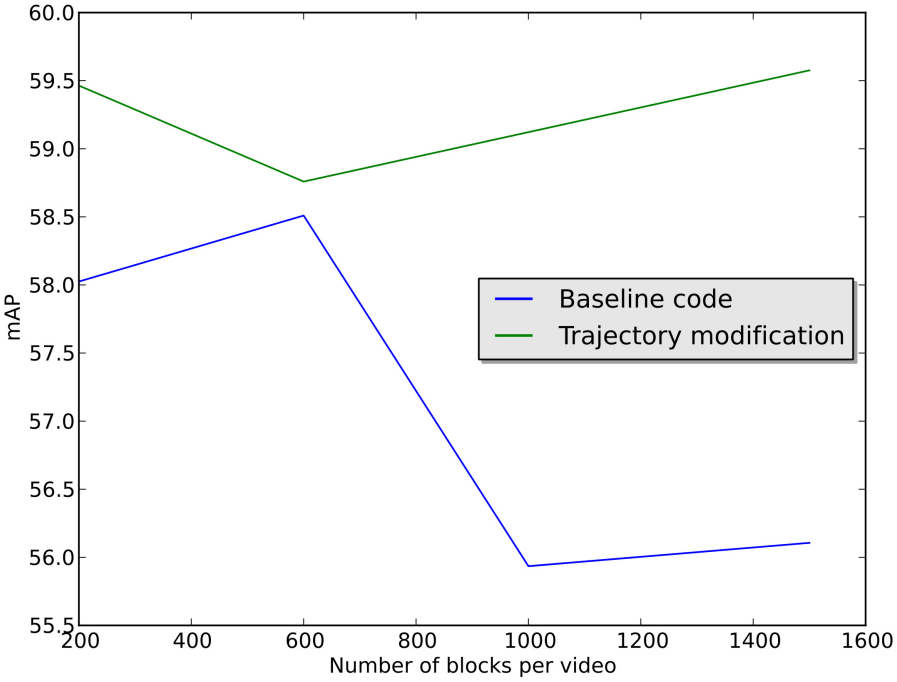

Fig. 2. Comparison of performance of the baseline code and our trajectory modification

The best result of our modification, which is about 1% (mAP) better than the best result of the baseline code, is obtained when training with the most samples per video (1500 BpV). On the other hand, too many training blocks per video reduces the performance of the baseline code significantly. As we only experiment with one version of a fixed resolution of the dataset each time, we expect the performance gain will increase when we train and test with many versions of multiple resolutions at the same time.

5 Discussion

Dense trajectory is a very powerful tool to capture motion inside a video. However, the strength of its coverage is also its weakness when applied to unsupervised learning algorithms. While dense coverage ensures that the resulting trajectories are unlikely to miss any motion in videos, it creates many irrelevant trajectories, which are not only a great computational burden but also a source of noise to train networks. We think that, for human activity datasets like Hollywood 2, a good algorithm for human detection and tracking will definitely help with the removal of unnecessary trajectories, thus alleviate the above mentioned problems. We plan to investigate this direction further in our future work.

As pointed out by Le et. al. [12], when applying ISA with video blocks, the features learn to detect a moving edge and they are selective to the velocity of motion. It means that the features learned by ISA are probably similar to motion features like HOF, MBH. If it is indeed the case, ISA features combined with form features such as HOG, SIFT might increase the performance of the system. Recently, Zhou et al. [31] use a similar stacked, convolutional network as the one described in the subsection 2.3 to implement temporal slowness [32], [33], [34]. One interesting result from their paper is that they trained the network with temporal slowness using a natural video dataset and from the trained network, they are able to extract features from static images of different image datasets. These features consistently improve the classification results for the image datasets by 4% or 5%. Inspired by this, we plan to investigate the features extracted when applying temporal slowness to tracked human sequences in the videos of the Hollywood 2 dataset. We would like to see how these features perform in isolation and in combination with SC-ISA's features.

Given that there is a big gap in performance between unsupervised learning algorithms and the state-of-the-art feature engineering method [7] in action recognition, one can reasonably question whether efforts to design better unsupervised learning algorithms for this problem are worthwhile. We believe that the answer is yes. Unsupervised algorithms can arguably be more adaptable to different datasets than fixed engineering features. Furthermore, learning can produce unexpected features, different or complementary to engineering features, thus maintaining a healthy competition between these two approaches could be much more beneficial for the progress of the field than focusing only on a single approach. In addition, we would like to point out that even the state-of-the-art combines different engineering features together. If unsupervised algorithms like SC-ISA also combine with different features either from other unsupervised algorithms or from feature engineering methods, the performance of the resulting systems will more likely increase and be comparable to the state-of-the-art. Finally, brain-inspired algorithms can make use of the proven principles in the human visual cortex, which is still the best general vision system at the moment.

6 Conclusion

We incorporate motion information into SC-ISA by training and testing SC-ISA using blocks following motion trajectories. We also show that, empirically, SC-ISA's performance degrades significantly when we train it with many more straight blocks that that of [12]. Interestingly, when SC-ISA is trained with more trajectory blocks, the performance decreases at first and then increases as more training blocks are added. Even though we only experiment SC-ISA with trajectory using one resolution version of a subset of the Hollywood 2 dataset, we expect the performance gain (1% better than the baseline code) will increase as we run with multiple resolution versions of the dataset. Unsupervised learning algorithms like SC-ISA are very interesting because potentially they can make use of the huge number of unlabeled videos available. However, we think that good human detection, or object detection in general, and tracking algorithms are needed in order to enable unsupervised learning algorithms to work with such large-scale video datasets.

References

1. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 3, pp. 32–36. IEEE (2004)
2. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR, pp. 2929–2936. IEEE (2009)
3. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: A large video database for human motion recognition. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2556–2563. IEEE (2011)
4. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
5. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2014)
6. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR, pp. 3169–3176. IEEE (2011)
7. Wang, H., Schmid, C.: Action Recognition with Improved Trajectories. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 3551–3558 (2013)
8. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324 (1998)
9. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR abs/1311.2524 (2013)
10. 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)

11. Le, Q.V., Ranzato, M., Monga, R., Devin, M., Corrado, G., Chen, K., Dean, J., Ng, A.Y.: Building high-level features using large scale unsupervised learning. In: ICML, icml.cc. Omnipress (2012)
12. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3361–3368. IEEE (2011)
13. 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, Part VI. LNCS, vol. 6316, pp. 140–153. Springer, Heidelberg (2010)
14. Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action representation, segmentation and recognition. Computer Vision and Image Understanding 115, 224–241 (2011)
15. Poppe, R.: A survey on vision-based human action recognition. Image Vision Comput. 28, 976–990 (2010)
16. Jiang, Y.G., Bhattacharya, S., Chang, S.F., Shah, M.: High-level event recognition in unconstrained videos. IJMIR 2, 73–101 (2013)
17. Perronnin, F., Sánchez, J., Mensink, T.: Improving the Fisher Kernel for Large-Scale Image Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
18. Jegou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: CVPR, pp. 3304–3311. IEEE (2010)
19. Wang, H., Schmid, C.: Lear-inria submission for the thumos workshop. In: ICCV Workshop on Action Recognition with a Large Number of Classes (2013)
20. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Heidelberg (2003)
21. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
22. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. IEEE Computer Society (2008)
23. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)
24. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 1996–2003. IEEE (2009)
25. Hyvärinen, A., Hoyer, P.: Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. Neural Computation 12, 1705–1720 (2000)
26. Hyvärinen, A., Hurri, J., Hoyer, P.O.: Natural Image Statistics: A Probabilistic Approach to Early Computational Vision, vol. 39. Springer (2009)
27. Comon, P.: Independent component analysis, a new concept? Signal Processing 36, 287–314 (1994)
28. Cardoso, J.: Multidimensional independent component analysis. In: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 4, pp. 1941–1944. IEEE (1998)
29. Kohonen, T.: Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map. Biological Cybernetics 75, 281–291 (1996)

30. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 609–616. ACM (2009)
31. Zou, W.Y., Ng, A.Y., Zhu, S., Yu, K.: Deep Learning of Invariant Features via Simulated Fixations in Video. In: NIPS, pp. 3212–3220 (2012)
32. Hinton, G.E.: Connectionist learning procedures. Artificial Intelligence 40, 185–234 (1989)
33. Mitchison, G.: Removing Time Variation with the Anti-Hebbian Differential Synapse, Neural Computation (1991)
34. Földiák, P.: Learning Invariance from Transformation Sequences. Neural Computation (1991)