Procedia Computer Science

Volume 53, 2015, Pages 391–399

2015 INNS Conference on Big Data

Generic Object Recognition with Local Receptive Fields Based Extreme Learning Machine

Zuo Bai, Liyanaarachchi Lekamalage Chamara Kasun and Guang-Bin Huang

School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, {zbai1, chamarak001}@e.ntu.edu.sg, egbhuang@ntu.edu.sg

Abstract

Generic object recognition is to classify the object to a generic category. Intra-class variabilities cause big troubles for this task. Traditional methods involve plenty of pre-processing steps, like model construction, feature extraction, etc. Moreover, these methods are only effective for some specific dataset. In this paper, we propose to use local receptive fields based extreme learning machine (ELM-LRF) as a general framework for object recognition. It is operated directly on the raw images and thus suitable for all different datasets. Additionally, the architecture is simple and only requires few computations, as most connection weights are randomly generated. Comparing to state-of-the-art results on NORB, ETH-80 and COIL datasets, it is on par with the best one on ETH-80 and sets the new records for NORB and COIL.

Keywords: Generic object recognition, local receptive fields, Extreme Learning Machine (ELM)

1 Introduction

Generic object recognition is to classify an unknown object to a certain generic category [16]. It remains as a challenging task due to its large amount of intra-class variabilities, such as different objects, poses, lighting conditions. The existing methods require plenty of human intervention, such as model construction [16], features extraction [12] , etc. Furthermore, these methods are only applicable for some specific tasks since features or models is not guaranteed for different tasks [2]. Convolutional neural network (CNN) is introduced in [15] and operated on the raw pixels. It presents remark performance on many image processing tasks [27]. Back-propagation (BP) algorithm [24] is adopted to adjust the connection weights in CNN. Thus, CNN inherits trivial issues from BP algorithm, such as local minima, time consuming, etc. Furthermore, CNN requires huge computations and training set to tune numerous connection weights.

Later, local receptive fields based extreme learning machine (ELM-LRF) is proposed in [7], which also handles the raw images directly. It generates the input weights randomly and calculates the output weights analytically, providing a simple and deterministic solution. Additionally, the requirement for computational capability and training samples are also largely reduced since most connection weights (the input ones) are simply generated randomly.

Selection and peer-review under responsibility of the Scientific Programme Committee of INNS-BigData2015

doi:10.1016/j.procs.2015.07.316

In this paper, we propose to use ELM-LRF as a general framework for generic object recognition. ELM-LRF has several advantages: 1) it does not use task specific information for learning; 2) it is a simple learning algorithm; 3) it is computational efficient as most connection weights are generated randomly. Subsequently, we evaluate ELM-LRF on different generic object recognition datasets, NORB [15], ETH-80 [16], COIL [21]. It shows better results than current state-of-the-art for NORB, COIL and is comparable with the best result on ETH-80.

2 Reviews of related works

2.1 Generic object recognition

Generic object recognition is to classify an individual object to a certain category and is also called *object categorization* [16]. Various method are proposed as follows:

i *Shape-based methods:* Shape models are constructed explicitly for subsequent recognition, while other attributes, like color or texture, are ignored [16, 17].

ii *Appearance-based methods:* Appearance information, like texture and color histograms, may be useful. PCA or other compression methods are used to generate compact representations for the highly-correlated information. Subsequently, each object is classified based on the similarity between itself and the compact representation [30].

iii *Local feature-based methods:* Different local features are proposed, including scale invariant descriptors (SIDs)[12], SIFT features [30], etc. Classifiers are followed to handle these features.

2.2 Convolutional neural network (CNN)

CNN [14] is a variant of multilayer feedforward neural networks inspired from biology [10]. Unlike aforementioned methods, CNN is operated directly on the raw pixels and requires no pre-processing. Additionally, CNN is more general than traditional methods, as the features are learned by the network itself. And the common approach to train a CNN is the back-propagation (BP) algorithm [24].

In some recent variants of CNN, such as GoogLeNet [27], superior performance is presented on super-large image datasets, like ImageNet [25]. However, with vast parameters to be tuned, huge computational capability is required. In addition, the training set has to be large enough to train the network properly. Therefore, it comes to a question: *is there any simple network that can handle raw images directly, while does not require learning with back-propagation algorithm?*

3 Local receptive fields based extreme learning machine (ELM-LRF)

3.1 Fully connected ELM

ELM is a generalized single-hidden layer feedforward neural networks (SLFNs) with many types of hidden nodes [9]. Input and hidden nodes are in full connection. It theoretically proves that hidden nodes can be generated randomly, as long as the activation functions of hidden nodes are nonlinear piecewise continuous [8]. Although ELM has some relationship with previous works

such as QuickNet [28] and random vector functional link (RVFL) [4], there exist significant differences between them. The detailed relationship and differences can be found in [6].

Unlike traditional learning methods, ELM does not require any iterative tunings. It presents better accuracy and high efficiency, in various applications such as system modelling, biomedical analysis, etc. [9]. Given a set of training data $(\mathbf{x}_i, \mathbf{t}_i), i = 1, \cdots, N, \mathbf{x}_i \in \mathbf{R}^{1\times d}, \mathbf{t}_i \in \mathbf{R}^{1\times m}$, state ELM implementation in matrix form:

$$\begin{aligned} &\mathbf{x}_i \to \mathbf{h}(\mathbf{x}_i) = [h_1(\mathbf{x}_i), \cdots, h_L(\mathbf{x}_i)], \ i = 1, \cdots, N \\ &\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \end{aligned} \tag{1}$$

where $\mathbf{H}$ and $\mathbf{T}$ are:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix}_{N\times L}, \ \mathbf{T} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_N \end{bmatrix}_{N\times m} \tag{2}$$

There are various methods to calculate $\boldsymbol{\beta}$ [1]. An efficient closed-form solution is [9]:

$$\boldsymbol{\beta} = \begin{cases} \mathbf{H}^T \left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{T}, & \text{if } N \le L \\ \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1} \mathbf{H}^T\mathbf{T}, & \text{if } N > L \end{cases} \tag{3}$$

3.2 Locally connected ELM

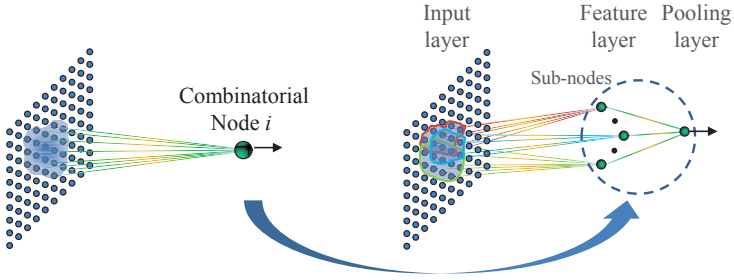


Figure 1: ELM combinatorial node

When facing locally correlated applications, like image processing and speech recognition, ELM-LRF is proposed to handle the local structures. It was shown in [7] that different shapes of local receptive fields may be suited for different applications. For instance, McDonnell *et al.* utilize random sampling method to generate the receptive fields and produce superior accuracy on the MNIST, NORB and SVHN datasets [18]. Subsequently, combinatorial node can be formulated to generate even more abstract representations of the raw inputs by combining several sub-nodes together, as shown in Fig. 1.

3.3 One feasible network of ELM-LRF

ELM-LRF is a two-stage network: (1) **tuning-free** nodes; (2) least-squares solution $\boldsymbol{\beta}$. Although many types of local receptive fields and combinatorial nodes are applicable, for simplicity, we use convolution operation and square/square-root pooling to construct one feasible

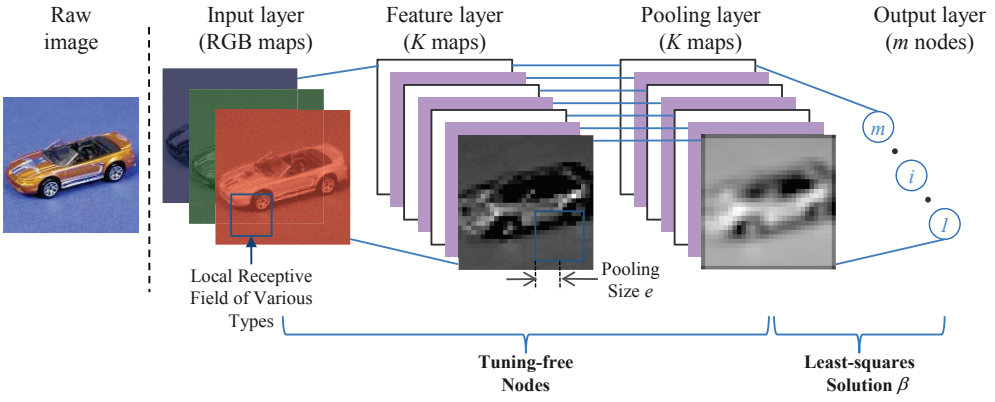


Figure 2: One feasible network of ELM-LRF: convolution and square/square-root pooling

network as in Fig. 2. The car image is chosen from ETH-80 ,where the input layer includes 3 RGB maps. And there are K maps in the feature and pooling layers in order to generate comprehensive representations for the raw image.

3.3.1 Tuning-free nodes

The nodes in the pooling layer are tuning-free, connecting with the input layer by random convolutional weights and pooling structure.

i *Random convolutional weights:* The convolutional weights between input and feature layer are random. The input image is $d \times d$ and the local receptive field is $r \times r$. Thus, the feature map is $(d-r+1) \times (d-r+1)$: 1) randomly generate $\hat{\mathbf{A}}^{\text{init}} \in \mathbf{R}^{r^2 \times K}$ based on standard Gaussian distribution; 2) orthogonalize into $\hat{\mathbf{A}} \in \mathbf{R}^{r^2 \times K}$ with SVD method; 3) reshape the columns of $\hat{\mathbf{A}}$, $\hat{\mathbf{a}}_k$, into $\mathbf{a}_k \in \mathbf{R}^{r \times r}, k = 1, \cdots, K$. Thus, node (i, j) in the k-th feature map, $c_{i,j,k}$ is calculated as:

$$c_{i,j,k}(\mathbf{x}) = \sum_{m=1}^{r} \sum_{n=1}^{r} x_{i+m-1,j+n-1} \cdot a_{m,n,k} \qquad i, j = 1, \cdots, (d-r+1) \tag{4}$$

ii *Square/square-root pooling:* As shown in Fig. 2, nodes in the feature layer are grouped within each pooling area, formulating subsequent pooling layer. Thus, node (p, q) in the k-th pooling map, $h_{p,q,ks}$ is:

$$h_{p,q,k} = \sqrt{\sum_{i=p-e}^{p+e} \sum_{j=q-e}^{q+e} c_{i,j,k}^2} \qquad p, q = 1, \cdots, (d-r+1) \qquad c_{i,j,k} = 0 \text{ if } (i, j) \text{ out of bound} \tag{5}$$

3.3.2 Regularized least-squares solution

All pooling nodes are calculated by solving (4) and (5) sequentially. Only the output weight $\boldsymbol{\beta}$ needs computation. Concatenating all pooling nodes into a row vector and putting all rows of N training samples together, the matrix $\mathbf{H} \in \mathbf{R}^{N \times (d-r+1)^2}$ is generated:

Table 1: Datasets descriptions

Dataset	# of categories	# of training data	# of testing data	# of input channels
NORB	5	24300	24300	2
ETH-80	8	1640	1640	3
COIL	100	1800	5400	3

$$\boldsymbol{\beta} = \begin{cases} \mathbf{H}^T \left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{T}, & \text{if } N \leq K \cdot (d - r + 1)^2 \\ \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1} \mathbf{H}^T\mathbf{T}, & \text{if } N > K \cdot (d - r + 1)^2 \end{cases} \tag{6}$$

ELM-LRF is different from conventional CNNs: 1) ELM-LRF is more flexible as different types of local receptive fields generated randomly according to some continuous probability distribution are applicable; 2) hidden nodes in ELM-LRF are tuning-free and the output weight $\boldsymbol{\beta}$ is calculated analytically. Thus, ELM-LRF provides an efficient and deterministic solution. It should be noted that ELM-LRF is different from [26] in the sense that conventional neurons can be naturally used in ELM while keeping the rest of ELM architectures and solutions unchanged.

4 Experiments

In this section, we conduct thorough investigations of ELM-LRF on generic object recognition tasks, NORB [15], ETH-80 [16] and COIL [21]. All experiments are conducted in MATLAB 2013a, on a Windows Server 2012, with Intel Xeon E5-2650, 2GHz CPU, 256G RAM.

4.1 Datasets descriptions

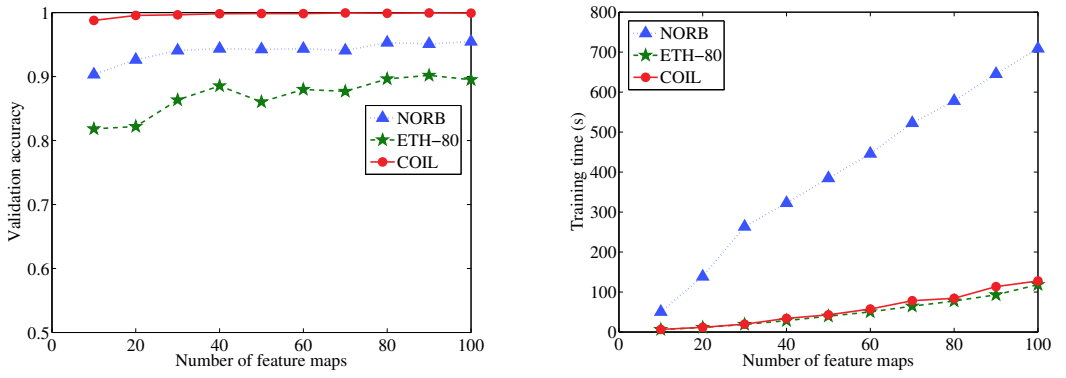
These datasets are subject to different variations: poses, lighting conditions, scales, positions and camera settings. All images are resized into 32×32 and used directly without pre-processing. NORB contains stereo images, thus 2 channels. Others contain RGB images, thus 3 channels.

NORB includes 48600 pairs of stereo images from 5 generic categories under different angles, lightings and azimuths. Half is used for training and the other half for testing based on the standard partition in [15]. ETH-80 [16] contains 8 generic categories, under 41 viewing angles. Each category is equally split into training and testing sets as done in [11]. COIL [21] includes 100 objects under 72 rotated views (5° increment). The testing set includes images every 20° $(0°, 20°, \ldots)$ and is consisted of 18 views. The training set includes the remaining images. We reserve a hold-out validation set for each problem, consisting of 20% of the training set.

4.2 Influence of the number of feature maps K

In essence, the purpose of multiple feature maps is to obtain thorough representations for the raw images. The more feature maps, the more exhaustive representations. However, after the number passes a threshold, more feature maps may hurt the performance because of overfitting.

At here, we fix other parameters (receptive field 4×4, pooling size 5 and $C = 0.01$.) and vary the number of feature maps K from 10 to 100, to examine the influence of K. As in Fig. 3(a), the validation accuracy increases with more feature maps, indicating that the threshold has not been reached. Additionally, more feature maps require more training time as depicted in Fig. 3(b). Thus, we make a compromise between the accuracy and K and fix $K = 50$, though not optimal, for later experiments to reduce computations.



(a) Validation accuracy (b) Training time

Figure 3: Validation accuracy and training time with varying feature maps

Table 2: Parameter selections

Dataset	receptive field	pooling size	C	# of feature maps (fixed)
NORB	4×4	3	0.01	50
ETH-80	3×3	6	1	50
COIL	7×7	5	1	50

4.3 Parameter selection

After fixing $K = 50$, parameters to be chosen are: 1) size of receptive field; 2) pooling size; 3) value of C. And the optimal parameters will be selected by grid search based on the validation accuracy. Receptive field is tried with 5 values: 3×3, 4×4, 5×5, 6×6 and 7×7. Pooling size is also tried with 5 values: 3, 4, 5, 6 and 7. And C is tried with 3 values: 0.01, 1 and 100. Table 2 specifies the parameters for all these datasets.

4.4 Performance on NORB

The test error rates and training time of different methods on the NORB dataset are compared in Table 3 [1]. ELM-LRF achieves the *best accuracy* with much faster training speed, up to *200 times* compared with deep belief network (DBN) [20] and CNN [15].

Table 3: Test error rates and training time on the NORB dataset

Methods	Test error rates	Training time (s)
ELM-LRF	**2.76%**	**400.78**
Random weights [26]	4.8%	1764.28
K-means + soft activation [5]	2.8%	6920.47
Tiled CNN [13]	3.9%	15104.55
CNN [15]	6.6%	53378.16
DBN [20]	6.5%	85717.14

[1] In this paper, we cite the error rates of other methods from corresponding papers directly. On the contrary, the training and testing time are all recorded on our experimental platform in order to conduct fair comparisons.

Table 4: Test error rates on the ETH-80 dataset

Methods	**Test error rates**
ELM-LRF	10.0%
Discriminant Analysis of Canonical Correlations (DCC) [11]	**8.3%**
Orthogonal Subspace Method (OSM) [11]	9.5%
Constrained Mutual Subspace Method (CMSM) [22]	10.3%
kNN-LDA [3]	24.8%
kNN-PCA	23.8%

Table 5: Test error rates on the COIL dataset

Methods	**Test error rates**
ELM-LRF	**0.02%**
Local Affine Frames (LAFs) [23] [1]	0.1%
Linear SVM [29]	8.7%
Spin-Glass Markov Random Field (MRF) [17]	3.2%
Standard CNN [19]	28.51%
CNN+*video (test images of COIL)* [19] [2]	7.75%
CNN+*video (COIL-like images)* [19] [3]	20.23%

[1] The current state-of-the-art result.
[2] Use the unlabeled test images as additional learning source. It is a semi-supervised method together with the labeled training images.
[3] Use COIL-like images as additional learning source.

4.5 Performance on ETH-80

The results of ELM-LRF and some leading methods are listed in Table 4. The test error rate of ELM-LRF is comparable with state-of-the-art result achieved by DCC [11] method. And ELM-LRF is also exceptionally efficient that it only requires 48.64 seconds for training and 15.35 seconds for testing.

4.6 Performance on COIL

As seen from Table 5, ELM-LRF also sets the new record for COIL dataset. Additionally, there are some works using CNN methods to handle the COIL dataset [19]. It can be observed that ELM-LRF is quite advantageous over CNN when dealing with COIL. Even if CNN uses unlabeled test images or COIL-like images as additional information for further training and achieves significant improvements, ELM-LRF still outperforms CNN by a big gap for this task. The authors believe that it is caused by the relatively too few training samples. In CNN, numerous parameters need to be tuned. And when there are not enough training samples, the parameters (connection weights) cannot be well trained, which degenerates the generalization capability of the network.

4.7 High efficiency of ELM-LRF

Let us inspect the efficiency of ELM-LRF as a general framework.The training and testing time are summarized in Table 6. As easily observed from the table, ELM-LRF is highly efficient that it requires less than 0.03 seconds per image for training and less than 0.01 seconds per image for testing. In addition, ELM-LRF can be easily extended to real-time applications, since it is able to test more than 100 images per second after properly trained.

Table 6: Training and testing time (seconds) on different datasets

Dataset	Training stage		Testing stage	
	Total training time	Per image	Total testing time	Per image
NORB	400.78	0.0165	113.7	0.0047
ETH-80	48.64	0.0297	15.35	0.0094
COIL	33.23	0.0185	34.18	0.0063

5 Conclusions

In this paper, we propose to use ELM-LRF as a general framework for generic object recognition. Distinct merits exist for ELM-LRF compared with traditional methods: 1) task non-specific for not utilizing any task-specific information; 2) simple to use that it requires no pre-processing, like design of suitable features, shape model construction or anything else; 3) highly efficient as only a small portion of connection weights need to be calculated. Additionally, unlike the newly-emerging CNN, where connection weights are iteratively tuned, most weights in ELM-LRF are simply generated randomly and only the output weights $\boldsymbol{\beta}$ is calculated deterministically. Comparing to CNN, it significantly reduces: 1) computational complexity; 2) requirement for huge training set. In the experiments, the general framework of ELM-LRF presents superior accuracy with exceptionally high speed.

References

[1] Zuo Bai, G-B Huang, Danwei Wang, Han Wang, and M Brandon Westover. Sparse extreme learning machine for classification. *Cybernetics, IEEE Transactions on*, 44(10):1858–1870, 2014.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

[3] Marco Bressan and Jordi Vitria. Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24(15):2743–2749, 2003.

[4] C.L.P. Chen. A rapid supervised learning neural network for function interpolation and approximation. *Neural Networks, IEEE Transactions on*, 7(5):1220–1230, Sep 1996.

[5] Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

[6] Guang-Bin Huang. What are extreme learning machines? filling the gap between frank rosenblatt's dream and john von neumann's puzzle. *Cognitive Computation*, 7(3):263–278, 2015.

[7] Guang-Bin Huang, Zuo Bai, L.L.C. Kasun, and Chi Man Vong. Local receptive fields based extreme learning machine. *Computational Intelligence Magazine, IEEE*, 10(2):18–29, 2015.

[8] Guang-Bin Huang, Lei Chen, and Chee-Kheong Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17(4):879–892, 2006.

[9] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 42(2):513–529, 2012.

[10] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243, 1968.

[11] Tae-Kyun Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *Pattern Analysis and Machine Intelligence, IEEE Transactions*

on, 29(6):1005–1018, June 2007.

[12] Iasonas Kokkinos and Alan Yuille. Scale invariance without scale selection. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1–8. IEEE, 2008.

[13] Quoc V Le, Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang Wei Koh, and Andrew Y Ng. Tiled convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1279–1287, 2010.

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–97–104, 2004.

[16] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 2, pages II–409–15, June 2003.

[17] J. Matas and S. Obdrzalek. Object recognition methods based on transformation covariant features. In *Signal Processing Conference, 2004 12th European*, pages 1721–1728, Sept 2004.

[18] M. D. McDonnell and T. Vladusich. Enhanced Image Classification With a Fast-Learning Shallow Convolutional Neural Network. *ArXiv e-prints*, 2015.

[19] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744. ACM, 2009.

[20] Vinod Nair and Geoffrey E Hinton. 3D object recognition with deep belief nets. In *Advances in Neural Information Processing Systems*, pages 1339–1347, 2009.

[21] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). Technical report, 1996.

[22] Masashi Nishiyama, Osamu Yamaguchi, and Kazuhiro Fukui. Face recognition with the multiple constrained mutual subspace method. In *Audio-and Video-Based Biometric Person Authentication*, pages 71–80. Springer, 2005.

[23] Štěpán Obdržálek and Jiří Matas. Local affine frames for image retrieval. In MichaelS. Lew, Nicu Sebe, and JohnP. Eakins, editors, *Image and Video Retrieval*, volume 2383 of *Lecture Notes in Computer Science*, pages 318–327. Springer, 2002.

[24] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagation errors. *Nature*, 323:533–536, 1986.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014.

[26] Andrew Saxe, Pang W Koh, Zhenghao Chen, Maneesh Bhand, Bipin Suresh, and Andrew Y Ng. On random weights and unsupervised feature learning. In *International Conference on Machine Learning*, pages 1089–1096, 2011.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. *ArXiv e-prints*, 2014.

[28] H. White. An additional hidden unit test for neglected nonlinearity in multilayer feedforward networks. In *Neural Networks, International Joint Conference on*, volume 2, pages 451–455, 1989.

[29] Ming-Hsuan Yang, Dan Roth, and Narendra Ahuja. Learning to recognize 3d objects with snow. In *Computer Vision - ECCV*, pages 439–454. Springer, 2000.

[30] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision*, 73(2):213–238, 2007.