Deep learning vs. bag of features in machine learning for image classification

Conference Paper · March 2018

DOI: 10.1109/ASET.2018.8379825

2 authors:

Sehla Loussaief
Ecole national d'ingénieur de carthage, tunis ,tunisie
6 PUBLICATIONS 29 CITATIONS
SEE PROFILE

Afef Abdelkrim
École Nationale d'Ingénieurs de Tunis
101 PUBLICATIONS 290 CITATIONS
SEE PROFILE

Deep Learning vs. Bag of Features in Machine Learning for Image Classification

Sehla Loussaief, Afef Abdelkrim
LA.R.A, Ecole Nationale d'Ingénieurs de Tunis, Université Tunis El Manar. BP 32, le Belvédère 1002.
ENICarthage, Université de Carthage, 35 rue des Entrepreneurs. Charguia II.
Tunis, Tunisie.
sehla.Loussaief@enicarthage.rnu.tn, afef.a.abdelkrim@ieee.org

Abstract— **The main issue in computer vision and notably image classification problems is image feature extraction and image encoding. Here we show and compare two approaches to solve this problem: the first approach uses the Bag of Features (BoF) paradigm. The second one is based on deep learning and especially Convolutional Neural Networks (CNN). Specifically, we use the "Caffe" CNN model trained to perform well on the ImageNet dataset. Our results shed light on how the use of CNN is more performant than the BoF in the process of feature extraction in a machine learning framework for image classification. This performance is shown by a series of experimentations that we carried out using the Caltech dataset and many classifier algorithms.**

Keywords— ***computer vision; image classification; feature extraction; machine learning; bag of features; deep learning; convolutional neural network***

I. INTRODUCTION

The past decade has seen the rise of different approaches for pattern-recognition task in computer vision. A study has enabled us to identify two methods: the Bag of Features (BoF) and the deep learning technique with the use of convolutional neural network (CNN).

The BoF methods have been applied to image classification, object detection, image retrieval, and even visual localization for robots. BoF approaches are characterized by the use of an orderless collection of image features. Lacking any structure or spatial information, it is perhaps surprising that this choice of image representation would be powerful enough to match or exceed state-of-the-art performance in many of the applications to which it has been applied. Due to its simplicity and performance, the Bag of Features approach has become well-established in the field.

CNN used in deep learning learn hierarchical layers of representation from input in order to perform image classification [1-2]. Recently, these deep architectures have demonstrated impressive, state-of-the-art, and sometimes human-competitive results on many pattern recognition tasks, especially vision classification problems [3-5].

We focus on the application of BoF and Deep Learning to image classification task. As discussed above, to deploy a machine learning framework for image classification we can use either the CNN capabilities or the BoF approach for feature extraction and image encoding. The feature vectors are fed to different classifiers. The purpose of our research is to experiment and compare the performance of classifier algorithms when we use these technics. We held experimentation with the Caffe CNN which is a well-trained CNN on ImageNet dataset, and as input to our image classification framework we use the Caltech dataset.

This paper seeks to compare the Bag of Features and deep learning paradigms, providing a survey of relevant literature and an experimental evaluation of their performance in the context of an image classification problem.

This paper is organized as follows. Section II provides an overview of image feature extraction deployed in machine learning framework. Section III provides details on the feature detection and extraction techniques commonly employed in BoF and deep learning approaches. Section IV looks at the evaluation of BoF and deep learning methods, including common performance measures, data sets, and challenges involved with comparative evaluation. The last section includes our concluding remarks.

II. IMAGE FEATURE EXTRACTION IN MACHINE LEARNING

When the input data to an algorithm is too large to be processed and is suspected to be redundant, then it can be transformed into a reduced set of features (also named a feature vector). This process is called feature extraction.

The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

Selecting suitable variables is a critical step for successfully implementing an image classification. The use of too many variables in a classification procedure may decrease classification accuracy. It is important to select only the variables that are most useful. That's why in machine learning and in image processing, feature extraction starts from an initial set of measured data and builds derived values (features). These values are intended to be informative, non-redundant, facilitating the subsequent learning steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction.

Feature extraction techniques have been widely used in the literature to reduce the data dimensionality. While the classical Principal Component Analysis (PCA) [6] is still one of the most popular choices, a plethora of non-linear dimensionality reduction methods have been introduced in the last decade.

In section III we expose the two investigated methods for image feature extraction which are Bag of Features and deep learning.

III. Methods

This section introduces the concepts and strategies employed to image feature extraction. In III.A, we expose the Bag of Features paradigm. In III.B, we outline the strategy to learn features by the use of Convolutional Neural Network.

A. Bag of Features image representation

A Bag of Features method represents images as orderless collections of local features. Features are extracted from training images and vector quantized to develop a visual vocabulary. A visual vocabulary is constructed to represent the dictionary by clustering features extracted from a set of training images. The image features represent local areas of the image. Novel image's features are assigned to the nearest code in the codebook. The image is reduced to the set of codes it contains, represented as a histogram. The normalized histogram of codes is exactly the same as the normalized histogram of visual words used in document classification. Both "codebook" and "visual vocabulary" terminology is present in the surveyed literature. The BoF term vector is a compact representation of an image which discards largescale spatial information and the relative locations, scales, and orientations of the features [7].

At a high level, the procedure for generating a Bag of Features image representation is shown in Fig. 1 and summarized as follows: (1) Build visual vocabulary: Extract features from all images in a training set. Then cluster, these features into a "visual vocabulary," where each cluster represents a "visual word" or "term." In some works, the vocabulary is called the "visual codebook." Terms in the vocabulary are the codes in the codebook. (2) Assign Terms: Extract features from a novel image. Use Nearest Neighbors or a related strategy to assign the features to the closest terms in the vocabulary. (3) Generate Term Vector: Record the counts of each term that appears in the image to create a normalized histogram representing a "term vector." This term vector is the Bag of Features representation of the image.

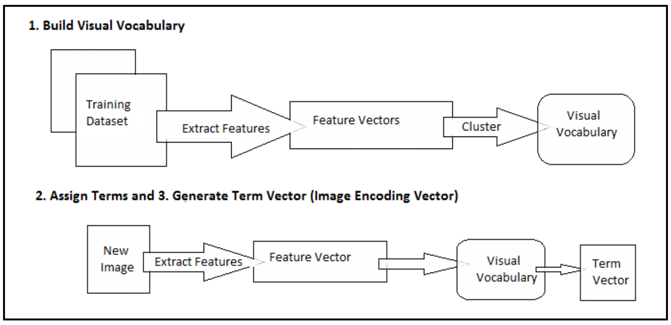

Fig. 1: BoF image representation process

Term vectors may also be represented in ways other than simple term frequency. There are a number of design choices involved at each step in the BoF representation.

One key decision involves the choice of feature detection and representation.

For features detection and extraction, we use the Speed Up Robust Features (SURF) method. It extracts salient features and descriptors from images.

This extractor is preferred over Scale-Invariant Feature Transform (SIFT) due to its concise descriptor length. In SURF, a descriptor vector of length 64 is constructed using a histogram of gradient orientations in the local neighborhood around each key-point [8].

In the clustering step, we use the K-means algorithm. It is selected over Expectation Maximization (EM) to group the descriptors into N visual words and to construct the visual vocabulary as experimental methods have verified the computational efficiency of K-means as opposed to EM [9].

B. Deep Learning: Convolutional Neural Network

Deep neural networks are models that capture hierarchical representations of data. These models are based on the sequential application of a computation "module", where the output of the previous module is the input to the next one; these modules are called layers. Each layer provides one representation level. Layers are parameterized by a set of weights connecting input units to output units and a set of biases. In the case of Convolutional Neural Networks (CNN), weights are shared locally, i.e. the same weights are applied at every location of the input. The weights connected to the same output unit form a filter. CNN layers consist of: (1) a convolution of the input with a set of learnable filters to extract local features; (2) a point-wise non-linearity, e.g. the logistic function, to allow deep architectures to learn non-linear representations of the input data; and (3) a pooling operator, which aggregates the statistics of the features at nearby locations, to reduce the computational cost (by reducing the spatial size of the image), while providing a local translational invariance in the previously extracted features. The last convolutional layer is followed by a fully-connected output layer [10]. Fig. 2 illustrates a graphical representation of the used deep convolutional architecture [11].

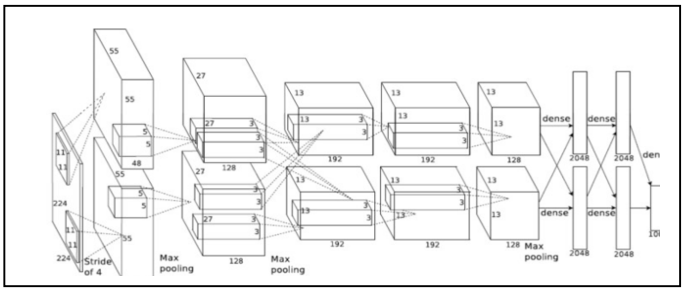

Fig. 2: ImageNet CNN layers

Fig. 2 demonstrates the different network layers required by the ImageNet Convolutional Neural Network. There are 5 convolution and 3 fully connected layers. These layers occupy more than 99% of the processing time for this network. There are 3 different filter sizes for the different convolution layers, 11×11, 5×5 and 3×3. The operations performed in a single convolutional layer can be summarized as:

$$O^l = pool_P\big(\sigma(O^{l-1} \star W^l + b^l)\big) \qquad (1)$$

Where O^{l-1} is the input feature map to the *l-th* layer; $\theta^l=\{w^l, b^l\}$ is the set of learnable parameters (weights and biases) of the layer, *σ(.)* is the point-wise non-linearity, pool is a subsampling operation, *P* is the size of the pooling region, and the symbol ⋆ denotes linear convolution. Note that in the context of CNN, the convolution is multi-dimensional with each filter. The input of the first layer is the input data, in this case a multi/hyper-spectral image, i.e. $O^o=I$, where $\mathrm{I} \in \mathbb{R}^{R^o x C^o x N_h^o}$ is the input image, R^o and C^o are its width and height N_h^o is the number of spectral channels (bands).

The pooling region is usually square, in this case formed by *P*×*P* pixels.

CNN architectures have a significant number of meta-parameters. The most relevant ones may be: (1) the number of layers; (2) the number of outputs per layer; (3) the size of the filters, also called receptive field; and (4) the size and type of spatial pooling.

Another important aspect is how to train such architectures. Deep convolutional networks can be trained in a supervised fashion, e.g. by means of standard back-propagation [12-13], or in an unsupervised fashion, by means of greedy layer-wise pre-training [14].

IV. EXPERIMENTAL RESULTS

This section is devoted to illustrate the capabilities of the presented feature extraction approaches in different scenarios of image classification.

It is devoted to present a series of results of image classification based on the BoF and CNN strategies. Our results are reported on Caltech101[1] image dataset to which we have added some new images of existing categories.

For all experiments, we run 10 times and the average results were presented. Here we are interested in measuring the classifier accuracy.

A. BoF experimental environment

The deployed BoF environment shown in Fig. 3, includes the use of the SURF technique as a feature extractor and image encoder. This step returns a big number of image feature vectors.

In the clustering step we use the K-means algorithm. It aims to reduce the number of feature vectors. Only cluster centers will be considered as the visual words of vocabulary. BoF encoder step aims to encode each input image of the training dataset into histogram of visual words frequency. Image encoding vectors are than fed into classifiers.

B. CNN experimental environment

To investigate the CNN capabilities in image feature extraction for classification purpose, we choose the well-known AlexNet architecture [15], which is a convolutional neural network trained on the 1.3-million-image ILSVRC 2012 ImageNet dataset [16-17].

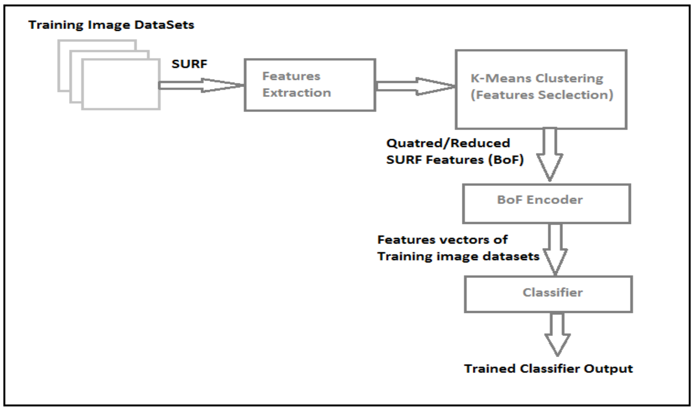
Training Image DataSets
SURF
Features Extraction
K-Means Clustering (Features Seclection)
Quatred/Reduced SURF Features (BoF)
BoF Encoder
Features vectors of Training image datasets
Classifier
Trained Classifier Output

Fig. 3: Image classification process

This already-trained AlexNet CNN is provided by the Caffe software package [18].

We chose AlexNet because it is widely known and publicly available. We conduct experiments with the Caffe provided model trained on the ImageNet dataset. The Caffe version has a minor difference from the original architecture in [18] that its neural activation functions are rectified linear units (ReLUs) [19] instead of sigmoids.

C. Results and discussion

For the performance measurement of our feature extraction techniques we use the Caltech101[1] dataset to which we add some images.

1) Scenario 1

Here, an approach for classification by a Linear SVM analysis is evaluated with particular regard to the effect of training set size on classification accuracy. The image feature vector size generated by our pre-trained CNN model is equal to 4096. To ensure the same dimension of image encoding vector, we chose the same value as visual dictionary size in the BoF assessment. Thus, in our machine learning framework each image is encoded with a vector of 4096 size.

The performance measures show that deep learning extractor and encoding technique is more robust compared to the increase of the size of the image dataset.

As presented in Fig. 4, experimentations indicate that the SVM classifier's accuracy significantly degrades with the increase of categories number when we use the BoF approach. In contrast the SVM classifier maintain a better accuracy when we use the deep learning extractor and encoding technique (>90%), as shown in Fig. 5.

With 12 categories in image dataset we notice that the SVM accuracy is 30% better when images are encoded based on the CNN technique.

2) Scenario 2

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

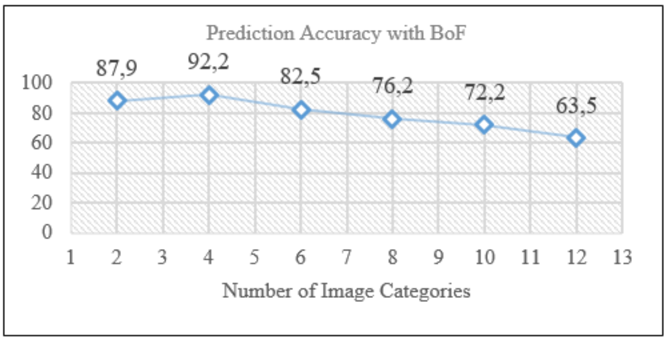

Fig. 4: Classification prediction accuracy with BoF technique

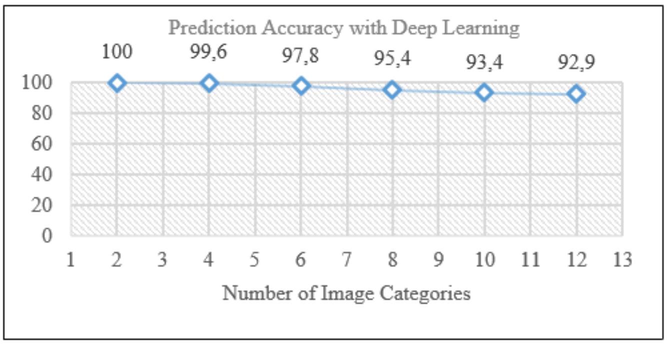

Fig. 5: Classification prediction accuracy with Deep Learning technique

Next, we fix the category number in the dataset and evaluate the accuracy of our machine learning regarding the classifier's type. In these experimentations we use the SVM, KNN and Decision Trees family.

a) Support Vector Machine Classification: Here we are interested in evaluating the accuracy of support vector machine (SVM) classifiers based on BoF or Deep Learning technique. More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. To reduce the multiclass classification problem to a set of binary (two class) classification sub-problems, with one SVM learner for each sub-problem we can use different Kernel function to compute the classifier [20]: Linear kernel, Gaussian kernel, Quadratic and Cubic.

Measurements, presented in Fig. 6, show that the use of deep learning gives a better prediction accuracy. Among the SVM classifier the Quadratic SVM gives a superior result either with the deployment of the BOF technique or that of the CNN based Deep Learning.

b) K-Nearest Neighbors Classification: K-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the most common class among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. KNN-based algorithms differ in the number of nearest neighbors to find for classifying each point when predicting and the distance metric used [21].

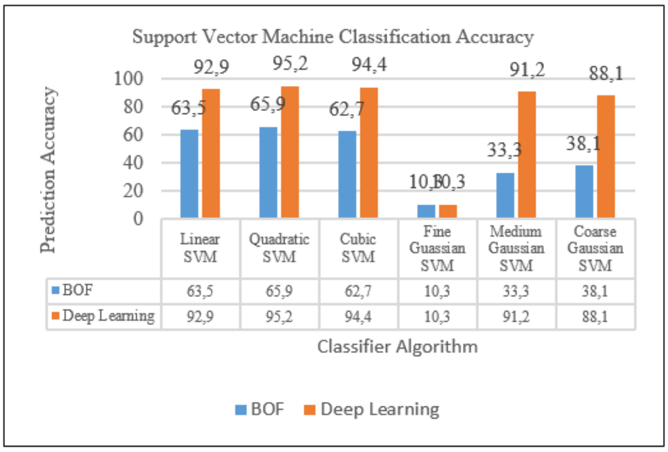

Fig 6: SVM Classifiers accuracies

KNN classifiers investigation, displayed in Fig. 7, confirm that the CNN based deep learning approach provide significantly better results than those based on the BoF mechanism.

It is shown that the Cosine KNN is the most precise during category prediction process.

c) Decision Trees Classification: Decision tree learning uses a decision tree as a predictive model which maps observations about an item (represented in the branches) to conclusions about the item's target value represented in the leaves [21].

Fig. 8 illustrates the prediction accuracy achieved through the use of likhood decision trees classifiers.

Tests demonstrate that the CNN based Deep Learning technique is more accurate.

Measurements show that the Medium Tree which uses medium number of leaves for finer distinctions between classes

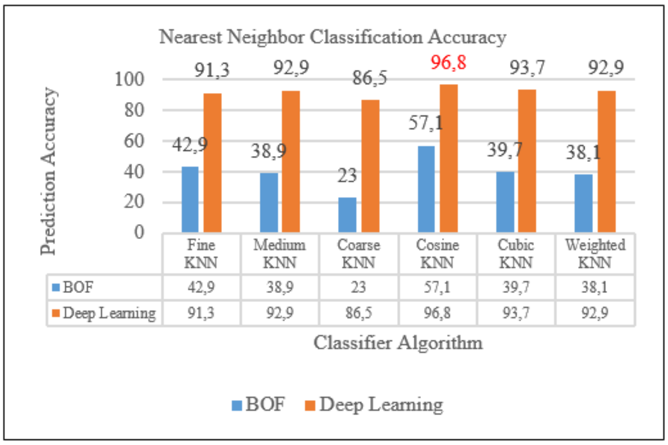

Fig. 7: KNN classifiers accuracies

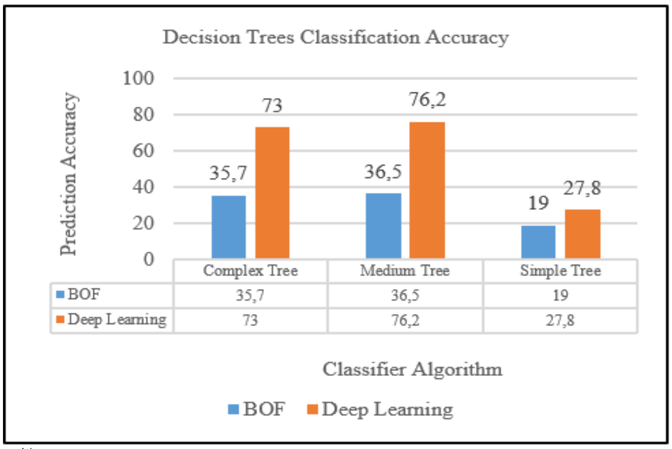

Fig. 8: Decision trees classifiers accuracies

and number of splits not exceeding a maximum of 20, gives better results than the complex and simple Tree ones.

V. Conclusion

Feature extraction is the fact of transforming the input data into the set of features. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

In this paper we related two feature extractor and image encoding techniques which are Bag of Features and Convolutional Neural Network based Deep Learning. Convolutional neural networks and BoF methods are investigated with results on Caltech101 Dataset. In BoF analysis, we choose a visual vocabulary size equal to 4096 to ensure the same features vector size as used by the pre-trained ImageNet Caffe CNN. State-of-the-art image classification algorithms are used to evaluate the two extractor feature approaches. Results show the highest classification accuracy for the Deep Learning approach in comparison with Bag of Features paradigm. Even if for both feature extractor techniques, classification accuracy was related with the size of the training set, the CNN based technique outperform the BoF one. Following the increase in categories number to 12, the Linear SVM classification performance decreased by 20% in the case where BoF was used as feature extracting approach and only by 0.08% with the use of CNN based deep learning. Research carried out in this paper demonstrate that the use of deep learning ImageNet Caffe CNN for image feature extraction has led to significant gains in accuracy. This result highlights that features extracted by CNN on the image training dataset are highly relevant. Results reveal that the trained networks are very effective at encoding information of the images.

In addition, experiments show that the most accurate classifications are derived from the KNN approach (Cosine KNN, 96.8%) followed by the quadratic SVM with 95.2%. The decision trees derived algorithms don't yield accurate classification (max=76.2%). The large performance gap between these two families of approaches make the BoF image encoding technique useless. That's why Deep Learning based classification has increasingly become important approach for data classification.

References

[1] Y. Bengio, "Learning deep architectures for ai". Foundations and trends R in Machine Learning, 2(1): pp. 1-127, 2009.

[2] G. E. Hinton, "Learning multiple layers of representation". Trends in cognitive sciences, 11(10): pp. 428-434, 2007.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks". In Advances in neural information processing systems, pp. 1097-1105, 2012.

[4] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Contextdependent pre-trained deep neural networks for largevocabulary speech recognition". Audio, Speech, and Language Processing, IEEE Transactions on, 20(1): pp.30-42, 2012.

[5] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification". In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 1701-1708. IEEE, 2014.

[6] I. Jolliffe, "Principal component analysis". Springer, 2002.

[7] S. O'hara and B A. Draper. "Introduction to the bag of features paradigm for image classification and retrieval". arXiv:1101.3354v1 [cs.CV] 17 January 2011.

[8] D.Lowe, "Towards a computational model for object recognition in IT cortex". Proc. Biologically Motivated Computer Vision, p. 2031, 2000.

[9] D. G.Lowe, "Distinctive image features from scale-invariant keypoints". IJCV, 60(2): pp. 91-110, 2004.

[10] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview. Neural Networks", Volume 61, pp. 10-28, January 2015.

[11] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks". Advances in Neural Information Processing Systems 25, 2012.

[12] Y. LeCun, L. Bottou, G. Orr, and K. Muller, "Efficient backprop," in ¨ Neural Networks: Tricks of the Trade. Springer Berlin, 1998, pp. 9-50.

[13] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in International Conference on Learning Representations (ICLR2014). CBLS, April 2014.

[14] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol. 18, no. 7, pp. 1527-1554, July 2006.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In Advances in neural information processing systems, pp. 1097-1105, 2012.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei, "Imagenet: A large-scale hierarchical image database". In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 248-255, 2009.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla and M. Bernstein, "Imagenet large scale visual recognition challenge". arXiv preprint arXiv:1409.0575, 2014.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding". arXiv preprint arXiv:1408.5093, 2014.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines". In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807-814, 2010.

[20] K. Crammer, Koby and Y.Singer, "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines" (PDF). Journal of Machine Learning Research. 2: pp. 265-292, 2001.

[21] P. Hall, B. Park and RJ. Samworth, "Choice of neighbor order in nearest-neighbor classification". Annals of Statistics. 36 (5): pp. 2135-2152. doi:10.1214/07-AOS537, 2008.

[22] L. Rokach and O. Maimon, "Data mining with decision trees: theory and applications". ISBN 978-9812771711, 2008.