

# The Investigation of Deep Data Representations Based on Decision Tree Ensembles for Classification Problems

Pavel Druzhkov and Valentina Kustikova

Lobachevsky State University of Nizhni Novgorod,  
Nizhni Novgorod, Russian Federation  
{itlab.ml@cs.vmk.unn.ru}

**Abstract.** A classification method based on deep representation of input data and ensembles of decision trees is introduced and evaluated solving the problem of vehicle classification and image classification with large number of categories.

**Keywords:** classification; decision trees; random forest, deep learning.

## 1 Introduction

*Deep learning* studies machine learning algorithms that enable automatic construction of feature description hierarchies thus making data representations more efficient in terms of the solved problem. Currently, the deep learning methods for solving image classification and object detection problems are making good headway (see, for example, a review of [1] and [2]) as a part of neural network (NN) models.

Although the deep learning ideas are for now best reflected in NN approaches, the integration of hierarchical data representation into other methods such as decision trees and bag-of-words is of special interest, because these combinations may help to reduce complexity of learning and/or the use of deep models and give a better understanding of their success. There were attempts to use decision trees for induction of new feature descriptions [3, 4]. Nevertheless, no examples of successful applications of such models stacks are known to the authors. This paper studies applicability of a deep classification method based on hierarchical representation of data obtained using decision tree ensembles.

## 2 The Proposed Classification Method

*The decision tree* divides the feature space into non-overlapping areas, corresponding to tree nodes, by means of a recursive procedure [5]. Given a data point decision tree performs a sequence of tests determining the way from the root to one of its leaves, that in turn defines the classification result. To ensure better statistical stability of the classifier, such methods for grouping the set of

trees as *boosting* [6] and *bagging* [7] are used. The output of a trees ensemble depends on the leaves that receive a certain object. In this context, Vens et al. [3] supposed that the set of tests performed for a certain data point could represent a better object view. Martyanov et al. [4] review different ways to use the internal structure of the random forest model [7] to construct efficient features. These ideas are further developed in this paper and a classification method based on hierarchical (deep) data representation is proposed. This method consists of the following stages:

1. Initial data representation is used to construct the ensemble of trees classifier.
2. Each tree in the ensemble defines a new feature whose value equals to the index of the leaf receiving the object in question.
3. New ensemble is constructed based on the derived features.
4. Steps 2 and 3 are repeated to ensure the required model depth.
5. The resulting feature descriptions are used to construct the classifier.

A possible advantage of the proposed method is that it can operate arbitrary initial feature descriptions. Therefore, using ones, that have been already successfully applied for computer vision purposes, one can improve both the classifier performance and the model quality. Deep model learning using this method can be based on widely known algorithms that are less complex compared to NN learning methods. The same is true for the use of a tuned model in the classification process, especially in case of cascaded classifiers enabling computation of feature values only when required.

Open-source C++ implementation of the proposed method utilizing random forest as a decision tree ensemble component was done [8]. Implementation allows an arbitrary initial feature description, however HOG [9] was used in all experiments described below.

### 3 Experimental Results

#### 3.1 Vehicle Classification

To put the proposed classifier to the test the vehicle classification problem was solved. The MIT2 dataset [10] from [11] was used. It contains images of four classes: *sedan*, *van*, *taxi* and *car*. Because of the essential classes overlap only two-class problems were considered: *car* vs *van* and *sedan* vs *taxi*. The training set consist of 200 images while the test one includes 730. Sizes of objects in the dataset vary considerably with the average of  $72 \times 38$  px. To properly apply different machine learning techniques, images were cropped and scaled to make all vehicles position and size identical.

It can be seen from the derived results (Table 1) that the proposed method has almost the same accuracy as (shallow) NN approaches: 97.5% and 97.75% correspondingly. According to Ma et al. [11], the best known result for the same data is 98.5% that is achieved by a Bayesian classifier applied to SIFT-like features. However this approach requires a complex parameter selection procedure.

In the absence of an open implementation nor a more detailed description it becomes even more tangled. Repeatability activities to obtain the results from [11] provided the accuracy of only 79.25% [12].

In case of less significant cross-class distinctive features the NN approaches showed a much lower classification accuracy than the proposed algorithm and the Bayesian classifier. The accuracy of convolutional NN on *sedans vs taxis* task is 36% compared to 94.24% and 96.06%, respectively (Table 1).

**Table 1.** Classification accuracy for MIT2 dataset

| <i>Classifier</i>                                       | <i>car vs van, %</i> | <i>sedan vs taxi, %</i> |
|---|----------------------|-------------------------|
| Linear NN: fully-connected and softmax layer            | 97.75                | 0                       |
| Convolutional NN: convolutional and softmax layer       | 50.3                 | 36.00                   |
| Bayesian classifier on SIFT-like features [11]          | 98.5                 | 96.06                   |
| Bayesian classifier on SIFT-like features [12]          | 79.25                | —                       |
| Proposed method: 3 levels, 500 trees of depth 2 at each | 97.5                 | 94.24                   |

### 3.2 Image Classification with Large Number of Categories

To expect how the proposed method scales with the growth of classes count, experiments on CIFAR-100 dataset [13] were carried out. Dataset contains 60,000  $32 \times 32$  px images. Each image belongs to one of the 100 classes that cover a wide range of visual objects. The whole set of images is divided into the learning (50,000 images) and test (10,000 images) parts.

Although the random forest, which is at the heart of the proposed method, is known as an algorithm that does not require careful selection of learning parameters, the optimal number of trees and their depth were determined on the basis of OOB accuracy [7] to find balance between model quality and size. Hyperparameters were selected in a greedy fashion: ones for the first layer were selected and fixed, then the second layer is explored.

For the first layer the OOB accuracy of 15.16% is obtained for 500 trees with the depth of 20. This is close enough to the observed maximum of 16.04% that corresponds to a much larger ensemble of 1000 trees. But this requires greater computational resources for learning and classification purposes and may also hinder learning of the second layer due to increased dimension of the new feature space. So 500 trees of depth 20 were fixed for the first layer setup. Achieved test accuracy for the one-layer model is 16.97%, while additional layer (1000 trees of depth 25) reduced it to 12.42%. It is much lower than the best known results of deep NN models for the same problem reaching 61–65% [14]. Such a gap may be explained by the nature of the solved problem, properties of the proposed algorithm and appropriateness of HOG description to which the proposed algorithm was applied.

Insufficient quality combined with uselessness of model depth increment for the considered problem indicates that the approach needs to be modified. The possible ways of improvement include the use of boosting instead of the random forest, replacement of the initial features or use of image patches of different size on different layers, to make connection between model layers and feature abstraction levels more explicit. Higher resolution images may be subject to quality improvement due to the use of keypoint descriptors by analogy with [4].

## 4 Conclusion

In this paper, a classification method based on hierarchical (deep) initial data representation was proposed. This method was tested by solving the vehicle classification problem and large scale image classification.

**Acknowledgments.** This work is supported by Russian Foundation for Basic Research (project No 14-07-31269) and has been done in the IT laboratory at CMC Department of Lobachevsky State University of Nizhni Novgorod.

## References

1. Kustikova, V.D., Druzhkov, P.N.: A Survey of Deep Learning Methods and Software for Image Classification and Object Detection. In: Proc. of OGRW. (2014)
2. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. In: arXiv:1409.0575. (2014)
3. Vens, C., Costa, F.: Random Forest Based Feature Induction. In: Proc. of ICDM. pp. 744-753. (2011)
4. Martyanov, V., Polovinkin, A., Tuv, E.: Feature Learning for Image Classification with Code Book Based on Decision Tree Ensembles. In: Proc. of ICIM. pp. 188-191. (2013)
5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth & Brooks. (1984)
6. Freund, Y., Schapire, R.: A decision-theoretic generalization of online learning and an application to boosting. In: Comp. and Sys. Sciences. Vol. 55. pp. 119-139. (1997)
7. Breiman, L.: Random Forests. Machine Learning. Vol. 45. No. 1. pp. 5-32. (2001)
8. Deep random forest implementation in SWOD (Sliding Window Object Detection) library, [https://github.com/druzhkov-paul/swod/tree/deep\\_random\\_forest](https://github.com/druzhkov-paul/swod/tree/deep_random_forest)
9. Dalal, N.: Finding people in images and videos. (2006)
10. Vehicle Classification in Surveillance (MIT2), [http://people.csail.mit.edu/xiaoxuma/proj/vehi\\_reco/index\\_vehi\\_reco.htm](http://people.csail.mit.edu/xiaoxuma/proj/vehi_reco/index_vehi_reco.htm)
11. Ma, X., Grimson, W.E.L.: Edge-based rich representation for vehicle classification. In: Proc. of the 10th IEEE Int. Conf. on CV (ICCV05). Vol. 2. pp. 1185-1192. (2005)
12. Khanova, T.A., Kustikova, V.D.: One approach to solving the problem of vehicle classification. In: Proc. of HPC-2014. pp. 453-459. (2014) (In Russian)
13. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. (2009), <http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>
14. Classification datasets results, [http://rodrigob.github.io/are\\_we\\_there\\_yet/build/classification\\_datasets\\_results.html](http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html)