New Object Detection Features in the OpenCV Library[1]

P. N. Druzhkov, V. L. Erukhimov, N. Yu. Zolotykh, E. A. Kozinov, V. D. Kustikova, I. B. Meerov, and A. N. Polovinkin

Lobachevskii State University of Nizhni Novgorod, pr. Gagarina 23, Nizhni Novgorod, 603950 Russia
e-mail: druzhkov_paul@mail.ru, victor.eruhimov@itseez.com, evgeniy.kozinov@gmail.com, valentina.kustikova@gmail.com, mib@uic.nnov.ru, alexey.polovinkin@gmail.com, nikolai.zolotykh@gmail.com

Abstract—In this work the object detection problem is considered. A short description of implementations of the object detection system with a discriminatively trained part based model and a gradient boosting trees algorithm (as part of OpenCV library) is given. Application of the gradient boosting trees learner to the object detection problem (in terms of the pedestrian detection problem) is explored.

DOI: 10.1134/S1054661811020271

INTRODUCTION

In this paper we consider an object detection problem, which is currently one of the most actively researched topics in the field of computer vision. The problem can be stated as follows: "Where are the instances of a particular object class in the image?"

One of the most common approaches to this problem is to slide a window across the image at multiple scales and to classify each such local window as containing the target or background. This approach was extended in so-called deformable part models, where object models are represented by a collection of parts arranged in a deformable configuration.

The main purpose of this work is high performance, open source implementation of two state-of-the-art algorithms for object detection and general machine learning problems: discriminatively trained part based models [4] and gradient boosting trees [6], respectively. The accuracy of the implemented algorithms has been evaluated on real world applications (PASCAL Visual Object Classes (VOC) challenge data [9], UCI [7], Daimler pedestrian detection benchmark [5]). The implementation has been integrated into one of the most popular open-source computer vision libraries.

OPENCV LIBRARY OVERVIEW

OpenCV [11] is an open source computer vision library. The library contains optimized implementation more than 500 optimized algorithms: general image processing functions, image pyramids, geometric descriptors, segmentation, transforms, features extraction, tracking, fitting, camera calibration, stereo vision, 3D, and machine learning: detection and recognition, matrix math, and auxiliary data structures and functions. The library has found use in different fields from interactive art to robotics. The current version of the library supports all popular OS: Windows, Linux, Mac OS; an Android version also exists.

OBJECT DETECTION WITH DISCRIMINATIVELY TRAINED PART BASED MODELS

In this section we describe the implementation of an object detection system based on mixtures of multiscale deformable part models, which relies on a new method for discriminative training with partially labeled data called Latent SVM [6] by the authors. Actually, Latent SVM is a reformulation of a multiple-instance SVM (support vector machine [10]) in terms of latent variables. The object model is represented by a set of components each of which contains a root filter (describes object in common), part filters (describe separate parts of object), "anchor" positions for parts relative to the root position, and coefficients of a quadratic functions defining a deformation cost for each possible placement of the part relative to the anchor position.

The object detection algorithm itself includes the following steps [6]:

—Feature pyramid computing (it describes different resolution levels of the original image with a set of histogram of oriented gradient (HoG) features [4]).

—Matching of the deformable model to an image by finding "maximum-score" locations. Scores (likelihoods) in different positions and scales in the image are estimated by applying linear filters to the feature pyramid and taking into account deformation cost for each possible placement of the object part.

In this work we implemented the full set of utilities which allows utilizing the object detection algorithm on arbitrary data:

—MATLAB script for converting the learned model from MAT to XML format.

[1]This article was translated by the authors.

Received December 29, 2010

—Parser for reading the learned model from XML format.

—Object detection algorithm (both feature pyramid computing and matching procedures).

—Samples.

Detection accuracy comparison with the original MATLAB based implementation on VOC 2007 data is shown in Table 1 (average precision measure is used; results are given for the first five classes of objects).

So the current implementation gives the same accuracy as the original, but usage of the original version is limited due to the need to have additional paid software (MATLAB with extension packages). Our implementation is written in the widely used C++ programming language and integrated into one of the most popular open source computer vision libraries.

Table 1. Average precision on VOC 2007 data

Class	Original	OpenCV
Aeroplane	0.282	0.277
Bicycle	0.525	0.54
Bird	0.007	0.005
Boat	0.125	0.133
Bottle	0.261	0.262

Table 2. CV-error on UCI data (classification problems)

Dataset	GBT	RT	XRT	SVM
Agaricus lepiota	0	0	0	0
Liver disorders	0.251	0.228	0.254	0.279
Car	0	0.036	0.039	0.051

GRADIENT BOOSTING TREES

Gradient boosting trees is a serial ensemble of decision trees [8] where every new tree constructed relies on previously built trees. At each iteration of GBT, a new tree is fitted to the generalized residuals with respect to a loss function, where the size of the ensemble is chosen to avoid overfitting. The GBT algorithm has all the properties of a universal learner: fast, works with mixed-type data, elegantly handles missing data, invariant to monotonic transformations of the input variables (and, therefore, resistant to outliers in input space). GBT has proven to be among the most accurate and versatile state-of-the-art learning.

Up to now universal open source implementation of GBT has not been known. The major part of available versions is limited to the small size of processed data or supported classes of the problem (regression only).

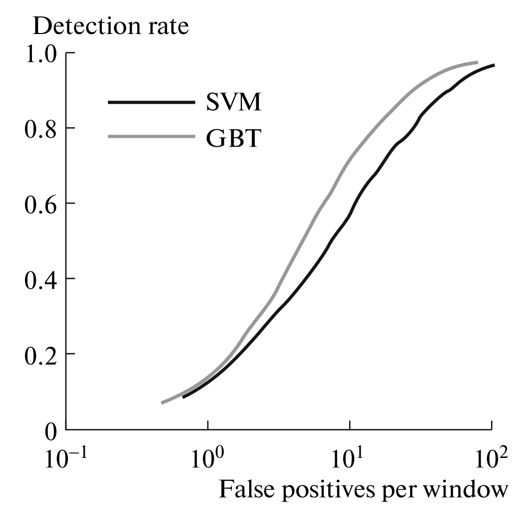
In this work we developed a fast and flexible version of the considered algorithm. It uses OpenCV decision tree implementation, which allows us to effectively operate with mixed-type data (both categorical and numerical) and handle missing values. We support both regression and classification problems, most loss functions, and some additional features for improving model quality: shrinkage heuristic and subsampling.

Prediction accuracy of GBT has been examined and compared with other well-known learners on several UCI [7] datasets. Cross-validation errors on part of them (10-fold CV was used) are shown in Table 2 (GBT (gradient boosting trees), RT (random forest) [3], XRT (extremely randomized trees) [4], SVM (support vector machine) [1, 2]).

We investigated the usage of the implemented learner for pedestrian detection task. We follow the approach of Dalal and Triggs [4] to model an object with well-normalized dense histograms of gradient orientation, and the resulting feature vector is subject to classification using a linear SVM. We suggest replacing the support vector machine classifier with gradient boosting trees and compared the modified implementation with the original.

The train part of the explored dataset consists of 15 660 pedestrian samples (in two resolutions each one: 48×96 (medium size) and 18×36 pixels (small size)) and 6744 full images (640×480 pixels) which do not contain any pedestrian objects. The test part of the dataset consists of 21 790 full images with 56 492 manual labels captured from a moving vehicle through urban traffic.

We learned classifiers (linear SVM with $C = 0.01$ and GBT with 2500 trees of depth 3, shrinkage = 0.1) on features extracted from 15660 "positive" images of medium size and 15660 "negative" images of the same size randomly sampled from the full "negative" images.



Detection rates for GBT and SVM algorithms in pedestrian detection task.

The following detection algorithm operates in classical sliding window fashion: detector windows are both shifted through scale and through a location at fractions of the base detector window size (so about 70 000 different subimages per image are examined).

The results are shown in the figure. As can be seen, the accuracy of the algorithm that uses GBT as a classifier outperforms the accuracy of the linear SVM version.

CONCLUSIONS

In this work we extended the functionality of one of the most popular open-source computer vision libraries with two state-of-the-art algorithms for object detection and general machine learning problems. The accuracy of the implemented methods was explored on some world-known benchmarks. In addition new modification of one of the state-of-the-art methods for the object detection problem using the implemented gradient boosting trees classifier was examined.

ACKNOWLEDGMENTS

This research was supported by the Russian Federal Program "Scientists and Educators in Russia of Innovations," contract no. 02.740.11.5131.

REFERENCES

1. V. N. Vapnik, *Relationships Restoration According to Empirical Data* (Nauka, Moscow, 1979) [in Russian].
2. B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proc. 5th Ann. ACM Workshop on COLT* (ACM, Pittsburgh Press, 1992), pp. 144–152.
3. L. Breiman, "Random Forests," Mach. Learn. **45** (1), 5–32 (2001).
4. N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition* (San Diego, 2005), pp. 886–893.
5. M. Enzweiler and D. Gavrila, "Monocular Pedestrian Detection: Survey and Experiments," IEEE Trans. Pattern Anal. Mach. Intell. **31** (12), 2179–2195 (2009).
6. P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models," IEEE Trans. Pattern Anal. and Mach. Intell. **32** (9), 1627–1645 (2010).
7. A. Frank and A. Asuncion, *UCI Machine Learning Repository* (Univ. of California, School of Information and Computer Sci., 2010), Available from: http://archive. ics.uci.edy/ml
8. J. H. Friedman, "Greedy Function Approximation: a Gradient Boosting Machine," Ann. Stat. **29** (5), 1189–1232 (2001).
9. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," Int. J. Comp. Vision **88** (2), 303–338 (2010).
10. P. Geurts, D. Ernst, and L. Wehenkel, "Extremely Randomized Trees," Mach. Learn. **36** (1), 3–42 (2006).
11. http://openev.willowgarage.com/wiki