ARTOS
Adaptive Real-Time Object Detection System

Björn Barz
BJOERN.BARZ@UNI-JENA.DE

Erik Rodner
ERIK.RODNER@UNI-JENA.DE

Joachim Denzler
JOACHIM.DENZLER@UNI-JENA.DE

Computer Vision Group
Friedrich Schiller University of Jena
Jena, Germany

Abstract
ARTOS is all about creating, tuning, and applying object detection models with just a few clicks. In particular, ARTOS facilitates learning of models for visual object detection by eliminating the burden of having to collect and annotate a large set of positive and negative samples manually and in addition it implements a fast learning technique to reduce the time needed for the learning step. A clean and friendly GUI guides the user through the process of model creation, adaptation of learned models to different domains using in-situ images, and object detection on both offline images and images from a video stream. A library written in C++ provides the main functionality of ARTOS with a C-style procedural interface, so that it can be easily integrated with any other project.

Keywords: object detection, efficient learning, large-scale image databases

1. How does ARTOS work?
Object detection is often one of the basic algorithms necessary for a lot of vision applications in robotics or related fields. Our work is based on the ideas of several state-of-the-art papers and the setup described in Göhring et al. (2014). Therefore, we do not claim any novelty in terms of methodology, but rather present an open source project that aims at making object detection learned on large-scale datasets available to a broader audience. However, we also extend the system of Göhring et al. (2014) with respect to the following aspects:

1. Multiple components for each detector similar by performing clustering

2. Threshold optimization using leave-one-out cross-validation and optimization of the mixture’s threshold combination

3. Flexible interactive model tuning, i.e. a user can remove components from a model and add multiple new models using in-situ images (images of the application environment)

©2014 Björn Barz, Erik Rodner, and Joachim Denzler.
Sample Acquisition  We use the ImageNet dataset (Deng et al., 2009) for automatic acquisition of a large set of samples for a specific object category. With more than 20,000 categories, ImageNet is one of the largest non-properiatery image databases available. It provides an average of 300-500 images with bounding box annotations (annotated by crowdsourcing) for more than 3,000 of those categories and, thus, is suitable for learning object detection models. Everything a user has to do in order to learn a new model using the ARTOS GUI is to search for a synset and to click “Learn!” (see Figure 2). For now, ARTOS requires access to a local copy of the ImageNet images and annotations (or a subset at least), which must be available on the file system, but we are planning to change this in future with a download interface.

Model Creation  As feature representation, we use Histograms of Oriented Gradients (HOG), originally proposed by Dalal and Triggs (2005), with the modifications of Felzenszwalb et al. (2010) as features. Hariharan et al. (2012) proposed a method for fast learning of models, even when only few positive and no negative samples are available. It is based on Linear Discriminant Analysis (LDA), which makes the following assumptions about class and feature probabilities:

\[ P[y] = \varphi^y (1 - \varphi)^{1-y} \]  
\[ P[x|y] \sim \mathcal{N}(\mu_y, \Sigma) \]

From this, a linear classifier of the form \( w^T x + b \) can be derived and \( w \) turns out to be:

\[ w = \Sigma^{-1} (\mu_{pos} - \mu_{neg}) \]

Important for a fast learning scheme is that \( \mu_{neg} \) and \( \Sigma \) do not depend on the positive samples and can be computed in advance and off-line.
Figure 2: The *Learn Model from ImageNet* dialog of the ARTOS GUI. It allows the user to select a synset and adjust some parameters related to clustering and threshold optimization.

In combination with HOG features, Hariharan et al. (2012) call the resulting features *Whitened Histogram of Orientations* (WHO), although the ideas of Hariharan et al. (2012) can be also used with other feature types that could be integrated into ARTOS.

ARTOS first performs two stages of clustering on the dataset obtained from ImageNet: first, the images are divided into clusters by an aspect-ratio criterion, and then, each resulting cluster is subdivided with respect to the WHO features of the samples. This is done using a simple k-Means algorithm. One model is learned for each cluster according to equation 3. Those models are then combined in a model mixture for the object class.

**Threshold Optimization** While Hariharan et al. (2012) gave an explicit formula for $w$ in $w^T x + b$, they kept quiet about how to obtain an appropriate bias $b$. To determine optimal biases, ARTOS finally runs a detector with the learned models on some of the positive samples and additional negative samples taken from other *synsets* of ImageNet in order to find a bias that maximizes the F$_1$-Measure.

But finding the optimal threshold for each model of the mixture independently is not sufficient. Since the models are combined and the final detection score will be the maximum of the detection scores of the single models, an optimal combination of biases is crucial. Thus, we employ the heuristic *Harmony Search* algorithm of Geem et al. (2001) to approximate an optimal bias combination that maximizes the F$_1$-Measure of the entire model. This could be easily adapted to other performance metrics or other optimization algorithms. In particular, we do not advocate for Harmony Search here and we believe that any other heuristic search method would work equally well.
Adaptation  After a model has been learned from ImageNet, it can be adapted easily to overcome domain-shift effects. PyARTOS, the Python based GUI to ARTOS, enables the user to take images using a camera (see Figure 3) or to annotate some image files, from which a new model will be learned and added to the model mixture.

Detection  For fast and almost real-time object detection, ARTOS incorporates the FFLD library (Fast Fourier Linear Detector) of Dubout and Fleuret (2012), which leverages the Convolution Theorem and some clever implementation techniques for fast template matching.

2. How to get ARTOS and what are the next steps?

A first (still not feature-complete) version of ARTOS has been released under the terms of the GNU GPL:

http://cvjena.github.io/artos/

There is also a related github repository and we invite everyone to contribute and use our code for various vision applications.

We are planning to add a public model catalogue to the website of ARTOS so that people can upload and download models of common objects. The project is part of the lifelong learning initiative of the computer vision group in Jena.

Enjoy object detection!
Acknowledgements  We would like to thank Dubout and Fleuret (2012) and Hariharan et al. (2012) for providing the source code of their research. Furthermore and most importantly, we also thank the authors of Göhring et al. (2014), who presented the approach on which our open source project is based on.

References

Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 248–255, June 2009.


