Detection of Dogs in Video Using Statistical Classifiers

Roman Juránek

Graph@FIT
Brno University of Technology, Faculty of Information Technology
Božetěchova 2, 612 66 Brno, Czech Republic
ijuranek@fit.vutbr.cz

Abstract. A common approach to pattern recognition and object detection is to use a statistical classifier. Widely used method is AdaBoost or its modifications which yields outstanding results in certain tasks like face detection. The aim of this work was to build real-time system for detection of dogs for surveillance purposes. The author of this paper thus explored the possibility that the AdaBoost based classifiers could be used for this task.

1 Introduction

Statistical classifiers can very well be used for object detection or pattern recognition in raster images. Current algorithms even exhibit real-time performance in detecting complex patterns, such as human faces [1], while achieving precision of detection which is sufficient for practical applications. Recent work of Šochman and Matas [2] even suggests that any existing detector can be efficiently emulated by a sequential classifier which is optimal in terms of computational complexity for desired detection precision. Tn their approach the effort is invested into selecting a set of suitable features which are then automatically combined into an ensemble by the WaldBoost [3] algorithm.

In this paper, dog detection for surveillance purposes is discussed. The aim of the work was to build simple system that will cope with low resolution and possibly low quality (noise, compression artifacts, etc.) input video. The general object detection approach with WaldBoost based classifiers with Haar-like features was used. The main problem to deal with in this work was the training data. Since there is no freely available database of suitable dog samples, the data had to be collected form the internet.

The following section summarizes algorithms commonly used for general object detection. Section 3 discusses specific issues related to detection of dogs. Experiments carried out with the detection of dogs on surveillance recordings and their results are described in section 4. The section also describes tools that were used for training and evaluation of the classifiers. The last section summarizes the paper and discuses possibilities of the future work.

2 Background

Object detection with classifiers was first used by Viola and Jones [1] for rapid face detection. In their work they used AdaBoost algorithm to select a set of critical features out of large set of Haar-like wavelets.

The basic algorithm of AdaBoost was described by Freund and Schapire in [4]. The response of weak hypothesis is restricted to binary value and thus the algorithm is referred to as discrete AdaBoost. Schapire and Singer [5] introduced real AdaBoost which allows confidence rated predictions and is most commonly used in combination with domain partitioning weak hypotheses (e.g. decision trees). Viola and Jones [1] used the AdaBoost classifiers to detect faces in images. In their work they used simple weak hypotheses consisting from a single Haar-like feature, and classifier cascade.

AdaBoost in its basic form, greedily selects weak hypotheses that are only moderately accurate to create very accurate classifier. The result of such classifier is based on linear combination of the selected weak hypotheses (Equation 1).

$$f_T(X) = \sum_{t=0}^{T} \alpha_t h_t(X) \tag{1}$$

The weak hypotheses selected by AdaBoost are not optimal as the process is greedy. There have been works addressing this fact, e.g. FloatBoost [6] or Total Corrective Step [7].

2.1 WaldBoost

The main drawback of AdaBoost classifiers is that each weak hypothesis within the classifier must be evaluated to obtain response. To reduce the number of evaluated hypotheses a classifier cascade [1] can be used. Another approach was introduced by Matas and Šochman [3] – in their WaldBoost they keep the linear structure of the classifier, and for each weak hypothesis selected by AdaBoost they calculate two early-termination thresholds θ_A and θ_B using Wald's *Sequential Probability Ratio Test* [8]. During classifier evaluation, when a strong classifier sum $f_t(X)$ exceeds a threshold evaluation ends, next stage is evaluated otherwise. The Equation 2 shows evaluation of $t - th$ stage of a classifier.

$$H_t(X) = \begin{cases} +1, & f_t(X) \geq \theta_A^{(t)} \\ -1, & f_t(X) \leq \theta_B^{(t)} \\ \text{continue } H_{t+1}, & \theta_A^{(t)} < f_t(X) < \theta_B^{(t)} \end{cases} \tag{2}$$

During the training, the thresholds for each stage are estimated according to parameters α and β, which represent *false negative rate* and *false positive rate* of the final classifier. In detection tasks, β is usually set to 0. As the result, positive threshold θ_A will be set to ∞ for each selected stage and thus a sample cannot be accepted by early termination mechanism (strong classifier sum cannot possibly

reach ∞). Early termination can thus only reject samples as negative. Positive detection can be reached only by evaluating the entire ensemble.

The *speed* of the WaldBoost classifiers is indicated by the average number of evaluated hypotheses per sample. The speed largely depends on the classifier application, training settings (namely *alpha* and *beta* parameters) and also on the number of weak hypotheses in the classifier.

2.2 Features

The Properties of a classifier largely depend on the low level weak hypotheses and features. In many computer vision problems, like the face detection, are the Haar-like features, commonly used since in combination with integral image they exhibit extreme performance and provide good amount of information. Other features commonly used are Gabor wavelets [9], Local Binary Patterns [10] or Local Rank Differences [11].

Fig. 1. Typical shapes of Haar-like features

Haar-like features (on the Figure 1) are based on the difference of adjacent rectangular regions of an image. Due the integral image representation, the response of a Haar-like feature can be obtained in constant time regardless of its size. The main drawback of the features is that they are dependent on light conditions and their response must be normalized

3 Detection of Dogs in Video

Compared to other tasks like face detection, the dog detection is more difficult since the silhouette of dogs changes over time as the dog moves. Variety of dog shapes is very large (different postures, orientations, etc). The texture and brightness of dogs also varies in wide range. The detection of dogs simply lacks a visually well defined class. Single WaldBoost classifier is, therefore, not able to detect dogs viewed from arbitrary angle. For the above reason the task in this paper is limited to the detection of dogs viewed from the profile only. While this limitation may seem relatively severe, it does not introduce any serious limitation from the application point of view, as the objects in the video sequence can be tracked and whenever the classifier detects a dog seen from the profile, the whole track is known to represent the dog.

Another issue connected with dog detection is that the silhouette is not horizontally symmetric. Unlike the face detection, the input image must be searched at least two times – first time original image and second time image horizontally

Fig. 2. Detection of dog in a simple outdoor scene

flipped (or with horizontally flipped classifier). This ensures that all dogs could be detected regardless of their orientation.

3.1 Training Data

Since there is no freely available database with suitable dog samples, training data for our experiments – images of dogs viewed from profile, are collected from the Internet and also from our own video recordings with dogs.

Fig. 3. Excerpts from training datasets

A drawback of the data is that conditions in images vary in wide range (different background, lighting, compression artifacts, etc.) which would probably result in worse classifier performance. To bring best performance, the data should correspond to the conditions in the target application which in our case was dog detection in underground stations. It is clear that the data, which were taken mostly from outdoor scenes, do not correspond to the conditions in underground stations very well.

From the annotated data, samples for the training framework (see section 4.1) were generated. The size of the samples was set similarly to size commonly used in face detection to $24 \times 16px$. In smaller samples, there would be hard to catch the silhouette of a dog. Larger samples, on the other hand, would increase number of weak hypotheses in the training which would slow down the training

process rapidly. Number of *dog* samples in the current dataset is 618 and they are divided to *training set* (236 samples), *validation set* (190 samples) and *testing set* (192 samples).

In the Figure 3 you can see samples form *dog* class (left) and *background* class (right) from the dataset. As the *background* class holds potentially infinite number of images, the samples can be randomly extracted from different images during training. The number of background samples during the training can thus, in combination with bootstrapping in WaldBoost algorithm, reach millions.

4 Experiments and Results

The experiments included training of WaldBoost classifiers using data described in section 3.1. The experiments also involved testing of different false negative rate (training parameter α) and observe how this settings influence the properties of the resulting classifier.

The following two sections describe the classifier training framework and the real-time object detection engine. In the section 4.3 are actual results of performed experiments.

4.1 WaldBoost Training

The Boosting Framework [12] was used for classifier training. Overview of the training process is shown on the Figure 4. The input for WaldBoost training are configuration (algorithm settings and used image features and weak hypotheses) and training data. The classifier (selected hypotheses) and its evaluation are generated as an output of the process.

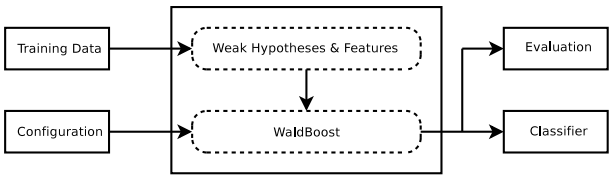

Fig. 4. Structure of the training framework

The framework supports many different types of image features and weak hypotheses as well as different training algorithms. For this paper the important is support of Haar-like features and WaldBoost training algorithm. Beside other parameters for the WaldBoost training, the configuration specifies the parameter α (see section 2.1) which has the greatest influence on the properties of classifiers.

The output of an experiment is a classifier and its evaluation on the training and testing sample set for each selected stage. Beside others, Receiver Operat-

ing Characteristic (ROC), Precision-Recall (PRC) and Negative Rejection Rate curves are generated.

Example of stage in the XML

```
<Stage posT="1E+10" negT="-1.3">
  <DomainPartitionWeakHypothesis responses="-0.2 -0.3 -0.4 -0.5 0.5 0.4 0.3 0.2">
    <Discretize min="-2.0" max="2.0" N="8">
      <HaarHorizontalDoubleFeature x="2" y="4" bw="5" bh="8" />
    </Discretize>
  </DomainPartitionWeakHypothesis>
</Stage>
```

Fig. 5. Example of a stage representation in XML

The resulting classifier is represented by a XML structure which contains a sequence of stages (Figure 5). Each stage has its early-termination thresholds and a single selected weak hypothesis. (`posT` corresponds to the θ_A and `negT` to the θ_B).

$$h_t(X) = W_k^{(t)} \tag{3}$$

The stage holds a domain partitioning weak hypothesis with an Haar-like image feature. Evaluation of the stage is explained by the Equation 3, where

$$k = \left\lfloor \frac{N(g(X) - min)}{max - min} \right\rfloor$$

is used as index to a table weights $W^{(t)}$ (`responses` in the XML code). The $g(X)$ is a response of the feature on sample X. The values N, min and max correspond to parameters of `Discretize` element in the XML code.

In the Figure 6 is illustrated how the weak hypothesis works. The quantized response of the feature is used as an index to the table of weights and thus to each interval of the feature response a weight value participating on the strong classifier sum is assigned.

4.2 Real-Time Object Detection

One of goals of this project was to build a real-time system for detection of dogs. For this purpose, detection engine that uses classifiers generated by the Boosting Framework was developed. The main properties of the engine is efficient Haar-like feature evaluation and multiscale object detection.

The actual object detection is achieved by scanning the image. When the detection is executed, first, input image is preprocessed and then scanned with a sliding window (figure 7), evaluating classifier on each position. Encountered detections are passed to a non-maxima suppression algorithm [2] which removes

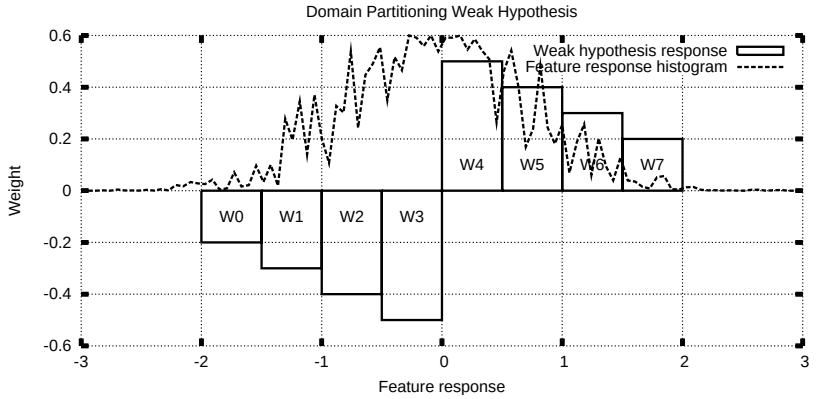

Fig. 6. Example of weak hypothesis with $min = -2$, $max = 2$ and $N = 8$ (corresponding to the XML code in the Figure 5).

possible multiple detections of same object. The image scanning technique can inherently detect multiple objects in an image.

The preprocessing stage involves calculation of standard integral image and integral image of squared values. The only reason to generate the other integral image is rapid calculation of standard deviation of classified samples, which is needed for Haar-like feature response normalization.

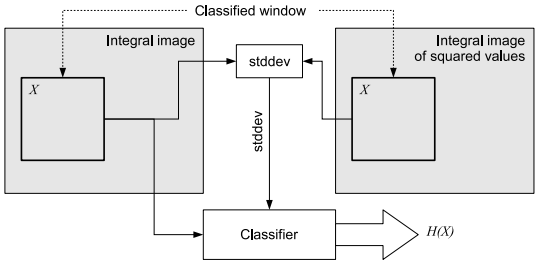

Fig. 7. Image processing in the detection engine

Detection of differently sized objects is achieved simply by resizing the classifier – position and size of each feature within the classifier is adjusted for particular scan window size. The image is scanned with the classifier afterwards.

4.3 Results

For the experiments, classifiers were trained using the framework described in the section 4.1. The training parameter α was set in range from 0.01 (few false negatives) to 0.2 (more false negatives).

The plot on the figure 8 shows *Receiver Operating Characteristics* (ROC) of the classifiers and on the figure 9 there are speed characteristics of the classifiers.

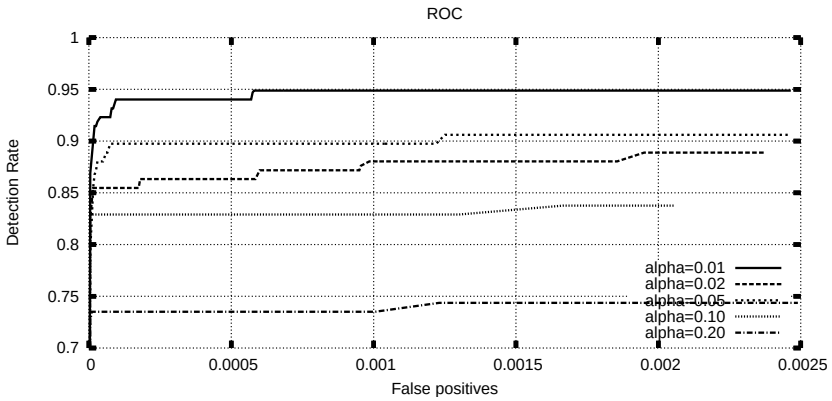

Fig. 8. ROC curves of the classifiers.

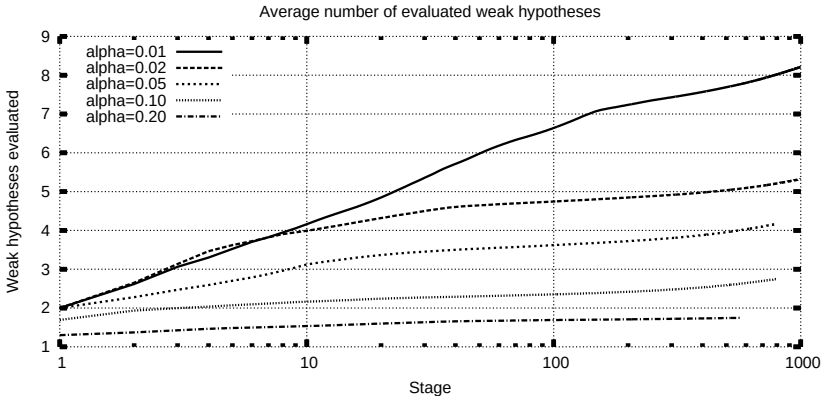

Fig. 9. Speed of the classifiers – average number of evaluated weak hypotheses on test set. The classifiers with higher α are faster due to more rapid rejection of negative samples. Note that the speed increases logarithmically with number of stages.

Another interesting property of WaldBoost classifiers is a *Negative Rejection Rate* (Figure 10). The curves show the ability of classifiers to reject negative samples by early-termination mechanism. When the classifier can reject vast majority of samples before final stage, only very few samples are needed to be classified by thresholding the final classifier response. The $\alpha = 0.1$ classifier can reject 99 % of negative samples before 10-th stage, whereas the $\alpha = 0.01$ classifier needs 100 stages to do same job.

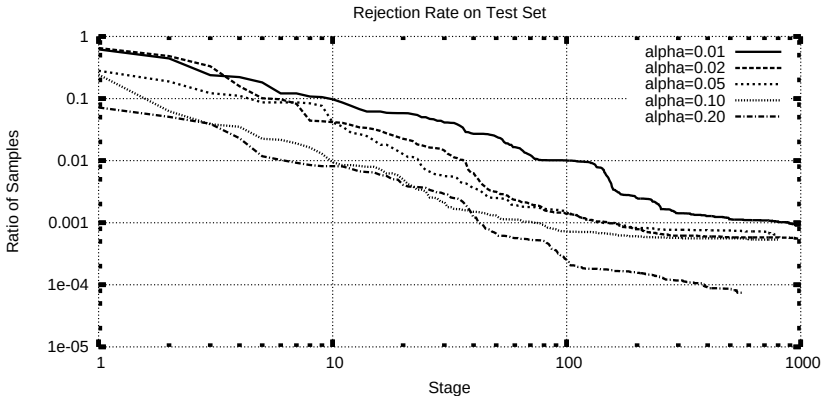

Fig. 10. Negative Rejection Rate of the classifiers. Ratio of negative (*background*) samples not rejected by the early-termination. More accurate classifiers (lower α) rejects samples slowly.

The experiments shows that classifiers with the α set to higher value are faster and as ROC curves shows, faster classifiers are less accurate because of rapid rejection of samples.

The classifiers were tested in the real-time detection engine with a recordings of an underground station. The detection was executed on $352 \times 288px$ image and the engine achieved performance of $50fps$ (Intel Core2 Duo, 2.66 GHz) which was more than sufficient (typical frame rate in a surveillance system is $5fps$). The detection speed of course depends on many parameters, mainly on accuracy of used classifier (more accurate classifiers takes longer time to evaluate). On the Figure 11, there are examples of detections in an underground station. The images (d) and (e) on the figure shows false detections. The false positive rate was higher due the fact that the detector is static and does not use any motion information.

5 Conclusion and Future Work

In this paper, real-time detection of dogs in low resolution videos was studied. The used method included statistical classifier training by the WaldBoost algo-

(a) (b) (c)

(d) (e)

Fig. 11. Detections in the Roma underground station (classifier with $\alpha = 0.05$ was used). The images (d) and (e) shows false detection.

rithm (section 2) and building of real-time detection engine – efficient evaluation of WladBoost classifiers on image (section 4.2). In the section 3, specific issues connected with detection of dogs were summarized and also data that was used in our experiments was described.

The method of general object detection without any extra knowledge related to dogs seems promising. The experiments (see section 4.3) shows that the method is viable despite the fact that the shape of detected objects rapidly changes over time. Even on existing dataset of images that were mostly collected from the Internet, the classifiers exhibit good characteristics. In scenes with simple background the classifiers were able to detect dogs with high accuracy. In complex scenes (lots of moving people in the station) there were more false positive detections. This was caused by the static character of the detector and also by rather small positive training set.

For the future, promising appears utilization of image features extracted from more than one frame (e.g. approach similar to [13]) instead of simple Haar-like features. Detector could be thus trained to that take advantage of both appearance and motion information to detect a dog. This approach would of course bring problems with training data because current dataset does not contain any motion information.

Acknowledgements

The author would like to thank to Ivo Řezníček for his help with collecting the data, and also to Pavel Zemčík and Adam Herout for their guidance. This work was supported by by European project CareTaker (FP6-027231).

References

1. Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. pages 511–518, 2001.
2. Jan Šochman and Jiří Matas. Learning a fast emulator of a binary decision process. In Yasushi Yagi, Sing Bing Kang, In So Kweon, and Hongbin Zha, editors, *Computer Vision - ACCV 2007. Proceedings 8th Asian Conference on Computer Vision*, volume II of *LNSC*, pages 236–245, Heidelberg, Germany, November 2007. Springer. Sang Uk Lee Outstanding Paper Award.
3. Jan Sochman and Jiri Matas. Waldboost " learning for time constrained sequential detection. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 150–156, Washington, DC, USA, 2005. IEEE Computer Society.
4. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag.
5. Robert E. Schapire and Yoram Singer. Machine learning, 37(3):297-336, 1999. improved boosting algorithms using confidence-rated predictions, 1999.
6. S. Li, Z. Zhang, H. Shum, and H. Zhang. Floatboost learning for classification, 2002.
7. Jan Sochman and Jiri Matas. Adaboost with totally corrective updates for fast face detection. In *FGR*, pages 445–450, 2004.
8. A. Wald. *Sequential Analysis*. John Wiley and Sons, Inc., 1947.
9. Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):959–971, 1996.
10. Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.
11. Pavel Zemčk, Michal Hradiš, and Adam Herout. Local rank differences - novel features for image. In *Proceedings of SCCG 2007*, pages 1–12, 2007.
12. Michal Hradiš. Framework for research on detection classifiers. In *Proceedings of Spring Conference on Computer Graphics*, pages 171–177, 2008.
13. Michael Jones, Paul Viola, Paul Viola, Michael J. Jones, Daniel Snow, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. In *In ICCV*, pages 734–741, 2003.