Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Determining Posterior Probabilities on the Basis of Cascaded Classifiers as used in Pedestrian Detection Systems

Roland Schweiger[*], Henning Hamer[†] and Otto Löhlein[‡]
[*]University of Ulm, Dept. of Measurement, Control and Microtechnology, 89069 Ulm
uni-ulm.schweiger@daimlerchrysler.com
[†]University of Ulm, Dept. of Neural Information Processing, 89069 Ulm
henning.hamer@uni-ulm.de
[‡]DaimlerChrysler AG, REI/AI, 89081 Ulm, Germany
otto.loehlein@daimlerchrysler.com

Abstract— **Cascaded classifiers are widely spread in automotive pedestrian detection systems. Since there has been no research on probabilistic information derivable on the basis of a cascade, these systems are limited in the sense that they only exploit the binary classification results. In contrast to that, this paper presents a mathematically founded model regarding the computation of posterior probabilities on the basis of such classifiers. This is highly relevant in respect of the further development of robust and reliable detection systems.**

I. Introduction and Related Work

Many of the recently developed pedestrian detection systems in automotive applications [1], [2] are based on cascaded classifiers, aiming at high detection rates in combination with low computation time. A cascade consists of several concatenated classifiers and corresponds to a degenerated decision tree. It is applied to a set of hypotheses, generated at every time frame and covering all relevant image locations and scalings. A detection occurs, if the corresponding hypothesis has passed a specified layer called detection layer. By adding layers to or removing layers from the end of the cascade, the working point of the classifier is adjusted. This is a rather heuristical approach and it would be more systematic to adjust the detection rate by applying a threshold to the mathematically modeled posterior probability of each sample.

Furthermore, detection information of previous frames could be used by the searching strategy at the next frame. One possible approach to accomplish this is to use a particle filter in order to track the region of interest over time [3], [4]. Generally speaking, a particle filter reduces a searching problem to a verification problem [5], [6]: Each particle can represent a hypothesis and therefore be classified by the cascade detector. Over time, search narrows down to image areas where the appearance of a pedestrian is more likely. However, this requires the posterior probability of each sample in respect of the measurement update stage of the particle filter.

Most important, recent developments based on subsequent stages of tracking allow the usage of unresolved sensor data [7], [8]. This means that there is no necessity for an explicit decision of the detector. Instead, the main focus here is to provide target candidates with probabilities, each associated with a belief of existence. Detectors of such type are called probabilistic detectors. In this context, the tracker simultaneously provides a-posteriori estimates of the state variables and the probability of existence. The advantage compared to the classical approach is, that temporal a-priori knowledge can be incorporated into the detection decision. This is state of the art in the field of state estimation. In order to use the cascaded classifiers in such a manner, it must assign a probability instead of a binary decision to each sample. Here, a mathematical model is key. This is the focus of this work.

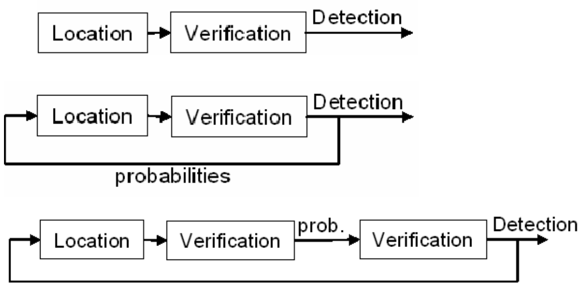


Fig. 1. Different system designs of common detection systems. The availability of the posterior probabilities after the verification step can improve these whole detection system. First, the probabilities can be used to force a desired detection rate. Secondly, the posterior probability can supply information for the searching strategy regarding the next frame. Thirdly, the detector can be used as a probabilistic detector. In this context, the tracker provides the a-posteriori estimates concerning the existence itself.

A. Structure of the Paper

This paper is structured as follows: Section II-A gives a theoretical overview of the AdaBoost algorithm, which is used to train each cascade stage. The section also recapitulates the theoretical upper bound on misclassification error. Furthermore, it briefly summarizes the connection of boosting and logistic regression and recapitulates the posterior probability presented by Friedman, Hastie and Tibshirani [9]. This forms the basis for the deduction of posterior

1-4244-1068-1/07/$25.00 ©2007 IEEE.

probabilities based on cascaded classifiers discussed in detail in Section III. Using the principle of Probabilistic Boosting Trees [10], we present a mathematical model for obtaining a-posteriori estimates on the basis of the cascade results. Section IV finally provides first experimental results using artificial and real data in order to verify the presented theory.

II. AdaBoost and Logistic Regression

A. The AdaBoost Approach to Machine Learning

Boosted classifiers and the related AdaBoost algorithm introduced by Freund and Schapire [11] are very well described in [12]–[14]. A pseudo code description of the algorithm regarding a two-class classification scenario is given in Figure 2. It requires a training set $(x_1, y_1), \dots, (x_N, y_N)$ as input with x_j being a vector valued feature and $y_j \in \{-1, +1\}$ a corresponding label. In every round $t = 1, \dots, T$, AdaBoost calls a weak learning algorithm which trains the classifier h_t on a weighted version of the training set by minimizing the weighted error

```
Given: samples $(x_1, x_1), \dots, (x_N, y_N)$ where $y_j \in \{-1, +1\}$
Initialize $w_j = \frac{1}{N}$.
For $t = 1, \dots, T$:
    • Train base classifier incorporating the weights $w_j, j = 1, \dots, N$ and determine base classifier $h_t$ that minimizes the weighted error on the training data.
    • Choose $\alpha_t$
    • Update sample weights and normalize so that $w$ will remain a distribution:
          $w_j = \frac{w_j e^{-\alpha_t y_j h_t(x_j)}}{Z_t}$
      $Z_t$ is the normalization factor.
The final strong learner decision $H$ realizes a majority vote of all weak learner decisions $h_t$:
    $H(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$
```

Fig. 2. Pseudo code description of the boosting algorithm AdaBoost regarding a two-class classification scenario.

$$\epsilon_i = \sum_j w_{i,j} \left| \frac{h_i(x_j) - y_j}{2} \right|, \tag{1}$$

and determines a parameter $\alpha_t \in R$. This α_t intuitively measures the assigned importance of h_t. For a binary decision of h_t, we typically set

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right). \tag{2}$$

The final strong learner decision H is a weighted majority vote of the T weak classifiers where α_t is the weight assigned to h_t:

$$H(x) = \begin{cases} +1: & A(x) := \sum_t \alpha_t h_t(x) \geq \theta \\ -1: & \text{else.} \end{cases} \tag{3}$$

$A(x)$ is called activation of the strong learner.

B. Analyzing the Training Error

In [15], [11] an upper bound on the misclassification error of an AdaBoost classifier is presented and defined as follows:

$$\frac{1}{N} |\{j : H(x_j) \neq y_j\}| \leq \frac{1}{N} \sum_j e^{-y_j A(x_j)} = \prod_t Z_t, \tag{4}$$

with $A(x_j)$ being the strong learner's activation so that $H(x_j) = \mathrm{sign}(A(x_j))$. The inequality can be proved by the fact that $e^{-y_j A(x_j)} \geq 1$ in case $H(x_j) \neq y_j$. The equality follows by unraveling the recursive definition of w. Note that α_t in Eq. 2 is chosen in order to minimize Z_t at each round. So, at heart, AdaBoost is a procedure for finding a linear combination $A(\cdot)$ of weak classifiers which minimizes the upper bound on misclassification error.

C. Boosting and Logistic Regression

In order to estimate the probability that a sample corresponds to a particular label, Friedman et al. have shown in [9] that AdaBoost is effectively approximating logistic regression. Minimizing the exponential criterion

$$\mathrm{E}\left[e^{-yA(x)}\right] \tag{5}$$

with respect to $A(x)$ leads to

$$A(x) = \frac{1}{2} \log \frac{p(y = +1|x)}{p(y = -1|x)}. \tag{6}$$

Although $\mathrm{E}[\cdot]$ stands for the expectation of the worst case misclassification error regarding all samples, it is sufficient to minimize the criterion conditional on x. Resolving Eq. 6 leads to

$$q(y = +1|x) := p(y = \pm 1|x) = \frac{e^{y \cdot A(x)}}{e^{-A(x)} + e^{A(x)}}. \tag{7}$$

This function is plotted in Figure 3.

III. Analyzing Cascaded Classifiers

A. Principle of Cascaded Classifiers

The cascade classifier approach, first presented in [16], is an object detection framework, capable of processing images extremely rapidly while achieving high detection rates. For this purpose, classifiers with increasing complexity are combined in a cascaded manner as illustrated in Figure 4. Each cascade stage discards samples originating from the image background and passes promising samples to the successive and more complex stage. Therefore the most computation effort is spent on promising (object-like) regions. The classifiers themselves are based on Haarwavelet-like features and trained with the AdaBoost algorithm presented in Section II-A.

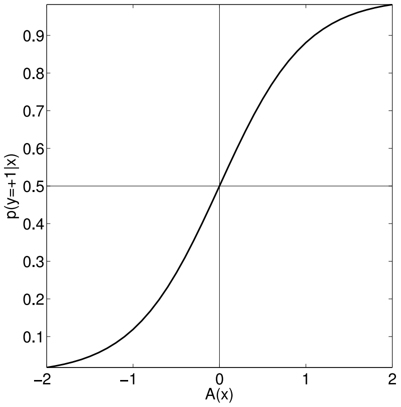


Fig. 3. Logistic function (see [9]) for the estimation of the posterior probability of a sample x, depending on the activation $A(x)$ assigned by a boosted classifier.

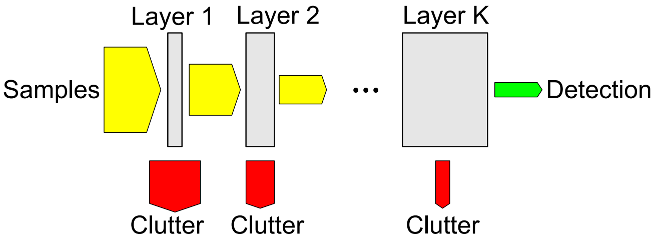


Fig. 4. Cascade principle: each layer discards clutter and passes promising samples to the successive and more complex stage.

In order to guarantee a certain overall detection rate, an user defined detection rate is forced at each layer by adjusting the threshold θ in Eq. 3. Note that this adjustment invalidates the upper bound on misclassification error presented in Section II-B and consequently makes the posterior probability presented in Section II-C invalid. This issue will be discussed in detail further below. The posterior probability in Eq. 7 is only defined for an individual cascade stage. The next section therefore describes the principle regarding a Probabilistic Boosting Tree presented in [10] which we will adopt to cascaded classifiers. In fact, a cascaded classifier is a degenerated tree itself.

B. Probabilistic Boosting Trees

In [10], a learning framework called Probabilistic Boosting Tree (PBT) is presented. One goal of this paper is to determine the posterior probability of a given sample x and a label $y \in \{-1, +1\}$ based on such a PBT. Every node of a PBT is an AdaBoost classifier with the activation threshold $\theta = 0$. As a result, the sample set is divided into two subsets depending on the posterior probability $q(y = +1|x)$ (Eq. 7). If $q(y = +1|x) < 0.5 - \epsilon$, the sample is passed to the left sub-tree, if $q(y = +1|x) > 0.5 + \epsilon$, the sample is passed to the right sub-tree, and if $0.5 - \epsilon \leq q(y + 1|x) \leq 0.5 + \epsilon$, the sample is passed to both sub-trees. Here, ϵ was introduced in order to avoid over fitting of the data. The whole PBT is trained recursively and expanded until a predefined depth is reached. In order to determine the posterior probability of an unseen sample x, it traverses the tree, again by continuously applying the posterior probability threshold 0.5. The authors of [10] then use the principle of total probability to derive a posterior probability based on the output of all nodes of the PBT. This approach results in the following recursive definition:

$$
\begin{aligned}
p(y|x) &= \sum_{l_1} q(l_1|x) \cdot p(y|l_1, x) \\
&= \sum_{l_1,l_2} q(l_1|x) \cdot q(l_2|l_1, x) \cdot p(y|l_2, l_1, x) \\
&= \dots \\
&= \sum_{l_1,\dots,l_n} q(l_1|x) \cdot \dots \cdot p(y|l_n, \dots, l_1, x) \qquad (8)
\end{aligned}
$$

Each $p(y| \dots, x)$ is a recursive element. l_n stands for the branch at level n of the tree. $q(l_n|l_{n-1}, \dots, l_1, x)$ represents the posterior probability based on the corresponding individual classifier at level n and is computed in analogy to Eq. 7. When the recursion reaches a leaf-node, the empirical probability of a sample of class $y \in \{-1, +1\}$ is returned and the recursion ends. In this case

$$
p(y|l_1, \dots, l_n, x) := p_{\text{emp}}(y|l_1, \dots, l_n). \qquad (9)
$$

In [10], the empirical probabilities are determined during training stage and defined as

$$
p_{\text{emp}}(y|l_1, \dots, l_n) = \sum_j w_j \delta(y_j = y). \qquad (10)
$$

Figure 5 illustrates the composition of the posterior probability as described above.

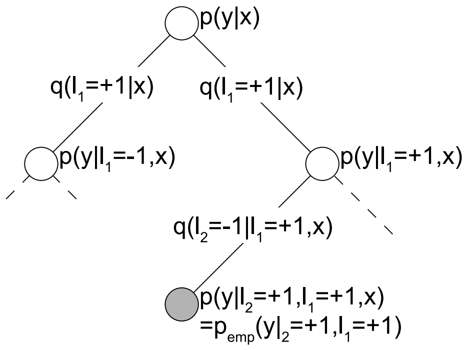


Fig. 5. Calculation of the posterior probability based on the Probabilistic Boosting Tree described in [10]. l_n stands for the branch at level n of the tree. $q(l_n|l_{n-1}, \dots, l_1, x)$ represents the posterior probability based on the corresponding individual classifier at level n. $p(y|x)$ is then calculated recursively by weighting the left and right sub-trees according to q. At the leaf nodes, the recursion ends with $p(y|\cdot, x) := p_{\text{emp}}(y|\cdot)$.

In order to reduce computation time, [10] does not consider the complete PBT but only those nodes that the samples passes on its way to a leaf node. The contribution of each ignored sub-tree is approximated by the empirical probability of the corresponding learnset. The ϵ-case is disregarded during testing stage, so the decision as to whether a sample is directed to the left or to the right sub-tree depends on the posterior probability based on the activation assigned by the corresponding individual classifier. However, applying a posterior probability threshold of 0.5 is just the same as using an activation threshold of 0.

C. Posterior Probability and Adapted Thresholds

As mentioned earlier, determining posterior probabilities in a PBT-like manner cannot be transfered to cascaded classifiers directly. This is due to two reasons:

- The cascade is not fully expanded to a defined tree depth (there are no left sub-trees).
- After the training of each stage, the threshold θ of the strong learner is adapted in order to guarantee a certain detection rate, i.e. $\theta \neq 0$.

While the first issue can be solved by using empirical probabilities (like done in [10] during testing), the second point directly affects the calculation of the posterior probabilities in accordance to Eq. 7. In [10], the posterior probability $q(y|x)$ of a sample computed on the basis of one node of the PBT is used to assign a weight to each of the two sub-tree probabilities emerging from the corresponding left and right sub-tree (see Eq. 8). If a sample is assigned the activation $A(x) = 0$ and the stronglearner threshold $\theta = 0$, it is very uncertain which of the two branches is the appropriate one.This is reflected by the posterior probability $q(y = +1|x) = 0.5$. In consequence, the two probabilities are averaged. In order to ensure the same behavior regarding $\theta \neq 0$, 7 must be adapted to

$$q^*(y = +1|x) = \frac{e^{(A(x)-\theta)}}{1 + e^{2(A(x)-\theta)}}. \tag{11}$$

This formula can be derived by minimizing the expectation

$$\mathrm{E}\left[e^{-y(A(x)-\theta)}\right] \tag{12}$$

with respect to $A(x)$, which leads to

$$A(x) = \frac{1}{2}\log\frac{q(y = +1|x)}{q(y = -1|x)} + \theta. \tag{13}$$

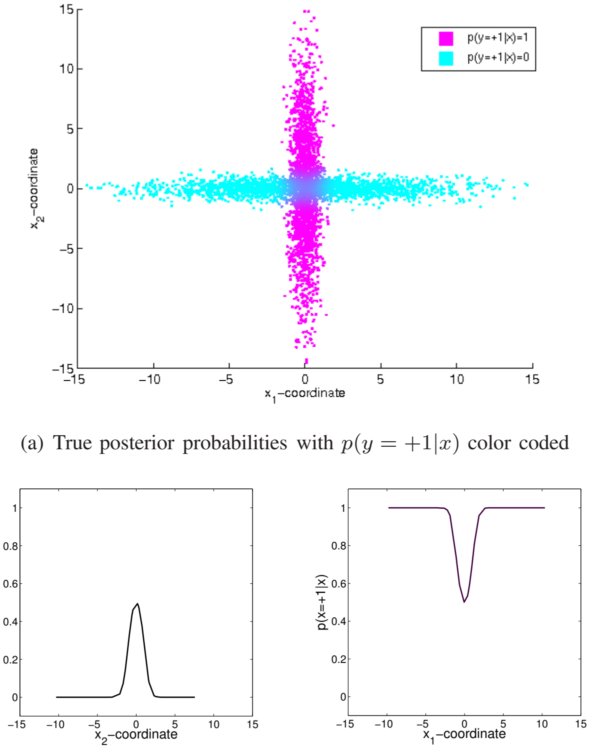
As in [9], Eq. 12 again is motivated by the upper bound on misclassification error concerning the strong learner's decision. The adapted threshold $\theta \neq 0$ is explicitly considered:

$$\frac{1}{N}\left|\{j : H(x_j) \neq y_j\}\right| \leq \frac{1}{N}\sum_j e^{-y_j(A(x_j)-\theta)}. \tag{14}$$

IV. Experiments

A. Artificial Data Set

In order to verify the presented theory, initially some experiments with artificially generated samples where conducted. Two 2D normal probability density function were used to draw the samples of learn and test set, with $(\sigma_{x_1}^{(\text{pos})}, \sigma_{x_2}^{(\text{pos})}) = (0.5, 5.0)$ for positive samples and $(\sigma_{x_1}^{(\text{neg})}, \sigma_{x_2}^{(\text{neg})}) = (5.0, 0.5)$ for negative samples. Thus, a cross shape is created. Using Bayes' theorem, a ground truth of the posterior probabilities is available and plotted in Figure 6. In order to receive a more precise impression of the exact probability values, a vertical and a horizontal cut through the cross in respect of the posterior probability is also shown. The vertical cut of the cross contains mostly positive samples (positive cut). In contrast, the horizontal cut generally relates to negative samples (negative cut). For an independent learn set, a cascade classifier was trained. The classifier characteristics are listed in table I.



(a) True posterior probabilities with $p(y = +1|x)$ color coded

(b) horizontal cut

(c) vertical cut

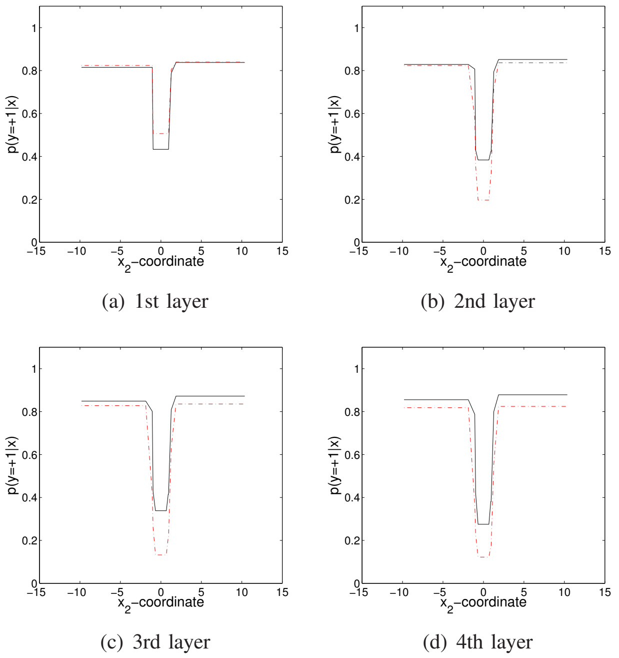
Fig. 6. True posterior probabilities for an artificial data set: (a) the probabilities $p(y = +1|x)$ derived by Bayes's Theorem are color coded. Samples with low posterior probability are blue, samples with high probabilities are red. (b) Horizontal cut (samples with $x_2 = 0$). (c) Vertical cut samples with $x_1 = 0$.

TABLE I

Characteristics of cascade classifier trained on an artificial data set.

Amount of classifiers:	4
Amount of weakleaners for every stage:	$7, 15, 15, 15$
Detection rate (on test set) D_{test}:	98.0%
False positive rate (on test set) F_{test}:	11.6%

In [10], empirical probabilities used for estimating the a-priori knowledge in the leaf nodes were obtained on basis of the learnset. In contrast to that, we used an independent data set (containing $10 \cdot 10^8$ samples of each class) for the computation of those empirical probabilities. The posterior probabilities were finally determined regarding a third and also independent data set. Figure 7 shows the vertical cut of the posterior probabilities. For completeness, the dashed lines in this figure also show the posterior probabilities that were calculated disregarding the adapted thresholds θ_k. First, referring to the solid lines, it can be observed that the posterior probabilities of samples within the vertical wings of the cross increase with cascade depth. This is not the case, without explicit consideration of the adapted thresholds. Secondly, the posterior probabilities of samples in the center



(a) 1st layer

(b) 2nd layer

(c) 3rd layer

(d) 4th layer

Fig. 7. Vertical cuts of posterior probabilities for the artificial data set from Figure 6. At $x_2 = 0$ a probability of 0.5, at $x_2 > 0.5$ and $x_2 < -0.5$ a probability of 1 is expected. (a) shows the probabilities calculated after the first cascade layer, (b),(c) and (d) respectively the values calculated after the second, third and 4th layer.

of the cross decrease less when considering the adapted threshold as done in Eq. 11.

B. Cascaded Classifiers for Pedestrian Detection

In order to evaluate the mathematical model described in Section III, a cascade for the application of a pedestrian detection system was trained. All data originates from 20 near infrared image sequences, each with a length of 45s; manually created label information is available. Samples correspond to rectangles of various size and position within an image. A sample is assigned to labels by considering the corresponding coverage. The coverage $\text{cov}(A, B)$ is a measure for the degree of intersection of two rectangular areas A and B:

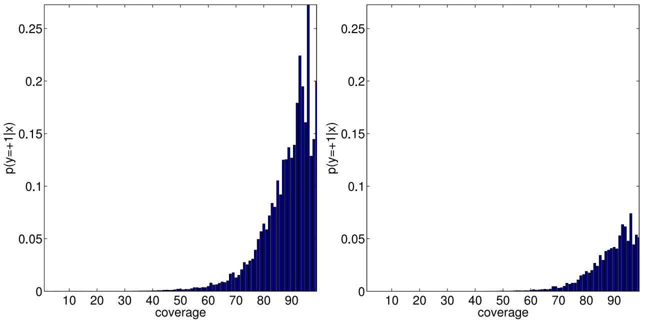
$$\text{cov}(A, B) = \frac{A \cap B}{A \cup B}. \tag{15}$$

If there is no intersection, coverage is 0. In contrast, if the sample boundaries correspond exactly to a label, the resulting coverage is 1.

Positive samples for the training were extracted on the basis of the label information (2500 positive samples). Negative samples were obtained by making use of a hypotheses generator that generates 300.000 hypotheses per image. In our training configuration, each stage of the cascade is restricted to a maximum number of weak learners, as given in table II. An independent empirical data set was used in order to determine the empirical probabilities. The posterior probabilities were then calculated for a third data set. Figure 8 displays the mean posterior probabilities plotted over coverage. High probabilities result from high coverages. Note, that the absolute values of the probabilities also reflect the a-priori information (i.e. the proportion of foreground samples in the data set) used to calculate the empirical probabilities.

TABLE II
MAXIMUM WEAK LEARNERS PER STAGE

stage	1	2	3	4	5	6,7	8-20	≥ 21
no.	3	5	7	10	13	15	25	50



(a) mean values

(b) variances

Fig. 8. Posterior probabilities over coverage calculated with a cascaded classifier for a pedestrian detection system. (a) shows the mean posterior probabilities achieved by all samples over the respective coverage with a label, (b) demonstrates the corresponding variances.

In Figure 9, two receiver operator characteristic (ROC) curves are displayed. One arose from the common cascade classification method, i.e. a detection occurs, if the corresponding sample has passed a specified detection layer. The other originates from the application of a threshold to each calculated posterior probability. Obviously, both ROC curves are similar. This is strong evidence for the validity of the presented mathematical model.

Figure 10 shows the posterior probability for two typical hypotheses. Further research will be needed in order to investigate the influence of the posterior probabilities to robust detection systems. As an example, the probabilities can help to understand typical false alarms. However, the practical relevance of posterior probability is already demonstrated in [8] and [4].

V. CONCLUSIONS AND FURTHER WORK

Cascaded classifiers are widely spread in automotive applications. In spite of this fact, there has been no research on probabilistic information derivable on the basis of cascades. Therefore, these systems are limited in the sens that they only exploit the binary classification results. In contrast to that, this paper presents a mathematically founded model regarding the computation of posterior probabilities on the basis of such classifiers.

The principle of determining the posterior probabilities of each sample is similar to the evaluation of a Probabilistic Boosting Tree. In fact, a cascaded classifier is a degenerated tree itself. At each node, the information from its descendants is weighted according to the posterior probability of the

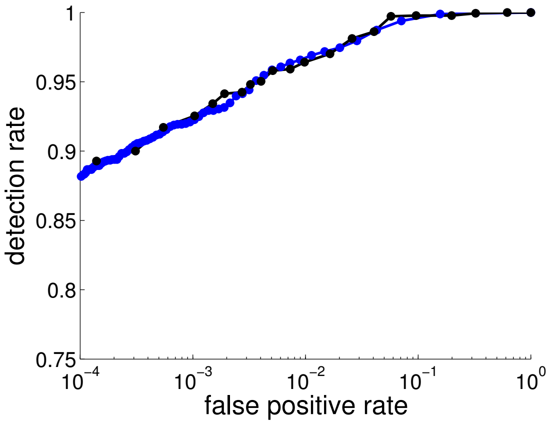


Fig. 9. Receiver Operator Characteristic curves for the trained cascaded classifier on an independent test set. The black curve arose from the common cascade classification method (i.e. a detection occurs, if the corresponding hypothesis has passed a specific layer called detection layer), the blue curve originates from the application of a threshold to the calculated posterior probability of each sample. Note that the false-positive rate axis is logarithmic.

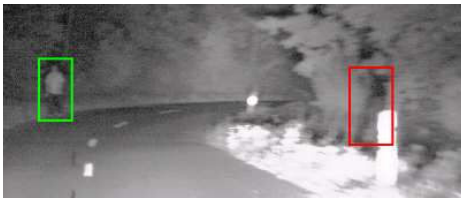


Fig. 10. Example of the posterior probability for two typical hypotheses. The detected foreground sample is displayed in green, the corresponding posterior probability is 0.023. The red sample was discarded with a value of 0.00012. The detection threshold was chosen to be 0.015. Note, that the absolute values of the probabilities also reflect the proportion of foreground samples in the data set.

corresponding strong learner and form an approximated posterior distribution.

It makes use of the fact that AdaBoost is effectively approximating logistic regression. This cannot be transfered directly to the strong learner decision made in cascaded classifiers: Whereas in Probabilistic Boosting Trees the final threshold of each individual classifier is $\theta = 0$, in cascaded classifiers the threshold of each strong learner is adapted in order to guarantee a certain detection rate. In order to address this issue, in this paper a mathematical model that also incorporates adapted thresholds $\theta \neq 0$ is derived.

Furthermore, a cascade has no left sub-trees. In this work, the probabilities of those leaves are approximated using empirical probabilities which incorporate the a-priori knowledge.

The validity of the presented mathematical model is demonstrated are based on experimental results using artificial data and data from a pedestrian detection system. These results show, that samples with high coverage to a label receive high probabilities as expected.

The availability of the posterior probabilities is highly relevant with regard to further development of robust detection systems in automotive applications. As an example, the probabilities can help to understand typical false alarms of cascaded classifiers or be used to force a desired detection rate. Furthermore, the posterior probabilities can supply information for the search strategy regarding the next time frame. Last but not least, cascaded classifiers can now be used as probabilistic detectors. In this context, future work will be the further development of a tracker stage that provides the a-posterior estimate concerning the existence of an object itself.

ACKNOWLEDGMENT

This work was supported by NIRWARN (Near-Infrared Warning), BMBF 01M3157B, Germany.

REFERENCES

[1] M. Maehlisch, M. Oberlaender, O. Loehlein, D. Gavrilla and W. Ritter, "A multiple detector approach to low-resolution fir pedestrian recognition," in *The IEEE Intelligent Vehicles Symposium, IV 2005*, June 2005, pp. 325–330.

[2] I. Kallenbach, R. Schweiger, G. Palm and O. Loehlein, "Multi-class Object Detection in Vision Systems Using a Hierarchy of Cascaded Classifiers," in *Proceedings of IEEE Intelligent Vehicles Symposium*, Tokyo, Japan, 2006.

[3] C. Idler, R. Schweiger, D. Paulus, M. Maehlisch, W. Ritter, "Realtime Multi-Target-Tracking with Particle Filters in Night View Automotive Applications," in *Proceedings of IEEE Intelligent Vehicles Symposium*, Tokyo, Japan, 2006.

[4] R. Arndt, R. Schweiger, W. Ritter, D. Paulus and O. Loehlein, "Detection and tracking of multiple pedestrians in automotive applications," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2007.

[5] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[6] A. Doucet, N. Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, ser. Statistics for Engineering and Information Science. LLC, 175 Fifth Avenue, New York, NY 10010, USA: Springer-Verlag, 2001.

[7] M. Maehlisch, R. Hering, W. Ritter and K. Dietmayer, "Multisensor vehicle tracking with the probability hypothesis density filter," in *Proceedings of the ISIF/IEEE 9'th International Conference on Information Fusion*, 2006.

[8] M. Maehlisch, W. Ritter and K. Dietmayer, "Decluttering with intergrated probabilistic data association for multisensor multitarget vehicle tracking," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2007.

[9] J. Friedman, T. Hastie and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," *Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 1998.

[10] Z. Tu, "Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering," in *Proceedings of the Tenth International Conference on Computer Vision*. IEEE Computer Society, 2005, pp. 1589–1596.

[11] Y. Freund and R. E. Schapire, "A Decision-theoretic Generalization of On-line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[12] ——, "Experiments with a New Boosting Algorithm," *Machine Learning: Proceedings of the Thirteenth International Conference*, vol. 148, p. 156, 1996.

[13] ——, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

[14] R. E. Schapire, "The boosting approach to machine learning: An overview," *MSRI Workshop on Nonlinear Estimation and Classification*, vol. 2002, 2002.

[15] R. E. Shapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, December 2000.

[16] P. Viola and M. Jones, "Robust Real-time Object Detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2002.