Decision-Level Fusion for Vehicle Detection

ZEHANG SUN and GEORGE BEBIS
University of Nevada
Department of Computer Science
Reno, Nevada, USA

NIKOLAOS BOURBAKIS
Wright State University
Department of Computer Science
Dayton, OH, USA

Abstract: This paper deals with the problem of decision-level fusion for vehicle detection from gray-scale images. Specifically, the outputs of some classifiers are simply ”distances”, that is, they represent ”distance measurements” between a query pattern and a decision boundary. We argue that the distance component is very helpful for decision fusion. Unfortunately, some of the most popular statistical decision fusion rules, such as the Sum rule and Product rule, do not take advantage of the ”distance” property. Even worse, these rules make assumptions about data independence and distribution models which do not hold in practice. Motivated by these observations, we propose a simple decision-level fusion rule in the context of vehicle detection. Our fusion rule takes advantage of ”distance” information and does not make any assumptions. We have applied this rule on a vehicle detection problem, showing that it outperforms well known statistical fusion rules.

Key–Words: Decision fusion, Haar wavelet transform, Gabor filter, Support Vector Machines

1 Introduction

With the aim of reducing injury and accident severity, vision-guided vehicle detection is an active research area among automotive manufacturers, suppliers and universities. Various vehicle detection approaches have been reported in the computer vision literature. For a recent review, please see [1]. An onboard vision system is expected to help drivers making some critical decisions, or even in some circumstances, take control of the vehicle, which might save people’s lives. Given this expectation, most of the current systems are not satisfactory due to the difficulties of the problem itself. Several factors make on-road vehicle detection very challenging including variability in scale, location, orientation, and pose. Vehicles, for example, come into view with different speeds and may vary in shape, size, and color. Vehicle appearance depends on its pose and is affected by nearby objects. In-class variability, occlusion, and lighting conditions also change the overall appearance of vehicles. Landscape along the road changes continuously while the lighting conditions depend on the time of the day and the weather.

Building a vehicle detection system consists of two steps: First, training a classifier using of a training set of data to estimate the parameters of the chosen model, and then a test set, patterns previously unseen by the classifier, is used to determine the classification performance. This ability to respond to novel patterns meaningfully, i.e., generalize, is an important aspect of a classifier. Given a finite and noisy data set, different classifiers typically give different generalizations. For instance, we have built two detection systems: one using Gabor filter as feature extraction method and SVM as classifiers (GaborSVM), and the other Haar Wavelet decomposition together with SVMs (HaarSVM). Although, they produce similar accuracy on our test data, the sets of pattern misclassified are not completely overlap. This observation suggests that different classifiers, offer complementary information about the patterns to be classified, which could be used to improve the overall performance of our vehicle detection system. This active research topic is formally known as Decision Fusion or Classifier Combination. Decision fusion approaches are particularly useful for difficult problems, such as those that involve a large amount of noise, limited number of training data or unusually within class varieties. As we discussed before, on-road vehicle detection instantiates those difficulties.

Many approaches for fusing different classifiers have been proposed in the literature. Xu et al. [2] combined multiple classifiers using linear combination and applied this scheme on a handwriting recognition problem. Cho et al. presented a fuzzy integral based multiple neural network combining scheme in [3], where not only the objective evidence provided by various classifiers, but the system’s expectation of the importance of that evidence were considered in the fusion process. Rogova [4] investigated the combining several Neural Network classifiers using the

Dempster-Shafer theory of evidence. Kittler et al. [5] developed a common theoretical framework for classifier combination based on Bayesian theory and demonstrated its effectiveness on M2VTS database and a handwritten digit data set.

In this paper, a decision fusion based vehicle detection method is proposed and tested on our vehicle detection data sets. The proposed method outperforms the popular probabilistic rules[5]. The reasons that the proposed simple fusion rule is better than probabilistic rules are: First, our fusion rule takes advantage of the physical meaning of the classifier outputs, which are some kind of distance measures. While the probabilistic rules discard those physical meaning implicitly when they manage to fit a distribution on the classifier outputs. Second, in order to avoid the intractable high order density function, people usually assume that the class-conditional densities are independent and follow Gaussian distribution. These two assumptions do not usually hold. We have statistically demonstrated that neither of these two assumptions hold in our vehicle detection application, though the histograms of the classifiers' outputs do look like normal distribution. In contrast, our fusion method doesn't depend on any assumptions, where it gains its strength.

2 Vehicle Detection Systems

We have developed two vehicle detection systems - one using Gabor filter as feature extraction method and SVM as classifiers (GaborSVM), see Section 2.3, and the other Haar Wavelet decomposition together with SVMs (HaarSVM), which is described in Section 2.2. A brief introduction of SVMs is provided in Section 2.1. Both of the two systems learn the characteristics of the vehicle and non-vehicle classes from a set of training images, and are evaluated on the same set of test images. The description of our data sets can be found in Section 2.4.

2.1 SVMs

SVMs are primarily two-class classifiers that have been shown to be an attractive and more systematic approach to learning linear or non-linear decision boundaries [6] [7]. Given a set of points, which belong to either of two classes, *SVM* finds the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. This is equivalent to performing structural risk minimization to achieve good generalization [6] [7]. In most applications, the SVM outputs are thresholded (i.e., "-1" or "+1"). In this paper, we do not do any thresholding; instead, we use the actual outputs, which are essentially "distances" in a high-dimensional space.

2.2 HaarSVM

Wavelets capture visually plausible features of the shape and interior structure of objects. Wavelet features at different scales capture different levels of detail. Coarse scale features encode large regions while fine scale features describe smaller, local regions. All these features together disclose the structure of an object in different resolutions.

In this system, we use Haar wavelet decomposition. The wavelet decomposition coefficients are used as our feature directly. Each of the subimages is scaled to 32x32 and then a 5 level Haar wavelet decomposition is performed on it, which yields 1024 coefficients. We do not keep the coefficients in the HH subband of the fist level since it encodes mostly noise. The final feature set contains 768 coefficients.

2.3 GaborSVM

Gabor filters provide a mechanism for obtaining some degree of invariance to intensity due to global illumination, selectivity in scale, as well as selectivity in orientation. Essentially, they are orientation and scale tunable edge and line detectors. Vehicles do contain strong edges and lines at different orientation and scales, thus, these features are very powerful for vehicle detection. The general functional $g(x, y)$ of the two-dimensional Gabor filter family can be represented as a Gaussian function modulated by an oriented complex sinusoidal. Gabor filters act as local bandpass filters. In our system, we use the Gabor filter design strategy described in [8]. Given an input image $I(x, y)$, Gabor feature extraction is performed by convolving $I(x, y)$ with a Gabor filter bank. We use Gabor features based on moments, extracted from 9 overlapped subwindows of the input image.

2.4 Dataset

The images used in our experiments were collected in Dearborn, Michigan during two different sessions, one in the Summer of 2001 and one in the Fall of 2001, using a low-light camera. To ensure a good variety of data in each session, the images were caught during different times, different days, and on five different highways. The training set contains subimages of rear vehicle views and non-vehicles which were extracted manually from the Fall 2001 data set. A total of 1051 vehicle subimages and 1051 non-vehicle subimages were extracted by several students in our lab. Although specific instructions were given to the students, there is some variability in the way the subimages were extracted. For example, certain subimages cover the whole vehicle, others cover the vehicle partially, and others contain the vehicle and some background (see Figure 1). We have not attempted to align the data in our case since alignment requires detecting certain features on the vehicle accurately. Moreover, we believe that some variability in the extraction of the subimages can actually improve performance. Light correction and histogram equalization were utilized to preprocess the images.

To evaluate the performance of the proposed approach, the average error (*ER*), false positives (*FPs*), and false negatives (*FNs*), were recorded using a three-fold cross-validation procedure. For testing, we used a fixed set of 231 vehicle and non-vehicle subimages which were extracted from the Summer 2001 data set.

Figure 1: Subimages for training.

3 Probabilistic Fusion

For a general $m-$class pattern classification problem, the ultimate goal is to assign a query pattern, z, to one of the m possible classes $(\omega_1, \cdots, \omega_m)$ with minimum risk. If we put $m-$class pattern classification problem in the N-classifier decision fusion context, the final decision, which class the pattern z belongs to, is based upon the compromise among the N classifiers. Let us assume that x_i, the output of applying ith classifier on pattern z, is a scalar for simplicity. The outputs of applying ith classifier on patterns from class ω_k can be modelled by a probability density function $p(x_i|\omega_k)$, and its prior probability $P(\omega_k)$. For any pattern, the outputs of applying the N classifiers form the output vector $\mathbf{X} = [x_1, x_2, \cdots, x_N]$. According to Bayesian theory, given classifier output vector $\mathbf{X}$, the query pattern z should be assigned to class ω_j provided the a posteriori probability of that interpretation is maximum [5], i.e.,

$$assign\ z \longrightarrow \omega_j \quad if$$

$$P(\omega_j|\mathbf{X}) = \max_k (P(\omega_k|\mathbf{X}) \tag{1}$$

In order to assign a pattern to a class by utilizing Eq.1, we need to know the high-order measurement statistics described in terms of joint probability density function $p(x_1, \cdots, x_N|\omega_k)$,which is intractable in practice. Rewriting the posterior probability using Bayes' theorem, we have

$$P(\omega_k|x_1, \cdots, x_N) = \frac{p(x_1, \cdots, x_N|\omega_k)P(\omega_k)}{p(x_1, \cdots, x_N)} \tag{2}$$

where $p(x_1, \cdots, x_N)$ is the unconditional measurement joint probability density and can be described as:

$$p(x_1, \cdots, x_N) = \sum_{j=1}^{m} p(x_1, \cdots, x_N|\omega_k)P(\omega_j) \tag{3}$$

Assuming that the outputs of the classifiers are conditionally statistically independent, we can write the joint distribution as

$$p(x_1, \cdots, x_N|\omega k) = \prod_{i=1}^{N} p(x_i|\omega_k) \tag{4}$$

Thus the fusion rule can be expressed as:

$$assign\ z \longrightarrow \omega_j \quad if$$

$$P(\omega_j)\prod_{i=1}^{N} p(x_i|\omega_j) = \max_k P(\omega_k)\prod_{i=1}^{N} p(x_i|\omega_k) \tag{5}$$

By assuming equal prior, we end up with:

$$assign\ z \longrightarrow \omega_j \quad if$$

$$\prod_{i=1}^{N} p(x_i|\omega_j) = \max_k \prod_{i=1}^{N} p(x_i|\omega_k) \tag{6}$$

In terms of the a posteriori probabilities, Eq.6 is equivalent to:

$$assign\ z \longrightarrow \omega_j \quad if$$

$$\prod_{i=1}^{N} P(\omega_j|x_i) = \max_k \prod_{i=1}^{N} P(\omega_k|x_i) \tag{7}$$

Under the assumption that the posterior probabilities computed by the respective classifiers will not deviate dramatically from the prior probabilities, the most commonly used Sum rule is derived in[5] as:

$$assign\ z \longrightarrow \omega_j \quad if$$

$$\sum_{i=1}^{N} p(x_i|\omega_j) = \max_k \sum_{i=1}^{N} p(x_i|\omega_k) \tag{8}$$

In terms of the a posteriori probabilities, we obtain the decision rules:

$$assign\ z \longrightarrow \omega_j \quad if$$

$$\sum_{i=1}^{N} P(\omega_j|x_i) = \max_k \sum_{i=1}^{N} P(\omega_k|x_i) \tag{9}$$

Since we know $p(x_i|\omega_k)$, $k = 1 \dots N$, Bayes' rule can be used to compute the posterior probability via:

$$P(\omega_k|x_i) = \frac{p(x_i|\omega_k)}{\sum_{k=1}^{N} p(x_i|\omega_k)} \tag{10}$$

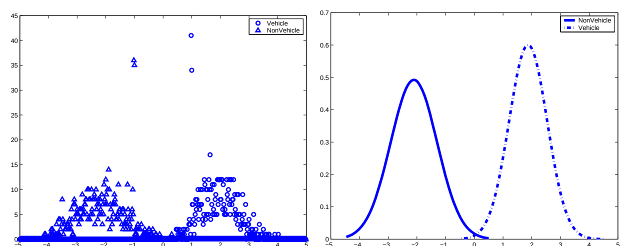
Eq.6-9 have been used extensively in decision fusion area.

4 Why Probabilistic Decision Fusion Rules Fail in Vehicle Detection

4.1 Classifier Outputs

By using Eq. 6- Eq. 8, we implicitly discard the actual physical meaning of the classifier outputs. The output of some classifier are essentially a "distance" measure between a query pattern and the decision boundary. Density function is only a relative frequency measure, and $p(x|\omega_k)$ indicates the "chance" that we are going to get the particular output value x given a query pattern z is from class ω_k. Traditionally, we consider this "chance" as the degree of confidence if we assign that z to ω_k. This is correct from the view point of pure statistics, not from the physical meaning of classifier outputs. Here is an example. In our vehicle detection context, we have two classes, vehicle class - ω_1 and non-vehicle class - ω_2. And the classifier $GaborSVM$ (see Section 2.3 for details) models the boundary between this two classes in high dimensional space. Fig. 2.a shows the histogram of the classifier's outputs, for simplicity, people tend to model the outputs for the two classes as two Gaussian distributions: $N(1.7019, 0.5427)$ and $N(-2.1048, 0.6567)$(Fig. 2).b. We'll show that the normality assumption doesn't hold in Section 4.2, even though Fig.2.a do look like Gaussian distribution.

We have four distinct query patterns, see Fig. 3.a-d, and outputs we get using $GaborSVM$ are $x_a = 3.1929$, $x_b = 0.1956$, $x_c = 3.0018$ and $x_d = 0.3965$. As we mentioned, they are essentially the "distance" measures between the query patterns and the decision boundary. The outputs of the classifier indicate that Fig. 3(a) and Fig. 3(c) are far away from the decision boundary, while Fig. 3(b) and Fig. 3(c) pretty close to the boundary. That x_a is greater than x_b indicates that we can assign pattern Fig.3(a) to class ω_1 with much higher confidence than pattern Fig. 3(a). The visualization conforms these. However, $p(x_a|\omega_1)$ and $p(x_b|\omega_1)$ are fairly similar to each other, so are the $p(x_c|\omega_1)$ and $p(x_d|\omega_1)$. It's obvious that, by using Eq. 8 or Eq. 6, we end up with very close measurements. The posterior probabilities, computed using Eq. 10, are not as distinguishable as we expected, though slightly better than the density values. They are: $P(x_a|\omega_1) = 1$, $P(x_b|\omega_1) = 0.8849$,$P(x_c|\omega_1) = 1$, and $P(x_d|\omega_1) = 0.9640$. These fairly similar values imply that Eq. 6-9 don't appreciate physical meaning ("distance") of the classifier output.



(a) (b)

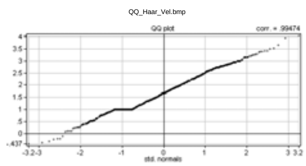
Figure 2: (a) Histogram of classifier outputs of GaborSVM; (b) Fitted density functions

(a) (b) (c) (d)

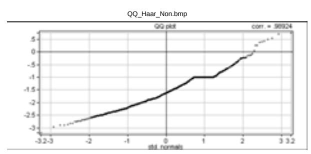
Figure 3: Patterns with similar $p(x|\omega)$ values: (a) $x = 3.1929$ and $p(x|\omega) = 0.0036$, (b) $x = 0.1965$ and $p(x|\omega) = 0.0033$, (c) $x = 3.0018$ and $p(x|\omega) = 0.0096$, (d) $x = 0.3965$ and $p(x|\omega) = 0.0093$

4.2 Do those assumptions hold?

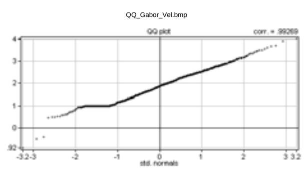
In order to derive the decision rules Eq.6-9, people make two assumptions: First, the class-conditional densities $p(x|\omega_k)$, $k = 1, 2$ are Gaussian, and class-conditional distributions are independent. Conditional independence assumption is unrealistic in many situations, though, because of the intractable higher order joint density fusions, people accept this assumption routinely in practice. From statistical point of view, the cross-correlation fusion (CCF) [12] is a tool used to measure dependence. We plug our data in and end up with $\rho = 0.376$, which is statistically significant and indicates that our data are dependent. Therefore, assuming that the class-conditional density function are independent is not proper.



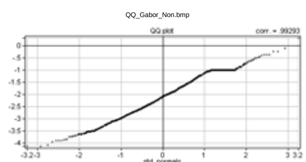
(a)



(b)



(c)



(d)

Figure 4: Q-Q plot of: (a) Outputs of HaarSVM on vehicle data, (b) Outputs of HaarSVM on nonvehicle data, (c) Outputs of GaborSVM on vehicle data, (d) Outputs of GaborSVM on nonvehicle data.

From the visualization of the histograms illustrated in Fig.2, people might be convinced the normality of the data. However, it is risky to assume a normal distribution, when the real distribution is not. There are a number of methods that can be used to check the deviations of the data from the normal distribution. The most useful tool for assessing normality is a quantile quantile or QQ plot [12]. It is a scatterplot with the quantiles of the scores on the horizontal axis and the expected normal scores on the vertical axis. A plot of these scores against the expected normal scores should reveal a straight line. Curvature of the points indicates departures of normality. Fig.4 shows that none of the four data follows Gaussian distribution, because each of them has a quite flat part on the QQ plot, i.e. each of them has a certain range where too many data fail in.

Hypothesis tests H_0-data is normal, are also carried out for all those four data sets. As we expected, all the four H_0 are rejected at the level of significance 0.01. QQ plots Fig.4 and Hypothesis tests disclose strong evidences that none of the data follows Gaussian distribution. We can see neither of the two assumptions(independence and normality) holds for our problem handy. This is one of the reason why probabilistic rules don't provide good performance.

5 Decision Fusion

The above observation suggests that ”distance” might be a good base for decision fusion. In this report, both of the two methods (GaborSVM and HaarSVM) use SVMs as classifier, which provides normalized distance measure. We propose using the following decision rule:

$$sign(x_1 + x_2) \quad = \quad \begin{cases} 1 & vehicle \\ -1 & nonvehicle \end{cases} \qquad (11)$$

where x_1 and x_2 are outputs by applying GaborSVM and HaarSVM on a same query pattern, $sign$ is the signum function. Base on Eq.11, the detection results will depend on the output of two detection system whichever has the bigger absolute output value. This is reasonable because a bigger absolute output value implies higher confidence level, since the outputs are from SVM.

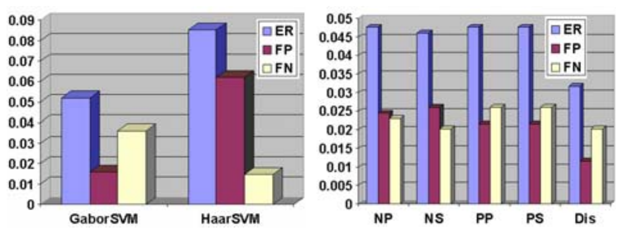
6 Experimental Results

We have performed a number of experiments and comparisons to demonstrate the performance of the proposed approach. First, GaborSVM and HaarSVM methods are evaluated using the test data individually. Fig.5.a shows the error rates, as well as the FP/FN rates. As we mentioned in Section 2.4, all the results reported in this report are the average rates using the three-fold cross-validation procedure. Overall the error rate of the GaborSVM is 5.19%, which is better than 8.52% obtained using HaarSVM method. However, HaarSVM has lower FN (1.47%) than GaborSVM does (3.61%). Therefore, there are patterns where HaarSVM does the correct classification, while GaborSVM doesn't, even though the overall performance of GaborSVM is better than that of HaarSVM. And this is also the justification of applying fusion methods.

Our literature review in Section1 shows that the Product rule Eq.6-7 and Sum rule Eq.8-9 have been used quite extensively for decision fusion. For comparison purposes, we have evaluated the performances of the Sum and Product rule in the context of vehicle detection. Let's refer the Sum rule using the density function directly Eq.8 to as method NS, and Sum rule using the posterior probability Eq.9 as method PS, the Product rule using the density function directly

Eq.6 as NP and the Product rule using the posterior probability Eq.7 as PP. The average error rates, as well as FP/FN, are illustrated in Fig.5.b. We can see the performances of these four probabilistic rules are fairly close to each other: error rates from NS is 4.61%, and error rates from all other three rules (NP, PP, and PS) are 4.76%. Compared to the GaborSVM and HaarSVM, we can see the probabilistic fusion methods only improve the performance a little bit, give the GaborSVM's error rate is 5.19%.

The performance of the proposed method is shown in the Fig.5.b. The average error rate is 3.17%, which is about two percent better than the GaborSVM method.



(a) (b)

Figure 5: (a) Gabor filter bank with 3 scales and 5 orientations; (b) Gabor filter bank with 4 scales and 6 orientations

7 Conclusions and Future Work

We have considered the applying decision fusion approaches on the vehicle detection problem. In particular, we investigated popular Sum and Product rules, and we found that the physical meaning of the classifier outputs, which are ”distance measurement”, were discarded implicitly by using these rules. We believe that ”distance measurement” are very helpful in the decision fusion context, because it reflects the confidence level. We also demonstrate that that two basic assumptions of the probabilistic fusion rules don't hold for our vehicle detection problem. Motivated by these observations, we define a simple fusion rule, which takes advantage of the ”distance” property of the classifier outputs and doesn't depend on any assumptions. Our experimental results demonstrate that our simple fusion methods give better performance in the context of vehicle detection than the commonly used Sum or Product rule.

References:

[1] G. Bebis Z. Sun and R. Miller, “On-road vehicle detection: A review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, 2006.

[2] A. Krzyzak L. Xu and C. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” *IEEE trans. Systems, Man, and Cybernetics*, vol. 22, pp. 418–435, 1992.

[3] S. Cho and J. Kim, “Combing multiple neural networks by fuzzy integral for robust classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, pp. 380–384, 1995.

[4] G. Rogova, “Combing the results of several neural network classifers,” *Neural Networks*, vol. 17, pp. 777–781, 1994.

[5] R. Duin J. Kittler, M. Hatef and J. Matas, “On combining classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.

[6] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.

[7] C. Burges, “Tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 955–974, 1998.

[8] B. Manjunath and W. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[9] C. Papageorgiou and T. Poggio, “A trainable system for object detection,” *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.

[10] S. Uthiram G. Bebis and M. Georgiopoulos, “Face detection and verification using genetic search,” *International Journal on Artificial Intelligence Tools*, vol. 9, no. 2, pp. 225–246, 2000.

[11] S. Shah and J. Aggarwal, “Multiple feature integration for robust object localization,” in *Proceedings of Computer Vision and Pattern Recognition*, 1998, pp. 1014–1020.

[12] R. Shumway and D. Stoffer, *Time series analysis and its applications*, Springer, 2000.