Learning Exemplar-Based Categorization for the Detection of Multi-View Multi-Pose Objects

Ying Shan Feng Han Harpreet S. Sawhney Rakesh Kumar
Sarnoff Corporation
201 Washington Road
Princeton, NJ 08540, USA
{yshan, fhan, hsawhney, rkumar}@sarnoff.com

Abstract

This paper proposes a novel approach for multi-view multi-pose object detection using discriminative shape-based exemplars. The key idea underlying this method is motivated by numerous previous observations that manually clustering multi-view multi-pose training data into different categories and then combining the separately trained two-class classifiers greatly improved the detection performance. A novel computational framework is proposed to unify different processes of categorization, training individual classifier for each intra-class category, and training a strong classifier combining the individual classifiers. The individual processes employ a single objective function that is optimized using two nested AdaBoost loops. The outer AdaBoost loop is used to select discriminative exemplars and the inner AdaBoost is used to select discriminative features on the selected exemplars. The proposed approach replaces the manual time-consuming process of exemplar selection as well as addresses the problem of labeling ambiguity inherent in this process. Also, our approach fully complies with the standard AdaBoost-based object detection framework in terms of real-time implementation. Experiments on multi-view multi-pose people and vehicle data demonstrate the efficacy of the proposed approach.

Figure 1. How to detect object in real-time with the amount of variability shown in this picture? Previous work using AbaBoost-based object detection algorithms required manually categorizing multi-view multi-pose training data into pre-defined categories and combining two-class classifiers trained individually to form a strong classifier. These work will have difficulty handling the above training data set because manual categorization is time-consuming and inherently ambiguous in both defining the categories and assigning samples to each category. Being able to address these issues while still comply with the real-time AdaBoost framework is the goal of this paper.

1. Introduction

The AdaBoost based method presented by Viola and Jones for face detection demonstrated a novel approach to real-time object detection. The use of AdaBoost in the traditional methodology is good at handling single view, relatively rigid objects such as frontal faces, but is inaccurate in detecting objects such as people and vehicle in their general poses and viewing aspects. Many researchers, including Viola and Jones, have noticed that by manually clustering training data into pre-defined intra-class categories of aspects and other variations, by training individual two-class classifiers for each category, and combining the individual classifiers into a single strong classifier greatly helps to improve the detection performance. However, the manual delineation of objects into training sets for various poses etc. puts a strong limitation on the overall efficacy of this approach. For instance, for a training set with the degree of variability shown in Fig. 1, the manual categorization process is inherently ambiguous, and may take significant amount of time. Moreover, the errors caused by improperly defined categories and incorrectly assigned labels will eventually be propagated into the final classifier and deter the

object detection performance. Addressing the automated categorization issues during the training phase while still meeting the real-time constraint during the detection phase is a challenging problem.

In this paper, we propose a novel computational framework that unifies automatic categorization, through training of a classifier for each intra-class exemplar, and the training of a strong classifier combining the individual exemplar-based classifiers with a single objective function. The function is optimized using two nested AdaBoost loops. The outer AdaBoost loop is used to select discriminative exemplars, each of which is conceptually equivalent to a manual category, and best represents a class of training samples that are similar according to certain distance measures. The inner AdaBoost loop is used to select discriminative features on the selected exemplars. The latter is similar to standard AdaBoost. The proposed approach replaces the manual time-consuming categorization process as well as addresses the problem of labeling ambiguity inherent in the process. Also, since the overall framework complies with the original AdaBoost-based object detection framework, our approach inherits the computational advantages of the standard approach.

2. Related Work

Multi-view multi-pose object detection is an active area of research. Viola et. al. [18] used a decision tree structure to extended their seminal work in [17] to cope with the detection of multi-view faces, while Li et. al. [7] proposed a pyramid-structured multi-view face detector. Mohan et. al. [8] and Shashua et. al. [13] proposed to construct component detectors and combine them into a single strong classifier for the detection of pedestrians with large aspect and pose variations. Huang et. al. [4] proposed a rotation invariant multi-view face detector using vector boosting. A common problem with these approaches is that they all required significant amount of manual categorization of the training data. Since the training data set is large, and is usually collected in uncontrolled environments, manual categorization can become prohibitively expensive especially with the increase in object variability and the number of object classes. Moreover, because of the fundamental ambiguity in labeling different poses and viewing aspects, manual categorization is also an error-prone procedure that may introduce significant bias into the training process. Recently, Tu [16] proposed a decision tree-based probabilistic boosting algorithm that naturally embedded the clustering into the learning process. This approach is designed to be a general multi-class classifier that handles large in-class variations. However, its success on multi-view object detection experiments is uncertain at best. As compared with Tu's work, our approach simulates a proven training pipeline while extending it for exemplar selection, and hence the results are more predictable.

The proposed approach is closely related to the shape exemplar-based methods for object classification and tracking. A representative is that of Gavrila et. al. [3], where a hierarchy of edge maps are used to model the different viewing aspects of pedestrian and road sign shapes. Toyama et. al. [14] used edge-based shape exemplars as the centers of a Gaussian mixture model, coined as the Metric Mixture Model, for pedestrian tracking. Song et. al. [15] used edge-based vehicle exemplars for vehicle segmentation and tracking. The shape exemplar methods have successfully demonstrated that the shape exemplar-based approach, combined with carefully selected match measures, is able to model pose and aspect changes, and is resilient to scene clutter and illumination changes. A key issue in the exemplar-based approach is the optimal selection of exemplars. Previous approaches addressed this problem by employing hierarchical clustering [3], the EM algorithm [14], and nearest neighbor representations [15]. For the object detection tasks, the exemplars thus selected are not discriminative. Specifically, the exemplar selection is not designed to select exemplars to maximize the discrimination between the objects of interest and the background (or other classes). A major contribution of this paper is the integration of exemplar selection within the training process of the classifier, where both object samples and background samples compete with each other to maximize the classification performance. This is in spirit similar to the idea of Learning Vector Quantization (LVQ) [6] proposed by Kohonen. Unlike LVQ, for which there is no proof of the convergence, our training process is based on the computational framework of AdaBoost, and therefore the convergence is guaranteed.

Other related works include Wu et. al.'s statistical field model [19] where object model and background model are trained separately and applied to pedestrian detection using mean field approximation. Similar to [19, 3, 14, 15], our approach also uses edge maps during the exemplar selection stage. Many previous works use clean edge maps for at least one of the edge maps. Truncated Chamfer distance [5] or robust Hausdorff distance [9] may work for these cases, but are less effective for the cases when both edge maps are not clean and there is significant clutter in the scene. Our approach uses the robust edge-based distance proposed in [12] that has been proved to work with the cases when both query and model are not clean.

3. Problem Statement and Notations

Let $\mathcal{B} = \{(I_1, y_1), \cdots, (I_l, y_l)\}$ be a set of training samples, where I_i is the ith sample image, $y_i = \{-1, 1\}$ is the class label of the image, and $l = m + n$ is the number of training samples including m positives and n negatives. We want to find an optimal classifier by minimizing an expo-

nential loss function L over the training set $\mathcal{B}$,

$$\min_{\{\alpha_t,\Theta_t\}_1^T} \sum_{i=1}^{l} L\left(y_i, F(I_i;\{\alpha_t,\Theta_t\}_1^T)\right), \quad (1)$$

where the additive model

$$F(I;\{\alpha_t,\Theta_t\}_1^T) \equiv \sum_{t=1}^{T} \alpha_t f(I;\Theta_t), \quad (2)$$

is used to map an image I to its corresponding class label, and the exponential loss function L is defined as

$$L(y, F(I)) = \exp(-y\ F(I)). \quad (3)$$

The parameter α_t in (1) is the weight of the t^{th} basis function F, and Θ_t is defined as

$$\Theta_t \equiv \{E_t, \mathcal{S}_t, \Gamma_t, \Lambda_t\}, \quad (4)$$

where E_t is the t^{th} exemplar - a representative training image, $\mathcal{S}_t$ is the set of features selected from E_t, Γ_t is the set of weights corresponding to each feature in $\mathcal{S}_t$, and Λ_t is the set of the parameters for each classifier constructed based on corresponding features in $\mathcal{S}_t$. The function f in (2) is defined as

$$f(I;\Theta_t) \equiv \sum_{\tau=1}^{T_t} \gamma_\tau^t\ g(I; s_\tau^t, \lambda_\tau^t), \quad (5)$$

where T_t is the number of features in E_t, g is the classifier constructed from each feature $s_\tau^t \in E_t$, $\gamma_\tau^t \in \Gamma_t$ is the weight of g, and $\lambda_\tau^t \in \Lambda_t$ is the set of parameters for g. From (5) it is clear that function $f(I;\Theta_t)$ is indeed an exemplar-based classifier that combines features selected from the exemplar as a weak classifiers. This is different from Viola's original work [17] where each weak classifier f is computed from a single Haar feature.

The rationale for using the exponential loss function in (1) and the additive model for classification is well-established in the AdaBoost literatures such as [10] and [2]. The basis function f, defined later, is called a "weak classifier" in this context. As compared with other state-of-art classification algorithms such as SVM, Viola et. al.'s work in [17] demonstrated additional advantages of using AdaBoost for object detection, i.e., effective feature selection and real time classification.

The key contribution in the above formulation is to introduce the concept of exemplar E_t - the equivalence of manual category - and integrate it into a unified computational framework so that the categorization process can be automated.

4. Learning Classifier

Following the problem formulation in Sec. 3, the goal of the training process is to determine the optimal parameter set $\{\alpha_t,\Theta_t\}_1^T$. Since the objective function in (1) contains two nested additive models F and f, the problem can be naturally solved with two nested AdaBoost procedures. More specifically, we use the outer AdaBoost to select discriminative exemplars and combine them into the final strong classifier F as in (1), and use the inner AdaBoost to select image features for each exemplar and combine them into an exemplar-based classifier f as in (5). In the following discussion, we will refer to the outer AdaBoost (detailed in Algorithm 1) as the "strong classifier", and the inner AdaBoost (detailed in Algorithm 2) the "exemplar-based weak classifier" or simply the "weak classifier".

4.1. Learning Strong Classifier and Discriminative Exemplars

Algorithm 1 Learning Strong Classifier and Discriminative Exemplars

Input: Candidate exemplar set $\mathcal{B}_c = \{(I_j^c, y_j^c)\}$, and sample set $\mathcal{B}_s = \{(I_i^s, y_i^s)\}$, where $\mathcal{B}_c \cup \mathcal{B}_s = \mathcal{B}$.

1: Initialize sample weights $w_{1,i} = \frac{1}{2m_s}, \frac{1}{2n_s}$, for $y_j^s = 0, 1$ respectively, where m_s and n_s are the number of positives and negatives respectively.
2: **for** $t = 1, \cdots, T$ **do**
3: Normalize the weights: $w_{t,i} \leftarrow w_{t,i} / \sum_{j=1}^{l} w_{t,j}$
4: **for** each candidate exemplar $c = 1, \cdots, l_c$ **do**
5: Train an exemplar-based classifier $f(I;\Theta_c)$ as in Algorithm 2.
6: Compute error rate $\epsilon_c = \sum_i w_{t,i} |f(I;\Theta_c) - y_i^s|$.
7: **end for**
8: Choose $f(I;\Theta_t)$ to be the classifier with the lowest error ϵ_t
9: Update the weights: $w_{t+1,i} \leftarrow w_{t,i}\beta_t^{1-e_i}$, where $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$, and $e_i = 0, 1$ for incorrect classification and correct classification respectively.
10: **end for**

Output: The set of discriminative exemplars $\mathcal{E} = \{E_t\}_1^T$, and the strong classifier $\sum_{t=1}^{T} \alpha_t f(I;\Theta_t)$, where $\alpha_t = \log \frac{1}{\beta_t}$.

As shown in Algorithm 1, the input of the outer AdaBoost is a candidate exemplar set $\mathcal{B}_c$ and a sample set $\mathcal{B}_s$. The samples in $\mathcal{B}_c$ are randomly selected and removed from the original sample set $\mathcal{B}$, and $\mathcal{B}_s$ contains the remaining samples. The output of this algorithm is the strong classifier as in (1) and the discriminative exemplar set $\mathcal{E}$, which is a subset of $\mathcal{B}_c$.

Steps from 1 to 3 and 8 to 10 are the standard AdaBoost steps initializing and updating sample weights, and combin-

ing the weak classifiers into a strong classifier according to the training error of the best weak classifier f at each iteration t. Steps from 4 to 7 iterate through all the candidate exemplars, compute a weak classifier based on each exemplar, and compute the training error rate for each weak classifier.

It is clear from Algorithm 1 how the parameters α_t in (1) are computed, and hypothetical exemplars in (4) are generated and selected. The remaining parameters in Θ_t are related to individual image features of each exemplar, and are computed with the inner AdaBoost detailed in Algorithm 2.

4.2. Learning Exemplars-Based Weak Classifier

Algorithm 2 builds a weak classifier with the image features in the c^{th} candidate exemplar E_c in the 4^{th} step of Algorithm 1. Note here the slight notation change of an exemplar. We use E_c, which is a hypothetical exemplar, instead of E_t in the precious section, which is the optimal exemplar selected at the t^{th} iteration in Algorithm 1. Algorithm 2 is similar in formality to the standard AdaBoost approach proposed in [17], except one major difference. In the standard AdaBoost algorithm, the weak classifier is trained based on image features extracted from each individual training image. Instead, the classification function g in our approach is trained based on the distances d_b between features on the exemplar and their corresponding features on the training samples. The output of this process is the exemplar-based classifier $f(I; \Theta_c)$ for the hypothetical E_c.

Algorithm 2 Learning Exemplar-based Classifier

Input: Exemplar E_c selected in the 4^{th} step of Algorithm 1, and the sample set $\mathcal{B}_s = \{(I_i^s, y_i^s)\}$.

1: Initialize sample weights $w'_{1,i} = \frac{1}{2m_s}, \frac{1}{2n_s}$.
2: **for** $\tau = 1, \cdots, T_c$ **do**
3: Normalize the weights: $w'_{\tau,i} \leftarrow w'_{\tau,i} / \sum_{j=1}^{l} w'_{\tau,j}$
4: **for** each feature $b = 1, \cdots, O_c$ of the exemplar **do**
5: Train a classifier $g(I; s_b^c, \lambda_b^c)$ based on the distances d_b from the exemplar feature s_b^c to the corresponding features of all the samples in $\mathcal{B}_s$.
6: Compute $\epsilon'_b = \sum_i w'_{\tau,i} |g(I; s_b^c, \lambda_b^c) - y_i^s|$.
7: **end for**
8: Choose $g(I; s_\tau^c, \lambda_\tau^c)$ to be the classifier with the lowest error ϵ'_τ.
9: Update the weights: $w'_{\tau+1,i} \leftarrow w'_{\tau,i} {\beta'_t}^{1-e_i}$, where $\beta'_\tau = \frac{\epsilon'_\tau}{1-\epsilon'_\tau}$.
10: **end for**

Output: The exemplar-based classifier for E_c as in (5): $\sum_{\tau=1}^{T_c} \gamma_\tau^c \, g(I; s_\tau^c, \lambda_\tau^c)$, where $\gamma_\tau = \log \frac{1}{\beta'_\tau}$.

4.3. Insights behind the Algorithm

We now do a simple thought experiment to see why the proposed algorithm is simulating the manual categorization process as in [18, 8, 13, 4]. Suppose that our training set consists of $\{\mathcal{F}_f \cup \mathcal{F}_s, \mathcal{N}\}$, where $\mathcal{F}_f$ and $\mathcal{F}_s$ are the positive face samples for frontal view and side view, respectively, and $\mathcal{N}$ is the non-face background samples. To train a face detector, the manual approaches will essentially train a frontal face detector with $\{\mathcal{F}_f, \mathcal{N}\}$ and a side face detector separately with $\{\mathcal{F}_s, \mathcal{N}\}$, and then combine them into a single detector. With our approach, a set of exemplars will be automatically selected. Suppose that one exemplar is selected for the frontal faces. Any effective features on this exemplar will have relatively small distances d_b to the corresponding features on the frontal faces, while having large distances to those on the side faces and background images. As a result, when training the classifier based on the frontal exemplar features, the side view faces will act like background images. The classifier thus trained is then similar to the frontal face detector computed by the manual approach, except that the training set becomes $\{\mathcal{F}_f, \mathcal{F}_s \cup \mathcal{N}\}$, i.e., the background sample set is now augmented with the side view face sample set. Similar reasoning also applies to the training of the side view face detector. The exemplars may not have explicit semantic significance as for the manually selected categories, but they are meaningful from the viewpoint of classification - they are selected by minimizing the loss function in (1). Consequently, one of the major advantages of our approach is to replace the inherently ambiguous and error-prone manual categorization with a principled and automated categorization process.

5. Implementation Details

Depending on the applications, the general framework provided by the proposed approach can be customized by changing features, distance measures, and different design philosophies. We present in this section two representative approaches and discuss their pros and cons from a practical viewpoint. These two approaches differ in the training of the exemplar-based classifier. The first approach follows the design philosophy of using a single strong classifier for each exemplar, while the second uses multiple weak features. The first approach is flexible in choosing different distance measures but is computationally expensive. The second approach has computational cost as in the original AdaBoost approach, but is less flexible in selecting the types of features.

5.1. Classification Using Exemplars with Single Strong Feature

During the past several decades, the object and pattern recognition community has successfully developed several

features and associated distance measures such as SIFT, Earth Mover's Distance (EMD), shape context, and Chamfer distance. When integrated with our approach, these provide a powerful classification mechanism in the presence of illumination changes, local deformations, and scene clutter. Our implementation specifically uses edge map as the single "super feature" for each exemplar, and computes distance $d = 1-\gamma$ between an exemplar and a training sample. The scalar score γ is defined as:

$$\gamma = \frac{\sum_{A \mapsto BI} h(d^p, \delta) h(a^p, \alpha) + \sum_{B \mapsto A} h(d^p, \delta) h(a^p, \alpha)}{N(A) + N(B)}, \tag{6}$$

where $N(A)$ and $N(B)$ are the numbers of edge pixels of the edge maps A and B, $\gamma \equiv \gamma_{A,B}$, $h(x, c) = (1 - |x|/c)$ for $|x| < c$, $h(x, c) = \rho$ for $|x| \geq c$, ρ is a small positive number, and d^p and a^p are the pixel-wise distance and angle difference between a pair of closest points from A and B. The constants δ and α can either be predefined or statistically computed from the training databy estimating the inlier and outlier processes as in [12].

The exemplar-based classifier $f(I; \Theta)$ is a simple "stump" trained based on the distances between the exemplar and the training samples. The threshold for f is also automatically determined from the training data. The final output of the training process is a classifier combining the discriminative edge-based classifier and its corresponding exemplars, which are in concept similar to the manual categories. This approach is in some degrees similar to [3, 14]. The major difference is that our exemplars are selected to minimize the classification error and are therefore more "discriminative". With the help of the distance transform, the computation of robust distance measure as in (6) can be accelerated. Given a hypothesized region of a target object, computing distances against multiple exemplars in the classifier can be accelerated by pre-aligning exemplar edgemaps and performing a batch matching process as in [11]. Following the idea in [13], a focus of attention component is used to speed up the overall object detection process while taking the advantage of the discriminative power provided by the proposed approach.

5.2. Classification Using Exemplars with Multiple Weak Features

This approach is designed to comply with the original AdaBoost algorithm [17]. For each exemplar, we partition it into a grid of regions, and compute the average gradient direction as in [19, 1] over each region as a feature. Only those features whose corresponding regions contain strong edges are retained for feature selection in the inner loop. The exemplar-based classifier $f(I; \Theta)$ combines feature-based weak classifiers g, which are trained based on the distance between the exemplar and the training samples for each feature. The threshold for each "stump" g is learned from the train data. The major advantage of this approach is that computationally it is comparable with the original AdaBoost approach since run-time feature extraction is similar. Moreover, the increase in the total number of features is limited due to the fact that the use of exemplar reduces the complexity of the each individual classifier $f(I; \Theta)$ and hence reduces the number of features needed for each exemplar-based classifier. Note that other features such as in [16] can also be integrated within our approach, and computed quickly during run-time since they comply with the integral image structure proposed in the original AdaBoost approach. As compared with the single feature approach, the selection of feature type is not as flexible as in the previous approach.

6. Experiments

The algorithm proposed in this paper is tested with different data sets with large in-class variations, and is compared with some existing AdaBoost-based algorithms. Since auto-categorization with discriminative exemplars idea is the key claim of the paper, our effort has been focused on just proofing the efficacy of this idea using a single Nested Adaboost, instead of cascading it into a full system. For a fair comparison, the two competing algorithms in our experiments are set to be single-layered. The standard AdaBoost uses OpenCV implementation, and the BPT algorithm uses the original implementation from the inventor. Ideally, we need to compare our results with manually categorization-based AdaBoost algorithms. However, it is hard to justify the label correctness for the databases with the amount of variability.

The people database is publicly downloadable from an MIT site[1], and the vehicle database is our private collection. Since end-to-end comparison was not the goal, these databases were sufficient to achieve our goals. In order for the training results to be meaningful also to real applications, all the negative samples are selected from a focus of attention mechanism. As in [13], the focus of attention algorithm is a classifier based on simple image features and other domain knowledge to generate target object hypotheses. The typical number of hypotheses generated is in the range of 50-100 for each image. In the following discussion, we will refer to the approach discussed in 5.1 as the *single feature exemplar-based classifier*, denoted as SFEC. The approach given in 5.2 as the *multiple feature exemplar-based classifier*, denoted as MFEC. The standard AdaBoost algorithm will be denoted as SADB, and Tu's approach [16] is denoted as PBT, i.e., probabilistic boosting tree.

[1] http://cbcl.mit.edu/cbcl/software-datasets/PedestrianData.html

6.1. People Classification

Our people database contains 2000 standing people images with various aspects, poses, and illumination conditions. The resolution of each image is 64×128. To clearly illustrate the idea of the exemplar-based approach, we use the SFEC implementation discussed in Sec. 5.1 to start with. Since the edge map is the only one feature for SFEC and the feature selection process in Algorithm 1 is not invoked, any performance gain demonstrated by this approach can be attributed to the success of the exemplar-based idea, which is one of the main contributions of this paper. The people database $\mathcal{B}$ is separated into three groups of positive samples, where the random exemplar candidate set has 200 samples, the training set has 1000 samples, and the test set has 800 samples. The number of negative samples for training and test are 1000 and 800, respectively. These are the background samples selected from the focus of attention output.

Figure 3. Discriminative exemplars automatically selected with the SFEC algorithm. Only 18 out of 20 exemplars are displayed for better layout of the images. Also note that the actual exemplar representation is edge map. The original images are presented here for the illustration purpose.

Figure 2 shows the training error rate and testing error rate with respect to the number of selected exemplars, which is the number of iterations T of the outer AdaBoost. It can be seen that as T increases, the training errors (both miss detection and false alarm) decrease. The test errors also decrease when T increases from 5 to 20, but the miss detection does not decrease beyond $T = 20$. Figure 3 shows 18 out of the 20 exemplars selected by the SFEC algorithm. Visually, these exemplars nicely cover different aspects such as frontal, back, left and right sides, different poses, and different aspect ratios. Without automatic exemplar selection, it would be very difficult to perform manual categorization at this level of detail.

Table 1 compares the training and test errors for different approaches applied to exactly the same data set. The SFEC numbers are based on the results with 30 exemplars. Since candidate exemplars are part of the training data, we include them into the training data for other approaches for a fair comparison. As compared with the standard AdaBoost, SFEC algorithm has over 4 times less miss detections, and has comparable false alarm rate. The SFEC algorithm also outperforms Tu's PBT [16] by around 2 times less miss detection and false alarm rates.

Approach	SFEC	SADB	PBT	MFEC
MD	2.50	10.125	5.25	2.0
FA	2.0	1.75	3.75	1.25

Table 1. Test errors with different approaches. The SFEC results are based on 30 exemplars. The MFEC uses 25 exemplars, and each exemplar uses 30 features. The miss detection rate, denoted as MD, and the false alarm rate, denoted as FA, are all in percentages.

To test the limit of the SFEC algorithm, we applied a classifier with 20 exemplars trained based on the previous database to a difficult data set. Figure 4 show some examples of this dataset. These image samples are cropped out from the bounding boxes provided by a tracking algorithm. Regardless of heavy shadows, significant amount of shape distortion, and backgrounds never seen before, both the miss detection and false alarm rates increase only about 7%. This demonstrates the superior generalization capability of our approach.

The last column in Table 1 shows the error rates for the MFEC approach, which is slightly better than the SFEC approach. This is encouraging since the MFEC algorithm complies with the infrastructure for real-time object detection, and there are still a lot of rooms to improve the algorithm by properly selecting the type of features and refine the implementation details. The MFEC algorithm uses 25 exemplars , and each exemplar uses about 30 features on average.

6.2. Vehicle Classification

Our vehicle database contains 1000 vehicle images, of which some examples are shown in Figure 5. The resolution of each image is 128×64. As compared with people, vehicle is rigid, but has more variabilities of types and viewing aspects. We split the database into three groups, where the random exemplar candidate set has 200 samples, the training set and the test set both have 400 samples. The number of negative samples for training and test are both 800. Figure 6 shows 20 exemplars selected automatically by the algorithm. Note here the number of iterations T is actually 30, but 10 of the exemplars selected at different iterations are repeated twice. It can be seen that the exemplars nicely

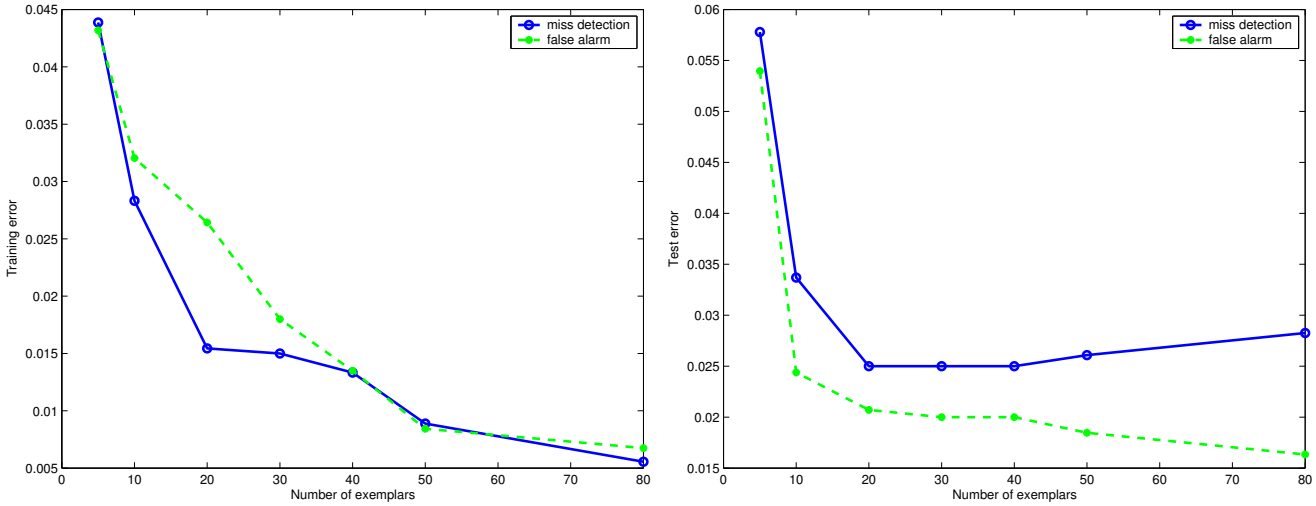

Figure 2. Training error rates (left plot) and test error rates (right plot) vs. the number of selected discriminative exemplars used in the SFEC algorithm. Note that the scales of the vertical axes are different.

Figure 4. A difficult test set with inaccurate target bounding boxes, large distortion, and heavy shadows.

cover different poses and viewing aspects. Table 2 compares the results of different algorithms. As in the people case, our algorithms consistently outperform the other two approaches.

Approach	SFEC	SADB	PBT	MFEC
MD	2.25	6.5	4.0	2.0
FA	4.25	4.25	5.75	3.5

Table 2. Test errors with different approaches. The SFEC results are based on 35 exemplars. The MFEC uses 30 exemplars, and each exemplar uses 25 features. The miss detection rate, denoted as MD, and the false alarm rate, denoted as FA, are all in percentages.

Figure 5. Examples from the database with 1000 vehicle images showing large variations of vehicle types and viewing aspects.

Figure 6. Discriminative exemplars automatically selected with the MFEC algorithm.

7. Conclusion and Future Work

We have developed a unified computational framework for detecting multi-view multi-pose objects without manual categorization for the training data. We have demon-

strated the efficacy of the proposed approach with both people and vehicle objects with large intra-class variations of pose, view aspects, scene clutter, and illumination conditions. Depending on features, distance measures, and different design philosophies, the proposed approach can be customized to work with focus of attention or run end-to-end by cascading the same type of detectors but in a coarse-to-fine fashion.

In our experiments, we observed that choosing the initial candidate exemplar set does have an impact on the performance. In the future, we will explore different approaches for a better selection of the candidate set. We will extend this approach to handle multiple classes and investigate the possibility of sharing features and exemplars by multiple classes. We will look into the feasibility of applying a hierarchical exemplar representation so that only a small portion of the exemplars needs to be compared during the run time. We will also study the impact of different kind of features and examine the potential of improving the classification performance on the feature level.

Acknowledgements

We would like to thank Professor Zhuowen Tu at UCLA for his support of Probabilistic Boosting Tree (PBT) related experiments and inspiring discussions on general object detection with AdaBoost. We would also like to thank Professor Ying Wu and Dr. Ting Yu at NWU for valuable discussions on their MRF-based pedestrian detection work.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR05)*, volume 2, pages 886–893, June 2005.

[2] Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Science*, 55, 1997.

[3] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *International Conference on Computer Vision (ICCV99)*, pages 87–93, 1999.

[4] C. Huang, H. AI, Y. Li, and S. Lao. Vector boosting for rotation invariant multi-view face detection. In *International Conference on Computer Vision (ICCV05)*, 2005.

[5] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 9(15):850–863, 1993.

[6] T. Kohonen. The self-organizing map. In *Proceedings of the IEEE, 78(9):1464-1480*, 1990.

[7] S. Li, L. Zhu, Z. Zhang, A. Blake, and H. Shum. Statistical learning of multi-view face detection. In *Proceedings of the 7th European Conference on Computer Vision (ECCV02)*, 2002.

[8] A. Mohan, C. Papageorgiou, T. Poggio, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 23(4):349–361, 2001.

[9] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Trans. Image Processing*, 6(1):103–113, 1997.

[10] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[11] Y. Shan, B. Matei, H. S. Sawhney, R. Kumar, D. Huber, and M. Hebert. Linear model hashing and batch ransac for rapid and accurate object recognition. In *International Conference on Computer Vision and Pattern Recognition (CVPR04)*, 2004.

[12] Y. Shan, H. S. Sawhney, and R. Kumar. Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR05)*, 2005.

[13] A. Shashua, Y. Gdalyahu, and G. Hayon. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV2004)*, 2004.

[14] X. Song and R. Nevatia. A model-based vehicle segmentation method for tracking. In *International Conference on Computer Vision (ICCV05)*, 2005.

[15] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *International Conference on Computer Vision (ICCV01)*, 2001.

[16] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *International Conference on Computer Vision (ICCV05)*, 2005.

[17] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR01)*, 2001.

[18] P. Viola and M. Jones. Fast multi-view face detection. In *Merl Technical Report TR2003-96*, 2003.

[19] Y. Wu, T. Yu, and G. Hua. A statistical field model for pedestrian detection. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR05)*, 2005.