Machine Vision and Applications
DOI 10.1007/s00138-011-0388-y

SPECIAL ISSUE PAPER

Active learning for on-road vehicle detection: a comparative study

Sayanan Sivaraman · Mohan M. Trivedi

Received: 23 October 2010 / Revised: 31 August 2011 / Accepted: 15 November 2011

Abstract In recent years, active learning has emerged as a powerful tool in building robust systems for object detection using computer vision. Indeed, active learning approaches to on-road vehicle detection have achieved impressive results. While active learning approaches for object detection have been explored and presented in the literature, few studies have been performed to comparatively assess costs and merits. In this study, we provide a cost-sensitive analysis of three popular active learning methods for on-road vehicle detection. The generality of active learning findings is demonstrated via learning experiments performed with detectors based on histogram of oriented gradient features and SVM classification (HOG–SVM), and Haar-like features and Adaboost classification (Haar–Adaboost). Experimental evaluation has been performed on static images and real-world on-road vehicle datasets. Learning approaches are assessed in terms of the time spent annotating, data required, recall, and precision.

Keywords Semi-supervised learning · Active learning · Annotation costs · Object detection · Active safety · Intelligent vehicles

1 Introduction

While navigation systems in consumer vehicles have become increasingly common, the need for a holistic approach to intelligent vehicles has become increasingly apparent. Navigation systems can locate the ego-vehicle and formulate driving directions, but do not comprise a comprehensive approach to driver assistance. Navigation systems cannot assess local traffic conditions, such as those affected by pedestrians, vehicles, or even road construction. In addition, the inclusion of navigation consoles and other associated infotainment adds a potential source distraction to the driver. Automotive accidents attributed to distracted drivers have increased significantly over the past decade [29]. The need for robust sensing as part of a holistic approach to driver assistance is as urgent as ever.

Between 1 and 3% of the world's gross domestic product is spent on the medical costs, property damage, and other costs associated with automotive accidents. Annually, some 1.2 million people die worldwide as a result of traffic accidents [46]. Research dealing with the development of sophisticated sensing systems for vehicle safety promises safer journeys by maintaining an awareness of the on-road environment for driver assistance. While sensing systems based on radar have made their way to consumer products, vision for on-road intelligent driver assistance systems has been a particularly active area of research in the intelligent vehicles community for the past decade [38,42].

The development of vision systems for intelligent vehicles presents a number of unique challenges. The performance of on-road vision systems is safety critical, and a given system must perform robustly, in a variety of backgrounds, weather, lighting, and traffic conditions. There are unique difficulties due to motion artifacts, and a high variability among vehicles typically encountered on roads and highways. Vision systems for intelligent vehicles also have a real-time execution requirement; a difference of a few hundred milliseconds can avoid or mitigate an accident.

On-road vision systems for driver assistance often are focused on perception of the on-road environment. Example applications may include pedestrian detection [1,23], lane

S. Sivaraman (✉) · M. M. Trivedi
Computer Vision and Robotics Research Lab, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0434, USA
e-mail: ssivaram@ucsd.edu

M. M. Trivedi
e-mail: mtrivedi@ucsd.edu

Published online: 16 December 2011

tracking [28,30], ego motion estimation [15], or vehicle detection [39]. On-road vehicle detection is a challenging problem that has been of great interest to the intelligent vehicles community. Recent studies have found success by incorporating active learning for on-road vehicle detection [36].

In the past decade, active learning has emerged as a powerful tool for building robust object detection systems in the computer vision community. Incorporating active learning approaches into computer vision systems promises training with fewer samples, more efficient use of data, less manual labeling of training examples, and better system recall and precision. In particular, various active learning approaches have demonstrated encouraging results for the detection of pedestrians [10,21], vehicles [36], faces [20], and other objects of interest.

Broadly speaking, active learning consists of initializing a classifier using conventional supervised learning and querying informative samples to retrain the classifier. The query function is central to the active learning process [26]. Generally, the query function serves to select difficult training examples, which are informative in updating a decision boundary [4,34]. Various approaches to the general query of samples in a binary problem have been explored [8].

While utilizing active learning for object detection promises training with fewer samples [8], less manual labeling of training examples [33], and better system performance [24,33]; there remain several open research issues. While it is generally accepted that active learning can reduce human annotation time in data labeling, few studies have performed the sort of annotation time experiments found in [35]. In addition to documenting the necessary labeling effort, few studies provide a side-by-side analysis of human time costs as a function of the sample query. While it is also accepted that active learning promises training with fewer samples, *how many* samples is not known. If a particular active learning approach outperforms its competitors with 1,000 training samples, does this necessarily mean that it will outperform them with 200, 5,000, or 10,000 training examples? In particular, when developing vision systems for vehicles, knowledge of the performance, labor, and data implications is very important.

In this study, we have implemented three widely used active learning frameworks for on-road vehicle detection. Two main sets of experiments have been performed. In the first set, HOG features and support vector machine [6,7,11,23] classification have been used. In the second set of experiments, the vehicle detector was trained using Haar-like features and Adaboost classification. Based on the initial classifiers, we have employed various separate querying methods. Based on confidence-based active learning [26], we have implemented Query by Confidence (QBC) with two variations. In the second, informative independent examples are queried from unlabeled data corpus, and a human oracle labels these examples. This method is compared against simply querying labeled examples from the initial corpus that are queried based on a confidence measure, as in [10,21]. The third learning framework we term Query by Misclassification (QBM), in which the initial detector is simply evaluated on independent data, and a human oracle labels false positives and missed detections. This approach has been used in [1,33].

The contributions of this paper include the following. A comparative study of active learning for on-road vehicle detection is presented. We have implemented three separate active learning approaches for vehicle detection, comparing the annotation costs, data costs, recall, and precision of the resulting classifiers. The implications of querying examples from labeled versus unlabeled data is explored. The learning approaches have been applied to the task of on-road vehicle detection, and a quantitative evaluation is provided on a real-world validation set.

The remainder of this paper is organized as follows. Section 2 provides a survey of related works in active learning and on-road vehicle detection. Section 3 provides background and theoretical justification for the active learning approaches evaluated in this research study. Section 3 provides implementation details for the object detection systems. Section 4.4 provides a quantitative experimental analysis of learning framework performance, which includes system performance, data necessity, and human annotation time involved in each. Section 5 provides concluding remarks.

2 Related research

We divide our literature review into subsections dealing with active learning and with vision-based on-road vehicle detection (Table 1).

2.1 Active learning for object detection

There is a rich literature in the machine learning community on active learning [4,8,9,26]. In certain separable learning scenarios, the decision boundary obtained by sequential active learning provably converges much faster than learning by random sampling from the distributions [4]. Active learning makes efficient use of training examples, which is especially useful when training examples are rare, or when extensive human annotation time in necessary to label images [10], which is generally known to be high cost [9]. Moreover, active learning provides a learning framework in which classifiers can be updated after the initial batch training and adapted to the environment to which they are deployed. This is of particular interest in vision-based object detection, as there may be significant differences between the training set and real-world conditions [25,33]. Active learning also

Table 1 Selected active learning based object detection approaches

Research study	Features	Learning	Sample query	Unlabeled samples?	Target object	Comments
Abramson and Freund [1]	Control points	Adaboost	Query by Misclassification	Yes	Pedestrians	Instrumented vehicle platform
Hewitt and Belongie [20]	Local features	one-shot learning	Manual initialization + tracking	Yes	Faces	Tracking used to generate samples for offline training
Leistner et al. [25]	Haar-like features	Adaboost + online boosting	Query by Misclassification	Yes	Pedestrians	Multi-view surveillance
Enzweiler and Gavrila [10]	Local features	SVM	Query by Confidence	No	Pedestrians	Additional samples created using generative model
Roth and Bischof [32]	Haar-like features	Online boosting	Automatic initialization + tracking	Yes	Faces, various objects	Online boosting for detection and for tracking
Roth et al. [33]	Haar-like features	Adaboost + online boosting	Query by Misclassification	Yes	Pedestrians	Annotation effort measured in mouse clicks
Sivaraman and Trivedi [36]	Haar-like features	Adaboost	Query by Misclassification	Yes	Vehicles	Instrumented vehicle platform
Lampert and Peters [24]	Pixel colors	Structured SVM	Tracking + misclassification	Yes	Ping pong ball	Structured output regression + active learning, implemented on GPU
Joshi and Porikli [21]	Histogram of oriented gradients	SVM	Query by Confidence	Yes	Pedestrians	Both human-supervise, and autonomous implementations, adaptive detection per scene
This study	Histogram of oriented gradients, Haar-like features	SVM, Adaboost	Query by Confidence, Query by Misclassification	No, Yes, Yes	Vehicles	Comparative study of active learning for on-road vehicle detection, instrumented vehicle platform

provides a solid framework for adaptive extensions, such as online learning for object detection [16,25,33].

Recent studies in the field address two overlapping questions. The first deals with formulating a query function for identifying informative examples [10,22,26]. The second deals with the labeling costs in active and online learning [22,33,35].

2.1.1 Sample query

Many active learning studies invoke the concept of the *Region of uncertainty* [4,9,10], a region in the decision space where classification is not unambiguously defined. The Region of Uncertainty is a function of the training data, whose size can be reduced by querying uncertain samples, identified by their probabilities or confidences [10,21,22], or the result of misclassification [4]. Defining a query protocol to identify informative examples is integral to active learning [10,26].

Confidence-based query uses a confidence metric to query examples that lie near the classification boundary in the decision space [26]. This can be done with a variety of metrics. In [26], such informative scores were identified by using a dynamic histogram of discriminant function values over the

entire training set. In [35], entropy was used to query the most uncertain examples. The euclidean distance from the decision boundary was used to query informative samples in [22].

The score-based query may simply be a threshold on the value of a classifier's discriminant function evaluated on given samples. An implicit probability or confidence measure for binary classifiers can also be obtained by feeding the value of the discriminant function to the logistic function as in [5,14,25]. In this case, the learning process queries samples x with class conditional probabilities near 0.5. This methodology has been used even when an explicitly probabilistic formulation has been used to model the data. In [10], a generative model of the target class was built. Random samples were generated from the generative model, and those samples lying near the decision boundary were queried using the logistic function for retraining. Training samples x were selectively sampled based on their class conditional probability, and these samples were used to retrain the SVM classifier [6,10].

Explicitly probabilistic approaches for active learning in vision are also explored in [10,22]. An online explicitly probabilistic formulation is used in [22], where sample query was implemented online using Gaussian processes for regression. In this study, query is performed using both the mean and variance of the expected class membership, which allows for integrating measures of uncertainty directly into selective sampling query [22].

However, in many object detection studies misclassification is utilized [1,16,25,33,36]. In this case, the human oracle labels false positives and missed detections, which are then archived for retraining. In [1,36], the learning process consists of an offline batch training, followed by a series of semi-supervised annotations, and a batch retraining. In [16,25,33], this is augmented by integrating online learning, so that each newly annotated frame immediately updates the classifier. In [20,24,32], tracking is used as a metric to identify regions of interest and gather more training samples. Table 2 summarizes the active learning approaches.

2.1.2 Labeling costs in active learning

Robust object detection often requires tens of thousands of training examples, which can require extensive annotation time [16,25,33]. As such, research studies address the cost of annotating unlabeled examples in terms of data required and human annotation time [22,33,35]. In [22], the number of necessary annotations is shown to be quite small with the use of online Gaussian process regression for object recognition.

In [35], the human cost of annotation is measured over multiple datasets, time frames, and individuals. It is found that annotation times are in general not constant, and that while a more precise query function may require fewer annotations, the time an individual takes may increase as the queried examples become more difficult [35]. In [44], the cost annotation costs per sample are predicted using an uncertainty model.

Identifying misclassified examples is a simple yet powerful approach to sample query than may increase annotation speed and lend itself easily to online learning for object detection [24,33]. While online learning for object detection has been shown to be quite effective [16], it has been shown that combining offline initial training with online updates [33] can reduce annotation time and improve detection performance in deployment scenarios.

2.2 On-road vehicle detection

While the most popular vehicle detection systems found in consumer products are the radars used for adaptive cruise control, it is known that commonly used commercial grade sensors may have limited angular range and temporal resolution [40]. Many of the radar-based systems for adaptive cruise control are meant only to detect vehicles directly in front of the ego vehicle. This may present problems during lane changes, or when the road has non-zero grade or curvature. In addition, such systems often do not provide information on vehicles in neighboring lanes. Using vision

Table 2 Definition of active learning approaches compared in this paper

Sample query	Description	Object detected
Query by Confidence, labeled examples	Examples with known class membership are queried based on a confidence or probability score	Pedestrians [10]
Query by Confidence, unlabeled examples	Unlabeled examples are queried based on a confidence or probability score, and labeled by human	Various objects [22]
Query by Misclassification	Unlabeled examples are queried based on raw classifier output, and labeled by human	Vehicles [36], Pedestrians [25]

Fig. 1 Examples of the varied environments where on-road vehicle detectors must perform

for vehicle detection can recognize and track vehicles across multiple lanes [37].

Robust recognition of other vehicles on the road using vision is a challenging problem and has been an active area of research over the past decade [40]. Highways and roads are dynamic environments, with ever-changing backgrounds and illuminations. The ego vehicle and the other vehicles on the road are generally in motion, so the sizes and locations of vehicles in the image plane are diverse. There is high variability in the shape, size, color, and appearance of vehicles found in typical driving scenarios [40].

Vehicle detection and tracking has been widely explored in the literature in recent years [37]. In [39], a variety of features were used for vehicle detection, including rectangular features and Gabor filter responses. The performance implications of classification with SVMs and NN classifiers was also explored. In [41], histogram of oriented gradient features were used for vehicle localization.

The set of Haar-like features, classified with Adaboost, has been widely used in the computer vision literature, originally introduced for detection of faces [45]. Various subsequent studies have applied this classification framework to vehicle detection [18,31]. In [17], the effect of varying the resolution of training examples for vehicle classifiers was explored, using rectangular features and Adaboost classification [13]. Rectangular features and Adaboost were also used in [36], integrated in an active learning framework for improved on-road performance.

In [19], vehicle detection was performed with a combination of triangular and rectangular features. In [18], a similar combination of rectangular and triangular features was used for vehicle detection and tracking, using Adaboost classification. In [2], a statistical model based on vertical and horizontal edge features was used for vehicle detection. In [12], vehicles are tracked at nighttime by locating the taillights (Fig. 1).

3 Active learning for on-road vehicle detection

Vision-based detection of other vehicles on the road is a difficult problem. Given that the end goal of on-road vehicle detection pertains to safety applications, such systems require robust performance from an automotive platform. Roads and highways are dynamic environments, with rapidly varying backgrounds and illuminations. The ego vehicle and the other vehicles on the road are in motion, so the sizes and locations of vehicles in the image plane are diverse. There is high intra-class variability found in vehicles, in the shape, size, color, and appearance of vehicles found in typical driving scenarios [39]. There are also motion artifacts, bumps, and vibrations from the road, and changes in pitch and yaw due to hills, curves, and other road structures. In addition, real-world on road scenes present many vehicle-like regions such as cast shadows, trees, and man-made structures, which tend to spur false positives. In this study, we have applied three active learning approaches to the task of on-road vehicle detection. We discuss the approaches below.

3.1 Query by Confidence

Query by Confidence (QBC) is based on the notion that the most uncertain and informative samples are those samples that lie closest to the decision boundary. These examples can be queried based on a confidence measure [26]. In [35], uncertainty was calculated using entropy over the label posteriors. While generative models used throughout the learning process can facilitate a confidence-based query based on probabilities as in [10,22], often visual object detectors are based on discriminative classification. In particular, support vector machines [6], and Adaboost [13] have been widely used [10,25,36].

Using common discriminative classifiers such as linear support vector machines and Adaboost, binary classification is based on a weighted sum of extracted features, which can be viewed as an inner product [13]. This is to say that a sample x's class y is given by the sign of a discriminant function $H(x)$ as given in Eqs. 1 and 2, where $h_n(x)$ are extracted features, and w_n are the weights.

$$y(x) = sgn\{H(x)\} \tag{1}$$

$$H(x) = \sum_{n=0}^{N-1} w_n h_n(x) \tag{2}$$

Using Eqs. 1 and 2 results in a hard classification. While discriminative classifiers are not explicitly probabilistic, a measure of confidence can be obtained by feeding $H(x)$ to a logistic function. The logistic function is monotonic and maps $H(x)$ to a value on the interval [0,1]. For Adaboost classification, the result of the logistic function can be interpreted as a class conditional probability $p(y = 1|x)$ [25], as in Eq. 3.

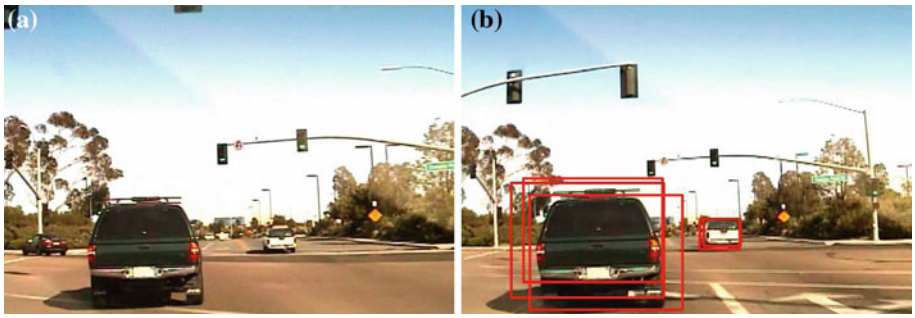


Fig. 2 **a** Training image from which the query function from Eq. 7 returned no informative training examples. **b** Training image from which multiple informative image subregions were returned

$$p(y = 1|x) = \frac{1}{1 + e^{-2H(x)}} \tag{3}$$

A rigorous proof of these equations can be found in [14]. For support vector machines using a linear kernel, with primal , the class conditional probability is computed via the following equation [3,27], where the parameters A and B are learned using maximum likelihood. Further discussion can be found in [27].

$$p(y = 1|x) = \frac{1}{1 + e^{-\{AH(x)+B\}}} \tag{4}$$

$$Q(x) = \{x : |p(y = 1|x) - 0.5| < \epsilon\} \tag{5}$$

With a confidence or probabilistic measure of a sample's class membership as defined by the classifier, we can define a query function $Q(x)$ that returns those samples that lie close to the decision boundary as in Eq. 5 (Fig. 2).

3.1.1 Query of unlabeled samples

Confidence-based query may also be used with independent samples by prompting a human oracle for annotations [22] of examples that satisfy Eq. 5. In this case, there are no prior hand labels for the data, and a human must decide whether the examples correspond to objects of interest. A human oracle's judgment of the appropriateness of the examples may add semantically meaningful examples to the data corpus that otherwise might be missed by automatic methods. This query method incorporates no prior knowledge of structure. In this case, it is assumed that while the initial corpus may not be adequate for training a classifier with good generality, that confidence regression properties of the classifier serve to query informative independent samples and minimize human annotation time. It is of note that fewer queried samples do not necessarily mean less human time spent on annotation, as humans may take longer to make decisions on annotating difficult samples [35].

By evaluating the query function $Q(x)$ on a corpus of already labeled training examples, we can retrieve those that lie closest to the decision boundary for retraining with no extra human effort. It is shown in Eqs. 1–5 that queried samples are those with almost equal class-conditional probabilities.

3.1.2 Query of labeled samples

While in [10] pre-cropped, hand-labeled training examples were used for confidence query, in this study the entire original image has been retained. For each location and scale in the search space of the image, we calculate the confidence value using the trained model. Using Eq. 5, we query those image subregions that lie close to the trained model's decision boundary. In the next step, we exploit structural information, using the location of the subregion queried with 5 and prior image annotations to compute the $Overlap$, as shown in Eq. 6. In this case, x' signifies a prior annotation, and x signifies the image subregion sample returned by 5.

Using Eq. 7, if $Q'(x)$ is non-zero, subregion x is retained for retraining. This approach provides a principled way for the learning process to query training examples that lie near the decision boundary and satisfy the structural constraint defined by hand labeling. It also allows us to obtain multiple informative training examples from a single hand-labeled example. This also means that queried examples are not strictly a subset of the initial training examples.

$$Overlap(x', x) = \frac{Area(x' \cap x)}{Area(x' \cup x)} \tag{6}$$

$$Q'(x) = Q(x) if\, Overlap(x', x) > \tau, else\ Q'(x) = 0 \tag{7}$$

Query of negative training examples is performed using Eqs. 1–4. We simply query negative image regions whose confidence lies close to the trained model's decision boundary, which are most informative for updating the classifier [10].

3.2 Query by Misclassification

There exist scenarios where although a classifier may have excellent performance over the training examples, the environments encountered in deployment differ substantially from the training examples, to the detriment of system recall and precision [25,33].

To ameliorate this problem,Query by Misclassification has been used in [1,16,25,33,36]. This method requires a human in the loop to label queried examples. The system typically

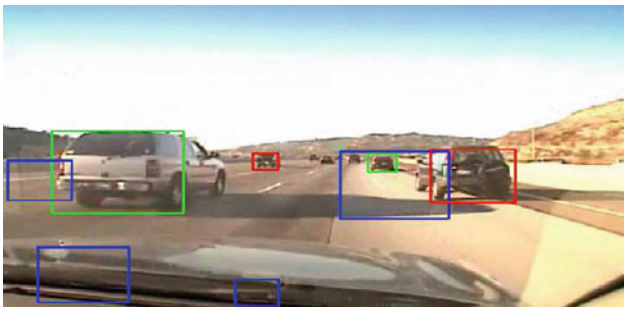


Fig. 3 Interface for Query by Misclassification. The interface evaluates the initial detector, providing an interface for a human to label ground truth. Detections are automatically marked. Missed detections and false positives are marked by the user

presents the user with the results of evaluating an initially trained classifier on independent datasets. Often, the independent data are more pertinent to actual deployment scenarios than the data that were used in initial training [25]. Users then mark these results as correct detections and false positives, and are able to also mark missed detections [32,36]. The annotation time used in the sample query is significantly faster than annotation associated with gathering initial training examples.

Figure 3 depicts the interface used in this study for Query by Misclassification. The interface evaluates the initial detector, providing an interface for a human to label ground truth. Detections are automatically marked green. Missed detections are marked red by the user, and false positives blue.

In this active learning paradigm, the most informative samples are those that result in misclassifications by the object detector [1,25,36]. However, correct detections are generally retained for retraining, to maintain a superset of the region of uncertainty and avoid overfitting [4]. Using this methodology, efficient on-line learning systems have been implemented in [16,25,33] in surveillance deployments for object detection.

4 Experimental evaluation

4.1 Learning considerations

Active learning is employed for object detection to train using less data [35], to minimize human annotation time [22], and to improve classifier performance [4,16,25,36]. We briefly examine the implications of these three factors in building robust active learning-based object detection systems.

4.1.1 Data considerations

The size and quality of available training data has major implications for active learning strategies. In certain applications, labeled data may be scarce, and learning methods may be aimed at using the scarce data as efficiently as possible, building models based on this data to query informative unlabeled samples [22]. However, if a large labeled data corpus is available, then learning methods can be aimed at sampling a subset of the extant corpus for retraining [10,43]. Of course, it is favorable to minimize the size of the training data, reducing computational load, while maintaining strong classifier performance.

Further data considerations include characteristics of the features, dimensionality of the features, and the resolution of training examples. These considerations present intertwined issues. A given feature set can be well suited for a given object detection task, but the features themselves may require a certain image resolution. For example, HOG features have been widely used for pedestrian detection [7,23], but require a higher image resolution than Haar features, which have been used widely for vehicles [17,36]. The required image resolution and feature set influences the dimensionality of the feature vector, which in turn influences the the required number of training examples for a given performance benchmark.

4.1.2 Supervision costs

Compiling a set of training examples requires much labor and time spent on collecting and annotating video sets [10]. It is favorable to minimize the effort spent on annotation while still maintaining high-quality data for training. While methods for querying the most difficult examples from unlabeled data corpuses have been proposed in [22,35], it is shown in [35] that querying the most difficult examples may increase the time a user spends on annotation.

4.2 Feature and classifiers sets

Two sets of experiments have been conducted to evaluate active learning approaches, using two different feature-classifier pairs for vehicle detection. The first set of experiments have used Histogram of Oriented Gradient features with linear support vector machine classification [3,6,7]. The second set of experiments have used Haar-like features and Adaboost cascade classification [13,45].

4.3 Training sets

We begin with a hand-labeled initial data corpus of 10,000 positive and 15,000 negative training examples, which was used to train the initial classifier. These examples were taken from video captured on highways and urban streets. For implementing Query by Confidence for labeled samples, we queried uncertain examples that satisfied Eq. 7. In querying independent unlabeled samples, Query by Misclassification and Query by Confidence, we queried examples using the methods described in the previous section, applied to an independent data corpus. A human oracle performed

Table 3 Comparison of labeling time for vehicle detectors trained with 1,000 samples, HOG–SVM

Method	Data samples	Total data	Labeling time (h)	Total time (h)	Indep. samples?
Random samples	1,000	1,000	27.8	27.8	No
Query by Misclassification	1,000	11,000	2.3	30.1	Yes
Unlabeled Query by Confidence	1,000	11,000	3.0	30.8	Yes

Table 4 HOG–SVM vehicle detection, training parameters

Parameter	Value
Number of orientations	4
Cell size	8×8
Training sample resolution	96×72
Kernel	Linear

Table 5 Active learning results: HOG–SVM vehicle detection, Caltech 1999 vehicle database

Sample query	Recall (%)	Precision (%)
Random examples	44.4	59.0
Query by Confidence	54.8	60
Query by Misclassification	77.8	81.4

the semi-supervised labeling. All labeling was timed. The independent data video corpus was acquired by concatenating nine high-density urban and highway traffic scenes, each lasting some 2 min.

4.4 Experiment 1: HOG–SVM vehicle detection

In this set of experiments, we train vehicle detectors using HOG features and linear SVM classifiers. The goal here was to evaluate what the potential performance would be with a very small number of training examples, in this case, 1,000. Vehicle detectors were evaluated on the Caltech 1999 vehicle database at http://www.vision.caltech.edu/archive.html, which consists of 127 still images of vehicles. Table 3 shows the labeling time for querying 1,000 samples. We note that Query by Confidence took somewhat longer than Query by Misclassification. This is due to the fact that ambiguous samples are often more difficult for users to decide labels [35]. Parameters used for the training are provided in Table 4, trained with LibSVM [3].

Table 5 shows the precision and recall of the classifiers, trained with 1,000 samples. We note that both active learning methods showed improved recall and precision over the vehicle detector trained with random examples. However, the results from Query by Misclassification were far better than those from Query by Confidence.

Table 6 Haar–Adaboost vehicle detection, training parameters

Parameter	Value
Number of cascade stages	15
Training error per stage	0.5
Expected training error	3e-5
Training sample resolution	24×18

Fig. 4 Example from *LISA_2009_Dense* validation set. The dataset consists of 1,600 consecutive frames, captured during rush hour, over a distance of roughly 2 km. The ground-truth dataset is publicly available to the academic and research communities at http://cvrr.ucsd.edu/LISA/index.html

4.5 Experiment 2: Haar features and Adaboost

In this set of experiments, we document the relative merits and trade-offs of active learning for vehicle detection using Haar-like features and Adaboost. Detectors are learned using training sets of 2,500, 5,000, and 10,000 training examples. We evaluate the vehicle detectors on a publicly available dataset, *LISA_2009_Dense*, which can be found at http://cvrr.ucsd.edu/LISA/index.html (Fig. 4). This is a difficult dataset consisting of 1,600 consecutive frames. Captured during rush hour, this scene contains complex shadows, dynamic driving maneuvers, and five lanes of traffic. There are 7,017 vehicles to be detected in this clip. The distance covered is roughly 2 km. Parameters used in training are provided in Table 6.

Table 7 Comparison of annotation time for vehicle detectors trained with 2,500 samples, Haar + Adaboost

Method	Data samples	Total data	Labeling time (h)	Total time (h)	Indep. samples?
Random samples	2,500	2,500	7 (projected)	7 (projected)	No
Labeled Query by Confidence	2,500	12,500	0	27.8	No
Query by Misclassification	2,500	12,500	2.5	30.3	Yes
Independent Query by Confidence	2,500	12,500	2.8	30.6	Yes

Table 8 Comparison of annotation time for vehicle detectors trained with 5,000 samples, Haar + Adaboost

Method	Data samples	Total data	Labeling time (h)	Total time (h)	Indep. samples?
Random samples	5,000	5,000	14 (projected)	14 (projected)	No
Labeled Query by Confidence	5,000	15,000	0	27.8	No
Query by Misclassification	5,000	15,000	4.0	31.8	Yes
Independent Query by Confidence	5,000	15,000	4.6	32.4	Yes

Table 9 Comparison of annotation time for vehicle detectors trained with 10,000 samples, Haar + Adaboost

Method	Data samples	Total data	Labeling time (h)	Total time (h)	Indep. samples?
Random samples	10,000	10,000	27.8	27.8	No
Labeled Query by Confidence	10,000	20,000	0	27.8	No
Query by Misclassification	10,000	20,000	7.0	34.8	Yes
Independent Query by Confidence	10,000	20,000	7.6	35.4	Yes

4.6 Analysis

4.6.1 Human labeling time

Tables 7, 8, and 9 detail the time costs associated with each learning method. While Query by Misclassification and Query by Confidence methods require extra labeling to assign class membership to independent examples, it is of note that even to query and label 10,000 independent training examples, it only took some seven additional hours of annotation. This is due to the fact that the respective query functions make labeling much more efficient.

For all of the active learning methods, we note that the bulk of human labor was spent labeling the initialization set. We also note that labeling independent examples using Query by Confidence consistently took longer than Query by Misclassification. This is a similar phenomenon as reported in [35]. Query by Confidence uses a more sophisticated query criterion to return the most difficult examples. A human annotator requires more time to annotate the most difficult examples. Querying the most difficult examples may result in a better trained classifier, but it does not reduce the time spend labeling.

4.6.2 Data implications and system performance

As we have trained each classifier with the same number of examples, using their respective active learning query functions, discussion of data implications and system performance are intertwined. Tables 7, 8, and 9 show the number of training examples used to train each classifier. Each classifier has used the same number of positive and negative training examples for each instantiation: 2,500, 5,000, or 10,000. However, active learning methods that use independent samples, Query by Confidence and Query by Misclassification, are based on the initial classifier. As such, we add the size of the initial training corpus to their data requirements.

We have plotted recall versus 1-precision for each classifier and each dataset. The performance of classifiers trained with 2,500 examples is plotted in Fig. 5a. The performance of classifiers trained with 5,000 examples is plotted in Fig. 5b. The performance of classifiers trained with 10,000 examples is plotted in Fig. 5c.

4.7 Analysis

We observe that in general, as the number of training examples increases, so does the performance of each classifier. Figure 5a–c are all plotted on the same axes scale. The overall improvement in system performance with increased training data is shown there.

In general, we find that each of the active learning methods outperforms training with random examples, for both HOG–SVM, and Haar–Adaboost detection algorithms. This confirms the valuable contribution of active learning to vehicle

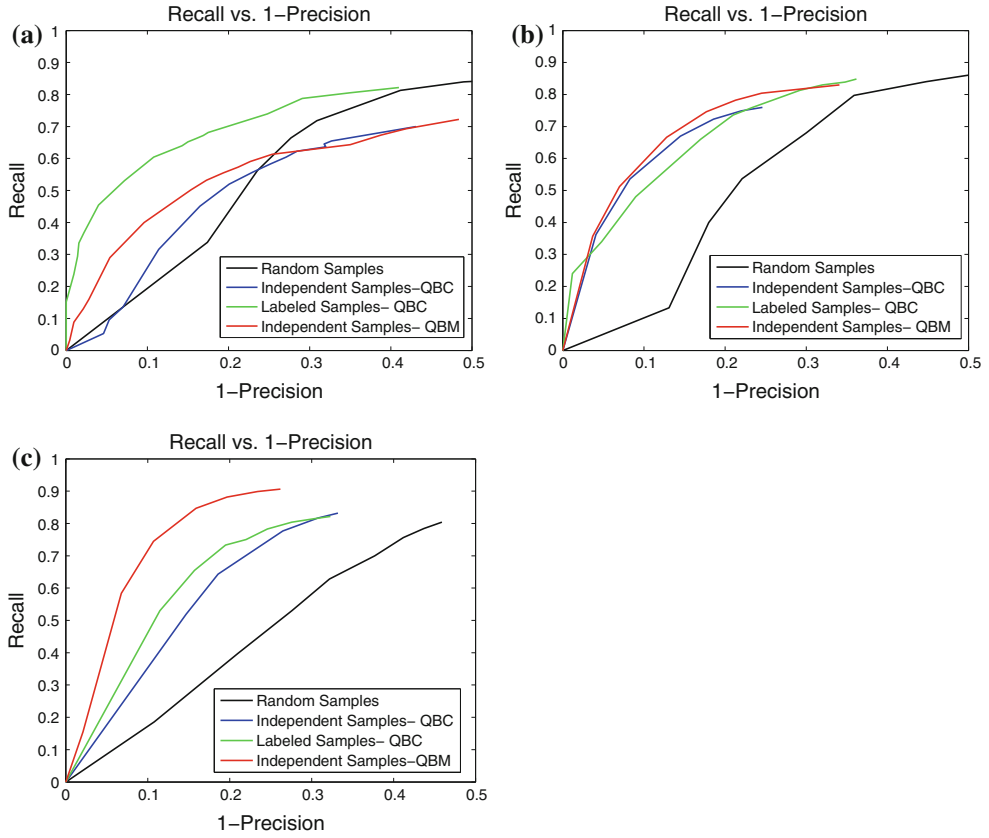


Fig. 5 Recall versus 1-precision for each vehicle detector, evaluated on the *LISA_2009_Dense* dataset. **a** Classifiers trained with 2,500 samples. **b** Classifiers trained with 5,000 samples. **c** Classifiers trained with 10,000 samples

Fig. 6 Frame 1136 of *LISA_2009_Dense*. Detectors trained with 2,500 samples. **a** Random samples. **b** Query by Misclassification. **c** Query by Confidence, independent samples. **d** Query by Confidence, labeled samples

detection. In the HOG–SVM experiment 1, QBM performed the best, followed by QBC, and finally training with random examples from the labeled corpus. We observe a similar performance trend in the Haar–Adaboost experiments (Fig. 6).

In experiment 2, as the number of training examples increases, the rankings of the three active learning methods changes. Using only 2,500 training examples, the best classifier is that built with labeled examples, using Query by Confidence as 2,500 samples is not enough to build a rich, representative data corpus for retraining at such a low resolution. Query by Confidence queries the most informative labeled examples from the initial training corpus and performs the best. In fact, training with random examples results in better performance than using independent samples for active learning, for such a small number of training examples, as shown in Fig. 5a.

In the next training set, we see changes in the performance rankings. Each of the active learning methods using 5,000 training examples outperforms training with random samples, as shown in Fig. 5b. Using 5,000 training samples, each of the active learning methods perform comparably well. Query by Misclassification yields the best performing classifier, but Query by Confidence of labeled and independent examples perform almost as well.

Throughout the experiments, we note that classifiers using Query by Confidence for labeled or independent samples perform similarly. This is to say that the inclusion of independent training samples does not make a large difference in the clas-

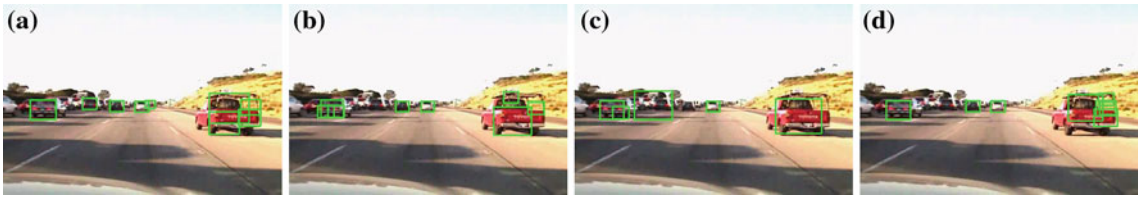


Fig. 7 Frame 1136 of *LISA_2009_Dense*. Detectors trained with 5,000 samples. **a** Random samples. **b** Query by Misclassification. **c** Query by Confidence, independent samples. **d** Query by Confidence, labeled samples

Fig. 8 Frame 1136 of *LISA_2009_Dense*. Detectors trained with 10,000 samples. **a** Random samples. **b** Query by Misclassification. **c** Query by Confidence, independent samples. **d** Query by Confidence, labeled samples

sifier's performance when using Query by Confidence. The reason for this lies in the similar query criterion. While one methods queries automatically from a labeled corpus, and the other queries from unlabeled independent samples, they both use the same query function, defined in Eqs. 1–4. As such, it makes sense that the classifiers that these methods yield perform comparably (Fig. 7).

We find that Query by Misclassification performs the best of the learning approaches for vehicle detection, using HOG–SVM and Haar–Adaboost. Each of the active learning methods far outperforms the initial classifier, training with random examples. Query by Misclassification has been used widely in the literature [1,24,25,32,33,36], and so it is expected that this method would perform well. This strong performance comes at the price of seven extra hours of annotation, and an extra independent data requirement. Of the active learning methods that use independent data that we have examined in this study, Query by Misclassification is the best choice for system performance and human labor (Fig. 8).

5 Concluding remarks

In this study, we have compared the labeling costs and performance pay-offs of three separate active learning approaches for on-road vehicle detection. Initial vehicle detectors were trained using on-road data. Using the initial classifiers, informative examples were queried using three approaches: Query by Confidence from the initial labeled data corpus, Query by Confidence of independent samples, and Query by Misclassification of independent samples. The human labeling costs have been documented. The recall and precision of the detectors have been evaluated on static images, and challenging real-world on-road datasets. The generality of the findings have been demonstrated by using detectors comprising HOG–SVM and Haar–Adaboost. We have examined the time, data, and performance implications of each active learning method. The performance of the detectors has been evaluated on publicly available vehicle datasets, as part of long-term research studies in intelligent driver assistance.

Acknowledgments The authors would like to thank colleagues Dr. B. Morris, Mr. C. Tran, Mr. S. Shivappa, Mr. A. Tawari, and Dr. A. Doshi for the useful comments and critiques. We would like to thank the UC Discovery Grant and VW Group of America for their sponsorship. Finally, we thank the reviewers and guest editors for their careful reading and constructive comments.

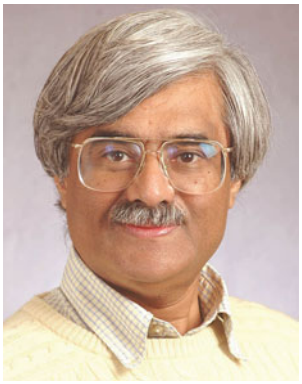
References

1. Abramson, Y., Freund, Y.: Active learning for visual object detection. UCSD Technical Report (2006). http://www.cse.ucsd.edu/~yfreund/papers/sevillePaper.pdf
2. Chan, Y., Huang, S., Fu, L., Hsiao, P.: Vehicle detection under various lighting conditions by incorporating particle filter. IEEE Intell. Transp. Syst. Conf. (2007)
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3) (2011) (Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm)
4. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. Mach. Learn. **15**(2), 201–221 (1994)
5. Collins, M., Schapire, R.E., Singer, Y.: Logistic regression, Adaboost and Bregman distances. Mach. Learn. (2002)
6. Cortes, C., Vapnik, V.: Support vector networks. Mach. Learn. **20**, 273–297 (1995)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. IEEE Conf. Comp. Vis. Patt. Recog. (2005)
8. Dasgupta, S.: Analysis of a greedy active learning strategy. Neural Inf. Process. Syst. (2004)
9. Dasgupta, S., Hsu, D.J., Monteleoni, C.: A general agnostic active learning algorithm. Neural Inf. Process. Syst. (2007)
10. Enzweiler, M., Gavrila, D.M.: A mixed generative-discriminative framework for pedestrian classification. IEEE Comput. Vis. Pattern Recognit. Conf. (2008)

11. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation and augmented reality tracking: an integrated system and evaluation for monitoring driver awareness. IEEE Trans. Intell. Transp. Syst. (2010)
12. Fossati, A., Schnmann, P., Fua, P.: Real-time vehicle tracking for driving assistance. Mach. Vis. Appl. (2010)
13. Freund, Y., Schapire, R.E.: A short introduction to boosting. J. Jpn. Soc. Artif. Intell. **14**(5), 771–780 (1999)
14. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression, a statistical view of boosting. Ann. Stat. (2000)
15. Gandhi, T., Trivedi, M.M.: Parametric ego-motion estimation for vehicle surround analysis using an omnidirectional camera. Mach. Vis. Appl. (2005)
16. Grabner, H., Bischof, H.: On-line boosting and vision. IEEE Comput. Vis. Pattern Recognit. Conf. **1**, 260–267 (2006)
17. Haselhoff, A., Schauland, S., Kummert, A.: A signal theoretic approach to measure the influence of image resolution for appearance-based vehicle detection. IEEE Intell. Veh. Symp. (2008)
18. Haselhoff, A., Kummert, A.: An evolutionary optimized vehicle tracker in collaboration with a detection system. IEEE Intell. Transp. Syst. Conf. (2009)
19. Haselhoff, A., Kummert, A.: A vehicle detection system based on Haar and triangle features. IEEE Intell. Veh. Symp. (2009)
20. Hewitt, R., Belongie, S.: Active learning in face recognition, using tracking to build a face model. IEEE Conf. Comp. Vis. Pattern. Recognit. Workshop (2006)
21. Joshi, A.J., Porikli, F.: Scene adaptive human detection with incremental active learning. IAPR Int'l Conf. Pattern. Recognit. (2010)
22. Kapoor, A., Grauman, K., Urtasun, R., Darrell, T.: Gaussian processes for object categorization. Int. J. Comput. Vis. (2009)
23. Krotosky, S.J., Trivedi, M.M.: On color-, infrared-, and multimodal-stereo approaches to pedestrian detection. IEEE Trans. Intell. Transp. Syst. (2007)
24. Lampert, C.H., Peters, J.: Active structured learning for high-speed object detection. DAGM Symposium on Pattern Recognition (2009)
25. Leistner, C., Roth, P., Grabner, H., Bischof, H., Stratzer, A., Rinner, B.: Visual on-line learning in distributed camera networks. IEEE Int Conf. Distrib. Smart Cameras (2008)
26. Li, M., Sethi, I.K.: Confidence-based active learning. IEEE Trans. Pattern. Anal. Mach. Intell. **28**(8), 1251–1261 (2006)
27. Lin, H.-T., Lin, C.-J., Weng, R.C.: A note on Platts probabilistic outputs for support vector machines. Mach. Learn. **68**, 267–276 (2007)
28. McCall, J., Trivedi, M.M.: Video based lane estimation and tracking for driver assistance: survey, system, and evaluation. IEEE Trans. Intell. Transp. Syst. vol. 7, no. 1 (2006)
29. National Highway Traffic Safety Administration. Distracted Driving 2009 (2010). http://www-nrd.nhtsa.dot.gov/Pubs/811379.pdf
30. Nieto, M., Laborda, J.A., Salgado, L.: Road environment modeling using robust perspective analysis and recursive Bayesian segmentation. Mach. Vis. Appl. (2010)
31. Ponsa, D., Lopez, A., Lumbreras, F., Serrat, J., Graf, T.: 3D vehicle sensor based on monocular vision. IEEE Intell. Transp. Syst. Conf., pp. 1096–1101 (2005)
32. Roth, P.M., Bischof, H.: Active sampling via tracking. IEEE Comput. Vis. Pattern Recognit. Conf. (2008)
33. Roth, P., Grabner, H., Leistner, C., Winter, M., Bischof, H.: Interactive learning a person detector: fewer clicks—less frustration. Workshop of the Austrian Association for Pattern Recognition (AAPR) (2008)
34. Roth, P., Sternig, S., Grabner, H., Bischof, H.: Classifier grids for robust adaptive object detection. IEEE Conf. Comput. Vis. Pattern. Recognit. (2009)
35. Settles, B., Craven, M., Friedland, L.: Active learning with real annotation costs. Nips Workshop on Cost-Sensitive Learning (2008)
36. Sivaraman, S., Trivedi, M.M.: A general active learning framework for on-road vehicle recognition and tracking. IEEE Trans. Intell Transp. Syst. (2010)
37. Sivaraman, S., Trivedi, M.M.: Improved vision-based lane tracker performance using vehicle localization. IEEE Intell. Veh. Symp. (2010)
38. Sivaraman, S., Trivedi, M.M.: Combining monocular and stereo-vision for real-time vehicle ranging and tracking on multilane highways. IEEE Intell. Transp. Syst. Conf. (2011)
39. Sun, Z., Bebis, G., Miller, R.: Monocular Precrash Vehicle Detection: Features and Classifiers. IEEE Trans. Image Proc. **15**(7), 2019–2034 (2006)
40. Sun, Z., Bebis, G., Miller, R.: On-road vehicle detection: a review. IEEE Trans. Pattern Anal. Mach. Intell. (2006)
41. Takeuchi, A., Mita, S., McAllester, D.: On-road vehicle tracking using deformable object model and particle filter with integrated likelihoods. IEEE Intell. Veh. Symp. (2010)
42. Trivedi, M.M., Gandhi, T., McCall, J.: Looking-in and looking-out of a vehicle: computer-vision-based enhanced vehicle safety. IEEE Trans. Intell. Transp. Syst. (2007)
43. Tsuruoka, Y., Tsujii, J., Ananiadou, S.: Accelerating the annotation of sparse named entities by dynamic sentence selection. BioNLP ACL workshop: themes in biomedical language processing (2008)
44. Vijayanarasimhan, S., Grauman, K.: Multi-level active prediction of useful image annotations for recognition. Adv. Neural Inf. Process. Syst. (2008)
45. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. IEEE Conf. Comput. Vis. Pattern. Recognit., vol. 1, pp. 511–518 (2001)
46. World report on road traffic injury prevention, World Health Organization. http://www.who.int/violence_injury_prevention/publications/road_traffic/world_report/factsheets/en/index.html

Author Biographies

Sayanan Sivaraman received his B.S. degree in electrical engineering from the University of Maryland in 2007, and his M.S. in electrical engineering from the University of California, San Diego in 2009. He is currently working towards his Ph.D. degree with specialization in intelligent systems, robotics, and controls. His research interests include computer vision, machine learning, intelligent vehicles, and transportation systems.



Mohan M. Trivedi received his B.E. (with honors) degree from Birla Institute of Technology and Science, Pilani, India, and his Ph.D. degree from Utah State University, Logan. He is Professor of electrical and computer engineering and the Founding Director of the Computer Vision and Robotics Research Laboratory, University of California, San Diego (UCSD). He and his team are currently pursuing research in machine and human perception, machine learning, distributed video systems, multimodal affect and gesture analysis, human-centered interfaces and intelligent driver assistance systems. He regularly serves as a consultant to industry and government agencies in the U.S. and abroad. He has given over 65 keynote/plenary talks. Prof. Trivedi served as the Editor-in-Chief of the Machine Vision and Applications journal. He is currently an editor for the IEEE Transactions on Intelligent Transportation Systems and Image and Vision Computing. He served as the General Chair for IEEE Intelligent Vehicles Symposium IV 2010. Trivedis team also designed and deployed the 'Eagle Eyes' system on the US-Mexico border in 2006. He served as a charter member, and Vice Chair of the Executive Committee of the University of California System wide UC Discovery Program. Trivedi served as an Expert Panelist for the Strategic Highway Research Program of the Transportation Research Board of the National Academies. He has received the Distinguished Alumnus Award from Utah State University, Pioneer Award and Meritorious Service Award from the IEEE Computer Society, and several Best Paper Awards. Trivedi is a Fellow of the IEEE, IAPR and the SPIE.