Learning to Use Scene Context for Object Classification in Surveillance

Biswajit Bose and Eric Grimson

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Abstract

Object classification in far-field outdoor video surveillance is a challenging problem because of low resolution, presence of shadows and projective image distortion. We present a classification system that learns to use scene context variables (such as position and orientation of objects), in addition to object variables (such as shape and size), to improve its performance in an arbitrary scene. The key feature of our system is that it needs only a small number of labeled examples from a few scenes, but extends well to novel scenes by extracting contextual information from unlabeled data. A three-step bootstrapping algorithm for adapting classifiers to a novel scene is proposed. A baseline classifier is first trained using object-specific features. This classifier is applied to a novel scene to calculate 'labels' for unlabeled data. Some of these labels are then incorporated into the training set of a scene-dependent classifier which uses both object and context features. Experimental results demonstrate the effectiveness of our adaptive classifiers for multiple urban scenes.

1. Introduction

Object classification in far-field visual surveillance is a challenging problem. Objects need to be detected against different backgrounds and changing lighting conditions. Even after successful detection, there are often very few image pixels per object, so not many local intensity-based features can be reliably extracted. This is quite different from object detection/recognition problems in near-field settings (such as face recognition), where there are a sufficient number of pixels per object to be able to detect its distinguishing components (for instance, body parts for person detection, or parts of a car for vehicle detection). Extracted features in far-field surveillance are also greatly distorted by the projective transformation induced by the camera—objects close by appear to be larger in size and moving faster than objects far away. Finally, since most object classification techniques for surveillance focus on learning from examples [5, 7, 8], the need for a set of hand-labeled training examples that are representative of the distribution of objects in the specific scene is a major limitation.

Our work is motivated by two observations:

- Features that help discriminate between objects of interest such as vehicles and pedestrians are of two broad types: scene-independent object features (*e.g.* relative size and shape) and scene-dependent context features (*e.g.* image-position, orientation and direction of motion of the object)

- Classification performance in far-field surveillance using both the above types of features is significantly better than classification with only object features.

Context features provide useful information in most real scenes because of underlying regularities in the world. For instance, position is an informative context feature because vehicles mostly appear on roads while pedestrians tend to stay on footpaths. In addition, the projected sizes of objects vary with position, in a manner characteristic of a particular camera position and orientation. Thus pedestrians close to the camera may appear larger than cars away from the camera. Vehicles and pedestrians also tend to have characteristic directions of motion in a scene. Although calculating context features directly from a single image is difficult, context can be learned over time by observing a large number of objects pass through a scene, provided object class labels are available.

In this paper, we address the task of building an object classifier (for distinguishing between vehicles and pedestrians) that only needs to be trained once on a small set of labeled examples, but can adapt automatically to an arbitrary scene by learning to use scene context features. We call this the problem of scene transfer. With most current systems, classification performance will go down each time the scene changes (for instance, by changing the orientation or zoom of the camera, or because of changes in object distribution over time), and labeled data from the old scene may become useless. Similarly, in a large surveillance network involving hundreds of cameras deployed throughout a city, obtaining labeled training sets for every scene is clearly not a viable option.

There has recently been significant interest within the machine learning community in making use of unlabeled examples for improving performance on learning tasks such

as text categorization [1, 6, 11]. Large quantities of unlabeled examples are often available in vision systems at no extra cost (round-the-clock surveillance data, for instance). While the use of unlabeled data has been considered within the vision community on a few occasions (see Wu *et al.* [13], for instance), the questions of how to use unlabeled data to adapt an object classifier to scene characteristics over time, or to transfer classifiers across scenes, have not been addressed previously. The method presented here makes use of unlabeled data in a novel scene to learn the context features associated with object classes in an unsupervised manner.

To achieve scene transfer, we first separate object features and context features by calculating mutual information scores between groups of features and class labels for a pool of object data collected from multiple scenes. We then apply a bootstrapping algorithm comprising three steps:

1. training a baseline classifier on objects from a few representative scenes using only scene-independent object features,

2. applying this classifier to a novel scene to obtain 'labels' for unlabeled examples, and

3. incorporating some of the new, high-confidence 'labels' into the training set of a more accurate, scene-specific classifier, that uses both object and context features.

We demonstrate the effectiveness of our algorithm on a large data set for different types of scene changes, and analyze its behavior in detail.

The organization of this paper is as follows. Section 2 describes our basic object classifier for video data. Section 3 explains how we identify the scene-dependent and scene-independent features. Section 4 presents our bootstrapping algorithm for scene transfer. Experimental results are discussed in Section 5. The final section summarizes our contributions and highlights directions for future work.

2. Object Classification from Video

We use a real-time object tracker based on Stauffer and Grimson's adaptive background subtraction algorithm [12] for detecting objects of interest in surveillance video. The input to our object classification system consists of pixel intensity values for tracked motion-blobs (regions where motion was detected) and the position and velocity of each motion-blob centroid in every frame. Since the identity of each tracked object is preserved across frames of the video, labeling a single tracked object essentially gives us multiple labeled instances of that object (one instance for each video-frame). We make use of this fact, both in designing our classifier and in extracting context features from unlabeled data in a novel scene.

It is interesting to consider the trade-offs involved in far-field surveillance. A greater amount of temporal information about an object is available from far-field data than from mid- or near-field data, because objects stay within the field of view for a longer period of time. On the other hand, image features available from near-field data are more reliable because of the higher resolution and the smaller range of projective distortion while an object is visible. The large range of projective distortion is a significant problem in far-field systems where the horizon line often lies within the field of view, since object size diminishes to zero as it approaches that line. Thus, the success of a far-field object classification system depends in part on the extent to which it leverages the information obtained by tracking an object through time. We now discuss the types of features we extract from tracking data.

2.1. Extracting Useful Video Features

Video sequences provide us with two possible kinds of features, which we call instance features and temporal features. Instance features are those features that can be associated with every instance of an object (that is, with each frame of a tracking sequence). The size of an object's silhouette and the position of a motion-blob centroid are examples of instance features. Temporal features, on the other hand, are features that are associated with an entire track, and cannot be obtained from a single frame. For example, the mean aspect ratio of an object's image or the Fourier coefficients of image size variation with time—both of which calculated using all instances of a track—are temporal features. Temporal features provide dynamical information about the object, and can generally only be used after having observed an entire track. However, temporal features can be converted into instance features by calculating them over a small window of frames in the neighborhood of a given frame. For example, apparent velocity of the object (in its image) is a feature that is calculated in this manner.

The features we considered for our object classification task are image position, size of bounding-box, variation in size, speed, aspect ratio, direction of motion, orientation and percentage occupancy of bounding-box. The 2D bounding box is the minimum rectangle that encloses the object's motion-blob and has two of its sides aligned with the principal axis of the blob. The size of the bounding box is the area enclosed by it. Orientation is given by the angle made by the principal axis of the object's image with the horizontal axis of the image. Percentage occupancy of bounding box is simply the number of object pixels within the bounding box divided by the size of the bounding box. Variation in size is a temporal feature that treats the image size of the object as a one-dimensional signal (which is a function of time) and finds the number of 'edges' in this signal by applying the Marr-Hildreth edge detector [9]. This is a potentially useful feature because vehicles are expected to have smooth variations in image size, whereas the pro-

jected size of pedestrians changes frequently as they walk. Using the technique discussed above, we convert this temporal feature into an instance feature before it is processed by our classification system.

The reasons for choosing most of these features are intuitively obvious. For instance, object position is useful given a particular urban scene, since vehicles mostly stay on the road regions of the scene and pedestrians stay on footpaths (except while crossing a street). Vehicles generally move faster than persons, but occasionally slow down or even stop. Also, vehicles far away may appear to move slower than people close by, even if they are actually moving faster, because of projective distortion. The aspect ratio of pedestrians (height-to-width ratio) is typically slightly greater than the aspect ratio of vehicles (length-to-width ratio), but depends on the camera orientation. Orientation and direction of motion of objects are both useful, since vehicles move in the direction of their principal axis, and rarely move across the normal flow of traffic in a scene. Percentage occupancy can help distinguish between persons close by and vehicles far away, both of which may have bounding boxes of the same size in the projected image. Vehicles, being almost rectangular, tend to occupy most of their bounding-boxes, while pedestrians fill their bounding-boxes sparsely. It is important to note that most of these features are subject to the effects of shadows, inaccurate detection of object regions while tracking, and projective distortion of the scene. Of course, the use of other features is possible.

As will be discussed in Section 3, these features are separated into two groups—scene dependent and scene independent—for the purpose of training our classifiers. The relative usefulness of these features, in terms of the information they provide about the object class, is also evaluated.

2.2. Choice of Classifier

There are two options available for classifying tracked objects: classifying individual instances separately (using instance features) and then combining the instance labels to produce an object label, or classifying entire tracks using temporal features. We chose an instance classifier, for two reasons. First, labeling of a single object produces many labeled instances and hence we can obtain a more reliable classifier from a small labeled set of objects. Second, a single instance feature (*e.g.* position of an object in a frame) often provides more information than the corresponding temporal feature (*e.g.* mean position of an object).

We considered using a generative model, but decided against it to avoid having to approximate multi-dimensional densities with only a small amount of labeled data. Instead, we chose a discriminative model—support vector machine (SVM) with soft margin—as our instance classifier. The use of a soft margin is important since the data are non-separable in the 9-dimensional feature space discussed above. One disadvantage of using SVMs is that the output is simply the signed distance of the instance from the separating hyperplane, and not the posterior probability of the instance belonging to an object class. Posterior probabilities are needed to be able to correctly combine instance labels to obtain an object label. To get around this problem, we retrofit a logistic function $g(x)$ that maps the SVM outputs x_i into probabilities (or confidence values) [10]:

$$g(x_i) = \frac{1}{1 + exp(-x_i)}. \quad (1)$$

The posterior probability $P(y_i = 1 \mid x_i, \lambda)$ is then given by $g(\lambda x_i)$, where the parameter λ is chosen such as to maximize the associated log-likelihood

$$l(\lambda) = \sum_{i=1}^{n} log\, P(y_i \mid x_i, \lambda) \quad (2)$$

where n is the total number of instances on which the SVM is trained and y_i is the label assigned to instance i.

We take the mean of all the instance confidence values in the tracking sequence, and assign the label corresponding to the class with highest confidence. Of course, to avoid the high dimensional complexity of modeling the dependencies between instances, we are making the implicit, simplifying assumption that feature values in different instances are independent of each other. As we shall see, we still achieve reasonably good test performance. To clarify, test error in our case corresponds to the fractional number of incorrect *object* labels, not instance labels.

In practice, the solution of the instance-based SVM is unstable. This is to be expected, as there are many instances of vehicles and persons which look exactly the same, especially for objects far away, and because of the effects of shadows. Perturbing the parameters of the SVM slightly will cause a significant change in the fraction of correctly classified examples. However, the confidences associated with these instance labels (after post-fitting a logistic function to the SVM outputs) will be low irrespective of the label assigned to the instance. Thus, as long as labels are ambiguous in only a small fraction of instances within a track for a given object, the overall classification performance for objects will not be significantly affected. We make use of Gaussian kernels and select the kernel bandwidth by 5-fold cross-validation. We also set a high upper bound (= 1000) on the value of the Lagrange multipliers that appear in the solution for the soft-margin SVM classifier (see [2] for details).

3. Feature Selection

Choosing an appropriate feature space for representing data is an important step for any learning-based algorithm. By adjusting the feature space representation, the generalization error of a classifier can be significantly decreased. The goal is to seek object features that differ very little within a class, but differ significantly between classes. In our case,

Feature	M.I. in a single scene	M.I. across multiple scenes
x-coordinate	0.44	0.05
y-coordinate	0.31	0.11
bounding-box size	0.61	0.53
variation in size	0.56	0.40
speed	0.42	0.37
direction of motion	0.08	0.03
orientation	0.24	0.10
aspect ratio	0.33	0.14
percentage occupancy	0.20	0.08

Table 1: Mutual information (M.I.) between object features and labels

Feature 1	Feature 2	M.I.
x-coordinate	*y*-coordinate	0.81
y-coordinate	percentage occupancy	0.52
y-coordinate	speed	0.50
x-coordinate	percentage occupancy	0.50
orientation	direction of motion	0.47
y-coordinate	aspect ratio	0.45
x-coordinate	speed	0.37
x-coordinate	aspect ratio	0.37
percentage occupancy	bounding-box size	0.33
orientation	bounding-box size	0.31

Table 2: Ten highest mutual information (M.I.) values between pairs of object features and labels in a single scene

since there is a large amount of variation in the features we have considered, both within a scene and across scenes, it is not immediately obvious which features should be used. On the one hand, we would like the chosen features to be invariant across scenes, given a particular class of objects. On the other hand, as we shall show shortly, incorporating knowledge about scene constraints into the feature space helps improve classification performance.

We perform feature selection by calculating the mutual information $I(\mathbf{X}; Y)$ between features and labels. To this end, we estimate the marginal and conditional probability distributions, $p(\mathbf{x})$ and $p(\mathbf{x}|\, y)$, of instance features $\mathbf{x}$ and labels y non-parametrically, using Parzen-window density estimators [4]. Two sets of mutual information calculations are then performed: the first using examples from a single scene, and the second using examples collected from a group of three scenes. The mutual information of the j^{th} feature with the label, after discretizing the continuous probability densities, is given in either case by [3]

$$I(X_j; Y) = \sum_{x_j \in \mathcal{X}} \sum_{y \in \{-1,1\}} p(x_j, y)\, log\, \frac{p(x_j, y)}{p(x_j)\, p(y)}. \quad (3)$$

Mutual information scores for our chosen features in the two cases are shown in Table 1. Upon analyzing these scores, we find that our features can be grouped into two categories. Features that have high mutual information with labels within a scene, but low values across a group of scenes, are scene-dependent context features. These clearly include the *x*- and *y*- image coordinates of the object. Apparent aspect ratio, too, depends to a great extent on viewpoint. Features that have reasonably high values of mutual information both within a scene and across scenes are scene-independent object features. Speed, bounding-box size and the variation in bounding-box size are good examples of such features. A third possible category consists of those features which have low mutual information in both cases, and hence should not be considered (as they are likely lead to over-fitting while training a classifier on a small number labeled examples). For instance, direction of motion by itself does not seem to provide very useful information.

Unfortunately, simply calculating mutual information for individual features with labels and choosing the features with the highest scores is not guaranteed to give the most informative features for a classification task. In order for feature selection to be theoretically optimal, mutual information scores need to be calculated not only with single features, but also with sets of features. For instance, if a road appears diagonally across the scene and people tend to walk on sidewalks beside the road, the mutual information of *x*- and *y*- image coordinates with labels might not be high, but the two taken together may almost uniquely identify the object as vehicle or person. In principle, this argument holds for all possible sets of features. However, in practice, it is highly unlikely that three or more features calculated from real-world scenes will conspire to give significantly better classification results than pairs of features acting together. Therefore, we repeat our mutual information calculations for all 36 possible pairs of our 9 features. The results for a single scene (the most relevant of which are given in Table 2) are similar to those in the single-feature case, except that (i) direction of motion and orientation, taken together, and (ii) bounding-box size and percentage occupancy, considered together, have significantly higher mutual information with the object label when considered jointly instead of singly. These observations are explained by the facts that (i) vehicles tend to be oriented in the direction of their motion, and (ii) percentage occupancy might be low for vehicles if some outlying pixels (that were included in the object's silhouette due to noise) made the bounding-box size excessively large. The results (not shown here) for pairs of features across multiple scenes follow the same trend as the single feature case.

Our final classification of features is thus as follows:

- Scene-dependent context features: *x*- and *y*- image coordinates, direction of motion, aspect ratio and orientation.

- Scene-independent object features: bounding-box size, variation in size, speed and percentage occu-

Scene S1 Scene S2 Scene S3

Scene S4 Scene S5 Scene S6

Figure 1: The six urban scenes used in our experiments

pancy.

The context features identified by us are not typically used in object classification systems for visual surveillance. We now take a closer look at them to understand why this is so, and how they can help in the classification task.

3.1. Context Features: Benefits and Limitations

To better understand the role of scene-dependent features, we performed a set of experiments with and without the use of these features. We chose two scenes (scenes S1 and S4 in Figure 1) having a total of 500 tracked objects and randomly selected 30 labeled objects (that is, about 800 instances) from each scene as the training sets $T1$ and $T4$ for the respective scenes. We then trained two SVM classifiers for each scene. Classifiers C_{i1} and C_{i4} were trained on T_1 and T_4 respectively using only the four scene-independent features identified above, and then tested on other objects from the same scenes, giving test errors of 9.4% and 3.2%. Classifiers C_{b1} and C_{b4} were trained on T_1 and T_4 respectively using both scene-dependent and scene-independent features. The test errors obtained in this case were 0.7% and 0.8% respectively.

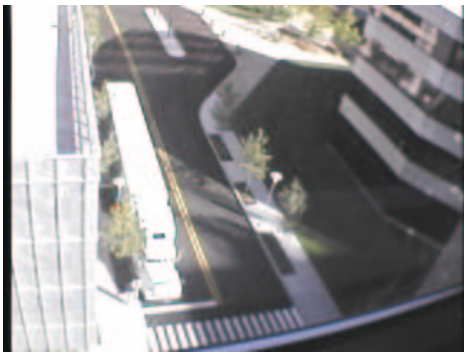
In both cases, the addition of scene-dependent features to the classifier's feature space led to significant improvement in test performance. There are two reasons for this improvement. The first reason is clearly the extra information that scene context features provide to help in classification, independent of the other features. The second reason has to do with the projective distortion introduced by the camera, as a result of which size, speed and direction of motion of objects are all affected by object position. This distortion implies that normalization of image measurements is needed before these features can be used to classify objects. Normalization with a single camera is a hard problem from the point of view of projective geometry, unless some assumptions are made about the scene (such as identifying some planes, angles and distances). However, we want to be able to do this simply by observing data. Using scene-dependent features such as position in the feature space is a non-parametric way of performing normalization. This is clearly demonstrated in Figure 2, where by simply using y-position in the image along with size of bounding-box as object features, and a linear SVM kernel, test error of 3% was obtained for scene S1. Since scene S4 does not have as much relative projective distortion as scene S1 – most of the cars move almost parallel to the image plane – the performance boost obtained by adding scene-dependent features is not as dramatic as in scene S1.

So far, we have established the usefulness of using scene-dependent features, but have not discussed how they can be used in practical surveillance systems. Within the standard supervised learning paradigm, fresh training labels would be required in every novel scene in order to extract scene-dependent features (since it would be a bad idea to train on position or orientation in one scene and test in another scene). Such a scheme is clearly too cumbersome to implement. To get around this limitation, we propose to exploit unlabeled data, as discussed in the next section.

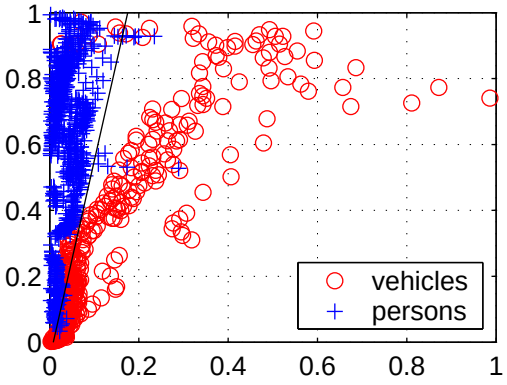
4. Bootstrapping for Scene Transfer

Having identified the scene-independent features, our bootstrapping algorithm for scene transfer is now quite simple.

1. Train a low-performance baseline classifier using the identified scene-independent features, on a set of labeled examples L from a group of urban scenes (or even a single scene).



(a)



(b)

Figure 2: (a) Video frame showing scene S1, in which a significant amount projective foreshortening is evident (*y*-coordinate in the frame increases from 0.0 at the bottom to 1.0 at the top). (b) Using the *y*-coordinate in the image as a normalizing feature for bounding-box size can greatly improve performance, as demonstrated by the 2D feature space for a data set of 32 randomly chosen objects (1227 instances). The abscissa represents image size (scaled by a factor of 10), while the ordinate is the *y*-coordinate of object centroid in the image.

2. Apply this baseline classifier to a large unlabeled data set U in a novel urban scene, after appropriately scaling feature values (so as to equalize the mean feature values in U and L, and ensure that the classification boundary of the baseline classifier is applicable in the new scene). This produces 'labels' (and associated confidence values) for objects in the new scene.

3. Treat the top 10 percent of the new labels in order of confidence values in each class as 'labeled' examples L' for supervised training of a new, scene-specific classifier, using both scene-independent and context features.

Our algorithm is similar to the co-training algorithm of Blum and Mitchell [1] in some respects. Both algorithms incorporate high-confidence labels into the training input for the new classifier. However, their reason for the performance gain when using unlabeled data for co-training is different from ours. While they use two competing types of classifiers that generate the same labels for examples using different techniques, we take advantage of the fact that a label once assigned to an object is valid for each of its instances. For example, if a vehicle's projected image grows smaller as it moves from one end of the field of view to the other, and it is labeled correctly with high confidence by the baseline classifier, then this scene-specific variation in size with position automatically appears in the feature space of training inputs to the new classifier. In fact, we can think of scene-independent and scene-dependent features being analogous to the features used by the two types of classifiers in the co-training algorithm, even though no explicit classifier using only scene-dependent features exists.

Another significant difference between our algorithm and co-training is that the aim of the latter scheme is to enhance performance of existing classifiers, so the new labels are simply used to augment the existing training set. In our case, on the other hand, we cannot use the original training examples in the development of the scene-specific classifier, since the values of the associated scene-dependent features are of no use. Instead, we develop a scene-specific classifier from scratch. Our approach can be considered as a discriminative analogue of decision-directed unsupervised learning [4] in which the scene-independent classifier represents our *a priori* information about the object classes in any scene.

It is useful to note here that the most useful objects for training a new SVM classifier are probably those that the original classifier is least certain about (as is the case in active learning with SVMs, for example). However, since a large proportion of low confidence labels are likely to be incorrect, they simply cannot be used for training in our setup.

The set of novel scenes for which our bootstrapping technique will work is restricted to urban scenes, because our baseline classifier only recognizes vehicles and persons, and will assume that a certain approximate fraction of objects in the scene are vehicles/persons. We use relative size and speed of objects, normalized with respect to the mean object size in a scene, as opposed to actual size, to achieve invariance to scale in our baseline classifier. As long as the fractions of vehicles and humans in the training set and in the novel scene are not very different, our method should work. This is mainly because we use high confidence examples in our scene-specific training set, which are very likely to be correct even if the discrimination of the baseline classifier is not well adapted to the distribution of object features in the particular scene. However, this method will not work when applied, for example, to highway traffic, since hardly any people will be tracked under these circumstances.

5. Experimental Results

We used a data set of more than 1000 object tracks from six different scenes (shown in Figure 1) for testing our al-

Training set used	T_1 (scene S1)	T_1 (scene S1)	T_L (scenes S4/S5)	T'_L (bootstrapped) (scene S1)	T'_L (bootstrapped) (scene S1)
Type of features	Scene-independent	Both	Scene-independent	Both	Scene-independent
Classifier used	C_{i1}	C_{b1}	C_{base}	C_{b1boot}	C_{i1boot}
% overall test error	9.4	0.7	16.5	**8.6**	12.6
% test errors per class	10.0/8.7	0.0/1.3	20.0/12.2	9.3/7.8	13.3/11.3

Table 3: Test errors for scene S1, using various types of classifiers and feature spaces. See Section 5 for descriptions of the training sets and classifiers referred to here. Errors per class are in the format vehicle error/person error. 'Both' refers to use of scene-dependent and scene-independent features together.

gorithms. Since our goal is to evaluate the improvement in classification performance obtained by using context features, we only included vehicles and persons in our data set (and left out other objects such as bicycles and clutter due to trees and reflections). Vehicles constitute 40% of these tracked objects. Objects that were tracked for less than 2 seconds were also filtered out, as reliable values of temporal features could not be obtained for them. Model selection for the scene-independent baseline classifier was carried out on a set R_1 of 50 objects from two scenes to fix the bandwidth parameter σ of the Gaussian kernels. The mutual information calculations described in Section 3 were performed on a separate set R_2 of 80 objects from three scenes. Neither of these sets was used for any subsequent calculations.

The scenes used for our experiments are shown in Figure 1. The average classification error on a test set of about 150 objects in a scene, when training on 30 examples from the same scene, using only scene-independent features, was 6.2%. In contrast, the average classification error when training with both scene-dependent and scene-independent features in each scene was 0.4%.

We present a detailed analysis of one of our scene transfer experiments. The labeled training set T_L, used for training the baseline classifier C_{base}, consisted of 33 objects (14 vehicles and 19 persons) chosen from scenes S4 and S5 (see Figure 1). These labeled objects gave rise to over 1200 labeled training instances, on which C_{base} was trained using only scene-independent features. In the second step, this baseline classifier was applied to a novel scene, S1. Of the 418 test objects in this new scene, the assigned labels $\hat{L}_1$ for 349 objects were correct. Thus, test error for scene transfer without bootstrapping was 16.5%. The average confidence for vehicle labels was 66%, while that for persons was 59%. The difference in confidences between classes is because the range of size and speed variation among persons is much less than the corresponding range of variation among vehicles. To complete the scene transfer process, we chose the top 21 (previously unlabeled) objects from each class in scene S1 which had highest confidence values associated with the labels $\hat{L}_1$, and formed a new training set T'_L specifically for scene S1. None of the labels assigned to the objects in T'_L were incorrect, and the minimum confidences for vehicle and person examples in this set were 78% and 65% respectively. A bootstrapped, scene-specific classifier, C_{b1boot}, was then trained on T'_L, using both scene-dependent and scene-independent features. The test error for this new scene-specific classifier in the same scene was 8.6%. Thus, our bootstrapping technique resulted in a performance boost of about 8% for this particular scene.

The above results are summarized in Table 3. For comparison, the results of training scene-independent and scene-specific classifiers (C_{i1} and C_{b1} respectively) on a labeled set T_1 taken from scene S1 itself are also provided. As expected, best classification results are obtained by training on T_1, and using both scene-dependent and scene-independent features. The bootstrapped classifier working with both types of features, C_{b1boot}, demonstrates a significant improvement over the baseline classifier C_{base} that uses only scene-independent features. By training another bootstrapped classifier using only scene-independent features, C_{i1boot}, it is found that about half the improvement is simply because the new classifier is able to better adapt to the values of scene-independent features in the novel scene than the baseline classifier, while the other half is because scene-dependent features are able to exploit scene-specific constraints.

We performed three more scene transfer experiments, by training in a single scene with scene-independent features and bootstrapping a scene-specific classifier in another scene. The average reduction in test error obtained by replacing the baseline classifier with the adapted classifier was 5.5%.

Cases where the classifiers trained on individual scenes failed included groups of three or more people walking together and vehicles that were always far away from the camera and small in size.

6. Conclusions and Future Work

We have demonstrated the advantages of using scene-dependent features for object classification from video data, and proposed a simple bootstrapping algorithm for scene-transfer without having to obtain labeled data from every scene. The system we present can be quickly set up for outdoor surveillance with a minimal period of unsupervised learning. Subsequently, even if camera parameters or the distribution of objects in the scene change, the object classifier can automatically adapt to the new scene characteristics by using the same bootstrapping technique.

In order to realize the significance of a 5% performance enhancement, one should keep in mind that object classification is often the first step towards event detection in surveillance systems. Practical systems may process over 5000 objects in a day, so even a 1% improvement would result in 50 fewer errors. This could be crucial if our goal is to detect anomalous events, which are expected to be rare in the first place.

The use of both labeled and unlabeled data can be thought of as a balance between two extremes: completely unsupervised learning (or clustering) of object classes (as in [12]) and purely supervised object recognition. A minimum amount of supervision is generally acknowledged to be necessary for obtaining perceptually meaningful object classes. On the other hand, since learning is mostly carried out in a high dimensional 2D image space, effective results may only be attained after providing a huge number of labeled examples, which is often intractable. One future possibility for research is to provide some information at later stages of learning, perhaps within an active learning or reinforcement learning framework.

In the present approach, a problem arises if the distribution of features of the 'high-confidence' objects is very different from that of the remaining objects. We are looking into other possibilities for selecting objects for training the scene-specific classifier, such as choosing objects which have wide variation in features in the course of their trajectories. Another issue that has not been addressed here is multi-class classification. While filtering out of clutter is mainly handled by the object tracker in our case, we would like to consider other classes of objects, such as dogs or bicycles, that may be important for event detection.

Acknowledgments

Tracking data were provided by Chris Stauffer, along with help on how to use them. Biswajit would like to thank Greg Shakhnarovich, Tommi Jaakkola, Kinh Tieu and Gerald Dalley for extremely useful discussions.

References

[1] Blum, A. and Mitchell, T., "Combining Labeled and Unlabeled Data with Co-Training," *Proc. Conf. on Computational Learning Theory*, 1998.

[2] Burges, C., "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998.

[3] Cover, T.M. and Thomas, J.A., *Elements of Information Theory*, John Wiley and Sons, Inc., 1991.

[4] Duda, R.O., Hart, P.E. and Stork, D.G., *Pattern Classification*, 2nd ed., John Wiley and Sons, Inc., 2001.

[5] Javed, O. and Shah, M., "Tracking and Object Classification for Automated Surveillance," *Proc. European Conf. on Computer Vision*, 2002.

[6] Joachims, T., "Transductive Inference for Text Classification using Support Vector Machines," *Proc. 16th Intl. Conf. on Machine Learning*, 1999.

[7] Kanade, T., Collins, R., Lipton, A., Burt, P. and Wixson, L., "Advances in Cooperative Multi-Sensor Video Surveillance," *DARPA Image Understanding Workshop*, November 1998, pp. 3-24.

[8] Lipton, A.J., Fujiyoshi, H. and Patil, R.S., "Moving target classification and tracking from real-time video," *Proc. IEEE Workshop on Applications of Computer Vision*, 1998.

[9] Marr, D. and E. Hildreth, "Theory of Edge Detection," *Proc. R. Soc. London*, B 207, pp. 187-217, 1980.

[10] Platt, J. C., "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," *Advances in Large Margin Classifiers*, Smola et al., eds., MIT Press, 1999.

[11] Seeger, M., "Learning with labeled and unlabeled data," Technical report, University of Edinburgh, 2000.

[12] Stauffer, C. and Grimson, E., "Learning patterns of activity using real-time tracking," *Proc. IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000.

[13] Wu, Y., Huang, T.S. and Toyama, K., "Self-Supervised Learning for Object Recognition based on Kernel Discriminant-EM Algorithm," *Proc. IEEE Int'l Conf. on Computer Vision*, 2001.