Improving Object Classification in Far-Field Video

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 video sequences is a challenging problem because of low resolution imagery and projective image distortion. Most existing far-field classification systems are trained to work well in a constrained set of scenes, but can fail dramatically when applied to new scenes, or even different views of the same scene. We identify discriminative object features for classifying vehicles and pedestrians and develop a scene-invariant classification system that is trained on a small number of labelled examples from a few scenes, but transfers well to a wide range of new scenes. Simultaneously, we demonstrate that use of scene-specific context features (such as image position and direction of motion of objects) can greatly improve classification in any given scene. To combine these ideas, we propose a new algorithm for adapting a scene-invariant classifier to scene-specific features by retraining with the help of unlabelled data in a novel scene. Experimental results demonstrate the effectiveness of our context features and scene-transfer/adaptation algorithm for multiple urban and highway scenes.

1. Introduction

There has been significant interest in the recent past in detection and classification of moving objects (such as vehicles and humans) in far-field video sequences. Examples include general purpose object tracking and classification systems [6, 9] as well as specialised systems for vehicle [7, 8] or pedestrian [16] detection. In contrast to near-field settings, the far-field domain is particularly challenging because of the low resolution of acquired imagery: objects are generally less than 100 pixels in height, and may be as small as 50 pixels in area. Under these conditions, local appearance-based features (such as body-parts of humans) cannot be reliably extracted. Further, as many far-field cameras have large fields-of-view, extracted object features may show significant projective distortion—nearby objects appear to be larger and to move faster than objects far away.

To achieve reasonable performance given these challenges, system designers typically train classifiers for optimal performance in a particular situation (*i.e.* specific position and orientation of camera, fixed scale and known distribution of objects). Such systems are restricted to interpreting activity in a fixed environment, and cannot be used in a new scene, or if the original scene characteristics change (*e.g.* due to change in camera position or zoom), without manual retraining. For projects involving hundreds of cameras deployed throughout a city or a network of highways, manually fixing the parameters of each camera is impractical. Also, when dealing with large quantities of data, low classification error is necessary to avoid overloading the system operator with false-positive results.

Our goal is to address the conflicting requirements of transferring classifiers across scenes without repeated manual supervision and of achieving high performance in any single scene. We show that certain 'scene-specific' local context features (such as image-position and direction of motion of objects) can help augment classification performance in any given scene. The challenge lies in the fact that these scene-specific features have different distributions in different scenes, which cannot all be represented by any single set of training data. To achieve both our goals simultaneously, we first identify 'scene-invariant' features and use them to design a baseline classifier. We then adapt this classifier to any given scene by learning scene-specific features with the help of unlabelled data.

In situations where labelled data are limited, semi-supervised learning methods have been used to improve classifier performance [12]. In these cases, the same feature space is populated by a few labelled examples and many unlabelled examples. In our learning problem, however, some features are scene-specific—they have completely unrelated distributions for the labelled and unlabelled data—and can thus only be learned from unlabelled data in the new scene. To this end, we propose a novel learning procedure which involves retraining a baseline scene-invariant classifier after appropriately weighting the 'labels' produced for unlabelled data by this classifier.

This paper makes three key contributions to object classification in far-field video sequences. The first is the choice of suitable features for far-field object classification from a single, static, uncalibrated camera and the design of a principled technique for classifying objects of interest (vehicles and pedestrians in our case). The second is the introduction

of scene-specific context features for improving far-field object classification performance, and the use of mutual information estimates to separate such features from scene-invariant features. The third is a composite learning algorithm that not only produces a scene-invariant baseline classifier that can be transferred across scenes, but also adapts this classifier to a specific scene (using context features) by passive observation of unlabelled data.

2. Related Work

Two complementary approaches are commonly used for object localisation and categorisation. One approach is to use background subtraction or optical flow for detection of moving objects [6, 9]. Detected objects are then tracked, and object classification is performed on the tracked object regions using simple descriptors of shape or motion. The other approach, direct image-based detection of object classes, does not rely on object motion. Instead, each video-frame is scanned for regions having the characteristic appearance of objects of interest, such as vehicles [7, 8] or pedestrians [10]. These methods typically use combinations of simple image features such as edges, wavelets, or rectangular filters, and require training on large labelled datasets, especially for low-resolution far-field images.

Most detection-based methods have severe problems with false positives, since even a false-positive rate as low as 1 in 50,000 can produce one false positive every frame [16]. To get around this problem, Viola et al. [16] have proposed a pedestrian detection system that combines appearance and motion cues by working on a pair of images. To achieve desired results, they use 4500 labelled training examples for detecting a single class of objects, and manually fix the scale to be used for detecting pedestrians.

Methods based on background subtraction followed by object tracking do not suffer from the problem of false positives or scale selection, and have been demonstrated to run in real-time [14]. These methods may also be able to track robustly through temporary occlusion and clutter. While these methods are only applicable to static (or pan-tilt-zoom) cameras, long-term scene analysis (*i.e.* more than a few hours) requires precisely such a setup.

Statistical priors on object characteristics in a scene can greatly help classification. Contextual information (such as likely scales or positions of objects) may be manually specified by an operator for a given scene or learnt from examples [15] and used to prime object detection. However, to the best of our knowledge, no previous work has been done on learning local context features (such as position and direction of motion of objects) solely from long-term observation of moving object, to improve classification performance in a given scene.

Obtaining hundreds of labelled training examples for object classifiers is not easy. Methods based on exploiting unlabelled data provide a useful alternative. Levin *et al.* [8] use a co-training algorithm [1] to help improve vehicle detection using unlabelled data. Stauffer [13] makes use of multiple observations of a single object (obtained from tracking data) to effectively exploit unlabelled data for object classification.

Scene transfer is not addressed by any of the above object classification methods. Most methods make implicit assumptions about the distribution of object features in a scene, as is evident from the training and test sets used (such as using only side views, only front views or only top views). Our goal is to develop a completely automated system that can be trained once on data from a few scenes, but will work when transferred to any of a wide range of far-field scenes.

3. Object Classification from Video Sequences

We use the background subtraction and tracking system of Stauffer and Grimson [14] to detect and track moving objects. Since the objects whose activities are of interest will move, they will (mostly) be detected by background subtraction. After tracking these foreground regions, classification is simply a matter of assigning a class label to each detected sequence of foreground images in a track. Our object-class detector can be much simpler than those used in detection-style algorithms, since it only needs to discriminate between foreground objects, not between object and background. Thus, fewer training examples are sufficient for good performance. In addition, tracking produces a sequence of images of the same object, thus providing a constraint that enables more robust classification than that possible from a single image.

To demonstrate our algorithms for scene-transfer and scene-adaptation, we consider classification of vehicles and pedestrians. As a pre-processing step, we automatically filter the tracking data to remove irrelevant clutter, thus reducing the classification task to a binary decision problem. Filtering is an important step for long-term surveillance, since (random sampling shows that) more than 80% of detected moving regions are actually spurious objects. The features we used for filtering clutter are minimum and maximum size of foreground region (to filter abrupt changes in lighting), minimum duration, minimum distance moved (to filter shaking trees and fluttering flags) and temporal continuity (since apparent size and position of objects should change smoothly). Even after filtering out clutter, there are certain classes of objects that are neither vehicles nor pedestrians, such as groups of people and bicycles. These were manually removed; including them in the analysis is left for future work.

3.1. Useful Video Features

Low resolution data in far-field video prevent us from reliably detecting parts-based features of objects using edge- and corner-descriptors. Instead, we use spatial moments of

object silhouettes (and their time derivatives), which provide a global description of the projected image of the object. The list of object features we consider is given in Table 1. Variation in area refers to the second derivative of number of pixels as a function of time, normalised by the mean area; it is expected to be higher for humans than for vehicles. Percentage occupancy is the number of silhouette pixels divided by the area of a bounding-box aligned with the principal axis of the silhouette. The significance of the mutual information scores mentioned in this table is discussed in Section 4.

Video sequences provide us with two kinds of features, which we call instance features and temporal features. Instance features, such as the position of a silhouette's centroid, are those that can be calculated from a single instance of an object (that is, from each frame of a tracking sequence). Temporal features, on the other hand, are features that cannot be obtained from a single frame: the mean aspect ratio or the Fourier coefficients of image-size variation, for example. Temporal features can usually be converted into instance features by calculating them over a small window of frames in the neighbourhood of a given frame. For example, apparent velocity of the projected object is calculated in this way.

3.2. Classifying Sequences of Observations

Tracked objects can be processed in two ways: classifying individual instances separately (using instance features) and combining the instance-labels to produce an object label, or classifying entire object tracks using temporal features. We chose an instance classifier, since labelling a single object produces many labelled instances. This helps in learning a more reliable classifier from a small set of labelled objects.

Each detected object is represented by a sequence of observations, $\mathbf{O} = \{O_i\}, 1 \leq i \leq n$, where each O_i is a vector in the object feature space and n is the number of frames for which the object was tracked. Classification of this object as a vehicle or pedestrian can be posed as a binary hypothesis testing problem, in which we choose the object class label l_j following the maximum-likelihood (ML) rule [5] (*i.e.* choose l_j corresponding to the higher class-conditional density $p(\mathbf{O}|l_j)$). We use the ML rule instead of the maximum a posteriori rule because we found the prior probabilities, $p(l_j)$, to be strongly scene-dependent. To develop a scene-invariant classifier, we assume $p(l_1) = p(l_2)$.

The likelihood-ratio test involves evaluation of $p(O_1, ..., O_n|l_j)$, the joint probability of all the observations conditioned on the class label. For images of a real moving object, this joint distribution depends on many factors such as object dynamics and imaging parameters. A simplifying Markovian approximation could be made, but the parameters of the resulting conditional probabilities (of each observation given its recent neighbours in the sequence) vary with the position of the observation due to projective distortion. Instead, we search for (approximately) independent observations in the sequence, and use the fact that the joint probability for independent samples is simply given by $\prod_{i=1}^{n} p(O_i|l_j)$. The probabilities $p(O_i|l_j)$ can in turn be obtained from the posterior probabilities of the labels given the observations, $p(l_j|O_i)$, by applying Bayes' rule. This means that our classifier can be run separately on each independent observation in a sequence, to produce the corresponding posterior probability of the class label. We approximate independent samples by looking for observations between which the imaged centroid of the object moves by a minimum distance. In our implementation, the minimum distance threshold is equal to the object-length. This is useful, for example, to avoid using repeated samples from a stopped object (which is quite common for both vehicles and persons in urban scenes).

3.3. Classification with Support Vector Machines

In choosing a suitable classifier, we considered using a generative model, but decided against it to avoid estimating multi-dimensional densities from a small amount of labelled data. Instead, we chose a discriminative model—support vector machine (SVM) with soft margin and Gaussian kernel—as our instance classifier. The use of a soft margin is necessary since the training data are non-separable. In terms of the SVM formulation, we are looking for the maximum-margin separating hyperplane for the N training points $\mathbf{x}_i \in \Re^k$ and corresponding labels $y_i \in \{-1, 1\}$, given the (dual) optimisation problem [3]:

$$\text{Maximise} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \qquad (1)$$

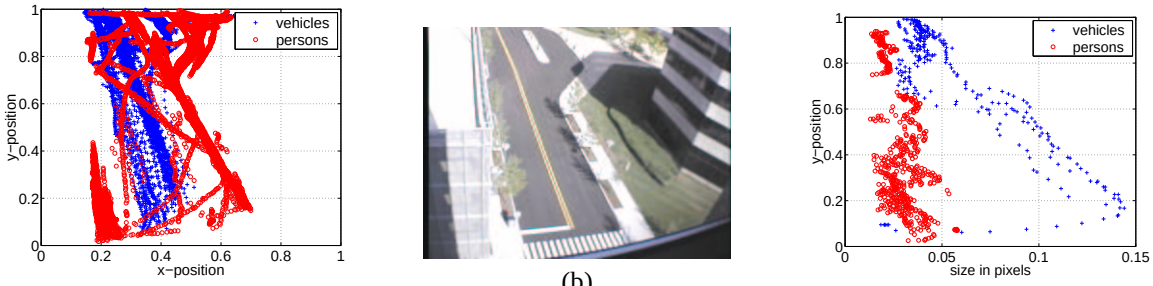
$$\text{subject to} \quad \sum y_i \alpha_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq B, \quad i = 1, 2, ..., N \qquad (2)$$

where α_i are the Lagrange multipliers (with upper-bound B) and K is the SVM kernel function.

It is useful to distinguish between support vectors for which $\alpha_i < B$ (margin vectors) and those for which $\alpha_i = B$ (mostly either misclassified points or correctly classified points within the margin). For training the baseline (scene-invariant) classifier, we fixed B to a large value (= 1,000). Our scene-adaptation algorithm (Section 5), however, uses different values of B for different 'labelled' examples.

One disadvantage of using SVMs is that the output, d_i, is simply the signed distance of the test instance from the separating hyperplane, and not the posterior probability of the instance belonging to an object class. Posterior probabilities are needed to correctly combine instance labels using the ML rule to obtain an object label. Thus, we retrofit a logistic function $g(d_i)$ that maps the SVM outputs d_i into probabilities [11]:

$$g(d_i) = \frac{1}{1 + exp(-d_i)}. \qquad (3)$$



(a) (b) (c)

Figure 1: a. Scatter plot illustrating spatial distribution of vehicles and persons in scene S1 (shown in b.), in which significant projective foreshortening is evident. c. Using the *y*-coordinate as a normalising feature for bounding-box size can greatly improve performance, as demonstrated by the fact that vehicles and pedestrians are clearly distinguishable in the 2D feature space.

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

$$l(\lambda) = \sum_{i=1}^{N} log\, P(y_i \mid d_i, \lambda). \quad (4)$$

The posterior probabilities thus obtained are used both for classifying sequences of observations and for associating confidences with the 'labels' of the classified objects for use in adapting the classifier to a specific scene (Section 5). To clarify, test error corresponds to the fractional number of incorrect *object* labels, not instance labels.

4. Identifying and Using Scene-Specific Features

Broadly speaking, we refer to those features which are useful for classification (*i.e.* training and testing) in any single scene, but not for training in one scene and testing in another scene, as scene-specific context features. All other (useful) features are said to be scene-invariant features.

For some features, such as image-position of an object, it is clear that the feature is scene-specific. However, for others features, such as aspect-ratio or orientation, it is not obvious whether the feature is scene-specific or scene-invariant[1]. Therefore, we estimate mutual information scores between features and labels for two data-sets—data taken from a single scene, and data drawn from multiple scenes—in order to identify scene-specific features. We formalise our definition of scene-specific features based on these scores. However, before describing these calculations, let us gain some insight into how scene-specific context features help.

4.1. Benefits of Using Scene-Specific Features

We performed a pair of experiments with and without scene-specific features, to demonstrate the role played by them. We chose 500 objects from two scenes (scenes S1 and S4 in Figure 2) and randomly selected 30 objects from each scene as training sets $T1$ and $T4$ for the respective scenes. Two SVM classifiers were trained for each scene, with orientation of principal axis and variation in area of silhouette used in every case. Classifiers C_{i1} and C_{i4} were trained on T_1 and T_4 respectively without using position or direction of motion as features, and then tested on other objects from the same scene, giving test errors of 9.4% and 3.2%. Classifiers C_{b1} and C_{b4} were trained on T_1 and T_4 respectively after including position and direction of motion in the feature space. The test errors obtained in this case were 0.7% and 0.8% respectively. Thus, in both scenes, the addition of context features to the classifier's feature space led to significant improvement in test performance.

Context features capture the inherent regularities in structured scenes. For instance, different spatial distributions of object classes in an urban scene are a result of the scene structure, *i.e.*, roads and footpaths. While detecting roads and footpaths reliably is a hard problem, it is much easier to learn the spatial class distribution from labelled data (as shown in Figure 1a), and use this for enhancing object classification. In the absence of structural regularities—in an open field, for example—context features would not provide extra information. Fortunately, most urban/highway scenes do exhibit some degree of structure.

As a result of the projective distortion introduced by the camera, area and speed of objects vary with object position. Normalisation of image measurements by correcting for this distortion will help to classify objects reliably. However, normalisation with a single camera is a difficult problem unless some constraining assumptions are made. Using image position as a feature is a non-parametric way of performing normalisation. This is demonstrated in Figure 1c, where by simply using *y*-position in the image along with area of bounding-box as object features, and a linear SVM kernel, test error of 3% was obtained for scene S1.

[1]It becomes even harder to decide when more features are added, as evident from our recent work [2].

Feature	M.I. for a single scene	M.I. for multiple scenes
x-coordinate	0.35	0.05
y-coordinate	0.28	0.04
area in pixels	0.71	0.33
speed	0.42	0.20
direction of motion	0.16	0.08
aspect ratio	0.49	0.07
variation in area	0.35	0.23
orientation	0.47	0.33
percentage occupancy	0.31	0.30
average intensity	0.15	0.08

Table 1: Mutual information (M.I.) between object features and labels, measured in bits (max. possible score = 1.0)

Feature 1	Feature 2	M.I.
x-coordinate	*y*-coordinate	0.89
y-coordinate	area	0.81
percentage occupancy	area	0.76
orientation	direction of motion	0.73
x-coordinate	area	0.72

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

4.2. Feature Selection and Grouping

Feature selection is necessary in order to remove non-informative features, and thus avoid overfitting when training using a small labelled data set. Feature grouping is necessary to separate scene-invariant features from scene-specific features, so that the former can be used in training a scene-invariant classifier.We perform both feature selection and feature grouping 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 $p(\mathbf{x} \mid y)$, of instance features $\mathbf{x}$ and labels y non-parametrically, using Parzen-window density estimators [5]. Window size is chosen by cross-validation. The mutual information of the j^{th} feature with the label (after discretising the continuous probability densities) is then given by [4]

$$I(X_j; Y) = \sum_{x_j \in \mathcal{X}} \sum_{y \in \{-1,1\}} p(x_j, y)\, log\, \frac{p(x_j, y)}{p(x_j)\, p(y)}. \quad (5)$$

Two sets of mutual information (MI) calculations were performed: the first using examples from a single scene, and the second using examples collected from a set of seven scenes. The scenes were chosen so as to adequately represent common variations in height, viewing angle and zoom of camera. Equal numbers of vehicle and pedestrian images from each scene were used.

MI scores for our chosen features in the two cases are shown in Table 1. We are mainly interested in comparing relative scores across features and scenes. Unfortunately, simply calculating MI between individual features and labels and choosing the features with the highest scores is not guaranteed to give the most informative features for a classification task. In order to be (theoretically) optimal, MI scores need to be calculated between all possible sets of features and labels. To see this, consider a road cutting diagonally across a scene and people walking on footpaths beside it: the MI of *x*- and *y*- image coordinates with the labels will be low, but the two coordinates taken together can accurately classify the object. In practice, it is unlikely that three or more features calculated from real-world scenes will conspire to give significantly better MI scores than pairs of features acting together. Therefore, we repeat our MI calculations for all possible pairs of features; the most relevant results (corresponding to the highest scores) are given in Table 2.

Based on these scores, our features can be grouped into two categories. Features (or groups of features) whose MI with labels is high within a scene, but drops significantly—by a factor of 2 or more—for a group of scenes, are considered to be scene-specific context features. These clearly include the *x*- and *y*- image coordinates and the area in pixels. Aspect-ratio, too, depends to a great extent on the elevation of the camera. Of the remaining features, those that have reasonably high MI values—greater than 0.2 bits—both within and across scenes are considered to be scene-invariant features. Orientation and variation in area are good examples of such features. Orientation is a useful feature since the vertical world direction projects to the vertical axis in the image for most camera setups, so that pedestrians have an almost constant orientation. Direction of motion and orientation have significantly higher MI with the object label when considered jointly rather than singly. This is because vehicles tend to be oriented in the direction of their motion in most scenes. However, the angle between these two features is corrupted by shadows in a different way in each scene, so that the MI with the labels across scenes is low (0.35). A third category consists of features which have low MI—less than 0.2 bits—in both cases, and are not used as they are likely to lead to over-fitting. Average brightness of the tracked object falls into this category.

Our final grouping of features is as follows:

- Scene-invariant features: orientation, variation in area, percentage occupancy
- Scene-specific features: *x*-, *y*- image coordinates, area in pixels, speed, direction of motion, aspect ratio
- Non-informative feature: average object brightness

Most classification methods do not transfer to novel scenes mainly because they use object area, aspect-ratio or similar features—features we identify here as scene-specific—in their classifier's feature space. Note that even though area has a reasonable score across a group of scenes, the fact that


Scene S1 Scene S2 Scene S3 Scene S4 Scene S5


Scene S6 Scene S7 Scene S8 Scene S9 Scene S10

Figure 2: The full set of scenes used in our experiments

this score is much lower than the score in a single scene implies that it is a rather poor feature when transferring to at least some subset of scenes.

We have shown that scene-specific context features can be used for reducing classification error when training and testing in the same scene. While this in itself is useful for many surveillance applications, we really would like to be able to transfer classifiers across scenes, while still enjoying the benefits of using context features. We now propose a scene transfer and adaptation algorithm to do exactly this.

5. Scene Transfer and Adaptation

Having identified scene-specific and scene-invariant features, we now describe the main learning algorithm for achieving both scene-transfer and scene-adaptation.

5.1. Baseline, scene-invariant classifier design

The design of a scene-invariant classifier is now perhaps obvious: train a classifier using only scene-invariant features on a labelled set of examples from 2 or 3 scenes (or even a single scene). The classification accuracy of our baseline classifier is around 80%. More importantly, the average posterior probability of the label given an observation in a new scene is higher for correctly classified test examples (0.41) than for incorrectly classified ones (0.17). Our scene-adaptation algorithm is now employed to further improve classification performance by using unlabelled data to learn scene-specific features.

5.2. Adapting a baseline classifier to a novel scene

We propose the following novel decision-directed learning [5] algorithm for scene-adaptation:

1. Apply the baseline classifier, C_{base}, to the new scene, to find 'labels', L_1, for unlabelled objects, along with associated posterior probabilities.

2. Convert posterior probabilities (0 to 1) to confidences (-100% to 100%) by shifting and scaling. (Note that the sign of the posterior probability of a single observation given a class may differ from the overall 'label' for the unlabelled object. This is indicated by a negative confidence value.)

3. Train a scene-specific classifier using only the scene-invariant features on both the original labelled examples L_0 and the 'labels' L_1 generated for unlabelled data, after making two changes:

   - For each unlabelled object, the 5% least confident instances are removed from further consideration (to afford some robustness to gross outliers). Each remaining instance is then assigned the same confidence value, equal to the mean confidence of these instances.
   - The bound on the Lagrange multiplier (in the SVM formulation: see Equation 2) for each training instance from the ith unlabelled object is set as $B_i = 1000 \times$confidence value.

   This step produces a partially-adapted classifier, C_{part}, which does not yet use scene-specific context features.

4. Apply C_{part} to the unlabelled data to generate a second set of 'labels', L_2, and associated confidences for unlabelled data.

5. Repeat step 3, this time using both scene-specific and scene-invariant features for training, but only the 'labels' in L_2, to obtain a fully-adapted classifier, C_{full}.

Since, by definition, true labels are not available for unlabelled data, there is a trade-off involved: high-confidence examples are far from the decision boundary, and thus not very informative, while low-confidence examples are close to the boundary and thus more informative, but also more likely to be incorrect. A balance is obtained by varying the Lagrange multipliers for these 'labelled' instances in

proportion to the confidences, since a large Lagrange multiplier heavily penalises an incorrect classification of the corresponding training example. Thus, our algorithm is able to allow points near the classification boundary (of the baseline classifier) to modify the adapted solution slightly, without letting incorrect (but low-confidence) 'labels' significantly disrupt the training process. The equalisation of confidences within a tracking sequence (in step 3) is done to avoid using high confidence outliers for retraining. Our underlying assumption is that incorrectly classified objects will have lower average confidence than correctly classified ones.

The classifier needs to be adapted in two steps (3 and 5) because the distribution of scene-specific features in the labelled and unlabelled data may be completely unrelated. Also, a two-step process gradually removes the information provided by true labels and increases reliance on the (uncertain) information provided by the new scene.

6. Experimental Results

We used a data set of more than 1000 object tracks from ten different scenes (Figure 2) for testing our algorithms. Model selection to fix the bandwidth of the SVM Gaussian kernel was carried out on a set of 50 objects. The MI calculations described in Section 4 were performed on a separate set of 80 objects from 7 scenes. The remaining object tracks were used for training and testing.

We performed two types of experiments:

- Without scene transfer: training on 30 objects, testing on 150 objects in the same scene.
- With scene transfer/adaptation: training on 30 objects from 2-3 scenes, testing on 150 objects in a new scene.

Average classification errors in these cases (averaged over 5 trials) are as follows:

- Without scene transfer, using only scene-invariant features: 8.2%
- Without scene transfer, using both scene-specific and scene independent features: 0.5%
- With scene transfer, using only scene-specific features (baseline classifier): 17.5%
- With scene transfer and scene adaptation (using both types of features): 8%

The average reduction in error due to adaptation with the help of scene-specific context features is thus 9.5%.

We present a detailed analysis of one scene-transfer/adaptation experiment. The labelled set T_L, used for training the baseline classifier C_{base}, consisted of 30 objects (17 vehicles and 13 persons—300 instances from each class) from scenes S4 and S5 (see Figure 2). This baseline classifier was applied to a novel scene, S1. Of the 150 test objects in this new scene, the assigned labels $\hat{L}_1$ for 125 objects were correct. Thus, test error (after scene transfer, but without adaptation) was 16.7%. This is comparable to some existing classification systems which are trained and tested on the same scene. The average confidence for vehicle labels was 47%, while that for persons was 37% (note that 0% represents no confidence, as it corresponds to a posterior probability of 0.5). Average confidence for correct labels was 51%, while that for incorrect labels was 14%. In the scene adaptation process, bounds on the Lagrange multipliers were varied according to the average object confidences, as described in Section 5.2. After partial adaptation, test error (using only scene-invariant features) decreased to 14.4%. After full adaptation, the error further decreased to 6.1%. Thus, our bootstrapping technique resulted in a performance boost of over 10% for this particular scene.

The above results are summarised in Table 3. For comparison, the results of training scene-invariant and scene-specific classifiers on a labelled set T_1 taken from scene S1 itself (Section 4.1) are also repeated there. As expected, best results are obtained by training on T_1 using both scene-specific and scene-invariant features. The fully-adapted classifier, working with both types of features, demonstrates a significant improvement over the baseline classifier, and even the partially-adapted classifier. This is because of the significant projective distortion evident in this scene, as well as the characteristic spatial distribution of vehicles and pedestrians. The resulting classification performance is better than simply using scene-invariant features for training in scene S1 itself.

In general, cases where the original scene-specific classifier (before transfer) failed include partial occlusion (as the object leaves the scene) and objects that are consistently far away from the camera (and hence have areas of around 10×5 pixels, producing very noisy features). Additional cases where the transferred and adapted classifier fails (but the original classifier works) include objects that show large feature variations as they move through the scene, or scenes with prominent shadows. Note that, in practice, overall classification error will be higher than that reported above (by an estimated 5 to 7 percent) because of tracking errors and the presence of multiple classes of objects.

7. Conclusions and Discussion

We have proposed a far-field object classification system that addresses some of the significant challenges in the field. Use of a discriminative (SVM) instance-classifier on simple object descriptors, along with a probabilistic method for combining instance confidences into object labels, allows for very low classification error (less than 1%) using only a small number of objects. Scene-specific context features are introduced and shown to provide numerous benefits. Scene-invariant and scene-specific features are identified using MI estimates in order to design scene-invariant

	a	b	c	d	e
Labelled training set scene	$(S1)$	$(S1)$	$(S4/S5)$	$(S4/S5)$	None
Unlabelled 'training set' scene	None	None	None	$(S1')$	$(S1'')$
Test set scene	$(S1)$	$(S1)$	$(S1)$	$(S1)$	$(S1)$
Transfer ?	No	No	Yes	Yes	Yes
Adaptation ?	No	No	No	Yes (Partial)	Yes (Full)
Type of features	S.I.	Both	S.I.	S.I.	Both
% Test error	9.4	0.7	16.7	14.4	6.1

Table 3: Performance evaluation for scene $S1$ (Figure 2): test errors using various classifiers and features. 'S.I.' = scene-invariant features. 'Both' = scene-specific + scene-invariant features. Labels for $S1'$ are produced in step c, and those for $S1''$ in step d.

classifiers. At the same time, a decision-directed learning algorithm has been proposed to adapt classifiers to scene-specific characteristics by carefully using unlabelled data. Our scene-invariant classifier has over 80% accuracy; further improvement of about 10% is obtained by using our scene-adaptation algorithm.

In future, we would like to extend the classification framework to other object classes (*e.g.* groups of people) or sub-classes (*e.g.* cars, vans and trucks). More features for classification have been added and analysed in our recent work [2]. We would also like to evaluate how much of the improvement in classification when using image position as a feature is a result of the implicit normalisation achieved, and how much is due to the structural regularities in the scene.

Acknowledgements

B.B. would like to thank Kinh Tieu, Polina Golland and John Fisher for extremely useful discussions. The work presented here was funded in part by DARPA grant N00014-00-1-0907.

References

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. COLT*, 1998.

[2] Biswajit Bose. Classifying tracked objects in far-field video surveillance. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2004.

[3] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

[4] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley and Sons, 1991.

[5] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[6] O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *Proceedings ECCV*, 2002.

[7] Z. Kim and J. Malik. Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In *Proc. ICCV*, 2003.

[8] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *Proc. ICCV*, 2003.

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

[10] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proc. IEEE CVPR*, pages 193–199, 1997.

[11] J.C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Smola et al., editor, *Advances in Large Margin Classifiers*. MIT Press, 1999.

[12] M. Seeger. Learning with labeled and unlabeled data. Technical report, University of Edinburgh, 2001.

[13] C. Stauffer. Minimally-supervised classification using multiple observation sets. In *Proc. ICCV*, 2003.

[14] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. PAMI*, 22(8):747–757, August 2000.

[15] A. Torralba and P. Sinha. Statistical context priming for object detection. In *Proc. ICCV*, pages 763–770, 2001.

[16] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. ICCV*, 2003.