Semi-Supervised Self-Training of Object Detection Models

Chuck Rosenberg
Google, Inc.
Mountain View, CA 94043
chuck@google.com

Martial Hebert
Carnegie Mellon University
Pittsburgh, PA 15213
hebert@ri.cmu.edu

Henry Schneiderman
Carnegie Mellon University
Pittsburgh, PA 15213
hws@ri.cmu.edu

Abstract

The construction of appearance-based object detection systems is time-consuming and difficult because a large number of training examples must be collected and manually labeled in order to capture variations in object appearance. Semi-supervised training is a means for reducing the effort needed to prepare the training set by training the model with a small number of fully labeled examples and an additional set of unlabeled or weakly labeled examples. In this work we present a semi-supervised approach to training object detection systems based on self-training. We implement our approach as a wrapper around the training process of an existing object detector and present empirical results. The key contributions of this empirical study is to demonstrate that a model trained in this manner can achieve results comparable to a model trained in the traditional manner using a much larger set of fully labeled data, and that a training data selection metric that is defined independently of the detector greatly outperforms a selection metric based on the detection confidence generated by the detector.

1. Introduction

1.1. Object Detection

Object detection systems based on statistical models of object appearance have been quite successful in recent years [18], [19], [17], [23]. Because these systems directly model an object's appearance in an image, a large amount of labeled training data is needed to provide good coverage over the space of possible appearance variations. However, collecting a large amount of labeled training data can be a difficult and time-consuming process. In the case of the training data for appearance-based statistical object detection, this typically entails labeling which regions of the image belong to the object of interest and which belong to the non-object part of the image, and, in some cases, marking landmark points. For many of the object detection techniques to be practical, it is crucial that a streamlined approach to training be used so that users are able to rapidly insert new object models in their systems.

The goal of the approach proposed here is to simplify the collection and preparation of this training data by utilizing a combination of data labeled in different ways. In what we call "weakly labeled" training data, the labeling of each of the image regions can take the form of a probability distribution over labels. This makes it possible to capture a variety of information about the training examples. For example, it is possible to indicate that the object of interest is more likely to be present toward the center of the image. Or it is possible to encode the knowledge that a specific image has a high likelihood of containing the object, but that the object's position is unknown. We refer to this type of training as "weakly labeled" or "semi-supervised".

In the recent literature, [20], [8], anecdotal evidence has been presented which suggests that semi-supervised training can provide a performance improvement when applied to the object detection problem. In the work presented here, we perform a comprehensive empirical evaluation with the goal of characterizing and understanding these issues in order to facilitate the broad practical application of semi-supervised training to the object detection problem. Although, for practical reasons, we use one detector for evaluation, the selected detector is representative of other recent algorithms in the literature. We believe that, in the context of computer vision, this is the first comprehensive scale study of semi-supervised training techniques which will be necessary in any practical application of object detection algorithms.

1.2. Training Approaches

In order to introduce the general approaches to semi-supervised training, let us first describe a generic detection algorithm that classifies a subwindow in an image as being a member of the "object" class or the "clutter" class. The classification is based on the values of the feature vectors associated with each subwindow in the image.

We designate the image feature vectors as X, with x_i being the data at a specific location in the image, where i indexes the image locations from $i = 1 \dots n$. Our goal is to compute $P(Y \mid X)$, where $Y = object$, or $Y = clutter$. Associated with each image class is a particular model, which is equal either to the foreground model f or background model b. We use θ_f to indicate the parameters of the foreground model and θ_b for the background model. If we

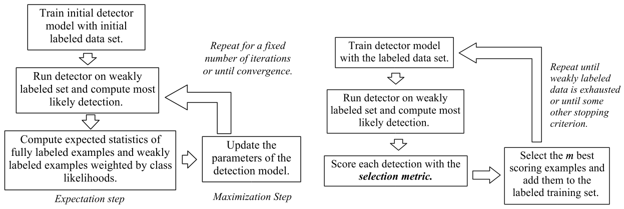

Train initial detector model with initial labeled data set.
Run detector on weakly labeled set and compute most likely detection.
Compute expected statistics of fully labeled examples and weakly labeled examples weighted by class likelihoods.
Expectation step
Update the parameters of the detection model.
Maximization Step
Repeat for a fixed number of iterations or until convergence.
Train detector model with the labeled data set.
Run detector on weakly labeled set and compute most likely detection.
Score each detection with the *selection metric.*
Select the *m* best scoring examples and add them to the labeled training set.
Repeat until weakly labeled data is exhausted or until some other stopping criterion.

Figure 1. **Schematic representation of the batch training approach with EM (left) and the incremental self-training approach (right).**

set $P(Y = object) = P(Y = clutter)$, the likelihood ratio over the entire window is:

$$\frac{P(Y{=}object|X)}{P(Y{=}clutter|X)} = \Pi_{i=1}^{n} \frac{P(x_i|\theta_f)P(M{=}f)}{P(x_i|\theta_b)P(M{=}b)}$$

The value of this likelihood ratio can be thresholded to determine the presence or absence of an object. In practice, the detection is performed in a subwindow of the image that is scanned across all possible locations in the input image.

For such a generic object detector, a natural approach to weakly-labeled training is Expectation-Maximization (EM) [2]. This is a very generic method for generating estimates of model parameters given unknown or missing data. This is implemented as an iterative process which alternates between estimating the expected values of the unknown variables and the maximum likelihood of values of the model parameters (Figure 1(left)).

It would seem that, local maxima and model selection issues not withstanding, EM would be the ideal approach to semi-supervised learning. Indeed, work using EM in the context of text classification [14], [12] has found that EM is a useful approach to training models using weakly labeled data. However, Nigam in [14] also found that there were instances in which EM did not perform well. There are many reasons why EM may not perform well in a particular semi-supervised training context. One reason is that EM solely finds a set of model parameters which maximize the likelihood of the data. The issue is that the fully labeled data may not sufficiently constrain the solution, which means that there may be solutions which maximize the data likelihood but do not optimize classification performance.

There have been a variety of approaches that attempt to incorporate new information into EM and to design alternate algorithms which can utilize additional prior information which we may have about a specific semi-supervised problem [22], [15], [6]. The alternative that we chose to evaluate in this work is often called self-training or incremental training [13]. In self-training, an initial model is constructed by using the fully labeled data. This model is used to estimate labels for the weakly labeled (or unlabeled) data. A selection metric is then used to decide which of the weakly labeled examples were labeled correctly. Those examples are then added to the training set and the process repeats. Obviously the selection metric chosen is crucial here; if incorrect detections are included in the training set then the final answer may be very wrong. This issue is explored throughout the paper. When discussing the incremental addition of training data, it is useful to define the following terms:

- The *initial labeled training set* is the initial set of fully labeled data: $\mathcal{L} = \{L_1 \ldots L_{m_l}\}$
- The *weakly labeled training set* is the current set of weakly labeled data: $\mathcal{W} = \{W_1 \ldots W_{m_w}\}$
- The *current labeled training set* is the initial training set in addition to any weakly labeled examples which have been assigned labels: $\mathcal{T} = \{T_1 \ldots T_{m_t}\}$

We use the object detection framework detailed in the previous sections as the basis of our weakly labeled data approach. This approach begins with an initial set of model parameters trained using the initial labeled training set provided, $\mathcal{M}^0 = \{\theta_f^0, \theta_b^0\}$. This serves as the starting point for our weakly labeled data approach during which we modify foreground model, θ_f.

The weakly labeled data approach relies on being able to estimate where the object is in the training image using the current model. However, since the initial model, $\mathcal{M}^0$, is trained using a limited amount of data, this may not be possible, especially for weakly labeled training data which differs significantly in appearance from the training images. One approach is to immediately add all of the weakly labeled data, $\mathcal{W}$, to the training set, $\mathcal{T}$. However, incorrect labels can potentially "corrupt" the model statistics.

In the approach described here, we attempt to reduce the impact of this issue by labeling weakly labeled examples and adding them incrementally to the training set according to our confidence in those labels, similar to the methods described in [13], [12], [20]. Here, the order in which the images are added is critical to allow the model to first generalize to images which are most similar to the initial training set, and then incrementally extending to views which are quite different from those in the original training set.

A schematic representation of the training procedure, termed "self-training" or "incremental semi-supervised training", is presented in Figure 1(right). The incremental training procedure for using a combination of weakly and fully labeled data is summarized as follows:

Initialization:

1. Train the parameters of the initial model, $\mathcal{M}^0$, consisting of the foreground θ_f^0 and background θ_b^0 models using the fully labeled data subset. Initialize the initial labeled training set, $\mathcal{T}^0$, with the provided fully labeled data.

Beginning of Iteration j:

1. For each W_k in $\mathcal{W}^j$ compute the selection metric, $S_k = \mathrm{Sel}(\mathcal{M}^j, W_k)$.
2. Select the weakly labeled example, $W_{\hat{k}}$ where $\hat{k} = \mathrm{argmax}_k\, S_k$ with the highest score and update both the current training set and the weakly labeled training set, $\mathcal{T}^{j+1} \leftarrow \mathcal{T}^j \cup \{W_{\hat{k}}\}$, $\mathcal{W}^{j+1} \leftarrow \mathcal{W}^j - \{W_{\hat{k}}\}$.

3. Compute a new foreground model θ_f^{j+1} by using $\mathcal{T}^{j+1}$.

End of Iteration j: While $\mathcal{W} \neq \emptyset$

The practical implementation of the semi-supervised approach is not a straightforward application of the techniques described in this section because of the complex nature of real world image data. In this paper, we focus on and provide insight into specific two key issues which are fundamental to practical implementation of the semi-supervised training of object detection systems: 1) What metric should be used in deciding which examples to add to the training set during incremental training? 2) How is the performance of the detector affected by the size of the labeled and weakly labeled sets?

1.3. Previous Work

In recent years there has been a substantial amount of work that addresses the problem of incorporating unlabeled data into the training process, [14], [9], [11]. Some of the earliest work in the context of object detection is described in [1]. The authors used an Expectation-Maximization (EM) approach. More recent work by Selinger in [20] uses an incremental approach similar to our approach. One of the main differences is that a contour based detection model is used. Also the model output is used as the scoring metric to decide which image to add next, whereas we found that other metrics tended to work better. Recent work [8], [7] utilizes an EM based approach.

The work which is most similar to our work is that described in [24]. In this work an object detection system is trained using images which are labeled indicating the presence or the absence of the object of interest. A search approach is used to find likely correspondences between detected features and model features. The work described in this paper is similar, but extends the prior work in that an evaluation of a mix of labeled and weakly labeled data is performed, the problem is examined in a discriminative context and the incorporation of other types of labeled data can be accommodated.

A number of authors have taken the approach of representing the relationships between labeled and unlabeled data using a graph in which the edge weights are inversely related to the similarity between the different examples in feature space [3]. They use a minimum cut algorithm to decide the labeling of the unlabeled data. That method augments the graph of training examples with a pair of “class” nodes which represent the positive and negative classes. Infinite weight edges connect labeled examples to the appropriate “class” nodes. Their analysis showed that particular graph structures and edge weights correspond to optimizing specific learning criteria.

The next set of ideas in this area is based on using random walks through the graph to capture the notion that examples which are similar in feature space should have similar labels. Some of the earliest work in this area is that of Szummer and Jaakkola [21], which was analyzed and extended in [25]. A promising recent direction is that of information regularization by [22] and [5]. The notion is to exactly capture the information that we hope to transfer from the unconditional underlying distribution $P(x)$ to the class label likelihood $P(y \mid x)$. The idea is that the unlabeled data will constrain the final hypothesis in a particular way. Specifically, we want hypotheses that tend not to split high density regions in the underlying distribution.

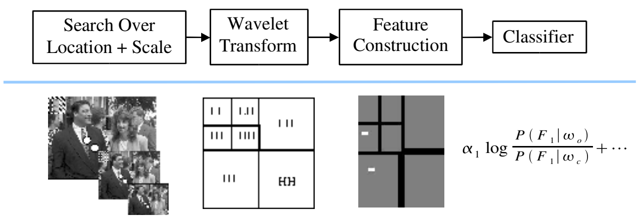

Search Over Location + Scale
Wavelet Transform
Feature Construction
Classifier
$\alpha_1 \log \frac{P(F_1 \mid \omega_o)}{P(F_1 \mid \omega_c)} + \cdots$

Figure 2. **Schematic representation of the detection process for a single stage of the detector.**

2. Experimental Setup

2.1. Detector Overview

We chose the detector described in [18], [19] to conduct our experiments. It has been used successfully for face detection and other rigid objects and has been demonstrated to be one of the most accurate for face detection. The detector is able to capture certain aspects of appearance variation such as intra-class variation. To handle large changes in scale and translation, the detector is scanned over the image at different scales and locations, and each corresponding subwindow is run through the detector.

A schematic representation of the detection process is presented in Figure 2: First, the subwindow is processed for lighting correction, then a two-level wavelet transform is applied, from which features are computed by vector-quantizing groups of wavelet coefficients. Finally, the subwindow is classified by thresholding a linear combination of the log-likelihood ratios of the features. The detector uses a cascade architecture, in which a number of detectors are placed in series; only image patches which are accepted by the first detector are passed on to the next. In this work we only use a single stage of the cascade to simplify the training process. Accordingly, detection performance is lower than what is typically achieved for the detector. We chose to limit the processing to one stage to facilitate many repetitions of the training process in the experiments, which would not have been possible with the full detector because of the long training times.

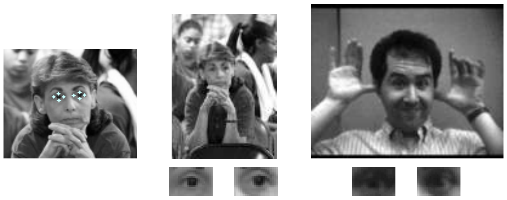


Figure 3. Landmark used on a typical training image (left); sample training images and the training examples associated with them (right).

2.2. Data

The object chosen for these experiments is a human eye as seen in a full or near frontal face. In each fully labeled example, four landmark locations on the eye were labeled (Figure 3). The labeled regions of each training image were rotated to a canonical orientation, scaled and cropped to result in a 24 × 16 training example image.

The full set of images with positive examples consists of 231 images. In each of these images, there were from two to six training examples per image for a total of 480 training examples. The independent test set consisted of 44 images. In each of these images, there were from two to 10 testing examples for a total of 102 test examples. In addition to the object training examples, we used a set of 15,000 negative examples. In all the experiments described in this paper, we used a fixed set of negative examples. Negative examples are assumed to be plentiful [23] and can be collected cheaply.

The training and test images were typically in the range of 200-300 pixels high and 300-400 pixels wide. While the training examples are all normalized to 24 × 16 images, scale invariance is achieved by scaling the image during the detection process. A total of 80 synthetic variations are applied to each training example including: ± 0.5 translation, 0.945 to 1.055 scale, ± 12 degrees rotation.

2.3. Training

Training the model with fully labeled data consists of the following steps:

1. Given the training data landmark locations, geometrically normalize the training example subimages, apply lighting normalization to the subimages, and generate synthetic training examples. The latter consists of scaling, shifting, and rotating the images by small amounts.
2. Compute the wavelet transform of the subimages.
3. Quantize each group of wavelet coefficients and build a naive Bayes model with respect to each group to discriminate between positive and negative examples.
4. Adjust the naive Bayes model using boosting, but maintaining a linear decision function, effectively performing gradient descent on the margin.
5. Compute an ROC curve for the detector using a cross validation set.
6. Choose a threshold for the linear function, based on the final performance desired.

If the full detector cascade is trained, these steps are repeated by setting a threshold that achieves a low false negative rate at each stage. The positive examples at each iteration are those images in the current training set which passed the detection test for the previous iteration. For computational reasons, we limit ourselves to one stage.

The goal of our experiments is to train the detector with different combinations of fully labeled and weakly labeled data and to evaluate the resulting detector performance.

The semi-supervised, incremental version of the training procedure that we used in the experiments reported here can be summarized as follows:

1. Train the detector using a limited amount of fully labeled positive examples and the full set of negative examples.
2. Run the detector over the weakly labeled portion of the data set and find the locations and scales corresponding to maxima of the likelihood ratio.
3. Use the output of the detector to label the unlabeled training examples and assign a selection score to each detection.
4. Select a subset of the newly labeled examples using the selection metric.
5. Iterate and go back to step 1. Stop after a fixed number of iterations or after all of the training images have been added.

Typically, once an image has been added to the training set, it is not removed, and the values of the latent variables are fixed.

2.4. Selection metrics

The type of selection metrics used for selecting the next example to add from the weakly labeled data set in Step 4. above is crucial to the performance of the training. We evaluated the difference in performance between a selection metric based on the classification confidence and a selection metric based on an distance measure between patches that is defined independently from the detector. A key observation is that, because it is independently defined, the second metric will have failure modes that are “orthogonal” to the failure modes of the detector, leading to better performance, as supported by the empirical results below. The effect of using an independently-defined metric had not been previously investigated and it appears to contribute in a critical way to training performance.

The first selection metric, termed *confidence selection metric*, is computed at every iteration by applying the detector trained from the current set of labeled data to the weakly labeled set. The detection with the highest detection confidence are selected and added to the training set. The second selection metric, termed *MSE selection metric* is calculated for each weakly labeled example by evaluating the distance between the corresponding image window and all of the other templates in the training data (including the original labeled examples and the weakly labeled examples added in

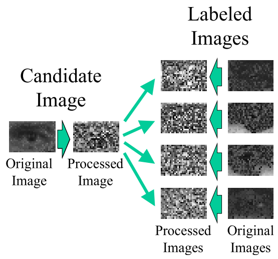


Figure 4. A schematic representation of the computation of the MSE score metric. The candidate image and the labeled images are first normalized with a specific set of processing steps before the MSE based score metric is computed.

prior iterations). The distance is computed after normalization of the detected template for scale, position, and orientation, based on the values computed by the detector, and after 3x3 high-pass filtering and normalization to zero mean and unit variance (Figure 4). Each candidate image is assigned a score which is the minimum of these distances. The candidate images with the smallest scores are selected for addition to the training set. If we define W_i to be the weakly labeled image under consideration, j to be the index over labeled images, $L_j$to be a specific image from the set of labeled images, $g(X)$ to be the transformation performed by the image preprocessing step, and Σ to be the weights for computing the Mahalanobis distance, then the overall computation can be written as:

$$\text{Score}(W_i) = \min_j \text{Mahalanobis}\left(g(W_i), g(L_j), \Sigma\right)$$

It is important to note that, in the computation of the MSE selection metric, the key information that is used is the position and scale returned by the currently detector. As a result, the detector must be accurate in localization but need not be accurate in detection since false detection will be discarded due to large their large MSE distances to all of the training examples. This is crucial to ensure the performance of the training algorithm with small initial training sets. This is also part of the reason for the MSE to outperform the confidence metric, which requires the detector to be accurate in *both* localization and detection performance.

3. Experiments and Analysis

3.1. Experiment Scenarios

We found that there was quite a bit of variance in the final detector performance and in the behavior of the semi-supervised training process. Much of this variance arose from the specific set of images randomly selected in the small initial training subset. To overcome this limitation, each experiment was repeated using a different initial random subset. We call a specific set of experimental conditions an *experiment,* and each repetition of that experiment we call a *run*. In most cases 5 runs were performed for each experiment.

Another parameter of the experiments is the number of images added at each iteration. Ideally, only a single image would be added at each iteration. However, because of the substantial training time of the detector, more than one image was added at each iteration. Adding more images reduces the average training time per weakly labeled image, but increases the chance that there will be an incorrect detection included in the weakly labeled data set. Typically 20 weakly labeled images were added to the training set at each iteration.

One of the challenges in performing such experiments is that the inner loop of the algorithm, training the detector on one specific training set, takes on the order of twelve hours on 3.0 GHz level machines. If the detector is trained during 10 iterations and 5 repetitions of an experiment are performed, then each experiment takes $12 \times 10 \times 5 = 600$ hours of compute time. As a result, the total computation time necessary to investigate all the variations of parameters and training conditions increases rapidly (to approximately 3 CPU-years).[1]

3.2. Evaluation Metrics

Each "run" was evaluated by using the area under the ROC curve (AUC). Because different experimental conditions affect performance, the AUCs were normalized relative to the full data performance of that run. So a reported performance level of 1.0 would mean that the model being evaluated has the same performance as it would if all of the labeled data was utilized. A value of less than 1.0 would mean that the model has a lower performance than that achieved with the full data set. To compute the full data performance, each specific run is trained with the full data set and its performance is recorded. The performance from all of the runs of a specific experiment are aggregated and we compute a single set of performance measures: the mean, the standard deviation, and the 95% significance interval, computed as the mean plus and minus 1.64 times the standard error of the mean. The plots show either or both the standard deviation or the 95% significance interval as error bars.

3.3. Baseline training configuration

It is very important to characterize sensitivity to training set size because we want to perform our experiments under conditions where the addition of weakly labeled data will make a difference. If the performance of the detector is already at its maximum, given a labeled training set of a specific size, then we cannot expect weakly labeled data to

[1] For reasons of space, we present only a summary of the experiments. Detailed analysis of the influence of the number of features, the geometric variations, and other training variations are reported in [16].

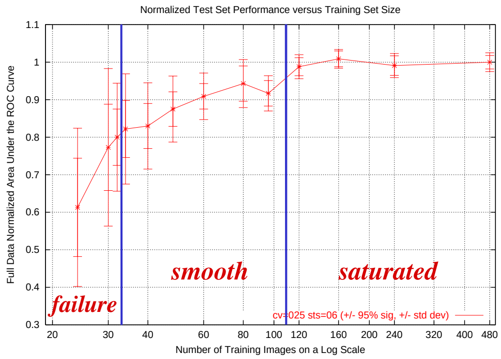


Figure 5. **Normalized AUC performance of the detector plotted against training set size on a log scale with the three regimes of operation labeled. The inner error bars indicate the 95% significance interval, and the outer error bars indicate the standard deviation of the mean.**

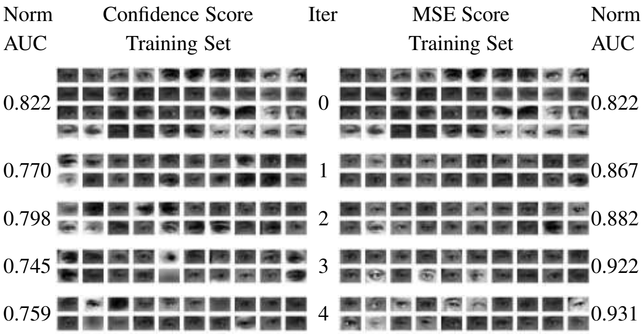


Figure 6. **Comparison of the training images selected at each iteration for the confidence and the MSE selection metrics. The initial training set of 40 images is the same for both metrics and is 1/12 of the initial training set size.**

help. In order to establish a baseline for the typical number of examples needed to train the detector, we ran the detector with different training set sizes and recorded the AUC performance (Figure 5). Our interpretation of this data is that is there are three regimes in which the training process operates. We call the first the "saturated" regime, which in this case appears to be from approximately 160 to 480 training examples. In this regime, 160 examples are sufficient for the detector to learn the requisite parameters; more data does not result in better performance. Similarly, variation in performance is relatively constant and small in this range. We call the second regime the "smooth" regime, which appears in this case to be between 35 and 160 training examples. In this regime, performance decreases and variation increases relatively smoothly as training set size decreases. In the third regime, the "failure" regime, there is both a precipitous drop in performance and a very large increase in performance variation. This third regime occurs when the training algorithm does not have sufficient data to estimate some set of parameters. An extreme case of this would be when the parameter estimation problem is ill conditioned. Based on this set of experiments, we chose the size of the labeled training set to be in the smooth regime for the experiments with weakly-labeled data.

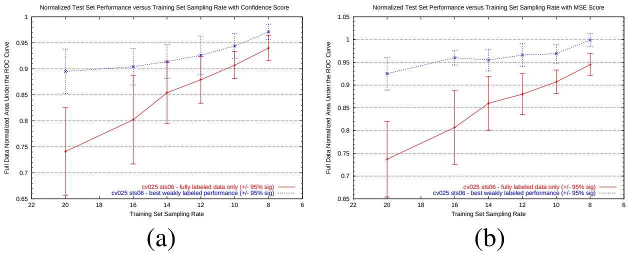
3.4. Selection Metrics

The next question is whether the choice of the selection metric makes a substantial difference in the performance of the semi-supervised training. We conducted experiments to compare the two main options: A confidence metric based on the most natural approach of selecting the example with the highest detector confidence, and the MSE metric that is defined independently of the detector confidence. The overall result is that the detector-independent MSE metric outperforms the more intuitive confidence metric.

The comparison between the two selection metrics is summarized in Figure 7. In these plots, the horizontal axis indicates the frequency at which the training data is sampled in order to select the initial labeled training set for each run ("8" means that 1/8th of the full training data was used as initial labeled training data, while the rest was used as unlabeled data). The plots show that performance is improved by the addition of weakly labeled data over the range of data set sizes. However, the improvements are not significant at the 95% level for the confidence metric. For the MSE metric however, the improvement in performance is significant for all the data set sizes. This observation is supported by other experimental variations in which the MSE metric consistently outperforms the confidence metric. Figure 6 shows montages of the examples selected from the weakly labeled training images selected at each iteration using the confidence metric and the MSE metric for a single run. The performance of the detector trained with the MSE metric improves with each iteration, whereas the performance of the confidence-based one decreases. For the confidence metric, there are clearly incorrect detections included in the training set past the first iteration. In contrast, all of the images that the MSE metric selects are valid except for one outlier at iteration 4.

3.5. Relative size of Fully Labeled Data

It is also important to evaluate the number of weakly labeled exemplars that need to be added to the labeled set in order to reach the best detector performance. For this evaluation, we recorded the number of examples that need to be added to the initial set in order to reach the point at which the performance of the detector does not change appreciably for every training run. The data is summarized in Figure 8, in which we plotted the ratio of weakly labeled data to labeled data at which the training procedure converged,



(a) (b)

Figure 7. Normalized performance of the detector, incorporating weakly labeled data by using the confidence metric (a) or the MSE metric (b), as the fully labeled training set size varies. The bottom plot line is the performance with labeled data only and the top plot line is the performance with the addition of weakly labeled data. Error bars indicate the 95% significance interval of the mean value.

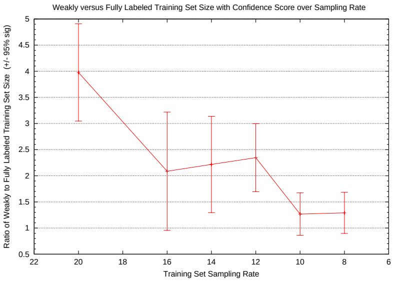


Figure 8. Ratio of weakly labeled to fully labeled data as the fully labeled training set size increases.

against the size of the initial training set (or, more precisely, the sampling rate that was used for generating the initial training set). This data shows that, as expected, the ratio increases as the size of the initial training set decreases since more weakly labeled examples are needed to compensate for smaller training sets. More importantly, the total size of the training set (initial labeled training images + examples added during training) is within the “saturated” operating regime identified in Figure 5. This is important because it shows that, even for small initial training sets, the total number of examples is on the same order as the number that would be needed to train the detector with a single set of labeled examples. In other words, using a small set of labeled examples does not cause us to pay a penalty in terms of a greater size of the total training set.

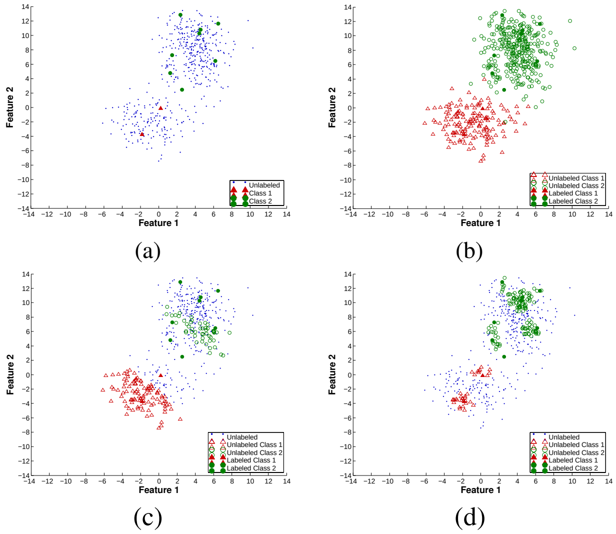
3.6. Discussion

These experiments lead us to several observations that will be useful in developing future detection systems based on weakly-labeled training. First, the results show that it is possible to achieve detection performance that is close to the base performance obtained with the fully labeled data, even when a small fraction of the training data is used in the initial training set. This observation remains valid even when taking into account the high degree of variability in performance across different choices of initial training sets (as illustrated by the error bars in the graphs presented, and the fact that we normalize the detector performance with respect to the base detector trained with all the labeled data). Second, as a practical matter, the experiments show that the self-training approach to semi-supervised training can be applied to an existing detector that was originally designed for supervised training. In fact, in our case, we used a detector that was already highly optimized and we were able to integrate it in the training framework. This suggests a general procedure for using semi-supervised training with existing detectors.

Finally, a more fundamental observation is that the MSE selection metric consistently outperforms the confidence metric. Experiments with simulated data and other, filter-based detectors (from [16], not reported here from reasons of space) show that, more generally, the self-training approach using an independently-defined selection metric outperforms both the same approach with confidence metrics, but also batch EM approaches. These results bring to light an important aspect of the the self-training process which is often overlooked. The issue is that during the training process, the distribution of the labeled data at any particular iteration may not match the actual underlying distribution of the data. As a result, confidence metrics may perform poorly because the labeled data distribution created by this metric is quite different from the underlying distribution, even when all of the weakly labeled data selected by the metric is correctly labeled. To illustrate this observation, Figure 9 shows a simple simulated example in which the labeled and unlabeled examples are drawn from two Gaussian distributions in the plane. Comparing the labels obtained after five iterations by using the confidence metric (Figure9(c)) and the Euclidean metric, we see that the labeled points cluster around existing data points. We believe a closer examination of this issue from both a theoretical and practical standpoint is an important interesting topic for future research toward the effective application of the semi-supervised approaches to object detection problems.

4. Summary and Conclusions

The goal of this work was to explore and evaluate approaches to semi-supervised training using weakly labeled data for appearance-based object detection. We conducted extensive experiments with a state-of-the art detector that led to several important conclusions including a quantitative evaluation of the performance gained by adding weakly labeled data to an initial small set of labeled data; a demonstration of the feasibility of modifying an existing detector



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

Figure 9. **(a) Original unlabeled data and labeled data; (b) Plot of the true labels for the unlabeled data; (c),(d) The points labeled by the incremental self-training algorithm after 5 iterations using the confidence metric and the Euclidean metric, respectively.**

to use weakly labeled data; and insights into the choice of selection metric used for training.

Many important issues that are critical to practical applications of these training ideas remain to be explored. First, it might be important to use a different version of the detector for initial training and for actual use on test images. For example, we found that the position and scale accuracy of the detector are important for semi-supervised training, whereas they may be less important when the detector is used in an application. Second, one alternative explanation for the success of the nearest neighbor approach (based on the appropriate selection metric) is that it is performing a type of co-training [4], [13], [10]. It would be interesting to study the relation between the semi-supervised training approach evaluated here with the co-training approaches. As shown in the experiments, the choice of the initial training set has a large effect on performance. Although we have performed experiments that compare different selections of the initial training set, it would be useful to develop more precise guidelines for selecting it. Finally, the approach could be extended to training examples that are labeled in different ways. For example, some images may be provided with scale information and nothing else. Additional information may be provided such as the rough shape of the object, or a prior distribution over its location in the image.

References

[1] S. Baluja. Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. NIPS, 1998.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. ICML, 2001.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. COLT, 1998.

[5] A. Corduneanu and T. Jaakkola. On information regularization. UAI, 2003.

[6] F. Cozman, I. Cohen, and M. Cirelo. Semi-supervised learning and model search. ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining., 2003.

[7] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. ICCV, 2003.

[8] R. Fergus, O. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. CVPR, 2003.

[9] T. Joachims. Transductive inference for text classification using support vector machines. ICML, 1999.

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

[11] A. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. KDD, 2000.

[12] K. Nigam. *Using Unlabeled Data to Improve Text Classification*. PhD thesis, Carnegie Mellon University Computer Science Dept., 2001. CMU-CS-01-126.

[13] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. CIKM, 2000.

[14] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. AAAI, 1998.

[15] Y. Rachlin. A general algorithmic framework for discovering discriminative and generative structure in data. Master's thesis, ECE Dept. Carnegie Mellon University, 2002.

[16] C. Rosenberg. *Semi-Supervised Training of Models for Appearance-Based Statistical Object Detection Methods*. PhD thesis, Carnegie Mellon University Computer Science Dept., May 2004. CMU-CS-04-150.

[17] B. Schiele and J. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *IJCV*, 36(1):31–52, 2000.

[18] H. Schneiderman. Feature-centric evaluation for efficient cascaded object detection. CVPR, 2004.

[19] H. Schneiderman. Learning a restricted bayesian network for object detection. CVPR, 2004.

[20] A. Selinger. Minimally supervised acquisition of 3d recognition models from cluttered images. CVPR, 2001.

[21] M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. NIPS, 2001.

[22] M. Szummer and T. Jaakkola. Information regularization with partially labeled data. NIPS, 2002.

[23] P. Viola and M. J. Jones. Robust real-time object detection. Technical report, Compaq Cambridge Research Lab, 2001.

[24] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. ECCV, 2000.

[25] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. ICML, 2003.