Training Object Detection Models with Weakly Labeled Data

Charles Rosenberg and Martial Hebert
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
chuck,hebert@cs.cmu.edu

Abstract

Appearance based object detection systems utilizing statistical models to capture real world variations in appearance have been shown to exhibit good detection performance. The parameters of these statistical models are typically learned automatically from labeled training images. This process can be difficult in that a large number of labeled training examples may be needed to accurately model appearance variation. In this work we describe a method whereby a training set consisting of a small number of fully labeled training examples augmented with a set of weakly labeled examples can be used to train a detector which exhibits performance better than that which can be obtained with a reduced set of fully labeled training examples alone.

1 Introduction

In recent years a number of successful appearance based object detection systems have been reported in the literature, [7, 9, 10, 12, 13]. Each of these systems has a different approach to how the appearance of an object and its variations are modeled. Certain types of variations like lighting, translation, and rotation are often modeled explicitly with analytic models. Variations which are not modeled explicitly are typically learned from a set of labeled training images. In order for the detector to achieve good performance this set of images needs to span the space of appearance variation of the object which is not captured by other aspects of the system. Depending on the object, a large set of images may be necessary to adequately span this space. Labeling these training images typically consists either of indicating the location of one or more landmarks on the object of interest or providing a mask which indicates which pixels in the training image belong to the object and which do not. Sometimes both landmarks and a mask are necessary. Collecting and labeling a training set of a sufficient size is a difficult process. When the training images are captured under carefully controlled conditions, for example in front of a uniform background, labels can be generated automatically. However, collecting a large number of images under controlled conditions may be infeasible and those images may not be representative of the appearance variation in the typical operating environment. The alternative is manual labeling which can be time consuming and expensive.

The goal of this work is to reduce the effort necessary to construct the set of images needed to train an appearance based object detection system. We propose to do this by utilizing a training set which consists of a small set of *fully labeled* examples in conjunction with an additional set of *weakly labeled* examples. We define *fully labeled* training examples as images labeled in the traditional sense, where object landmark locations and

pixel class membership is indicated. We define *weakly labeled* training examples as images for which some information has been provided about each example image, but less than what is typical for full labeling. The weak labeling information for a training image could take many forms. For example the exact location of the object, but not its scale or pose could be provided or just the object's pose in the training image, but not its location. In this work weakly labeled data is weakly labeled in the sense that the object of interest is known to be in the training image, but the location of the object and which pixels correspond to the object is not known. Note that *weakly labeled* data is not the same as *unlabeled data* where the only information provided is that the image is a typical image, but no information is provided about the absence or presence of the object of interest.

In this paper we first introduce a simple object detection model which is used as a means of exploring and evaluating the use of weakly labeled training data. We then discuss our approach to the use of weakly labeled training data and related prior work. Finally we present the quantitative results of a set of experiments on real world images and conclude.

2 Object Detection Model and Weakly Labeled Data Approach

2.1 Introduction

To facilitate the evaluation of the use of weakly labeled training data when training an appearance based object detection model we developed a relatively basic statistical object detection framework. It is important to note that our goal was to construct a detection system which provided good detection performance and facilitated our approach to the integration of weakly labeled data in the training process. We did not attempt to construct a state of the art detection system, example of which are presented in [7, 9, 10, 12, 13].

2.2 Basic Model

Our object detection framework is based on a generative image model which captures the statistics of a feature vector computed at each pixel location. The features in our system are the outputs from a set of oriented separable Gaussian derivative filters, the filter design according to [2]. Specifically, in our implementation, we utilized four filter scales and three orientations at each scale, with the following kernel sizes: 15×15, 21×21, 31×31, and 43×43, for a total of 12 filter values at each pixel location. An example of the filters at a single scale can be seen in Figure 1.

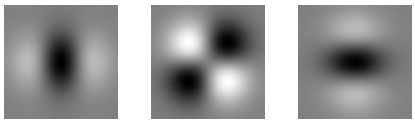


Figure 1: Plots of the three filter orientations used as features for the generative model. Positive values are indicated as light colored pixels, negative values as black, and zero as gray.

In our generative model, the distribution of values over this twelve dimensional vector of filter outputs is modeled using a mixture of full covariance Gaussians. In the experiments detailed here we use one mixture model with 40 components to capture the filter output statistics for each object to be detected and another mixture model with 20 components to model the clutter class. (Empirically we found that more mixture components

were needed to accurately model the object distribution.) Even though each object class is represented with a different mixture model, within each object class the feature vector statistics are modeled with a single mixture of Gaussians irrespective of their relative location on the object of interest. This is similar to the model described in [10], but in that work the authors utilize a histogram based model which typically requires a larger number of parameters to handle high dimensional feature spaces.

The appeal of this type of model is its simplicity, the issue, however, is how to map from filter response probabilities at individual pixel locations to the desired quantity which is the probability of the object of interest being present at a particular location in the image. One approach is to combine the probabilities from nearby pixels into a single quantity, a principled method for doing this is described in the following section.

2.3 Model of the Spatial Distribution of Filter Responses

To move from a pixel based generative model to an object based one, we utilize the generative model which is depicted as a graphical model in Figure 2. Figure 2a depicts the model for an image containing an object and clutter and Figure 2b depicts the model for an image containing solely clutter. Note that this model can be generalized to handle multiple objects.

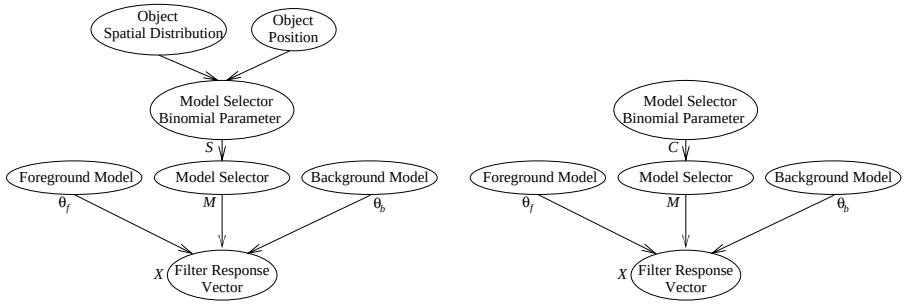
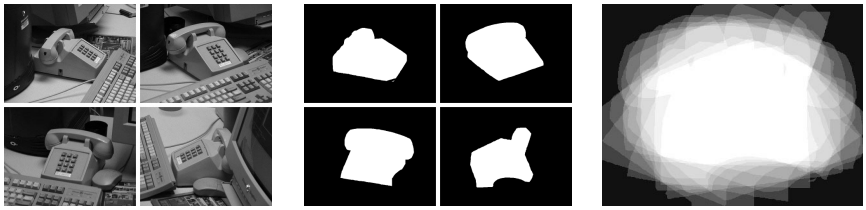


Figure 2: Diagram (a) depicts a graphical model which corresponds to the generative model of the spatial distribution of filter responses and its associated dependencies for an image containing an object. Diagram (b) depicts the generative model for an image which only contains clutter.

The selection of one of the two models in Figure 2 determines whether an image will consist of both the object and clutter or just clutter alone. First we will discuss the model in Figure 2a which depicts an image containing an object. It is first important to note the top down nature of the model. The top two nodes determine where the object will be in the image and the distribution of the features generated in the image. The *object spatial distribution* determines the probability that a feature vector will be generated from the *foreground model* distribution or the *background model* distribution at each location in the image. It is important to note the difference between these distributions. The foreground and background models are the location independent distributions described in the previous section. The foreground model captures the distribution of the feature vectors which are only generated by the object and the background model captures the distribution of feature vectors which are only generated by the background. As described previously, the foreground and background models are each composed of a separate mixture of Gaussians. The purpose of the *spatial distribution* is to capture the approximate shape and size of the object. The object spatial distribution specifies the parameters of a probability distribution at each location which selects whether a particular feature at a specific location

in the image is generated by the foreground distribution or the background distribution. In our implementation the *spatial distribution* consists of a collection of parameters for a set of Bernoulli distributions and each feature vector is generated independently of the others. Therefore each Bernoulli distribution has a different set of parameters at each image location. Examples of such a model for multiple poses of a desktop telephone can be seen in Figure 3c where lighter pixels indicate locations are more likely to be foreground and darker pixels indicate locations which are more likely to be background. For pixels near the center of an object the foreground probability will be close to 1, as can be seen in Figure 3c. In our model pixels outside the bounding box of the object are all modeled by the clutter class Bernoulli distribution.

The case where an image consists solely of clutter, as depicted in Figure 2b, is simpler. In this case all of the pixels in the image are simply generated by a fixed mixture of the *background model* and the *foreground model*. The foreground model is included to capture incorrect labeling of foreground pixels as background.



(a) (b) (c)

Figure 3: Four example training images (a) and corresponding mask images (b). Image (c) is a plot of the spatial distribution of the object viewed from a wide range of different angles. Lighter pixels indicate pixels which are more likely to be part of the phone, darker pixels indicates pixels which are less likely to be part of the phone.

In the experiments reported here the objects appear at approximately the same scale in both the training and test images. However, scale could be handled in a manner similar to that described in [13] as a post process. It should also be noted that we currently assume that the probability of the object being at any particular location in the image is uniform.

To make things more formal, 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 $i = 1 \ldots n$. Our goal is to compute $P(Y \mid X)$, where Y is the image class, either an image containing an object, $Y = object$, or not, $Y = clutter$. For the model that captures the presence of an object, we designate the spatial distribution over model selection probabilities when the object is present as S, the object distribution. The parameters of the Bernoulli distribution at a specific image location is indicated S_i, where i indexes image location. So the probability that a particular model is selected at a specific location is $P(M_i = m_i) = S_i$. In the case of an image which solely contains clutter, C is used to signify the parameters of a fixed Bernoulli distribution over the image. We use θ_f to indicate the parameters of the mixture of Gaussians for the foreground model and θ_b the background model. Note that the parameters of these models do not vary with image location, as described in the previous section. By applying Bayes rule we can compute $P(Y \mid X)$ in terms of quantities from our generative model:

$$P(Y \mid X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Taking $P(Y)$ as the prior, if we can find an expression to compute $P(X \mid Y)$ and $P(X)$ we can compute $P(Y \mid X)$. When computing $P(X \mid Y = object)$ we note that given $Y = object$, the probability of the observed feature data at each image location is

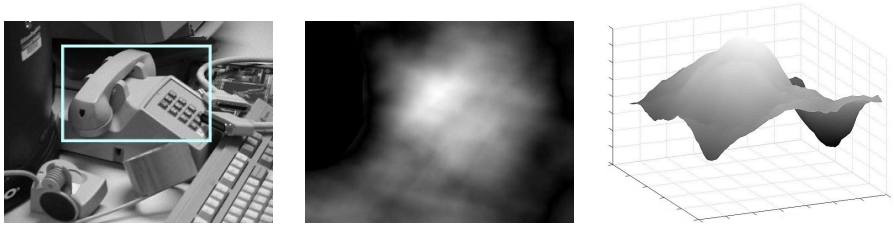
independent:

$$P(X \mid Y = object) = \Pi_{i=1}^{n}(P(x_i \mid \theta_f)P(M_i = \theta_f \mid S_i) + P(x_i \mid \theta_b)P(M_i = \theta_b \mid S_i))$$

For images which contain both the object and clutter pixels outside the bounding box are modeled identically to pixels generated in an image which consists solely of clutter. Because of this, only pixels inside the bounding box need to be considered when computing the likelihood ratio. If we set $P(Y = object) = P(Y = clutter)$, the likelihood ratio over the entire image is:

$$\frac{P(Y=object|X)}{P(Y=clutter|X)} = \Pi_{i=1}^{n} \frac{P(x_i|\theta_f)P(M_i=\theta_f|S_i)+P(x_i|\theta_b)P(M_i=\theta_b|S_i)}{P(x_i|\theta_f)P(M_i=\theta_f|C)+P(x_i|\theta_b)P(M_i=\theta_b|C)}$$

The maximum likelihood location of the object can be found by varying the location of object spatial distribution in the image and finding the location with the maximum likelihood ratio. The value of this likelihood ratio can also be thresholded to determine the presence or absence of an object. Figure 4 shows an example detection in (a), a grayscale plot of the log likelihood ratio in (b), and a surface plot of the same data in (c).



(a) (b) (c)

Figure 4: These images are examples of a detections of the desktop phone object on the test set. The (a) image is the detection, the (b) image is a grayscale plot of the log likelihood ratio where larger values are plotted with lighter pixels, the (c) image is a 3d plot of the same data.

2.4 Training the Model with Labeled Data

The parameters of the probability distributions in this model can be learned from training data. When the training data is labeled this can be done in a relatively straightforward manner because of the structure of the generative model as depicted in Figure 2. The structure indicates that the parameters of S_i can be computed simply by knowing which pixels are part of the object and which are background in the training examples. So, given an aligned mask for each training image which indicates the class membership of each pixel, it is possible to compute the parameters of the Bernoulli distribution, for example see Figure 3, for each location simply by counting the number of times that location was indicated as belonging to the object class.

Also, because of the structure of the model, once the class of a pixel is known the probability distributions $P(x_i \mid \theta_f)$ and $P(x_i \mid \theta_b)$ can be computed. In our implementation, as described previously, each of these distributions is modeled by a mixture of Gaussians. In our case each of these distributions does not vary with image location, so it is only necessary to learn a single set of parameters for each. We learn the parameters of these Gaussians in the standard manner using Expectation-Maximization, as described in [1].

Some example detection results achieved when training this model with labeled data can be seen in Figure 5.

Figure 5: Examples of a detections of the mug, phone, and chair objects using a detector trained with fully labeled data.

2.5 Weakly Labeled Data Approach

In our approach weakly labeled data is weakly labeled in the sense that the object of interest is known to be in the training image, but the location of the object and which pixels correspond to the object is not known. For the work described here we assume that the objects are all present at approximately the same scale.

We utilize the object detection framework detailed in the previous section and train it using EM. The first step in our training process is to train the spatial distributions and the foreground and background models as described in the previous section with the fully labeled data subset. This serves as the starting point for our weakly labeled data approach. During training with weakly labeled data we fix the background model and the spatial distributions.

We then train the foreground model with the combination of fully labeled and weakly labeled images using EM. The main difference from the fully labeled data case is that we weight the contribution of each training example to the sufficient statistics in the expectation step according to our confidence in that data. The weight for fully labeled data is 1. For weakly labeled examples the weight is the probability that the observed feature vector was generated by the foreground model given the current model.

To compute this probability we first use the current model to determine the most likely location of the object in each weakly labeled data example. Once the most likely location of the object in each weakly labeled examples is known the desired probability can be computed as follows:

$$P(M_i = \theta_f \mid x_i, S_i) = \frac{P(x_i|\theta_f)P(M_i=\theta_f|S_i)}{P(x_i|\theta_f)P(M_i=\theta_f|S_i)+P(x_i|\theta_b)P(M_i=\theta_b|S_i)}$$

The complete training procedure for using a combination of weakly and fully labeled data is as follows:

1. Train the object and clutter spatial distributions and the foreground and background models using fully labeled data subset and EM.
2. Compute the most likely object location for each weakly labeled data example using the current model.
3. Execute the "E" step of EM, by computing the expected values of the sufficient statistics of the fully labeled and weakly labeled object data, weighting the weakly labeled examples by $P(M_i = \theta_f \mid x_i, S_i)$.
4. Execute the "M" step of EM, by updating the parameters of the foreground model using the sufficient statistics computed in step 3.
5. Repeat steps 2-4 for a fixed number of iterations or until convergence.

In this work we repeat the inner loop of this algorithm for a fixed number of iterations, typically 40. Also filter responses outside of the bounding box of the object spatial distribution were not included in the weakly labeled training set for the foreground model.

2.6 Implementation Efficiency Issues

Conceptually detecting objects using the generative model described is relatively straightforward, the likelihood ratio at each possible x, y location in the image is computed and is then thresholded to find detections or the most likely location can be determined using the maximum of the likelihood ratio. However, computational efficiency can be an issue because these spatial distributions can be quite large and therefore can be very costly to compute. For example the spatial distribution for the phone object for a 640×480 image is 405 pixels wide $\times$ 314 pixels high and encompasses over 120,000 pixels. To speed up this computation, which is basically a large correlation, we introduce an approximation which facilitates an incremental approach to computing the desired likelihood ratio. It should also be noted that there are other signal processing methods for speeding up this computation, such as frequency domain processing, for example as described in [4].

The approximation we use here is to represent the probabilities in the spatial distribution using only a small number of discrete values, to quantize the probability values. In our experiments we used only two values, a background rate and a foreground rate. In the quantized approximation, a bitmap indicates which pixels belong to a specific quantization level. Because of the nature of this approximation a faster incremental computation can be implemented which takes advantage of prior computations. Specifically, as the bitmap is moved across the image, not all of the values enter or leave the sum or product being computed, only a small number of values change. Taking advantage of this can greatly reduce the number of pixels which need to be processed.

In our experiments when a two level quantized approximation was combined with the incremental approach a speed up of 45 times over the naive approach was observed, reducing the run time per 640×480 image from 90 minutes to under 2 minutes.

3 Related Work

The work that is most closely related to the work described here is that by Frey and Jojic as described in [3] and [6]. In that work a generative model of image formation is assumed where each pixel intensity value in a "latent image" is modeled by a separate Gaussian with a small noise variance. The observed image is formed by applying a transformation, such as translation to the latent image. Our work differs in two primary ways. The first is that our method is semi-supervised. This supervision is provided via a small set of labeled training images and an explicit clutter model. This guides the algorithm to model the object of interest, so no training image alignment is needed. The second way our work differs is in the details of our generative model. Instead of modeling pixel intensities directly, we model a vector of filter responses. The difference is that our goal is to build a generative model which is useful for the detection task whereas their goal is to build a generative model which generates a realistic looking image. So even though our model is incapable of generating realistic looking images, it is more robust to small variations in the input when used for discriminative purposes.

Similar work to Frey and Jojic is described in [14], but differs from our work in its feature set and generative model. In this work the generative model captures the statistics of wavelet coefficients and the experiments involved compensating for translation, rotation and lighting effects for unlabeled data. The work described in [16] uses an unsupervised framework and is also similar, but again utilizes a different feature set and generative model. In this work models were constructed for a set of weakly labeled object and clutter images. Experiments were performed training a face and a car detector and good results were reported. The work described in [11] is similar, but differs from our approach in that it utilizes hard clustering for training and classification purposes. The

work described in [15] is also quite similar to our work in that it seeds the model with a small amount of labeled data. However their object model is different in that it is based on deformable templates of object contours. Also, in their method weakly labeled training examples are added to the training set one by one, whereas in our method all new training examples are included at once.

Weakly labeled data in conjunction with Expectation-Maximization has also been discussed in [5]. Although this work appears quite similar on the surface it is different in two important ways. The first is that it is targeted toward solving reinforcement learning problems. The second is that the approach requires an oracle which can be repeatedly queried for a weak learning signal for unlabeled data during training.

Work has also been reported regarding the use of unlabeled data for training models for content based document classification by Nigam, et al. [8].

4 Experimental Results

4.1 Description

In our experiments our goal was to detect a desktop telephone. There was significant variation in the appearance of the phone as seen in Figure 6, however, lighting and scale remained relatively constant. A set of twelve training images was used which included significant clutter and the viewing position of the phone was varied ± 90 degrees of frontal. The test set consisted of a different set of twelve images taken under similar conditions but which also contained minor partial occlusion of the phone object. Test set lighting and scale conditions were similar to the training set.

In our experiments we evaluated the performance of models trained with different subsets of the fully labeled training set both with and without the addition of weakly labeled data. The "Full" set included all twelve images as fully labeled data and is our basis for comparison. Set "A" included 6 images evenly sampled with respect to viewing angle, so the full range of viewing angles was covered but were not as densely sampled as the full training set. Set "B" included 3 images evenly sampled with respect to viewing angle. Set "C" included 6 images in a contiguous range of viewing angles, starting at 90 degrees and going until approximately frontal. Set "D" included the 3 images closest to the 90 degree viewing angle.

To evaluate the performance of our algorithm we computed the distance in pixels from the location of the true detection to the maximum likelihood location found by the algorithm. The location of the ground truth detection was approximately at the center of the object and was determined by the highest likelihood match between a mask image indicating which pixels belonged to the object and the learned spatial distribution. In these experiments the C parameter for clutter images, the probability of a pixel being generated from the background model, was set to be 0.99.

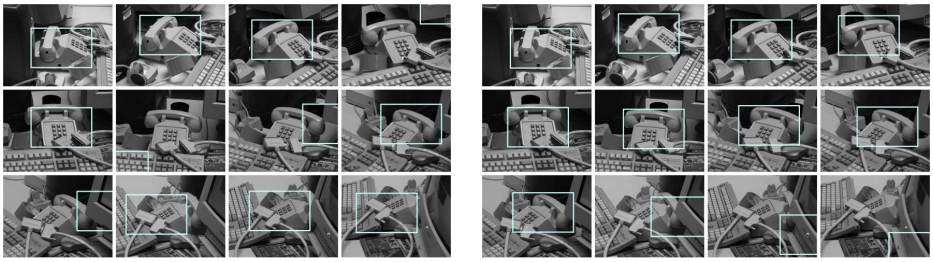
4.2 Results

Typical detections results on the test set can be seen in Figure 6. For visualization purposes we assume that the object is present and plot a fixed size turquoise rectangle centered at the location of the maximum likelihood detection. The images in Figure 6a show the result on the test set for a model trained with training set "D" alone. The results in Figure 6b are when weakly labeled data is added to the training set. The first three images in the test set are very close in viewing angle to the training images in set "D". The important thing to note is that the weakly labeled data allowed the model to generalize to nearby viewing angles.

The quantitative results of our experiments are in detailed Table 1. Mean error is the average distance in pixels from the true object location to the detected location under that experimental condition and "Std Err" is the standard error of that mean. As can be seen an improvement was realized with the addition of weakly labeled data to the training set in all cases except for Set A. It seemed that in that condition the model had achieved maximum performance with the labeled data alone. The largest improvement in performance was realized for set D. As can be seen in Figure 6b, the system effectively generalized from the limited set of views in the fully labeled training data, similar to the first three views in the figure to the larger set of views in the weakly labeled data. This is evidenced by the improved detection performance in the first two rows of images in the figure.

Data	Description	Mean Err	Std Err	Description	Mean Err	Std Err
A	6 Full Only	37.13	6.51	6 Full + 6 Weak	71.35	20.09
B	3 Full Only	126.53	45.68	3 Full + 9 Weak	89.87	30.19
C	6 Full Only	106.34	37.68	6 Full + 6 Weak	86.86	30.52
D	3 Full Only	146.71	40.65	3 Full + 9 Weak	86.31	30.06
Full	12 Full Only	38.20	7.07			

Table 1: Results of training with weakly labeled data. The column labeled "Mean Err" is the mean distance in pixels from the correct object location. The column labeled "Std Err" is the standard error of this mean, smaller values signifying higher confidence.



(a) (b)

Figure 6: Detection results. Image set (a) are results for a detector trained on data set "D" alone and (b) are results for a detector trained on results data set "D" and weakly labeled data.

5 Conclusions

The appearance based approach to object detection has been proven quite successful as reported recently in the literature. The strength of this approach is that it directly models the expected variation in the appearance of an object which is typically learned from labeled training images. The issue is that a large amount of labeled data may be necessary to capture this appearance variation and acquiring and labeling this data can be difficult and costly. The goal of this work was to demonstrate that weakly labeled, and thus less costly, data can be used in the training process to improve system performance. To test this hypothesis we evaluated the performance of an object detection system based on a generative model. In the majority of our experiments an improvement in performance was observed when weakly labeled data was added to the training set.

There are a number of issues we would like to investigate in the future: weak labeling which just indicates object position, weakly labeled data with scale variation, and an analysis of when weakly labeled data may hurt performance instead of helping it.

Acknowledgments

We would like to thank Sanjiv Kumar for collecting the images used in this work and Tom Minka, Henry Schneiderman and Sebastian Thrun for useful discussions and comments related to this work. This research was supported in part by NSF Grant IIS-9907142.

6 References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition.* New York: Oxford University Press, 1995.

[2] William T. Freeman and Edward H. Adelson. The Design and Use of Steerable Filters. *Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891-906, September 1991.

[3] Brendan Frey and Nebojsa Jojic. Estimating Mixture Models of Images and Inferring Spatial Transformations Using the EM algorithm. *Conference on Computer Vision and Pattern Recognition*, pp. 416-422, 1999.

[4] Brendan J. Frey and Nebojsa Jojic. Fast, large-scale transformation-invariant clustering. *Conference on Neural Information Processing Systems*, 2001.

[5] Yuri Ivanov, Bruce Blumberg and Alex Pentland. Expectation Maximization for Weakly Labeled Data. *International Conference on Machine Learning*, 2001.

[6] Nebojsa Jojic, Brendan Frey. Learning Flexible Sprites in Video Layers. *Conference on Computer Vision and Pattern Recognition*, 2001.

[7] B. Moghaddam and A. Pentland. Probabilistic Visual Learning for Object Representation. *Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[8] Kamal Nigam, Andrew McCallum, Sebastian Thrun and Tom Mitchell. Learning to Classify Text from Labeled and Unlabeled Documents. *Fifteenth National Conference on Artificial Intelligence*, pp. 792-799. 1998.

[9] H. A. Rowley, S. Baluja and T. Kanade. Neural network-based faced detection. *Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23-28, 1998.

[10] Bernt Schiele and James L. Crowley. Recognition Without Correspondence using Multidimensional Receptive Field Histograms. *International Journal of Computer Vision*, vol. 36, no. 1, pp. 31-52, 2000.

[11] Cordelia Schmid. Constructing models for content-based image retrieval. *Conference on Computer Vision and Pattern Recognition*, 2001.

[12] H. Schneiderman and T. Kanade. Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition. *Conference on Computer Vision and Pattern Recognition*, pp. 45-51. 1998. Santa Barbara, CA.

[13] H. Schneiderman and T. Kanade. A Statistical Method for 3D Object Detection Applied to Faces and Cars. *Conference on Computer Vision and Pattern Recognition*, 2000.

[14] C. Scott and R. Nowak. Template Learning from Atomic Representations: A Wavelet Based Approach to Pattern Analysis. *International Conference on Computer Vision Workshop on Statistical and Computational Theories of Vision* , 2001.

[15] Andrea Selinger and Randal C. Nelson. Minimally Supervised Acquisition of 3D Recognition Models from Cluttered Images. *Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 213-220, 2001.

[16] M. Weber, M. Welling and P. Perona. Unsupervised Learning of Models for Recognition. *Sixth European Conference on Computer Vision*, 2000.