Mammographic mass detection with a hierarchical image probability (HIP) model

Clay Spence, Lucas Parra, and Paul Sajda

Sarnoff Corporation CN5300 Princeton, NJ 08543-5300

ABSTRACT

We formulate a model for probability distributions on image spaces. We show that any distribution of images can be factored exactly into conditional distributions of feature vectors at one resolution (pyramid level) conditioned on the image information at lower resolutions. We would like to factor this over positions in the pyramid levels to make it tractable, but such factoring may miss long-range dependencies. To fix this, we introduce hidden class labels at each pixel in the pyramid. The result is a hierarchical mixture of conditional probabilities, similar to a hidden Markov model on a tree. The model parameters can be found with maximum likelihood estimation using the EM algorithm. We have obtained encouraging preliminary results on the problems of detecting masses in mammograms.

Keywords: Mammography, CAD, Image Probability

1. INTRODUCTION

Many approaches to object recognition in images estimate $\Pr(\text{class} \,|\, \text{image})$. By contrast, a model of the probability distribution of images, $\Pr(\text{image})$, has many attractive features. We could use this for object recognition in the usual way by training a distribution for each object class and using Bayes' rule to get $\Pr(\text{class} \,|\, \text{image}) = \Pr(\text{image} \,|\, \text{class}) \Pr(\text{class}) / \Pr(\text{image})$. Clearly there are many other benefits of having a model of the distribution of images, since any kind of data analysis task can be approached using knowledge of the distribution of the data. For classification we could attempt to detect unusual examples and reject them, rather than trusting the classifier's output. We could also compress, interpolate, suppress noise, extend resolution, fuse multiple images, etc.

Many image analysis algorithms use probability concepts, but few treat the distribution of images. One of the few examples of image distribution models was constructed by Zhu, Wu and Mumford.[1] They compute the maximum entropy distribution given a set of statistics for some features, which seems to work well for textures but it is not clear how well it will model the appearance of more structured objects.

There are several algorithms for modeling the distributions of features extracted from the image, instead of the image itself. The Markov Random Field (*MRF*) models are an example of this line of development; see, e.g., References 2,3. However, they tend to be very computationally expensive.

In De Bonet and Viola's flexible histogram approach,[4,5] features are extracted at multiple image scales, and the resulting feature vectors are treated as a set of independent samples drawn from a distribution. The distribution of feature vectors is then modeled using Parzen windows. This has given good results, but the feature vectors from neighboring pixels are treated as independent when in fact they share exactly the same components from lower-resolutions. To fix this one might build a model in which the features at one pixel of one pyramid level condition the features at each of several child pixels at the next higher-resolution pyramid level. The multiscale stochastic process (*MSP*) methods do exactly that. Luettgen and Willsky,[6] for example, applied a scale-space auto-regression (AR) model to texture discrimination. They use a quadtree or quadtree-like organization of the pixels in an image pyramid, and model the features in the pyramid as a stochastic process from coarse-to-fine levels along the tree. The variables in the process are hidden, and the observations are sums of these hidden variables plus noise. The Gaussian distributions are a limitation of MSP models. The result is also a model of the probability of the observations on the tree, not of the image.

All of these methods seem well-suited for modeling texture, but it is unclear how one might build models to capture the appearance of more structured objects. We will argue below that the presence of objects in images can make local conditioning like that of the flexible histogram and MSP approaches inappropriate. In the following we

E-mail: {cspence, lparra, psajda}@sarnoff.com

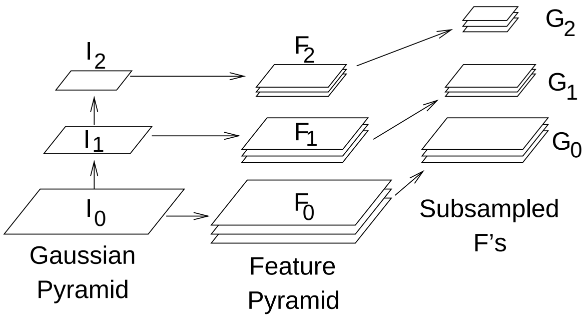

Figure 1. Pyramids and feature notation.

present a model for probability distributions of images, in which we try to move beyond texture modeling. This hierarchical image probability (*HIP*) model is similar to a hidden Markov model on a tree, and can be learned with the EM algorithm. In preliminary tests of the model on classification tasks the performance was comparable to that of other algorithms.

2. COARSE-TO-FINE FACTORING OF IMAGE DISTRIBUTIONS

Our goal will be to write the image distribution in a form similar to $\Pr(I) \sim \Pr(\mathbf{F}_0 \,|\, \mathbf{F}_1) \Pr(\mathbf{F}_1 \,|\, \mathbf{F}_2) \ldots$, where $\mathbf{F}_l$ is the set of feature images at pyramid level l. We expect that the short-range dependencies can be captured by the model's distribution of individual feature vectors, while the long-range dependencies can be captured somehow at low resolution. The large-scale structures affect finer scales by the conditioning.

In fact we can prove that a coarse-to-fine factoring like this is correct. From an image I we build a Gaussian pyramid (repeatedly blur-and-subsample, with a Gaussian filter). Call the l-th level I_l, e.g., the original image is I_0 (Figure 1). From each Gaussian level I_l we extract some set of feature images $\mathbf{F}_l$. Sub-sample these to get feature images $\mathbf{G}_l$. Note that the images in $\mathbf{G}_l$ have the same dimensions as I_{l+1}. We denote by $\tilde{\mathbf{G}}_l$ the set of images containing I_{l+1} and the images in $\mathbf{G}_l$. We further denote the mapping from I_l to $\tilde{\mathbf{G}}_l$ by $\tilde{\mathcal{G}}_l$.

Suppose now that $\tilde{\mathcal{G}}_0 : I_0 \mapsto \tilde{\mathbf{G}}_0$ is invertible. Then we can think of $\tilde{\mathcal{G}}_0$ as a change of variables. If we have a distribution on a space, its expressions in two different coordinate systems are related by multiplying by the Jacobian. In this case we get $\Pr(I_0) = |\tilde{\mathcal{G}}_0| \Pr(\tilde{\mathbf{G}}_0)$. Since $\tilde{\mathbf{G}}_0 = (\mathbf{G}_0, I_1)$, we can factor $\Pr(\tilde{\mathbf{G}}_0)$ to get $\Pr(I_0) = |\tilde{\mathcal{G}}_0| \Pr(\mathbf{G}_0 \,|\, I_1) \Pr(I_1)$. If $\tilde{\mathcal{G}}_l$ is invertible for all $l \in \{0, \ldots, L-1\}$ then we can simply repeat this change of variable and factoring procedure to get

$$\Pr(I) = \left[\prod_{l=0}^{L-1} |\tilde{\mathcal{G}}_l| \Pr(\mathbf{G}_l \,|\, I_{l+1}) \right] \Pr(I_L) \tag{1}$$

This is a very general result, valid for all $\Pr(I)$, no doubt with some rather mild restrictions to make the change of variables valid. The restriction that $\tilde{\mathcal{G}}_l$ be invertible is strong, but many such feature sets are known to exist, e.g., most wavelet transforms on images.

3. THE NEED FOR HIDDEN VARIABLES

For the sake of tractability we want to factor $\Pr(\mathbf{G}_l \,|\, I_{l+1})$ over positions, something like

$$\Pr(I) \sim \prod_l \prod_{x \in I_{l+1}} \Pr\big(\mathbf{g}_l(x) \,|\, \mathbf{f}_{l+1}(x)\big)$$

where $\mathbf{g}_l(x)$ and $\mathbf{f}_{l+1}(x)$ are the feature vectors at position x. The dependence of $\mathbf{g}_l$ on $\mathbf{f}_{l+1}$ expresses the persistence of image structures across scale, e.g., an edge is usually detectable as such in several neighboring pyramid levels. The flexible histogram and MSP methods share this structure.

While it may be plausible that $\mathbf{f}_{l+1}(x)$ has a strong influence on $\mathbf{g}_l(x)$, a model distribution with this factorization and conditioning cannot capture some properties of real images. Objects in the world cause correlations and non-local dependencies in images. For example, the presence of a particular object might cause a certain kind of texture to be visible at level l. Usually local features $\mathbf{f}_{l+1}$ by themselves will not contain enough information to infer the object's presence, but the entire image I_{l+1} at that layer might. Thus $\mathbf{g}_l(x)$ is influenced by more of I_{l+1} than the local feature vector.

Similarly, objects create long-range dependencies. For example, an object class might result in a kind of texture across a large area of the image. If an object of this class is always present, the distribution may factor, but if such objects aren't always present and can't be inferred from lower-resolution information, the presence of the texture at one location affects the probability of its presence elsewhere.

We introduce hidden variables to represent the non-local information that is not captured by local features. They should also constrain the variability of features at the next finer scale. Denoting them collectively by A, we assume that conditioning on A allows the distributions over feature vectors to factor. In general, the distribution over images becomes

$$\Pr(I) \propto \sum_{A} \left\{ \prod_{l=0}^{L} \prod_{x \in I_{l+1}} \Pr\big(\mathbf{g}_l(x) \,|\, \mathbf{f}_{l+1}(x), A\big) \Pr(A \,|\, I_{L+1}) \right\} \Pr(I_{L+1}). \tag{2}$$

As written this is absolutely general, so we need to be more specific. In particular we would like to preserve the conditioning of higher-resolution information on coarser-resolution information, and the ability to factor over positions.

As a first model we have chosen the following structure for our HIP model:[*]

$$\Pr(I) \propto \sum_{A_0,\ldots,A_L} \prod_{l=0}^{L} \prod_{x \in I_{l+1}} \Big[\Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l, x) \Pr(a_l \,|\, a_{l+1}, x) \Big] \tag{3}$$

To each position x at each level l we attach a hidden discrete index or label $a_l(x)$. The resulting label image A_l for level l has the same dimensions as the images in $\tilde{\mathbf{G}}_l$.

Since $a_l(x)$ codes non-local information we can think of the labels A_l as a segmentation or classification at the l-th pyramid level. By conditioning $a_l(x)$ on $a_{l+1}(x)$, we mean that $a_l(x)$ is conditioned on a_{l+1} at the *parent* pixel of x. This parent-child relationship follows from the sub-sampling operation. For example, if we sub-sample by two in each direction to get $\mathbf{G}_l$ from $\mathbf{F}_l$, we condition the variable a_l at (x, y) in level l on a_{l+1} at location $(\lfloor x/2 \rfloor, \lfloor y/2 \rfloor)$ in level $l+1$ (Figure 2). This gives the dependency graph of the hidden variables a tree structure. Such a probabilistic tree of discrete variables is sometimes referred to as a belief network. By conditioning child labels on their parents information propagates though the layers to other areas of the image while accumulating information along the way.

For the sake of simplicity we've chosen $\Pr(\mathbf{g}_l \,|\, \mathbf{f}_{l+1}, a_l)$ to be normal with mean $\bar{\mathbf{g}}_{l,a_l} + M_{a_l}\mathbf{f}_{l+1}$ and covariance Σ_{a_l}, that is,

$$\Pr(\mathbf{g} \,|\, \mathbf{f}, a) = \mathcal{N}(\mathbf{g}, M_a \mathbf{f} + \bar{\mathbf{g}}_a, \Lambda_a) \tag{4}$$

4. EM ALGORITHM

Due to the tree structure, the belief network for the hidden variables is relatively easy to train with an EM algorithm. The expectation step (summing over a_l's) can be performed directly. If we had chosen a more densely-connected structure with each child having several parents, we would need either an approximate algorithm or Monte Carlo techniques. The expectation is weighted by the probability of a label or a parent-child pair of labels given the image. This can be computed in a fine-to-coarse-to-fine procedure, i.e. working from leaves to the root and then back out to the leaves. The method is based on belief propagation.[7]

[*] In principle there is also a factor of $\Pr(I_{L+1})$. In many cases I_{L+1} will be a single pixel that is approximately the mean brightness in the image. We ignore this, which is equivalent to assuming that $\Pr(I_{L+1})$ is flat over some range. In this case $\mathbf{f}_{L+1}$ is zero for typical features. In addition, there is no hidden variable a_{L+1}. If we combine these considerations we see that the $l = L$ factor should be read as $\prod_x \Pr(\mathbf{g}_L \,|\, a_L, x) \Pr(a_L, x)$.

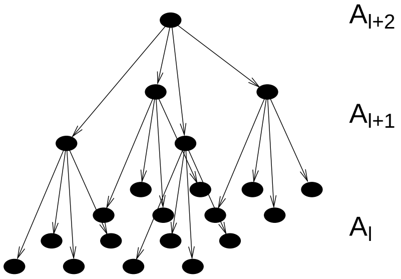

Figure 2. Tree structure of the conditional dependency between hidden variables in the HIP model. With subsampling by two, this is sometimes called a quadtree structure.

Once we can compute the expectations, the normal distribution makes the M-step tractable; we simply compute the updated $\bar{\mathbf{g}}_{a_l}$, Σ_{a_l}, M_{a_l}, and $\Pr(a_l \,|\, a_{l+1})$ as combinations of various expectation values.

In order to apply the EM algorithm, we need to choose a parameterization for the model. The parameterization of $\Pr(\mathbf{g} \,|\, \mathbf{f}, a)$ is given above in Equation 4. For $\Pr(a_l \,|\, a_{l+1})$ we use the parameterization

$$\Pr(a_l \mid a_{l+1}) = \frac{\pi_{a_l,a_{l+1}}}{\sum_{a_l} \pi_{a_l,a_{l+1}}} \tag{5}$$

in order to ensure proper normalization.

Below, we denote the new parameter values computed during the t-th maximization step as θ^{t+1} and the old values as θ^t.

4.1. MAXIMIZATION

Maximizing the expectation of the likelihood over the hidden variables with respect to the model parameters gives the following update formulae:

$$\pi^{t+1}_{a_l,a_{l+1}} = \sum_x \Pr(a_l, a_{l+1}, x | I, \theta^t), \tag{6}$$

$$M^{t+1}_{a_l} = \left(\left\langle \mathbf{g}_l \mathbf{f}^T_{l+1} \right\rangle_{t,a_l} - \left\langle \mathbf{g}_l \right\rangle_{t,a_l} \left\langle \mathbf{f}^T_{l+1} \right\rangle_{t,a_l} \right) \left(\left\langle \mathbf{f}_{l+1} \mathbf{f}^T_{l+1} \right\rangle_{t,a_l} - \left\langle \mathbf{f}_{l+1} \right\rangle_{t,a_l} \left\langle \mathbf{f}^T_{l+1} \right\rangle_{t,a_l} \right)^{-1}, \tag{7}$$

$$\bar{\mathbf{g}}^{t+1}_{a_l} = \left\langle \mathbf{g}_l \right\rangle_{t,a_l} - M^{t+1}_{a_l} \left\langle \mathbf{f}_{l+1} \right\rangle_{t,a_l}, \tag{8}$$

and

$$\Lambda^{t+1}_{a_l} = \left\langle \left(\mathbf{g}_l - M^{t+1}_{a_l} \mathbf{f}_{l+1} \right) \left(\mathbf{g}_l - M^{t+1}_{a_l} \mathbf{f}_{l+1} \right)^T \right\rangle_{t,a_l} - \bar{\mathbf{g}}^{t+1}_{a_l} \bar{\mathbf{g}}^{t+1\,T}_{a_l}. \tag{9}$$

Here the brackets $\langle . \rangle_{t,a_l}$ denotes the expectation value

$$\langle X \rangle_{t,a_l} = \frac{\sum_x \Pr(a_l, x \,|\, I, \theta^t) X(x)}{\sum_x \Pr(a_l, x \,|\, I, \theta^t)}. \tag{10}$$

4.2. EXPECTATION

In the E-step we need to compute the probabilities of pairs of labels from neighboring layers $\Pr(a_l, a_{l+1}, x_l \,|\, I, \theta^t)$ for given image data. But note that in all occurrences of the reestimation equations, i.e. (5,6) and (10), we need that quantity only up to an overall factor. We can choose that factor to be $\Pr(I|\theta^t)$ and can therefore compute $\Pr(a_l, a_{l+1}, x_l, I|\theta^t)$ instead using

$$\Pr(a_l, a_{l+1}, x \,|\, I, \theta^t) \Pr(I \,|\, \theta^t) = \Pr(a_l, a_{l+1}, x, I \,|\, \theta^t) = \sum_{A \backslash a_l(x), a_{l+1}(x)} \Pr(I, A|\theta^t) \tag{11}$$

The computation of these quantities can be cast as recursion formulae, defined in terms of quantities u and d, which approximately represent upwards and downwards propagating probabilities. The recursion formulae are

$$\begin{aligned} u_l(a_l,x) &= \Pr(\mathbf{g}_l\,|\,\mathbf{f}_{l+1},a_l,x) \prod_{x'\in \mathsf{Ch}(x)} \tilde{u}_{l-1}(a_l,x') & (12)\\ \tilde{u}_l(a_{l+1},x) &= \sum_{a_l} \Pr(a_l|\,a_{l+1}) u_l(a_l,x) & (13)\\ d_l(a_l,x) &= \sum_{a_{l+1}} \Pr(a_l|\,a_{l+1}) \tilde{d}_l(a_{l+1},x) & (14)\\ \tilde{d}_l(a_{l+1},x) &= \frac{u_{l+1}(a_{l+1},\mathsf{Par}(x))}{\tilde{u}_l(a_{l+1},x)}\, d_{l+1}(a_{l+1},\mathsf{Par}(x)) & (15) \end{aligned}$$

The upward recursion relations (12–13) are initialized at $l=0$ with $u_0(a_0,x)=\Pr(\mathbf{g}\,|\,\mathbf{f}_1,a_0,x)$ and end at $l=L$. At layer L Equation 13 reduces to $\tilde{u}_L(a_{L+1},x)=\tilde{u}_L(x)$.[†] Since we do not model any further dependencies beyond layer L, the pixels at layer L are assumed independent. Considering the definition of u, it is evident that the product of all $\tilde{u}_L(x)$ coincides with the total image probability,

$$\Pr(I|\theta^t) = \prod_{x\in I_L} \tilde{u}_L(x) = u_{L+1}. \qquad (16)$$

The downward recursion (14 - 15) can be executed, starting with equation (15) at $l=L$ with $d_{L+1}(a_{L+1},x)=d_{L+1}(x)=1$.[†] The downwards recursion ends at $l=0$ with equation (14).

We can now compute (11) as

$$\begin{aligned} \Pr(a_l,a_{l+1},x,I\,|\,\theta^t) &= u_l(a_l,x)\tilde{d}_l(a_{l+1},x)\Pr(a_l|a_{l+1}) & (17)\\ \Pr(a_l,x,I\,|\,\theta^t) &= u_l(a_l,x)d_l(a_l,x) & (18) \end{aligned}$$

Obviously computations (12–18) in the E-step at iteration t need to be completed with fixed parameters θ^t.

Because of the dependence of $\mathbf{g}_l$ on $\mathbf{f}_{l+1}$, these u's and d's are not, in general, actual probabilities. In spite of this it can be shown that these recursion relations are correct.

5. EXPERIMENTS

5.1. CLASSIFICATION OF VEHICLES IN SAR IMAGERY

Though not a medical imaging problem, we first present the results of our experiments on synthetic aperture radar (*SAR*) imagery, since SAR imagery is noisy and involves detecting an extended textured object, much like a breast mass and many other medical imaging problems. The problem was to discriminate between three target classes in the MSTAR public targets data set, to compare with the results of the flexible histogram approach of De Bonet, et al.[5] We trained three HIP models, one for each of the target vehicles BMP-2, BTR-70 and T-72 (Figure 3). As in Reference 5 we trained each model on ten images of its class, one image for each of ten aspect angles, spaced approximately 36° apart. We trained one model for all ten images of a target, whereas De Bonet et al trained one model per image.

We first tried discriminating between vehicles of one class and other objects by thresholding $\log\Pr(I\,|\,\text{class})$, i.e., no model of other objects is used. In essence this discriminates simply by judging whether an image looks sufficiently similar to the training examples. For the tests, the other objects were taken from the test data for the two other vehicle classes, plus seven other vehicle classes. There were 1,838 image from these seven other classes, 391 BMP2 test images, 196 BTR70 test images, and 386 T72 test images. The resulting ROC curves are shown in Figure 4a.

We then tried discriminating between pairs of target classes using HIP model likelihood ratios, i.e., $\log\Pr(I\,|\,\text{class1}) - \log\Pr(I\,|\,\text{class2})$. Here we could not use the extra seven vehicle classes. The resulting ROC curves are shown in Figure 4b. The performance is comparable to that of the flexible histogram approach.

[†] The (non-existent) label a_{L+1} can be thought of as a label with a single possible value, which is always set. The conditional $\Pr(a_L|\,a_{L+1})$ turns then into a prior $\Pr(a_L)$

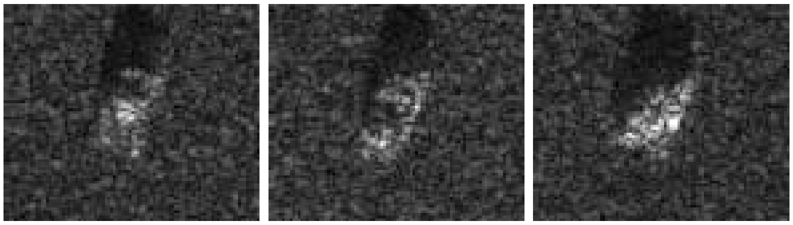

Figure 3. SAR images of three types of vehicles to be detected.

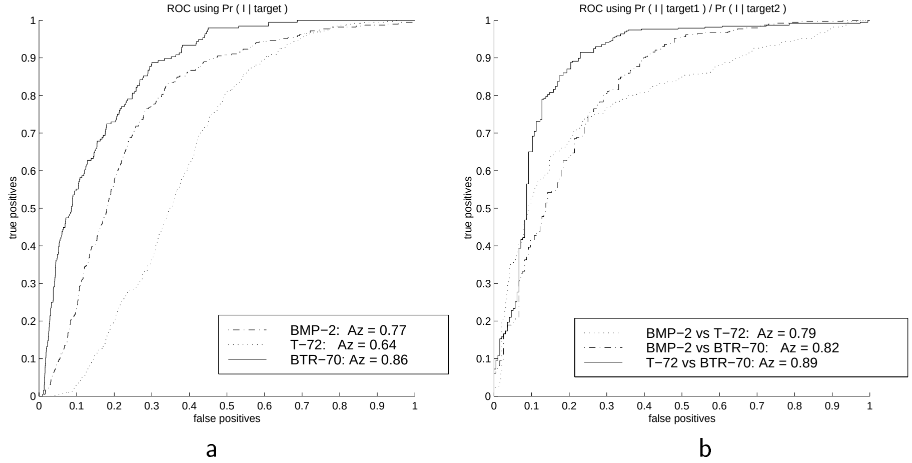

Figure 4. ROC curves for vehicle detection in SAR imagery. (a) ROC curves by thresholding HIP likelihood of desired class. (b) ROC curves for inter-class discrimination using ratios of likelihoods as given by HIP models.

5.2. MASS DETECTION

We applied HIP to the problem of detecting masses in ROIs taken from mammograms, as detected by a CAD system at the University of Chicago. We trained a HIP model of the distribution of positive images on 36 randomly-chosen ROIs that contained masses, and a second HIP model on 48 randomly-chosen ROIs without masses. The likelihood ratio was then used as the test criterion, i.e., a threshold on this ratio is used to decide which ROIs will be called masses. The true and false positive rates as a function of the threshold were measured on a test set with 36 mass and 49 non-mass ROIs.

A search was performed over the number of hidden labels values at each level. The search criterion was the negative log-likelihood on the training data plus the minimum-description-length penalty term, $d\log(N)/2$, where d is the number of model parameters and N is the the number of training examples. The maximum number of labels in a level was bounded (somewhat arbitrarily) at 17, since doubling the number of components in a level at this point was observed to decrease the MDL criterion, but very little, and the computation time would approximately double.

The best architecture had 17, 17, 11, 2, and 1 hidden label in levels 0–4, respectively. For this architecture, A_z was 0.73. This detector had a specificity of 33 % at a sensitivity of 95 %. The ROC curve is shown in Figure 5. While this performance is not as good as we might hope, being worse than our own HPNN classifier,[8] for instance, it demonstrates that the model captures relevant information for classification. We hope that further work, particularly in model and feature selection, will improve on these results.

6. CONCLUSION

We have developed a class of image probability models we call hierarchical image probability or HIP models. To justify these, we showed that image distributions can be exactly represented as products over pyramid levels of distributions of sub-sampled feature images conditioned on coarser-scale image information. We argued that hidden variables are needed to capture long-range dependencies while allowing us to further factor the distributions over position. In our current model the hidden variables act as indices of mixture components. The resulting model is

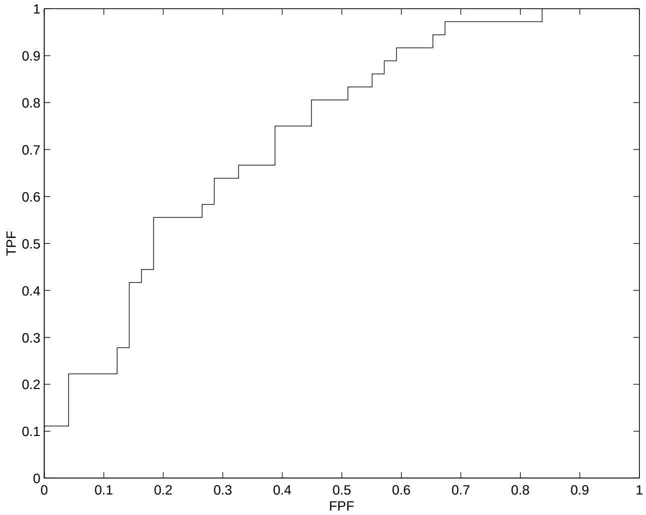

Figure 5. ROC curve for HIP detector of Mass ROIs generated by U. Chicago CAD.

somewhat like a hidden Markov model on a tree. The HIP model can be used for a wide range of image processing tasks besides classification, e.g., compression, noise-suppression, up-sampling, error correction, etc.

There is much room for further work on variations of the specific HIP model presented here. The tree-structured discrete hidden variables lend themselves well to exact marginalization, but they fail to capture certain image properties. For example, contrast level and orientation could be given continuous parameterizations. See, for example, the work of Simoncelli and Wainwright, who developed a very similar model to capture the statistics of contrast level (which they refer to as "scale"), though they did not formulate their model as an image probability.[9] Furthermore, as is well known, the tree structure of the hidden variable dependencies will tend to artificially suppress the statistical dependence between some neighboring pixels, but not others. Allowing multiple parents would alleviate this. Unfortunately, either of these modifications would make it impractical to marginalize over the hidden variables, which is the proper probabilistic procedure. There are approximate alternatives to exact marginalization, which should allow a far wider variety of hidden variable structures.

ACKNOWLEDGEMENTS

We thank Drs. Robert Nishikawa and Maryellen Giger of The University of Chicago for useful discussions and providing the data. This work was supported by the US Department of the Army under grant number DAMD17-98-1-8061. This paper does not necessarily reflect the position or the policy of the US government, and no official endorsement should be inferred.

REFERENCES

1. S. C. Zhu, Y. N. Wu, and D. Mumford, "Minimax entropy principle and its application to texture modeling," *Neural Computation* **9**(8), pp. 1627–1660, 1997.
2. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. PAMI* **PAMI-6**, pp. 194–207, Nov. 1984.
3. R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. ASSP* **33**, pp. 959–963, 1985.
4. J. S. D. Bonet and P. Viola, "Texture recognition using a non-parametric multi-scale statistical model," in *Conference on Computer Vision and Pattern Recognition*, IEEE, 1998.
5. J. S. D. Bonet, P. Viola, and J. W. F. III, "Flexible histograms: A multiresolution target discrimination model," in *Proceedings of SPIE*, E. G. Zelnio, ed., vol. 3370, 1998.
6. M. R. Luettgen and A. S. Willsky, "Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination," *IEEE Trans. Image Proc.* **4**(2), pp. 194–207, 1995.

7. M. I. Jordan, ed., *Learning in Graphical Models*, vol. 89 of *NATO Science Series D: Behavioral and Brain Sciences*, Kluwer Academic, 1998.
8. C. D. Spence and P. Sajda, “Applications of multi-resolution neural networks to mammography,” in *Advances in Neural Information Processing Systems 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, eds., pp. 981–988, MIT Press, (Cambridge, MA), 1998.
9. M. J. Wainwright and E. P. Simoncelli, “Scale mixtures of Gaussians and the statistics of natural images,” in *Advances in Neural Information Processing Systems 12*, S. A. Solla, T. Leen, and K.-R. Müller, eds., MIT Press, (Cambridge, MA), 1999.