Unsupervised Learning for Object Recognition

Samuel Audet

***Abstract*— This report consists of a literature review of papers dealing with object recognition using unsupervised learning techniques. Five papers that brought important contributions to the field are summarized, analyzed and compared. It was found that unsupervised object recognition was considered first as an image segmentation problem, but new unsupervised object learning techniques have been developed requiring no image segmentation at all. In this report, we however stipulate that the latter techniques could possibly benefit from unsupervised image segmentation to provide even better unsupervised object recognition.**

***Index Terms*— Unsupervised learning, computer vision, image segmentation, object recognition, content-based image retrieval, expectation-maximization, clustering.**

I. Introduction

In the future, unsupervised learning techniques could have a great impact on how new problems are approached. The traditional supervised learning methods used in many fields, including computer vision, requires a supervisor to guide the machine, labeling inputs with the outputs we want the machine to learn from. However, this labeling process is a very tedious, especially when the number of data to label is in the millions of items, such as images on the Internet, where content-based image retrieval would be very useful. To overcome this problem, unsupervised learning techniques are required. In computer vision, unsupervised object recognition has traditionally been approached as an image segmentation problem first [1], [2], [3], but more recently new method requiring no segmentation whatsoever [4], [5] have emerged.

II. Literature Review

With regards to unsupervised techniques in general, the main application in computer vision has traditionally been image segmentation [1], [2], [3]. The segmentation here refers to the splitting of an image into regions with similar color or texture, or both. More recently, however, a different approach to object recognition [4], [5] was born. With this approach, images are not segmented at all, but the program can still learn and recognize objects within the background clutter of the images without any supervision other than providing the algorithm with a set of training images all containing the object to be learned and recognized.

A. Image Segmentation

Jain and Fakkorhnia [1] appear to be the firsts to develop a remarkably efficient unsupervised image segmentation technique. Their approach is inspired by the multi-channel filtering theory of the human visual system which describes how the eyes and the related regions of the brain filter and interpret textures. The multi-channel theory stipulates that a visual signal is split and processed in parallel by a bank of filters very similar in properties to Gabor filters, which they make use of for their segmentation technique.

A 2D Gabor function is a function that consists of a sinusoidal wave-front function of a given frequency and phase, in a given orientation on the plane, modulated by a 2D Gaussian function of given variances to limit the sinusoidal in space. A Gabor filter is the use of such a Gabor function as a basis function for a wavelet transform. A wavelet transform is similar in nature to a window Fourier transform, but unlike the Fourier transform where the window size is fixed, the window size for the wavelet transform changes according to the frequency, allowing better localization for higher frequencies. However, Gabor functions are not perfectly orthogonal wavelets, and their use is mainly justified by the evidence from biological systems.

The authors use a fixed set of Gabor filters at orientation intervals of 45 degrees, and the chosen frequencies (cycles/image-width) are all powers of 2 (i.e.: octaves), up to 1/4 of the image width, multiplied by $\sqrt{2}$. For computational efficiency, they select a subset of these filters by choosing those that can reconstruct the test images with a maximum relative pixel-to-pixel error of 5%. After filtering an image with this reduced bank of filters, each of the many produced images are again transformed with a non-linear function, $\tanh(\alpha I_{Gabor}(x, y))$, which acts as an *activation function*, again justified by biological evidence. For each filtered image, a feature image is computed for which each pixel is set to the *texture energy*: the sum of the activated pixels over a small Gaussian window. In addition to the previous feature images, two more images consisting of the pixel coordinates are used. They emphasize the fact that nearby pixels are more likely to be part of the same cluster. To cluster regions together, it is first assumed there are K texture categories. A square-error clustering algorithm known as CLUSTER [6] is then used to group together in an optimal manner the K categories. For computational performance, the original underlying intensity patterns, which are in the first iterations clustered in small regions, are used directly with a minimum distance classifier to more quickly classify other similar regions of the image.

The experimental results are very convincing, and moreover its failure modes are very similar to humans'. However, the number of texture categories needs to be specified as a parameter, which is a serious limitation when applied to unsupervised learning for object recognition. Also no color information is used, only texture information. Panjwani and Healy [2] use a different a approach. Using Markov random fields, they are able to overcome the limitations of the previous

Samuel Audet is with the Electrical and Computer Engineering Department and the Centre for Intelligent Machines, McGill University, Montréal, Québec, Canada. McGill ID: 260184380. E-mail: saudet@cim.mcgill.ca.

Gabor filter approach. Color information is also added to the models of the textures.

In their work, they make use of a special case of the Markov random fields: the Gaussian Markov random fields, which work well in modeling a wide range of textures. With this approach, a given pixel in a given color channel is linearly dependent on its 26-connected immediate neighbors in both geometric and "warped" color space (the standard RGB color space). The 3×26 model parameters are coefficients that dictate the linear relationship between a pixel and all its neighbors. They also represent the colored texture.

The conditional probability density function of a colored pixel given its neighborhood (texture) is then defined as a zero-mean 3D Gaussian function, whose covariance noise matrix is computed from the pixels in the region and the parameters. In this way, if a pixel and its neighbors completely agree with the texture parameters, the input value for the PDF will be zero.

The segmentation algorithm uses Gaussian Markov random fields, and proceeds in three phases: region splitting, conservative clustering, and stepwise optimal clustering.

In the region splitting phase, the image is divided into smaller and smaller square regions, until it is safely determined that the texture in each region is uniform. The initial regions could simply be the pixels and their immediate neighborhood, but it is more computationally optimal to not split regions which are "obviously" uniform. The uniformity test consists of computing the mean color error of all pixels in the region, and comparing it with the mean color error of each of the four possible sub-square-regions. If the difference is below some threshold, and if the covariance of the region is also below another certain threshold, then the region is not split. The threshold values are very conservative in order to split regions unless uniformity is very statistically significant.

Then, the conservative merging proceeds locally by merging similar adjacent regions. This phase is used to remove "obvious" workload from the next phase. A few tricks are used to estimate the similarity between two adjacent regions candidate for merging: the color mean differences need to be below a certain threshold, the covariance matrix needs to be small (tested with a threshold similar to the one used during the region splitting phase), and if the regions are not too small for pseudo-likelihood estimation (see next paragraph), it needs to be below another threshold. Again, the threshold values are very conservative in order not to merge regions unless uniformity is very statistically significant. All candidate regions are processed until equilibrium.

Finally, stepwise optimal merging continues to process merging operations until a given stopping criterion is reached. Since we are interested in the parameters of the texture, the PDF associated with each pixel is the likelihood of the parameters. Multiplying PDF values of all pixels in a given region provides a pseudo-likelihood. This is obviously not a true likelihood since neighboring pixels are dependent of each other by definition, but using this value for a maximum-likelihood approach proved to give good results for parameter estimation. This approach can be partially justified by the fact that a locally defined Markov random field is equivalent to a globally defined Gibbs random field. In this manner, for two adjacent regions candidate for merging, the algorithm proceeds by comparing the pseudo-likelihood of the whole image (by multiplying the pseudo-likelihood of all the regions) when the two regions are left separate, and when they are merged. All adjacent regions to a region are likewise tested, and the pair with the maximum pseudo-likelihood wins, and are merged if the difference in global pseudo-likelihood, which will increase as merging progresses, is less than a given stopping criterion. Other candidate regions are processed until the stopping criterion is reached. Again, if one region of the pair is too small for pseudo-likelihood estimation, color mean difference tricks are used, but without thresholds. They will be merged if no better pairs exist.

Mainly thanks to its use of colored textures, this unsupervised technique for image segmentation gives better results than the previous one using Gabor filters, especially in the case of natural images. It also does not required the number of regions to be specified in advance. The algorithm will find by itself the best segmentation and number of regions given the stopping criterion. Also, given its local "Markovian" nature, the algorithm can easily be implemented on a parallel architecture for improved performance. Nonetheless, image segmentation is only part of the object recognition problem. Blobworld [3] uses image segmentation as the basis for a content-based image retrieval system. Its expectation maximization approach to image segmentation is again quite different from, but similar in the results to the "Markovian" approach.

In Blobworld, color, position and texture features are extracted from each pixel. The color feature is simply the color components of the pixel in the L*a*b color space, more perceptually uniform to humans than the RGB color space. Likewise, the position feature is simply the (x, y) position of the pixel in the image. The texture features are however more elaborate. The texture information is taken solely from the grayscale intensity component of the pixel, or in this case the L* component of the L*a*b components. An approximation of the gradient $\nabla I(x, y)$ is computed using first order finite differences. The second moment matrix is approximated by convolution $M(x, y) = G(x, y) * \nabla I(x, y) \nabla I(x, y)^T$, where $G(x, y)$ is a 2D symmetric Gaussian function with parameter σ denoting the scale of the texture feature. In order to choose σ, a measure of *polarity* is computed. The polarity indicates the extent to which the orientation of the neighboring pixels' gradient is similar. It is computed for all pixels with $\sigma_k = k/2, k = 0, 1, ..., 7$. Each polarity image p_k thus obtained is smoothed with a Gaussian kernel of standard deviation $2\sigma_k$ producing $p'_k(x, y)$. The selected σ for a given pixel is the one for which $\frac{p'_k(x,y) - p'_{k-1}(x,y)}{p'_k(x,y)} < 0.02$. This criterion is met when the scale of the Gaussian and the polarity encompasses the spatial frequency of the texture, and thus gives a measure of the scale of the texture. If the contrast of the underpinning pixels is less than 0.1, the scale is set to 0. This said, the texture feature for a pixel is actually described by ac, $p_{k^*}c$ and c, where k^* corresponds to the chosen scale, $a = 1 - \lambda_2/\lambda_1$, to the anisotropy, $c = 2\sqrt{\lambda_1 + \lambda_2}$

to the normalized texture contrast, where $\lambda_1 >= \lambda_2$ are the eigenvalues of the matrix $M(x, y)$. a and p_k are modulated by the contrast c so they approach 0 when the contrast also approaches 0 (i.e.: no texture). This texture descriptor is also thus invariant to orientation and scale. The original image is then smoothed with a Gaussian filter of the given scale at each pixel, effectively removing texture information while keeping color and structure. Note that the L*a*b color features are affected by this smoothing.

The expectation maximization algorithm [7] is then used on this eight-dimensional feature space. The algorithm can be briefly summarized by this equation:

$$\theta_{n+1} = \arg\max_{\theta} \sum_{z} p(z|x, \theta_n) \ln p(x, z|\theta), \tag{1}$$

where θ_{n+1} is the new estimate of the parameters, θ_n is the previous estimate of the parameters, z is the latent or hidden variables, and x is the observed data. $p(x, z|\theta)$ is the likelihood of the parameters for our observed data and hidden variables. The summation calculates the expectation of the log-likelihood with respect to the probability of the hidden variables knowing our data and the previous estimate of the parameters. In other words, it returns how likely the new parameters are given what we know about the data and the previous estimate of the parameters. The maximum over possible new parameters is used as part of the next iteration. The algorithm iterates until convergence to a certain stopping criterion based on the improvement of the log-likelihood or on the convergence of the estimated parameters. The algorithm is only guaranteed to converge to a local maximum, so domain specific initialization tricks might be required if it does not converge to satisfactory values.

In the case of Blobworld, the parameters are the means and covariances of the mixture of Gaussians representing each feature for each of the K segmented regions. The observed data are the eight features per pixel, and the hidden variables indicate which segmented region each pixel belongs to. In Blobworld, the algorithm is restarted four times with added Gaussian noise to the initial mean estimates in order to avoid low local maxima. The algorithm is further run over a range of K from 2 to 5, and the log-likelihood is used as a measure of how good of a fit the segmentation is. A Minimum Description Length criterion, which is an operational form of Occam's razor,

$$\arg\max_{K} \left[\ln p(x, z|\theta_K) - \frac{\dim \theta_K}{2} \ln N \right], \tag{2}$$

where N is the number of pixels, is used to determine the minimum value of K needed to represent the data.

The description of each region or blob is stored in memory as $5 \times 10 \times 10$ bins in L*a*b color space, and as mean contrast and anisotropy. In order to get a matching score for recognition of blobs within other images, a difference measure between histograms and mean values is calculated using weighted Euclidean distances.

One of the disadvantages of this segmentation method compared to the previous ones is that it will sometimes split large uniform regions. This is due to the use of pixel positions in the EM algorithm, but for image retrieval uses it is not deemed by the authors to be a serious problem. It segments images sufficiently well for this practical application it was developed for. Nonetheless, as the authors point out, sophisticated object recognition is the ultimate goal. The simple comparison of histograms is an insufficient descriptor for real-world objects. Recently, such unsupervised object recognition algorithms have emerged.

B. Feature Based Recognition

There are two important contributions [4], [5] to the problem of completely unsupervised object recognition, where the latter follows on the former's work. In this framework, an object model is represented as the *shape* of the constellation of rigid *parts* (features). The registration of training images, the part selection and the estimation of the model parameters are all accomplished automatically. This technique however requires a data set composed of images all containing the object to be later recognized. Before training, it cannot discriminate between multiple object classes by itself.

First, Weber et al. [4] define a generative probabilistic model with three random variables: observed rigid parts X^o (comprised of a number of observed part candidates, denoted by their (x, y) coordinates, for each part type), and two hidden variables: $\mathbf{h}$ being the *hidden hypothesis* of which candidate parts belong to the foreground (the object to be recognized), and $\mathbf{x^m}$ consisting of valid foreground rigid parts that are not observed (*missing*). Then they define two new auxiliary hidden variables: $\mathbf{b}$ and $\mathbf{n}$. $\mathbf{b}$ encodes in a binary fashion which part types have been detected (1), and which have been missed or occluded (0). $\mathbf{n}$ indicates the number of background candidates for each part type. Noting that $\mathbf{b}$ and $\mathbf{n}$ are fully determined by $\mathbf{h}$ and X^o (sufficient statistics), the probabilistic model is defined as:

$$p(X^o, \mathbf{x^m}, \mathbf{h}, \mathbf{n}, \mathbf{b}) = p(X^o, \mathbf{x^m}|\mathbf{h}, \mathbf{n}) p(\mathbf{h}|\mathbf{n}, \mathbf{b}) p(\mathbf{n}) p(\mathbf{b}), \tag{3}$$

where $p(\mathbf{n})$ is modeled by a Poisson distribution, which relays the assumption of independence between part types in the background as well as the independence of their positions. For simplicity, $p(\mathbf{b})$ is modeled with a table of independent probabilities over the total number of part types, but modeling joint probabilities might be more powerful. Also note that $\mathbf{b}$ becomes parameters to be estimated. $p(\mathbf{h}|\mathbf{n}, \mathbf{b})$ is defined simply as a uniform density for all $\mathbf{h}$ consistent with $\mathbf{b}$ and $\mathbf{n}$, and 0 elsewhere. $p(X^o, \mathbf{x^m}|\mathbf{h}, \mathbf{n})$ is rewritten as $p_{fg}(\mathbf{z}) p_{bg}(\mathbf{x_{bg}})$ where $\mathbf{z}$ is the set of all observed and missing foreground parts, and $\mathbf{x_{bg}}$ is the observed background parts. These two densities, the foreground and the background, are assumed to be independent. $p_{fg}(\mathbf{z})$ is modeled as a joint Gaussian density whose means are centered around the position of the foreground parts and are parameters to be estimated, and the observed background parts $p_{bg}(\mathbf{x_{bg}})$ are modeled as a uniform density over the image.

In order to select parts that describe the objects well, points of interest are first detected from the training images using the

Förstner operator. This operator selects corners, intersections and center points of circular patterns. It produces about 150 points for each of the training images used by the authors. To reduce this number, a k-means clustering algorithm is applied to the patterns (small grayscale sub-image) of those points to reduce their number to about 100 for the whole set of training images. Only clusters with more than 10 patterns are kept, leaving out unpopular patterns, and new part images are produced by averaging over the patterns in each cluster. Although the number of parts remaining after this operation is much reduced, background parts are still present.

In order to train the model, foreground parts need to be identified. The first issue that needs to be addressed is the number of parts that should be used. If not enough parts are used, then the model is not descriptive enough and does not perform well. If too many parts are used, the complexity of the model increases and we risk overfitting the training data. This is again the concept behind Occam's razor. In order to select which parts to keep a greedy search is carried out over the possible set of parts. For a given number of parts that will be used (F), which parts to choose is determined by random selection. The model parameters are then estimated with the training images by expectation maximization, as described previously, on the probability density function of equation 3 given the observed data, the hidden variables and the parameters to estimate. The classification performance is then observed on the validation image set, which is separate from the training image set in order to statistically validate the new model. New part configurations that improve performance are kept. This process is akin to Monte Carlo sampling. The process ends when all part configurations have been tried, or after a given number of trials. The variation in performance with different values of F is observed as well. Tests with cars and faces show that values for F of 4 for faces and 5 for cars are optimal.

Classification is defined in terms of the object being *present* or *absent* from the image. The maximum a posteriori probability (MAP) hypothesis (present or absent) is chosen. Once model parameters are estimated, by application of the Bayes' theorem, the ratio $\frac{P(present|X^o)}{P(absent|X^o)}$ can be computed and is used along with a threshold to make a decision for classification. Given the data, the classification provides a clue as to how close to the model the *appearance* of each part is, and how closely their position matches with the model, the *shape*.

The algorithm was validated by experimenting with 200 images showing cars and 200 images showing faces. 100 background images of each environment are also included in each set. After training on 100 face images, and classifying the remaining 100 face images and 100 background images from the face environment, 93.5% of the images were correctly classified. For cars, the accuracy was of 86.5%. This method although very effective has some limitations. It is not invariant to scale, and does not take into account changes in appearance (viewpoint and lighting conditions). The approach by Fergus et al. [5] follows on from this work. It adds to the model representation better invariance to scale and appearance.

First, they replace the Förstner operator by the one developed by Kadir and Brady [8]. This new operator computes the histogram $P(x, y, s)$ over the image at all regions with center (x, y) and scale (radius) s. The entropy (from information theory) of the histograms $H(P(x, y, s))$ at different scales is then calculated and the points with local maximum entropy (meaning the most different ones) are selected as candidate parts at scale s. Parts with maximum saliency $\frac{dP}{ds}H$ (scale normalized) are used as parts for learning and recognition.

This new scale information from the operator is added to the location and shape variable X^o by normalizing all position measurements to the scale of the parts. The scale information is also added to the probabilistic model as a new relative scale variable S, modeled by another Gaussian density.

Another variable added to the model is the appearance A of a part. It is modeled by PCA where the 10 to 15 first principal components (eigenvectors of the covariance matrix) are used to describe the appearance of 11×11 pixels patches. A is also modeled by a joint Gaussian density, but with no covariances, only variances. A background model is also stored as a joint Gaussian in the same manner.

The new variables are added as such into the probabilistic model designed by Weber et al. [4] and the same expectation maximization algorithm is used over the new set of variables. However, as optimization, the A* space-search algorithm is used and makes for a "considerable improvement" in performance, although they provide no basis for comparison. Recognition is also performed similarly to the previous method, but again with the new set of variables. To avoid overfitting data, large datasets (up to 400 images in size) are used. The performance of the classifier was found to be remarkably consistent with only very slight variations (e.g.: $\sim$1%) in classification error when giving different initial estimates to the EM algorithm.

During experiments, the number of parts used was 6, compared to 4 and 5 for the previous method [4]. Both of these algorithms are compared in the new paper [5]. In the case of motorbikes, performance increases from 84% (ROC) to 92.5% using the new scale and appearance variables. For airplanes, it jumps from 68% to 90.2%. Moreover, when executing the recognition phase on images without the object of interest, it was found that the ROC was about 50% on average, so it can be considered good at discriminating images. The general formulation of the framework is very general, and these particular implementations could be improved upon. A possibly better feature extractor operator, like SIFT, as well as use of color information should provide even better recognition rates. Also multi-modal densities are not supported. They could be useful for example in the case of radically different appearances belonging to the same part type (for example, faces with and without sunglasses). Affine and projective transform invariance would also improve recognition under 3D out-of-plane rotation.

III. Conclusion

Unsupervised learning techniques in computer vision have come a long way in just a few years. Algorithms for detecting and recognizing objects in a scene without any required preprocessing of the images are now available. These techniques

cannot however cluster objects in classes by themselves. Images in the training data set must be manually separated into all the different classes of objects before training. On the other hand, image segmentation techniques have also evolved up to the point where the segmented regions could be used in conjunction with such object recognition techniques. Merging these techniques could greatly reduce the number of hypotheses generated by feature (part) detection, allowing more processing power to be dedicated to automatic clustering of multiple classes of objects during the training phase.

REFERENCES

[1] A. Jain and F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters," *Pattern Rcognition*, vol. 24/12, pp. 1167–1186, 1991.

[2] D. K. Panjwani and G. Healey, "Markov Random Field Models for Unsupervised Segmentation of Textured Color Images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 10, pp. 939–954, 1995.

[3] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1026–1038, 2002.

[4] M. Weber, M. Welling, and P. Perona, "Unsupervised Learning of Models for Recognition," *Proc. 6th European Conference Computer Vision (ECCV)*, pp. 18–32, June 2000.

[5] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 264–271, June 2003.

[6] B. Julesz and J. Bergen, "Textons, The fundamental Elements in Preattentive Vision and Perception of Textures," *Bell Syst. Tech. J.*, vol. 63, no. 6, pp. 1619–1645, 1983.

[7] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[8] T. Kadir and M. Brady, "Saliency, Scale and Image Description," *Int. J. Comput. Vision*, vol. 45, no. 2, pp. 83–105, 2001.