SUPERVISED TEXTURE SEGMENTATION THROUGH A MULTI-LEVEL PIXEL-BASED CLASSIFIER BASED ON SPECIFICALLY DESIGNED FILTERS

Jaime Melendez, Xavier Girones, Domenec Puig

Intelligent Robotics and Computer Vision Group
Department of Computer Science and Mathematics
Rovira i Virgili University
{jaime.melendez,xavier.girones,domenec.puig}@urv.cat

ABSTRACT

This paper presents a new, efficient technique for supervised texture segmentation based on a set of specifically designed filters and a multi-level pixel-based classifier. Filter design is carried out by means of a neural network, which is trained to maximize the filters' discrimination power among the texture classes under consideration. Texture features obtained with these filters are then processed by a classification scheme that utilizes multiple evaluation window sizes following a top-down approach, which iteratively refines the resulting segmentation. The proposed technique is compared to previous supervised texture segmenters by using both synthetic compositions and real outdoor textured images.

Index Terms— Supervised texture segmentation, Specific texture filters, neural networks, multi-level classification

1. INTRODUCTION

One of the most actively studied application domains related to texture analysis is texture segmentation, which aims at partitioning a given image into disjoint regions of uniform texture. In order to achieve this objective, features for each pixel are computed by applying a set of texture feature extraction methods (texture methods, in short) considering that pixel and its surrounding neighborhood (evaluation window). Among the variety of texture methods proposed in the literature, those based on multichannel filtering have received especial attention due to their versatility and neuropsychological support (e.g., [1]).

The main problem in the design of a multichannel filtering system is the selection of an appropriate filter bank. One possibility is to localize the filter responses in the spatial-frequency domain in such a way that the frequency plane is covered as much as possible (e.g., [2]), thus obtaining a filter set that is expected to discriminate among a wide variety textures. On the other hand, filters can also be designed adaptively and only for the limited number of textures under consideration (e.g., [3]).

One of the main advantages of adaptive filter design is that this problem can be outlined as an optimization problem, thus well-known optimization techniques such as those corresponding to neural networks (NNs) and support vector machines (SVMs), can be easily applied. Remarkable examples are the approaches introduced in [3],[4], which are a generalization of the multichannel filtering method, as they include both feature extraction and classification into a single network, or the one-against-all SVMs utilized in [5] as feature extractors and classifiers.

In this paper, a similar approach as in [3],[4] is followed, but considering a simpler NN (a standard multilayer perceptron (MLP)) in order to design a specific filter bank for supervised texture segmentation. Furthermore, instead of letting the NN to perform the pixel classification task as in [3],[4], only the weights in the first layer are kept and are utilized as convolution masks in order to obtain texture features, which are then processed by a powerful, multi-level pixel-based classifier based on SVMs, which yields the final segmented image. It is shown through experiments that this new classification methodology leads to better segmentation results than previous supervised segmenters both in terms of accuracy and processing time.

The organization of this paper is as follows. Section 2 elaborates on the filter design methodology and feature extraction. Section 3 describes the proposed multi-level classification scheme. Section 4 shows and discusses the experimental results. Finally, conclusions and further improvements are given in section 5.

2. FILTER BANK DESIGN USING AN MLP

In an MLP, the output of the *i*-th unit for a given layer can be described by

This work has been partially supported by the Spanish Ministry of Education and Science under project CSD2007-00004.

$$O_i = g(h_i) = g\left(\sum_k w_{ik} v_k - \theta_i\right), \quad (1)$$

where v_k is the k-th input, w_{ik} is the weight connecting input k with output i, θ_i is the bias term, and $g(h)$ is the activation function. Now, utilizing vector notation for the weights and inputs, h_i can be expressed in terms of a dot product:

$$O_i = g\left(\mathbf{w}_i \mathbf{v} - \theta_i\right). \quad (2)$$

A way of using an MLP for supervised texture segmentation is to supply it with vectors corresponding to image pixels contained in rectangular windows. In this case, the vector of weights $\mathbf{w}_i$ acts as a convolution kernel (see Figure 1) and the set of $\mathbf{w}_i$ corresponding to the units in the first layer (L1) is equivalent to a filter bank. From this perspective, the first layer behaves as a feature extraction stage, while the upper layers perform the classification.

Alternatively, the set of $\mathbf{w}_i$ can be taken separately and used for feature extraction with a different classifier, which is the approach followed in this paper: an MLP is first trained for supervised texture segmentation and the weights in the first layer are used as a replacement for a filter bank within an SVM-based classifier. The rationale behind this methodology is that, at the end of the MLP training process, the resulting weights are optimal in the least squares sense and thus likely to perform better than a generic filter bank for the specific problem the MLP has been trained for. After convolution with these filters, texture features such as the mean and standard deviation computed from a sliding window of N x N can be obtained.

2.1. MLP architecture

The MLP used in this work consists of three fully connected layers. The first layer comprises 24 square filters, each one containing N x N weights. A different MLP is trained for each pattern set and mask size (N = 3, 5 and 9). The hidden layer consists of 48 units. This number has been determined empirically and is associated with the minimum capacity required to obtain a good set of filters. The output layer contains as many units as texture patterns M to be classified. Finally, the selected activation function is the conventional sigmoid, which, for this specific problem, performs better than the hyperbolic tangent.

2.2. MLP training

Around 20000 random N x N samples are taken from each training pattern. The M output classes are encoded using a 1 of M coding scheme. Weights are initialized by sampling from a uniform distribution in the interval [0, 25/W], where W is the number of weights in the unit.

Four MLPs are trained and the best performing one in terms of classification error against an independent test set

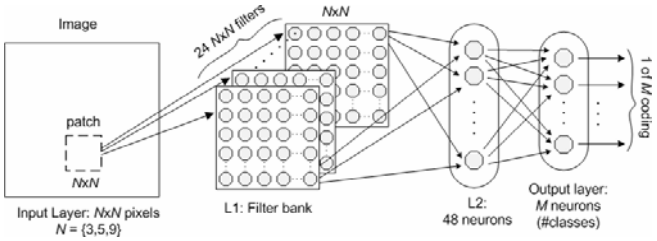


Figure 1. MLP architecture used for filter computation.

is kept. Training consists of two stages. First, a variant of gradient descent with back-propagation, momentum and weight decay [6] is used. This stage is semi-online, with duration of 525000 epochs and an epoch size of 512 samples. The values of the learning rate η and weight decay λ are reduced progressively, ranging from $\{\eta = 0.04, \lambda = -5.10^{-4}\}$ to $\{\eta = 0.01, \lambda = -1.10^{-4}\}$, whereas the momentum is kept constant to 0.9. The second stage refines the solution by combining the conjugate gradient method [7] and simulated annealing [8] to escape local minima. It is run for a maximum of 4 iterations, each one stopping when no significant improvement in MSE is measured.

3. PIXEL-BASED CLASSIFICATION SCHEME

In order to take advantage of the output of the specific filters developed in the previous section, the texture features computed for the different window sizes are processed by following the top-down approach summarized below [9]:

1. Select the largest available window and classify the test image pixels labeled as unknown (initially, all pixels are labeled as unknown).
2. Locate the pixels of the classified image that constitute the frontiers between regions of different texture and mark them as unknown, as well as their neighborhoods. The size of the neighborhood corresponds to the size of the evaluation window used to classify the image.
3. Discard the current evaluation window.
4. Repeat steps 1 to 3 until the smallest window has been utilized.

In this way, large windows are applied inside regions of homogeneous texture in order to avoid noisy classified pixels and small windows are applied near the frontiers between those regions in order to refine them. Furthermore, the above strategy renders classifying every image pixel with all the available windows unnecessary. Therefore, it leads to low computation times.

Pixel classification in the previous scheme is performed by means of SVMs following the one-against-one multiclass extension, which requires a binary classifier for each pair of classes. The final classification considering every pair of classes (k,k') is obtained by the following rule: if feature vector $\mathbf{x}$ belongs to class k according to the sign of the decision function, then the votes for class k are incremented

by one. Otherwise, the votes associated with class k' are incremented by one. In the end, $\mathbf{x}$ is assigned to the class with the largest number of votes. In order to obtain an accurate classification, probability estimates have been included instead of discrete votes.

Finally, in order to achieve the maximum classification speed, the support vector reduction method described in [10] is used. Unfortunately, this method is only applicable to SVMs with linear kernels, which means that more sophisticated kernels, which are often shown to yield higher accuracy, cannot be utilized. However, it will be shown that with the proposed top-down classification scheme, the seemingly "weak" linear SVMs are competitive enough.

4. EXPERIMENTATION

The proposed methodology for supervised texture segmentation has been evaluated on composite images of well-known Brodatz [11], MeasTex [12] and Vistex [13] textures, and on real outdoor images taken both at ground level and by aerial devices. Figure 2 shows some examples of these 256 x 256 pixels sized input test images.

In order to further validate this methodology, a comparison with other supervised pixel-based texture classifiers in terms of segmentation quality (measured by the classification rate) and processing time has been performed. These alternative techniques are: (a) The multi-level, prototype-based classifier proposed in [9], which utilizes texture features from a Gabor filter bank [2] evaluated over five window sizes. (b) The classifier based on integration of multiple methods and windows described in [14], with the same texture features as the previous one but with six window sizes. (c) The SVM/NN-based approach for texture feature extraction and classification proposed in [5]. (d) An extension to pixel-based classification of the local binary pattern-based classifier proposed in [15]. (e) A similar extension of the classifiers included in the MeasTex suite [12], which have been independently evaluated with features derived from classical Gabor filters, the fractal dimension, gray level co-occurrence matrices and Markov random fields.

Figures 3 and 4 show the segmentation maps produced by the most accurate of the evaluated approaches for the images in Figure 2. Black areas correspond to regions that do not belong to any of the sought texture patterns. Pixels belonging to these "unknown" patterns have not been taken into account when computing classification rates. In addition, black borders in the segmentation maps correspond to pixels that could not be classified due to the utilized window size. They have also been discarded.

The average results for this comparison are summarized in Table 1. Experiments have been run on a Pentium 4 at 3.2 GHz with 2 GB of RAM. Only the best

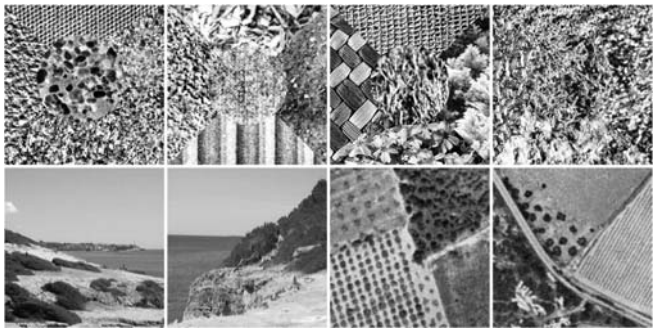


Figure 2. Examples of synthetic compositions and outdoor scenes.

classification results for the MeasTex and LBP-based classifiers have been considered. As shown in Table 1, the proposed classifier is the best in terms of average classification rate. This should be attributed to: (a) The specificity achieved by the MLP-designed filters, which characterize the salient features of the texture patterns better than the remaining texture methods. (b) The multi-level classification strategy, which iteratively improves the result yield by a "weak" classifier, such as the single-window, linear SVM-based classifier utilized in this work.

In terms of processing time, the proposed approach is clearly the fastest one. For instance, it is more than six times faster than the closest (and most similar) alternative classifier, which is the multi-level, prototype-based classifier in [9]. The reasons for this improved performance are the following: (a) Due to the specificity achieved by the MLP-designed filters, only three window sizes are necessary to achieve good results instead of the five window sizes required by the Gabor filters, which leads to much faster feature extraction and classification processes. (b) The MLP-designed filters have real coefficients, in contrast to the complex coefficients of the Gabor filters, which renders faster convolutions. (c) The considerable speed-up yield by the applied support vector reduction method.

5. CONCLUSIONS

This paper presents a new efficient technique for supervised texture segmentation based on a specifically designed filter bank and a multi-level pixel-based classification scheme that achieves good segmentation results with low processing times. Comparisons with alternative supervised pixel-based classifiers have been favorable both in terms of segmentation accuracy and execution time.

Future work will consist of improving the proposed filter design by employing convolutional networks to optimize the filter bank for capturing the most salient features in a neighborhood of pixels rather than just in a single location, thus leading to the increase of the current segmentation accuracy.

Table 1. Average classification rates and classification times (in seconds) for the evaluated approaches.

Pixel-based classifier	Average classification rate	Average CPU time
Proposed	91.95	0.80
Prototype-based [9]	89.39	5.11
Integration-based [14]	84.05	279.65
SVM-NN [5]	72.78	122.83
LBP-G [15]	78.95	56.85
MeasTex [12]	72.09	387.79

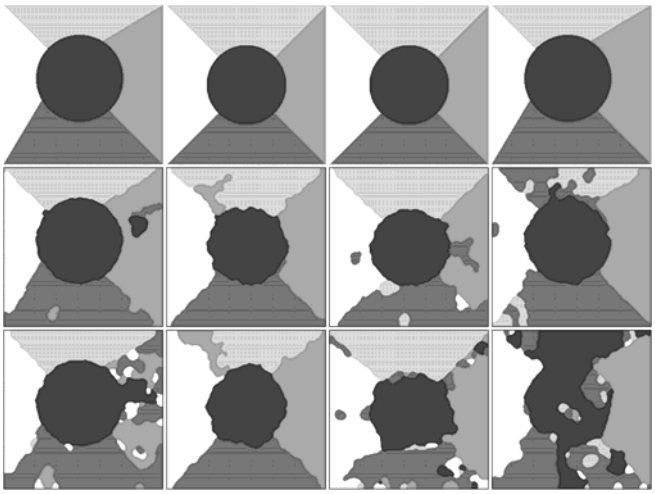


Figure 3. Segmentation maps corresponding to the compositions in the first row of Figure 1. Ground-truth (*first row*), proposed classifier (*second row*) and prototype-based classifier in [9] (*third row*).

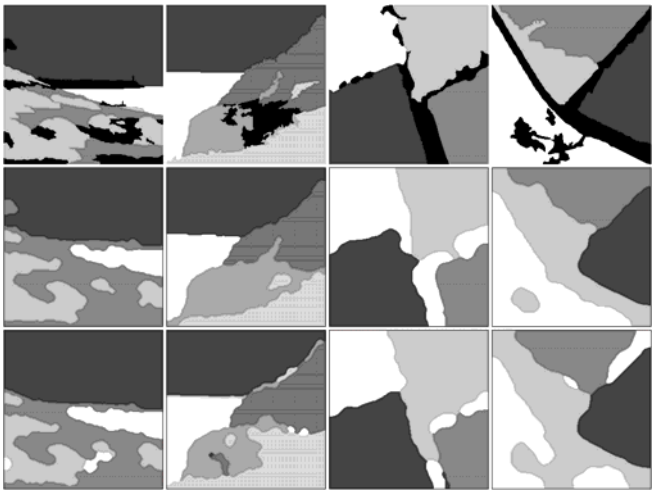


Figure 4. Segmentation maps corresponding to the outdoor scenes in the second row of Figure 1. Ground-truth (*first row*), proposed classifier (*second row*) and prototype-based classifier in [9] (*third row*).

6. REFERENCES

[1] J.G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Res.*, 20, pp. 847-856, 1980.

[2] B.S. Manjunath and W.Y. Ma, "Texture features for browsing and image retrieval," *IEEE T. Pattern Anal.*, 18(8), pp. 837-842, 1996.

[3] A.K. Jain and K. Karu, "Learning texture discrimination masks," *IEEE T. Pattern Anal.*, 18(2), pp. 195-205, 1996.

[4] F.H.C. Tivive and A. Bouzerdoum, "Texture classification using convolutional neural networks," *IEEE Region 10 Conference*, pp. 1-4, 2006.

[5] K.I. Kim et al., "Support vector machines for texture classification," *IEEE T. Pattern Anal.*, 24(11), pp. 1542-1550, 2002.

[6] A. Gupta and M. Lam, "Weight decay backpropagation for noisy data," *Neural Netw*., 11, pp. 1127–1138, 1998.

[7] Y. LeCun, et al., "Efficient BackProp", in Orr, G. and Muller K. (Eds), *Neural Networks: Tricks of the trade,* Springer*,* 1998.

[8] Aarts E. and Korst J., *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, Wiley, December 1988.

[9] J. Melendez et al., "Multi-level pixel-based texture classification through efficient prototype selection via normalized cut," *Pattern Recogn.*, 43(12), pp. 4113-4123, 2010.

[10] D. Tsujinishi et al., "Why pairwise is better than one-against-all or all-at-once," *Proc. IEEE IJCNN*, pp. 693-698, 2004.

[11] Brodatz P, *Textures: A Photographic Album for Artists and Designers*, Dover & Gree Publishing Company, 1999.

[12] G. Smith and I. Burns, "Measuring texture classification algorithms," *Pattern Recogn. Lett.*, 18(14), pp. 1495-1501, 1997.

[13] MIT VisMod Group, http://vismod.media.mit.edu/, 1998.

[14] D. Puig and M.A. Garcia, "Automatic texture feature selection for image pixel classification," *Pattern Recogn.*, 39(11), pp. 1996-2009, 2006.

[15] T. Ojala et al., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE T. Pattern Anal*, 24(7), pp. 971-987, 2002.