A Neural System for Deforestation Monitoring on Landsat Images of the Amazon Region

Valmir C. Barbosa*
Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

Ricardo J. Machado, and Frederico dos S. Liporace
IBM Rio Scientific Center, Rio de Janeiro, Brazil

ABSTRACT

We deal with the problem of automating the interpretation of satellite images of the Amazon region for deforestation monitoring. Our approach is based on a combination of image segmentation and classification techniques, the latter employing a neural-network architecture that works on a fuzzy model of classification. The architecture implements a relaxation mechanism on top of a feedforward neural network, in order to take advantage of the interrelations among neighboring image segments. Our fuzzy, segment-based approach has numerous advantages over more traditional, pixel-based approaches employing statistical techniques. These advantages range from the possibility of treating transition and interference phenomena in the images to the ease with which complex information related to a region's geometry, texture, and contextual setting can be used. We report on a great variety of experiments on representative portions of the Amazon region, employing neural networks trained by the back-propagation algorithm. The results indicate very good overall performance, and allow us to draw some conclusions regarding the effectiveness of the various sources of information available

Address correspondence to Dr. Ricardo J. Machado, IBM Rio Scientific Center, Caixa Postal 4624, 20001-970 Rio de Janeiro, Brazil.

* Valmir C. Barbosa participated in this research as a visiting scientist at the IBM Rio Scientific Center.

Received November 1993; accepted May 1994.

International Journal of Approximate Reasoning 1994; 11:321–359

655 Avenue of the Americas, New York, NY 10010 0888-613X/94/$7.00

as input to the system. In particular, it appears that simple spectral information, together with textural information on a region's entropy and correlation and simple contextual information, are effective in the classification for deforestation monitoring. It also appears that the effective incorporation of geometric information would require further investigation on possible enhancements to the system.

KEYWORDS: ***Remote sensing, Landsat imagery, deforestation in the Amazon region, image segmentation, image classification, fuzzy neural networks.***

1. INTRODUCTION

The Amazon region is one of the major components of the planet's environment. Covering several million square kilometers, the region has the world's largest rain forest and river system, playing a very important role in many global processes. Recent decades have witnessed a dramatic increase in human activity in the region, including deforestation for various types of settlement, dam and road building, mining, agriculture, and cattle raising, all with potential effects on environmental stability.

Although of great relevance in the context of several global issues, the understanding of the extent to which human activity in the Amazon region can be harmful is rather superficial, particularly because of the lack of reliable information. The use of satellite imagery for nearly a decade has improved the situation somewhat, but the process of extracting significant information from the remotely sensed images is still rudimentary, representing a serious bottleneck, as it has to be repeated yearly for incremental monitoring.

The essential problem to be tackled is the yearly determination of the fraction of the Amazon region that has undergone deforestation, as well as of the locations where this process has been most pronounced. Currently, the extraction of information from the satellite images is to a large extent achieved manually, and essentially can be regarded as comprising two phases. In the first phase the images are examined by photointerpreters, who by visual inspection partition each of them into relatively "homogeneous" regions, which are then labeled as belonging to one of the various thematic categories of interest. Typically, these categories are identified beforehand on the basis of the aid they are expected to provide in determining the deforestation status of the region. The professionals who carry out this first phase rely heavily on their knowledge and experience concerning the general appearance of the images and the region under study. This knowledge encompasses spectral and textural characteristics of the images, as well as various geometric features of the regions identified. In addition, contextual information related to each region's vicinity and more broadly to the geography of the entire region under examination is

also used. The second phase consists in the digitization of the labeled image for incorporation into geographic information systems.

Not only is this two-phase process in its entirety too costly, but also it renders some images virtually intractable. As a consequence, a reliable overall assessment of the deforestation in the Amazon region, with its trends for future years clearly identified, is still unavailable. One major effort toward the availability of such an assessment is to automate considerably more of the entire process, which necessarily involves the use of automatic image classification techniques. Several such techniques can be borrowed from the field of image analysis [1, p. 571], and within the realm of remotely sensed images have indeed been employed. Many of these techniques approach the image on a pixel-by-pixel basis (or on the basis of small, fixed-size neighborhoods around pixels) and have a statistical nature [2–6] (although variations employing neural networks have appeared as well [7–11]). In general, these techniques rely on spectral data alone to perform the classification.

In this paper we describe an approach to image classification which proceeds in two phases. The first phase, contrasting with the traditional pixel-by-pixel approaches, consists in the segmentation of the image into spectrally homogeneous portions (called *segments*). There is in the literature a great variety of techniques to perform the segmentation of an image. The one we employ is relatively simple, and belongs to the class of the so-called region-growing techniques [12]. In broad terms, the image is partitioned into regions, of which initially there are as many as there are pixels in the image. The segments then grow from these regions by the aggregation of adjacent regions whose average spectral characteristics do not differ from each other beyond a prespecified tolerance. Regions to be merged must also be closest to each other, in terms of their average spectral characteristics, among all the regions adjacent to them. The choice of the tolerance is critical, as it may lead to an excessive fragmentation of the image (if the tolerance to join regions is too strict), or to segments that are not spectrally homogeneous enough to be of significant help in the subsequent classification (if the tolerance is too loose). This classification constitutes the second phase, during which each segment is classified into one or more of the thematic categories forest (F), savannah (S), water (W), deforested area (D), cloud (C), and shadow (Sh). The categories F, S, W, and D embody all the relevant information to be monitored, and are referred to as *basic categories*. The remaining categories, C and Sh, called *interfering categories*, are needed to allow for the interference caused by clouds and shaded areas in the classification process.

The classification of segments into these categories follows a fuzzy-logic approach [13–17], that is, a segment may belong to multiple categories

with partial degrees of membership. In this way, it is possible to model transition phenomena (e.g., the recovery of a deforested area) and uncertainties. When one of the categories involved is an interfering category, then it is possible for a segment to belong fully to more than one category (e.g., a tiny, opaque cloud surrounded by a large forest suggests the full membership of the cloudy segment in both C and F, as there is little or no uncertainty as to what lies under the cloud).

Our two-phase method (image segmentation followed by segment classification), together with the fuzzy criterion adopted to describe the degree of membership of each segment in each of the six possible categories, endows our approach with distinctive features that make it stand out from the traditional, pixel-oriented approaches. The first of these features is that a region-oriented approach resembles much more closely the approach taken by human photointerpreters. The classification provided by these experts is the best available standard against which to compare the result of automatic classifiers (ruling out, of course, the *in loco* inspection of representative portions of the Amazon region), so this resemblance may well be exploited. Secondly, region-oriented approaches allow the incorporation of geometric, textural, and contextual information into the classification process. As we remarked previously, such additional sources of information are used widely by human photointerpreters, a clear indication that the classification task should not be attempted solely on a spectral basis. The third important feature is the already mentioned ease with which interference and transition phenomena can be captured with the aid of partial degrees of membership in the various categories. In fact, this fuzzy membership in multiple categories can be regarded as the possibility of having countless (or a great many, as the degrees of membership are in practice discrete) categories to which segments may belong fully and exclusively.

The results we have obtained with this approach appear to be superior to what can be achieved with more traditional approaches, although, as we shall see later on, various improvement directions can be considered. What we have found as a result of extensive experimentation on representative images of the Amazon region is that the use of some forms of textural and contextual information in addition to simple spectral data tends to lead to better results than the use of the simple spectral data alone. We have also found that the addition of geometric information on top of simple spectral data tends to be of no help, but that is probably to be accounted for by imperfections during the segmentation phase.

Related work employing a combination of segmentation and classification techniques can be found in two other publications that we know of. One of them reports on a preliminary version of our own system [18]. The

other describes a similar approach developed independently for the analysis of SPOT images, aiming at their crisp (as opposed to fuzzy) classification from textural data [19].

This paper is divided into four additional sections. Section 2 contains the architectures of our classifying system, and in Section 3 we discuss the various features, called *segment descriptors*, used to characterize each of the segments for classification. Experimental results are given in Section 4, and Section 5 contains concluding remarks.

2. SYSTEM ARCHITECTURES

The architectures of our classifying system are centered around a neural network comprising feedforward connections only, which we shall refer to as the *core network*. Each segment is presented to the core network for classification as a collection of segment descriptors, of which some describe spectral, geometric, and textural characteristics of the segment, and others are related to neighboring segments in the image, accounting for some contextual information for that segment. For each segment S, the core network outputs a vector in $[0, 1]^6$ whose components are to be interpreted as the degrees of membership of S in each of the six categories. All the segment descriptors will be explained in detail in Section 3. However, those related to neighboring segments, called *neighborhood descriptors*, are of special relevance for the design of the classifying system, and are for this reason introduced at this earlier stage in our discussion (although details are left to Section 3 as well).

For a segment S, let $\mathcal{N}(S)$ denote the set of segments which are neighbors of S in the image, and $\xi(S)$ denote the classification of S as given by the core network. The neighborhood descriptors of S are functions of $\xi(S')$ for all $S' \in \mathcal{N}(S)$. Clearly, this definition introduces a cyclic dependence of the classification of a segment upon the classification of its neighbors, and creates the need for two special mechanisms to be introduced in our classifying system. The first is a relaxation scheme whereby segments are presented to the core network for classification, possibly many times (at least once), until the classification of no segment deviates significantly from its previous classification. The second mechanism is needed to provide initial conditions for this relaxation scheme. In our system, these initial conditions are provided by another neural network with feedforward connections, called the *startup network*, to which every segment is presented exactly once at the beginning of the classification process for a preliminary classification based solely on spectral, geometric, and textural information. It should now be noted, for the sake of greater

precision, that the classifications of a segment's neighbors used to compute its neighborhood descriptors may also have been given by the startup network.

The resulting system architecture is given next in the form of an algorithm, called CLASSIFY($\mathscr{S}$), where $\mathscr{S}$ is a set of segments of an image to be classified. For the sake of presenting the algorithm, we regard the set $\mathscr{S}$ as being implemented by a list, which we refer to as $\mathscr{S}$ as well. This algorithm starts off by presenting each segment in $\mathscr{S}$ to the startup network for a preliminary classification, and then iterates by employing the core network to classify the segments until no segment is given a classification sufficiently different from its previous one. Along the iterations, segments are picked for classification from the head of $\mathscr{S}$. If the classification of a segment deviates significantly from its previous classification, then its neighbors are added to the tail of $\mathscr{S}$ (if not already there) for another classification by the core network. The computations carried out by the startup network and the core network on a segment S are represented, respectively, by the functions STARTUP(S) and CORE(S). Two successive classifications $\xi(S)$ and $\xi'(S)$ of a segment S will be considered significantly different from each other if the Euclidean distance between $\xi(S)$ and $\xi'(S)$ in $\mathbf{R}^6$ exceeds a tolerance $\varepsilon_1 > 0$.

```
CLASSIFY(𝒮):
  for S in 𝒮 do
    ξ(S) := STARTUP(S);
  k := 0;
  while 𝒮 is not empty and k < K do
    begin
      Remove S from 𝒮;
      ξ'(S) := CORE(S);
      if {Σ_{i=1}^{6} [ξ_i(S) − ξ_i'(S)]²}^{1/2} > ε_1 then
        begin
          ξ(S) := ξ'(S);
          for S' ∈ 𝒩(S) do
            if S' is not in 𝒮 then
              Add S' to 𝒮
        end;
      k := k + 1
    end.
```

The relaxation mechanism embedded in algorithm CLASSIFY($\mathscr{S}$) is not guaranteed to stabilize, that is, reach a situation in which no segment is classified by the core network significantly differently from its previous classification. For this reason, a maximum number of classifications by the core network, denoted by K in CLASSIFY($\mathscr{S}$), is employed as an alternative termination criterion.

As we already remarked, both the startup and the core networks operate on segment descriptors that include descriptors of the geometric features of a segment. These segment descriptors will be discussed in detail in Section 3, but need to be addressed now as well, as they motivate modifications in our system architecture. Descriptors of geometric features are important because deforested areas usually have polygonal boundaries or boundaries with a predominating polygonal section. In addition, such areas may also contain off-boundary rectilinear features indicating the presence of artificial structures. However, as we remarked in Section 1, when the image is segmented during the first phase of our two-phase classification scheme, the resulting segments may be excessively fragmented as a result of the tradeoff that exists between image fragmentation and segment homogeneity. As a consequence, very often a deforested area gives rise to numerous segments, and most of the geometric regularity present in the area's boundary is not reflected in the contours of most of the resulting segments.

One possible attempt at circumventing this problem is to try and coalesce neighboring segments into larger segments so that the original deforested areas are mostly recovered and so the geometric characteristics of their boundaries appear in the resulting segment and are once again meaningful. This possibility is realized in the algorithm that we give next, called COALESCE($\mathscr{S}$), which transforms the set of segments $\mathscr{S}$ originally produced by the segmentation phase into a set $\mathscr{S}'$ with (expectedly) fewer segments, which can then be classified by CLASSIFY($\mathscr{S}'$) while hopefully taking full advantage of the geometric characteristics of the segments' contours.

COALESCE($\mathscr{S}$) relies on a neural network with feedforward connections to perform a preliminary classification of the segment in $\mathscr{S}$ based solely on the segments' spectral and textural characteristics. This neural network is referred to as the *alternate startup network*, and the computation that it carries out on a segment S is represented by the function ALTERNATE_STARTUP(S). Neighboring segments are then coalesced pairwise into a new segment that replaces them in $\mathscr{S}$ as long as their classifications are sufficiently close to each other. Closeness is determined in this case by checking whether the degrees of membership of the two segments in each of the six categories do not differ from each other beyond a certain tolerance. [Note that, in analogy with algorithm CLASSIFY($\mathscr{S}$), we might judge the closeness of two neighboring segments' classifications by the Euclidean distance between them in $\mathbf{R}^6$. Our choice has in this case been dictated by the outcome of experiments on how well the two criteria lead the coalescing process to perform.] The new segment is classified by the neural network as it enters $\mathscr{S}$. Whenever a segment cannot be coalesced to any of its neighbors, it is removed from $\mathscr{S}$ and added to $\mathscr{S}'$. The

criterion to terminate the algorithm is the emptiness of $\mathscr{S}$. The largest difference allowed for each category between the classifications of two segments for them to be coalesced into a single segment is determined by a tolerance $\varepsilon_2 > 0$. As in algorithm CLASSIFY($\mathscr{S}$), in COALESCE($\mathscr{S}$) the set $\mathscr{S}$ is treated as a list of segments.

```
COALESCE(𝒮):
  for S in 𝒮 do
    ξ(S) := ALTERNATE_STARTUP(S);
  while 𝒮 is not empty do
    begin
      Remove S from 𝒮;
      if there exists S' ∈ 𝒩(S)
      such that |ξ_i(S) − ξ_i(S')| ≤ ε_2 for 1 ≤ i ≤ 6 then
        begin
          Remove S' from 𝒮;
          Coalesce S and S' into S'';
          Add S'' to 𝒮;
          ξ(S'') := ALTERNATE_STARTUP(S'')
        end
      else
        Add S to 𝒮'
    end.
```

Implicit in the functioning of COALESCE($\mathscr{S}$) is the fact that the segment S'' obtained from the coalescence of S and S' inherits the neighbors of those two segments (except of course S and S' themselves). Conversely, these neighbors no longer have S or S' as neighbors, but S'' instead.

3. SEGMENT DESCRIPTORS

The satellite images used by our classifying system are produced by the Landsat Thematic Mapper (TM) in six spectral bands [20]. (Actually, there are seven TM bands, but the one occupying the thermal wavelength is known to be ineffective for discrimination purposes like ours.) In this section we describe the segment descriptors employed by our system. These descriptors come in four types: *spectral descriptors*, *geometric descriptors*, *textural descriptors*, and the already introduced neighborhood descriptors. Spectral and textural descriptors depend directly on the information contained in the image in the six TM bands, while the geometric descriptors depend on the process of image segmentation and the algorithm for segment coalescence, and the neighborhood descriptors on the algorithms for classification.

For each TM band the image is viewed as an $m \times n$ matrix I, where m is the number of pixel rows of the image and n its number of pixel columns. For $1 \leq i \leq m$ and $1 \leq j \leq n$, the element I_{ij} of the matrix gives the so-called *gray level* of the corresponding pixel in that band. We assume throughout that gray levels are integers in the set $\{0, \ldots, L-1\}$. Note, in addition, that whenever possible the matrix notation we have introduced is used without explicit mention to the particular band under consideration.

Before introducing the details of the descriptors in each of the four classes, we present a brief summary of them. The spectral descriptors of a segment are the gray-level averages, taken over all the pixels of the segment, in each of the six Landsat TM bands. The textural descriptors for each TM band are of two types, one based on the gray-level variance, and the other encompassing four functions of the *cooccurrence matrix* for the segment [1, p. 506]. These functions yield the segment's *angular second momentum*, *contrast*, *entropy*, and *correlation*. Geometric descriptors, in turn, come in two types, one involving a number of *Fourier descriptors* of the segment's contour (represented by their magnitudes, for invariance), the other comprising a "rectitude" indicator based on the *Hough transform* of the image in selected TM bands [1, p. 432].

Each segment has six neighborhood descriptors, one for each category, indicating the average degree of membership of the segment's neighbors in that category. This average is weighted by the length of perimeter (given in number of pixels) that the two segments share.

3.1. Spectral Descriptors

For each TM band, the spectral descriptor of a segment S, denoted by $s(S)$, is simply the gray-level average of the pixels in S. If we let the notation $(i, j) \in S$ stand for the fact that the pixel in row i and column j of I is in segment S, then we have

$$s(S) = \frac{1}{|S|} \sum_{(i,j) \in S} I_{ij},$$

where $|S|$ is the number of pixels in S.

3.2. Geometric Descriptors

There is a great variety of geometric descriptors that can be used to characterize image segments. Some are of quite simple nature, as for example a segment's *area* (number of pixels), *perimeter* (number of contour pixels), *compactness* (some measure decreasing in the segment's perimeter and increasing in its area), *eccentricity* (the ratio of the segment's

largest to its smallest axis), and *connectedness* (generally related to the number of "holes" in the segment) [1, p. 483]. In the case of our classification problem, what seems to be needed is descriptors that can indicate the presence of polygonal contours and more generally the presence of rectilinear structures in the segment. Our choice has been to adopt Fourier descriptors for the detection of polygonal contours, and *Hough descriptors*, to be defined shortly, for the more general detection of straight lines in the segment's contour or in its interior. It should be recalled, however, that the success of the Fourier descriptors is expected to be intimately related to the ability of algorithm COALESCE($\mathcal{S}$) in producing segments (from those in $\mathcal{S}$) that correspond to regions of meaningful geometry.

The Fourier descriptors for a segment are obtained from a sequence of contour pixels of that segment by computing the discrete Fourier transform (DFT) of the sequence, with each pixel being regarded as a point in the complex plane. With $j = \sqrt{-1}$, let this sequence comprise b pixels and be denoted by $\langle x_0 + jy_0, \ldots, x_{b-1} + jy_{b-1} \rangle$. The DFT of this sequence is another sequence of b points in the complex plane, denoted by $\langle X_0 + jY_0, \ldots, X_{b-1} + jY_{b-1} \rangle$, such that

$$X_k + jY_k = \frac{1}{b} \sum_{i=0}^{b-1} (x_i + jy_i) \exp(-j2\pi ki/b)$$

for $0 \leq k \leq b - 1$. The original sequence can be recovered quite easily by computing the inverse DFT, which always exists.

The sequence obtained by the DFT has several properties of great appeal in our context. First, any translation of the segment within the complex plane affects X_0 and Y_0 only. Second, if the segment is rotated in the complex plane or the DFT is computed starting at a different point in the sequence, then only the phases of the resulting points are affected, not their magnitudes. Third, so long as the points are used in the DFT computation in the same order as they appear in the sequence, then the magnitude of $X_1 + jY_1$ is greatest among those of all the points, except possibly $X_0 + jY_0$. If only the magnitudes of $X_1 + jY_1$ through $X_{b-1} + jY_{b-1}$ are employed as Fourier descriptors, then position and rotation invariance are achieved. If in addition these magnitudes are normalized to the magnitude of $X_1 + jY_1$, then scale invariance is achieved as well.

Then given a segment S and a sequence of points $x_0 + jy_0$ through $x_{b-1} + jy_{b-1}$ on its contour, we define $b - 2$ Fourier descriptors, given by

$$f(S, k) = \frac{(X_k^2 + Y_k^2)^{1/2}}{(X_1^2 + Y_1^2)^{1/2}}$$

for $2 \leq k \leq b - 1$. Note that $f(S,0)$ and $f(S,1)$ could be defined likewise. However, they would be useless, because $f(S,0)$ is not invariant under translation and scale, and $f(S,1)$ is necessarily equal to 1.

Two remaining points to be discussed in connection with the Fourier descriptors are the selection of the value of b and of the b contour points. For the sake of computational efficiency when calculating the Fourier descriptors, it is advisable to utilize the fast Fourier transform (FFT) algorithm, which in its radix-2 variant then imposes on b the requirement that it has to be a power of two. Another requirement is that the b contour points be selected so that they delimit approximately rectilinear portions of the contour, so that not only can the inverse DFT recover the original points, but also it can recover a good approximation of the contour. These two requirements on the selection of contour points are conflicting, as not every segment allows a power-of-two number of contour points to delimit approximately rectilinear portions if the criterion for rectitude is not flexible. Our approach is then the following. Points are chosen on the segment's contour to delimit approximately rectilinear portions (employing the algorithm given in [21]). Additional points are then picked on the middle of some of these portions of the contour in order to fill up a power-of-two number of points. After the DFT has been computed, the 16 Fourier descriptors $f(S,2)$ through $f(S,17)$ are used (note that our neural networks must have a fixed number of inputs reserved for Fourier descriptors; 16 is the fixed number we have chosen). If our power-of-two number is less than 18, then the unavailable Fourier descriptors are taken to be zero (although we have not given details, this is consistent with the correctness of the inverse DFT). If the number is greater than 18, then the excess Fourier descriptors are ignored (meaning that the inverse DFT on the 18 descriptors would only be able to recover a less accurate approximation of the contour).

The Hough transform provides the basis for the establishment of our Hough descriptors. Essentially, the Hough transform allows straight lines in an image to be detected within certain limits, and our Hough descriptors for a given segment are a measure of how much of the rectitude present in the image is associated with that segment.

The Hough transform is based on the idea that a straight line $y = a'x + b'$ in the x-y space corresponds to a point in the a-b space such that $a = a'$ and $b = b'$, and that through this point pass as many straight lines as there are points on the line $y = a'x + b'$. The goal of the Hough transform is to compute, for all values of a and b (within an appropriate quantization of the a-b space), the number of pixels in the x-y space found to lie on the straight line $y = ax + b$. In order for a pixel to be taken as lying on some straight line, the gradient of the image for the TM band under consideration is computed at all pixels, and checked against a certain threshold.

Pixels where the gradient exceeds this threshold are taken as contributing to a straight line. A practical implementation of this procedure employs a grid of accumulators on the *a-b* space, which are initialized to zero and incremented whenever pixels are found to be lying on the corresponding straight lines. Accumulators exceeding another threshold indicate prominent straight lines in that TM band. In this paper we have employed a computationally efficient algorithm for the Hough transform known as the *binary Hough transform* [22, 23].

After the Hough transform has been computed for the entire image on a given TM band, let $l(S)$ denote the number of pixels of the segment S found to be lying on straight lines. Our Hough descriptor for S in that band, denoted by $h(S)$, is then given by

$$h(S) = \frac{l(S)}{|S|}.$$

3.3. Textural Descriptors

For a given TM band and $0 \leq z \leq L - 1$, let $p(S, z)$ denote the number of pixels (i, j) of a segment S such that $I_{ij} = z$. The *gray-level variance* of S, denoted by $\sigma^2(S)$, is given by

$$\sigma^2(S) = \sum_{z=0}^{L-1} [z - s(S)]^2 \frac{p(S, z)}{|S|},$$

where $s(S)$ is the gray-level average of S (cf. Section 3.1). The first textural descriptor we utilize for the segment S is denoted by $t_1(S)$ and given as a function of the gray-level variance of S as

$$t_1(S) = 1 - \frac{1}{1 + \sigma^2(S)}.$$

If every pixel in S has the same gray level, then $t_1(S) = 0$. As the gray levels of the pixels in S become more and more varied, and consequently $\sigma^2(S)$ increases, then $t_1(S)$ approaches 1.

The other textural descriptors that we utilize attempt to take into account the relative positions of the pixels inside S, and are based on the already mentioned cooccurrence matrix for S. For an integer $d \geq 1$ and an angle $\theta \in [0, \pi)$, the cooccurrence matrix for S, denoted by $C(S, d, \theta)$, is aimed at capturing the frequencies of occurrence in S of pairs of pixels with certain gray-level characteristics that are $d - 1$ pixels apart in S along direction θ. This matrix has L rows and L columns, and for

$0 \le z_1, z_2 \le L - 1$ its element in row z_1 and column z_2 is given by

$$C_{z_1 z_2}(S, d, \theta) = \frac{A_{z_1 z_2}(S, d, \theta)}{\sum_{x=0}^{L-1} \sum_{y=0}^{L-1} A_{xy}(S, d, \theta)},$$

where $A_{z_1 z_2}(S, d, \theta)$ is the number of times that pairs of pixels in S, one of gray level z_1 and the other of gray level z_2, are positioned with respect to each other so that they are $d - 1$ pixels apart along direction θ. Clearly, $C(S, d, \theta)$ is symmetrical with respect to its main diagonal, and $C(S, d, \theta) = C(S, d, \theta + \pi)$ (whence the limits imposed earlier on the variation of θ).

One difficulty often encountered with the several possible uses of the cooccurrence matrix is that it is normally calculated for entire images, which is computationally demanding. When it is computed on typically small image segments, however, this difficulty tends to be considerably smaller.

Various texture descriptors based on the cooccurrence matrix have been proposed and evaluated in a variety of contexts [24]. What has been found is that small values of d (usually in $\{1, 2, 3\}$) and values of θ that are multiples of $\pi/4$ (i.e., in $\{0, \pi/4, \pi/2, 3\pi/4\}$) tend to be the most effective choices [25, 26]. We have selected four of the most successful such functions, and adopted $d = 1$ and $\theta \in \{0, \pi/4, \pi/2, 3\pi/4\}$. These functions are the already mentioned angular second momentum (also sometimes referred to as *energy* or *uniformity*), contrast (often referred to as *inertia*), entropy, and correlation. Given a TM band, we denote these functions for a segment S and θ in the appropriate range respectively by $t_2(S, \theta)$, $t_3(S, \theta)$, $t_4(S, \theta)$, and $t_5(S, \theta)$, given below:

Angular second momentum:

$$t_2(S, \theta) = \sum_{z_1=0}^{L-1} \sum_{z_2=0}^{L-1} C_{z_1 z_2}^2(S, 1, \theta).$$

Contrast:

$$t_3(S, \theta) = \sum_{z_1=0}^{L-1} \sum_{z_2=0}^{L-1} (z_1 - z_2)^2 C_{z_1 z_2}(S, 1, \theta).$$

Entropy:

$$t_4(S, \theta) = -\sum_{z_1=0}^{L-1} \sum_{z_2=0}^{L-1} C_{z_1 z_2}(S, 1, \theta) \ln C_{z_1 z_2}(S, 1, \theta).$$

Correlation:

$$t_5(S,\theta) = \sum_{z_1=0}^{L-1}\sum_{z_2=0}^{L-1} \frac{[z_1 - \mu_1(S,\theta)][z_2 - \mu_2(S,\theta)]C_{z_1z_2}(S,1,\theta)}{\sigma_1(S,\theta)\sigma_2(S,\theta)},$$

where

$$\mu_1(S,\theta) = \sum_{z_1=0}^{L-1} z_1 \sum_{z_2=0}^{L-1} C_{z_1z_2}(S,1,\theta),$$

$$\mu_2(S,\theta) = \sum_{z_1=0}^{L-1}\sum_{z_2=0}^{L-1} z_2 C_{z_1z_2}(S,1,\theta),$$

$$\sigma_1^2(S,\theta) = \sum_{z_1=0}^{L-1} [z_1 - \mu_1(S,\theta)]^2 \sum_{z_2=0}^{L-1} C_{z_1z_2}(S,1,\theta),$$

$$\sigma_2^2(S,\theta) = \sum_{z_1=0}^{L-1}\sum_{z_2=0}^{L-1} [z_2 - \mu_2(S,\theta)]^2 C_{z_1z_2}(S,1,\theta).$$

Each of these functions of the cooccurrence matrix is aimed at capturing a particular property of the arrangement of pixels in the segment. Although their detailed behavior is not immediate to grasp, some trends can be inferred rather easily. For example, $t_2(S,\theta)$ is minimum when all the elements of $C(S,1,\theta)$ have the same value, while under the same circumstances $t_4(S,\theta)$ is maximum and $t_5(S,\theta)$ is zero. Similarly, if the largest elements of $C(S,1,\theta)$ tend to concentrate near the matrix's main diagonal, then $t_3(S,\theta)$ acquires a relatively low value and $t_5(S,\theta)$ tends to its maximum.

3.4. Neighborhood Descriptors

Given any two segments S, S', let $b(S,S')$ denote the number of pixels in the contour of S that are adjacent to at least one pixel of S'. Clearly, $b(S,S)$ yields the perimeter of S, and $b(S,S')$ is equal to zero if $S' \notin \mathcal{N}(S)$.

Following our earlier discussion, each segment has six neighborhood descriptors, one for each of the six categories of interest to our classification problem. For $1 \le i \le 6$ and a segment S, the neighborhood descriptor that corresponds to the ith category is denoted by $n_i(S)$ and given by the weighted average of the degrees of membership of the neighbors of S in that category according to their current classifications [cf. algorithm CLASSIFY($\mathcal{S}$)]. The weight to be used for each neighbor S' is the ratio

$b(S, S')/b(S, S)$, that is, the fraction of S's perimeter that is adjacent to S'. In summary,

$$n_i(S) = \sum_{S' \in \mathcal{N}(S)} \frac{b(S, S')}{b(S, S)} \xi_i(S'),$$

where $\xi(S')$ is the current classification of S'.

4. RESULTS

In this section we present the results of a variety of experiments which were conducted to assess the relative merits of the segment descriptors described in Section 3, as well as to evaluate the performance of the algorithms introduced in Section 2. All the neural networks employed in those algorithms [the startup network and the core network of algorithm CLASSIFY($\mathcal{S}$), and the alternate startup network of algorithm COALESCE($\mathcal{S}$)] are obtained in this section by the back-propagation training algorithm, incorporating various features selected via early experimentation regarding the choice of transfer functions for the neurons and the adaptive selection of step sizes during the gradient descent in the training phase [27, p. 124].

All neural networks have one single layer of hidden neurons, and each network is structured as six independent modules, one for each category, sharing only the inputs (i.e., the segment descriptors). Each module has a single output neuron and its own private set of hidden neurons. This arrangement into independent modules appears to be generally preferable for multiple-category problems [28], as was in our case demonstrated by early experimentation as well.

Approximately 17,000 segments were used in our experiments, divided into approximately two-thirds to constitute the *training set* and the remaining one-third to constitute the *test set*. These segments were obtained from five representative images of the Amazon region, chosen to contain a great variety of the possible situations, including many of great complexity. These five images come from the states of Acre, Mato Grosso, Pará, and Rondônia, and from around Tucuruí dam.

All segments were previously classified by a photointerpreter into the six categories, using (for simplicity) only five possibilities (0, 0.25, 0.5, 0.75, and 1) for the degrees of membership in each of the categories. This classification was subject to the constraints that the degrees of membership of a segment in the basic categories must add up to one, and that its degrees of membership in all six categories must add up to some number between one and two. However, on a few occasions the photointerpreter was

allowed to disobey these constraints. This happened for segments that were erroneously kept from being split into more segments during the segmentation phase, and for which the photointerpreter felt compelled to assign degrees of membership in the basic categories adding to more than one. These segments are referred to as *anomalous* segments.

As we remarked in Section 1, the fuzzy membership of each segment in multiple categories allows for the possibility of a very large number of compound categories from the standpoint of full (and exclusive) membership. However, not all of these categories are available for use by the photointerpreter in his classification, owing to the constraint that he is only allowed five distinct degrees of membership in each class. What this amounts to is that the number of possible compound categories indirectly available to the photointerpreter is approximately 600. There still seem to be plenty of possibilities, as all the segments extracted from the five representative images turned out to require only 35 different compound categories when classified by the photointerpreter. We shall henceforth let $\bar{\xi}(S)$ denote the photointerpreter's classification of segment S.

Before we proceed, we should once again make the important remark that this classification of segments by a photointerpreter is the best available source of training data and the best available standard against which the performance of automatic classifiers can be checked. Surely the use of *in loco* observations during field trips would be better, but this is of course beyond all sorts of practical limitations in most cases. We shall return to this point with additional comments in Section 5.

The segments thus classified by the photointerpreter were then divided into the training set and the test set by a semiautomatic procedure designed to ensure that every one of the six categories would be represented in both sets by segments of relatively large area, and that neighboring segments would be cast into the same set as much as possible. The functioning of this procedure goes as follows. First all segments S such that $\bar{\xi}_i(S) = 1$ for the ith category only ($1 \leq i \leq 6$) and considered to be "large enough" are distributed between the training set and the test set in the proportion of two to one per category. Then the neighbors of segments already in the training set are added to this set until two-thirds of the entire population of segments is exhausted. The remaining segments are taken to complete the one-third of test-set segments. It is not hard to anticipate that the possibility that neighboring segments belong to different sets can be a source of difficulties when neighborhood descriptors are used. We shall return to this issue shortly, when discussing the use of those descriptors.

A very welcome by-product of the structuring of neural networks as collections of independent modules (one per category) is that each module can be trained separately. Not only is this attractive from a purely

computational standpoint (smaller modules can be trained faster, and the various modules can be spread out for training over a network of workstations or the processors of a parallel-processing machine), but also it allows the training set to be especially calibrated to provide the most efficient training for each module. Specifically, each module can be trained on an extension of the training set obtained for that module by the unbiased replication of some of its members. This replication is aimed at correcting the uneven distribution of examples concurring toward and against establishing the membership in the category of the module being trained. In our case, when training a module for the ith category, $1 \leq i \leq 6$, segments were replicated in such a way that the sums, over segments S in the training set extension, of $\bar{\xi}_i(S)$ and of $1 - \bar{\xi}_i(S)$ were roughly the same.

Another type of segment replication has been employed, but only for the modules corresponding to category D. This has proven necessary because the images we selected were all taken from a time of the year when the distinction between categories D and S is extremely difficult, as both of them appear spectrally very similar in the images. The result of the aforementioned replication of segments to train category-D modules turned out to leave too few segments concurring against this category in category S, and then we had to unbalance the replication a little so that those segments became more numerous. This additional replication has endowed category-D modules with a considerably enhanced ability to discriminate between the problematic categories.

As one last remark concerning this replication of segments for training, it is worth mentioning that no actual replication need take place. All we need is a means of providing different weights to the various segments as they are presented to the neural network for training, so that some measure of the relative "importance" of each segment can be conveyed. As it turns out, the use of such weights is a straightforward matter in the case of most traning algorithms, including the one we employ in the sequel.

In the remainder of this section we provide experimental data on the startup network and the core network, first on the original set of segments and then on the set of segments coalesced with the aid of the alternate startup network. In order to obtain these data, we have employed a set of four performance measures: the *mean squared error* (MSE), the *hit ratio*, the *sensitivity*, and the *specificity*.

For a set of segments $\mathscr{S}$, the MSE, denoted by $\text{mse}(\mathscr{S})$, is given by

$$\text{mse}(\mathscr{S}) = \frac{1}{2|\mathscr{S}|} \sum_{S \in \mathscr{S}} \sum_{i=1}^{6} \left[\xi_i(S) - \bar{\xi}_i(S) \right]^2,$$

where $\xi(S)$ is the output of a classifier on segment $S \in \mathscr{S}$.

The hit ratio depends on the definition of what a *correct* classification is, which is far from consensual in view of the fuzzy character of our classification scheme. In this paper we take a classification $\xi(S)$ of a segment S to be correct or not, relative to a threshold τ, $0 \leq \tau < 1$, according to the outcome of the following simple steps. First mark the categories corresponding to those components of $\bar{\xi}(S)$ which are greater than τ. Let N be the number of marked categories. If $N = 0$, then say that $\xi(S)$ is correct if none of its components is greater than τ either. If $N > 0$, then check whether the categories corresponding to the N largest components of $\xi(S)$ are the same as the marked categories. Say that $\xi(S)$ is correct in the affirmative case. The classification $\xi(S)$ is said not be correct in all other situations. The hit ratio for a set of segments $\mathscr{S}$ is then defined to be the fraction of $|\mathscr{S}|$ corresponding to segments $S \in \mathscr{S}$ such that $\xi(S)$ is correct.

The definition of a correct classification can be changed slightly so that only the components belonging to a certain group of categories is looked at during the steps we just outlined. This allows us to treat, in the context of this paper, the correctness of a classification with respect to the group of basic categories, regardless of the interfering categories, and conversely. In addition, it should be noted that, as long as $0.5 < \tau < 1$ and a segment S is not anomalous, then necessarily $N = 1$ if we restrict our correctness analysis to the basic categories only. This allows us to compute the hit ratio for a specific category as the fraction of the number of segments the photointerpreter classified as belonging to that category corresponding to segments which were classified correctly by the automatic classifier. Likewise, we may now split the corresponding fraction of incorrectly classified segments among the remaining basic categories to see in which direction the classifier erred. The same is valid, of course, if we restrict ourselves to the group of interfering categories. Similarly, it is possible to evaluate the classifier's ability to infer in the presence of interfering categories. This can be achieved by computing the hit ratio for a specific basic category as the fraction of the number of segments the photointerpreter classified as belonging to that category and to one of the interfering categories corresponding to segments which were classified as belonging to that basic category by the automatic classifier.

The other two performance measures, sensitivity and specificity, are also dependent upon the threshold τ, and should be taken in the context of a specific category, say the ith, for $1 \leq i \leq 6$. For a set of segments $\mathscr{S}$, let $\mathscr{S}_1 \subseteq \mathscr{S}$ be such that $\bar{\xi}_i(S) > \tau$ for all $S \in \mathscr{S}_1$ and $\mathscr{S}_2 = \mathscr{S} - \mathscr{S}_1$. The sensitivity of a classifier to the ith category on the set $\mathscr{S}$ is the fraction of $|\mathscr{S}_1|$ corresponding to those segments $S \in \mathscr{S}_1$ such that $\xi_i(S) > \tau$, where $\xi_i(S)$ is the classification of S as given by the classifier. Similarly, the

specificity of the classifier to the ith category on the set $\mathscr{S}$ is the fraction $|\mathscr{S}_2|$ corresponding to those segments $S \in \mathscr{S}_2$ such that $\xi_i(S) \leq \tau$.

All four performance measures can have their definitions extended to refer to the areas of the segments, as opposed to the number of segments. In this way, the definition of the MSE can be extended to that of the *area mean squared error* (AMSE), which for a set of segments $\mathscr{S}$ is denoted by amse($\mathscr{S}$) and given by

$$\text{amse}(\mathscr{S}) = \frac{1}{2\Sigma_{S \in \mathscr{S}}|S|} \sum_{S \in \mathscr{S}} |S| \sum_{i=1}^{6} \left[\xi_i(S) - \bar{\xi}_i(S) \right]^2,$$

where ξ(S) is the output of a classifier on segment $S \in \mathscr{S}$ and $|S|$ is the number of pixels in S (its area). Likewise, the other three measures can be extended in the straightforward way by considering the classifications of individual pixels.

4.1. Results on the Original Segments

Figures 1 through 6 show the evolution of the MSE during experiments to determine which of the segment descriptors given in Section 3 are to be used as inputs to the startup network. The MSE evolution in these figures is relative to the training set, enlarged for each module as explained previously. An *epoch* in these figures (and henceforth) refers to an iteration of the back-propagation training algorithm during which all the examples in the training set contribute to the computation of the descent direction. Also, a *base* set of descriptors is henceforth taken as the set of twelve descriptors comprising, for a segment S, the spectral descriptor $s(S)$ and the textural descriptor $t_1(S)$ for each of the six TM bands. As we recall, these two types of descriptors embody information on the segment's gray-level average and variance. In Figures 1 through 4, each module contains six hidden neurons.

Figures 1 and 2 show, respectively for the modules of categories F and S, a comparison of the base set of descriptors and enlargements of this set to encompass each of the other textural descriptors, $t_2(S, \theta)$ through $t_5(S, \theta)$ for a segment S and an angle $\theta \in \{0, \pi/4, \pi/2, 3\pi/4\}$, all based on the segment's cooccurrence matrix for $d = 1$ and the corresponding value of θ. These four functions yield the segment's angular second momentum, contrast, entropy, and correlation, respectively. Each of these enlargements adds to the base set twelve other descriptors, all corresponding to one of the four functions. These twelve additional descriptors include one for each of the possible angles and each of the three TM bands 3, 4, and 5 (texture measures on the other three bands are usually ineffective).

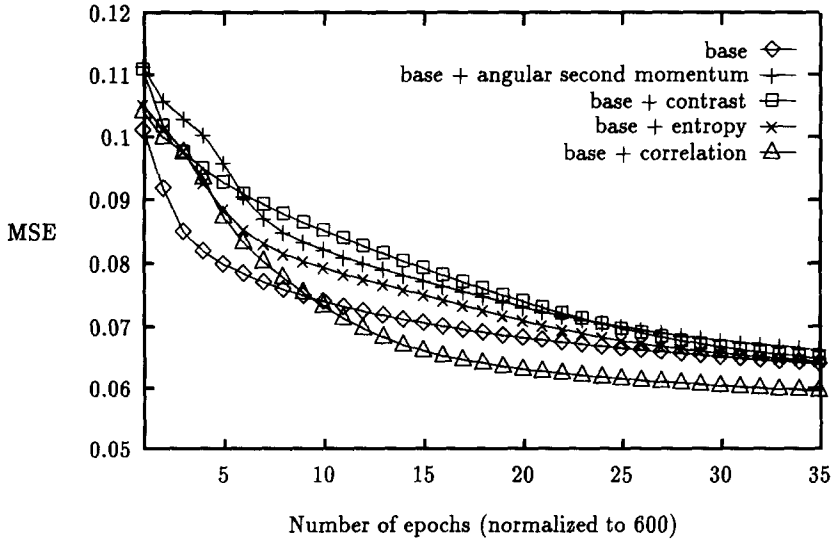

Figure 1. Effects of textural descriptors on category-F modules.

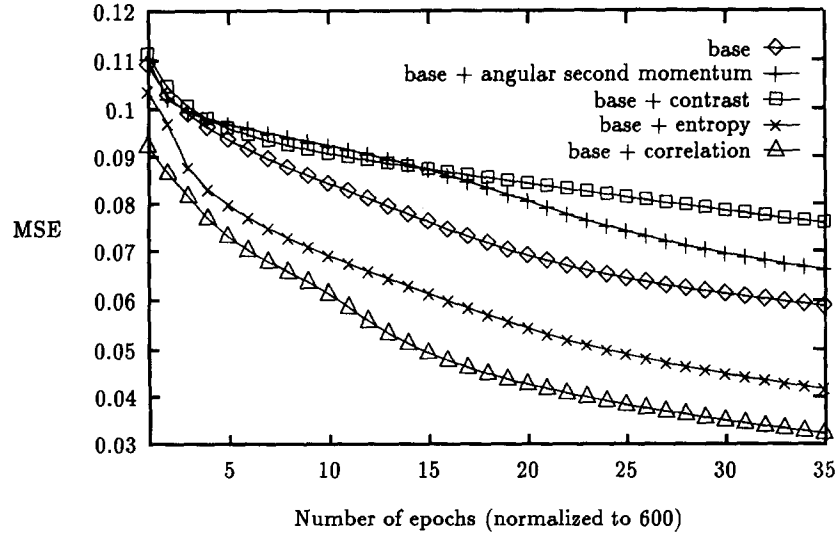

Figure 2. Effects of textural descriptors on category-S modules.

Figures 1 and 2 are extremes of the typical behavior that occurred for the other modules as well. What they indicate (Figure 2 does this with special clarity) is that the use of the entropy or of the correlation as an additional segment descriptor is an improvement over the use of the base set of descriptors alone. The other two texture measures (angular second momentum and contrast), on the other hand, do not typically improve the network's behavior.

Figure 3 and 4 show the effect of enlarging the base set, for a segment S, by the sixteen Fourier descriptors $f(S, 2)$ through $f(S, 17)$ or the Hough descriptor $h(S)$ for TM band 5 (Hough descriptors for the other TM bands were found by visual inspection to be too ineffective to be considered). These two figures correspond respectively to category-S and category-D modules, which as we remarked tend to be spectrally very similar, and in whose discrimination geometric descriptors can be expected to provide some aid, given the typical polygonal contours of category-D segments. However, as we anticipated earlier in this paper, the excessive fragmentation caused by the segmentation phase is such that most of the geometric information present in the contour of a deforested area is lost. As a result, the use of geometric descriptors does not appear to add to the discriminatory power of the descriptors in the base set.

Having identified the entropy and correlation texture measures as potentially good additions to the base set of segment descriptors, we set out at last to check the effect of employing both texture measures together,

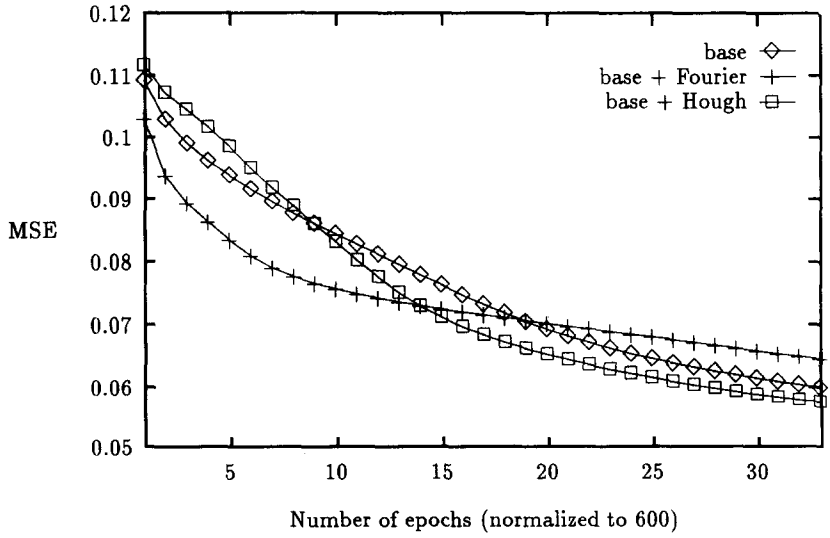

Figure 3. Effects of geometric descriptors on category-S modules.

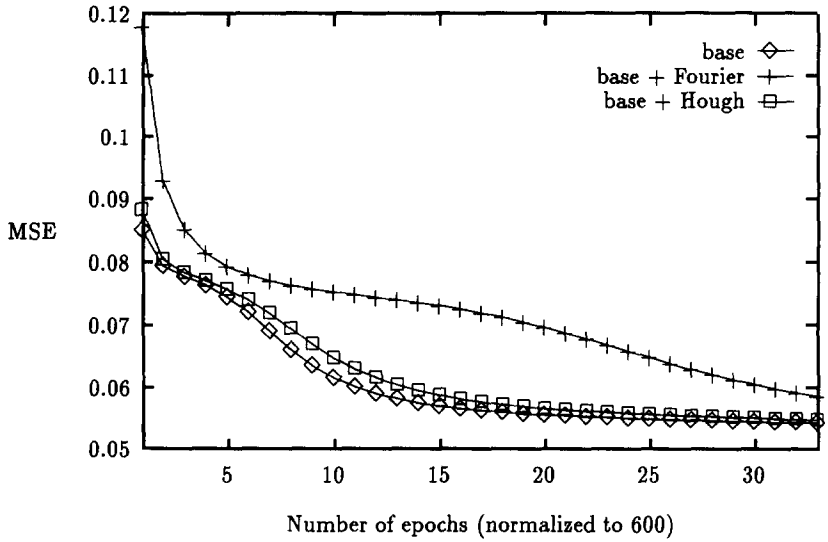

Figure 4. Effects of geometric descriptors on category-D modules.

allowing for the possible need to increase the number of hidden neurons (from six to twelve per module). The results are summarized in Figures 5 and 6, which show two extremes, respectively for categories S and W, of the typical behavior observed. Clearly, the joint use of both texture measures in addition to the base set seems to be an improvement over the addition to the base set of each measure individually, and even more so if an enlarged hidden layer is employed.

The startup network we employ henceforth in this section is then the following. Each module has 12 hidden neurons, and the total number of inputs is 36. Of these, 12 are in the base set of descriptors, 12 refer to the segment's entropy, and 12 to its correlation.

Our core network has all the inputs that the startup network does, and in addition the six neighborhood descriptors $n_1(S)$ through $n_6(S)$ for each segment S. These descriptors depend on all of S's neighbors, and this may be a problem in view of the possibility that some of these neighbors are not in the same set (training or test) as S. Of course this is a possibility only because we have forced all of our images to contribute with segments to both the training set and the test set. Normally, though, the core network will be given all the segments of an image for classification, and the absence of some of a segment's neighbors will simply not occur (except at the image's boundaries). In the case of this paper, the following is how we have dealt with this issue. Neighborhood descriptors for segments in the training set have taken into account all of its neighbors, even those in

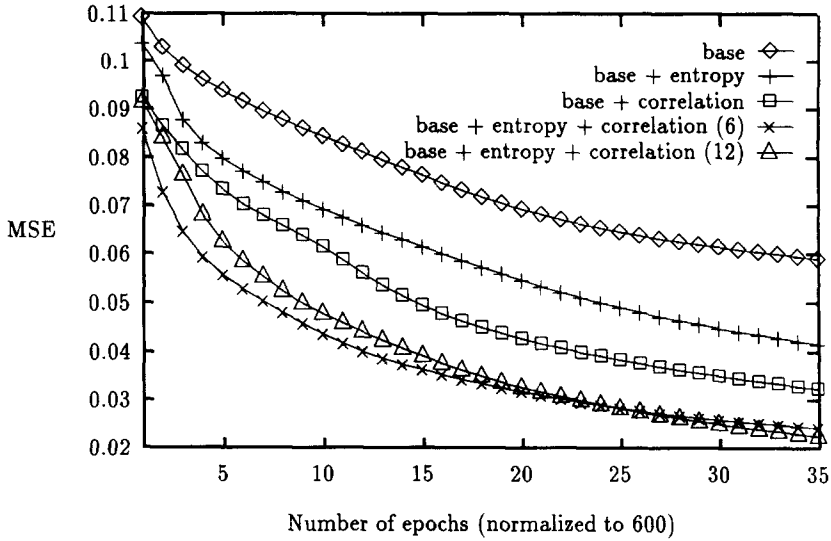

Figure 5. Effects of the size of the hidden layer on category-S modules.

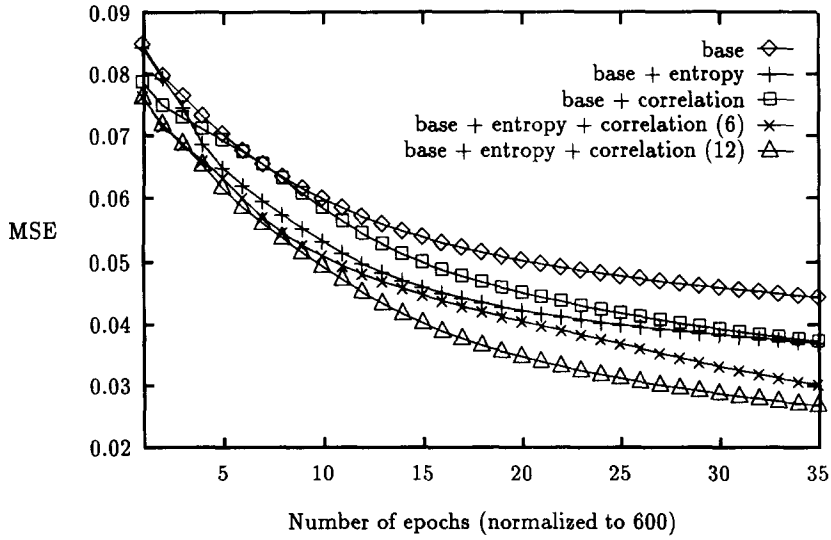

Figure 6. Effects of the size of the hidden layer on category-W modules.

the test set. For segments in the test set, as we shall see shortly, we have considered in our experiments both the inclusion of all neighbors in the computation of a segment's neighborhood descriptors and the inclusion of only those neighbors which are members of the test set as well.

It is important to mention that, during the training of the core network, the neighborhood descriptors for segments in the training set were computed using the classification provided by the startup network. Another possibility would have been to employ the photointerpreter's classification in the computation of the neighborhood descriptors, but clearly this would have provided the network with a distorted view of these descriptors.

The performance of CLASSIFY($\mathscr{S}$) on segments from our five images is shown in Figures 7 through 11 where we depict the evolution of the MSE as segments are presented to the core network for classification (in these figures, the initial value of the MSE is then the value obtained after all segments in $\mathscr{S}$ were classified by the startup network). We have employed $\varepsilon_1 = 0.2$ (empirically determined) and $K = \infty$. Three runs of CLASSIFY($\mathscr{S}$) are shown in each of these figures, one for $\mathscr{S}$ as the portion of the training set contained in the corresponding image, another for $\mathscr{S}$ as the portion of the test set contained in the corresponding image with neighborhood descriptors taking into account neighbors in the training set as well (complete neighborhood), and a third one for $\mathscr{S}$ as the portion of the test set contained in the corresponding image with neighborhood descriptors computed solely on test-set neighbors (incomplete neighborhood). When

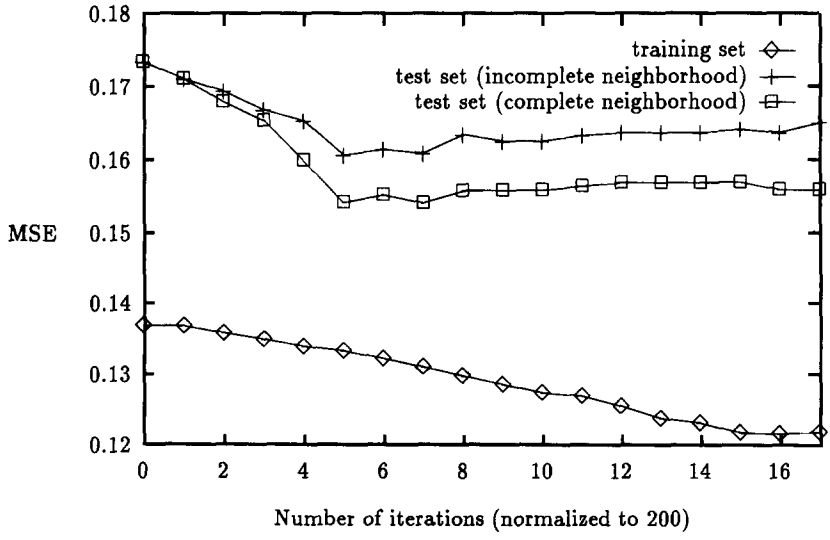

Figure 7. Evolution of MSE for a run of CLASSIFY ($\mathscr{S}$) on the Acre image.

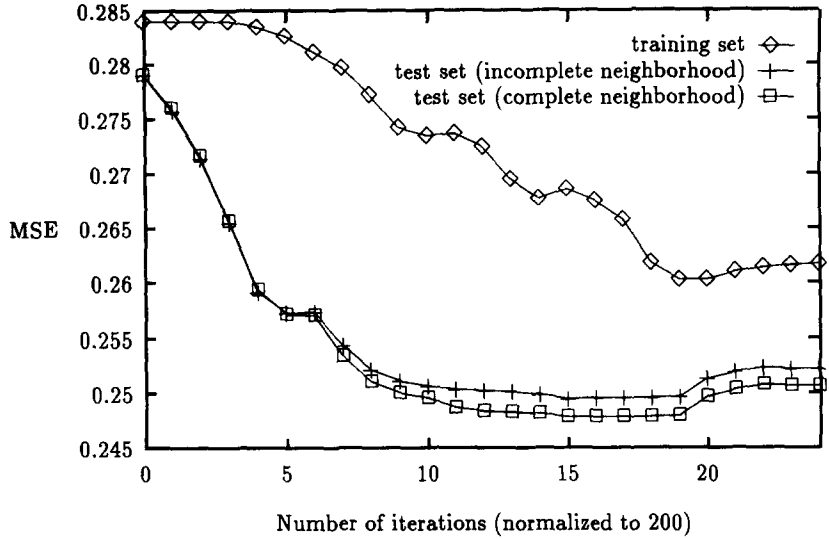

Figure 8. Evolution of MSE for a run of CLASSIFY ($\mathscr{S}$) on the Mato Grosso image.

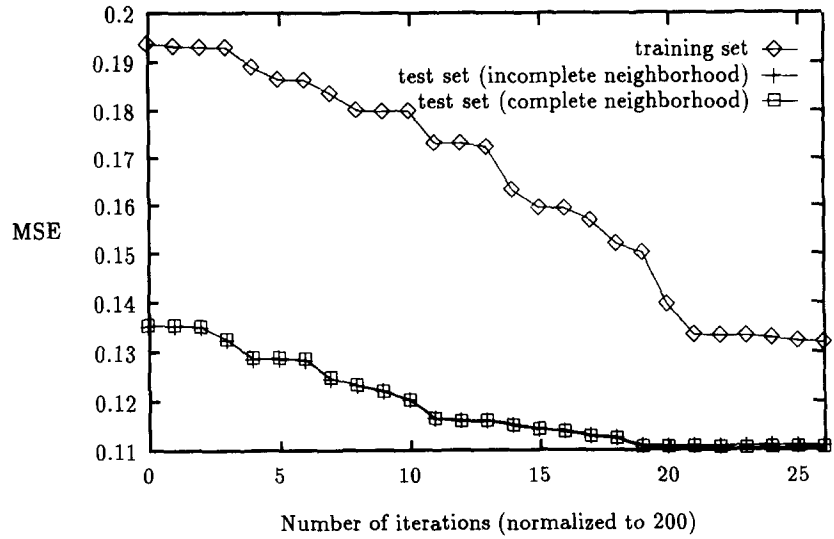

Figure 9. Evolution of MSE for a run of CLASSIFY ($\mathscr{S}$) on the Pará image.

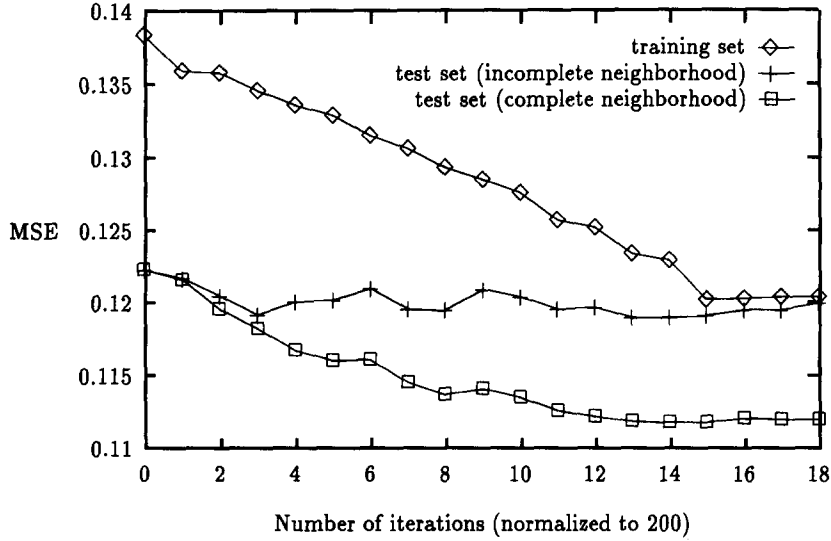

Figure 10. Evolution of MSE for a run of CLASSIFY ($\mathscr{S}$) on the Rondônia image.

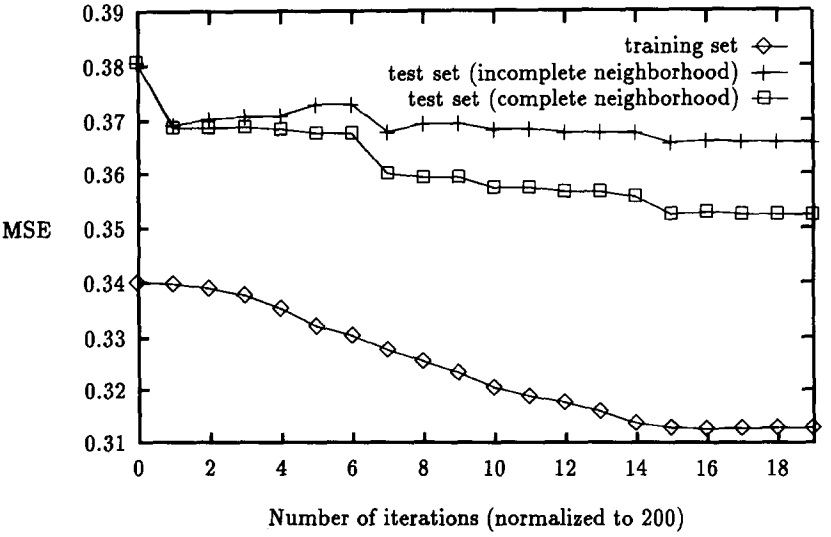

Figure 11. Evolution of MSE for a run of CLASSIFY ($\mathscr{S}$) on the Tucuruí image.

$\mathscr{S}$ is contained in the training set, the complete neighborhood of segments is used, but the classification of test-set segments employed to compute neighborhood descriptors is obtained with an incomplete neighborhood.

The first interesting trend that can be inferred from Figures 7 through 11 is that the MSE tends to evolve at lower values when a segment's neighborhood descriptors are computed on its complete neighborhood. This suggests the conjecture than an evolution somewhere in between the two curves corresponding to the test set would have occurred had the image contained test-set segments only. Secondly, the use of neighborhood descriptors does tend to lead the MSE to drop further from its value after the classification by the startup network, although not monotonically so. Even in images for which the use of neighborhood descriptors appears not to have been markedly effective, there were segments indicating otherwise. For example, in the image corresponding to Figure 11 there are small segments that the photointerpreter indicated to be primarily in category `Sh` but were classified by the startup network as belonging primarily to category `W`. This is understandable because of the spectral similarity of members of the two categories. These segments, when presented to the core network, had their classifications gradually changed from category `W` to category `Sh`. We perceive this as confirming our earlier expectation that neighborhood descriptors would be especially relevant for small segments belonging to an interfering category, in which cases it would be possible to infer the basic categories to which those segments belonged. In fact, much of the "reluctance" of the MSE to drop in Figure 11 can also be accounted for by this very inference mechanism, as the use of neighborhood descriptors led to the classification of many small segments as members of both categories `F` and `Sh`, whereas the photointerpreter was more conservative and did not infer (i.e., he classified these segments as belonging to category `Sh` solely). Later on we shall have more to say about the accuracy of a photointerpreter's classifications.

We show in Figure 12 the overall evolution of the MSE along the five runs of CLASSIFY($\mathscr{S}$) depicted in Figures 7 through 11. For each number of iterations, the MSE shown in Figure 12 has been obtained as a weighted sum of those of Figures 7 through 11, each weight being the fraction of the total number of segments in the training (or test) set corresponding to each image. Note that the initial and final values of the MSE are the same that would be obtained if CLASSIFY($\mathscr{S}$) were executed on the entire training (or test) set.

The results shown in Figure 12 indicate a clear tendency toward improvement in the value of the MSE with the use of the neighborhood descriptors. The initial and final MSE values shown in this figure are also given in Table 1, where we also show the corresponding values of the AMSE (in this table, i.n. stands for incomplete neighborhood and c.n. for

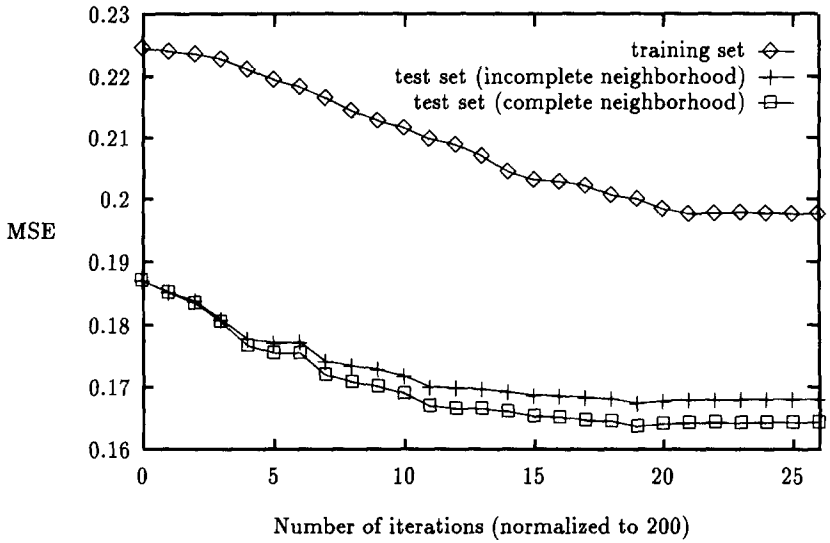

Figure 12. Overall evolution of MSE for the runs of CLASSIFY ($\mathscr{S}$) on the five images.

complete neighborhood). These values of the AMSE reflect the same trend with the use of the neighborhood descriptors.

The data displayed in Figures 7 through 12 may seem in disaccord with the usual expectation regarding the behavior of neural classifiers, according to which the value of the MSE for the training set should be lower than for the test set. It should be noted, in the present case, that the procedure we described earlier for the determination of the training and test sets may at times lead to the allocation of most "difficult" segments to one of the two sets, thereby causing the evolution of the MSE to run at higher values for the set to which those segments happened to be assigned.

As a final remark on the iterative use of the core network in algorithm CLASSIFY($\mathscr{S}$), we recall from our earlier observations that there does not

Table 1. MSE and AMSE Values Yielded by the Startup Network and by CLASSIFY($\mathscr{S}$)

	MSE		AMSE	
	Startup	CLASSIFY ($\mathscr{S}$)	Startup	CLASSIFY ($\mathscr{S}$)
Training set	0.22463	0.19753	0.13357	0.11650
Test set (i.n.)	0.18709	0.16431	0.16900	0.14011
Test set (c.n.)	0.18709	0.16806	0.16900	0.13823

Table 2. Hit Ratios of the Startup Network on the Basic Categories

	F		S		W		D	
F	82.98	(98.19)	4.12	(0.12)	1.00	(0.07)	4.36	(2.93)
S	0.59	(0.10)	51.55	(87.50)	0.00	(0.00)	1.60	(7.56)
W	0.59	(0.03)	0.00	(0.00)	99.00	(99.93)	0.25	(0.07)
D	15.83	(1.68)	44.33	(12.39)	0.00	(0.00)	93.80	(89.43)

appear to be any assurance that this relaxation scheme will always stabilize. Because our networks were trained by the back-propagation algorithm, it may be instructive to compare this possibility of unstable behavior with the same possibility in the context of the so-called *recurrent* back-propagation networks [27, p. 172]. Nevertheless, all the runs shown in Figures 7 through 11 did stabilize, as the algorithm did stop and the value we adopted for K (infinity) would allow no other stopping criterion but reaching stability [i.e., in CLASSIFY($\mathscr{S}$), the list that implements $\mathscr{S}$ must have become empty].

Let us now review some hit ratios computed using $\tau = 0.65$. Overall, the hit ratio of the startup network was 86.90% segments, or equivalently 94.07% pixels, while the hit ratio of CLASSIFY($\mathscr{S}$) was 86.90% segments, or equivalently 93.33% pixels, taking $\mathscr{S}$ to be the test set with incomplete neighborhood (complete neighborhoods brought these values up slightly, to 87.32% and 94.13%, respectively). So apparently the use of neighborhood descriptors has not affected these measures for the value of τ we chose. However, we can, as we mentioned earlier, focus on selected groups of categories to investigate other types of hit ratio, as we do next.

Other hit-ratio values are shown in Tables 2 through 9. Tables 2 and 3 refer to the basic categories only, Tables 4 and 5 refer to the interfering categories only, Tables 6 and 7 refer to inference in the presence of the interfering category C, and Tables 8 and 9 refer to inference in the presence of the interfering category Sh. In these pairs of tables, the first one depicts the performance of the startup network, and the second one the performance of CLASSIFY($\mathscr{S}$) when $\mathscr{S}$ is the test set with incomplete neighborhood.

Table 3. Hit Ratios of CLASSIFY ($\mathscr{S}$) on the Basic Categories

	F		S		W		D	
F	82.52	(97.98)	7.22	(0.34)	0.50	(0.04)	4.45	(6.03)
S	0.46	(0.05)	51.55	(88.26)	0.00	(0.00)	1.07	(6.93)
W	0.26	(0.02)	0.00	(0.00	99.50	(99.96)	0.21	(0.07)
D	16.75	(1.96)	41.24	(11.40)	0.00	(0.00)	94.26	(86.97)

Table 4. Hit Ratios of the Startup Network on the Interfering Categories

	C		Sh	
C	95.88	(94.39)	2.28	(1.96)
Sh	4.12	(2.97)	97.72	(95.07)

Table 5. Hit Ratios of CLASSIFY ($\mathscr{S}$) on the Interfering Categories

	C		Sh	
C	96.47	(94.47)	2.28	(1.96)
Sh	2.94	(2.17)	97.72	(95.19)

Table 6. Hit ratios of the Startup Network with Interference from C

	F		S		W		D	
F	86.67	(93.21)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
S	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
W	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
D	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	50.00	(28.05)

Table 7. Hit Ratios of CLASSIFY ($\mathscr{S}$) with Interference From C

	F		S		W		D	
F	90.00	(94.42)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
S	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
W	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
D	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	50.00	(28.05)

Table 8. Hit Ratios of the Startup Network with Interference from Sh

	F		S		W		D	
F	87.63	(89.75)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
S	1.03	(2.42)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
W	2.06	(0.85)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
D	1.03	(0.22)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)

Table 9. Hit Ratios of CLASSIFY ($\mathscr{S}$) with Interference from Sh

	F		S		W		D	
F	90.72	(92.67)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
S	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
W	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)
D	1.03	(0.22)	0.00	(0.00)	0.00	(0.00)	0.00	(0.00)

Columns in Tables 2 through 9 refer to the photointerpreter's classification, while rows refer to the outcome of the classifier under consideration in each table. Two numbers are shown in each table entry, one giving a hit ratio (multiplied by 100) in terms of numbers of segments, the other (in parentheses) giving the corresponding hit ratio (also multiplied by 100) in terms of numbers of pixels (areas). So, for example, in Table 2 we have that, of the segments that the photointerpreter indicated as belonging primarily to F, 82.98% were also classified as such by the startup network. Likewise, Table 6 indicates that, of the segments that the photointerpreter said belonged to both C and F, 86.67% were also classified by the startup network as belonging to F. Anomalous segments have not been considered in any of the tables, so the ratios in some columns do not add up to 100.

The high values of most diagonal entries in all of Tables 2 through 9 indicate very good performance. In addition, very often the inclusion of the neighborhood descriptors has elicited an improvement in performance.

To finalize our presentation of results on the original segments, we show in Table 10 figures for the sensitivity and the specificity of the startup network and of CLASSIFY($\mathscr{S}$) to each of the six categories when $\mathscr{S}$ is the test set with incomplete neighborhood. There is a slight trend toward increased sensitivity when neighborhood descriptors are used, except for

Table 10. Sensitivity and Specificity of the Startup Network and of CLASSIFY ($\mathscr{S}$)

	Sensitivity		Specificity	
	Startup	CLASSIFY ($\mathscr{S}$)	Startup	CLASSIFY ($\mathscr{S}$)
F	0.75	0.75	0.95	0.95
S	0.43	0.44	0.99	0.99
W	0.82	0.80	1.00	1.00
D	0.89	0.90	0.88	0.87
C	0.87	0.87	1.00	1.00
Sh	0.81	0.83	0.98	0.99

the interfering categories, while the classifiers' specificity appears to be generally unaffected by the use of these descriptors.

4.2. Results on Coalesced Segments

Our nearly 17,000 image segments were presented to algorithm COALESCE($\mathscr{S}$) so that neighboring segments given similar classifications by the alternate startup network could be merged into larger segments. This alternate startup network happened in this case to be precisely the startup network we selected to use in our evaluation of CLASSIFY($\mathscr{S}$), i.e., its inputs were a segment's entropy and correlation, in addition to the basic spectral and simple textural descriptors. Segment merging was performed separately for $\mathscr{S}$ as the training set and for $\mathscr{S}$ as the test set, meaning that the numbers of pixels in the resulting training and test sets were kept constant as the segments were merged. Coincidentally, it also happened that the proportion of two to one in the numbers of segments respectively in the training set and in the test set was also approximately maintained. We employed $\varepsilon_2 = 0.2$ as a result of preliminary experiments.

We show in Figures 13 and 14 the evolution of the MSE as we trained two modules (for categories S and D, respectively) of a potential startup network to act on coalesced segments. The base set of descriptors used in these figures is no longer the one we have been using so far, but comprises,

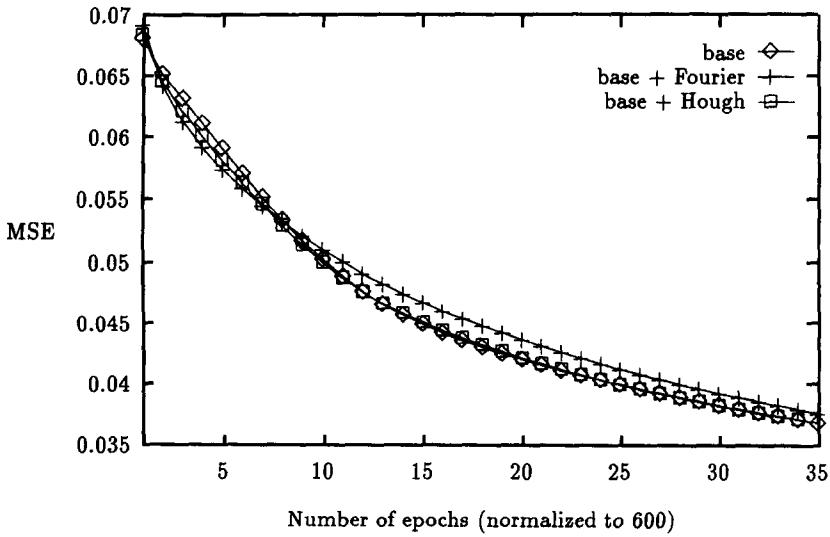

Figure 13. Effects of geometric descriptors for coalesced segments on category-S modules.

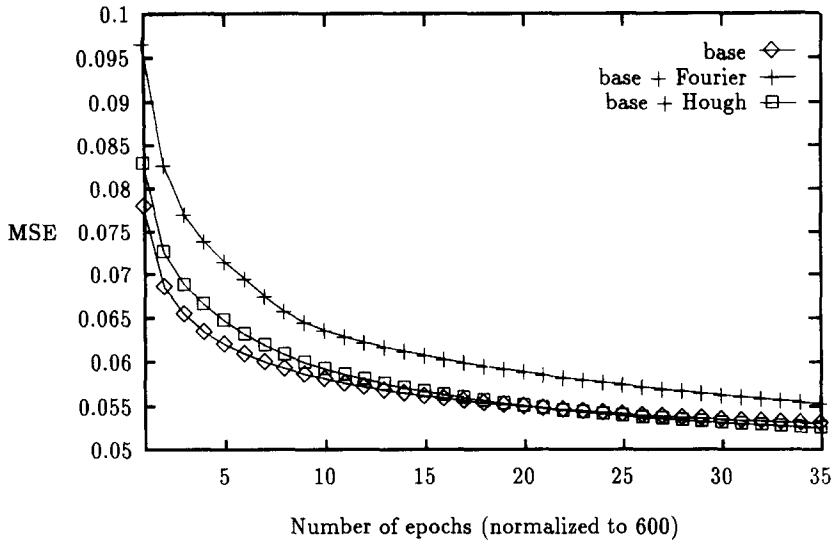

Figure 14. Effects of geometric descriptors for coalesced segments on category-D modules.

in addition to that set, a segment's entropy and correlation. These two figures may be thought of as the counterparts of Figures 3 and 4, respectively, presented earlier. Just as in those two earlier figures, the addition of Fourier or Hough descriptors to the base set does not seem to be effective.

The outcome of COALESCE($\mathscr{S}$) depends on the order assumed for $\mathscr{S}$ when it is regarded as a list. Although one might expect that different orders might provide segment groupings of more relevant geometry, our expectation is that such differences would be negligible and the resulting geometric descriptors equally ineffective.

5. CONCLUDING REMARKS

We have addressed in this paper the problem of automatic interpretation of Landsat images of the Amazon region for deforestation monitoring. The need to automate this monitoring process stems, as we argued, from the high degree of complexity of the task and from the need to perform it yearly for incremental monitoring. The approach we adopted comprises two independent phases. During the first phase the satellite images are segmented into spectrally homogeneous regions. In the second phase each segment is presented, in the form of a series of segment descriptors, to a classifier, whose output is the degree of membership of that segment in

each of the six categories of interest. These categories were chosen to aid in the deforestation assessment.

Two classifier architectures have been proposed and evaluated in this paper. The first architecture operates directly on the segments produced by the segmentation phase and employs two feedforward neural networks for segment classification. The startup network classifies a segment based on its spectral, geometric, and textural descriptors, while the core network employs the segment's neighborhood descriptors as well. Because of the nature of these neighborhood descriptors, this first architecture embodies a relaxation scheme during which the core network may be used more than once on each segment. The second classifier architecture follows the same general guidelines as the first one, but in contrast operates on fewer segments, obtained from the original segments by the coalescence of neighboring segments that were classified similarly by the alternate startup network. Inputs to this network are solely the segment's spectral and textural descriptors. The purpose of this second architecture is to restore regions of meaningful geometry, which are frequently excessively fragmented during the segmentation phase.

We have reported on extensive experiments with both classifier architectures. All of our experiments were based on neural networks trained by the back-propagation training algorithm. In order to train these neural networks, several thousand segments from representative portions of the Amazon region were manually classified by a photointerpreter and then employed as examples. Another batch of these segments, also classified by the photointerpreter, was kept aside for later assessment of the classifiers' ability to generalize.

Training each neural-network module consumed many hours on an IBM RISC System/6000 model 580 workstation. Once trained, the network was in general capable of processing each of the images in a few minutes on the same machine.

The classifiers' performance was evaluated on the basis of four different measures, namely the mean squared error, hit ratios, sensitivity, and specificity. Each of these measures may be employed having either segments or pixels as basic units. In the latter case, the performance figures relate more directly to the classifiers' ultimate goal (i.e., to assess the evolution of deforestation in the Amazon region), as they refer to the same units as end users do, namely areas (as numbers of pixels). However, it appears to be more appropriate to employ segments as the basic units to evaluate the classifiers, as the area-related measures tend to be too sensitive to the presence of too large segments in either the training set or the test set of segments.

Our results on the original segments indicate that in general two of the textural descriptors based on the cooccurrence matrix for the segment

(entropy and correlation) and the neighborhood descriptors tend to provide improvements on classifications obtained solely from spectral descriptors and very simple textural descriptors. They also indicate that the geometric descriptors and the other textural descriptors are in general ineffective. Our results on the coalesced segments, which to some degree we had expected to benefit from the inclusion of geometric information, pointed toward the ineffectiveness of our geometric descriptors as well. If we believe in the importance of geometric information in the detection of deforestation (and apparently this is indeed to be believed, as human photointerpreters seem to rely on such information to a large extent), then the conclusion is that simply coalescing neighboring segments does not suffice to restore the polygonal geometry of deforested regions. Instead, as we discuss further on, the segmentation phase must be made such that it does not distort so much the presence of polygonal contours.

Overall, our results appear to be comparable to the results typically reported on the performance of other classifiers employing traditional pixel-based techniques. However, apparently these other techniques have a very hard time dealing with the transition and interference phenomena we mentioned earlier, which are a fundamental motivation for our fuzzy approach. For example, results recently reported on the same images that we employed in our evaluation are comparable to ours but only look at those portions of the image to which the photointerpreter assigned crisp degrees of membership [29]. What this means is that approximately one-fifth of the entire area of the images had to be left untouched, owing to that classifier's inability to deal with fuzzy information. In addition to the ease with which our approach deals with the inherent fuzziness of the domain, there is also the intrinsic ability that it has to cope with the geometric, textural, and contextual (in the form of immediate neighborhoods) characteristics of spectrally homogeneous regions of an image. Pixel-oriented approaches seem to lack this ability.

We feel the results we obtained, albeit good, can still be polished if we carry out the training of our neural networks perhaps just a bit further. Training a neural network by the back-propagation algorithm may require quite some time, especially for complex domains such as the one we have been considering. For this reason, it would be inaccurate to say that the many networks we used were trained to their best performances, and therefore it is quite possible that improvements can still be achieved.

The overall methodology we employed in this paper can of course be used in other contexts, including some that involve more specialized inquiries concerning environmental issues in the Amazon region. Just as the study of deforestation patterns led to the consideration of the four basic categories we used throughout this paper, other studies based on the same methodology would require their own sets of categories of interest.

In this vein, the development of another system is being undertaken to identify in greater detail the types of vegetation within the forest and savannah areas, as well as some types of sediments in the watercourses in the region. Likewise, more sophisticated applications, such as the identification of particular types of human settlement (urban settlements, farming areas, and so on), might also in various degrees benefit from our methodology.

In addition to this application of our methodology to tackle a different problem within the Amazon region, the work we have described is currently the subject of extensions in many directions. Some of these extensions are methodological in nature, while some others are considerably more technical. For example, one of the methodological issues we have been considering has to do with the evaluation of our classifiers. As we mentioned earlier, the classification provided by a photointerpreter is in general the best we can hope to have as a standard against which to compare the output of the classifiers. However, no photointerpreter is fully consistent with himself, which amounts to yet another degree of uncertainty in the standard he provides us with. The photointerpreter who provided us with the classifications we used throughout as a standard appears to be approximately 90% consistent with himself (i.e., on two occasions separated by a few months he provided two sets of classifications on the same set of segments with 90% of matches in number of segments, although nearly 98% of matches in number of pixels). In the future we plan to find out how consistent he is with other photointerpreters, and to consider taking a few field trips to selected locations to enhance the determination of our standard for comparisons.

One important technical issue that we feel needs to be addressed has to do with the current complete separation between the segmentation and classification phases in our approach. Perhaps a higher degree of integration between the two phases might be able to produce less fragmented segments in the situations in which this fragmentation leads to the loss of most of the geometric information of interest. Although we are as yet uncertain as to how this might be achieved, one possibility seems to be the development of a segmentation system that somehow takes into account the categories of interest at later stages, or a system that does not merely follow a region-growing technique but also tries to preserve significant rectilinear boundaries untouched. This latter possibility has already received some attention elsewhere [30, 31].

Image boundaries constitute the source of other troublesome possibilities, as we have not in any way been treating image-boundary segments differently than inner segments. Segments located at the boundaries of images possess artificial rectilinear portions in their contours, and a considerable portion of their neighborhoods is altogether absent. Clearly,

such artificial rectitude will tend to confuse the classification system and perhaps cause it to misinterpret the importance of rectilinear segment boundaries as indicators of the presence of human activity in the corresponding region. In addition, for these image-boundary segments neighborhood descriptors will tend to do poorly. One possible solution to these problems is the use of an additional (symbolic) segment descriptor to indicate that the segment is located at the boundary of an image, and that therefore its rectitude and neighborhood are not to be given undue importance.

Another technical issue with many interesting ramifications is concerned with the neural model and corresponding training algorithm employed in the various neural networks that constitute our classifiers. We are currently investigating the use of an AND-OR fuzzy model which holds the promise of efficient training, especially when compared with the back-propagation algorithm we employed [32].

ACKNOWLEDGMENTS

We are thankful to the following colleagues, who in various ways contributed to the research described in this paper. Cláudia Ferlin participated in the early stages of the design of our classifiers, and Ralph Linsker of the IBM Thomas J. Watson Research Center offered several insightful comments during the same phase. Dave McCall of the Machine Vision Group at IBM Greenock gave us very useful suggestions for our implementation of the back-propagation algorithm. Guaraci Erthal and Leonardo Bins, of Brazil's Institute for Space Research (INPE), implemented the software for the segmentation of the images, and produced all the segments we used in our experiments. Adriano Venturieri, under the supervision of João Roberto dos Santos, both of INPE as well, did the manual classification (photointerpretation) of all the segments.

References

1. Gonzalez, R. C., and Woods, R. E., *Digital Image Processing*, Addison-Wesley, Reading, Mass., 1992.
2. Townshend, J. R. G., and Justice, C. O., Unsupervised classification of MSS Landsat data for mapping spatially complex vegetation, *Internat. J. Remote Sensing* 1(2), 105–120, 1980.
3. Schowengerdt, R. A., *Techniques for Image Processing and Classification in Remote Sensing*, Academic, Orlando, Fla., 1983.

4. Wu, J.-K., Zheng, Q.-F., Wang, T.-T., and Li, K.-W., A forest inventory using LANDSAT imagery in the Mao-shan area of China, *Internat. J. Remote Sensing* 6(12), 1783–1795, 1985.

5. Nelson, R., and Holben, B., Identifying deforestation in Brazil using multiresolution satellite data, *Internat. J. Remote Sensing* 7(3), 429–448, 1986.

6. Williams, D. L., and Nelson, R. F., Use of remotely sensed data for assessing forest stand conditions in the eastern United States, *IEEE Trans. Geosci. and Remote Sensing* GE-24(1), 130–138, 1986.

7. Benediktsson, J. A., Swain, P. H., and Ersoy, O. K., Neural network approaches versus statistical methods in classification of multisource remote sensing data, *IEEE Trans. Geosci. and Remote Sensing* 28(4), 540–552, 1990.

8. Excell, P. S., and Studholme, C., A neural network for spatial feature recognition, in *Proceedings of the 16th Annual Conference of the Remote Sensing Society*, Swansea, U.K., 224–231, 1990.

9. Kanellopoulos, I., Varfis, A., Wilkinson, G. G., and Mégier, J., Neural network classification of multi-date satellite imagery, in *Proceedings of the International Geoscience and Remote Sensing Symposium* (Vol. IV), Helsinki, Finland, 2215–2218, 1991.

10. Mulder, N. J., and Spreeuwers, L., Neural networks applied to the classification of remotely sensed data, in *Proceedings of the International Geoscience and Remote Sensing Symposium* (Vol. IV), Helsinki, Finland, 2211–2213, 1991.

11. Heermann, P. D., and Khazenie, N., Classification of multispectral remote sensing data using a back-propagation neural network, *IEEE Trans. Geosci. and Remote Sensing* 30(1), 81–88, 1992.

12. Haralick, R. M., and Shapiro, L. G., Image segmentation techniques, *Comput. Vision, Graphics and Image Process.* 29(1), 100–132, 1985.

13. Huntsberger, T. L., Rangarajan, C., and Jayaramamurthy, S. N., Representation of uncertainty in computer vision using fuzzy sets, *IEEE Trans. Comput.* C-35(2), 145–156, 1986.

14. Krishnapuram, R., and Keller, J., Fuzzy set theoretic approach to computer vision: An overview, in *Proceedings of the IEEE International Conference on Fuzzy Systems*, San Diego, 135–142, 1992.

15. Pal, S. K., Fuzzy sets in image processing and recognition, in *Proceedings of the IEEE International Conference on Fuzzy Systems*, San Diego, 119–126, 1992.

16. Terano, T., Asai, K., and Sugeno, M., *Fuzzy Systems Theory and Its Applications*, Academic, Boston, 1992.

17. Kent, J. T., and Mardia, K. V., Spatial classification using fuzzy membership models, *IEEE Trans. Pattern Anal. and Machine Intell.* 10(5), 659–671, 1988.

18. Machado, R. J., Barbosa, V. C., Liporace, F. dos S., and Ferlin, C., Monitoring the deforestation of the Amazon region with neural networks, in *Proceedings of the International Joint Conference on Neural Networks*, Nagoya, Japan, 1239–1242, 1993.

19. Dreyer, P., Classification of land cover using optimized neural nets on SPOT data, *Photogrammetric Eng. & Remote Sensing* 59(5), 617–621, 1993.

20. Lillesand, T. M., and Kiefer, R. W., *Remote Sensing and Image Interpretation*, Wiley, New York, 1987.

21. Wall, K., and Danielsson, P.-E., A fast sequential method for polygonal approximation of digitized curves, *Comput. Vision, Graphics and Image Process.* 28, 220–227, 1984.

22. Costa, L. da F., and Sandler, M. B., The binary Hough transform and its implementation, in *Proceedings of the SPIE / SPSE Symposium on Electronic Imaging Science and Technology*, Santa Clara, Calif., 183–193, 1990.

23. Costa, L. da F., and Sandler, M. B., Detecting digital straight line segments in $O(N^2)$, in *Proceedings of the International Workshop on Visual Form*, Capri, Italy, 165–174, 1991.

24. Peckinpaugh, S. H., An improved method for computing gray-level cooccurrence matrix based texture measures, *Comput. Vision, Graphics and Image Process.* 35(6), 574–580, 1991.

25. Haralick, R. M., Shanmugam, K., and Dinstein, I., Textural features for image classification, *IEEE Trans. Systems Man Cybernet.* SMC-3(6), 610–621, 1973.

26. Visa, A., A texture classifier based on neural network principles, in *Proceedings of the International Joint Conference on Neural Networks*, San Diego, I-491–I-496, 1990.

27. Hertz, J., Krogh, A., and Palmer, R. G., *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, Calif., 1991.

28. Bailey, D., and Thompson, D., How to develop neural-network applications, *AI Expert* 5(6), 38–47, 1990.

29. Machado e Silva, A. J. F., An evaluation of the impact of compression techniques on digital image classification (in Portuguese), Tech. Report CCR-153, IBM Rio Scientific Center, Rio de Janeiro, Brazil, June 1993.

30. Haddon, J. F., and Boyce, J. F., Image segmentation by unifying region and boundary information, *IEEE Trans. Pattern Anal. and Machine Intell.* 12(10), 929–948, 1990.

31. Pavlidis, T., and Liow, Y.-T., Integrating region growing and edge detection, *IEEE Trans. Pattern Anal. and Machine Intell.* 12(3), 225–233, 1990.

32. Machado, R. J., and Rocha, A. F., Evolutive fuzzy neural networks, in *Proceedings of the IEEE International Conference on Fuzzy Systems*, San Diego, 493–500, 1992.