Brain volumes characterisation using hierarchical neural networks

Sergio Di Bona[a], Heinrich Niemann[b], Gabriele Pieri[a], Ovidio Salvetti[a,*]

[a] *Institute of Information Science and Technologies, Italian National Research Council, Via G. Moruzzi, 1-56124 Pisa, Italy*

[b] *Bavarian Research Center for Knowledge-Based Systems Haberstrasse, 2-D-91058 Erlangen, Germany*

Abstract

Objective knowledge of tissue density distribution in CT/MRI brain datasets can be related to anatomical or neuro-functional regions for assessing pathologic conditions characterised by slight differences. The process of monitoring illness and its treatment could be then improved by a suitable detection of these variations. In this paper, we present an approach for three-dimensional (3D) classification of brain tissue densities based on a hierarchical artificial neural network (ANN) able to classify the single voxels of the examined datasets. The method developed was tested on case studies selected by an expert neuro-radiologist and consisting of both normal and pathological conditions. The results obtained were submitted for validation to a group of physicians and they judged the system to be really effective in practical applications.

Keywords: Hierarchical neural networks; Artificial neural networks; 3D density classification; Brain imaging

1. Introduction

The nature of a brain pathology observed through the analysis of tomographic scans can be associated with tissue density variations since the radiation absorption of biotissue is directly related to its density. As tissue density is related to water concentration, in a number of relevant cases, a neuropathology presents biological alterations related to density modifications, such as the appearance of a hypodensity area due the growth of

* Corresponding author. Tel.: +39-050-315-3124; fax: +39-050-315-2810.
E-mail addresses: dibona@iei.pi.cnr.it (S. Di Bona), niemann@informatik.uni-erlangen.de (H. Niemann), pieri@iei.pi.cnr.it (G. Pieri), ovidio.salvetti@isti.cnr.it (O. Salvetti).

interstitial water. Moreover, it is well known that a brain tumour is characterised by several cell proliferations that strengthen their concentration in a volume unit. As a consequence, a hypodensity region may not be clearly distinguishable in a tomographic image compared to a normal tissue. Moreover, there are numerous cases of possible uncertainty, such as low degree malignant neoplastic forms, brain tissue early ischemic lesions or oedema. In this context, the ability to measure even slight tissue density variations in MRI/CT brain scans could help to make the subject's analysis more objective, especially when an integration of both three-dimensional (3D) morphological and densitometric features can be performed to increase the effectiveness of the measurements themselves, since this would refer to a more complete brain model [7,10,16]. Starting from well-defined cerebral regions, a densitometric property can be locally measured and then assigned to a given class in order to characterise a specific tissue. Each class can be then considered to assess modifications occurring in different 3D models of the brain, assuming that the same brain or different brains have been made spatially corresponding and analogous encephalic structures can thus be compared [2,3]. The 3D brain data derived from the tomographic scans, can be organised into a knowledge base in order to automatically compare the densitometric features of an unknown brain, when possibly damaged regions cannot be recognised accurately by visual examination.

In this paper, we present an approach for brain tissues density analysis and classification using artificial neural networks (ANN). In particular, the method developed is based on a hierarchical ANN model applied to 3D brain volumes. The ability of ANNs to recognise a set of inputs belonging to a given stimuli space is used here to classify sets of features extracted from the voxels belonging to the volumes examined. The selection of the features is a crucial aspect of the classification process and the problem faced does not allow us to identify a unique optimal set a priori with any great certainty.

For this reason, an architectural network paradigm was taken to model different cases of the same problem instantiation by adopting a two-level hierarchy of independent specialised networks, the first one being suitable to map the feature sets, the second one capable of performing a final classification of the voxels on the basis of the output given by the previous net.

We started from the assumption that the use of a single ANN to model such a complex problem, would entail retraining the entire network whenever the description of the problem (i.e. the feature set) changed. In fact, as yet there is no knowledge available that would predict the most accurate result.

On the other hand, the adoption of a multilevel ANN allows for specialisation and adaptability at the same time, since its individual levels can be finely tuned to the characteristics of the feature set and also the entire architecture can be easily modified if the problem description changes, in particular by training only those levels involved in the modification.

Furthermore, the approach followed allows us to optimise the computational complexity of the individual levels independently. In particular, the addition of a new feature would only have a partial effect on an individual level of the ANN architecture, making it easier to work on the problem of density volume classification. These features are extracted taking into account the 3D properties of the densitometric volumes, in such a way as to be able to pursue a *full 3D* approach.

In the following, the hierarchical ANN developed to classify brain densities is presented in detail. The results of the case studies selected by an expert neuro-radiologist, consisting of both normal and pathological brain conditions, effectively demonstrated the proposed method and encourage further improvements for developing a more complete tool capable of supporting the analysis of disease diagnosis and follow-up.

2. Materials and methods

A 3D brain volume is defined by a dataset of parallel equidistant slices (i.e. brain scans) assuming that the acquisition sampling rate is accurate enough to avoid any loss of information.

As a first step, in order to significantly reduce the number of voxels to be analysed during the classification process, the entire background region located outside the skull is disregarded. For this purpose, an algorithm for the automatic segmentation of the internal and external skull boundaries, based on a priori knowledge of the images' characteristics, was implemented [5]. This was done in such a way that the region belonging to the skull had already been classified and the region of interest (ROI) to be analysed corresponded to the region inside the internal boundary of the skull itself [4].

Once the algorithm has been applied to all the slices of the dataset, the set of ROIs obtained define the volume of voxels to be classified (*3D-ROI*).

All the voxels are then processed to compute a set of features that are used to obtain the classification of the voxels themselves. In the following, each voxel coordinate is computed relative to the skull; in order to overcome problems related to the displacement of the skull between slices.

Voxel density is classified using a hierarchical ANN architecture. Several ANN-based approaches have been presented in the literature, including some that use hierarchical structures [6,8,11,12,14,15].

Sahiner et al. [14] propose a *convolution neural network* with two-dimensional (2D) weight kernels to classify density regions in mammograph images. Two different classification techniques are studied: classification by image subsampling and classification based on two different sets of textural features which are extracted from the original images either as '*grey level difference statistics*' or as '*spatial grey level dependencies*'. The features extracted are first arranged as texture images and then used as input for the net.

Ersoy and Deng [6] describe a hierarchy composed of one level of parallel error back-propagation (EBP)-based networks, each of equal complexity. In this approach, there is evidence of a global network consisting of small parallel EBP stages which converge faster than a single EBP network of the same total size in terms of similar error performance.

Sajda et al. [15] introduce a hierarchical architecture composed of two levels of classical EBP-based networks that is more accurate than a non-hierarchical architecture. The architecture uses the output of the first network, together with extracted features of higher resolution, such as a contextual input to the other network.

Reddick et al. [12] propose an interesting architecture based on a combination of a self-organising map (SOM) [9] with an EBP network. The first net, composed by an input layer of three neurons fully connected with nine output neurons, is used to *segment* the three

signal intensities from a single pixel in a T1-weighted, T2-weighted and PD magnetic resonance images. The output neurons have a 3D weight vector that represents the input for the second net that is composed by three inputs, seven hidden and output neurons. Each output neuron corresponds to one of the seven different classification values.

Keem et al. [8] propose a parallel hierarchy of EBP networks, one specialised in the densitometric analysis of the coronary arteriography and the other specialised in the geometric aspect. Their system was developed to assist an observer in artery lesion detection.

Mavrovouniotis and Chang [11] give an overview of hierarchical neural networks and a comparison of common three layer ANN architectures. Each of the sub-net within the hierarchy is intended to capture specific aspects of the input data. A sub-net models a particular subset of the input variables, but the exact patterns and relationships among variables are determined by training the network as a whole. The advantages of the hierarchy of sub-nets are summarised in two main points: the modular organisation makes the networks easier to analyse and the hierarchy gives the network hints about the most promising directions to look for patterns.

In our approach, we considered important to achieve the following:

- to exploit the differences among the extracted features to improve the classification capability;
- to be able to easily change the number of features;
- to implement a *full 3D* approach, in terms of spatial geometric relations among the neuro-functional structures.

To this end, a hierarchical ANN architecture was implemented as shown in Fig. 1.

In particular the recognition system uses two kinds of neural network that can be trained separately to perform the basic recognition task. The implementation of the global network by means of two independent network levels implies a rapid and efficient training of each individual level.

This classification mainly consists of two phases: at a lower level, the classification of the individual features extracted from each voxel is performed; at a higher level, the results of the first phase are the input for the final classification.

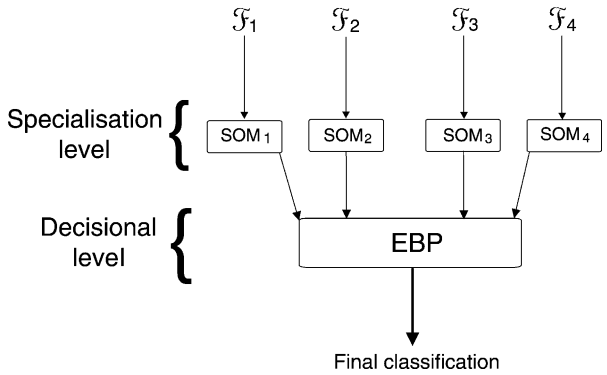

Fig. 1. Hierarchical classification architecture.

The lower level (*specialisation level*) is composed of a set of different *experts* (i.e. classifiers). Each expert is based on an unsupervised SOM model [9], and the training is performed with the aim of clustering each input value into crisp classes, without using any information related to the brain tissue class which the voxel belongs to. Each specific feature is the input to only one classifier of the specialisation level; in this way, each SOM can be individually optimised without affecting the other components of the global ANN architecture. This is done in order to reduce locally the computational complexity and, at the same time, to implement a flexible system.

The higher level (*decisional level*) is composed of a single *final* classifier based on an EBP model.

In the following sections a detailed description of each step is given.

2.1. Classification features

Each 3D-ROI is classified on the basis of specific properties related to the voxels themselves and their spherical neighbourhood.

The set of features chosen was selected in such a way as to reduce the probability of having overlapping classification values from voxels belonging to different density classes. To do this, we defined a set of independent features which represent both geometric and densitometric properties of the 3D-ROIs to be classified.

The features described below include information related to the 3D spatial configuration of the tissues modelled and they experimentally demonstrated a discriminating set to effectively distinguish adjacent structures in the domain examined.

Assuming that the 3D-ROI is defined by the function $R(V) = (x_V, y_V, z_V) = k_V$, where k_V is the grey value of V, the following set of features is computed [4]:

- grey level and position of V:

$$F_{\mathrm{p}}(V) = (k_V, x_V, y_V, z_V), \tag{1}$$

- local mean grey value computed in a spherical neighbourhood $S(V, r)$ with centre in V and radius r:

$$F_{\mathrm{m}}(V) = \frac{\sum_{P \in S(V,r)} R(P)}{|S(V,r)|} \tag{2}$$

  where $|S(V, r)|$ is the number of voxels in the sphere,
- difference between the local mean grey value computed in $S(V, r)$ and the grey value of V:

$$F_{\mathrm{d}}(V) = |F_{\mathrm{m}}(V) - R(V)|, \tag{3}$$

- gradient computed in $S(V, r)$:

$$F_{\mathrm{g}}(V) = \alpha \vec{G} \tag{4}$$

$$\alpha = \frac{1}{|S(V,r)|}$$

$$\vec{G} = (G^x(V,r), G^y(V,r), G^z(V,r))$$

$$G^x(V,r) = \left(\sum_{(x,y_V,z_V)\in S(V,r)} R(x+1,y_V,z_V) - R(x,y_V,z_V) \right)$$

$$G^y(V,r) = \left(\sum_{(x_V,y,z_V)\in S(V,r)} R(x_V,y+1,z_V) - R(x_V,y,z_V) \right)$$

$$G^z(V,r) = \left(\sum_{(x_V,y_V,z)\in S(V,r)} R(x_V,y_V,z+1) - R(x_V,y_V,z) \right).$$

Each voxel is represented by its set of features which are, in turn, used as inputs for the specialisation level of the classifier. The classification of the features also implies the classification of the corresponding voxel.

2.2. Classification process

Hierarchical classification is composed of different levels of parallel classifiers. Our approach consists of two levels, the specialisation level gets as input the features computed in the previous step and sends its output to the decisional level, whose task is to combine the output values of the previous level and to perform the final classification.

Since there are two classifiers, the problem of combining their (possibly) conflicting decisions arises. There are usually two main approaches: parallel and sequential. In the parallel approach, the results of the two classifiers are merged by means of some kind of voting mechanism. In the sequential approach, the simplest and thus more efficient classifier is run first: if the input is recognised with a high confidence degree the second classifier is not applied, otherwise the second classification is also performed and the two results are merged [1].

The disadvantage of this last approach is that any misclassification performed by the first classifier cannot be overruled by the second and, therefore, there must be a guarantee that this classifier yields a very low error rate.

In the architecture proposed in this paper, the approach is slightly different: the two classification levels have a complementary role to each other and they are always run together. The specialisation level performs a partial classification, each of the experts trying to separate the input data into crisp classes. This classification is performed without using any information on the density class the voxel belongs to (i.e. unsupervised learning), but using only the appropriate feature for each of the experts. The output of the specialisation level is used to form the input pattern for the decisional level, which refines the classification and gives a final response, i.e. the density class which the voxel belongs to.

2.2.1. Specialisation level classification

This classification is performed by a series of modules, each implementing a SOM model. The number of modules corresponds to the number of features to be recognised, i.e. $F_{\mathrm{p}}(V)$, $F_{\mathrm{m}}(V)$, $F_{\mathrm{d}}(V)$, $F_{\mathrm{g}}(V)$.

SOMs are one-layer ANNs. This means that every output neuron receives the input stimulus and is connected to the output neurons that are spatially close (i.e. in an interaction range ρ).

Each SOM acts as an independent classification module, i.e. it gives a classification value for each voxel of the dataset without taking into account the output of the other SOMs. The input is different for each SOM: every module is trained and then used for the classification of a specific feature, therefore its input is only one of the four processed features (i.e. a quadruple for the first feature, a single value for the second and third feature, and a triple for the last feature).

These SOMs are *mono-dimensional* (1D) networks, i.e. the only layer of the network is composed of an array of neurons; this means that the dimension of the weights depends on the feature: 1D for $F_{\mathrm{m}}(V)$ and $F_{\mathrm{d}}(V)$, and multidimensional for $F_{\mathrm{p}}(V)$ and $F_{\mathrm{g}}(V)$.

We chose to use 1D SOMs due to the high separability of the classes identified by the features.

Furthermore, our system is very open to the addition of new computed features, which could give more information about the classification of the voxels. In fact the addition of new features does not influence the existing specialisation modules and only requires minor changes to the decisional level: i.e. the addition of new 1D/2D SOM modules dedicated to receiving the new evaluated features and to classifying them.

The weights in the networks are defined for the output neurons and are changed during the training phase.

For each input feature $F_k(V)$ ($k \in \{m, d, p, g\}$), the neuron i that best proximates the projection of the input itself into the network (in terms of Euclidean distance) is considered to be the neuron most excited by the stimulus; its weights w_i are modified in such a way as to specialise i on the specific input. In addition, the weights of the neurons within a fixed neighbourhood are modified in order to preserve the topological architecture of the net; in other words neighbour neurons are *excited* by similar inputs [9].

At the end of the training, each network is able to recognise and to classify the values of a specific feature.

The weights $w_j(it)$ of a generic neuron j at time it, for the input $F_k(V)$ ($k \in \{m, d, p, g\}$) are modified as follows:

$$w_j(it+1) = \begin{cases} w_j(it) + \alpha(it)[F_k(V)] - w_i(it)] & \text{if } j \in N_i(it) \\ w_i(it) & \text{if } j \notin N_i(it) \end{cases} \tag{5}$$

where

- $w_i(it)$ represents the weights at time it of the neuron i most excited by the stimulus;
- $\alpha(it) = \left(1 - \left(\frac{it-1}{T}\right)\right) \in]0, \ldots, 1[$ is the learning coefficient, and it depends on the iteration number it, and on the maximal number of iterations T;
- $F_k(V)$ is the value of the input feature computed for the voxel V;
- $N_i(it)$ is the set of neurons in a fixed neighbourhood of radius ρ and centred in i, influenced by the modification of $w_i(it)$.

The output of each SOM is the classification of the voxel on the basis of the specific feature considered and it is identified as the number of the winning node for that voxel.

```
begin{
    Randomly initialise the weights
    Set constant T to the maximal number of iterations
    Set it = 0
    while (weights are changing) and (it < T) do
        Randomly choose an input value F_k(V)
        Locate the neuron i most excited by F_k(V)
        Neuron i represents the classification of F_k(V)
        Modify weights of neuron i and of the neurons in a
            predefined neighbourhood (Eq. 5)
        it = it + 1
    end_while
} end
```

Fig. 2. Algorithm used to train the SOM network.

Therefore, the output of the specialisation level is a n-dimensional vector containing the responses of the n SOMs (in this case $n = 4$); this vector defines the input stimulus of the decisional level.

The algorithm used to train a SOM to classify a generic feature $F_k(V)$ is shown in Fig. 2.

After the SOMs have been trained over the set of voxels selected for the *learning* process, they are ready to classify each input (see the algorithm in Fig. 3).

The number of neurons chosen in each SOM can be greater than the number of density classes and then reduced (optimised) by applying a clustering algorithm based on distance [16].

2.2.2. *Decisional level classification*

The set of classifications computed by the specialisation level is used as input for the top level classifier implemented with an EBP network.

The algorithm implemented has the following parameters:

- The network used is a feed-forward back-propagation network,
- The training function updates weights according to a resilient back-propagation algorithm [13],
- The dimension of the input stimuli corresponds to the number of different features taken into account (in this case the input stimuli are defined by a four-dimensional (4D) array),
- The input layer is composed by n_{I} neurons,
- The output layer is composed by n_{O} neurons that correspond to the number of density classes to be identified,

```
begin{
    while (there are features to classify) do
        Get input F_k(V)
        Locate the neuron i most excited by F_k(V)
        Neuron i represents the classification of F_k(V)
        return(i)
    end_while
} end
```

Fig. 3. Algorithm used to classify a generic feature with the SOM ANNs.

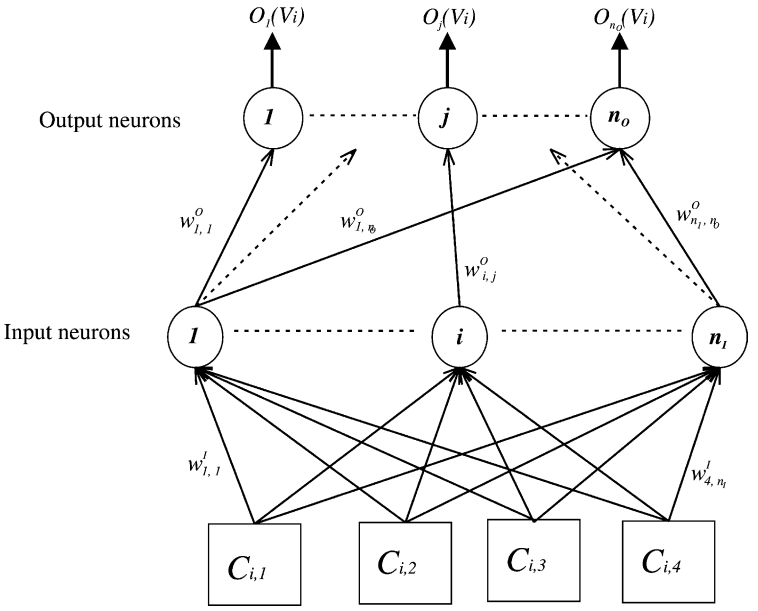

Fig. 4. EBP architecture. $C_{i,1},\ldots,C_{i,4}$: input stimulus; $W^{\mathrm{O}}_{l,m}$, $w^{\mathrm{I}}_{k,j}$: weights; n_{I}: number of input neurons; n_{O}: number of output neurons; $O_j(V_i)$: final classification of V_i. In our case studies $n_{\mathrm{I}} = 25$ and $n_{\mathrm{O}} = 8$.

- The network is used with no hidden layer; experimental tests have shown that hidden layers do not improve either the performance or the quality of the results,

Fig. 4 shows the architecture of the decisional level. The format of the input is an array containing, for each voxel to be classified, the 4D response given by the specialisation level to the specific feature extracted. Each array is used as an input stimulus for each of the n_{I} neurons of the input layer.

During the training phase of the EBP, a matrix is used for the input where every row consists of the above mentioned array extended with a value $C_{i,T}$ that represents the correct classification value of the voxel V_i (*target class*). This last value is used to evaluate the *error signal*, i.e. the difference between the network response and the correct classification value, and then to modify the weights (i.e. the synapses) in order to better match the target class.

In order to have a significant training-set, an expert neuro-radiologist selected a set of sample voxels from each image in such a way as to be representative of the class which they belong to. Thus, the training-set is composed of a set of voxels specifically selected from each image.

The supervised algorithm will be able to train the network by propagating backward the resulting error $\delta_j(V)$. This error is calculated as a function of the neurons' output and the difference between the network output and the target class for the specific voxel V.

$$\delta_j(V) = \begin{cases} f_j'(\mathrm{net}_j(V))(C_{V,T} - O_j(V)) & \text{if } j \in \mathrm{OutN} \\ f_j'(\mathrm{net}_j(V)) \sum_{k \in \mathrm{OutN}} (\delta_k(V) w_{j,k}) & \text{if } j \in \mathrm{InN} \end{cases} \tag{6}$$

```
begin{
  n = total number of input pattern (i.e. sample voxels)
  for_each voxel V_i do
      Get the input from specialisation level as
        the classification of the features considered
      Compute the output of the network propagating
        the input through weights
      Compare the output with the target class C_{i,T}
        and evaluate the error δ_j(V_i) (Eq. 6)
      Back-propagate the error and update the weights
        up to the input layer (Eq. 7)
  end
} end
```

Fig. 5. Algorithm implemented to train the EBP network.

where $O_j(V)$ is the output value of the neuron j for the input V, $\text{net}_j(V)$ is the weighted sum of the ingoing signals to the neuron j for the input V, $f_j'(\text{net}_j(V))$ is the derivative of an activation function f used to compute the output; this function is usually defined as the identity, OutN is the set of output neurons, and InN is the set of input neurons.

The output neurons propagate this error backwards to each input neuron. The error is used by each neuron to update its weights on the incoming connections and to calculate the error to be propagated back itself. Input neurons only update the weights of their connections, without propagating the error.

Having an input V, the weights from neuron i to neuron j (i.e. $w_{i,j}$) are updated by a value $\Delta_{i,j}(V)$ computed as follows:

$$\Delta_{i,j}(V) = \eta \delta_j(V) O_i(V) \tag{7}$$

where η represents the *learning rate* and can be chosen either as a constant or a function of the number of iteration, like in the SOM algorithm.

An outline of the EBP algorithm is shown in Fig. 5.

The network update procedure is iterated over the set of voxels until either an error threshold is reached, or the number of iterations reaches a pre-defined value (which depends on the specific domain under examination). After the training, the network can be used as classifier (see Fig. 6 for the classification algorithm).

```
begin{
  for_each voxel V_i to be classified do
      Get the input from specialisation level as
        the classification of the features considered
      Compute the output of the network propagating the input
      Take the most excited neuron i of the output level
        as the final classification of V_i
  end
} end
```

Fig. 6. Algorithm implemented by the EBP network to perform the final classification.

The low number of inputs and the absence of intermediate neurons allow the network to be trainable in a very short time. The output of this network is the final classification of the voxel processed.

3. Results

The model proposed was applied to real cases: 3D CT/MRI data sets relative to different patients and representing both normal and pathological conditions were selected as study cases by an expert neuro-radiologist.

The brain scans were acquired at the Neurosurgery Department of the University of Pisa, with a *GE Medical Systems* machinery, model *Genesis Jupiter.*

Our study involved a set of 15 patients who were used to train the net, and a set of eight patients used to test the hierarchical ANN (the two sets are separate). The training set was composed of seven women and eight men, aged between 50 and 70 years old. Each CT/MRI sequence had a number of slices varying between 15 and 23.

The CT/MRI slices were used to obtain a 3D representation of the volume to be classified.

A graphical interface tool was also developed in order to assist the neuro-radiologist in selecting the set of training stimuli, i.e. a set of voxels chosen as being representative of the class which they belong to. The set of voxels extracted and used for the training phase varied in the interval $[10, \ldots, 70]$ thus obtaining a total number of 4500 learning stimuli. The number of training iteration was fixed at 300.

The features described above were extracted from these voxels and used as inputs for the SOMs.

The SOM network specialised on $F_{\mathrm{p}}(V)$ had a 4D input (i.e. $\langle X$-coord, Y-coord, Z-coord, grey level$\rangle$); the SOM network specialised on $F_{\mathrm{g}}(V)$ had a 3D input, and the other two SOMs had a 1D input (since the last two features had real values).

Since in our experiments eight different density classes were taken into account (see Table 1), each SOM had a $\langle 8 \times 1 \rangle$ neurons layer. Each voxel was firstly classified by each of the four SOMs; the outputs were then used as inputs for the EBP network.

All these inputs were sent to all the 25 neurons which make up the input layer of the EBP network; the output layer of the decisional level network is composed of eight neurons,

Table 1
Typology specification of density classes

Density class	Typology
1	Bone
2	Grey matter
3	White matter
4	Liquor
5	Air
6	Hypodensity
7	Hyperdensity
8	Blood

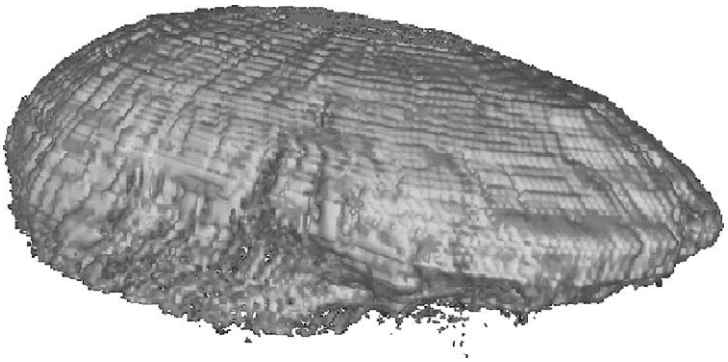

Fig. 7. Example of 3D reconstruction of a brain section relative to the density classified as grey matter. The volume computed is 372 cm^3.

corresponding to the number of possible density classes. The output neuron most excited by the inputs represents the final classification.

The number of operations computed for each voxel is 148 for the features extraction phase and 700 for the classification phase (132 computations are performed by the specialisation level while 568 by the decisional level classifier).

The following examples refer to both normal and pathologic brain conditions. In both cases a number of approximately 850,000 voxels were classified for a total number of about 720,800,000 operations computed in an average elapsed time of 20 s.

Figs. 7 and 8 show the 3D representations of two single density classes extracted from the normal case volume, ''grey matter'' and ''white matter'', respectively. Fig. 9 shows both densities simultaneously in the same portion of the brain. Reconstructions were obtained using AV-Express^{TM} software package by ''*Advanced Visual System Inc.*''.

Figs. 10 and 11 show examples of classification where all the eight different classes are considered (see Table 1): a 3D section of a pathological case (Fig. 10) and a non-pathological case (Fig. 11).

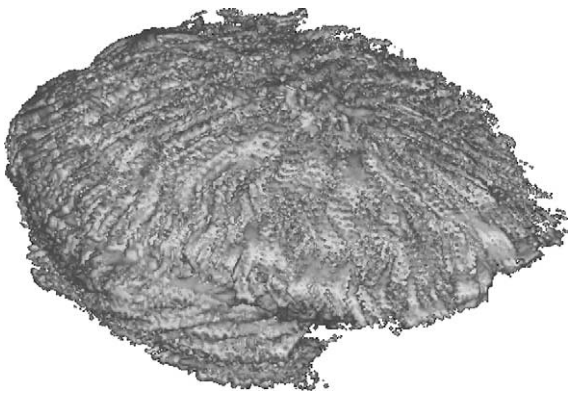

Fig. 8. Example of 3D reconstruction of a brain section relative to the density classified as white matter. The volume computed is 391 cm^3.

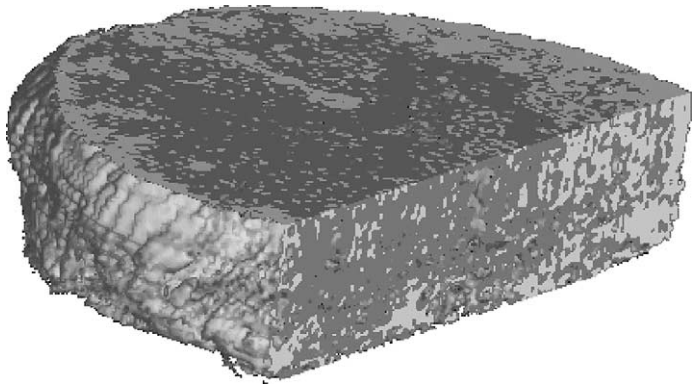

Fig. 9. 3D density reconstruction of ‘‘white matter’’ (dark color) and ‘‘grey matter’’ (light color), in a normal patient. The volume is shown with a vertical cut for better evidencing the different internal structures.

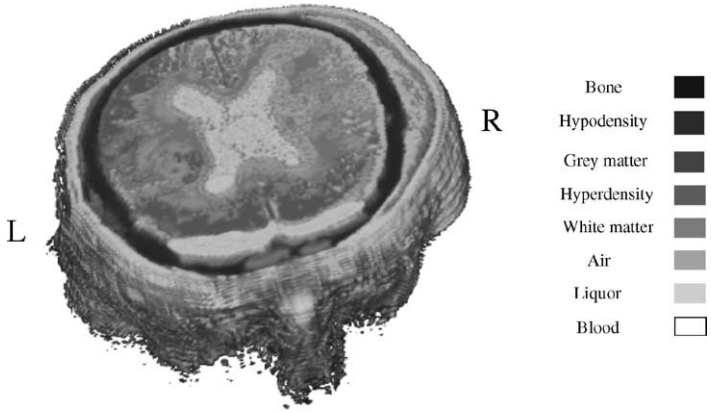

Fig. 10. 3D visualisation of the voxels classification in a pathological case of the encephalon (cerebral ischemia).

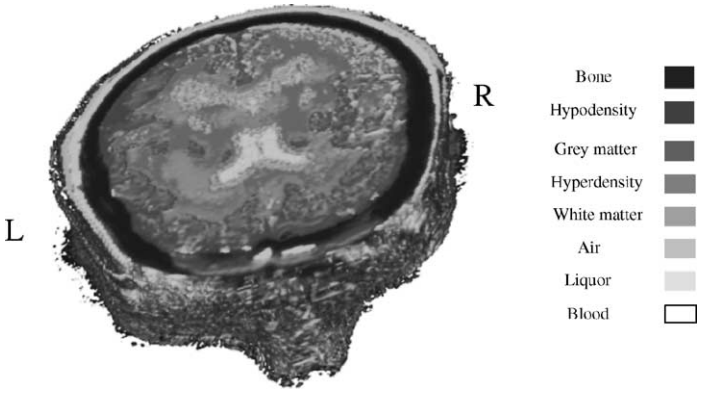

Fig. 11. 3D visualisation of the voxels classification in a normal case of the encephalon.

4. Conclusions

A diagnosis of cerebral anomalies is usually based on the examination and comparison of digital scans of the tissues involved. This leads to qualitative evaluations of densitometric and volumetric modifications of the anomalies themselves. Although visually inspecting tomographic scans performed by an expert physician is a fundamental step when diagnosing pathologies, the automatic quantification of densitometric and volumetric properties of a lesion is an important aid when a static evaluation of the perturbing agent is not sufficient and the evolution of the pathology needs to be studied.

From a biological point of view, a cerebral lesion is characterised by two elements: its size and tissue composition. From a radiological point of view these two aspects correspond to the morphological and densitometric characteristics of the anomaly. The volume of the lesion is undoubtedly a valuable measure of its morphological properties, whereas a densitometric characterisation is more complicated to obtain.

An important point is that quantitative measurements related to the densitometric properties of the tissues cannot replace a qualitative inspection of the digital scans. However, they can integrate and complete the physical characterisation of a lesion and the description of its temporal evolution. These measurements are aimed at differentiating the pathologic tissues from the normal ones and can be computed by analysing the textural properties and the grey level distribution of the digital scans.

In the light of this, we have defined a method for the textural characterisation of anatomical soft tissues in order to support the diagnosis of cerebral pathologies. For this purpose, real complex cases with evident anomalies were taken into account to test the effectiveness of the approach.

We introduced a multilevel approach for the classification of tissue density in 3D image data sets of the brain. This approach is based on the use of SOM and EBP networks to classify a set of pre-defined features extracted from each voxel of the volumetric data set; these features are considered as being representative of the voxel itself and thus lead to its final classification.

CT/MRI data sets referring to both normal and pathological conditions, selected by an expert neuro-radiologist, were used as case studies to test our approach.

The data set was selected by the experts in such a way as to avoid ambiguous conditions and to be able to assess the objectivity of the model proposed. We are confident that ambiguous conditions can be classified by introducing, at the first level of the architecture, networks specialised in the analysis of those ambiguities (e.g. partial volumes). The flexibility of our model allows new modules to be added with only slight changes to the whole architecture.

The results obtained from the case studies were submitted for validation to a group of physicians at the Department of Neurosurgery, University of Pisa. They judged the system to be really effective in practical applications. This fact could be a basis for designing a more complete instrument to support the analysis of disease diagnosis and follow-up.

Performance is a critical aspect of the architecture described, since the use of an EBP-based network introduces a significant computational overhead. In order to improve efficiency, specific algorithms for the reduction of the number of voxels to be examined are currently being studied.

As for the EBP network, the performance can be improved by the exploitation of a *rejection rate*. In fact, when more than one class can be assigned to an individual voxel, the network could either try to refine the result or reject the input. In these cases, we found advantageous to reject an individual voxel and optionally assign a default classification value (e.g. *blank*) or direct the expert to that region, rather than accepting an erroneous classification or trying to refine the result.

Acknowledgements

The authors would like to thank Prof. L. Lutzemberger and his staff at the Department of Neurosurgery (University of Pisa), for their contribution in the preparation of the case studies. This paper has been partially supported by the Italian Ministry of Technological and Scientific Research.

References

[1] Cao J, Ahmadi M, Shridar M. A hierarchical neural network architecture for handwritten numeral recognition. Pattern Recognit 1997;30(2):289–94.

[2] Di Bona S, Huwer S, Niemann H, Salvetti O. The brain matcher. In: Radig B, Niemann H, Zhuravlev Y, Gourevitch I, Laptev I, editors. Proceedings of the 5th Open German–Russian Workshop on Pattern Recognition and Image Understanding. Sankt Augustin, Germany: Infix; 1999

[3] Di Bona S, Huwer S, Niemann H, Salvetti O. Non-linear neural enhancement of anatomical differences in deformed brain MR-images. Pattern Recognit Image Anal 1999;9:55–65.

[4] Di Bona S, Pieri G, Salvetti O. A multilevel neural network model for density volumes classification. In: Loncaric S, Babic H, editors. Proceedings of the IEEE 2nd International Symposium on Image and Signal Processing and Analysis—ISPA'01. Zagreb, Croatia: University of Zagreb; 2001. p. 213–8.

[5] Di Bona S, Salvetti O. Approach for 3D volumes matching. In: Nasrabadi NM, Katsaggelos AK, editors. Proceedings of the IS&TI SPIE 13th International Symposium on Electronic Imaging 2001, vol. 4305. Artificial Neural Networks in Image Processing. Bellingham, WA, USA: SPIE- Int. Soc. Opt. Eng; 2001. p. 73–80.

[6] Ersoy OK, Deng SW. Parallel, self-organizing, hierarchical neural networks with continuous inputs and outputs. IEEE Trans Neural Networks 1995;6(5):1037–44.

[7] Evangelista R, Salvetti O. A morphometric and densitometric approach to image interpretation. Pattern Recognit Image Anal 1993;3:103–10.

[8] Keem S, Meadows H, Kemp H. Hierarchical neural networks in quantitative coronary arteriography. In: Proceedings of the 4th International Conference on Artificial Neural Networks—ANN'95. London, UK: IEE; 1995.

[9] Kohonen T. Self-Organizing Maps. Springer Series in Information Sciences, second ed., vol. 30. Berlin: Springer; 1997.

[10] Lutzemberger L, Salvetti O. Brain modelling by structural and functional data fusion. Acta Neurochir 1996;138(5):613–4.

[11] Mavrovouniotis ML, Chang S. Hierarchical neural networks. Comput Chem Eng 1992;16(4):347–69.

[12] Reddick WE, Glass JO, Cook EN, Elkin TD, Deaton RJ. Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks. IEEE Trans Med Imaging 1997;16:911–8.

[13] Rumelhart DE, McClelland JL. Parallel distributed processing: explorations in the microstructure of cognition, vols. 1 and 2. MIT Press, Cambridge (MA): Bradford; 1986.

[14] Sahiner B, Chan H, Petrick N, Wei D, Helvie MA, Adler DD, et al. Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images. IEEE Trans Med Imaging 1996;15:598–610.

[15] Sajda P, Spence C, Pearson J. A hierarchical neural network architecture that learns target context: applications to digital mammography. In: Proceedings of the 1995 International Conference on Image Processing, vol. 3. Los Alamitos, CA, USA: IEEE Comput. Soc. Press; 1995. p. 3149–51.

[16] Salvetti O, Braccini G, Evangelista R, Freschi M. An intelligent system for the diagnosis of complex images. Artif Intell Med 1996;8:167–85.