

Wide field imaging – I. Applications of neural networks to object detection and star/galaxy classification

S. Andreon,^{1★} G. Gargiulo,^{2,3} G. Longo,¹ R. Tagliaferri^{4,5} and N. Capuano²

¹*Osservatorio Astronomico di Capodimonte, via Moiariello 16, 80131 Napoli, Italy*

²*Facoltà di Scienze, Università di Salerno, via S. Allende, 84081 Baronissi (Salerno), Italy*

³*IASS ‘E. R. Caianiello’, via G. Pellegrino, 19, 84019 Vietri s/m (Salerno), Italy*

⁴*DMI, Università di Salerno, via S. Allende, 84081 Baronissi (Salerno), Italy*

⁵*INFN Unità di Salerno, via S. Allende, 84081 Baronissi (Salerno), Italy*

Accepted 2000 May 4. Received 2000 April 28; in original form 1999 November 26

ABSTRACT

Astronomical wide-field imaging performed with new large-format CCD detectors poses data reduction problems of unprecedented scale, which are difficult to deal with using traditional interactive tools. We present here NEXt (Neural Extractor), a new neural network (NN) based package capable of detecting objects and performing both deblending and star/galaxy classification in an automatic way. Traditionally, in astronomical images, objects are first distinguished from the noisy background by searching for sets of connected pixels having brightnesses above a given threshold; they are then classified as stars or as galaxies through diagnostic diagrams having variables chosen according to the astronomer’s taste and experience. In the extraction step, assuming that images are well sampled, NEXt requires only the simplest a priori definition of ‘what an object is’ (i.e. it keeps all structures composed of more than one pixel) and performs the detection via an unsupervised NN, approaching detection as a clustering problem that has been thoroughly studied in the artificial intelligence literature. The first part of the NEXt procedure consists of an optimal compression of the redundant information contained in the pixels via a mapping from pixel intensities to a subspace individualized through principal component analysis. At magnitudes fainter than the completeness limit, stars are usually almost indistinguishable from galaxies, and therefore the parameters characterizing the two classes do not lie in disconnected subspaces, thus preventing the use of unsupervised methods. We therefore adopted a supervised NN (i.e. a NN that first finds the rules to classify objects from examples and then applies them to the whole data set). In practice, each object is classified depending on its membership of the regions mapping the input feature space in the training set. In order to obtain an objective and reliable classification, instead of using an arbitrarily defined set of features we use a NN to select the most significant features among the large number of measured ones, and then we use these selected features to perform the classification task. In order to optimize the performance of the system, we implemented and tested several different models of NN. The comparison of the NEXt performance with that of the best detection and classification package known to the authors (SEXTRACTOR) shows that NEXt is at least as effective as the best traditional packages.

Key words: methods: data analysis – techniques: image processing – catalogues.

1 INTRODUCTION

Astronomical wide-field (hereafter WF) imaging encompasses the use of images larger than 4000² pixels (Lipovestky 1994) and is the only tool to tackle problems based on rare objects or on

statistically significant samples of optically selected objects. Therefore, WF imaging has been and still is of paramount relevance to almost the whole field of astrophysics: from the structure and dynamics of our Galaxy to the environmental effects on galaxy formation and evolution to the large-scale structure of the Universe. In the past, WF was the almost exclusive domain of Schmidt telescopes equipped with large photographic plates and

★ E-mail: andreon@na.astro.it

was the main source of targets for photometric and spectroscopic follow-ups using telescopes in the 4-m class. Nowadays, the exploitation of the new generation 8-m class telescopes, which are designed to observe targets that are often too faint to be even detected on photographic material (the POSS-II detection limit in B is ~ 21.5 mag), requires digitized surveys realized with large-format CCD detectors mounted on dedicated telescopes. Much effort has therefore been devoted worldwide to constructing such facilities: the MEGACAM project at the CFHT, the ESO Wide Field Imager at the 2.2-m telescope, the Sloan DSS and the ESO–OAC VST (Arnaboldi et al. 1999) are only a few of the ongoing or planned experiments.

One aspect which is never stressed too often is the enormous problem posed by the handling, processing and archiving of the data produced by these instruments: the VST alone, for instance, is expected to produce a flow of almost 30 Gb of data per night or more than 10 Tb per year of operation.

The scientific exploitation of such a huge amount of data calls for new data reduction tools, which must be reliable, must require a small amount of interactions with the operators and must be as independent of a priori choices as possible.

In processing a WF image, the final goal is usually the construction of a catalogue containing as many as possible astrometric, geometric, morphological and photometric parameters for each individual object present in the image. The first step in any catalogue construction is therefore the detection of the objects, a step which, as soon as the quality of the images increases (both in depth and in resolution), becomes much less obvious than it may seem at first glance. The traditional definition of ‘object’ as a set of connected pixels having brightness higher than a given threshold has in fact several well-known pitfalls. For instance, low surface brightness galaxies very often escape recognition because (i) their central brightness is often comparable to or fainter than the detection threshold, and (ii) their shape is clumpy, which implies that even though there may be several nearby pixels above the threshold, they can often be unconnected and thus escape the assumed definition.

A similar problem is also encountered in the catalogues extracted from the *Hubble Deep Field* (HDF), where a plethora of small ‘clumpy’ objects is detected but it is not clear whether each clump represents an individual object or instead is a fragment of a larger one. Ferguson (1998) stresses some even stronger limitations of the traditional approach to object detection: (i) a comparison of catalogues obtained by different groups from the same raw material and using the same software shows that, near the detection limits, the results are strongly dependent on the assumed definition of ‘object’; (ii) object detection performed by the different groups is worse than that attained by even an untrained astronomer by visually inspecting an image – many objects that are present in the image are lost by the software, while others that are missing on the image are detected (hereafter spurious objects; see Fig. 1). In other words, the silent assumption that faint objects consist of connected and amorphous sets of pixels makes the definition of ‘object’ astronomer-dependent and produces quite ambiguous results at very low signal-to-noise ratios (S/N).

The classification based only on morphological grounds of an object as a star or as a galaxy relies instead on whether the object is spatially resolved or not. Human experts can usually classify objects either directly from the appearance of the objects on an image (either photographic or digital) or from the value of some finite set of derived parameters via diagnostic diagrams (such as magnitude versus area). This approach, however, is much too

time-consuming and much too dependent on the ‘know how’ and personal experience of the observer: (i) the choice of the most suitable parameters varies greatly from author to author, making comparisons difficult if not impossible, and (ii) regardless the complexity of the problem, owing to the obvious limitations of representing three or more dimensions in space on a two-dimensional graph, only two features are often considered. In recent years much effort has therefore been devoted to implementing and fine-tuning artificial intelligence (AI) tools to perform star/galaxy classification on automatic grounds. The powerful package *SEXTRACTOR* (hereafter *SEX*: Bertin & Arnouts 1996), for instance, uses nine features (eight isophotal areas and the peak intensity) and a neural network to classify objects. The *SEX* output is an index, ranging from 0 to 1, which gives the degree of ‘stellarity’ of the object. This, however, still implies a fair degree of arbitrariness in choosing these features and not any other set. Other approaches to the same problems will be reviewed in the last section of this paper.

This paper is divided into two major parts: in Section 2 we present the AI theory used in the paper and in Section 3 the experiments. Finally, in Section 4 we discuss our results and draw some conclusions.

2 THE THEORY

In the AI domain there are dozens of different NNs used and

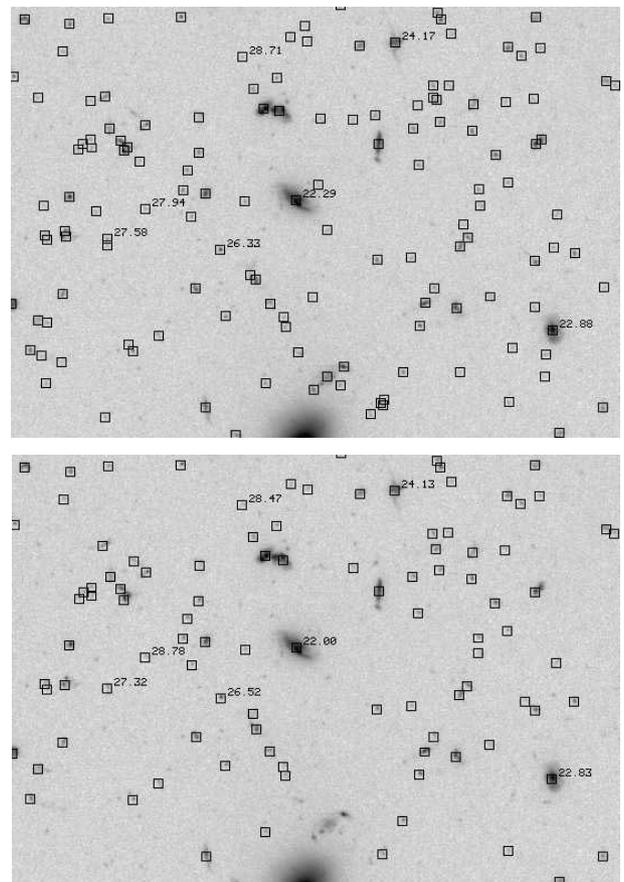


Figure 1. A portion of the *Hubble Deep Field*. The two panels show the objects in the Couch (1996) and Lanzetta, Yahil & Fernandez-Soto (1996) catalogues. The figures are taken from Ferguson (1998), and are reproduced with the permission of Cambridge University Press.

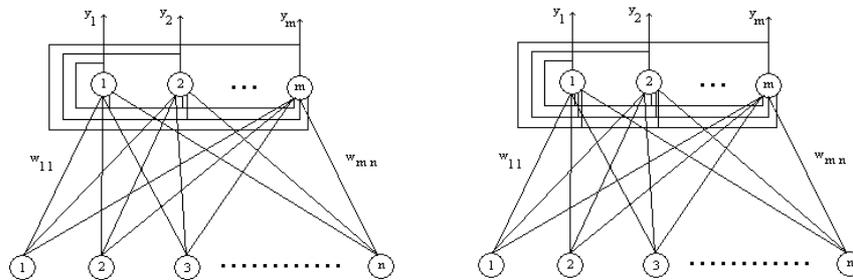


Figure 2. Hierarchical PCA NN (left) and symmetric PCA NN(right).

optimized to perform the most various tasks. In astronomical literature, in contrast, only two types of NNs are used: the Artificial NN, called in AI literature ‘multilayer perceptron (MLP) with back-propagation learning algorithm’, and Kohonen’s self-organizing maps (or their supervised generalization).

We followed a rather complex approach, which can be summarized as follows. Principal Component Analysis (PCA) NNs were used to reduce the dimensionality of the input space. Supervised NNs need a large number of labelled data to obtain a good classification, while unsupervised NNs overcome this need, but do not provide good performances when classes are not well separated. Hybrid and unsupervised hierarchical NNs are therefore very often introduced to simplify the expensive post-processing step of labelling the output neurons in classes (such as objects/background), in the object detection process. In the following subsections we illustrate the properties of several types of NNs that were used in one or another of the various tasks. All the discussed models were implemented, trained and tested and the results of the best performing ones are illustrated in detail in the next sections.

2.1 PCA neural nets

A pattern can be represented as a point in a L -dimensional parameter space. To simplify the computations, a more compact description is often needed, where each pattern is described by M , with $M < L$, parameters. Each L -dimensional vector can be written as a linear combination of L orthonormal vectors or as a smaller number of orthonormal vectors plus a residual error. PCA is used to select the orthonormal basis that minimizes the residual error.

Let \mathbf{x} be the L -dimensional zero mean input data vectors and $\mathbf{C} = \mathbf{E}(\mathbf{x}\mathbf{x}^T) = \langle \mathbf{x}\mathbf{x}^T \rangle$ be the covariance matrix of the input vectors \mathbf{x} . The i th principal component of \mathbf{x} is defined as $\mathbf{x}^T \mathbf{c}(i)$, where $\mathbf{c}(i)$ is the normalized eigenvector of \mathbf{C} corresponding to the i th largest eigenvalue $\lambda(i)$.

The subspace spanned by the principal eigenvectors $\mathbf{c}(1), \dots, \mathbf{c}(M)$, ($M < L$) is called the PCA subspace (with dimensionality M : Oja 1982; Oja et al. 1996). In order to perform PCA, in some cases and especially in the non-linear one, it is convenient to use NNs that can be implemented in various ways (Oja 1982; Baldi & Hornik 1989; Sanger 1989; Jutten & Herault 1991; Oja, Ogawa & Wangviwattana 1991; Plumbley 1996). The PCA NN used by us was a feedforward neural network with only one layer, which is able to extract the principal components of the stream of input vectors. Fig. 2 summarizes the structure of the PCA NNs. As can be seen, there is one input layer, and one

forward layer of neurons that is totally connected to the inputs. During the learning phase there are feedback links among neurons, the topology of which classifies the network structure as either hierarchical or symmetric depending on the feedback connections of the output layer neurons.

Typically, Hebbian type learning rules are used, based on the one-unit learning algorithm originally proposed by Oja (1982). The adaptation step of the learning algorithm – in this case the network is composed of only one output neuron – is then written as

$$w_j^{(t+1)} = w_j^{(t)} + \mu y^{(t)} [x_j^{(t)} - y^{(t)} w_j^{(t)}], \quad (1)$$

where $x_j^{(t)}$, $w_j(t)$ and $y^{(t)}$ are, respectively, the value of the j th input, the j th weight and the network output at time t , while μ is the learning rate. $\mu y^{(t)} x_j^{(t)}$ is the Hebbian increment and equation (1) satisfies the condition

$$\sum_{j=1}^M (w_j^{(t)})^2 \leq 1. \quad (2)$$

Many different versions and extensions of this basic algorithm have been proposed in recent years (Sanger 1989; Karhunen & Joutsensalo 1994, hereafter KJ94; Karhunen & Joutsensalo 1995, hereafter KJ95; Oja et al. 1996).

The extension from one to more output neurons and to the hierarchical case gives the well-known generalized Hebbian algorithm (GHA) (Sanger 1989; KJ95):

$$w_{ji}^{(t+1)} = w_{ji}^{(t)} + \mu y_j^{(t)} \left[x_i^{(t)} - \sum_{k=1}^j y_k^{(t)} w_{jk}^{(t)} \right], \quad (3)$$

while the extension to the symmetric case gives Oja’s Subspace Network (Oja 1982):

$$w_{ji}^{(t+1)} = w_{ji}^{(t)} + \mu y_j^{(t)} \left[x_i^{(t)} - \sum_{k=1}^M y_k^{(t)} w_{jk}^{(t)} \right]. \quad (4)$$

In both cases the weight vectors must be orthonormalized and the algorithm stops when

$$t_n = \sqrt{\sum_{j=1}^M \sum_{i=1}^L (w_{ji}^{(t)} - w_{ji}^{(t-1)})^2} < \varepsilon,$$

where ε is an arbitrarily chosen small value. After the learning phase, the network becomes purely feedforward. KJ94 and KJ95 proved that PCA neural algorithms can be derived from optimization problems, such as variance maximization and representation error minimization. They generalized these problems to non-linear

problems, deriving non-linear algorithms (and the relative networks) having the same structure as the linear ones: either hierarchical or symmetric. These learning algorithms can be further classified as robust PCA algorithms and non-linear PCA algorithms. KJ95 defined robust PCA algorithms as those in which the objective function grows more slowly than a quadratic one. The non-linear learning function appears at selected places only. In non-linear PCA algorithms all the outputs of the neurons are non-linear functions of the responses.

More precisely, in the robust generalization of variance maximization, the objective function $f(z)$ is assumed to be a valid cost function such as $\ln \cos(z)$ or $|z|$. This leads to the adaptation step of the learning algorithm:

$$w_{ji}^{(t+1)} = w_{ji}^{(t)} + \mu g(y_j^{(t)}) e_{ji}^{(t)}, \quad (5)$$

where

$$e_{ji} = x_i^{(t)} - \sum_{k=1}^{l(j)} y_k^{(t)} w_{jk}^{(t)},$$

$$g = \frac{df}{dz}.$$

In the hierarchical case $l(j) = j$. In the symmetric case $l(j) = M$, the error vector $e_j^{(t)}$ becomes the same $e^{(t)}$ for all the neurons, and equation (5) can be compactly written as

$$\begin{aligned} \mathbf{W}^{(t+1)} &= \mathbf{W}^{(t)} + \mu(\mathbf{I} - \mathbf{W}^{(t)}\mathbf{W}^{(t)\text{T}})xg(x^{\text{T}}\mathbf{W}^{(t)}) \\ &= \mathbf{W}^{(t)} + \mu\mathbf{e}^{(t)}g(\mathbf{y}^{(t)\text{T}}), \end{aligned} \quad (6)$$

where $\mathbf{y}^{(t)\text{T}} = \mathbf{x}^{\text{T}}\mathbf{W}^{(t)}$ is the instantaneous vector of neuron responses at time t . The learning function g , derivative of f , is applied separately to each component of the argument vector.

The robust generalization of the representation error problem (KJ95) with $f(t) \leq t^2$ leads to the stochastic gradient algorithm:

$$w_j^{(t+1)} = w_j^{(t)} + \mu[w_j^{(t)\text{T}}g(e_j^{(t)})\mathbf{x}^{(t)} + \mathbf{x}^{(t)\text{T}}w_j^{(t)}g(e_j^{(t)})]. \quad (7)$$

This algorithm can again be considered in both the hierarchical and symmetric cases. In the symmetric case $l(j) = M$, the error vector is the same $e^{(t)}$ for all the weights $w^{(t)}$. In the hierarchical case $l(j) = j$, equation (7) gives the robust counterparts of the principal eigenvectors $\mathbf{c}(i)$.

In equation (7) the first update term

$$w_j^{(t)\text{T}}g(e_j^{(t)})\mathbf{x}^{(t)}$$

is proportional to the same vector $\mathbf{x}^{(t)}$ for all weights $w_j^{(t)}$. Furthermore, we can assume that the error vector $e^{(t)}$ is relatively small after the initial convergence. Hence, we can neglect the first term in equation (7), and this leads to

$$w_j^{(t+1)} = w_j^{(t)} + \mu y_j^{(t)} g(e_j^{(t)}). \quad (8)$$

Let us consider now the non-linear extensions of PCA algorithms that can be obtained in a heuristic way by requiring all neuron outputs to be always non-linear in equation (5). Then

$$w_j^{(t+1)} = w_j^{(t)} + \mu g(y_j^{(t)}) b_j^{(t)}, \quad (9)$$

where

$$b_j^{(t)} = \mathbf{x}^{(t)} - \sum_{k=1}^{l(j)} g(y_k^{(t)}) w_k^{(t)}.$$

In previous experiments (Tagliaferri et al. 1998, 1999a) we

found that the hierarchical robust NN of equation (5) with learning function $g = \tanh(ax)$ achieves the best performance with respect to all the other mentioned PCA NNs and linear PCA.

2.2 Unsupervised neural nets

Unsupervised NNs partition the input space into clusters and assign to each neuron a weight vector, which unequivocally individualizes the template characteristic of one cluster in the input feature space. After the learning phase, all the input patterns are classified.

Kohonen (1982, 1988) self organizing maps (SOM) are composed of one neuron layer structured in a rectangular grid of m neurons. When a pattern x is presented to the NN, each neuron i receives the input and computes the distance d_i between its weight vector w_i and x . The neuron that has the minimum d_i is the winner. The adaptation step consists of modifying the weights of the neurons in the following way:

$$w_j^{(t+1)} = w_j^{(t)} + \varepsilon^{(t)} h_{\sigma^{(t)}}[d(j, k)](x - w_j^{(t)}), \quad (10)$$

where $\varepsilon^{(t)}$ is the learning rate ($0 \leq \varepsilon^{(t)} \leq 1$) decreasing in time, $d(j, k)$ is the distance in the grid between the j and the k neurons and $h_{\sigma^{(t)}}(x)$ is a unimodal function with variance $\sigma^{(t)}$ decreasing with x .

The neural-gas NN is composed of a linear layer of neurons and a modified learning algorithm (Martinetz, Berkovitch & Shulten 1993). It classifies the neurons in an ordered list (j_1, \dots, j_m) according to their distance from the input pattern. The weight adaptation depends on the position $\text{rank}(j)$ of the j th neuron in the list:

$$w_j^{(t+1)} = w_j^{(t)} + \varepsilon^{(t)} h_{\sigma^{(t)}}[\text{rank}(j)](x - w_j^{(t)}), \quad (11)$$

and works better than the preceding one: in fact, it is quicker and reaches a lower average distortion value.¹

The Growing Cell Structure (GCS) (Fritzke 1994) is a NN which is capable of changing its structure depending on the data set. The aim of the net is to map the input pattern space into a two-dimensional discrete structure S in such a way that similar patterns are represented by topological neighbouring elements. The structure S is a two-dimensional simplex where the vertices are the neurons and the edges attain the topological information. Every modification of the net always maintains the simplex properties. The learning algorithm starts with a simple three-node simplex and tries to obtain an optimal network by a controlled growing process, i.e. for each pattern x of the training set, the winner k and the weights of the neighbours are adapted as follows:

$$w_k^{(t+1)} = w_k^{(t)} + \varepsilon_b(x - w_k^{(t)}); \quad w_j^{(t+1)} = w_j^{(t)} + \varepsilon_n(x - w_j^{(t)}) \quad (12)$$

$\forall j$ connected to k , where ε_b and ε_n are constants which determine the adaptation strength for the winner and for the neighbours, respectively.

The insertion of a new node is made after a fixed number λ of adaptation steps. The new neuron is inserted between the most frequent winner neuron and the more distant of its topological neighbours. The algorithm stops when the network reaches a pre-defined number of elements.

¹ Let $P(x)$ be the pattern probability distribution over the set $V \subseteq \mathfrak{R}^n$ and let $w_i(x)$ be the weight vector of the neuron that classifies the pattern x . The average distortion is defined as $E = \int P(x)[x - w_i(x)]^2 dx$.

The online K-means clustering algorithm (Lloyd 1982) is a simpler algorithm which applies the Gradient Descent (GD) directly to the average distortion function as follows:

$$\mathbf{w}_j^{(t+1)} = \mathbf{w}_j^{(t)} + \varepsilon^{(t)}(\mathbf{x} - \mathbf{w}_j^{(t)}). \quad (13)$$

The main limitation of this technique is that the error function presents many local minima, where the learning could stop before reaching the optimal configuration.

Finally, the Maximum Entropy NN (Rose, Gurewitz & Fox 1990) applies the GD to the error function to obtain the adaptation step:

$$\mathbf{w}_j^{(t+1)} = \mathbf{w}_j^{(t)} + \varepsilon^{(t)} \frac{\exp(-\beta^{(t)} d_j)}{\sum_{k=1}^m \exp(-\beta^{(t)} d_k)} (\mathbf{x} - \mathbf{w}_j^{(t)}), \quad (14)$$

where β is the inverse temperature and takes a value increasing in time and d_j is the distance between the j th and the winner neurons.

2.3 Hybrid neural nets

Hybrid NNs are composed of a clustering algorithm which makes use of the information derived by one unsupervised single-layer NN. After the learning phase of the NN, the clustering algorithm splits the output neurons into a number of subsets, equal to the number of the desired output classes. As the aim is to put similar input patterns in the same class and dissimilar input patterns in different classes, a good strategy consists of applying a clustering algorithm directly to the weight vectors of the unsupervised NN.

A non-neural agglomeration clustering algorithm that divides the pattern set (in this case the weights of the neurons) $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_m\}$ into l clusters (with $l < m$) can be briefly summarized as follows:

- (i) initially it divides \mathbf{W} into m clusters C_1, \dots, C_m such that $C_j = \{\mathbf{w}_j\}$;
- (ii) it then computes the distance matrix \mathbf{D} with elements $D_{ij} = d(C_i, C_j)$;
- (iii) it then finds the smallest element D_{ij} and unifies the clusters C_i and C_j into a new one $C_{ij} = C_i \cup C_j$;
- (iv) if the number of clusters is greater than l then it goes back to step 2, otherwise it finally stops.

Many algorithms quoted in the literature (Everitt 1977) differ only in the way in which the distance function is computed. For example,

$$d(C_i, C_j) = \min_{\substack{w_{ik} \in C_i \\ \text{and} \\ w_{jl} \in C_j}} \|w_{ik} - w_{jl}\|$$

(nearest neighbour algorithm);

$$d(C_i, C_j) = \left\| \frac{1}{|C_i|} \sum_{w_{ik} \in C_i} w_{ik} - \frac{1}{|C_j|} \sum_{w_{jl} \in C_j} w_{jl} \right\|$$

(centroid method);

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{w_{ik} \in C_i, w_{jl} \in C_j} \|w_{ik} - w_{jl}\|$$

(average between groups).

The output of the clustering algorithm will be a labelling of the patterns (in this case neurons) in l different classes.

2.4 Unsupervised hierarchical neural nets

Unsupervised hierarchical NNs add one or more unsupervised single-layer NNs to any unsupervised NN, instead of a clustering algorithm as is the case for hybrid NNs.

In this way, the second-layer NN learns from the weights of the first-layer NN and clusters the neurons on the basis of a similarity measure or a distance. The iteration of this process to a few layers gives the unsupervised hierarchical NN.

The number of neurons in each layer decreases from the first to the output layer and, as a consequence, the NN takes the pyramidal aspect shown in Fig. 3. The NN takes as input a pattern \mathbf{x} and then the first layer finds the winner neuron. The second layer takes the first-layer winner weight vector as input and finds the second-layer winner neuron and so on up to the top layer. The activation value of the output layer neurons is 1 for the winner unit and 0 for all the others. In short: the learning steps of a s -layer hierarchical NN with training set \mathbf{X} are the following:

- (i) the first layer is trained on the patterns of \mathbf{X} with one of the learning algorithms for unsupervised NNs;
- (ii) the second layer is trained on the elements of the set \mathbf{X}_2 , which is composed of the weight vectors of the first-layer winner units;
- (iii) the process is iterated to the i th layer NN ($i > 2$) on the training set, which is composed of the weight vectors of the winner neurons of the $(i - 1)$ th layer when presenting \mathbf{X} to the first layer NN, \mathbf{X}_2 to the second layer and so on.

By varying the learning algorithms, we obtain different NNs with different properties and abilities. For instance, by using only SOMs we have a multilayer SOM (ML-SOM) (Koh, Suk & Bhandarkar 1995) where every layer is a two-dimensional grid. We can easily obtain (Tagliaferri, Capuano & Gargiulo 1999a) *ML-NeuralGas*, *ML-Maximum Entropy* or *ML-K means* organized on a hierarchy of linear layers. The ML-GCS has a more complex architecture and has at least three units per layer.

By varying the learning algorithms in the different layers, we can take advantage of the properties of each model (for instance, because we cannot have a ML-GCS with two output units we can use another NN in the output layer).

A hierarchical NN with a number of output layer neurons equal to the number of the output classes simplifies the expensive

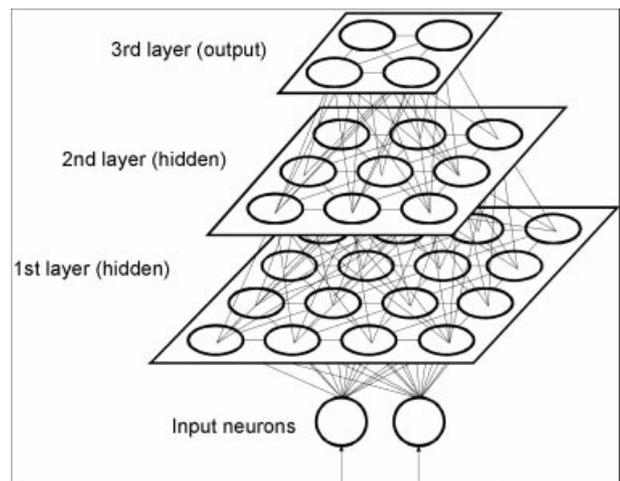


Figure 3. The structure of a hierarchical unsupervised NN.

post-processing step of labelling the output neurons in classes, without reducing the generalization capacity of the NN.

2.5 Multilayer perceptron

A *Multilayer Perceptron* (MLP) is a layered NN composed of

- (i) one input layer of neurons which transmit the input patterns to the first hidden layer;
- (ii) one or more hidden layers with units computing a non-linear function of their inputs;
- (iii) one output layer with elements calculating a linear or a non-linear function of their inputs.

The aim of the network is to minimize an error function, which generally is the sum of squares of the difference between the desired output (target) and the output of the NN. The learning algorithm is called back-propagation because the error is back-propagated in the previous layers of the NN in order to change the weights. In formulae, let \mathbf{x}^p be an L -dimensional input vector with corresponding target output c^p . The error function is defined as follows:

$$E_p = \frac{1}{2} \sum_i (c_i^p - y_i^p)^2,$$

where y_i^p is the output of the i th output neuron. The learning algorithm updates the weights by using the gradient descent (GD) of the error function with respect to the weights. We can define the input and the output of the neuron j respectively as

$$\text{net}_j^p = \sum_i w_{ji} y_i^p$$

and

$$y_j^p = f(\text{net}_j^p),$$

where w_{ji} is the connection weight from the neuron i to the neuron j , and $f(x)$ is linear or sigmoidal for the output nodes and sigmoidal for the hidden nodes. It is well known in the literature (Bishop 1995) that these facts lead to the following adaptation steps:

$$w_{ji}^{(t+1)} = w_{ji}^{(t)} + \Delta w_{ji}^{(t)} \quad \text{where} \quad \Delta w_{ji}^{(t)} = \eta \delta_j^p y_i^p \quad (15)$$

and

$$\delta_j^p = (c_j^p - y_j^p) y_j^p (1 - y_j^p) \quad \text{or} \quad \delta_j^p = (c_j^p - y_j^p) \sum_k \delta_k^p w_{kj} \quad (16)$$

for the output and hidden units, respectively. The value of the learning rate η is small and causes a slow convergence of the algorithm. A simple technique often used to improve it is to sum a momentum term to equation (15), which becomes

$$\Delta w_{ji}^{(t)} = \eta \delta_j^p y_i^p + \mu \Delta w_{ji}^{(t-1)}. \quad (17)$$

This technique generally leads to a significant improvement in the performances of GD algorithms but it introduces a new parameter μ which has to be empirically chosen and tuned.

Bishop (1995) and Press et al. (1993) summarize several methods to overcome the problems related to the local minima and to the slow time convergence of the above algorithm. In a preliminary step of our experiments, we tried all the algorithms discussed in chapter 7 of Bishop (1995), finding that a hybrid algorithm based on the scaled conjugate gradient for the first steps and on the Newton method for the next ones gives the best results with respect to both computing time and relative number of errors. In this paper we used it in the MLP experiments.

3 THE EXPERIMENTS

3.1 The data

In this work we use a $2000 \times 2000 \text{ arcsec}^2$ area centred on the North Galactic Pole extracted from the slightly compressed POSS-II F plate n. 443, available via the World Wide Web at the Canadian Astronomy Data Center (<http://cadwww.dao.nrc.ca>). Fig. 4 shows the studied field. POSS-II data were linearized using the sensitometric spots recorded on the plate. The seeing full width at half-maximum (FWHM) of our data was 3 arcsec. The same area has been widely studied by others and, in particular, by Infante & Pritchett (1992, hereafter IP92) and Infante, Pritchett & Hertling (1995), who used deep observations obtained at the 3.6-m CFHT telescope in the F photographic band under good seeing conditions (FWHM < 1 arcsec) to derive a catalogue of objects complete down to $m_F \sim 23$, i.e. much deeper than the completeness limit of our plate. Their catalogue is therefore based on data of much better quality and accuracy than ours, and it was because of the availability of such a good template that we decided to use this region for our experiments. We also studied a second region in the Coma cluster (which happens to be in the same n. 443 plate) but, because none of the catalogues available in the literature is much better than our data, we were forced to neglect it in most of the following discussion.

The characteristics of the selected region, a relatively empty one, slightly penalize our NN detection algorithms, which can easily recognize objects of quite different sizes. In contrast to what happens to other algorithms, NEXt works well even in areas where both very large and very small objects are present such as, for instance, the centres of nearby clusters of galaxies, as our preliminary test on a portion of the Coma clusters clearly shows (Tagliaferri et al. 1998).

3.2 Structure of NEXt

The detection and classification of the objects is a multistep task.

(1) First of all, following a widely used AI approach, we mathematically transform the detection task into a classification one by compressing the redundant information contained in nearby pixels by means of a non-linear PCA NN. Principal vectors of the PCA are computed by the NN on a portion of the whole image. The values of the pixels in the transformed M -dimensional eigenspace obtained via the principal vectors of the PCA NN are then used as inputs to unsupervised NNs to classify pixels into a few classes. We wish to stress that, in this step, we are still classifying pixels, and not objects.

The adopted NN is unsupervised, i.e. we never feed into the detection algorithm any a priori definition of what an object is, and we leave it free to find its own object definition. It turns out that image pixels are split into a few classes, one coincident with what astronomers call background and some others for the objects (in the astronomical sense). Afterwards, the class containing the background pixels is kept separated from the other classes, which are instead merged together. Therefore, as final output, the pixels in the image are divided into ‘object’ or ‘background’.

(2) Since objects are seldom isolated in the sky, we need a method to recognize overlapping objects and deblend them. We adopt a generalization of the method used by Focas (Jarvis & Tyson 1981).

(3) Owing to the noise, object edges are quite irregular. We

therefore apply a contour regularization to the edges of the objects in order to improve the following star/galaxy classification step.

(4) We define and measure the features used, or suitable, for the star/galaxy classification, then we choose the best performing features for the classification step, through the sequential backward elimination strategy (Bishop 1995).

(5) We then use a subset of the IP92 catalogue to learn, validate and test the classification performed by NEXt on our images. The training set was used to train the NN, while the validation was used for model selection, i.e. to select the most performing parameters using an independent data set. As template classifier, we used SEX, the classifier for which is also based on NNs.

The detection and classification performances of our algorithm were then compared with those of traditional algorithms, such as SEX. We wish to stress that, in both the detection and classification phases, we were not interested in knowing how well NEXt can reproduce SEX or the performance of an astronomer's eye, but rather to see whether the SEX and NEXt catalogues are or are not similar to the 'true' one, represented in our case by the IP92 catalogue.

Finally, we would like to stress that in statistical pattern recognition, one of the main problems in evaluating the system performances is the optimization of all the compared systems in order not to give any unfair advantage to one of the systems with respect to the others (just because it is better optimized than the

others). For instance, because the background subtraction is crucial to the detection, all algorithms, including SEX, were run on the same background-subtracted image.

3.3 Segmentation

From the astronomical point of view, segmentation allows us to disentangle objects from a noisy background. From a mathematical point of view, instead, the segmentation of an image F consists of splitting it into disconnected homogeneous (according to a uniformity predicate P) regions $\{S_1, \dots, S_n\}$, in such a way that their union is not homogeneous:

$$\bigcup_{i=1}^n S_i = F \quad \text{with} \quad S_i \cap S_j = \emptyset, i \neq j,$$

where $P(S_i) = \text{true} \forall i$ and $P(S_i \cup S_j) = \text{false}$ when S_i is adjacent to S_j . The two regions are adjacent when they share a boundary, i.e. when they are neighbours.

A segmentation problem can be easily transformed into a classification one if classes are defined on pixels and P is written in such a way that $P(S_i) = \text{true}$ if and only if all the pixels of S_i belong to the same class. For instance, the segmentation of an astronomical image into background and objects leads us to assign each pixel to one of the two classes. Among the various methods discussed in the literature, unsupervised NNs usually provide

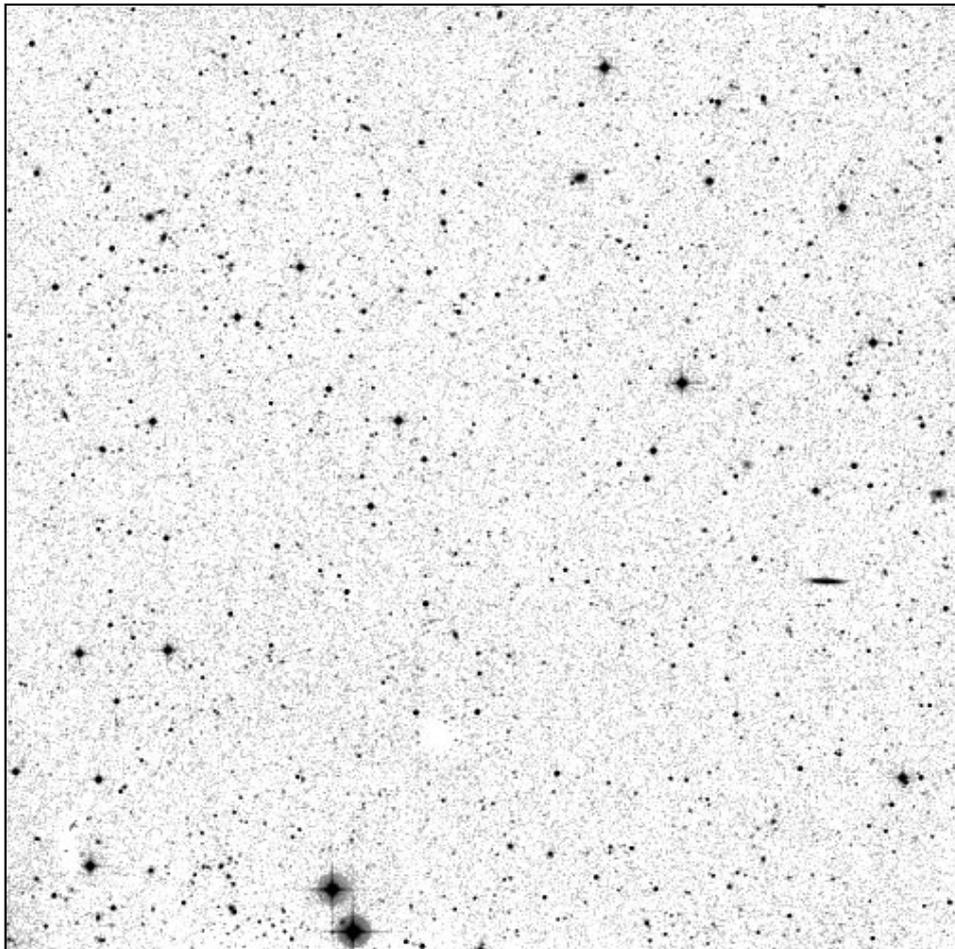


Figure 4. The studied field. The field is 2000×2000 arcsec wide. North is up and east is left. The image is binned at 4×4 .

better performance than any other NN-type on noisy data (Pal & Pal 1993) and have the great advantage of not requiring a definition (or exhaustive examples) of an ‘object’.

The first step of the segmentation process consists of creating a numerical mask where different values discriminate between the background and the object (Fig. 5).

In well-sampled images, the attribution of a pixel to either the background or the object classes depends on both the pixel value

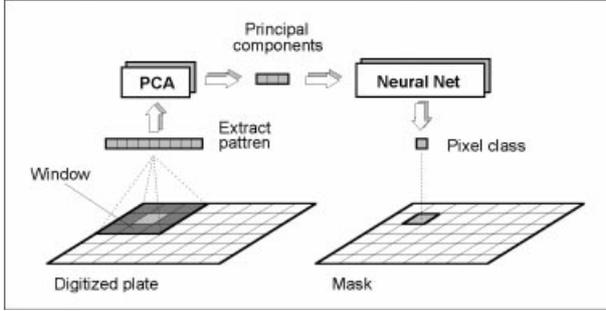


Figure 5. Single-band segmentation: process scheme.

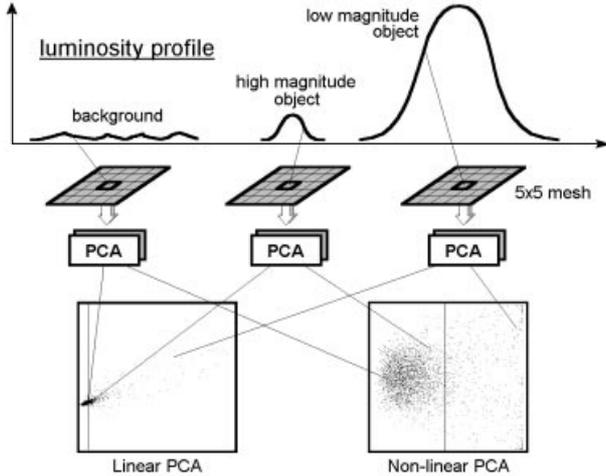


Figure 6. Illustration of the coefficients distribution in two dimensions of the three-dimensional eigenvector space of the input patterns. Non-linear PCA NNs are the best performing because the spatial distributions obtained in the input transformed space are more spread out, as shown in the example.

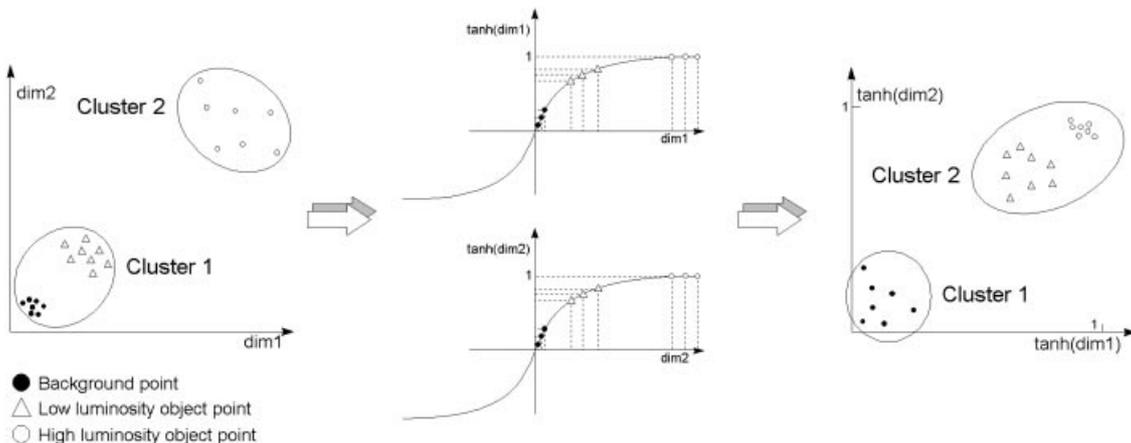


Figure 7. Clustering before and after the application of the tanh function in the learning of the PCA NN.

and the properties of its neighbours: for instance, a ‘bright’ isolated pixel in a ‘dark’ environment is usually just noise. Therefore, in order to classify a pixel, we need to take into account the properties of all the pixels in a $(n \times n)$ window centred on it. This approach can be easily extended to the case of multiband images. $n \times n$, however, is too high a dimensionality to be effectively handled (in terms of learning and computing time) by any classification algorithm. Therefore, in order to lower the dimensionality, we first use a PCA to identify the M (with $M \ll n \times n$) most significant features. Details are as follows.

(i) We first run the $(n \times n)$ window on a subimage containing representative parts of the image. We used both 3×3 and 5×5 windows.

(ii) Then we train the PCA NNs on these patterns. The result is a projection matrix \mathbf{W} with dimensionality $(n \times n) \times M$, which allows us to reduce the input feature number from $(n \times n)$ to M . We considered only the first three components because, according to the PCA, they contain almost 93 per cent of the information, while the remaining 7 per cent is distributed over all the other components.

(iii) The M -dimensional projected vector $\mathbf{W} \cdot \mathbf{I}$ is the input of a second NN which classifies the pixels in the various classes.

(iv) Finally, we merge all classes except the background one in order to reduce the classification problem to the usual ‘object/background’ dichotomy.

Much attention also has to be paid to the choice of the type of PCA. After several experiments, we found that – for our specific task, which is characterized by a large dynamical range in the luminosities of the objects (or, equivalently, in the pixel values) – PCAs can be split into two large groups: PCAs with linear input–output mapping (hereafter linear PCA NNs) and PCAs with non-linear input–output mapping (non-linear PCA NNs) (see Section 2.1). Linear PCA NNs were found to misclassify faint objects as background. Non-linear PCA NNs based on a sigmoidal function allowed, instead, the detection of faint sources. This can be better understood from Figs 6 and 7, which give the distributions of the training points in the simpler case of two-dimensional inputs for the two types of PCA NNs.

Linear PCA NNs produce distributions with a very dense core (background and faint objects) and only a few points (luminous objects) spread over a wide area. Such behaviour results from the presence of very luminous objects in the training set, which compress the faint ones to the bottom of the scale. This problem

Table 1. First transposed eigenvector 3×3 .

1	1	1
1	1	1
1	1	1

Table 2. Second transposed eigenvector 3×3 .

-2	-4	-4
1	0	1
4	4	2

Table 3. Third transposed eigenvector 3×3 .

4	0	-3
4	0	-4
3	0	-4

Table 4. First transposed eigenvector 5×5 .

0.184082	0.196004	0.192174	0.172190	0.135904
0.207895	0.225043	0.223247	0.202225	0.161833
0.216280	0.236501	0.236244	0.215310	0.173737
0.207416	0.228742	0.229733	0.210264	0.170427
0.181968	0.202628	0.204979	0.188628	0.153777

Table 5. Second transposed eigenvector 5×5 .

0.333702	0.313831	0.211281	0.095904	-0.015149
0.340559	0.244218	0.181150	0.022373	-0.018147
0.230683	0.121065	0.006941	-0.130654	-0.200468
0.052375	-0.053280	-0.130905	-0.292818	-0.308128
-0.039431	-0.078693	-0.241601	-0.257146	-0.256896

Table 6. Second transposed eigenvector 5×5 .

0.043911	-0.140093	-0.208548	-0.245815	-0.308660
0.114738	-0.053939	-0.109611	-0.273351	-0.340794
0.239802	0.210806	0.015948	-0.166681	-0.230214
0.300927	0.256129	0.042729	-0.009986	-0.113216
0.322900	0.244557	0.165326	0.004180	-0.122482

can be circumvented by avoiding very luminous objects in the training set, but this would make the whole procedure too much dependent on the choice of training set.

Non-linear PCA NNs, instead, produce better sampled distributions and a better contrast between background and faint objects. The sigmoidal function compresses the dynamical range, squeezing the very luminous objects into a narrow region (see Fig. 7).

Overall, the best performing NN (Tagliaferri et al. 1998) turned out to be the hierarchical robust PCA NN with learning function $g^{(l)} = \tanh(ax)$ given in equation (5). This NN was also the fastest among the non-linear PCA NNs.

The principal component matrices are detailed in Tables 1–3 and 4–6 for the 3×3 and 5×5 cases, respectively. In Tables 1–3, numbers are rounded to the closest integer because they differ from an integer only at the 7th decimal place. Not surprisingly, the first component turns out to be the mean in the 3×3 case. The other two matrices can be seen as antisymmetric filters with

respect to the centre. The convolution of these filters (see Fig. 8) with the input image gives images where the objects are the regions of high contrast. Similar results are obtained for the 5×5 case.

At this stage we have the principal vectors and, for each pixel, we can compute the values of the projection of each pixel in the eigenvector space. The second step of the segmentation process consists of using unsupervised NNs to classify the pixels into a few classes, having as input the reduced input patterns which have been just computed. Supervised NNs would require a training set specifying, for each pixel, whether that pixel belongs to an object or to the background. We no longer consider such a possibility, because of the arbitrariness of such a choice at low fluxes, the lack of elegance of the method and the problems that are encountered in the labelling phase. Unsupervised NNs are therefore necessary. We considered several types of NNs.

As already mentioned several times, our final goal is to classify the image pixels into just two classes: objects and background, which should correspond to two output neurons. This simple model, however, seldom suffice to reproduce real data in the bidimensional case (but similar results are obtained also for the three-dimensional or multidimensional cases), because any unsupervised algorithm fails to produce spatially well-separated clusters and more classes are needed. A trial and error procedure shows that a good choice of classes is 6: fewer classes produce poor classifications while more classes produce noisy ones. In all cases, only one class (containing the lowest luminosity pixels) represents the background, while the other classes represent different regions in the object images.

We compared hierarchical, hybrid and unsupervised NNs with 6 output neurons. From theoretical considerations and from preliminary work (Tagliaferri et al. 1998) we decided to consider only the best performing NNs, i.e. neural gas, ML–neural gas, ML–SOM and GCS + ML–neural gas. For a more quantitative and detailed discussion see Section 3.6, where the performance of these NNs is evaluated.

After this stage all pixels are classified into one of six classes. We merge together all classes with the exception of the background one, and reduce the classification to the usual astronomical dichotomy: object or background.

Finally, we create the masks, each one identifying one structure composed of one or more objects. This task is accomplished by a simple algorithm, which, while scanning the image row by row, when it finds one or more adjacent pixels belonging to the object class expands the structure including all equally labelled pixels adjacent to them.

Once objects have been identified we measure a first set of parameters. These are as follows. The photometric barycentre of the objects is computed as

$$\bar{x} = \frac{\sum_{(x,y) \in A} x \cdot I(x,y)}{\text{flux}} \quad \text{and} \quad \bar{y} = \frac{\sum_{(x,y) \in A} y \cdot I(x,y)}{\text{flux}},$$

where A is the set of pixels assigned to the object in the mask, $I(x,y)$ is the intensity of the pixel (x,y) , and

$$\text{flux} = \sum_{(x,y) \in A} I(x,y)$$

is the flux of the object integrated over the considered area. The semimajor axis of the object contour is defined as

$$a = \max_{(x,y) \in A} \|(x,y) - (\bar{x}, \bar{y})\| = \max_{(x,y) \in A} r(x,y),$$

with position angle defined as

$$\alpha = \arctan\left(\frac{y' - \bar{y}}{x' - \bar{x}}\right),$$

where (x', y') is the most distant pixel from the barycentre belonging to the object. The semiminor axis of the faintest isophote is given by

$$b = \max_{(x,y) \in A} \left| \sin \left[\arctan\left(\frac{y - \bar{y}}{x - \bar{x}}\right) - \alpha \right] \right| \cdot r(x, y).$$

These parameters are needed in order to disentangle overlapping objects.

3.4 Object deblending

Our method recognizes multiple objects by the presence of multiple peaks in the light distribution. The search for double peaks is performed along directions at position angles $\beta_i = \alpha + i\pi/n$ with $0 \leq i \leq n$. In contrast to FOCAS (Jarvis & Tyson 1981), we sample several position angles because objects are not

always aligned along the major axis of their light distribution, as FOCAS implicitly assumes. In our experiments the maximum n was set to 5. When a double peak is found, the object is split into two components by cutting it perpendicularly to the line joining the two peaks.

Spurious peaks can also be produced by noise fluctuations, a case which is very common in photographic plates near saturated objects. A good way to minimize such noise effects is, just for deblending purposes, to reduce the dynamical range of the pixel values, by rounding the intensity (or pixel values) into N equispaced levels.

Multiple (i.e. 3 or more components) objects pose a more complex problem. In the case shown in Fig. 9, the segmentation mask includes three partially overlapping sources. The search for double peaks produces a first split of the mask into two components which separate the third and faintest component into two fragments. Subsequent iterations would usually produce a set of four independent components, therefore introducing a spurious detection. In order to solve the problem posed by multiple ‘non-spurious’ objects erroneously split, a recomposition loop needs to be run. Most celestial objects – it does not matter

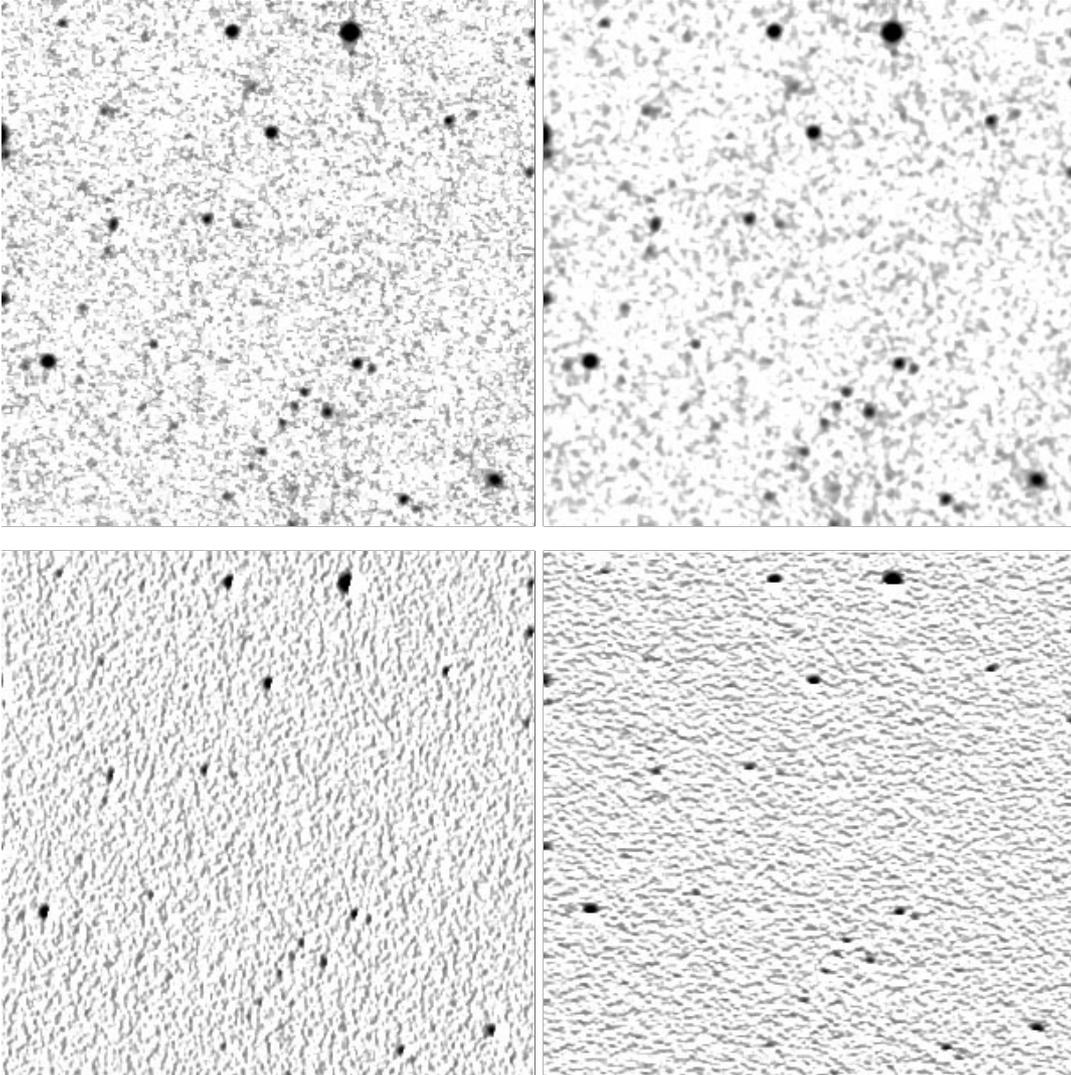


Figure 8. A portion 300×300 of the original image (top left). The same image convolved with the principal eigenvector matrices: first, second and third eigenvectors (top right, bottom left and bottom right, respectively).

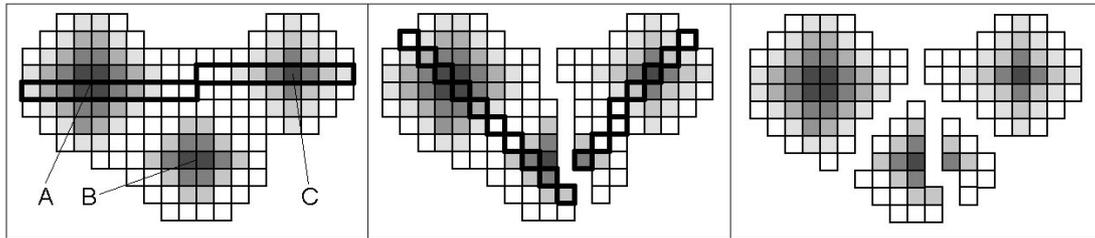


Figure 9. Deblending of three partially overlapping sources. In the left panel, the original image is plotted. In the central and right panels, we show the result after the first and the second steps. The correct result is achieved after a recomposition loop.

whether they are resolved or unresolved – present a light distribution rapidly decreasing outwards from the centre. If an object has been erroneously split into several components, then the adjacent pixels on the corresponding sides of the two masks will have very different values. The implemented algorithm checks each component (starting from the one with the highest average luminosity and proceeding to the fainter ones) against the others. Let us now consider two parts of an erroneously split object. When the edge pixels have a luminosity higher than the average luminosity of the faintest component, the two parts are recomposed. This procedure also takes care of all spurious components produced by the haloes of bright objects (an artefact which is a major shortcoming of many packages available in the astronomical community).

3.5 Contour regularization

The last operation before measuring the parameters of the objects consists of the regularization of the contours because, owing to noise, overlapping images, image defects, etc., segmentation produces patterns that are not similar to the original celestial objects that they must represent. For the contour regularization, we threshold the image at several sigma over the background and we then expand the ellipse describing the objects in order to include the whole area measured in the object detection.

3.6 Results on the object detection phase

After the above-described steps, it becomes possible to measure and compare the performances of the various NN models. We implemented and compared neural gas (NG3), ML–neural gas (MLNG3 or MLNG5), ML–SOM (K5) and GCS + ML–neural gas (NGCS5), where the last digit in the NN name indicates the dimensions of the running window.

Attention was paid to choosing the training set, which needed to be at the same time small but significant. By trial and error, we found that for PCA NNs and unsupervised NNs it was enough to choose ~ 10 subimages, each one $\sim 50 \times 50$ pixels wide and not containing very large objects. As all the experienced users know, the choice of the SEX parameters (minimum area, threshold in units of the background noise and deblending parameter) is not critical and the default values were chosen (4 pixel area, 1.5σ).

Table 7 shows the number of objects detected by the five NNs and SEX. It has to be stressed that ~ 2100 objects out of the 4819 available in the IP92 reference catalogue are beyond the detection limit of our plate material. SEX detects a larger number of objects but many of them (see Table 7) are spurious. NNs detect a slightly smaller number of objects but most of them are real. In particular,

Table 7. Number of objects grouped into ‘Total’, ‘True’ and ‘False’ detections integrated over the whole magnitude range. The reference catalogue consists of 4819 objects, among which ~ 2400 are too faint to be visible on our plate material.

Catalogues	Total	Detections	
		‘True’ objects	‘False’ objects
K5	1942	1738	204
MLNG3	2742	2059	683
MLNG5	3776	2310	1466
NG3	1584	1477	107
NGCS5	1862	1692	170
SEX	4256	2388	1866

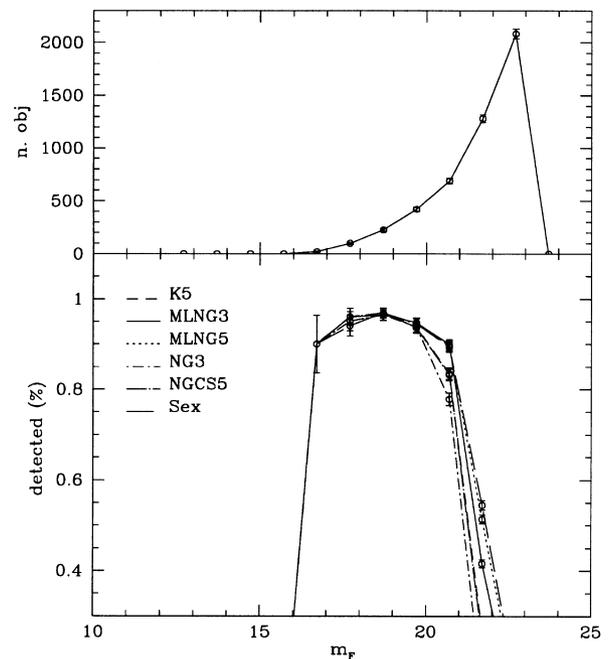


Figure 10. Upper panel: number of ‘True’ objects in the reference catalogue; lower panel: number of objects detected by a given NN and by SEX relative to the total number.

MNG5 loses, with respect to SEX, only 79 real objects but detects 400 fewer spurious objects; MNG3 has a slightly poorer performance in detecting true objects but is even freer of spurious detections.

The upper panel of Fig. 10 shows the number of ‘True’ objects (i.e. objects in the IP92 catalogue). Most of them are fainter than $m_F \sim 21.5$ mag, i.e. they are fainter than the plate limit. The lower panel shows instead the number of objects detected by the various

NNs relative to SEX. The curves largely coincide and, in particular, MLNG5 and SEX do not statistically differ in any magnitude bin while MLNG3 slightly differs only in the faintest bin ($m_F \sim 21.5$).

The class of ‘Missed’ objects (i.e. objects which are listed in the reference catalogue but are not in the NNs or SEX catalogues) needs a detailed discussion. We focus first on brighter objects. They can be divided into the following classes.

(i) A few ‘True’ objects with a nearby companion which are blended in our image but are resolved in IP92.

(ii) Parts of isolated single large objects incorrectly split by IP92 (a few cases).

(iii) A few detections aligned in the east–west direction on the two sides of the images of a bright star. They are likely false objects (diffraction spikes detected as individual objects in the IP92 catalogue).

(iv) Objects in IP92 that correspond to empty regions in our images: they can be missing because they are variable, fast-moving or with an overestimated luminosity in the reference catalogue; they can also be missed because they are spurious in the template catalogue.

Therefore, a fair fraction of the ‘Missed’ objects is truly non-existent and the performances of our detection tools are lower bounded at $m_F < 21$ mag. We wish to stress here that even though there is no such thing as a perfect catalogue, the IP92 template is among the best ever produced, to our knowledge.

The upper panel of Fig. 11 is the same as in Fig. 10. The lower panel shows instead the fraction of ‘false’ objects, i.e. the objects detected by the algorithms but not present in the reference catalogue. IP92 concentrated on faint objects and masked out the bright ones, therefore their catalogue may exclude a few ‘True’ objects (in particular at $m_F \sim 17$). We believe that all objects brighter than $m_F = 20$ mag are really ‘True’ because they are detected both by SEX and NNs with high significance. For objects brighter than $m_F = 20$ mag, the NNs and SEX have similar performance. They differ only at fainter magnitudes. The catalogue with the largest contamination by ‘False’ objects is SEX, followed by MLNG5, MLNG3 and the other NNs, which are much less contaminated. MLNG5 is quite efficient in detecting ‘True’ objects and has a 20 per cent cleaner detection rate in the highly populous bin $m_F = 21.7$ mag. MLNG3 is less efficient than MLNG5 in detecting ‘True’ objects but it is even cleaner than MLNG5 as regards false detections.

Let us now consider whether or not the detection efficiency depends on the degree of concentration of the light (stars have steeper central gradients than galaxies). In IP92 objects are classified into two major classes, stars and galaxies, and a few minor ones (merged, noise, spike, defects, etc.) that we neglect. The efficiency of the detection is shown in Fig. 12 for three representative detection algorithms: MLNG5, K5 and SEX. At $m_F < 21$ mag, the detection efficiency is large, close to 1 and independent of the central concentration of the light. Please note that there are no objects in the image having $m_F < 16$ mag and that in the first bin there are only four galaxies. At fainter magnitudes (~ 22 – 23 mag), detection efficiencies differ as a function of both the algorithm and the light concentration. In fact, SEX, MLNG5 and to a lesser extent K5, turn out to be more efficient in detecting galaxies than stars (in other words, ‘Missed’ objects are preferentially stars). For SEX, a possible explanation is that a minimal area above the background is required in order for the object to be detected at $m_F \sim 22$ – 23 mag

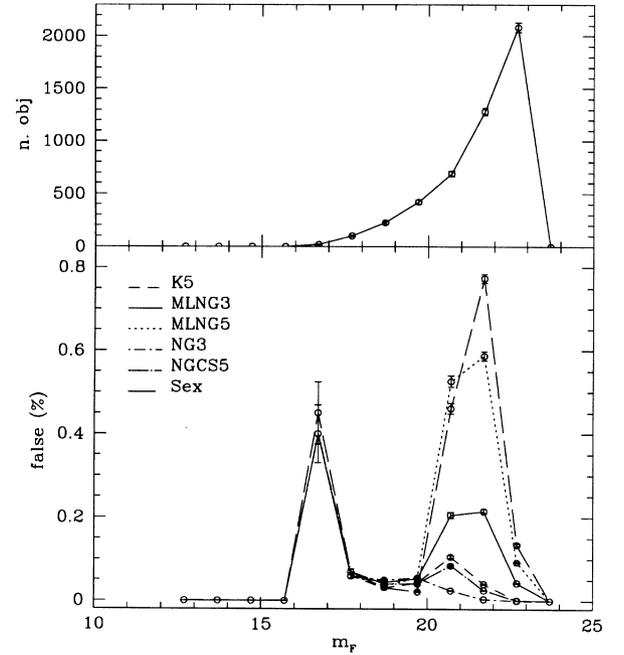


Figure 11. Upper panel: as in Fig. 10; lower panel: fraction of ‘False’ objects detected by the algorithms and not present in the reference catalogue (IP92).

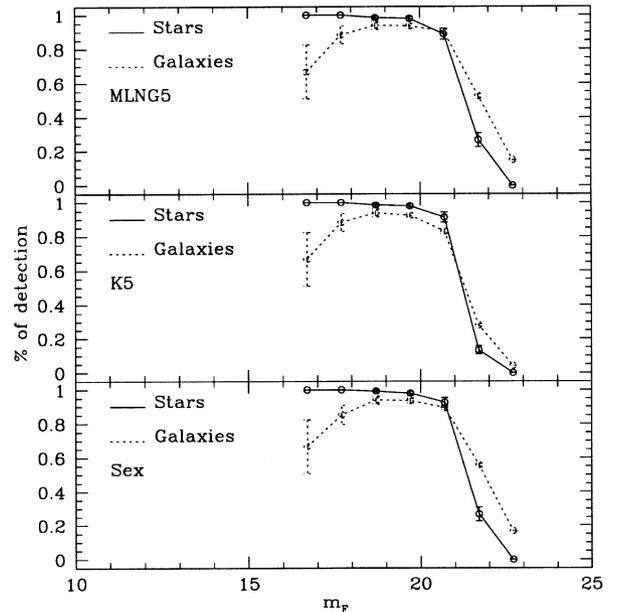


Figure 12. The percentage number of objects detected by MLNG5, K5 and SEX.

and noise fluctuations can affect the isophotal area of unresolved objects, bringing it below the assumed threshold (4 pixels). This bias is minimum for the K5 NN. However, this is more likely a result of the fact that K5 misses more galaxies than the other algorithms, rather than the fact that it detects more stars.

In conclusion, MLNG3 and MLNG5 turn out to have high performance in detecting objects: they produce catalogues that are freer of false detections at the price of a slightly larger

incompleteness than the SEX catalogues below the plate completeness magnitude.

We also want to stress that because the poorer performing NNs produce catalogues that are much freer of false detections, the selected objects are in large part ‘true’, and not just noise fluctuations. These NNs may therefore be very suitable to select candidates for possible follow-up detailed studies at magnitudes where many of the objects detected by SEX would be spurious. Deeper catalogues having a large number of spurious sources, such as those produced by SEX or other packages, are instead preferable if, for instance, they can be cleaned by subsequent processing (for instance by matching the detected objects with other catalogues).

A posteriori, one could argue that a performance similar to that of each of the NNs could be achieved by running SEX with appropriate settings. However, it would be unfair (and methodologically wrong) to make a fine tuning of any of the detection algorithms using a posteriori knowledge. It would also make cumbersome the automatic processing of the images that is the final goal of our procedure.

3.7 Feature extraction and selection

In this section we discuss the feature extraction and selection of the features that are useful for the star/galaxy classification. Features are chosen from the literature (Godwin & Peach 1977; Jarvis & Tyson 1981; Odewahn et al. 1992, hereafter O92; Miller & Coe 1996), and then selected by a sequential forward selection process (Bishop 1995), in order to extract the most performing ones for classification purposes.

The first five features are those defined in the previous section and describing the ellipses circumscribing the objects: the photometric barycentre coordinates (\bar{x}, \bar{y}) , the semimajor axis (a), the semiminor axis (b), and the position angle (α). The sixth one is the object area, A , i.e. the number of pixels forming the object.

The next twelve features have been inspired by the pioneering work of O92: the object diameter ($\text{dia} = 2a$); the ellipticity ($\text{ell} = 1 - b/a$); the average surface brightness [$\langle \text{SuBr} \rangle = (1/A) \times \sum_{(x,y) \in A} I(x,y)$]; the central intensity ($I_0 = I(\bar{x}, \bar{y})$); the filling factor ($f_{\text{fac}} = \pi ab/A$); the area logarithm [$c_2 = \log(A)$]; the harmonic radius (r_{-1}), which is defined as

$$r_{-1} = \frac{1}{\text{flux}} \sum_{(x,y) \in A} \frac{I(x,y)}{r(x,y)};$$

and five gradients G_{14} , G_{13} , G_{12} , G_{23} and G_{34} , defined as

$$G_{ij} = \frac{T_j - T_i}{r_i - r_j},$$

where T_i is the average surface brightness within an ellipse, with position angle α , semimajor axis $r_i = ia/4$, $i = 1, \dots, 4$, and ellipticity ell .

Two more features are added following Miller & Coe (1996): the ratios $T_r = \langle \text{SuBr} \rangle / I_0$ and $T_{cA} = I_0 / \sqrt{A}$.

Finally, five FOCAS features (Jarvis & Tyson 1981) have been included: the second (C_2) and the fourth (C_4) total moments, defined as

$$C_2 = \frac{M_{20} + M_{02}}{M_{00}} \quad \text{and} \quad C_4 = \frac{M_{40} + 2M_{22} + M_{04}}{M_{00}},$$

where M_{ij} are the object central momenta, computed as

$$M_{ij} = \sum_{(x,y) \in A} (x - \bar{x})^i (y - \bar{y})^j I(x,y);$$

the average ellipticity,

$$E = \frac{\sqrt{(M_{20} - M_{02})^2 + M_{11}^2}}{M_{02} + M_{20}};$$

the central intensity averaged in a 3×3 area; and, finally, the ‘Kron’ radius, defined as

$$r_{\text{Kron}} = \frac{1}{\text{flux}} \sum_{(x,y) \in A} I(x,y) r(x,y).$$

For each object we therefore measure 25 features, where the first 6 are reported only to ease the graphical representation of the objects and have a low discriminating power. The complete set of the extracted features is given in Table 8.

Our list of features therefore includes most of those usually used in the astronomical literature for the star/galaxy classification.

Are all these features truly needed? And, if this is not the case and a smaller subset contains all the required information, what are the most useful features? We tried to answer these questions by evaluating the classification performance of each set of features through a priori knowledge of the true classification of each object, as it is listed in a much deeper and higher quality reference catalogue.

Most of the defined features are not independent. The presence of redundant features decreases the classification performance,

Table 8. Extracted features.

Number	Features	Symbols	Number	Features	Symbols
1	Isophotal Area	A	14	Gradient 1–2	G_{12}
2	Photometric Barycenter Abscissa	\bar{x}	15	Gradient 2–3	G_{23}
3	Photometric Barycenter Ordinate	\bar{y}	16	Gradient 3–4	G_{34}
4	Semimajor Axis	a	17	Average Surface Brightness	$\langle \text{SuBr} \rangle$
5	Semiminor Axis	b	18	Central Intensity	I_0
6	Position Angle	α	19	Ratio 1	T_r
7	Object Diameter	dia	20	Ratio 2	T_{cA}
8	Ellipticity of the Object Boundary	ell	21	Second Total Moment	C_2
9	Filling Factor	f_{fac}	22	Fourth Total Moment	C_4
10	Area Logarithm	c_2	23	Ellipticity (Averaged over the whole Area)	E
11	Harmonic Radius	r_{-1}			
12	Gradient 1–4	G_{14}	24	Peak Intensity	I_p
13	Gradient 1–3	G_{13}	25	Kron Radius	r_{Kron}

because any algorithm would try to minimize the error with respect to features that are not particularly relevant for the task. Furthermore, by introducing useless features the computational speed would be lowered.

The feature selection phase was realized through the sequential backward elimination strategy (Bishop 1995), which works as follows: let us suppose we have M features in one set and run the classification phase with this set. Then, we build M different sets with $M - 1$ features in each one and run the classification phase for each set, keeping the set that attains the best classification. This procedure allows us to eliminate the less significant feature. Then, we repeat the procedure $M - 1$ times, eliminating one feature at each step. In order to reduce the computation time further we do not use the validation set and the classification error is evaluated directly on the test set. It has to be stressed that this procedure is common in the statistical pattern recognition literature where, very often, for this task simplified models are also introduced. This however could be avoided in our case, because of the speed and good performance of our NNs.

Unsupervised NNs were not successful in this task, because the input data feature space is not separated into two non-overlapping classes (or, in simpler terms, the images and therefore the parameters of stars and galaxies fainter than the completeness limit of the image are quite similar), and they reach a performance level much lower than supervised NNs.

Supervised learning NNs give far better results. We used an MLP with one hidden layer of 19 neurons and only one output, assuming value 0 for star and value 1 for galaxy. After the training, we calculate the NN output as 1 if it is greater than 0.5 and 0 otherwise for each pattern of the test set. The experiments produce a series of catalogues, one for each set of features.

Fig. 13 shows the classification performance as a function of the adopted features. After the first step, the classification error remains almost constant up to $n = 4$, i.e. up to the point where features that are important for the classification are removed.

A high performance can be reached using just six features. With a lower number of features the classification worsens, whereas a larger number of features is unjustified, because it does not increase the performance of the system. The best performing set of features consists of features 11, 12, 14, 19, 21 and 25 of Table 8. They are two radii, two gradients, the second total moment and a ratio that involves measures of intensity and area.

3.8 Star/galaxy classification

Let us discuss now how the star/galaxy classification takes place. The first step is accomplished by ‘teaching’ the MLP NN using the selected best features. In this case we divided the data set into three independent data sets: training, validation and test sets. The learning optimization is performed using the training set, while the early stopping technique (Bishop 1995) is used on the validation set to stop the learning to avoid overfitting. Finally, we run the MLP NN on the test set.

As comparison classifier, we adopt SEX, which is based on an MLP NN. As features useful for the classification, SEX uses eight isophotal areas and the peak intensity plus a parameter, the FWHM of stars. As the SEX NN training was already realized by Bertin & Arnouts (1996) on 10^6 simulated images of stars and galaxies, we limit ourselves to tuning SEX in order to obtain the best performances on the validation set. Both SEX and our system use NNs for the classification, but they follow two different,

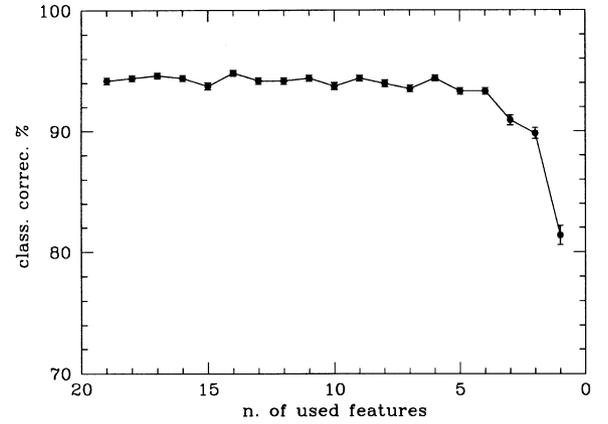


Figure 13. Classification performance as a function of the eliminated features.

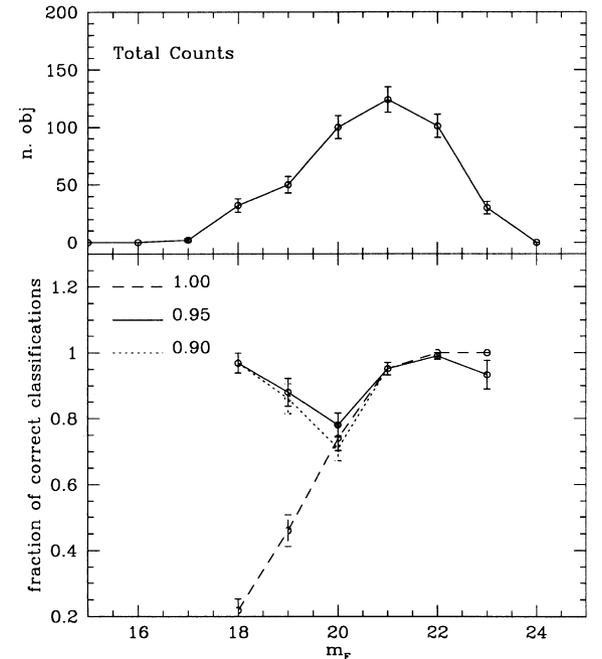


Figure 14. Optimization of the classification performance of SEX for different choices of the stellarity index parameter.

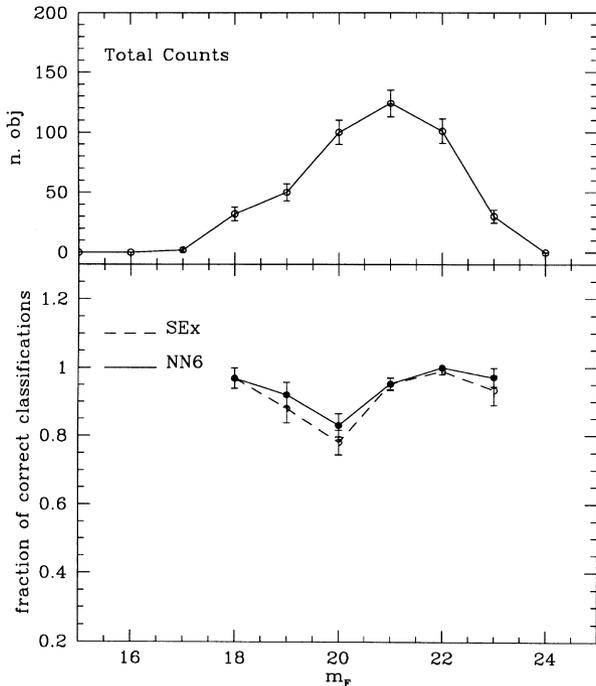
alternative approaches: SEX uses a very large training set of simulated stars and galaxies; our system uses noisy, real data. Furthermore, while the features of SEX are fixed by the authors, and the NNs output is a number x , $0 < x < 1$, our system selects the best performing ones and its output is an integer: 0 or 1 (i.e. star or galaxy). Therefore, we use the validation set for choosing the threshold that maximizes the number of correct classifications by SEX (see Fig. 14).

The experimental results are shown in Fig. 15 where the errors are plotted as a function of the magnitude. At all magnitudes NEXT misclassifies fewer objects than SEX. Out of 460 objects, SEX makes 41 incorrect classifications, NEXT just 28.

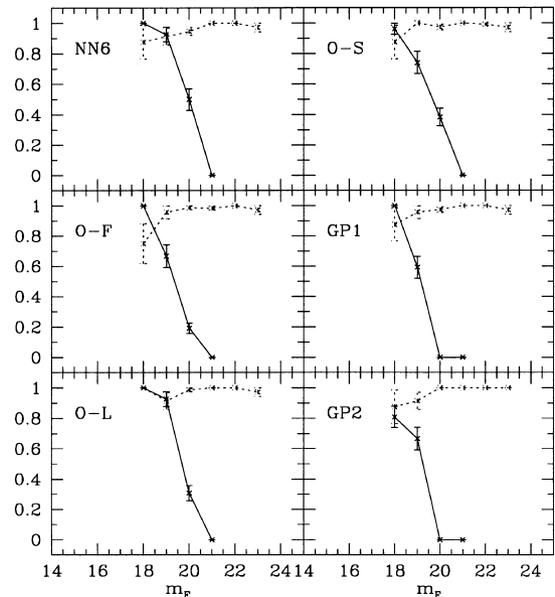
In order to check that our feature selection is optimal, we also compared our classification with those obtained using our MLP NNs with other feature sets, selected as shown in Table 9. The total number of misclassified objects in the test set of 460 elements was as follows: O-F, 43 errors; O-L, 30 errors; O-S, 35

Table 9. Adopted sets of features.

Source	Code	Number of the corresponding feature in Table 8
Odewhan full set	O-F	7, 8, 17, 18, 9, 10, 14, 15, 16
Odewhan large galaxies set	O-L	17, 14, 15, 16
Odewhan small galaxies set	O-S	17, 18, 16, 7
Godwin & Peach like 1977 first	GP1	17, flux
Godwin & Peach like 1977 second	GP2	10, flux

**Figure 15.** Classification performance of SEX and our NN-based methods.

errors; GP1, 48 errors; GP2, 49 errors. Fig. 16 shows the classification performance of the considered feature sets as a function of the magnitude of the objects. Results for stars are presented as solid lines, while for galaxies we used dotted lines. The performance of NEXt is presented in the top left panel: galaxies are correctly classified as long as they are detected, whereas the correctness of the classification of stars drops to 0 at $m_F = 21$. Fainter stars are almost absent in the IP92 catalogue, thus explaining why the classification performances for stars stop at brighter magnitudes than galaxies. O92 selected a 9-feature set (O-F) for the star/galaxy classification. Their set (central left panel) shows slightly poorer performance for bright ($m_F = 17$) galaxies and for faint stars ($m_F = 19$) than the set of features selected by us (upper left panel). They select also a smaller (four) set of features (O-F) quite useful for classifying large objects. The classification performance of this set, when applied to our images, turns out to be better than that of the larger feature data set: in fact, bright galaxies are not misclassified (see the bottom left panel). Even with respect to our data set O-F performs well: their set is slightly better in classifying bright galaxies, at the price of achieving poorer performance for faint stars. The further set of features by O92 (O-S) was aimed at the accurate detection of faint sources and performs similarly to their full set: it misclassifies bright galaxies and faint stars. The performance of the traditional classifiers, magnitude versus area (GP1) and magnitude versus

**Figure 16.** Classification performance of SEX and our NN based methods. The solid and dotted lines denote the performance for stars and galaxies, respectively.

brightness (GP2), is presented in the central and lower right panels. With just two features, all the faint objects are classified as galaxies, and owing to the absence of stars in our reference catalogue, the classification performances are 100 per cent. However, this is not a real classification. At bright magnitudes, the performance of the traditional classifiers is as good as, or slightly worse than, NEXt.

4 SUMMARY AND CONCLUSIONS

In this paper we discuss a novel approach to the problem of detection and classification of objects on WF images. In Section 2 we briefly review the theory of some types of NNs which are not familiar to the astronomical community. Based on these considerations, we implemented a neural network based procedure (NEXt) capable of performing the following tasks: (i) detecting objects against a noisy background; (ii) deblending partially overlapping objects; (iii) separating stars from galaxies. This is achieved by a combination of three different NNs, each performing a specific task. First we run a non-linear PCA NN to reduce the dimensionality of the input space via a mapping from pixels intensities to a subspace individuated through principal component analysis. For the second step we implemented a hierarchical unsupervised NN to segment the image and, finally, after a deblending and reconstruction loop we implemented a supervised MLP to separate stars from galaxies.

In order to identify the best performing NNs we implemented

and tested in homogeneous conditions several different models. NEXt offers several methodological and practical advantages with respect to other packages.

(i) It requires only the simplest a priori definition of what an ‘object’ is.

(ii) It uses unsupervised algorithms for all those tasks where both theory and extensive testing show that there is no loss in accuracy with respect to supervised methods. Supervised methods are in fact used only to perform star/galaxy separation because, at magnitudes fainter than the completeness limit, stars are usually almost indistinguishable from galaxies and the parameters characterizing the two classes do not lie in disconnected subspaces.

(iii) Instead of using an arbitrarily defined and often specifically tailored set of features for the classification task, NEXt, after measuring a large set of geometric and photometric parameters, uses a sequential backward elimination strategy (Bishop 1995) to select only the most significant ones. The optimal selection of the features was checked against the performances of other classifiers (see Section 3.8).

In order to evaluate the performance of NEXt, we tested it against the best performing package known to the authors (i.e. SEX) using a DPOSS field centred on the North Galactic Pole. We want also to stress here that – in order to have an objective test, and in contrast to what is currently done in the literature – NEXt was checked not against the performance of an arbitrarily chosen observer but rather against a much deeper catalogue of objects obtained from better quality material.

The comparison of NEXt performance against that of SEX shows that in the detection phase NEXt is at least as effective as SEX in detecting ‘true’ objects but much freer of spurious detections. As far as classification is concerned, the NEXt NN performs better than the SEX NN: 28 errors for NEXt against 41 for SEX for a total of 460 objects, most of the errors referring to objects fainter than the plate detection limit.

Other attempts, besides those described in the previous sections, to use NNs for similar tasks have been discussed in the literature. Bazell & Peng (1998) used the same North Galactic Pole field (but extracted from POSS-I plates) as used in this work. They tested their star/galaxy classification NN on objects which are both too few (60 galaxies and 27 stars) and too bright (a random check of their objects shows that most of the galaxies extend well over than 20 pixels) to be of real interest. It also needs to be stressed that, owing to their preprocessing strategy, their NNs are forced to perform cluster analysis on a huge multidimensional input space with scarcely populated samples.

Naim (1997) instead follows a strategy that is similar to ours and makes use of a fairly large data set extracted from POSS-I material. He, however, trained the networks to achieve the same performance as an experienced human observer while, as already mentioned, NEXt is checked against a catalogue of ‘True’ objects. Even though his target is the classification of objects fainter and larger than those we are dealing with, he tested the algorithm in a much more crowded and difficult region of the sky near the Galactic plane.

O92 make use of a traditional MLP and succeeded in demonstrating that AI methods can reproduce the star/galaxy classification obtained using traditional diagnostic diagrams by trained astronomers. Their aim, however, was less ambitious than that of ‘performing the correct star/galaxy classification’, which is the final goal of NEXt.

This paper is a first step toward the application of AI methods to

astronomy. Foreseen improvements of our approach are the use of Independent Component Analysis (ICA) NNs instead of PCA NNs and the adoption of Bayesian learning techniques to improve the classification performance of MLPs. These developments and the application of NEXt to other wide-field astronomical data sets obtained at large-format CCD detectors will be discussed in forthcoming papers.

ACKNOWLEDGMENTS

The authors wish to thank Chris Pritchett for providing them with a digital version of the IP92 catalogue. We also acknowledge the Canadian Astronomy Data Center for providing us with POSS-II material. This work was partly sponsored by the special grant MURST COFIN 1998, n.9802914427.

Fig. 1 was reproduced with permission from Ferguson (1998), STSC Vol. 11, The Hubble Deep Field, published by Cambridge University Press.

REFERENCES

- Arnaboldi M., Capaccioli M., Mancini D., Rafanelli P., Scaramella R., Sedmak G., Vettolani G. P., 1999, in 14th IAP Meeting, Wide Field Surveys in Cosmology. Editions Frontieres, Gif-sur-Yvette, p. 343
- Baldi P., Hornik K., 1989, *Neural Networks*, 2, 53
- Bazell D., Peng Y., 1998, *ApJS*, 116, 47
- Bertin E., Arnouts S., 1996, *A&AS*, 117, 393
- Bishop C. M., 1995, *Neural Networks for Pattern Recognition*. Oxford Univ. Press, Oxford
- Couch W. J., 1996, <http://ecf.hq.eso.org/hdf/catalogs>,
- Everitt B., 1977, *Cluster Analysis*. Heinemann, London
- Ferguson H. C., 1998, in Livio M., Fall S. M., Madau P., eds, *The Hubble Deep Field*, Proc. STScI Symp. Ser., Vol. 11. Cambridge Univ. Press, New York, p. 181
- Fritzke B., 1994, *Neural Networks*, 7, 1441
- Godwin J. G., Peach J. V., 1977, *MNRAS*, 181, 323
- Infante L., Pritchett C., 1992, *ApJS*, 83, 237 (IP92)
- Infante L., Pritchett C., Hertling G., 1995, *J. Astron. Data*, 1, 2
- Jarvis J. F., Tyson J. A., 1981, *AJ*, 86, 476
- Jutten C., Hérault J., 1991, *Signal Proc.*, 24, 1
- Karhunen J., Joutsensalo J., 1994, *Neural Networks*, 7, 113 (KJ94)
- Karhunen J., Joutsensalo J., 1995, *Neural Networks*, 8, 549 (KJ95)
- Koh J., Suk M., Bhandarkar S., 1995, *Neural Networks*, 8, 67
- Kohonen T., 1982, *Biological Cybernetics*, 43, 59
- Kohonen T., 1988, *Self-organization and associative memory*, 2nd edn. Springer-Verlag, Berlin
- Lanzetta K. M., Yahil A., Fernandez-Soto A., 1996, *Nat*, 381, 759
- Lipovetsky V. A., 1994, in MacGillivray H. T., Thomson E. B., Lasker B. M., Reid I. N., Malin D. F., West R. M., Lorenz H., eds, *IAU Symp. 161, Astronomy from Wide-Field Imaging*. Kluwer, Dordrecht, p. 3
- Lloyd S., 1982, *IEEE Trans. Inf. Theory*, IT-28, 2,
- Martinetz T., Berkovich S., Schulten K., 1993, *IEEE Trans. Neural Networks*, 4, 558
- Miller A. S., Coe M. J., 1996, *MNRAS*, 279, 293
- Naim A., 1997, *MNRAS*, in press (astro-ph/9701227)
- Odehahn S., Stockwell E., Pennington R., Humphreys R., Zumach W., 1992, *AJ*, 103, 318 (O92)
- Oja E., 1982, *Journal Math. Biology*, 15, 267
- Oja E., Ogawa H., Wangviwattana J., 1991, in Kohonen T. et al., eds, *Artificial neural networks*, IAU Symp., Vol. 161. North-Holland, Amsterdam, p. 385
- Oja E., Karhunen J., Wang L., Vigario R., 1996, in Marinaro M., Tagliaferri R., eds, *VII Italian Workshop on Neural Networks*, IAU Symp., Vol. 161. World Scientific, Singapore, p. 16
- Pal N. R., Pal S. K., 1993, *Pattern Recognition*, 26, 1277
- Plumbley M., 1996, *Appl. Intelligence*, 6, 185

Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., 1993, Numerical Recipes, Cambridge Univ. Press, Cambridge
Rose K., Gurewitz F., Fox G., 1990, Phys. Rev. Lett., 65, 945
Sanger T. D., 1989, Neural Networks, 2, 459
Tagliaferri R., Longo G., Andreon S., Zaggia S., Capuano N., Gargiulo G., 1998, in Marinaro M., Tagliaferri R., eds, X Italian Workshop on Neural Nets WIRN, IAU Symp., Vol. 161. Springer-Verlag, London, p. 169

Tagliaferri R., Capuano N., Gargiulo G., 1999a, IEEE Trans. Neural Networks, 10, 199
Tagliaferri R., Ciaramella A., Milano L., Barone F., Longo G., 1999b, A&AS, 137, 391

This paper has been typeset from a \TeX/L\AA\TeX file prepared by the author.