Expert Systems with Applications 40 (2013) 7258–7266

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

AutoAssociative Pyramidal Neural Network for one class pattern classification with implicit feature extraction

Bruno J.T. Fernandes [a,b], George D.C. Cavalcanti [a,*], Tsang I. Ren [a,c]

[a] Centro de Informática, Universidade Federal de Pernambuco, Av. Jornalista Anibal Fernandes, Cidade Universitária 50740-560, Recife-PE, Brazil
[b] Escola Politénica, Universidade de Pernambuco, Rua Benfica, 455, Madalena 50720-001, Recife-PE, Brazil
[c] iMinds – Vision Lab, Department of Physics, University of Antwerp, Universiteitsplein 1, Wilrijk B-2610, Belgium

ARTICLE INFO

Keywords:
Neural networks
Receptive fields
Autoassociative memory
One-class classification
Computer vision

ABSTRACT

Receptive fields and autoassociative memory are brain concepts that have individually inspired many artificial models, but models using both ideas have not been deeply studied. In this paper, we propose the AutoAssociative Pyramidal Neural Network (AAPNet), which is an artificial neural network for one-class classification that uses autoassociative memory and receptive field concepts in its pyramidal architecture. The proposed neural network performs implicit feature extraction and learns how to reconstruct a pattern from such features. The AAPNet is evaluated using the object categorization Caltech-101 database and presents better results when compared with other state-of-the-art methods.

1. Introduction

Motivated by the design and functioning of the human brain, biological models have been analyzed to develop powerful learning systems. Receptive fields (Hubel & Wiesel, 1965) and autoassociative memory (Rolls & Treves, 1998) are concepts that have been successfully used for the proposal of novel neural networks and pattern recognition models. Levine and Shefner (1991) defined a receptive field as "an area in which stimulation leads to response of a particular sensory neuron". In other words, this area represents a region of neurons covered by another neuron hierarchically located above them and tuned to extract features, like edges in specific orientations. Artificial neural networks have used the concept of receptive field to implicitly extract features from patterns (Fukushima, 1988; LeCun, Bottou, Bengio, & Haffner, 1998; Perez, Salinas, Estvez, & Valenzuela, 2003). Phung and Bouzerdoum (2007) and Fernandes, Cavalcanti, and Ren (2009) proposed neural networks in which the layers are organized in a pyramidal architecture (Gonzalez & Woods, 2010) that iteratively extract features through the receptive fields associated to each layer. These models preserve the spatial topology of the data, therefore, the features are linked to specific locations of the image.

The human brain uses a type of memory called autoassociative memory for episodic memory storage and for short term memory (Rolls & Treves, 1998). In this kind of memory, the learning process of new patterns is performed very fast, and besides, the entire pattern can be retrieved based only on a part of it. In pattern recognition, an autoassociative classifier is a particular case of one-class classifiers (OCC) (Moya, Koch, & Hostetler, 1993). OCC has already been applied in concept learning, in which the autoassociative memory is used to learn the inner structure of a class based only on the features collected from the patterns of this class. The autoassociative classifiers must learn the mapping of a pattern to a new feature space and then learn the inverse mapping with respect to the minimization of the distance between the input pattern and the output given by the classifier.

In this paper, we propose the AutoAssociative Pyramidal Neural Network (AAPNet), which is an autoassociative neural network with pyramidal architecture that receives as input an image, implicit extract features and returns the reconstruction of the input image based on these features. The AAPNet uses the concepts of receptive field and autoassociative memory in its pyramidal architecture. The combination of such concepts leads to a neural network model for computer vision that incorporates feature extraction and classification with closed decision boundaries in the same structure. This neural network model automatically learn suitable features to represent a given pattern in a non-local way. Thus, it is able to represent complex functions with the advantage of not requiring negative samples in the learning process. Previously proposed autoassociative neural networks (Hanif, Prevost, Belaroussi, & Milgram, 2008; Hinton & Salakhutdinov, 2006; Thompson et al., 2002; Cavalcanti, Pereira, & Filho, 2004) do not take advantage of the benefits of the combination of receptive fields concepts, autoassociative memory and pyramidal architecture in the same model.

* Corresponding author. Address: Universidade Federal de Pernambuco, Centro de Informática, Av. Jornalista Anibal Fernandes, Cidade Universitária 50740-560 Recife, PE, Brazil. Tel.: +55 81 2126 8430x4346; fax: +55 81 2126 8438.
E-mail address: gdcc@cin.ufpe.br (G.D.C. Cavalcanti).
URL: http://www.cin.ufpe.br/~viisar (G.D.C. Cavalcanti).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.06.080

The paper is organized as follows. In Section 2, related works are presented. In Section 3, the architecture of the proposed AAPNet is described. The AAPNet is evaluated using the Caltech-101 object categorization database and the results are shown in Section 4. Finally, in Section 5, some concluding remarks are given.

2. Related works

The neocognitron (Fukushima, 1988) is a neural network model based on the concept of receptive fields. It was motivated by the human brain behavior and the hypothesis of Hubel and Wiesel (1965) in an attempt to propose a neural network model of the visual system. The neocognitron performs the learning and recognition process inside its architecture with implicit feature extraction. In the neocognitron, the classification process begins with extraction of local features. The local features are gradually integrated into global features. The neocognitron was improved by Fukushima (2003), who proposed several modifications, such as inhibitory surround in the connections and supervised competitive learning. Many other models were developed based on the neocognitron, such as Convolutional Neural Network (CNN) (LeCun et al., 1998), Feature Extraction Network with Multilayer Perceptron (FEN + MLP) (Perez et al., 2003), Pyramidal Neural Network (PyraNet) (Phung & Bouzerdoum, 2007), and Inhibitory Pyramidal Neural Network (I-PyraNet) (Fernandes, Cavalcanti, & Ren, 2008; Fernandes et al., 2009).

Table 1
Notation and definitions used to describe the AAPNet.

Symbol	Description
l_n	2-D pyramidal layer in the position n
L	Last 2-D pyramidal layer
R	Reconstruction layer
I	Input image used as the first layer l_0 of the AAPNet
$I^k_{u,v}$	Value of the pixel in the (u, v) position of the k-th input image
r_n	Size of the receptive field of the neurons in the layer l_n
o_n	Size of the overlap between the receptive fields of the neurons in the layer l_n
$w^n_{i,j}$	Weight associated with the position (i,j) in the layer l_{n-1} to the layer l_n
γ_n	Area in the input image covered by a neuron in the layer l_n
κ_n	Size of the overlap between covered areas of the neurons in the layer l_n
$b^l_{u,v}$	Bias of the neuron at (u, v) in the layer $l \in \{l_1, \ldots, L\}$
$y^l_{u,v}$	Output of the neuron at (u, v) in the layer $l \in \{l_1, \ldots, L, R\}$
$w^R_{i,j}$	Weight associated with the position (i,j) in the layer L to the layer R
f	Activation function
$\delta^{l,k}_{u,v}$	Error sensitivity for the neuron at the position (u, v) in the layer $l \in \{l_1, \ldots, L, R\}$
$s^{l,k}_{u,v}$	Input weighted sum for the neuron at the position (u, v) in the layer $l \in \{l_1, \ldots, L, R\}$
$\Delta^{(t)}_{i,j}$	Adaption rule of the RPROP algorithm
$\eta^{[+,-]}$	Increase (+) and decrease (−) factors of the RPROP algorithm

CNN is a neural network composed of two kinds of cells: simple cells, which perform feature extraction, and complex cells, which perform subsample of the features in order to focus on the relationship between them. LeCun et al. (1998) took into consideration the fact that a specific neural network architecture based on *a priori* knowledge is able to improve the generalization ability of the model. So, CNN was proposed to implicitly extract the features of the patterns making it a better model to handle the great variability and richness of natural data.

Perez et al. (2003) emphasized the ability of simple cells in the detection of stimulus in specific orientations, while complex cells are activated every time their receptive fields composed of simple cells present a strong stimulus in their detection task. They proposed the FEN + MLP neural network simulating the behavior of such cells for feature extraction. A neural classifier on the top of the neural network decides the class of each pattern. Perez et al. (2003) used a genetic algorithm to find the optimal architecture of the FEN + MLP.

Phung and Bouzerdoum (2007) proposed the PyraNet, a neural network specifically developed for image recognition tasks. The PyraNet is based on CNN and on the pyramid images concepts (Gonzalez & Woods, 2010). The PyraNet learning algorithm tunes the nonlinear processing of each pyramid level to solve specific recognition problems. The PyraNet integrates the feature extraction and pattern classification steps in the same structure. Moreover, the PyraNet also maintains the spatial topology of the input image and presents a simple connection scheme with lower computational and memory costs than in other neural networks as demonstrated in Phung and Bouzerdoum (2007).

The I-PyraNet (Fernandes et al., 2008; Fernandes et al., 2009) extends the PyraNet by incorporating the inhibitory fields concept in the neural network. The results achieved by Fernandes et al. (2009) demonstrated that the I-PyraNet is a more stable model than the PyraNet and obtains higher classification rates in detection tasks. However, both models are not good approaches to multi-class classification tasks since the number of free parameters is not large enough to define many different decision boundaries between the classes.

Hopfield networks (Hopfield, 1982) have been applied in the development of autoassociative models. Their applications range from image restoration and reconstruction (Sun, 2000) to image storage and retrieval (Ramya, Kavitha, & Shreedhara, 2011). The autoassociative classifiers map the input pattern to a new feature space and then learn the inverse mapping that minimizes the

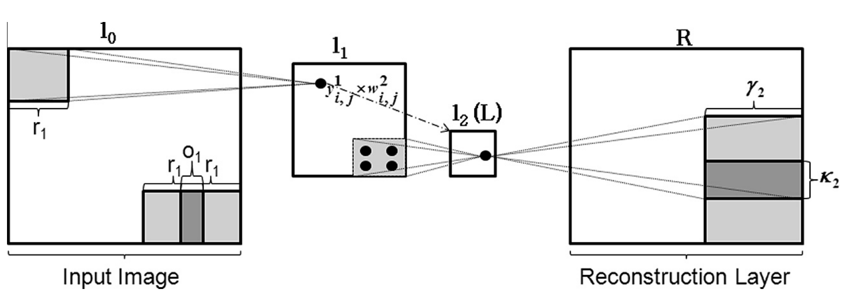


Fig. 1. AAPNet architecture in a 2-D bottleneck shape composed of pyramidal layers (layers l_0 to L) responsible for the features extraction of the input pattern and an output layer located at the top of the neural network responsible for the image reconstruction (layer R). Each neuron in a pyramidal layer l_n is connected to a receptive field in the previous layer with size $r_n \times r_n$ and an overlap region with size of o_n. The output of a neuron in a pyramidal layer l_n, given by $y^n_{i,j}$, is then multiplied by the weight associated to the connection from the neuron to the next layer, $w^{n+1}_{i,j}$, and it is used as an excitatory stimulus by the neurons in the layer l_{n+1}. The AAPNet is trained to approximate the output of the layer R to the neural network input.

distance between the input and the output patterns. Principal Component Analysis (PCA) is a method that can be applied in such a way. However, PCA is able only to identify linear correlations in a data set. Neural networks (Haykin, 2007), like the MLP, are also used as autoassociative classifiers (Cavalcanti et al., 2004) with the possibility of identify nonlinear correlations between the variables. Thompson et al. (2002) applied autoassociative neural networks for novelty detection showing that learning happens in a more substantial way than a simple memorization. On the other hand, Hinton and Salakhutdinov (2006) presented a model to train autoassociative neural networks with many hidden layers, leading to a more stable model. Moreover, Hanif et al. (2008) used an autoassociative neural network to localize specific facial features.

3. Autoassociative pyramidal neural network

The Autoassociative Pyramidal Neural Network (AAPNet) is a biologically inspired model designed to implicitly learn suitable features to represent visual patterns. The AAPNet is a one-class neural network that represents a specific visual pattern and outputs how close an image is to the class represented by the AAPNet. In this section, we present the neural network architecture (Section 3.1), the connectivity model (Section 3.2) and a description of the training algorithm (Section 3.3). Table 1 presents the notation and definitions used to describe the AAPNet.

3.1. AAPNet architecture

Fig. 1 presents the AAPNet architecture. The entire network is connected in cascade, i.e., the output of one layer works as the input to the next one. The features are iteratively extracted by the neurons connected to the receptive fields and they are used to reconstruct the input pattern presented to the AAPNet. The AAPNet presents a 2-D bottleneck shape. Layers arranged in a pyramidal architecture (layers l_0 to L) are responsible for the feature extraction of the input pattern, and an output layer with the same size of the input image is responsible for the image reconstruction (layer R). Each neuron in the last pyramidal layer, L, is connected to a receptive field in the output layer and the receptive field size is given by the area of the image covered by such neuron defined in the next section.

The first layer of the AAPNet, l_0, is represented by the input image, which is iteratively subsampled through the layers l_1 to L. Then, the image is reconstructed based on the features of the last feature extraction layer L.

The reconstruction process preserves the spatial topology of the extracted features since the position of each neuron in the same layer denotes the position of the respective receptive field in the previous layer. The output of the neurons in R is the output of the AAPNet.

The reconstruction layer R has the same size of the input layer l_0 and the output of its neurons represents the output of the AAPNet. The distance between the input image and the AAPNet output is used to decide whether or not the input image belongs to the visual pattern represented by the AAPNet.

3.2. Connectivity model

The first layer of the AAPNet, l_0, is the input image. Each neuron in a pyramidal layer l_n is connected to a region of affectation in the previous layer l_{n-1} and $r_n \times r_n$ is the size of such region, called receptive field. Adjacent neurons share connections from an overlapped area in their receptive fields and the number of overlapped neurons in l_{n-1} is given by o_n. Each neuron in the layer l_1, for example, is connected to a receptive field composed of $r_1 \times r_1$ neurons in the input layer. An adjustable weight, $w^1_{i,j}$, is associated with each neuron in the (i,j) position at the layer l_0.

Each neuron in the last pyramidal layer L is connected to a specific area in the reconstruction layer. This area denotes the covered area of a neuron in the input image, given by γ, and it is recursively calculated from the union of the receptive fields in the previous layers that are connected to such neuron. Adjacent neurons also share some connections in the covered area, given by κ. These parameters are calculated by:

$$\gamma_n = \begin{cases} r_0, & \text{if } n = 0 \\ (r_n\gamma_{n-1}) - ((r_n - 1)\kappa_{n-1}), & \text{otherwise} \end{cases} \tag{1}$$

$$\kappa_n = \begin{cases} o_0, & \text{if } n = 0 \\ (o_n\gamma_{n-1}) - ((o_n - 1)\kappa_{n-1}), & \text{otherwise} \end{cases} \tag{2}$$

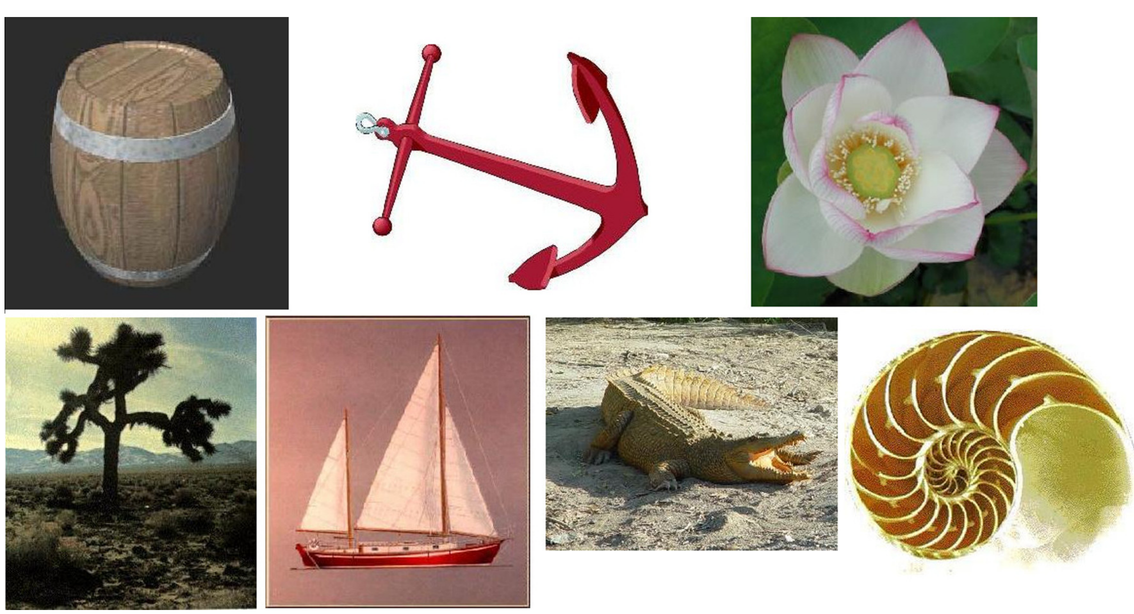


Fig. 2. Image samples from the Caltech-101 subset used to find the optimal parameters of the AAPNet.

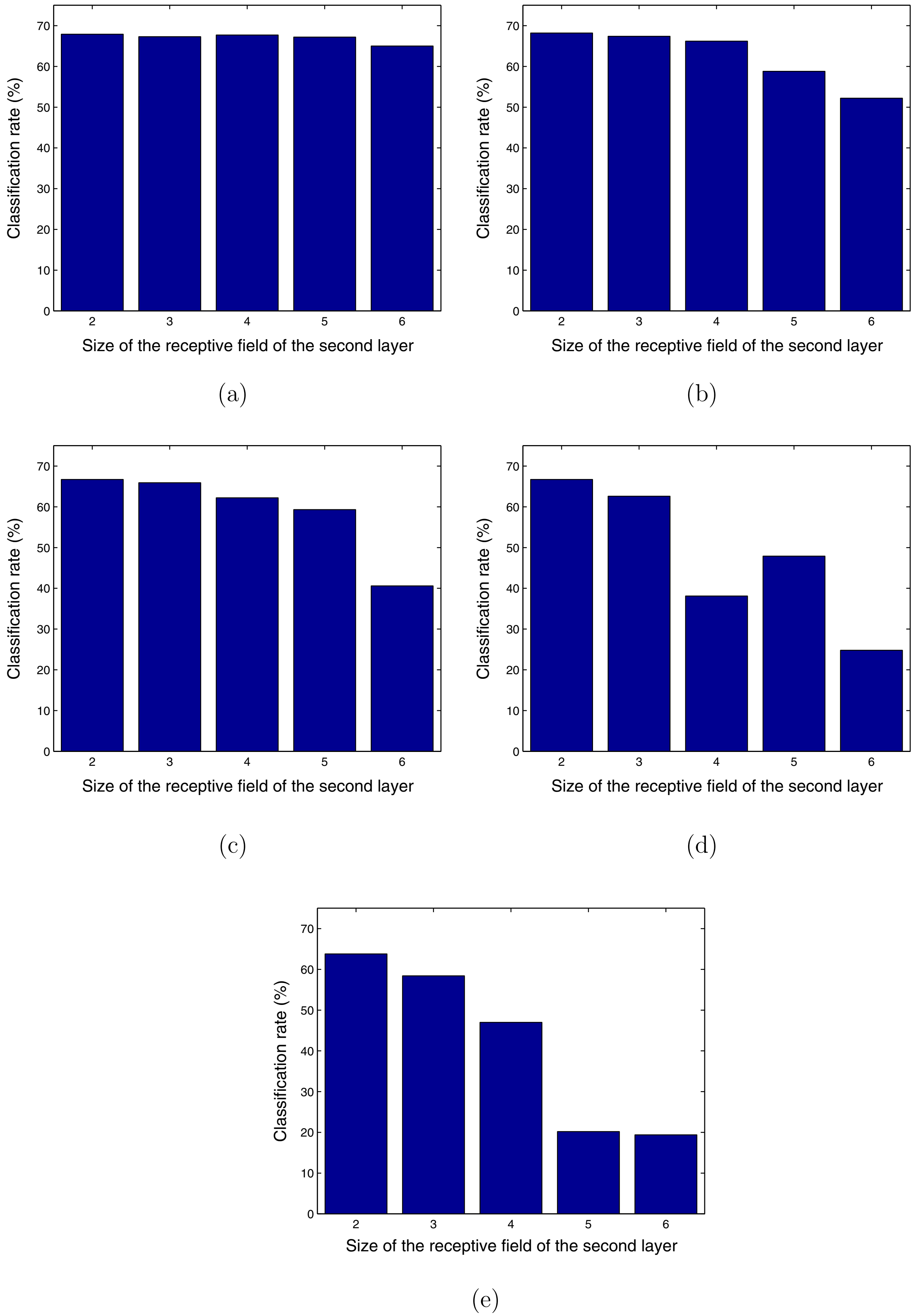


Fig. 3. Classification rates for different receptive fields configurations with a fixed overlap factor of 1 for both pyramidal layers. Receptive field sizes of the first layer are equal to: (a) 2, (b) 3, (c) 4, (d) 5, (e) 6.

where γ_n and κ_n are the image covered area and the overlap region size between adjacent covered areas of the layer l_n in the input image, respectively.

The output of a neuron is defined as a non-linear activation function over the weighted summation of the neurons inside its receptive field. Thus, being (u, v) the position of a neuron in layer l_n, (i,j) the position of the neuron in the previous layer l_{n-1} and $b^n_{u,v}$ the bias of the neuron at (u, v), the output $y^n_{u,v}$ of the neuron in a pyramidal layer is given by:

$$y^n_{u,v} = f\left(\sum_{i=i^{n,0}_{min}}^{i^{n,0}_{max}} \sum_{j=j^{n,0}_{min}}^{j^{n,0}_{max}} w^n_{i,j} y^{n-1}_{i,j} + b^n_{u,v}\right). \quad (3)$$

The output of a neuron in the reconstruction layer, y^R, depends on the output of the neurons in the last pyramidal layer that contains such neuron in their covered area and it is given by:

$$y^R_{u,v} = f\left(\sum_{i=i^1_{min}(\gamma,\kappa)}^{i^1_{max}(\gamma,\kappa)} \sum_{j=j^1_{min}(\gamma,\kappa)}^{j^1_{max}(\gamma,\kappa)} w^R_{i,j} y^L_{i,j}\right), \quad (4)$$

where $w^R_{i,j}$ denotes the weight associate with the input position (i,j) at the last pyramidal layer L to the reconstruction layer R, $y^L_{i,j}$ is the output of the neuron (i,j) and i_{min},i_{max},j_{min} and j_{max} are defined as:

$$i^p_{min}(x,y) = \begin{cases} u(x-y), & p = 0 \\ \left\lceil \frac{u-x}{x-y} \right\rceil + 1, & otherwise \end{cases}, \quad (5)$$

$$i^p_{max}(x,y) = \begin{cases} u(x-y) + x, & p = 0 \\ \left\lfloor \frac{u-1}{x-y} \right\rfloor + 1, & otherwise \end{cases}, \quad (6)$$

$$j^p_{min}(x,y) = \begin{cases} v(x-y), & p = 0 \\ \left\lceil \frac{v-r_n}{x-y} \right\rceil + 1, & otherwise \end{cases}, \quad (7)$$

$$j^p_{max}(x,y) = \begin{cases} v(x-y) + x, & p = 0 \\ \left\lfloor \frac{v-1}{x-y} \right\rfloor + 1, & otherwise \end{cases}. \quad (8)$$

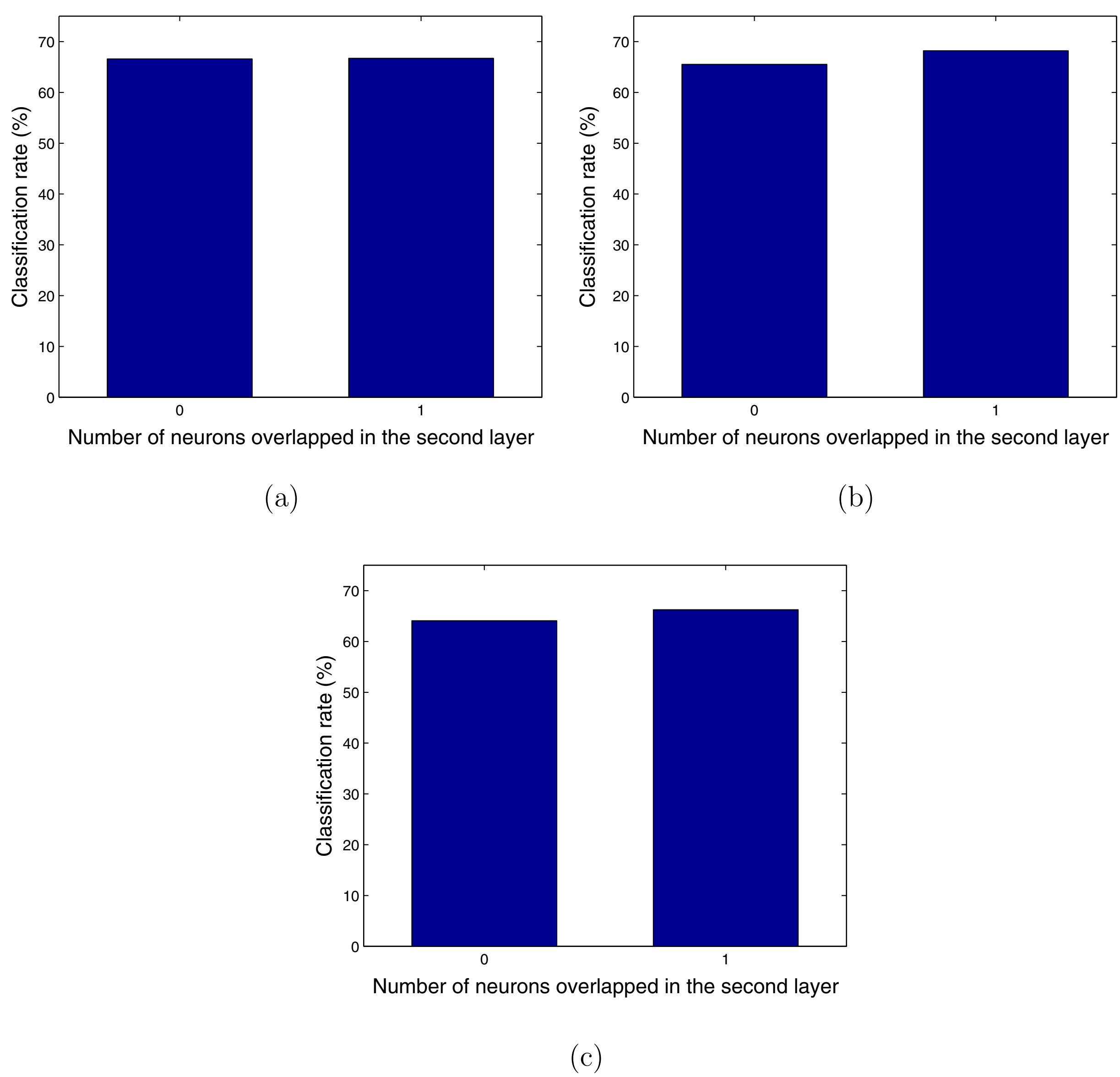


Fig. 4. Classification rates for different overlap configurations with receptive fields of size 3 in the first pyramidal layer and 2 in the second pyramidal layer. Number of overlapped neurons in the first layer equal to: (a) 0, (b) 1, (c) 2.

3.3. AAPNet training

The AAPNet is a supervised neural network and its objective is to reduce the difference between the input image and the obtained output. This is performed by adjusting the weights in the AAPNet.

The error sensitivity δ for each neuron in the reconstruction layer for an input image I is given by:

$$\delta^{R,k}_{u,v} = y^{R}_{u,v} - I^{k}_{u,v} \times f'\left(s^{R,k}_{u,v}\right), \tag{9}$$

where $s^{R,k}_{u,v}$ is applied as the weighted sum input for the neuron (u,v) at the reconstruction layer, f' is the differential of the activation function f and k is the index representing each training image. Moreover, the error sensitivity for the neurons in the last pyramidal layer is given by:

$$\delta^{L,k}_{u,v} = f'\left(s^{n,k}_{u,v}\right) \times w^{R}_{u,v} \times \sum_{i=i^{0}_{min}(\gamma,\kappa)}^{i^{0}_{max}(\gamma,\kappa)} \sum_{j=j^{0}_{min}(\gamma,\kappa)}^{j^{0}_{max}(\gamma,\kappa)} \delta^{R,k}_{i,j}. \tag{10}$$

The error sensitivity for the neurons in the other pyramidal layers is given by:

$$\delta^{l,k}_{u,v} = f'\left(s^{n,k}_{u,v}\right) \times w^{n+1}_{u,v} \times \sum_{i=i^{1}_{min}(r_{n+1},o_{n+1})}^{i^{1}_{max}(r_{n+1},o_{n+1})} \sum_{j=j^{1}_{min}(r_{n+1},o_{n+1})}^{j^{1}_{max}(r_{n+1},o_{n+1})} \delta^{n+1,k}_{i,j}. \tag{11}$$

Furthermore, the error gradient of the weights and the biases can be derived as shown in the next equations:

- Last pyramidal layer:

$$\frac{\partial E}{\partial w^{R}_{i,j}} = \sum_{k=1}^{K} \left\{ y^{L,k}_{i,j} \times \sum_{i=i^{0}_{min}(\gamma,\kappa)}^{i^{0}_{max}(\gamma,\kappa)} \sum_{j=j^{0}_{min}(\gamma,\kappa)}^{j^{0}_{max}(\gamma,\kappa)} \delta^{R,k}_{u,v} \right\}. \tag{12}$$

- Other pyramidal layers:

$$\frac{\partial E}{\partial w^{n}_{i,j}} = \sum_{k=1}^{K} \left\{ y^{n-1,k}_{i,j} \times \sum_{i=i^{1}_{min}(r_n,o_n)}^{i^{1}_{max}(r_n,o_n)} \sum_{j=j^{1}_{min}(r_n,o_n)}^{j^{1}_{max}(r_n,o_n)} \delta^{n,k}_{u,v} \right\}. \tag{13}$$

- Biases:

$$\frac{\partial E}{\partial b_n} = \sum_{k=1}^{K} \delta^{k}_{n}, \quad \frac{\partial E}{\partial b_{u,v}} = \sum_{k=1}^{K} \delta^{k}_{u,v}. \tag{14}$$

Finally, the weights in the neural network might be updated following a given learning rule. Resilient Propagation rule (Riedmiller & Braun, 1993) is used in this regard in the paper. Thus, the weights are adaptively updated based on the gradient signal, according to the following rule:

$$w^{(t)}_{i,j} = w^{(t)}_{i,j} \times \left(-sign\left(\frac{\partial E}{\partial w^{(t)}_{i,j}}(t)\right)\right) \times \Delta^{(t)}_{i,j} \tag{15}$$

and $\Delta^{(t)}_{i,j}$ is the adaptation rule given by:

$$\Delta^{(t)}_{i,j} = \begin{cases} \eta^{+} \times \Delta^{(t-1)}_{i,j}, & \frac{\partial E}{\partial w^{(t)}_{i,j}}(t) \times \frac{\partial E}{\partial w^{(t)}_{i,j}}(t-1) > 0 \\ \eta^{-} \times \Delta^{(t-1)}_{i,j}, & \frac{\partial E}{\partial w^{(t)}_{i,j}}(t) \times \frac{\partial E}{\partial w^{(t)}_{i,j}}(t-1) < 0 \\ 0, & otherwise \end{cases} \tag{16}$$

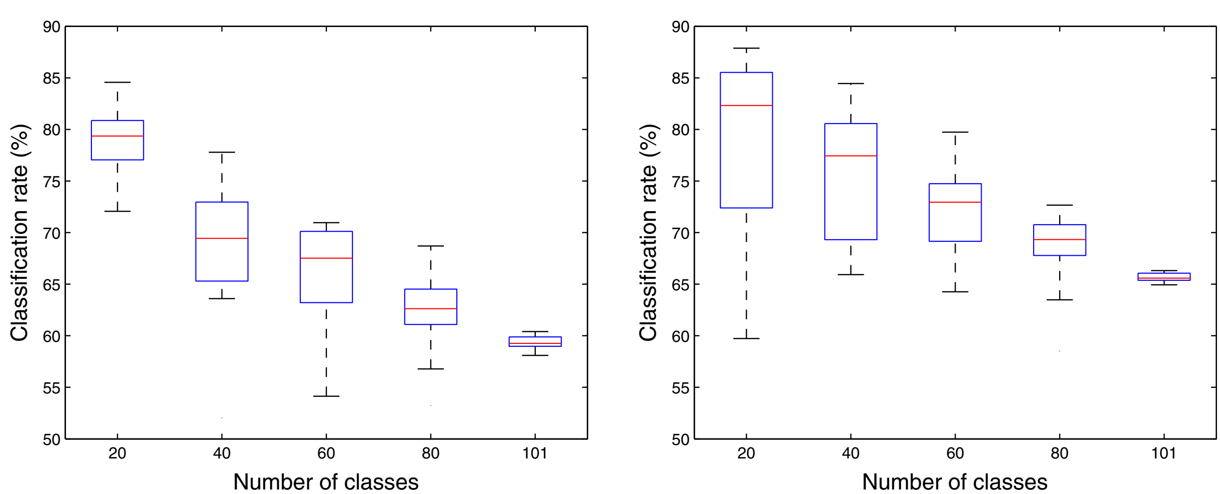
where $\eta^{+} > 1$ and $0 < \eta^{-} < 1$ are the increase and decrease factors, respectively, that define the jump given in each learning step.

4. Object categorization

Object categorization algorithms aims to assign a category to an image. This is a challenging task because objects may present variation in shape, texture, position, occlusion, noise, or background clutter (Galleguillos & Belongie, 2010). The AAPNet is evaluated using the Caltech-101 object categorization database (Fei-Fei, Fergus, & Perona, 2006) and its results are compared with other of state-of-the-art methods.

4.1. Methodological protocol

The used AAPNet has three pyramidal layers, including the input layer and one reconstruction layer. The activation function used in all the neurons of the AAPNet is the sigmoid-logistic. An ensemble averaging (Verikas, Lipnickas, Malmqvist, Bacauskiene, & Gelzinis, 1999) of AAPNets is trained for each class. In this committee, there is one neural network for each training image. The test pattern is classified based on the minimum distance between the neural network input and the reconstructed image. The closer is the input image to the AAPNet output, the higher is the probability of the image be associated to the category represented by the AAPNet.



(a) (b)

Fig. 5. Box plot for different number of classes with (a) 15 and (b) 30 training images.

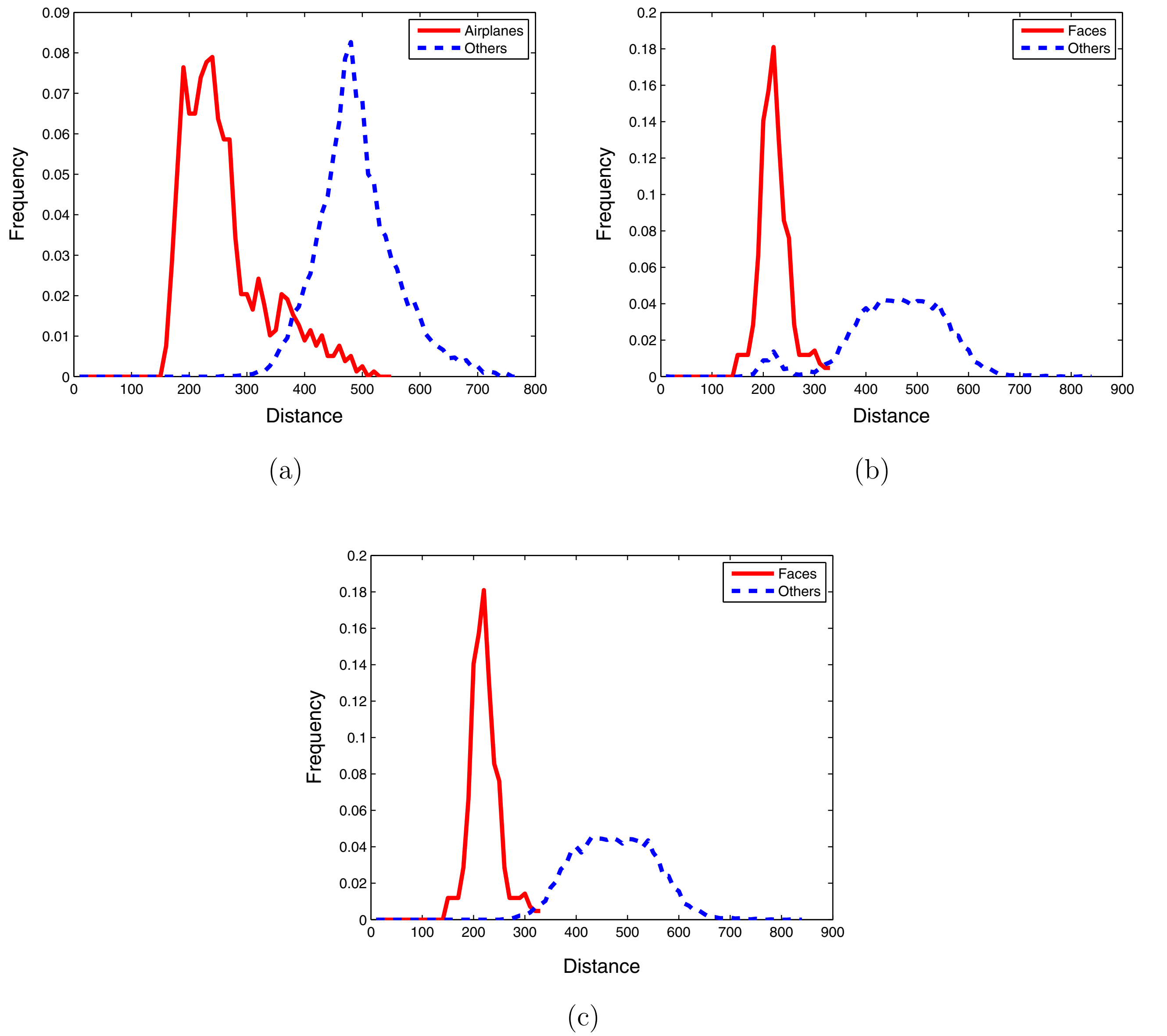


Fig. 6. Comparison between the distances: (a) from airplanes to objects from other classes; (b) from faces to objects of other classes; (c) from faces to objects of other classes without the “faces_easy” patterns.

The Caltech-101 object database has varied images from 101 objects. The images were converted to gray scale, had their histogram equalized and were downsampled to 40 × 40 pixels.

A subset of the Caltech-101 containing seven classes (anchor, barrel, crocodile, joshua tree, ketch, lotus and nautilus) was used to determine the best configuration of the AAPNet. Five training images were randomly chosen per class and the remaining images were used for testing. Fig. 2 presents one image example from the seven classes contained in the Caltech-101 subset. The configuration used in the test with the complete database is the one that presents the highest classification rate in the parameter determination step presented in Section 4.2.

In the experimental setup for the test with the complete database, 15 or 30 images are randomly chosen per class for training and the remaining images are used for testing. Such procedure is repeated ten times and the average results are presented. Wu, Zheng, You, and Du (2007) used the same experimental setup to present their results. Section 4.3 presents the results achieved by the AAPNet using the whole Caltech-101 database.

Table 2
Classification rate (%) for object categorization in the Caltech-101 database with the standard deviation in parenthesis.

Classifier	15 training images	30 training images
AAPNet	**59.38**(**0.64**)	**65.64**(**0.46**)
Serre et al. (2005)	35.00	42.00
Mutch and Lowe (2006)	51.00	56.00
Wolf et al. (2006)	51.18(1.20)	–
Wu et al. (2007)	52.16(1.00)	60.23(0.80)
Linear SVM (Maji et al., 2013)	38.79(0.94)	44.33(1.33)
Kernel SVM (Maji et al., 2013)	44.27(1.45)	50.13(1.19)
IKSVM (Maji et al., 2013)	50.10(0.65)	56.59(0.77)

The characters in bold present the best classification rate for each training configuration.

4.2. Parameters determination

Fig. 3 presents the classification rate obtained with different receptive field configurations in order to find the optimal parameters for the AAPNet in the object categorization task. The receptive fields sizes range from 2 to 6 neurons in both pyramidal layers.

A large receptive field size leads to a bad classification rate, as shown in the configuration with receptive field size 6 for both pyramidal layers. This is an expected behavior since the receptive field of a layer increases when the number of neurons in the layer is reduced, leading to fewer features extracted and creating larger homogeneous regions in the neural network output. For example, if there is only one neuron in the last feature extraction layer L of the AAPNet, the output of all the neurons in the reconstruction layer R is the same. On the other hand, if the size of the receptive field is too small, many features are going to be extracted and the model is more sensitive to the variance in the objects. The optimal configuration of the receptive fields sizes is 3 and 2 for the first and the second pyramidal layers, respectively. This configuration obtained a classification rate of 68.21% in the subset used to determine the optimal configuration.

Fig. 4 presents the classification rate obtained with different overlap configurations using the optimal receptive fields configuration shown in Fig. 3. The overlap sizes range from 0 to 2 in the first pyramidal layer and from 0 to 1 in the second pyramidal layer.

The overlap configuration of size 1 for both pyramidal layers presented the highest classification rate. The absence of an overlap between adjacent receptive fields in both layers led to the worst classification rate. It can be speculated that this absence reduces the neural network fault tolerance since no redundancy is shown between the neurons.

4.3. Experimental results

Fig. 5(a) and (b) show the performance of the AAPNet with the parameters determined in the previous section varying the number of classes (20, 40, 60, 80 and 101) with 15 and 30 training images per class, respectively. The experiments were performed with 20, 40, 60, 80, and 101 classes randomly selected and it is presented as a box plot defined based on 10 runs for each number of classes. The difference between the upper and lower quartiles increased with a reduced number of classes which indicates that some classes are more difficult to be recognized or that there are very similar classes. The results obtained with 101 classes presented the lowest skewness and median classification rate since all the classes are used in each run.

Fig. 6(a) and (b) present a histogram comparing the distances from the categories "airplanes" and "faces" to the objects of the other 100 classes, respectively. The frequency in the histogram results from the number of images with the distance to the class belonging to intervals of size 10. The category "faces" presents a big region of overlap with short distances with some patterns in the Caltech-101 database. However, it is important to note that the Caltech-101 has another face class, called "faces_easy". Fig. 6(c) presents the comparison between the distances from the "faces" to the objects of other classes without the "faces_easy" patterns. In this scenery, the distance overlap of the "faces" patterns from patterns of the other classes is much lower. The area under the ROC curve (Fawcett, 2006) for these categories are: 0.98 for "airplanes"; 0.97 for "faces"; and approximately 1.00 for "faces" without the "faces_easy" patterns. The results show that the AAPNet is able to detect objects and to define a large separation surface between true and false examples.

Table 2 shows the results obtained with the AAPNet in the complete Caltech-101 database compared with the results obtained by state-of-the-art approaches. Two experiments, with 15 or 30 images randomly chosen per class for a single training run, were performed. Wu et al. (2007) proposed a biologically inspired method for learning the visual features and presented the highest classification rate among the state-of-the-art classifiers. The AAPNet achieved the best classification rate in both experiments with 7.22 and 5.41 percentile points higher than the ones obtained by Wu et al. (2007) for 15 and 30 training images, respectively. The AAPNet and the Support Vector Machines (SVMs) (Maji, Berg, & Malik, 2013) do not use any feature extraction step, but the former presents a recognition rate of nearly 10 percentiles points higher than the latter. Moreover, the AAPNet is shown as the most stable model since its standard deviation is the lowest one.

5. Concluding remarks

This work presented a novel neural network with implicit feature extraction for visual pattern recognition, called AAPNet. The motivation for the development of such neural network relies on the good results presented by the models that use the concepts of receptive fields and autoassociative memory. The AAPNet is an autoassociative pyramidal neural network that is able to learn the inner structure of the patterns without the need of negative examples. The experiments herein performed demonstrated that the AAPNet is a valuable alternative in pattern recognition tasks. The AAPNet obtained the best classification results in the object categorization task. Another advantage of the AAPNet is its modularization, where the emergence of a new class leads to the construction of a new classifier without affecting the other already trained neural networks. Moreover, the AAPNet does not need negative information to find the boundaries of a given pattern and it can be appropriately applied in other pattern recognition tasks with the advantages of the one-class classifiers.

We are currently investigating approaches to automatically find the best parameters for the AAPNet architecture. In AAPNet, the size of the receptive fields is the same for all neurons located in a given layer. Another direction for future research is to develop an algorithm to calculate a different receptive field size per neuron.

Acknowledgment

This research was partially supported by CNPq, Capes and Facepe.

References

Cavalcanti, G. D. C., Pereira, C. S., & Filho, E. C. B. C. (2004). Auto-associative neural networks and eigenbands fusion for frontal face verification. In *Proceedings of the brazilian symposium on artificial neural networks* (No. 3656, pp. 1–6).

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*, 861–874.

Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE Transaction of Pattern Analysis and Machine Intelligence, 28*, 594–611.

Fernandes, B. J. T., Cavalcanti, G. D. C., & Ren, T. I. (2009). Nonclassical receptive field inhibition applied to image segmentation. *Neural Network World, 19*, 21–36.

Fernandes, B. J. T., Cavalcanti, G. D. C., & Ren, T. I. (2008). Classification and segmentation of visual patterns based on receptive and inhibitory fields. In *Proceedings of the IEEE international conference on hybrid intelligent systems* (pp. 126–131).

Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks, 1*, 119–130.

Fukushima, K. (2003). Neocognitron for handwritten digit recognition. *Neurocomputing*, 161–180.

Galleguillos, C., & Belongie, S. (2010). Context based object categorization: A critical survey. *Computer Vision and Image Understanding, 114*, 712–722.

Gonzalez, R. C., & Woods, R. E. (2010). *Processamento de imagens digitais*. Prentice-Hall.

Hanif, S. M., Prevost, L., Belaroussi, R., & Milgram, M. (2008). Real-time facial feature localization by combining space displacement neural networks. *Pattern Recognition Letters, 28*, 1094–1104.

Haykin, S. (2007). *Neural networks: A comprehensive foundation*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc..

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5768), 504–507.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the national academy of sciences* (Vol. 79, pp. 2554–2558).

Hubel, D. H., & Wiesel, T. N. (1965). Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology, 28*(2), 229–289.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE* (Vol. 86, pp. 2278–2324).

Levine, M., & Shefner, J. (1991). *Fundamentals of sensation and perception*. Oxford, NY, USA: Oxford University Press.

Maji, S., Berg, A. C., & Malik, J. (2013). Efficient classification for additive kernel SVMs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*, 66–77.

Moya, M., Koch, M., & Hostetler, L. (1993). One-class classifier networks for target recognition applications. In *World congress on neural networks* (pp. 797–801).

Mutch, J., & Lowe, D. G. (2006). Multiclass object recognition with sparse, localized features. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 11–18).

Perez, C., Salinas, C., Estvez, P., & Valenzuela, P. (2003). Genetic design of biologically inspired receptive fields for neural pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics, 33*(2), 258–270.

Phung, S. L., & Bouzerdoum, A. (2007). A pyramidal neural network for visual pattern recognition. *IEEE Transactions on Neural Networks, 18*(2), 329–343.

Ramya, C., Kavitha, G., & Shreedhara, K. S. (2011). Recalling of images using hopfield neural network model. *CoRR, 11*, 2–5.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In *Proceedings of the IEEE international conference on neural networks* (pp. 586–591).

Rolls, E., & Treves, A. (1998). *Neural networks and brain function*. Oxford, NY, USA: Oxford University Press.

Serre, T., Wolf, L., & Poggio, T. (2005). Object recognition with features inspired by visual cortex. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 994–1000).

Sun, Y. (2000). Hopfield neural network based algorithms for image restoration and reconstruction. *IEEE Transaction on Signal Processing, 48*(7), 2105–2118.

Thompson, B. B., Marks, I. R. J., Choi, J. J., El-Sharkawi, M. A., Huang, M. -Y., & Bunje, C. (2002). Implicit learning in autoencoder novelty assessment. In *Proceedings of the IEEE international joint conference on neural networks* (pp. 2878–2883).

Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M., & Gelzinis, A. (1999). Soft combination of neural classifiers: A comparative study. *Pattern Recognition Letters, 20*, 429–444.

Wolf, L., Bileschi, S., & Meyers, E. (2006). Perception strategies in hierarchical vision systems. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2153–2160).

Wu, Y., Zheng, N., You, Q., & Du, S. (2007). Object Recognition by Learning Informative, Biologically Inspired Visual Features. In *Proceedings of the IEEE international conference on image processing* (pp. 181–184).