Scientific Research and Essays Vol. 7(16), pp. 1662-1673, 30 April, 2012
Available online at http://www.academicjournals.org/SRE
DOI: 10.5897/SRE11.1892
ISSN 1992-2248 © 2012 Academic Journals

Full Length Research Paper

Automatic detection of urban areas using the Hierarchical Temporal Memory of Numenta®

Alberto J. Perea[1]*, José E. Meroño[2], Ricardo Crespo[2] and María J. Aguilera[1]

[1]Department of Applied Physics, University of Cordoba, Campus Rabanales, 14071 Cordoba, Spain.
[2]Department of Graphics Engineering and Geomatics, University of Cordoba, Campus Rabanales, 14071 Cordoba, Spain.

Accepted 11 April, 2012

The aim of the study is to develop a new methodology for the identification of urbanized and not urbanized areas using high spatial resolution remote sensing data and adapting a new digital classification algorithm based on the functionality of the human Neocortex. We used multispectral Quickbird images for the classification of different urban areas in the Province of Cordoba (Spain). The Memory-Prediction Theory, implemented in the form of a Hierarchical Temporal Memory (HTM), was applied by means of the Nupic software, in order to obtain the classification of urban land cover types. HTM is a new computing technology that replicates the structure and function of the human neocortex. The conclusion indicates that Hierarchical Temporal Memory has great potential for extracting urban areas information from high satellite imagery and the 93.8% of the parcels has been well classified. As conclusion, this methodology can improve the level of automatization of digital classifications using high remote sensing data.

Key words: Hierarchical temporal memory, urban areas, memory-prediction theory, objects based classification, Neocortex.

INTRODUCTION

Nowadays very high resolution commercial satellite images from urban areas are now available for very important applications and they provide more detailed spatial information such as texture, shape size and context rather than spectral information. In contrast, the high resolution images from satellites, each pixel no longer refers to a complete object, character or area, but rather to a portion of the components of these, which means that classic techniques of classification based on pixels present some limitations (Wilkinson et al., 1991): (1) the spectral information contained in pixels is not sufficient in the majority of cases, such as to identify vegetation species or the types of surface cover; and (2) normally pixels include a radiometric mixture from their neighbors and consequently few zones have total homogeneity. In the area of the digital treatment of images, there is currently great interest in the development of new classification algorithms (Ayala and Menenti, 2002). The combination of spectral data with other sources of auxiliary data allows for the use of more information that can improve classifications (Abkar et al., 2000). Although their spatial resolution enables the identification of urban and sub-urban objects, these images are difficult to classify on a pixel-by-pixel basis due to their high level of information (Van der Sande et al., 2003). Consequently, classical algorithms of pixel based image analysis are becoming less important for high resolution classification (Antunes et al., 2003).

An alternative to a pixel based classification is the

*Corresponding author. E-mail: g12pemoa@uco.es. Tel: +34-957-218-553. Fax: +34-957-218-553.

classification based on objects that takes into account other information, such as the shape, textures and spectral information.

The detection of high vegetation in urban areas, such as parks and small forests next to residential areas, is needed for several applications and the urban development is one of the key issues facing land-use planning departments today. The extraction of large scale geographical information from very high resolution satellite images is an important research topic in urban studies, especially in areas with an elevated rate of urban changes, as a way to update the geographical information (Dinis et al., 2010). Several studies estimate vegetation in urban areas from satellite images for different purposes, such as carbon storage modeling (Myeong et al., 2006) or the comparison of vegetation occurrence in different cities (Small, 2007).

Matikainen et al. (2007) distinguished buildings, high vegetation and ground using a decision tree. A digital surface model (DSM) derived from last pulse laser scanner data was first segmented and segments were classified into classes 'ground' and 'building or tree'. Compared with a building map, a mean accuracy of almost 90% was achieved.

Hug (1996) identified and extracted surface objects from the data of imaging laser altimeters. Surface objects were detected by applying a morphology-based filter to the elevation data. Then, these surface objects were separated into artificial objects (buildings) and natural objects (vegetation) using surface reflectance data, and/or elevation 'texture' and surface orientation. The best classification error was 21% that was reached using reflectance data.

Vosselman et al. (2004) first separated bare earth LIDAR points from object points and then further classified the object points as building points or vegetation points. The overall classification accuracy over the three classes; bare earth, buildings and vegetation was 90%.

Rottensteiner et al. (2005) evaluated a method for building detection by the Dempster-Shafer fusion of LIDAR data and multispectral images. For that purpose, ground truth was digitised for two test sites with quite different characteristics. Using these data sets, the heuristic model for the probability mass assignments of the method and the contributions of the individual cues used in the classification process were evaluated. This investigation showed that the NDVI increases the correctness by up to 15% for smaller buildings.

Forlani et al. (2006) presented a three-stage frame work for a robust automatic classification of raw LIDAR data as buildings, ground and vegetation. First the raw data was filtered and interpolated over a grid. In the second stage, double raw data segmentation was performed and then geometric and topological relationships among regions were taking into consideration. Finally, a rule-based scheme was applied for the classification of the regions.

Zhang et al. (2006) presented a framework that applied a series of algorithms to automatically extract building footprints from airbone light detection and ranging (LIDAR) measurements. In the proposed framework, the ground and non-ground LIDAR measurements was first separated using a progressive morphological filter. To test the proposed framework urbanized areas were employed and the total of omission and commission errors for extracted footprints was about 12%.

Vögtle and Steinle (2000) presented a methodology for recognition and 3D reconstruction of buildings in urban environment. Laser scanning data and spectral information were used. The main advantages of this method was that the method is not limited to predefined building types like gable or hip roofed ones and it is not necessary to use data sources like digital city maps or digital cadastral maps.

On the other hand, new progresses in neuroscience have increased the knowledge about the organization and operation of the cerebral cortex. Therefore it's possible to apply its operation algorithms to the software, which was simplistic and had limited results using neuronal networks up to now.

For decades, most artificial intelligence researchers tried to build intelligent machines that did not closely model the actual architecture and processes of the human brain. One of the reasons was that neuroscience provided many details about the brain, but an overall theory of brain function that could be used for designing such models was conspicuously lacking.

A new theory called memory-prediction theory offers a large-scale framework of the processes in the human brain and invites computer scientists to use it in their quest of machine intelligence (Hawkins and Blakeslee, 2004).

The memory-prediction theory is based on the functioning of the human neocortex. It has a hierarchical network structure where each region performs the same basic operation (Hawkins and Blakeslee, 2004).

Hawkins and Blaskeslee (2004) focus their theory on a unified model of how the human neocortex works, but in truth you do not need to have deep interest in neurobiology to see the power of the model. The basic idea is as follow: the brain uses large amounts of memory to create a hierarchical model of the world and uses it to create, by analogy, continuous predictions about future events.

A hierarchical network structure guides the functioning of each region in the cortex. All regions in the hierarchy perform the same basic operation. The inputs to the regions at the lowest levels of the cortical hierarchy

come from our senses and are represented by spatial and temporal patterns. The neocortex learns sequences of patterns by storing them in an invariant form in a hierarchical neural network. It recalls the patterns auto-associatively when given only partial or distorted inputs. The structure of stored invariant representations captures the important relationships in the world, independent of the details. The primary function of the neocortex is to make predictions by comparing the knowledge of the invariant structure with the most recent observed details.

Parts of this theory, known as the Memory-Prediction Theory (MPT), are modeled in the Hierarchical Temporal Memory or HTM technology developed by a company called Numenta®.

HTM in urban environments has a lot of direct applications in land management issues. And thanks to the development of a large number of platforms mapping and navigation tools on the Internet (such as Google Maps or Google Earth) is necessary to develop programs and applications oriented to recognition and automatic classification of satellite images.

Taking advantage of this, they must integrate new technologies emerging in the process of classification and recognition in images.

The new technology of hierarchical temporal memory is able to develop processes of recognition and pattern classification in images with good results for the requirements discussed.

There are still few applications of the HTM algorithm to the analysis or classification of high spatial remote sensing data. Perea et al. (2009) carried out a land use classification of digital aerial photographs using a network based on the Hierarchical Temporal Memory. A photogram received by a photogrammetric UltracamD® sensor of Vexcel, and data on 1513 plots in Manzanilla (Huelva, Spain) were used to validate the classification, achieving an overall classification accuracy of 90.4%.

The general goal of this paper is to propound a methodology based on the Hierarchical Temporal Memory model, proposed by Numenta®, to improve the methodologies used nowadays in the characterization of urban areas. To achieve this, QuickBird imagery is used to develop an algorithm based on the Hierarchical Temporal Memory of Numenta® to produce updated thematic cartography for municipal activities.

MATERIALS AND METHODS

Study area

The area of study was located in Cordoba city, Spain (37º53'5'' N; 4º46'44'' O). This is a rectangular area of 18 x 9 km and covers 16200 ha (Figure 1). This area was selected for two reasons:

(1) It is an area known by the authors, thus facilitates the search areas of each class.
(2) To work with images of an urban area because the elements of the territory in this type of environments have geometric patterns, different between them (e.g. building rooftops patterns are very different from the shape of a tree). The aim is to study the processes of recognition and classification in this type of geometric patterns.

QuickBird images and preprocessing

One multispectral image was used (QuickBird, Ortho Ready Standard Imagery, Digital Globe, Longmont, Colorado, USA), in UTM coordinates (Universal Transverse Mercator) and georeferenced in the WGS84 system. This image was ortorrectified and referred to the European Datum 1950 of the International Ellipsoid. The image was codified in 16 bits with a resolution of 2.4 m and was composed of four bands (blue, green, red and near infrared). The image was taken on 16 August 2007, beginning at 10:56, with a solar elevation angle of 59.8°. The system was developed to distinguish the following land covers: Road, Building and Park (Figure 2).

The process began with the radiometric correction of the images. Radiometric corrections modify the original digital levels to assimilate them to values that will present the image in the case of ideal reception. QuickBird images already have a series of radiometric corrections that the distributing company applies to its commercial products. The main corrections in the images are: restoration of lost pixels from the image or the possible loss or addition to the image.

A transformation of the digital levels at radiance values in the atmospheric ceiling was made and a reflectivity image was obtained. The conversion to the spectral radiance of the atmospheric ceiling can be done simply in two steps: the value of the corrected pixels is multiplied by the appropriate absolute calibration factor and the result is divided by the effective bandwidth to obtain spectral radiance. The radiometric calibration factor is included in the metadata files of the image.

HTM configuration

Hierarchical Temporal Memory (HTM) is a machine learning technology that aims to capture the structural and algorithmic properties of the neocortex (Numenta Inc., 2010).

HTMs can be considered a form of Bayesian network where the network consists of a collection of nodes arranged in a tree-shaped hierarchy (Hawkins and George, 2007). Each node in the hierarchy self-discovers a set of causes in its input through a process of finding common spatial patterns and then detecting common temporal patterns (Numenta Inc., 2010). Unlike many Bayesian networks, HTMs are self-training, have a well-defined parent/child relationship between each node, inherently handle time-varying data, and afford mechanisms for covert attention. Sensory data are presented at the "bottom" of the hierarchy. To train an HTM, it is necessary to present continuous, time varying, sensory inputs while the causes underlying the same sensory data persist in the environment. In other words, you either move the senses of the HTM through the world, or the objects in the world move relative to the HTM's senses. Time is the fundamental component of an HTM, and can be thought of as a learning supervisor. HTM networks are made of nodes; each node receives as input a temporal sequence of patterns. The goal of each node is to group input patterns that are likely to have the same cause, thereby forming invariant representations of extrinsic causes.

An HTM node uses two grouping mechanisms to form invariants.

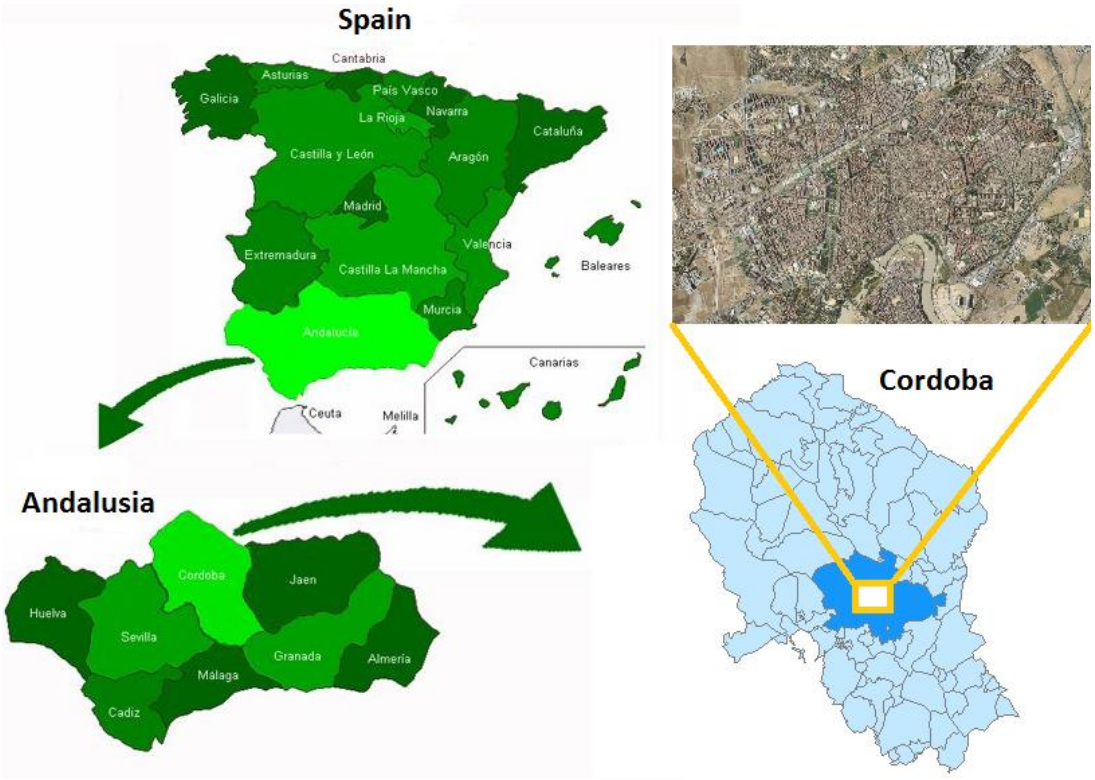

Figure 1. Investigated area and satellite imagery coverage.

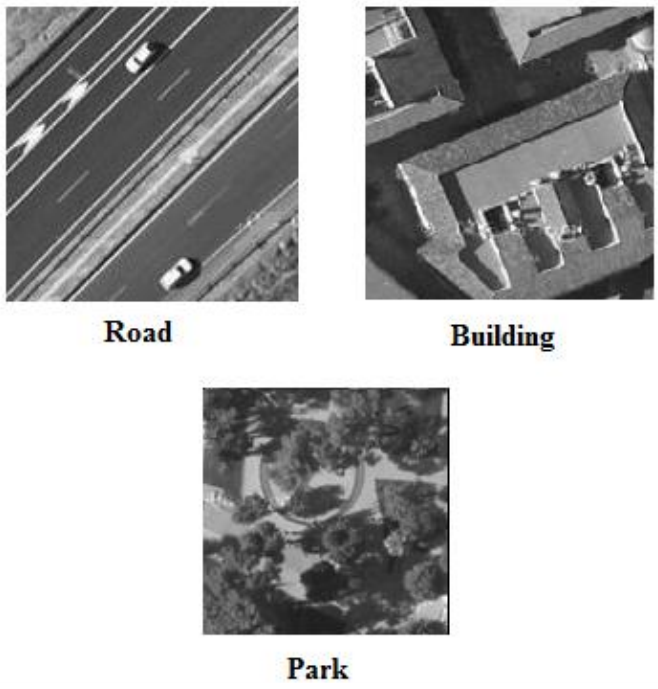

Figure 2. Classified categories.

Table 1. The number of training and test images.

Category	Training images	Testing images
Road	300	150
Building	300	150
Park	300	150

The first one is called spatial pooling, which receive raw data from the sensor; spatial poolers of higher nodes receive the outputs from their child nodes. The input of the spatial pooler in higher layers is the fixed-order concatenation of the output of its children. This input is represented by row vectors, and the role of the spatial pooler is to build a matrix (the coincidence matrix) from input vectors that occur frequently. There are multiple spatial pooler algorithms, e.g. Gaussian and 'Product'. The Gaussian spatial pooler algorithm is used for nodes at the input layer, whereas the nodes higher up the hierarchy use the 'Product' spatial pooler. The Gaussian spatial pooler algorithm compares the raw input vectors to the existing coincidences in the coincidence matrix. If the Euclidean distance between an input vector and an existing coincidence is small enough, the input is considered to be the same coincidence, and the count for that coincidence is incremented and stored in memory.

The 'Product' spatial pooler is always part of a node higher up the hierarchy, and receives the concatenation of the outputs of its child nodes. This vector is divided up into N portions, which is the number of children of the node. The 'Product' spatial pooler sets the highest value in each of these N distributions to 1, while the other values are set to 0. These new vectors are stored in the coincidence matrix, and the counts of the coincidences that already exist are incremented.

The second mechanism is called temporal pooling, which groups together patterns that are temporally close. This way, patterns that are very different, but that have a common cause, can be in the same group.

Both the spatial and temporal poolers switch from learning to inference mode at some point. In the case of the spatial pooler, its output is a vector of length equal to the number of patterns pooled by the node, and the ith position in this vector corresponds to the ith pattern inside this spatial pooler.

This output is a probability distribution of the similarity between the input pattern and the stored patterns, measured in terms of Euclidean distances. An assumption commonly made by the designers of HTM is that the probability that a pattern is closest to another pattern falls off as a Gaussian function of the Euclidean distance, therefore it can be calculated as proportional to $e^{-\frac{d_i^2}{\theta^2}}$ in a node, and the outputs of the spatial pooler are the inputs of the temporal pooler. As mentioned before, the temporal pooler forms groups of patterns that are likely to follow each other in time, since it would indicate that they are likely to have the same cause in the world.

The designers of HTM used a time-adjacency matrix partitioned with a 'greedy' algorithm. This algorithm creates groups by finding the most-connected pattern that is not part of a group, and picking the N most-connected patterns to this pattern recursively (Hawkins and George, 2007). For every input from the spatial pooler, the temporal pooler outputs a probability distribution over its groups, propagating the uncertainties up in the hierarchy in a Bayesian Belief Propagation way. The ambiguous information propagated from the bottom of the hierarchy is resolved higher up in the hierarchy.

We used Nupic® (Numenta Platform for Intelligent Computing), software for implementing HTMs developed by Numenta to implement our HTM network. The company provides examples of how to create and use HTMs in various scenarios. One of these examples trains an HTM to recognize black and white pictures (one bit per pixel) with different levels of deformations. Another example uses an HTM to classify fruit images (grayscale, 8 bits per pixel). We adapted these examples to solve problems related to the classification of urban areas. The segmentation of images for the training and classification processes of the HTM network has been fixed at 200 x 200 pixels, which is a size large enough to recognize patterns but not excessive. Large images are slow down this process. This size has also been chosen to define classes clearly and crisp, without outside interference patterns for each class.

To implement an HTM, two steps have to be taken: creating the architecture, and training it with a set of training patterns. After, we created architecture and trained the network on the remote sensing data train set and finally we tested the HTM with test set. Table 1 shows the number of training and testing images for the experiment.

HTM networks are built and configured by writing Python scripts. While the majority of the scripts follow a standard pattern, each network requires customization. One must leverage in-depth knowledge of data to design and configure the hierarchy of nodes. Each node algorithm needs to be customized based on the input values it is encountering. Owing to the large number of node parameters, node configuration values will most likely be 'tweaked' after each iteration in order to improve accuracy. The network structure usually remains the same, reducing the amount of code that must be changed.

Our HTM consists of 7 levels, three levels each with two sub-levels (a level which analyzes the spatial component and other level which analyze the temporal component) and a final classifier. It is the final element of the hierarchy and classifies the image into common categories (Figure 3). Through the parameter outputElementCount, the number of categories can be defined, three in this case.

MaxDistance on the first level defines the minimum value that the squares of the Euclidean distances between an input (x) and all the previously memorized inputs (y_i) have to take in order for x to be considered novel. maxGroupSize sets an upper limit for the number of quantized inputs that can form a group in the temporal pooler. The pooler algorithm used by the spatial pooler of higher levels is 'product', which means that the belief that an input during inference is similar to a given vector (previously memorized by the spatial pooler) is calculated as follows:

$$belief_i = \prod_{j=1}^{nchildren} y_i[child_j] * x\,[child_j] \quad (1)$$

where nchildren is the number of children the node has, x is the input vector, y_i are the vectors previously stored by the spatial pooler, and $a[child_n]$ is the part of vector a that is received from the n^{th} child.

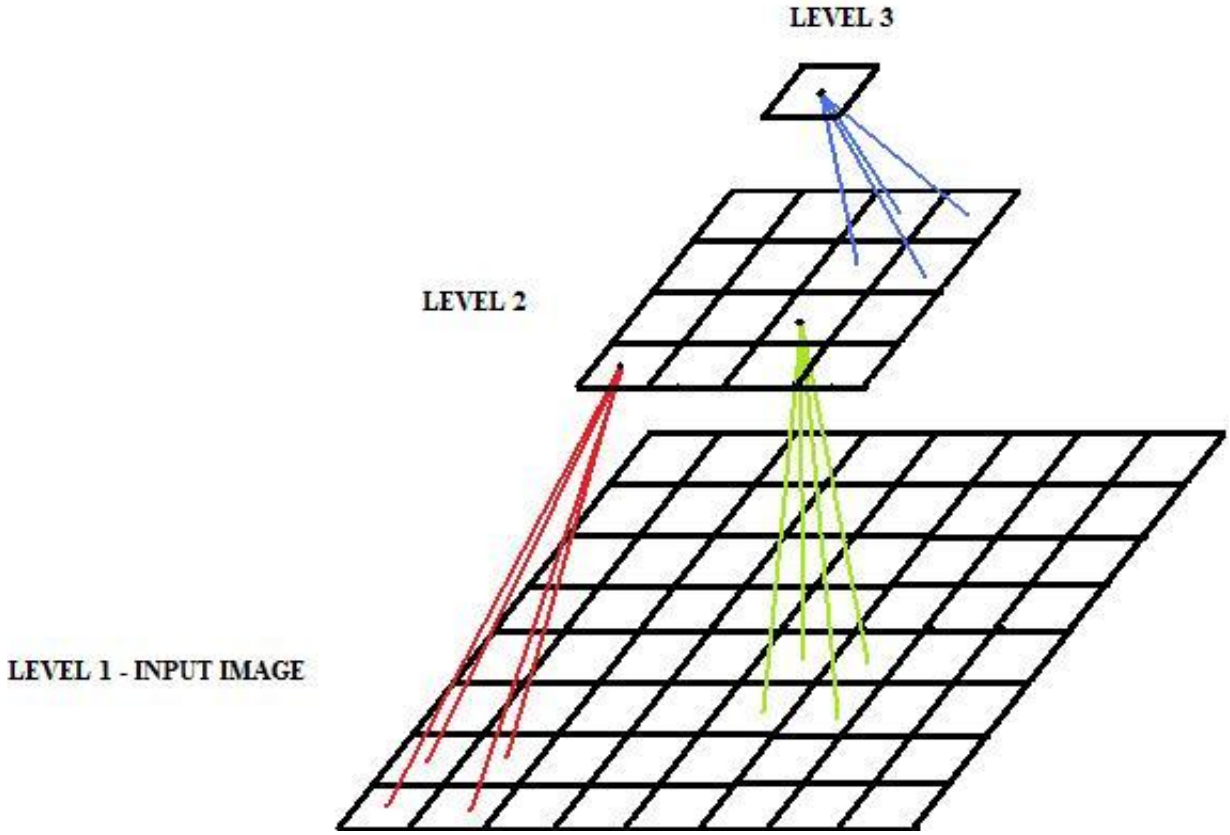

Figure 3. HTM with three layers of nodes.

Finally, the temporal pooler at each level uses the sumProp algorithm, which takes the highest belief from each group to generate a distribution of beliefs over temporal groups during inference.

Other parameters related with the scale of the images are; scaleRF- An integer specifying the number of scales (resolutions) in the multi-resolution topology from which each node should receive input. For example, a value of 2 means that each node should receive input from 2 scales. Note that unless scaleRF is 1, the number of resolutions seen by the parent level will be lower than the number seen at the current level. scaleOverlap; an integer specifying how many scales neighboring nodes should share in common. For example, if scaleRF is 2, scaleOverlap is 1, and there are 3 resolutions in the level below, some nodes will see the smaller and middle resolutions, and some nodes will see the middle and larger resolutions.

Training phase

Once the network is built, defining the architecture through which information flows, we set up the training process and the information processing. Thus, the key parameter is the number of iterations performed using the training images. In this case we have performed 2000 iterations in three levels. It has been shown experimentally that, if the iterations are increasing to the double value (4000), it is not observed a significant increase of accuracy in the analysis. NuPIC has a user interface that allows interacting with the network while the analysis process is carried out.

In Figures 4 and 5, the training of temporal pooler of level 1, sub-level 2 is showed. Next to the training image, a representation of information received by the spatial node of the first spatial pooler: GaborNode is also presented (Numenta Inc., 2008).

Inference phase

Once the network has been trained with the database provided, stating the categories, the inference stage is starting, where unknown images are analyzed by the network, according to the learned and memorized in the previous stage.

Figures 6 and 7 present the system working on the inference stage. Again, it shows the status of any of the nodes while the network is processing information. In Figure 6, the first sub-level of the node is shown, the GaborNode, which creates a representation according to the known patterns of similar shape and texture. After finishing the recognition and inference phases, the application shows the evaluation process.

RESULTS

We verified the capability of the model to learn invariant

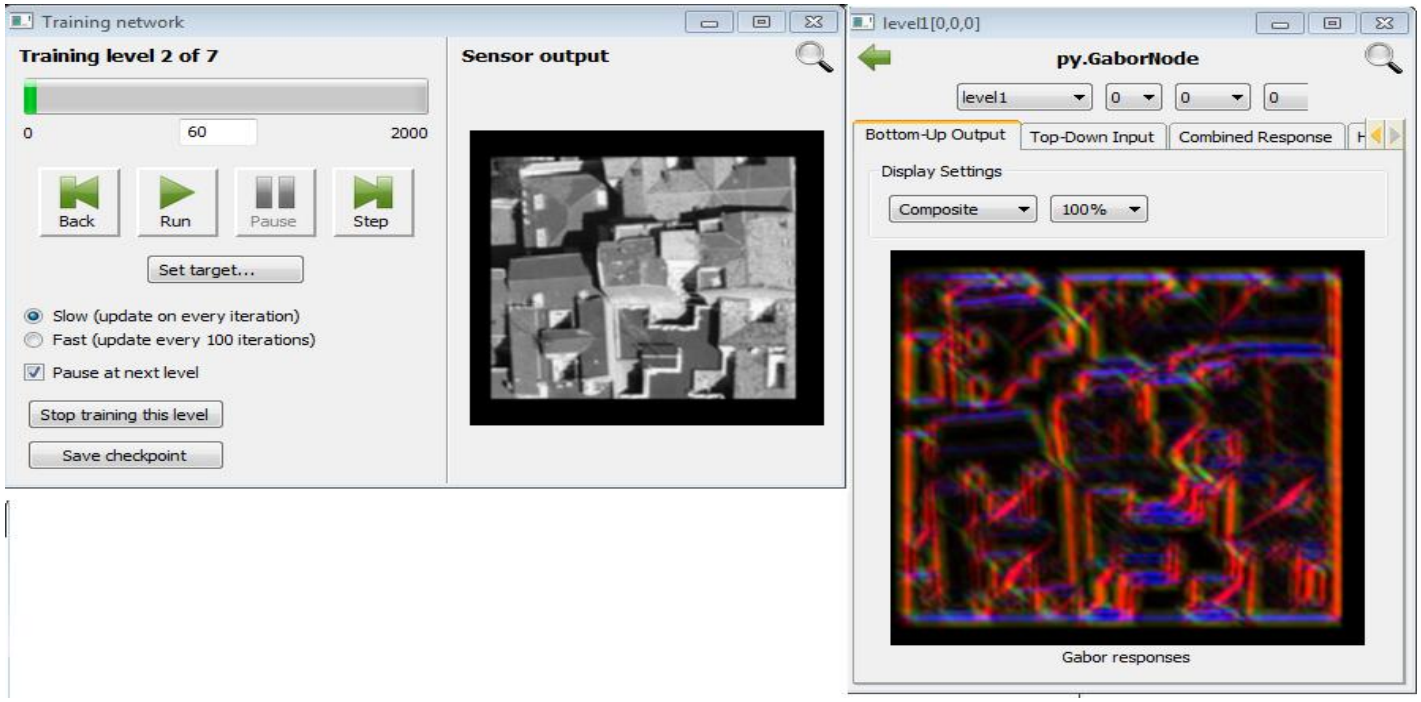

Figure 4. Training stage of level 1, sub-level 2.

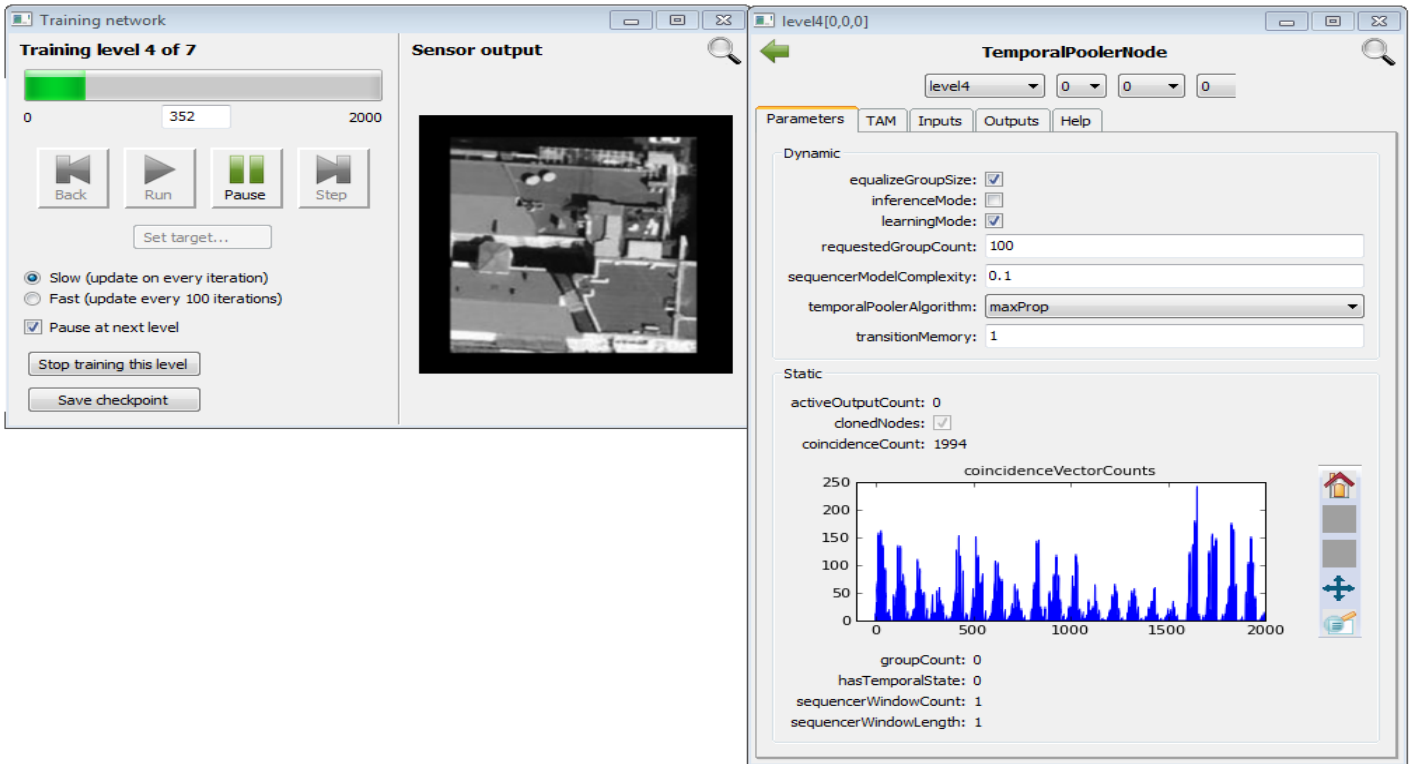

Figure 5. Training stage of level 2, sub-level 2.

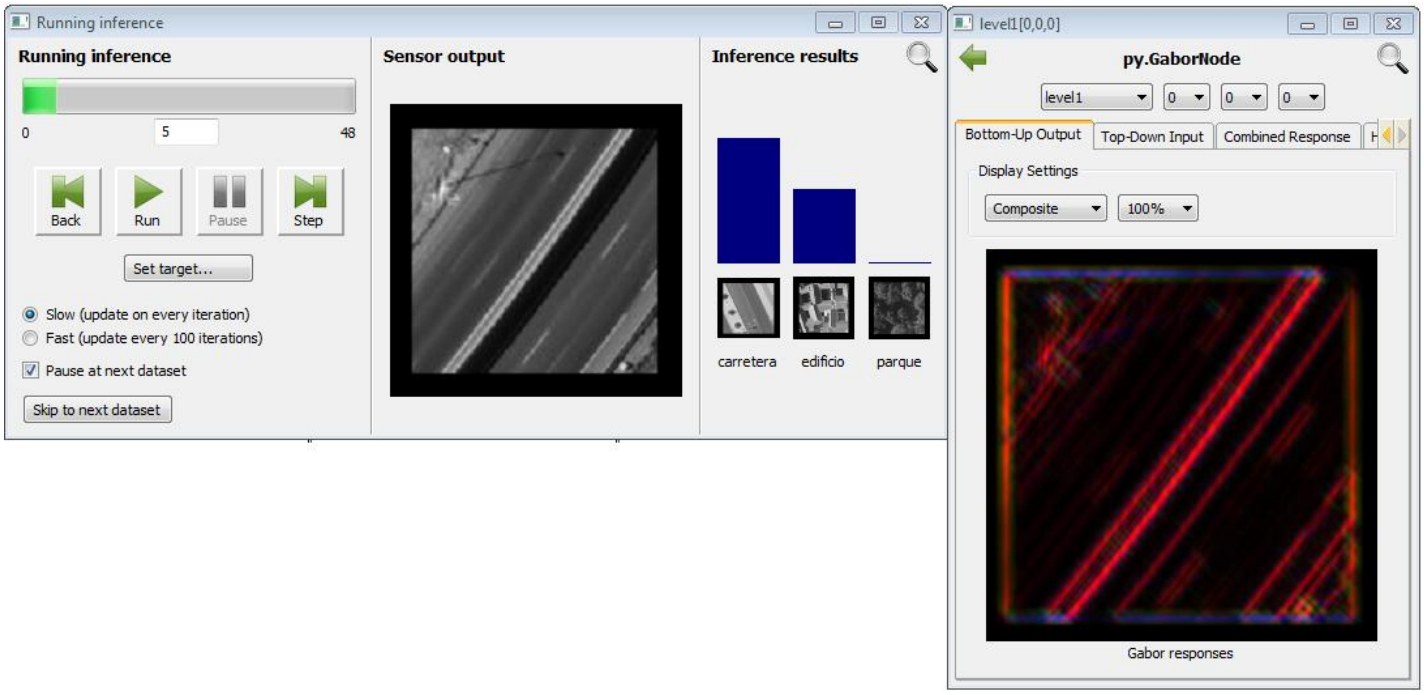

Figure 6. Example of inference stage.

representations from visual patterns and to store these patterns in the hierarchy and recall them auto-associatively. During the experimentation, we varied some internal parameters affecting the learning process, and also made modifications to the algorithms and data structures themselves. The most important parameters modified are (Numenta Inc., 2008):

Sparsify: Specifies whether a stored pattern is sparsified or not. If sparsify is true, then some components of the stored pattern are zeroed out. In domains like vision and speech, sparsifying the stored coincidence patterns increases the recognition and generalization performance. In this experiment the sparsify was true.

Sigma: During a node's inference stage, an input pattern is compared to the stored patterns assuming that the stored patterns are centres of radial basis functions with Gaussian tuning. The sigma parameter specifies the standard deviation of this Gaussian. The best value for this parameter and this type of images was 0.15.

RequestedGroupCount: It is a parameter that determines the maximum number of temporal groups that will be formed. The best value for this parameter and this type of images was 24.

TransitionMemory: This parameter determines how far back in time the temporal pooler node looks for temporal transitions. Having a high transition Memory has the effect of smoothing out the temporal transitions so that temporal jitter and repeated states in the input sequence do not produce undesired behaviour. Larger values of transitionMemory were used for the higher level nodes as the Numenta Node Algorithms Guide recommends.

We also investigated the effect of the parameters Maxdistance, scaleRF, scaleOverlap and the number of iterations on overall accuracy, the average number of coincidences and temporal groups learned in the bottom-level nodes.

The scaleRF and scaleOverlap parameters have been modified to obtain more accuracy results. These parameters are related to the image scale during the training phase, where the node receives the same information. The higher accuracy was obtained for a value of 3 for scaleRF and 2 for scaleOverlap. The maximum value obtained was 93.78%.

However, increasing the maximum distance to consider a new similar pattern to one already stored on the network, disparate patterns, unrelated to space, can be more easily classified as patterns that do not belong to the same class. The results obtained in this case show a decreasing of accuracy to 68.45%.

By modifying the training process, significantly increasing the number of iterations (up to 4000 by level of nodes), no change is observed with minimal

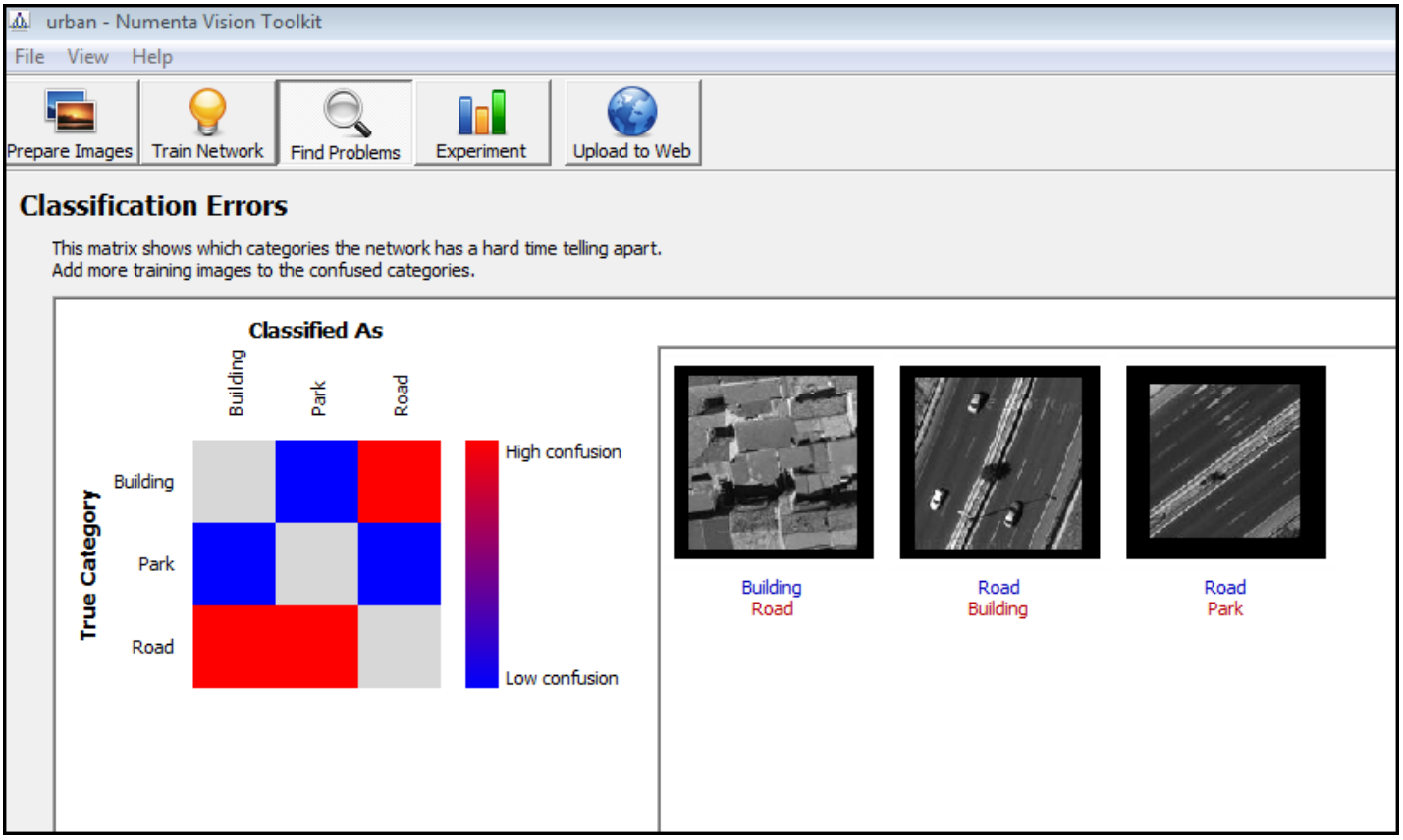

Figure 7. Classification errors during the inference stage.

representative.

DISCUSSION

Three parameters have been modified during the classification; MaxDistance parameter, which defines the minimum Euclidean distance between a known pattern and a new one, is critical in the pattern classification and recognition. In the original configuration, it is based on an intermediate parameter (0.2) and we studied the variation of this parameter between values. By reducing the parameter is expected to get a tighter result in the classification rate as we are telling the network that is stricter when classifying a pattern as known.

The other two parameters changed (scaleRF and ScaleOverlap) are related to the scale or resolution of the images presented to the network, so that changing these parameters we can modify the number of different scales which are presented and the overlap between them. This change is critical as the changes of the same image resolution allow the network to extract patterns of the same image at different levels so as to create better invariant representations (or models of memorized patterns) used to classify new images.

The basic network, as the original method, starts from intermediate values for ScaleOverlap and ScaleRF (2 and 1 respectively). They have been reduced to a minimum (a single resolution) and increased. Table 2 shows that the best overall accuracy is related to the increasing of the values for these parameters.

Overall, changing the minimum distance of pattern recognition and the patterns related to the number of resolutions, the network has the ability to extract patterns from an image at different resolutions.

Overall classification accuracy is 98% for parks, while for roads its value is lower, only 87%. It is confused mostly with building but also with parks. Finally, good results of the classification can be observed, obtaining an overall accuracy of 93.8% and problems associated to the use of high spatial resolution images have been resolved to a large extent, as in the case of the salt and

Table 2. Overall accuracy for different values of scaleRF and ScaleOverlap.

scaleRF	scaleOverlap	Overall accuracy
1	1	78.94
2	1	81.3 %
3	2	93.78 %

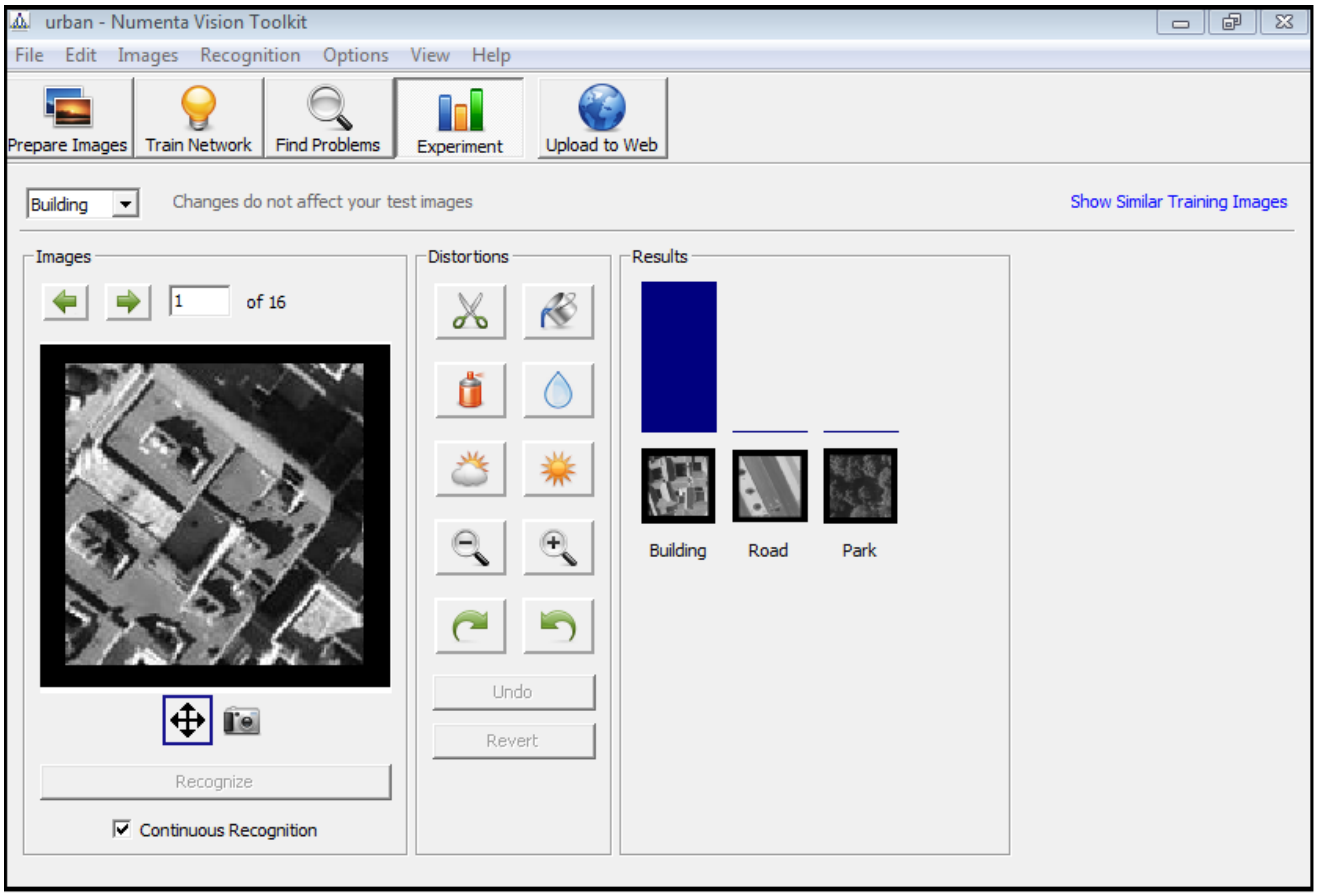

Figure 8. Example of classified image as building land use.

pepper effect (Figures 8 and 9).

The accuracy values obtained with the algorithm based on the Hierarchical Temporal Memory were similar to and/or higher than the values obtained by other authors, which shows that the methodology is adequate for detecting urban areas.

In the city of Nanjing, eastern China, new method based on Normalized Difference Built-up index to automate the process of mapping built-up area was proposed. The overall accuracy was 92.6% (Zha et al., 2003).

Mathieu et al. (2007) applied the oriented based classification to very high resolution multispectral Ikonos images to produce vegetation community maps in Dunedin City, New Zealand. The overall classification accuracy from the simplified classification was 77% with a κ value close to the excellent range (κ = 0.74).

Conclusions

This experiment analysis and classification of images of urban environments has been developed using the new technology of Hierarchical Temporal Memory algorithm (HTM). Several repetitions of the analysis of sample

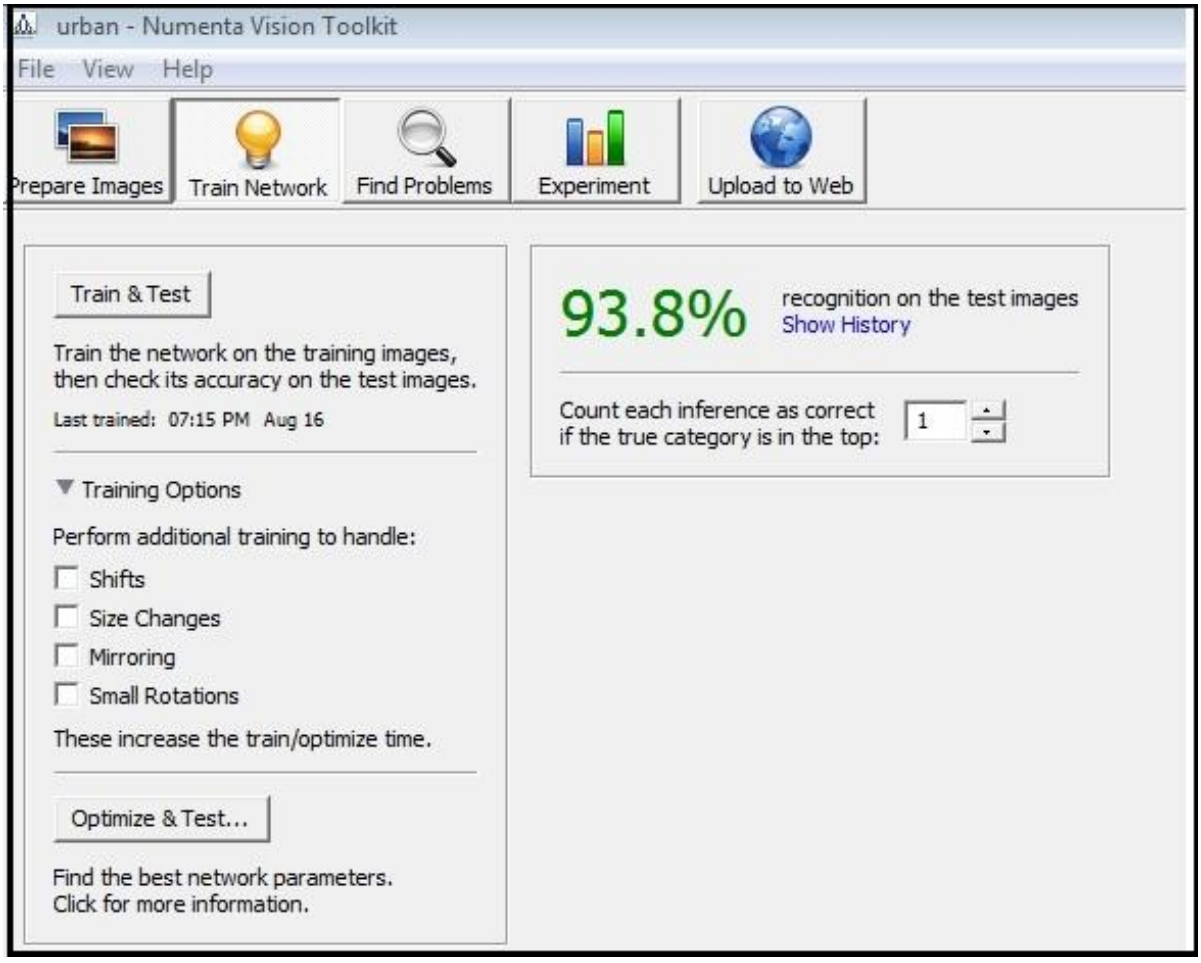

Figure 9. Recognition on the test images.

images have been carried out modifying various parameters of both architecture and inner workings of the network during the learning and analysis process.

This is a versatile tool in image analysis and in recognition-oriented classification that allows us to easily modify, without strong technical and programming requirements, the architecture of the network and to optimize its internal operations according to the information used.

The best result was obtained with the optimization of the parameters related with the image scale during the training phase, achieving a global classification accuracy of 93.8%.

REFERENCES

Abkar AA, Sharifi MA, Mulder NJ (2000). Likelihood-based image segmentation and classification: A framework for integration of expert knowledge in image classification procedures. Int. J. Appl. Earth Obs. Geoinf., 2: 104-119.

Antunes AFB, Lingnau C, Da Silva JC (2003). Object Oriented Analysis And Semantic Network For High Resolution Image Classification. In: Anais XI SBSR. Belo Horizonte, pp. 273-279.

Ayala RM, Menenti M (2002). Alternativas a los problemas presentados en un proceso de clasificación basado en el reconocimiento espectral de patrones. Mapping, 75: 72-76.

Dinis J, Navarro A, Soares F, Santos T, Freire S, Fonseca A, Afonso N, Tenedório J (2010). Hierarchical Object-based classification of dense urban areas by integrating high spatial resolution satellite images an LIDAR elevation data. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Ghent, Belgium, 38 (Part 4/C7): 280-286.

Forlani G, Nardinocchi C, Scaioni M, Zingaretti P (2006). Complete classification of raw LIDAR data and 3D reconstruction of buildings. Pattern Anal. Appl., 8(4): 357-374.

Hawkins J, Blakeslee S (2004). On Intelligence. New York, USA: Henry Holt., pp. 296.

Hawkins J, George D (2007). Hierarchical Temporal Memory, Concepts, Theory, and Terminology. Technical report. Numenta, Inc.: Redwood city, USA. Available from: http://www.numenta.com/htm-overview/education/Numenta_HTM_Concepts.pdf

Hug C (1996). Extracting artificial surface objects from airborne laser

scanner data. In: Gruen A, Baltsavias EP, Henricsson O (Eds) Automatic Extraction of Man-Made Objects from Aerial and Space Images (II): Birkhäuser Verlag: Basel, Switzerland, pp. 203-212.

Mathieu R, Aryal J, Chong AK (2007). Object-based classification of Ikonos imagery for mapping large-scale vegetation communities in urban areas. Sensors, 7: 2860-2880.

Matikainen L, Kaartinen H, Hyyppä J (2007). Classification tree based building detection from laser scanner and aerial image data. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Espoo, Finland, 36(Part 3/W52): 280-286.

Myeong S, Nowak DJ, Duggin MJ (2006). A temporal analysis of urban forest carbon storage using remote sensing. Remote Sens. Environ., 101: 277-282.

Numenta Inc (2008). Advanced Nupic Programming. Technical report. Numenta, Inc.: Redwood city, USA. Available from: http://www.numenta.com/archives/education/nupic_prog_guide.pdf

Numenta Inc (2008). Numenta Node Algorithms Guide-Nupic 1.7Technical report. Numenta, Inc.: Redwood city, USA. Available from: http://blog.mohammadzadeh.info/media/blogs/snf/Resources/HTM/NodeAlgorithmsGuide.pdf?mtime=1296775184

Numenta Inc (2010). HTM cortical Learning Algorithms. Technical report. Numenta, Inc.: Redwood city, USA. Available from: http://www.numenta.com/htm-overview/education/HTM_CorticalLearningAlgorithms.pdf

Perea AJ, Meroño JE, Aguilera MJ (2009). Application of Numenta® Hierarchical Temporal Memory for land-use classification. S. Afr. J. Sci., 105(9-10): 370-375.

Rottensteiner F, Summer G, Trinder J, Clode S, Kubik K (2005). Evaluation of a method for fusing LIDAR data and multispectral images for building detection. In: International Archives of Photogrammetry, Remote Sens. Spatial Inf. Sci. Vienna, Austria, 36: Part 3/W24.

Small C (2007). Comparative analysis of urban vegetation scale and abundance. In: Urban Remote Sensing Joint. Event, URBAN 2007 – URS 2007, Paris.

Van der Sande CJ, de Jong SM, de Roo APJ (2003). A segmentation and classification approach of IKONOS-2 imagery for land cover mapping to assist flood risk and flood damage assessment. Int. J. Appl. Earth Obs. Geoinf., 4: 217-229.

Vögtle T, Steinle E (2000). 3D modelling of buildings using laser scanning and spectral information. In: Int. Archiv. Photogramm. Remote Sens. Amsterdam, Netherlands, 38(Part B3): 927-934.

Vosselman G, Gorte B, Sithole G (2004). Change detection for updating medium scale maps using laser Altimetry. In: Int. Archiv. Photogramm. Remote Sens., 35(Part B3): 207-212.

Wilkinson GG, Kanellopoulos I, Kontoes C, Schoenmakers R (1991). Advances in the automatic processing of satellite images. In: Toselli, F., Meyer, J. (Eds) Conference on the Application of Remote Sensing to Agricultural Statistics, Belgirate, Italy, pp. 125-132.

Zha Y, Gao J, Ni S (2003). Use of normalized difference built-up index in automatically mapping urbana areas from TM imagery. Int. J. Remote Sens., 24(3):583-594.

Zhang K, Yan J, Chen SC (2006). Automatic construction of building footprints from airborne LIDAR data. In: IEEE Trans. Geosci. Remote Sens., 44(9): 2523-2533.