Application of Numenta® Hierarchical Temporal Memory for land-use classification

A.J. Perea[a*], J.E. Meroño[b] and M.J. Aguilera[a]

The aim of this paper is to present the application of memory-prediction theory, implemented in the form of a Hierarchical Temporal Memory (HTM), for land-use classification. Numenta® HTM is a new computing technology that replicates the structure and function of the human neocortex. In this study, a photogram, received by a photogrammetric UltraCamD® sensor of Vexcel, and data on 1 513 plots in Manzanilla (Huelva, Spain) were used to validate the classification, achieving an overall classification accuracy of 90.4%. The HTM approach appears to hold promise for land-use classification.

Key words: memory-prediction theory, NuPIC®, UltraCamD® sensor, Hierarchical Temporal Memory

Introduction

Vision is the primary sensory modality for humans—and most other mammals—by which they perceive the world. In humans, vision-related areas occupy about 30% of the neocortex.[1] Light rays are projected upon the retina, and the brain tries to make sense of the world by means of interpreting the visual input pattern. The sensitivity and specificity with which the brain solves this computationally complex problem cannot yet be replicated on a computer. The most imposing of these problems is that of invariant visual pattern recognition.

Recently it has been said that the prediction of future sensory input from salient features of current input is the keystone of intelligence.[2] The neocortex is the structure in the brain which is assumed to be responsible for the evolution of intelligence. Current sensory input patterns activate stored traces of previous inputs which then generate top-down expectations, which are verified against the bottom-up input signals. If the verification succeeds, the predicted pattern is recognised. This theory explains how humans, and mammals in general, can recognise images despite changes in location, size and lighting conditions, and in the presence of deformations and large amounts of noise. Parts of this theory, known as the memory-prediction theory (MPT), are modelled in the Hierarchical Temporal Memory or HTM technology developed by a company called Numenta®;[3] the model is an attempt to replicate the structural and algorithmic properties of the neocortex.[3] Spatial and temporal relations between features of the sensory signals are formed in an hierarchical memory architecture during a learning process. When a new pattern arrives, the recognition process can be viewed as choosing the stored representation that best predicts the pattern. Hierarchical Temporal Memory has been successfully applied to the recognition of relatively simple images,[4] showing invariance across several transformations and robustness with respect to noisy patterns.

We have applied the concept of HTM, as implemented by Numenta®, to land-use recognition, by building and testing a system to learn to recognise five different types of land use.

Overview of the HTM learning algorithm

Hierarchical Temporal Memory can be considered a form of a Bayesian network, where the network consists of a collection of nodes arranged in a tree-shaped hierarchy.[4] 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.[4] 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. Hierarchical Temporal Memory 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 (Fig. 1). The first mechanism is called spatial pooling, in which raw data are received by 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 with 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.

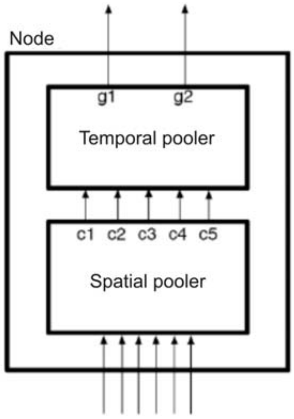


Fig. 1. The HTM node structure.[4]

[a]Dpto. Física aplicada, Edificio Einstein, University of Cordoba, Campus Rabanales, 14071 Spain.
[b]Dpto. Ingeniería Gráfica y Geomática, Edificio Gregor Méndel, University of Cordoba, Campus Rabanales, 14071 Spain.
*Author for correspondence E-mail: g12pemoa@uco.es

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 into *N* portions, with *N* being 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, by which patterns that are temporally close are grouped together. In 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 *i*th position in this vector corresponds to the *i*th 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 $e\frac{d_i^2}{e^2}$ 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.[4] For every input from the spatial pooler, the temporal pooler outputs a probability distribution over its groups, propagating the uncertainties in the hierarchy in a Bayesian belief propagation manner. The ambiguous information propagated from the bottom of the hierarchy is resolved higher in the hierarchy.

Materials and methods

The study area was located in the central plains of Huelva Province, Spain (Fig. 2), in the subregion known as Manzanilla (37°23′N; 6°25′W).

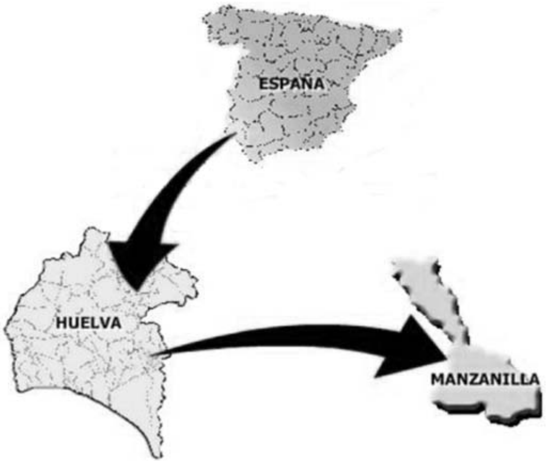


Fig. 2. A map showing the study area.

Digital aerial photograph

The dataset used in this research was a photogram received by a photogrammetric UltraCamD® sensor of Vexcel on 23 October 2007, with dimensions of 7 500 × 11 500 pixels. Its band combination was formed by red, green and blue. The digital aerial photographs had a special resolution of 30 cm. The photogram was segmented in small images of 128 × 128 pixels, as the HTM platform classifies only small images which contain only one pattern.

Map of crops and exploitation

A map of crops and exploitation of the region of Huelva (2007) was used to carry out the 'training' of the classification and its subsequent validation. The land use in this area is classified as: *Vitis vinifera* L. (vineyards), *Olea europaea* L. (olive groves), fallow land, irrigated land and built-up surface (Fig. 3). Table 1 shows the number of training and test images for the architecture demonstration.

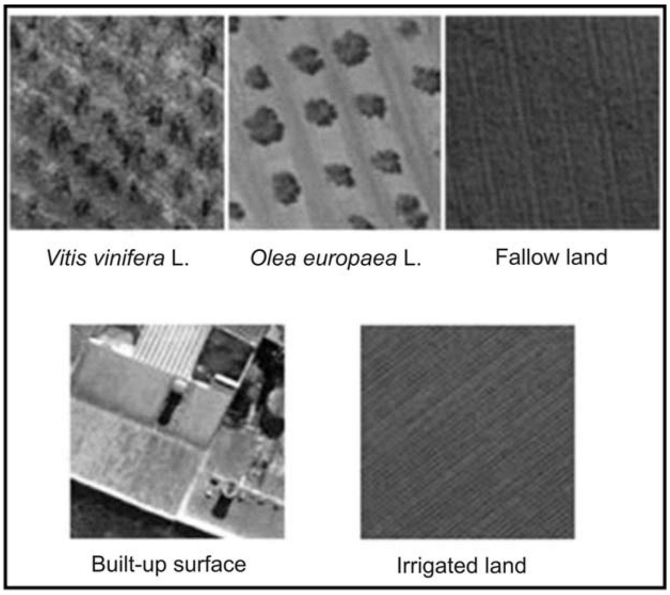


Fig. 3. Land-use classification within the study area.

We used NuPIC® (Numenta® Platform for Intelligent Computing) software developed by Numenta® for implementing HTMs 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 recognise black and white pictures (one bit per pixel) with different levels of deformations. Another example uses an HTM to classify fruit images (greyscale, eight bits per pixel). We adapted these examples to solve problems related to the classification of different land uses using small greyscale images (128 × 128 pixels), because HTMs classify only these kinds of images. To implement an HTM, two steps have to be undertaken: creating the architecture and training the architecture with a set of training patterns. After we created an architecture and trained the network on the digital aerial photographs, we tested the HTM with a test set.

Hierarchical Temporal Memory networks are built and config-

Table 1. The number of training and test images.

Category	Training images	Test images
Vitis vinifera L.	300	150
Irrigated land	300	150
Olea europaea L.	300	150
Fallow land	300	150
Built-up surface	300	150

ured by writing Python scripts. While the majority of the scripts follow a standard pattern, each network requires customisation. One must leverage in-depth knowledge of the data to design and configure the hierarchy of nodes. Each node algorithm needs to be customised based on the input values it is encountering. Because of 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 model consisted of three levels (Fig. 4): Level 1 (the input level) consisted of 16 nodes, each receiving a feature and the corresponding delta; Level 2 consisted of four nodes, each receiving the output of four input level child nodes; and Level 3 consisted of one top level node.

The parameters of the HTM network used were as follows:

Level 1:
- levelSize = 64
- pooler algorithm: gaussian; sigma = 0.4
- maxDistance = 5
- maxGroupSize = 1 435
- grouper algorithm: sumProp

Level 2:
- levelSize = 4
- pooler algorithm: product
- maxGroupSize = 1 435
- grouper algorithm: sumProp

Level 3:
- levelSize = 1
- pooler algorithm: product
- mapper algorithm: sumProp

'maxDistance' on the first level defines the minimum value that the squares of the Euclidean distances between an input (x) and all the previously memorised inputs (y_i) have to take in order for x to be considered novel. 'maxGroupSize' sets an upper limit for the number of quantised 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 memorised by the spatial pooler) is calculated as follows:

$$\text{belief}_i = \prod_{j=1}^{n\ \text{children}} y_i[\text{child}_j] * x[\text{child}_j]$$

where n children 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 $a[\text{child}_n]$ is the part of vector a that is received from the nth child.

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.[5]

This type of hierarchical network structure is analogous to the hierarchy of the visual region in human neocortex, which also is organised as a hierarchy of cortical regions. The receptive field size in the cortical regions also gradually increases in the higher levels of the hierarchy. The neural structures in higher regions of the cortex represent increasingly complex structures and the structures in the top visual region represent visual objects, just as they do in this model.

Accuracy evaluation and validation

The accuracy evaluation is a general term to compare the generated classification with known geographical information. Its main aim is therefore to determine the veracity of the classification process. A true-terrain image from the information contained in the crop maps and exploitations of the region of Huelva was prepared. The statistics used were: producer's accuracy, user's accuracy, overall accuracy and Kappa statistic.

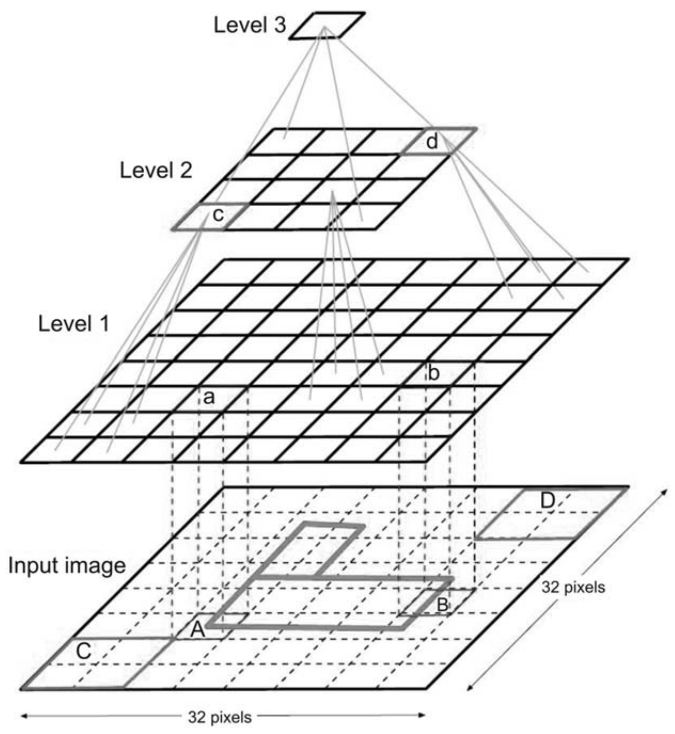


Fig. 4. The HTM model with three layers of nodes.[4]

The Kappa statistic is a measure of the difference between the observed accuracy and the random possibility of chance agreement between the reference data and the classification.[6] When the total number of correctly classified pixels in a class is divided by the total number of pixels that should have been classified in that class, it is known as the producer's accuracy.[7] If the total number of correctly classified pixels in a class is divided by the total number of pixels that were actually classified in that class (both correctly and incorrectly), the result is a measure of the user's accuracy.[7] The overall accuracy is the percentage of correctly classified pixels. We used Numenta® Vision Test App® software to validate our HTM network.

Results and discussion

The methodology proposed was applied to the region of study obtaining a final classification of land use. Table 2 shows the accuracy of classification in the digital aerial photograph according to its boundary analysis. The highest producer's accuracy was achieved for the 'built-up surface', having a value of 100%, while the lowest producer's accuracy was for '*Vitis vinifera* L.' (81.33%).

In the case of the user's accuracy, the highest value again was obtained for the 'built-up surface' class (100%), while the lowest corresponded to '*Olea europaea* L.' (73.03%). The HTM classification thus gave an high overall accuracy of 90.4%, and the Kappa statistic had a value of 0.80, showing that the classification was 80% better than a random one.

We also verified the capability of the model to learn invariant representations from visual patterns and to store these patterns in the hierarchy and recall them auto-associatively. During the study, we varied many internal constants affecting the learning process, and also made modifications to the algorithms and data structures themselves. Figures 5–8 illustrate the main recognition capabilities of the system, trained to recognise five categories of

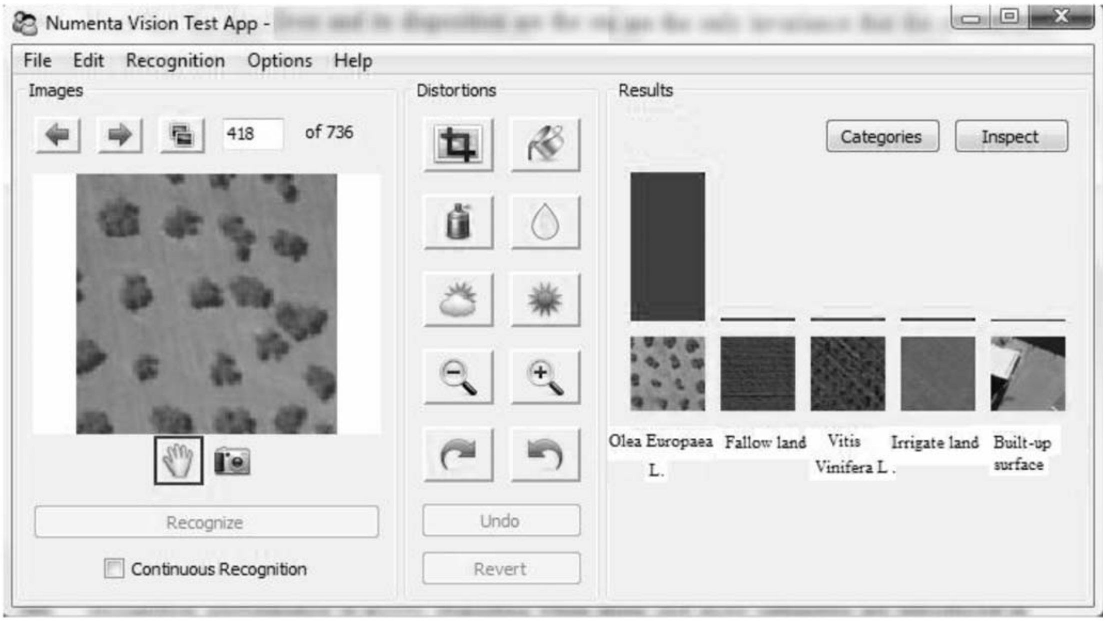


Fig. 5. Original image of the category '*Olea europaea* L.'.

images. One of the two original training images in the category '*Olea europaea* L.' is shown in Fig. 5. The system easily recognised the shifted version of the original image, shown in Fig. 6.

Note that the number of the '*Olea europaea* L.' and their arrangement were the only invariants that the system was explicitly exposed to during the training; hence the other invariants described below were discovered automatically by the system.

The system can function as an auto-associative memory, as demonstrated in Fig. 7. Given a part of the original image, the

Table 2. Producer's accuracy, user's accuracy, overall accuracy and Kappa statistic of the HTM classification obtained from the digital aerial photograph.

Category	*Vitis vinifera* L.	Irrigated land	*Olea europaea* L.	Fallow land	Built-up surface	Total
Vitis vinifera L.	**122**		28	0	0	150
Irrigated land	1	**134**	12	3	0	150
Olea europaea L.	20	0	**130**	0	0	150
Fallow land	0	0	8	**142**	0	150
Built-up surface	0	0	0	0	**150**	150
Producer's accuracy	81.33%	89.33%	86.66%	94.66%	100%	
User's accuracy	85.31%	100%	73.03%	97.93%	100%	
Overall accuracy	90.4%					
Kappa statistic	0.80					

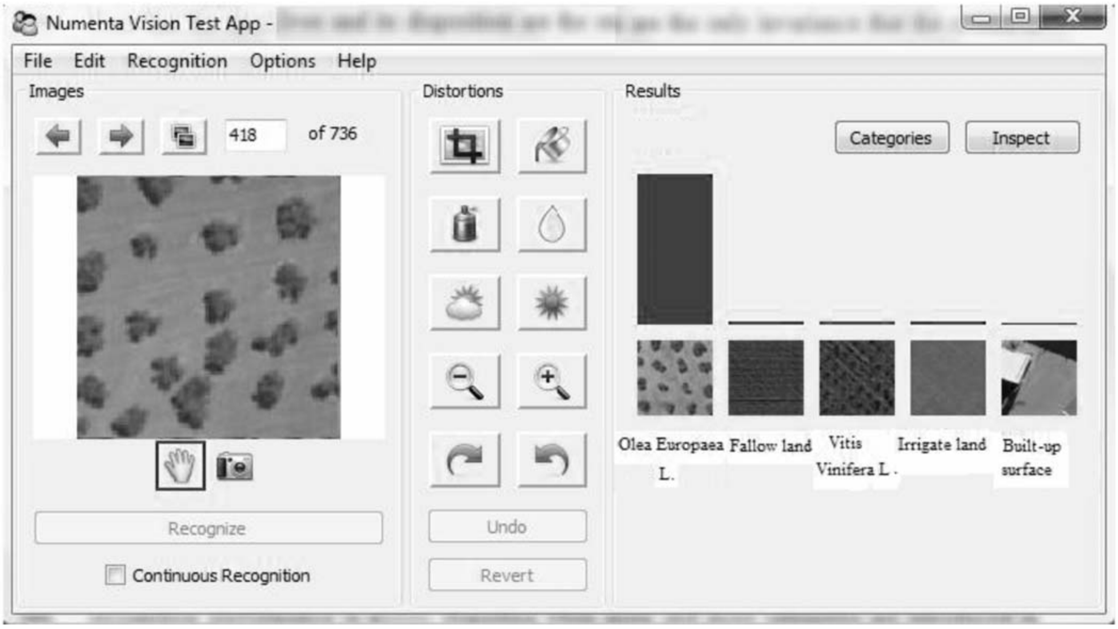


Fig. 6. Rotated image of the category '*Olea europaea* L.'.

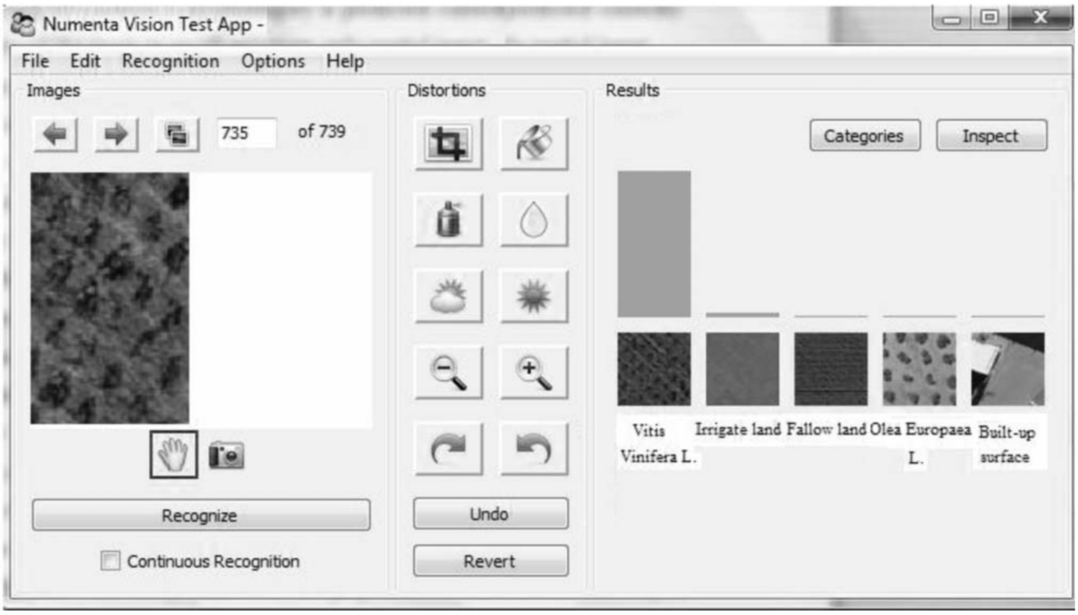


Fig. 7. Partial image of the category '*Olea europaea* L.'.

missing information was reconstructed and the category was correctly predicted. This resembles a capability of the brain to recall missing information given only partial input. The system can also tolerate a substantial amount of noise of various types and still discern and correctly recognise the category, as shown in Fig. 8.

We observed that the system performed better overall while recognising complex images, which have more discernible features, such as corners and line intersections. For example, images of 'fallow land' and 'irrigated land' were not recognised as easily as 'built-up surface' or '*Vitis vinifera* L.'. The system also sometimes tended to confuse categories sharing many similar shapes, such as '*Vitis vinifera* L.' and '*Olea europaea* L.'. We also observed that the recognition performance was slowly degraded when more categories were introduced in training, arising from the same confusion between similar images. For example, we tried to classify different irrigated crops (*Phaseolus vulgaris* L., *Triticum aestivum* L., *Vicia faba* L. and *Pisum sativum* L.), but the analysis and result showed a small overall accuracy of 69.65% and a Kappa Statistic of 0.72.

It is also useful to note the relative strength of beliefs of the ten best-predicted categories that is displayed by the system as a bar graph. When input image is not heavily distorted, and resembles its true category more than any other categories, we see a graph similar to the one shown in Fig. 9. We can judge from the graph that the winning prediction is very confident. When the input image is not readily recognisable, or seems similar to several categories, the graph would look like the one in Fig. 10.

Conclusions

The images from the digital aerial sensors in our model may be an extremely useful tool in the agriculture field, providing an accurate result for the use of land in a fixed area under certain conditions. By contrast, traditional classification techniques, basically pixel-based approaches, are limited in that they typically produce a characteristic 'salt and pepper' effect, and are unable to extract objects of interest. An HTM network considers spatial and temporal relations between features of the sensory

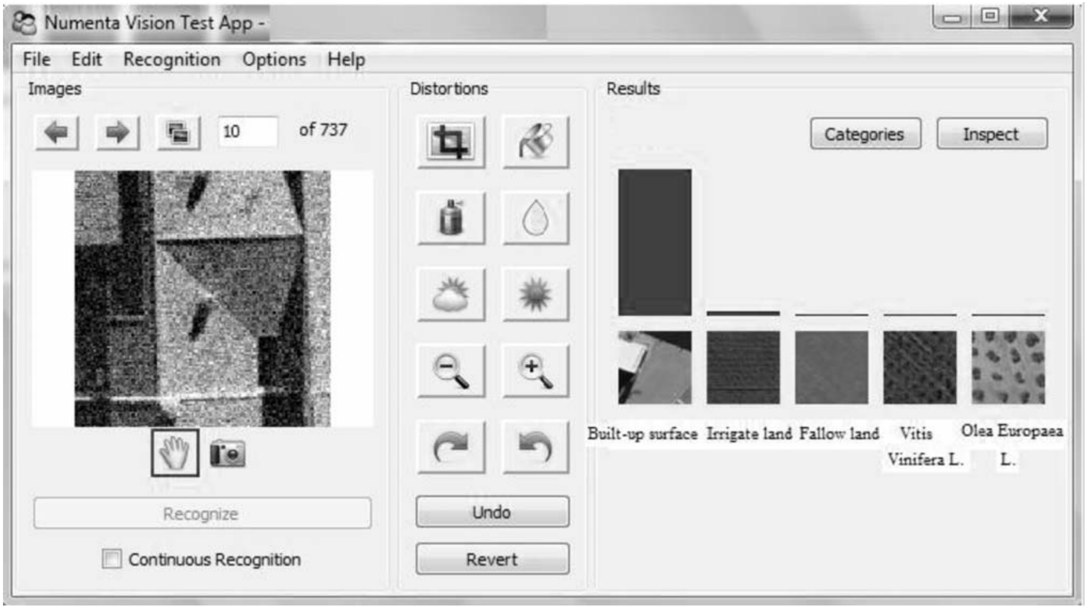
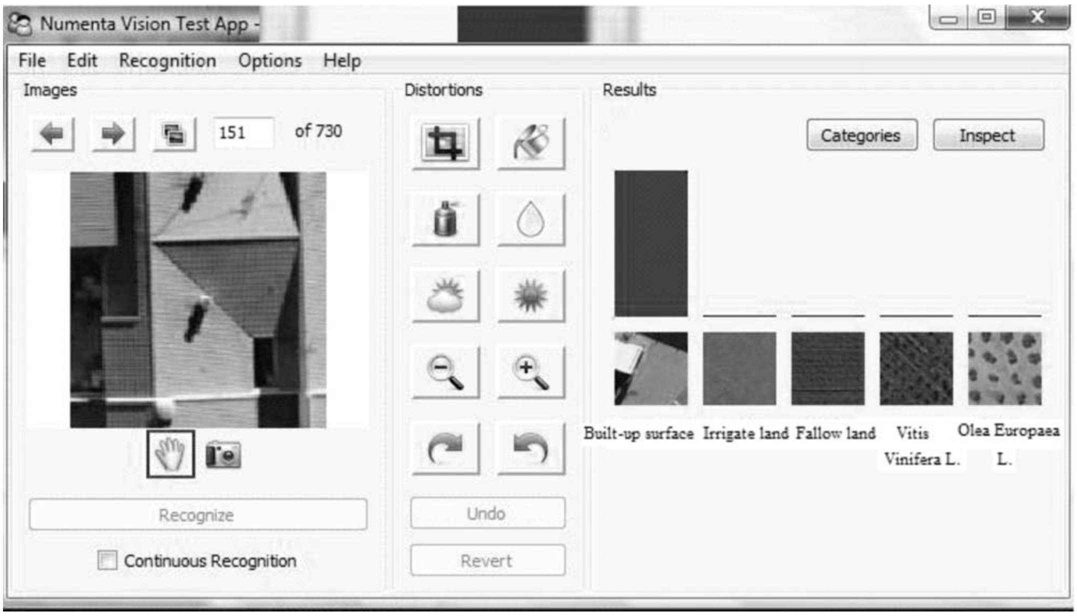


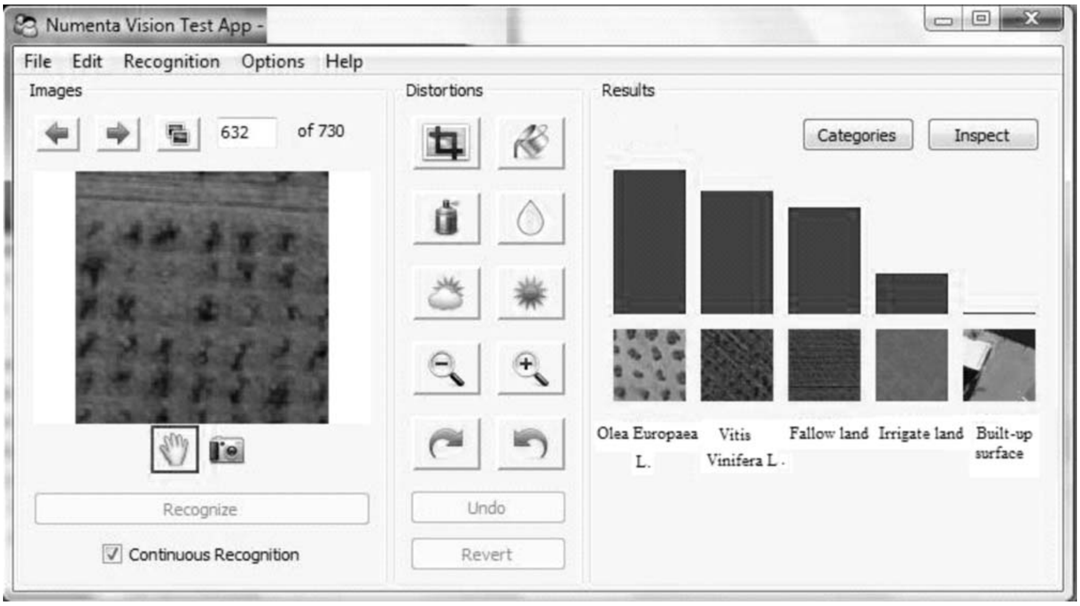
Fig. 8. Image of the category 'built-up surface' with noise.



151 of 730

□ Continuous Recognition

Fig. 9. Graph showing 100% confidence in category prediction (built-up surface).



632 of 730

☑ Continuous Recognition

Fig. 10. Graph showing lack of confidence in single-category prediction.

signals which are formed in an hierarchical memory architecture during a learning process. The methods are not, however, actually comparable because HTM classifies only small greyscale images with only one pattern. In the future, we expect that the platform will be able to classify more than one pattern, and the 'salt and pepper' effect could be eliminated.

This model shares many common ideas with traditional neural networks. The hierarchy consists of many relatively simple units (subregions) that do the same basic operation and can be made to run in parallel. It solves problems by using cooperation between subregions without a centralised algorithm. The knowledge and beliefs in the system are distributed between the subregions in various hierarchical levels. The system learns its skills by training and is able to generalise. The memory-prediction framework, however, is an inferential system that uses beliefs for learning and recognition. Nevertheless, due to their similarities, the model shares a number of advantages with neural networks; it clearly can function as an associative memory, can tolerate noise and can generalise training images. Finally, this model offers a greater promise of understanding what intelligence is by closely modelling the overall structure of the human neocortex.

Received 11 May. Accepted 27 August 2009.

1. Douglas R.J. and Martin K.A.C. (2004). Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.* **27**, 419–451.
2. Hawkins J. and Blakeslee S. (2004). *On Intelligence,* chap. 4, pp. 83–104. Henry Holt & Company, New York.
3. Hawkins J. and George D. (2007). *Hierarchical Temporal Memory: Concepts, Theory, and Terminology.* Numenta Inc., Redwood City, California. Online at: http://www.numenta.com/Numenta_HTM_Concepts.pdf
4. George D. and Jaros B. (2007). *The HTM Learning Algorithms.* Numenta Inc., Redwood City, California. Online at: http://www.numenta.com/for-developers/education/Numenta_HTM_Learning_Algos.pdf
5. Csapó A. and Baranyi P. (2007). VFA-driven Hierarchical Temporal Memory input for object categorization. In *Proceedings of the 8th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics,* Hungary. Online at: http://bmf.hu/conferences/cinti2007/3_CsapoAdam.pdf
6. Lillesand T.M. and Kiefer R.W. (1994). *Remote Sensing and Image Interpretation,* 3rd edn, pp. 616–617. John Wiley & Sons, New York.
7. Chuvieco E. (1990). *Fundamentos de teledetección espacial.* Ediciones Rialp, Madrid.