Introducing an Intelligent Computerized Tool to Detect and Predict Urban Growth Pattern

SITI Z. Z. ABIDIN[1], M. N. FIKRI JAMALUDDIN[1], M.ZAMANI Z. ABIDEN[2]
[1]Faculty of Computer and Mathematical Sciences,
[2]Faculty of Architecture, Planning and Surveying
University Technology MARA,
40450, Shah Alam, Selangor
MALAYSIA
sitizaleha533@salam.uitm.edu.my, {breakprimetime, mzabiden009}@gmail.com

Abstract: - Urban growth pattern is usually detected using spatial analysis. Spatial analysis is widely used in scientific research especially in the field of statistics, image processing and geoinformatics. In modeling urban growth, the analysis is mostly performed using statistical and mathematical techniques. With the advance computer technology, physical land (ground) situation for a place of interest can be represented in digital computerized form with the accurate and appropriate scale. In this way, measurement can be made on the digitized presentation for performing analysis. The change in land use is affected by many factors such as population growth, economic change, social structure, the change in rules and regulation, and many more. These influential factors have dynamic behaviors that require complex solutions. Much research has been undertaken to use several methods such as geographical information system (GIS) and cellular automata theory, to model the urban growth. Recently, an intelligent approach has been introduced that features dynamic behavior. Artificial Neural Network (ANN) has the capability to learn dynamic behavior and performs prediction based on its learning process.

In this paper, we present an intelligent computerized tool, called DIGMAP-Detector. This tool is able to learn a pattern of urban growth based on at least two digital maps (with 4-bit/pixel bitmaps or 8-bit/pixel bitmap in Bitmap File Format (BMP)). Implemented using Java programming language, the tool reads digital map files with the size of 847 pixels length and 474 pixels width. Classification on the map with two independent binary classes (value *1* for urban and *0* for rural) are prepared using GIS software. By applying a cellular automata theory that considers the affect on a center pixel is influenced by its surrounding pixels (eight pixels), the tool uses a back propagation neural network to read the values of surrounding pixels as its input layer nodes and the center pixel as the output node. Several analyses are performed to determine the appropriate values for the neural network configuration before its learning engine starts to learn the pattern of dynamic urban changes based on the digital map patterns. When the neural network engine has learnt the pattern, prediction can be carried out to predict the missing years and future urban growth. With good prediction accuracy, urban planning and monitoring can be performed with maintaining good ecological and environmental system. In addition, better planning also gives benefit to economic values.

Key-Words: - Computerized tool, Artificial Neural Network, Digital map, Urban Growth, Urban pattern, Cellular automata.

1 Introduction

Urban growth pattern is usually detected using spatial analysis. Spatial analysis is widely used in scientific research especially in the field of statistics [1], image processing [2] and geoinformatics [3-4]. The analysis is mostly performed using statistical and mathematical techniques [5]. Although much research has been undertaken to model urban growth using such analysis, the need to find a standard and universal solution to model the urban growth seems hard to achieve because there are many influential factors that contribute to the pattern of land use change. So far, the proposed models are only applicable to the studied areas because no single place can have similar influential factors. Mostly, the factors are caused by several issues such as population, social, environmental, economics and politics. Thus, the key challenge in modeling urban growth is that it requires dynamic features.

Cellular automata model [6] is found to be very successful in generating complex structures. Recently, an intelligent approach has been introduced to model the urban growth. Artificial Neural Network (ANN) [7] has the capability to learn dynamic behavior and performs prediction based on its learning process. The emergence of artificial neural network (ANN) as one of computational intelligence, has added more processing power when ANN is used within the spatial analysis.

The analysis is performed on digital maps. In some studies, topological maps are used and they are converted into digital forms [8]. The urban growth pattern can be studied by comparing the past history of a place with its current situation. The study usually focuses on each of the influential factor by collecting more information when checking the changes of the land used.

In this research work, different approach has been proposed by detecting the urban growth through the change in pixel values at specific points (or region) in digital maps. Then, the key factors that influence the land use change can be determined by identifying the pattern of changes in pixels for at least two digital maps. The digital images are acquired through satellite. This is, perhaps, the first attempt, in the field of urban planning, to provide a new tool to detect and predict urban growth pattern with flexible points (or region) of interest, through an abstract approach. In order to know the real scenario, further investigation need to be carried out at the specific location but generally, with the detection and prediction initiated by this tool, some information about the urban growth pattern is already known to the users.

As a case study, six satellite images of the same location in Klang Valley (one of the most rapid changes of urban growth in Malaysia) are used. The maps are initially processed using residual error approach [9] that concerned with getting the most accurate geo-coding and classification (when to compare the digital image and the ground measurement). All the digital images are classified into two independent classes; built and non-built area (urban and rural, respectively) using ENVI software. The built area is represented by value *1* and value *0* is for non-built.

Then, the input data is reprocessed for several steps before it is trained by a feed forward back propagation neural network learning engine. Several analyses are performed to get the best configuration of the artificial neural network so that prediction can be carried out at the best accuracy. The more accurate the prediction, the better it can be used to plan and monitor future urban growth. Thus, economical development and ecological system as well as the environment will be managed properly.

This paper is organized as follows: Section 2 discusses on the related work of the intelligent tools and urban growth pattern. Section 3 presents the design and implementation of the computerized tool. Section 4 illustrates the results, before the concluding remarks in Section 5.

2 Related Work

Much research has been undertaken to use intelligent techniques for pattern recognition and prediction. The artificial neural network (ANN) that was inspired by the biological neurons in human brain [10] has the ability to learn input pattern by using a transfer function. There are several types of ANN, however, in this research work, other types of ANN such as recurrent and self-organizing map (SOM) do not fit to the requirement of this research that deals with known input data and output values (exactly eight input nodes to produce one output at a time). In recurrent neural network, when input produces the output, the output is used back as input and the process is iterated. In contrary, in this research, when the output is produced, the pattern is learnt and none of the output will be used for further processing. In SOM, the learning process is performed using unsupervised approach and eventually, the output values are not known in advance and they are determined by the SOM engine. Therefore, it is clearly shown that SOM is not suitable to be selected as the intelligent engine.

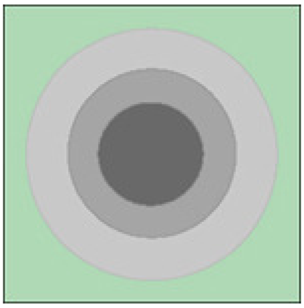
2.1 Application of Intelligent Techniques

Intelligent techniques are widely used in various disciplines such as medical [11-12], geosciences [13-14], economics [15-16], weather forecasting [17], mechanics [18], environmental issues [19-20] and many more. Its powerful capabilities have been proven in these research works, even though the problems are complex and multidisciplinary. For example, in the medical field [21], most ANN algorithms are used to detect any uncertainties in medical imaging for a certain diagnosis. Besides dealing with medical imaging, the intelligent technique can be used in expert systems for determining certain diseases. In geosciences, SOM can be used to predict the distribution of minerals at a particular place, while ANN is used to predict the potential of mineral existence with respect to other given minerals through the concept of mineralization process [13]. In economics, ANN can be used in cost estimation that helps in profit making for mass-production products. For weather forecasting, ANN can predict weather condition for a particular time period. It is also used to do a prediction model for lubrication oil consumption in mechanical field. Furthermore, the intelligent technique is also useful in decision support and modeling of rain harvesting system for the benefits of the environment.

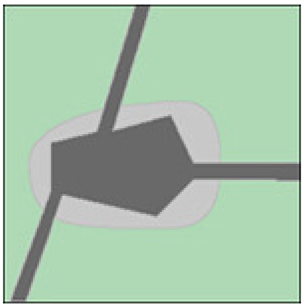
Most of these research works use intelligent techniques by applying their concepts or algorithms to build new models or doing simulation. The common tools to implement the simulation are using the existing software tools such as MATHLAB. It is hard to find intelligent tools that are implemented using high level programming languages, especially in the urban growth pattern and prediction.

2.2 Urban growth pattern

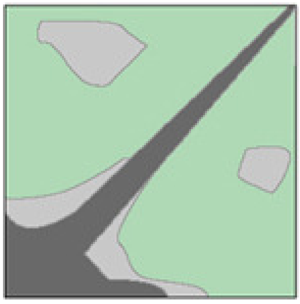
The urban growth pattern is usually influenced by many factors. The factors include population density, economic status, political issues and geographical condition [3,22,23]. The uncontrolled rapid urbanization process may expand urban areas and becoming a sprawl. Figure 1 shows the potential types of urban sprawl; low-density (radial), ribbon and leapfrog sprawl [24].



(a) low density



(b) ribbon



(c) leapfrog

Figure 1. Types of urban sprawl.[24]

In Figure 1 (a), the center and dark colored area is the original urban area. The lighter colors are the expansion area with regards to the original urban. The low density sprawl type is usually affected by public infrastructure, while the ribbon sprawl type happens with the influent of major transportation network. The leapfrog type develops several isolated and detached areas that require high cost of development in order to provide basic resources in everyday life (e.g. water, electricity, sewage, etc.) [24]. Therefore, urban planning and monitoring are crucial to ensure proper urbanization process.

Urban growth models were started to be developed from static models to comparative equilibrium (allows temporal data). Some of current models are able to deal with urban dynamic features. Cellular automata model is one of the most popular techniques used in modeling urban growth pattern [25-26]. In the model, there are cells, states, neighborhood and transition rules. It can generate complex structures and can be used to model dynamic and evolutional issues [27-28]. For example, Rongqun *et al.* [29] use cellular automata and geographical information system (GIS) to simulate wetlands evolution of Yinchuan City and Samat [30] simulates the urban growth for the area of Seberang Prai. The combination of cellular automata and neural network is becoming the interests of several researchers [31-32]. However, the emphasis was focused on the cellular automata model rather than the configuration of the neural network engine.

3 Design and Implementation

In this section, the design and implementation of the proposed intelligent tool are discussed. The classified temporal imagery data was processed using the Geographical Information System (GIS). All the imagery data (in the form of digital maps) was processed so that it is in bitmap (BMP) format and only contains binary values. Figure 2 shows the presentation of digital map with the size of 847 pixel width and 474 pixel height. The map is classified into two independent classes, built (white area) or binary value *1*, and black (black area) or binary value *0*.

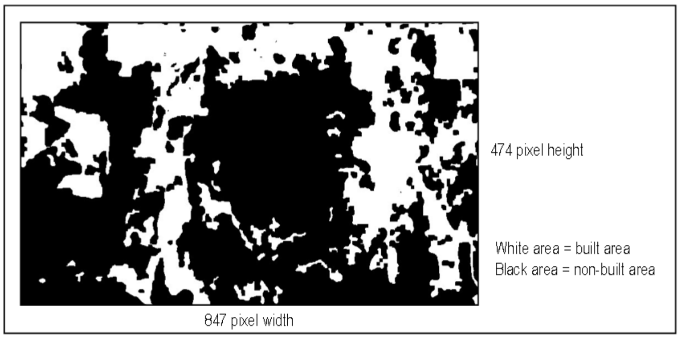


Figure 2. Example of a digital map for the case study area

In designing the computerized tool, namely DIGMAP-Detector, the imagery data is reprocessed to extract relevant pixels for the training process. It takes a lot of steps to prepare the data for the learning engine. As for the minimum (two datasets) of two different images, the minimum pixels to be processed are as follows:

$$\textit{Total pixels} = w \times l \times t \text{ , where} \quad (1)$$
$$(w\text{:width; } 15<w<847),$$
$$(l\text{:length; } 15<w<474),$$
$$(t\text{: total datasets; } t>2)$$

In a digital map, not all pixels are processed because in urban growth pattern, there are several influential factors involved that contribute to the change in the land used. Thus, the tool is designed so that users can choose one or more pixels (or area) of interest. One pixel represents an area of 30 meters by 30 meters for the ground. The framework of the prototype system is shown in Figure 3.

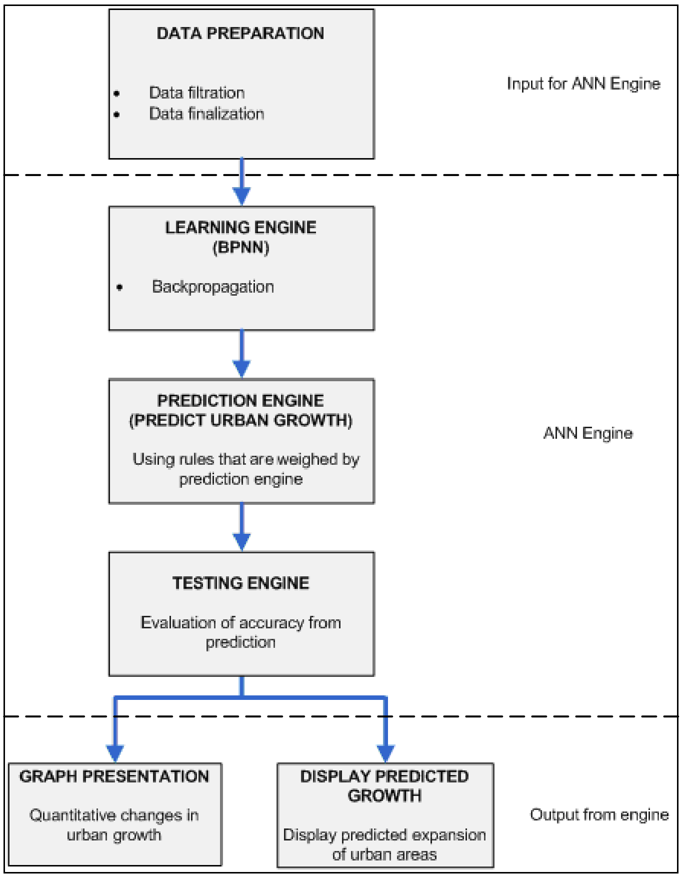


Figure 3. The framework

As illustrated in the figure, there are three major phases involved; data preparation, engine development and output setting. In preparing input, data has to be filtered before it is finalized. The intelligent engine will learn the data pattern, performs prediction based on the weight resulted from the data training process, and then it will test the accuracy of results. For output setting, there are two presentations of the result; first, a digital map presentation with growth indicator, and secondly, a line graph to illustrate the growth pattern based on years. Details on preparation of input and configuration of intelligent engine are described in the following section.

3.1 Preparation of Input

A study is performed by giving a special attention to the location of pixels in the digital maps. The growth of urban is usually becoming a sprawl [24] and it is likely to have radial sprawl in most urban expansion. Therefore, it is suitable to use the cellular automata method by focusing on how a pixel (value *0*) is influenced by its surrounding pixels' values. Therefore, for the first step, the DIGMAP-Detector reads all the pixels in the given maps to search for pixels with value *0* (urban area) and its surrounding pixels that have both *1* and *0* values. Figure 4 illustrates presentation of grid cells (representation of a pixel and its surrounding locations) and the selection of a location to be viewed in an ordinary text format.

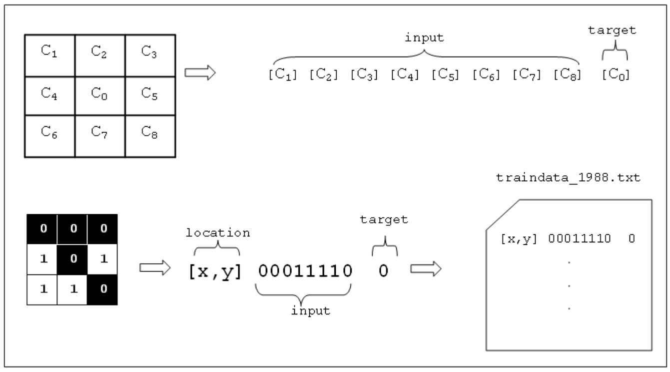


Figure 4. Steps in input representation

After the order of location (C_1 to C_8) for the surrounding pixels are determined, the appropriate pixel values are extracted from the first image file. When extracting the values, this tool writes the respective coordinate and the values in a line into a file of text format. After all the coordinates are examined, the content of the text file will be checked to make sure that all of the surrounding pixels that have the combination of *1*s and *0*s are used. In this way, only the area with the possible (land use) change will be processed for pattern detection.

There are four conditions to be avoided in preparing the input data. They are as follows:

- Condition 1: All the surrounding pixels' values are all *1*s. They have no significance because the surrounding is urban (built land) and no pattern can be learnt.
- Condition 2: All the surrounding pixels' values are all *0*s. They have no significance because the surrounding is rural (non-built land) and no pattern can be learnt.
- Condition 3: The center cell has changed from 1 to 0 with the meaning that the built land is transformed into non-built. This situation is very rare in real life so the change might be due to errors during initial image processing.
- Condition 4: Repetition of pattern for all the surrounding pixels' values that contain combination of *1*s and *0*s.

The last condition (Condition 4) is very important in determining the pattern to be learnt by the neural network engine. Since it is possible to have the repeated pattern of binary values, DIGMAP-Detector has to process the data once again to eliminate the unnecessary duplicate pattern. Figure 5 illustrates the process of getting unique train data after eliminating any duplicate values.

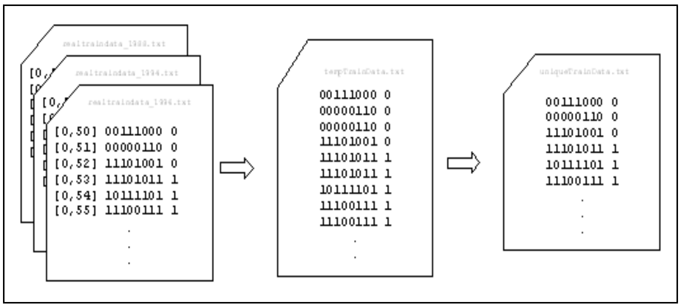


Figure 5. Steps in obtaining unique train data

The pattern must be unique so that the learning process will not be influenced by the same input pattern that leads to wrong results. At the same time, the coordinate values are no longer relevant and significant. Therefore, the final data contains only unique sequence of binary values.

3.2 Configuration of the Intelligent Engine

When the input data for DIGMAP-Detector is ready to be processed, it is important to make sure that the best network configuration is achieved. The number of input nodes is exactly eight to present the number of surrounding pixels. It is also known that the number of output node is one for reflecting the center pixel.

Several analyses are carried out to determine the number of hidden nodes, suitable learning rate, value for bias, and number of epoch. For the value of bias, they are set differently during learning and prediction due to the constraint of using the binary values that only have either *0* or *1*. By using the same value as learning, all pixel values tend to be *1*. Therefore, the sigmoid function [33] has to be applied to adjust the bias value during prediction. The function for creating the bias is as follows:

$$fAN(net-\theta)=\frac{1}{1+e^{-\lambda(net-\theta)}} \quad (2)$$

In equation (2), λ controls the steepness of the function where the value is usually λ= 1. Figure 6 shows the sigmoid function when the bias, θ = 0. During the development, a suitable bias value needs to be defined in order to balance the prediction result.

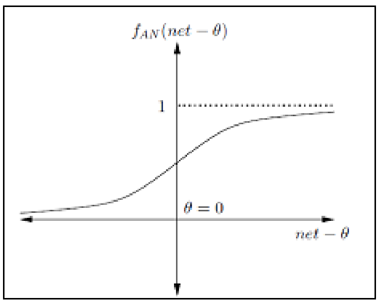


Figure 6. Sigmoid transfer function [33]

During the analyses, the number of hidden node is set to 6, learning rate is 0.01, bias (during learning) is 0.6, bias (during prediction) is 0.9 and epoch is 20. When the analysis is performed on the number of hidden nodes, its value becomes a variable and all other parameters are fixed to the previously assigned values. The accuracy is determined based on five out of six digital maps; year 1994, 1996, 2000, 2001 and 2003. Figure 7 shows the example of analysis on the number of hidden nodes.

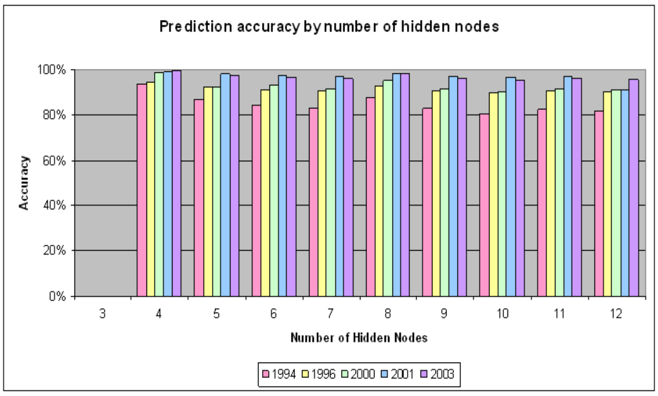


Figure 7. Analysis on finding the number of hidden nodes

From the analysis, the most accurate result based on the average of five datasets, is four hidden nodes with above 80%. However, four nodes cannot be used to present the hidden nodes since the input nodes are eight. In this case, the next best average accuracy is eight and it is appropriate to be used.

For analyzing the learning rate, similar concept is applied by keeping all the parameters to static values except the learning rate (as a variable). Figure 8 depicts the result of the analysis, where the value of 0.01 for the learning rate has accuracy more than 80% for all the given years. Other learning rate values cannot perform better and therefore, learning rate of 0.01 is selected as the best.

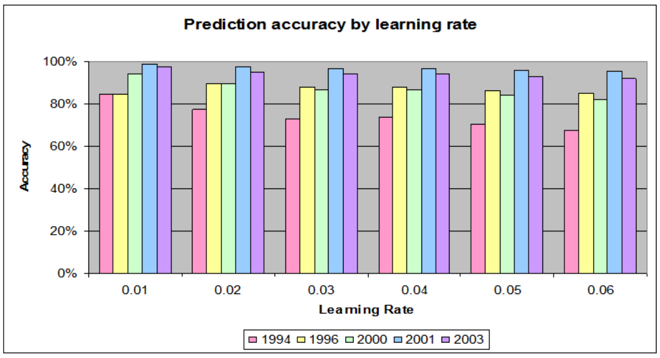


Figure 8. Analysis on finding the best value for learning rate

In finding the best value of bias, the bias for prediction is set as variable while other parameters remain static. Figure 9 illustrates the results of bias.

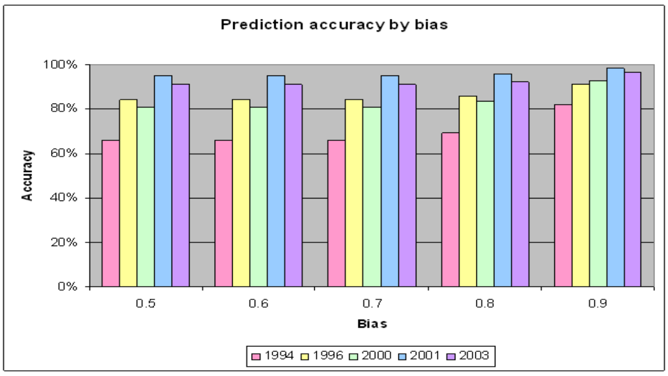


Figure 9. Analysis on finding the best value for bias

Referring to Figure 9, the increase in bias value gives better accuracy percentage for the year 1994 data. However, the percentages of accuracy for other years do not change much over the change in the value of bias. On average, the best value for bias is 0.09 (during prediction).

Analysis on configuring the best value of epoch is also performed by testing on the value of 20, 40, 60, 80 and 100. Figure 10 shows the results on testing the epoch.

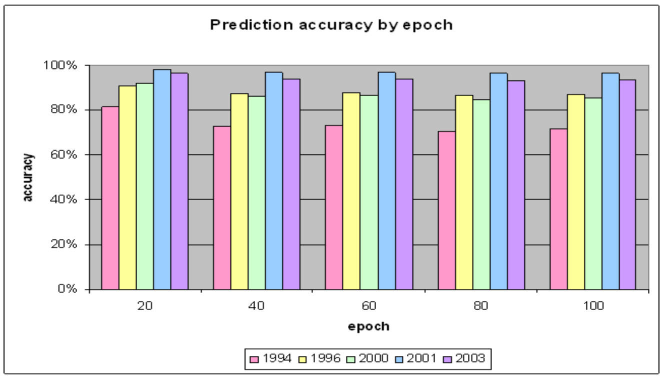


Figure 10. Analysis on finding the best epoch

In the analysis, epoch 20 gives average accuracy of 80% or more and can be considered the best, while other values give almost similar results. The value of epoch determines the number of iteration in learning the pattern of changes in the unique pixel values. If the value is too large, it will cause the neural network engine to remember the pattern that will never be adjusted. If this case happens, it will reduce the performance of the prediction engine. Therefore, the increase in number of epoch does not necessarily give better results.

When the best neural network configuration is known, the computerized intelligent tool is designed and implemented. Figure 11 shows the neural network configuration with eight nodes in the input layer, eight nodes in the hidden layer and one output node.

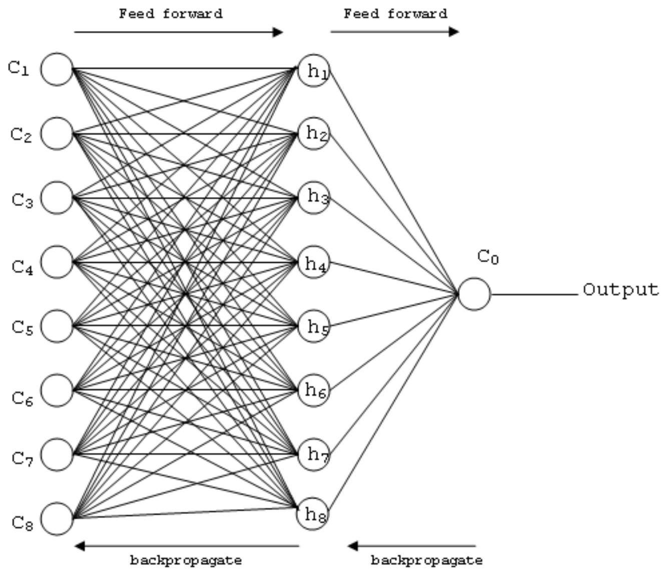


Figure 11. The neural network configuration

The transfer function algorithm of back propagation neural network is used to simulate the urban growth pattern. The tool is implemented using Java programming language. Java is an object oriented programming (OOP) that uses classes. Figure 12 shows the class diagram of the tool. There are three main components that reflect the three major phases in the framework. They are `data`, `bpnnEngine` and `result`. The `data` class is used to acquire the input in the form of image and performs the preparation of input to ANN using `preprocessing` class. The `bpnnEngine` class contains the learning and prediction engine that uses the back propagation neural network algorithm. The `result` class produces the results using wizard so that the tool provides users with user friendly environment.

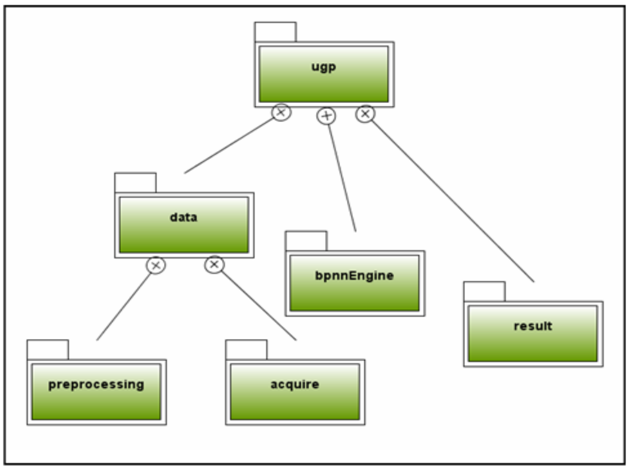


Figure 12. Class diagram

This tool is capable to detect pattern of any changes for pixels in at least two digital maps. It can be applied to any location or any digital map as long as the format of input data is as specified.

4 Results

The interface of DIGMAP-Detector tool composes of a tool bar and multi-dimensional interface (MDI). With the simple layout, users can work conveniently by clicking on the selected icon and has the MDI to show the available output for the learning and prediction processes. Figure 13 illustrates the layout of the interface.

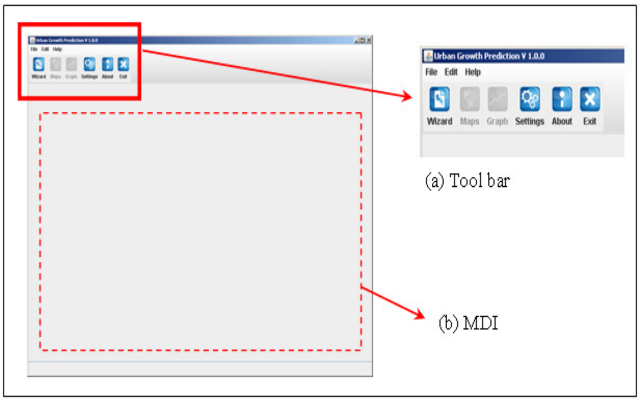


Figure 13. Layout of DIGMAP-Detector's Interface

A wizard is provided that guides users to use the computerized tool. There are three main steps in the computerized tool; process the digital maps, learn the maps' patterns and do the prediction.

- Step 1: Process the digital maps.
  Users are required to enter at least two file names with the correct file path so that the tool can search for files at the correct location of the file directory. Figure 14 shows the screenshot of interface for setting the input data that requires information on the year and filenames.

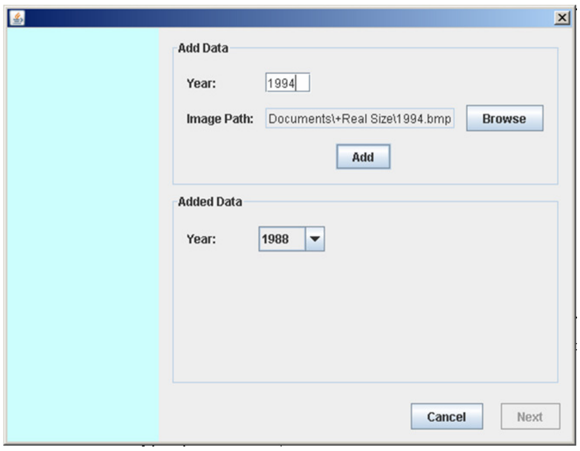


Figure 14. Screenshot of setting input files.

It is possible to enter more than two files for better prediction. Details on processing the input data from the image files are hidden from the users. Users are provided with a button 'NEXT' to proceed with the learning process.

- Step 2: Learning process.
  Users can see a progress bar to indicate that the percentage of progression in the learning. Since the learning process takes quite some time, the progress bar helps user to estimate the length of time to finish. At the same time, user can see other information such as increasing values for epoch and information on errors. Figure 15 shows the screenshot of the interface.

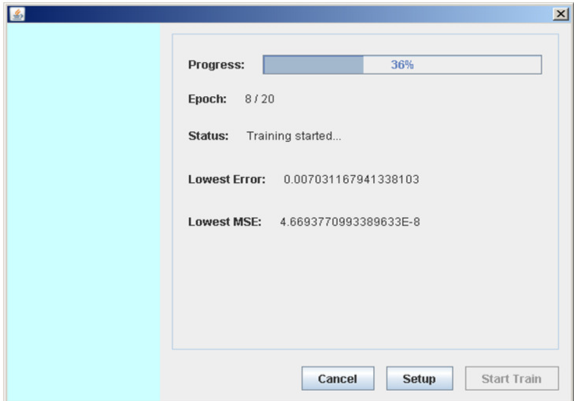


Figure 15. Screenshot of the learning process.

By providing users with the dynamic changed values during the training, users are able to see the learning process. It is also significant to provide users with another interface for manually change neural network configuration engine as discussed in Section 3.2. Figure 16 depicts the screenshot of the learning engine configuration settings.

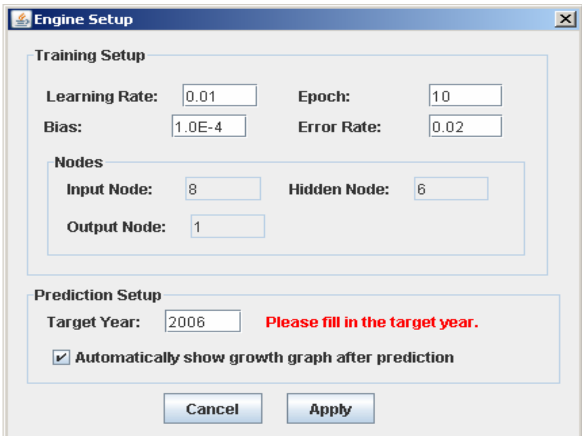


Figure 16. Screenshot of engine configuration setup.

In the dynamic setup, users are able to change values such as learning rate, bias, epoch and the number of hidden nodes. However, some values are fixed such as the number of input and output nodes. For the prediction set up, users are free to choose any year starting from the year 1989 since the data starts at year 1988. On top of that, users have the choice to see the prediction of the growth in term of graph besides producing the predicted digital map. Figure 17 depicts the graph that shows the growth of urban staring at year 1989 (starting year for enable prediction) until the year given by the user as the target.

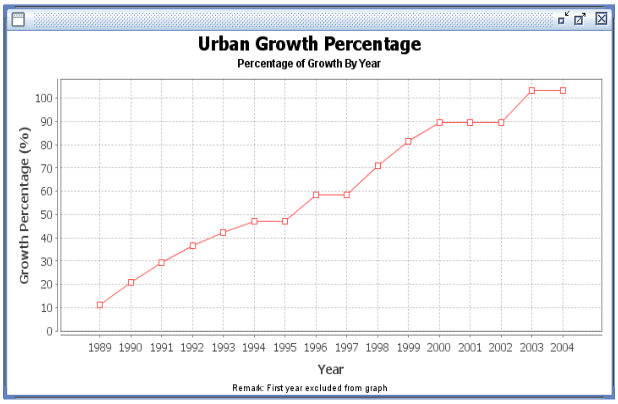


Figure 17. Growth percentages in graph presentation.

The graph shows the increase in percentage of growth.

- Step 3: Prediction process

After doing the prediction set up by setting the start year of prediction, users can see the information pertaining to the process is displayed while a progress bar is shown. The use of progress bars throughout the system is performed when the computerized engine needs some time to process users' requests. In this way, users can utilize their time instead of waiting without doing anything.

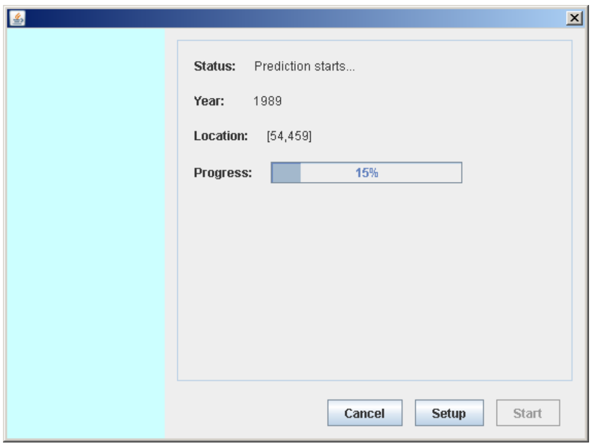


Figure 18. Screenshot of the prediction process.

Figure 19 shows the screenshot of the prediction results (in term of digital map) performed by the tool. It is a prediction pattern for the year 1990.

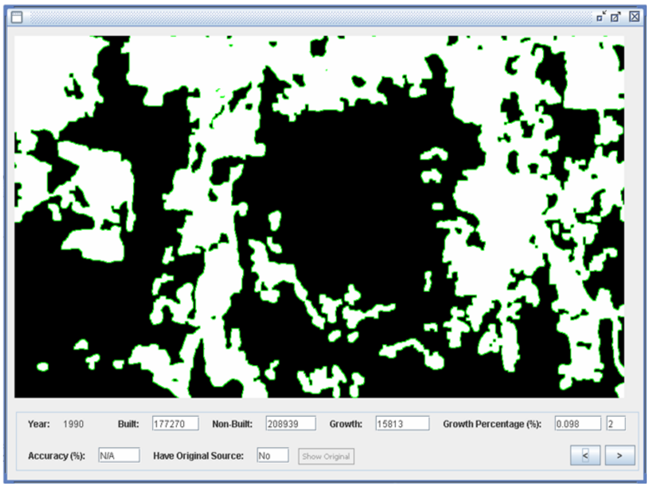
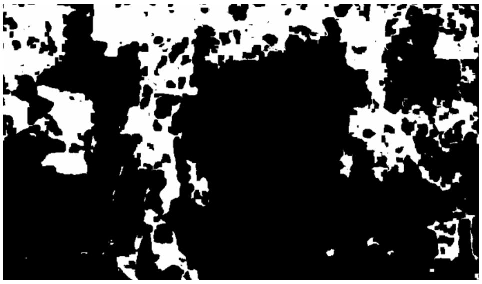


Figure 19. Predicted pattern for the year 1990 (missing year).

In the screenshot, at the bottom of the predicted digital map, there is some information about the predicted year, the total number of pixels with the value *1s* and *0s*, the number of 'growth' pixels, and the growth percentage. For this particular example, accuracy of the prediction cannot be determined because there is no original image map to be compared. If there is the original map in the given input file, the accuracy value will be given to the user together with an active button to view the original image. With the capability of the multi-dimensional interface (MDI), it is easy to view the predicted and original images alternately by pressing the button 'Show Original'. The accuracy is determined by the difference in the total pixels' values. The tool will check and calculate the pixels with different values when going through the original image.

With six temporal imagery data and focusing on fifteen years duration (1988 to 2004), this tool gives 80% to 93% accuracy in predicting the urban growth pattern. Figure 20 depicts the comparison between original and predicted images for the year 2000. The prediction image seems to have more growth as compared to the original image.


(a) Original image

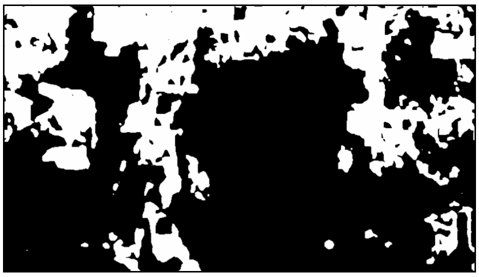

(b) Predicted image

Figure 20. Original and predicted images for the year 2000.

For the 15 years duration and only six datasets are available, it is the need to fill the gap between the years. The shorter the length of period between two datasets, the more percentage of accuracy can be produced. It is significant to analyze the differences because the ability of the tool to predict urban growth is depended on the percentage of accuracy. Table 1 illustrates the results.

Table 1. Comparison between period and percentage accuracy

Year	Period (years)	Accuracy (%)
1994	6	71.934
1996	2	86.488
2000	4	84.743
2001	1	95.210
2003	2	92.047

There are two results for the period of 2 years that give two distinct values. The former (year 1996) gives less accurate results, may be due to not much information from its previous years.

5 Conclusion

In this paper, we have presented a computerized tool called DIGMAP-Detector that uses back propagation artificial neural network technique to detect the pattern of changes (particularly urban growth) in at least two digital maps. The tool is implemented using Java programming language that reads input as image in 4-bit/pixel or 8-bit/pixel bitmap file (BMP). There are at least two temporal satellite images of the same location, need to be provided. The tool will do processing on the digital maps and produces a data set in ordinary text format that contains pixel's coordinate, eight surrounding pixels' values and the next available center pixel's value. The values are in binary forms (*1* is for built and *0* for non-built). The tool has the learning engine that is configured through four accuracy analyses; number of hidden nodes, the learning rate, bias and epoch. The results show that the tool successfully performs prediction on the pattern of urban growth with 80% to 93% accuracy. The accurate prediction helps in urban planning and monitoring that could facilitate future livings.

References:

[1] Lee, B.N.; Wang, L.-C. And Abadir, M.S., Refined statistical static timing analysis through learning spatial delay correlations, *43rd ACM/IEEE Design Automation Conference*, 2006, pp. 149-154.

[2] Ziliani, F. and Cavallaro, A., Image analysis for video surveillance based on spatial regularization of a statistical model-based change detection, *Conference on Image Analysis and Processing*, 1999, pp. 1108–1111.

[3] Yongfu Li; Anrong Dang and Tianhang Huang, Evolution of administrative division in Beijing and analysis on its spatial pattern based on GIS, *17th International Conference on Geoinformatics*, 2009, pp. 1-6.

[4] Zheng Ji-dong and Luo Li-ping, The Establishment and Application of Urban Environment Information System Based on MAPGIS: A Case Study on Jiaozuo City, *International Conference on Environmental Science and Information Application Technology*, *ESIAT 2009*,Vol. 3, 2009 , pp. 191-194.

[5] Zhixiang Yang, Xiufeng He and Zhiquan Hu, Evaluating Urban Expansion of Nanjing City Based on Remote Sensing and GIS, *International Conference on Environmental Science and Information Application Technology, ESIAT 2009*, Vol. 3, 2009 , pp. 432–435.

[6] Itami, Robert M., Simulating Spatial Dynamics: Cellular Automata Theory, *Landscape and Urban Planning (30),* 1994, pp. 27-47.

[7] Rosenblatt, F., *The Perception: A Probabilistic Model for Information Storage and Organization in the Brain*, Psychological Review, vol. 65(6), 1958

[8] Fadda, Eyad H.R, Kakesh, Maher and Al Azb, Tariq A., The implementation of satellite images and associated digital image processing in addition to GIS modeling for urban mapping in Amman area, Jordan, *WSEAS Transactions on Communications*, 2009, pp. 300-309.

[9] Zainal Abiden, M.Zamani and Abidin, Siti Z. Z., Introducing Residual Errors in Accuracy Assessment for Remotely Sensed Change Detection.

IEEE 5th *CSPA* 2009.

[10]Yu, Z., Feed-Forward Neural Networks and Their Applications in Forecasting, *http://free.beyond dream.com/neuralnet*, 2000.

[11]Burke, H.B., Rosen, D.B., and Goodman, P.H., Comparing artificial neural networks to other statistical methods for medical outcome prediction, *IEEE International Conference on Computational Intelligence*, Vol. 4 , 1994 , pp. 2213–2216.

[12]Abdul Khalid, N.E, Manaf, M. and Aziz, M.E., Fusion of Fuzzy Heuristic and Particle Swarm Optimization as an Edge Detector, *IEEE International Conference on Information Retrieval and Knowledge Management*, *CAMP10*, 2010, pp. 249-253.

[13]Abidin, S.Z.Z, Musa, N., Sulaiman, S. and Abiden, M.Z.Z., Mineral Potential Prediction using Hybrid Intelligent Approach, *IEEE International Conference on Environmental and Computer Science ICECS 2009*, Dubai, 2009, pp. 384-388.

[14]Skabar, A., Mineral potential mapping using feed-forward neural networks, *Proceedings of the International Joint Conference on Neural Network*, Vol. 3, 20-24 July 2003, pp. 1814–1819.

[15]Khoa, N.L.D., Sakakibara, K. and Nishikawa, I., Stock Price Forecasting using Back Propagation Neural Networks with Time and Profit Based Adjusted Weight Factors, *SICE-ICASE 2006*, 2006, pp. 5484 – 5488.

[16]Wang, H.S., Che, Z.H. and Wang, Y.N., Cost estimation of plastic injection products through back-propagation network, *the 8th WSEAS International Conference on Neural Network*, Canada, 2007.

[17]Taylor, J. W. and Buizza, R., Neural Network Load Forecasting with Weather Ensemble Predictions, *IEEE Power Engineering Review,* Vol. 22 , Issue 7, 2002 , pp. 59-59.

[18]Hsiao, Y.F. and Tarng, Y.S., Study of the lubrication oil consumption prediction of linear motion guide through the Grey Theory and the neural network, *WSEAS Transactions on Applied and Theoretical Mechanics*, 2009.

[19]Musa, N., Salleh, S. and Abidin, S.Z.Z., Cyclic Multi-tiers Knowledge Management Model in RAINS, *IEEE ITSim*, 2008.

[20]Yue Liu, Hui Liu, Bofeng Zhang and Gengfeng Wu, Extraction of if-then rules from trained neural network and its application to earthquake prediction, *Proceedings of the Third IEEE International Conference on Cognitive Informatics*, 2004, pp. 109-115.

[21]Liu Jianli and Zuo Baoqi, The Segmentation of Skin Cancer Image Based on Genetic Neural Network, *WRI World Congress on Computer Science and Information Engineering,* Vol. 5, 2009 , pp. 594-599.

[22]Jaafar J., Emerging Trends of Urbanisation in Malaysia. *Journal of The Department of Statistics, Malaysia*, vol. 1, 2004, pp. 43-54.

[23]Sudhira H. S., Ramachandra T. V., Wytzisk A and Jeganathan C., Framework for Integration of Agent-based and Cellular Automata Models for Dynamic Geospatial Simulations. *Technical Report*: 100. Centre for Ecological Sciences, Ind. Ins. of Sci., Bangalore, 2005.

[24]http://chesapeake.towson.edu/landscape/urbansprawl/forms.asp

[25]Yang Q., Lia X. and Shi X., Cellular automata for simulating land use changes based on support vector machines, *Comp & Geosci*., vol 34, 2008, pp. 592–602.

[26]Almeida C. M., Monteiro A. M. V, Camara G., Filho B. S. S., Cerqueria G. C., Araujo W. L., and Eduardo A., Simulating Urban Land Use Change Through CA-based Modelling and Logstic Regression, *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.6878* , 2003.

[27]White, R., Engelen, G. (1993). Cellular automata and fractal urban form: a cellular modelling approach to the evolution of urban land-use patterns. *Environment and Planning*, vol. 25, pages 175–1199

[28]Back T., Dornemann H., Hammel U. and Frankhauser P., Modeling urban growth by cellular automata, *Lecture Notes in Computer Science number 1141*, 1996, pp. 636 – 645

[29]Rongqun, Z., Huiqing, Z., Chengjie, T. and Suhua, M., *WSEAS Transactions on Information Science and Applications*, 2009.

[30]Samat N., Integrating GIS and Cellular Automata spatial model in evaluating urban growth: Prospects and Challenges. *Juru Alam Bina, vol 1 (9)*. 2007, pp. 79-93

[31]Mahajan, Y. and Venkatachalam, P., Neural Network Based Cellular Automata Model for Dynamic Spatial Model in GIS, *ICCSA 2009*, Part I, LNCS 5592, 2009, pp. 341-352

[32]Moghaddam, H.K. and Samadzadegan, F., Urban Simulation using Neural Networks and Cellular Automata for Land Use Planning, *REAL CORP 2009: CITIES 3.0 – Smart, Sustainable, Integrative strategies, concepts and technologies for planning the urban future*, 2009.

[33]Engelbrecht, A.P., *Computational Intelligence: An Introduction*, John Wiley & Sons, 2007.