1-1-2014

# Application of Cellular Neural Networks and Naive Bayes Classifier in Agriculture

Oluleye H. Babatunde
*Edith Cowan University*, obabatun@our.ecu.edu.au

Leisa Armstrong
*Edith Cowan University*, l.armstrong@ecu.edu.au

Jinsong Leng
*Edith Cowan University*, j.leng@ecu.edu.au

Dean Diepeveen

Babatunde, O. H., Armstrong, L. , Leng, J. , & Diepeveen, D. (2014). Application of Cellular Neural Networks and Naive Bayes Classifier in Agriculture. Proceedings of Asian Federation for Information Technology in Agriculture. (pp. 63-72). Perth, W.A. Australian Society of Information and C. Available here

# *Application of Cellular Neural Networks and NaiveBayes Classifier in Agriculture*

***Oluleye Babatunde*[1] *, Leisa Armstrong*[1]*, Jinsong Leng* [1] *and Dean Diepeveen* [2]**

[1]*eAgriculture Research Group, School of Computer and Security Science, Edith Cowan University, Mt Lawley WA, 6050, Australia.*
[2]*Department of Agriculture and Food, South Perth, 6067, WA, Australia.*
Corresponding author's e-mail: hezecomp@yahoo.com; obabatun@our.ecu.edu.au

## *Abstract*

This article describes the use of Cellular Neural Networks (a class of Ordinary Differential Equation (ODE)) , Fourier Descriptors (FD) and NaiveBayes Classifier (NBC) for automatic identification of images of plant leaves. The novelty of this article is seen in the use of CNN for image segmentation and a combination FDs with NBC. The main advantage of the segmentation method is the computation speed compared with other edge operators such as canny, sobel, Laplacian of Gaussian (LoG). The results herein show the potential of the methods in this paper for examining different agricultural images and distinguishing between different crops and weeds in the agricultural system.

**Keywords**: Naive Bayes Classifier, Cellular Neural Networks, Image Classification, Plant Leaves, Fourier Descriptors.

## *Introduction*

There are at least 250,000 to 270,000 plant species around the world (Chomtip, Chawin, Pitchayuk, & Nititat, 2011). Traditional recognition of these plant is carried out by manual matching of the plant's features extracted from the leaves, flowers, and bark of these plants (Meeta, Mrunali, Shubhada, Prajakta, & Neha, 2012). In view of the large number of plant species available, the manual approach to plant classification is tedious, tiring and prone to human error. As a result of this, attempts to automate this process have been made using features of plants extracted from images as input parameters to various classifier systems (Cope, Corney, Clark, & Remagnino, 2011) & (Pahalawatta, 2008). In this paper, a technique to augment already existing techniques of plant identification system is described. The main contribution of this paper is to increase the classification speed of the existing systems by incorporating Cellular Neural Networks for image segmentation. This approach in this work could also be useful for examining different agricultural herbicide damaged leaves or distinguishing between weeds and crops.

## *Cellular Neural Networks*

Cellular Neural Networks (CNN) are variants of ANNs having neighborhood communication as the main distinguishing feature(Chua & Yang, 1988). CNN is modelled from an electrical circuit shown in Figure 1. This circuit was invented in 1988 by Chua and his graduate student Yang at the Department of Electrical Engineering and Computer Sciences, university of California, Berkeley (Chua & Roska, 2002). A standard CNN topological structure is made up of an $M \times N$ or rectangular array of cells $C(i,j)$(or dynamic components) with Cartesian coordinates $(i,j)$, $i = 1(1)M, j = 1(1)N$ as shown in (Hezekiah, Akinwale,

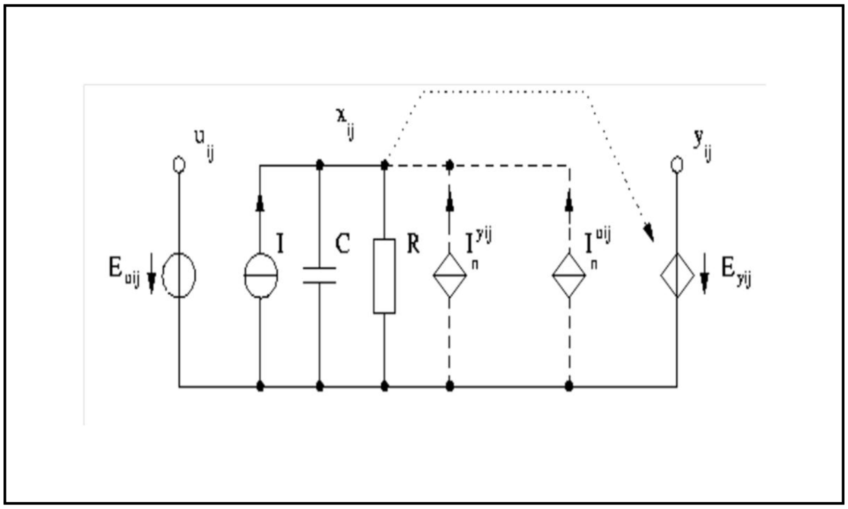


Figure 1: Circuit representing Cellular Neural Networks (Hezekiah et al, 2012)

& Folorunso, 2010). CNN is a hybrid model, sharing features from both Cellular Automata and Artificial Neural Networks. The circuit structure and element values of all cells of a CNN are homogenous. Equation 0.1 governing the behavior of a CNN cell circuit is a dynamical system (or Ordinary Differential Equation (ODE)) derived from evolution laws and circuit theory as shown in Figure 1. The output of the circuit (also called the amplifier), is given in Equation 0.2

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{(k,l)\in\mathcal{N}(i,j)} A(i,j;k,l)\cdot y_{kl}(t) + \sum_{(k,l)\in\mathcal{N}(i,j)} B(i,j;k,l)\cdot u_{kl} + I(i,j;k,l) \qquad (0.1)$$

$$y_{ij}(t) = f(x_{ij}(t)) = \frac{1}{2}(|x_{ij}(t)+1| - |x_{ij}(t)-1|) \qquad (0.2)$$

The variables appearing in Equations 0.1 and 0.2 are defined in the reference papers (Hezekiah et al., 2010) and (Chua & Yang, 1988).

### *The Flavia Dataset*

The source of images of leaves used in this study are images of leaves found in the Flavia dataset which is publicly available (Wu et al., 2007). The Flavia dataset is a constrained set of leaf images taken against a white background and without any stem present. The species in the dataset have a varying number of instances as shown (Babatunde, Armstrong, Leng, & Diepeveen, 2014). The dataset has 1907 images of 32 species of plants. For this study, the dataset was divided into two disjoint sets, each of which contains 1587 images and 320 images for both training and test set respectively.

### *Features Generated From The Flavia Dataset*

### *Image Pre-Processing*

The images found in (Wu et al., 2007) are first pre-processed. The original colored images are converted to grayscale images using the formular in Equation 0.7. One output of this conversion is shown in Figure 2.

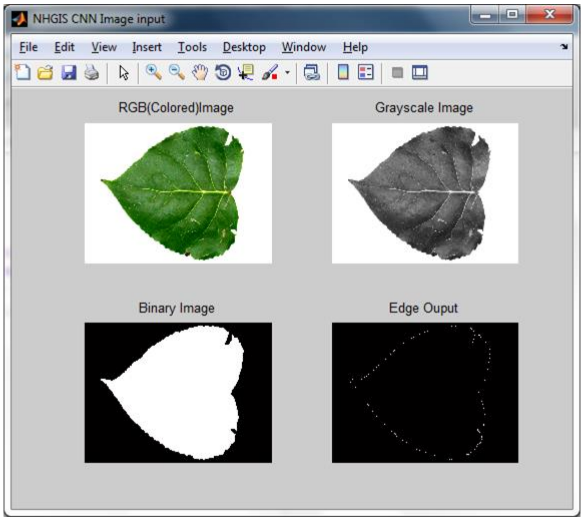


Figure 2: Image Segmentation Using CNN

### *Image Segmentation Using Cellular Neural Networks*

Next to image pre-processing, we employ CNN edge detection templates in Equation 0.3. These templates are the matrix coefficient of the systems of equations in 0.1 & 0.2. The outputs of the segmentation stage are finally passed on to feature extraction modules using FD.

$$\mathbf{A}=\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix},\mathbf{B}=\begin{pmatrix}-1&-1&-1\\-1&8&-1\\-1&-1&-1\end{pmatrix},I=(-1) \tag{0.3}$$

### *Features Extraction Using Fourier Descriptors*

Fourier descriptors (FD) methods have been traditionally used for shape recognition and are part of general methods used in encoding various shape signatures (Charles & Ralph, 1972; Dengsheng & Guojun, 2000; Tyler, 2006). The FT and its inverse are described in (MathWorks, 2007; Fourier, 1878) respectively by formulas in Equations 0.4 and 0.5

$$F(\alpha_1,\alpha_2)=\sum_{x=-\infty}^{\infty}\sum_{y=-\infty}^{\infty}f(x,y)e^{-i\alpha_1 x}e^{-i\alpha_2 y} \tag{0.4}$$

Where $\alpha_1$ and $\alpha_2$ are frequency variables of the image pixels representing the periods of $F$ . The original image of the plant's leaf can re-constructed by using the inverse Fourier Transform given as

$$f(x,y)=\frac{1}{4\pi^2}\int_{\alpha_1=-\pi}^{\pi}\int_{\alpha_2=-\pi}^{\pi}F(\alpha_1,\alpha_2)e^{i\alpha_1 x}e^{i\alpha_2 y}dxdy \tag{0.5}$$

It's assumed that a 2D image is given or generated from 3D image through appropriate color-to-grayscale conversion methods such as found in Equations 0.6 and 0.7. The 2D image generated can then be represented as $f(x_i,y_j),\ ,i=1(1)M,\ ,j=1(1)N$, (where $M,N \in Z^+$). A grayscale image is produced from using any of the formulars in Equation 0.6 and 0.7. The edge output from which the boundary pixels are extracted is shown in Figure (2)

$$Method1 = \frac{(R+G+B)}{3} \tag{0.6}$$

$$Method2 = 0.299R + 0.587G + 0.114B \tag{0.7}$$

### *Computation of Fourier Descriptors*

The FD is used in describing the boundary of shapes in 2D images using Fourier Transform (FT). The steps needed to compute FDs are as given in the subsections following.

#### *Boundary Parametization*

The boundaries of the image is given as the set $\big((x_1,y_1),(x_2,y_2),(x_3,y_3),...(x_N,y_N)\big)$ containing $N$ ordered points/pixels.

#### *Boundary Tracing*

Any arbitrary point, say, $(x_0,y_0)$, is chosen as the starting point and the travesal/tracing of all the boundaries to the left and right of point $(x_0,y_0)$. A complete list of boundary pixels will thus be obtained after this.

#### **Complex representation of boundary points**

The boundary pixels set $(x_i,y_j), i=0(1)N, j=1(1)N$, is then represented as complex variables $(x_n + jy_m), n=0(1)N, m=1(1), j^2=-1$

#### *Application of Fourier Transform*

Next to complex representation above, appropriate Fourier Transform such as FFT or DFT ((Mathworks, 2009))is applied to it. The coefficient thus obtained are called **Fourier Descriptors**.

#### *Invariant FDs*

Suppose the descriptors are given as $|FD_1|,|FD_2|,|FD_3|,...,|FD_N|$. To achieve invariant properties with respect to rotation, scaling, and translation, the following steps are performed:
(a) Set $|FD_1| = 0$
(b) Divide the remaining FDs by the first of the remaining FDs. This implies that $|FD_2| \mapsto \frac{|FD_2|}{|FD_2|}$,
$|FD_3| \mapsto \frac{|FD_3|}{|FD_2|}$,
$|FD_4| \mapsto \frac{|FD_4|}{|FD_2|}$,
$\vdots$
$|FD_N| \mapsto \frac{|FD_N|}{|FD_2|}$

### *Feature set*

The first twenty coefficients of FDs for all the images in Wu et al. (2007) were used to form the feature set in this work. These features are represented in Table 1.

Table 1: 20 features extracted from the Flavia Dataset (Babatunde et al, 2014)

| **Observation** | **F1 F2 F3 F4 F5 F6 F7 F8 F9 ...F20** | Class No |
|---|---|---|
| Image1 | $X_{1,1}$ $X_{1,2}$ $X_{1,3}$ $X_{1,4}$ $X_{1,5}$... $X_{1,20}$ | 1 |
| Image2 | $X_{2,1}$ $X_{2,2}$ $X_{2,3}$ $X_{2,4}$ $X_{2,5}$... $X_{2,20}$ | 1 |
| Image3 | $X_{3,1}$ $X_{3,2}$ $X_{3,3}$ $X_{3,4}$ $X_{3,5}$... $X_{3,20}$ | 1 |
| Image4 | $X_{4,1}$ $X_{4,2}$ $X_{4,3}$ $X_{4,4}$ $X_{4,5}$... $X_{4,20}$ | . |
| Image5 | $X_{5,1}$ $X_{5,2}$ $X_{5,3}$ $X_{5,4}$ $X_{5,5}$... $X_{5,20}$ | . |
| Image6 | $X_{6,1}$ $X_{6,2}$ $X_{6,3}$ $X_{6,4}$ $X_{6,5}$... $X_{6,20}$ | . |
| ... | ... | 32 |
| ... | ... | 32 |
| ... | ... | 32 |
| Image1907 | $X_{1907,1}$ $X_{1907,2}$ $X_{1907,3}$... $X_{1907,20}$ | 32 |

### *Baye's rule and Bayesian Classifier*

The FDs used for providing the training set for classification of the images used are represented in Table 1. Each class $c_i, i = 1(1)32$, has a priori probability $p_i$ of occuring in the dataset. The rationality behind the Baye's rule is to compute the a posteriori probabilities from the a priori probabilities and the evidence. The feature space in Table 1 can be represented as measurable pair $(X,c_i) \in \mathbb{R}^D \times \{1,2,3,...,32\}$ where $c_i = 1,2,3,...,32$ is the class label and $D$ is the number of features in the given input(feature space). This implies that the conditional distribution of $X$, given that the class information $c_i$ takes on a value $i \in Z^+$ is given by $X|c_i = i \dashrightarrow P_i$ for $i = 1(1)32$ where "$\dashrightarrow$ " means "distributed as ", and where $P_i$ is a probability distribution. A classifier is a functional that assigns to an observation $X = c_i$, a measure (mostly numeric) of what the class information for the unseen data actually was. In formal (mathematical) notation based on the dataset in Table 1, a classifier is a measurable function $ClassMap : \mathbb{R}^D \mapsto \{1,2,3,...,32\}$, where the $D$-dimensional input vectors are mapped to any of the intergers $1,2,3,...,32$. Bayesian Classifier is a probabilistic classifier model and a probabilistic classifier model (based on Table 1) is a conditional model $p(X|c_1,c_2,c_3,...,c_{32})$ over a dependent class variable $c_i$, i = 1(1)32, conditioned on several features $(F_1,F_2,F_3,...,F_{20})$. The definitions (0.1 to 0.3) are needed to understand Bayesian classifier expressed in definition 0.4.

**Definition 0.1.** A Posteriori $\mapsto p(X|c_i)$. This is the probability of $X$ given the evidence $c_i$. This can also be expressed as

$$posteriori = \frac{prior \times likelihood}{evidence}((Bishop,1995)).$$

**Definition 0.2.** A priori $\mapsto p(X)$. This is the probability of $X$ only.

**Definition 0.3.** Baye's rule is given as :

$$p(c_j|X) = \frac{f(X|c_j)p(c_j))}{f(X)}$$

The Bayesian Classifier is based on finding the class for a given test data $X$. Once the probability measure for every class $c_j$ in the pattern unit is calculated, the test data $X$ is classified as belonging to the class $c_j$ for which $p(c_j|X)$ is maximized . In order to estimate $p(c_j|X)$, we need to compute the class conditional probabilities $f(X|c_j)$ and the a priori probabilities $p(c_j)$ for every class $c_j$ in the class information. The calculation of $f(X)$ is not compuslory because its value is homogenous across all classes. The a priori probabilities are computed from the training dataset as discussed in (Babatunde et al., 2014). According to Parzen (1962), the class conditional probabilities $f(X|c_j)$ can be computed from the training data using the Equation 0.8.

$$f(X|c_j) = \frac{1}{(2\pi)^{M/2}\sigma^M N_j} \sum_{i=1}^{N_j} \exp\left[-\frac{(X-X_i^j)^T(X-X_i^j)}{2\sigma^2}\right] \tag{0.8}$$

where
$M$ = dimensionality of the features; $N_j$ is Number of training patterns belonging to class $j$; $X_i^j = i_{th}$ training pattern of $X^j$ and $\sigma$ is Gaussian smoothing parameter or standard deviation.

**Definition 0.4.** The Baye's classifier is thus defined as $CLASSIFIER_{Bayes}(x) = argmax(P(c_i = i|X = x))$

### *Learning system based on Naive Bayes Classifier*

The steps involved in the design of the classification system shown in Figure 4 are described in Figure 3. These steps are pretty much the same with most image classification systems. The entire implementation (including the GUI) was done using MATLAB 2013a. The naive Bayes classifier assumes that the presence or absence of a particular feature unrelated to the presence or absence of any other feature, given the class variable. With this assumption, conditional distribution over the class variable $c_j, j = 1(1)32$ is given as:

$$p(c_j|X_1, X_2, X_3, ..., X_n) = \frac{1}{\alpha} p(c_j) \prod_{i=1}^{n} p(X_i|c_j) \tag{0.9}$$

where $\alpha$ = evidential scale factor depending on the number of features in the training set. Thus the corresponding naive Bayes classifier is given as:

$$classify(x_i \in X) = argmax\ p(c = c_i) \prod_{i=1}^{n} p(X_i = x_i|c = c_i), i = 1(1)n, n > 1\ \&\ n \in Z^+. \tag{0.10}$$

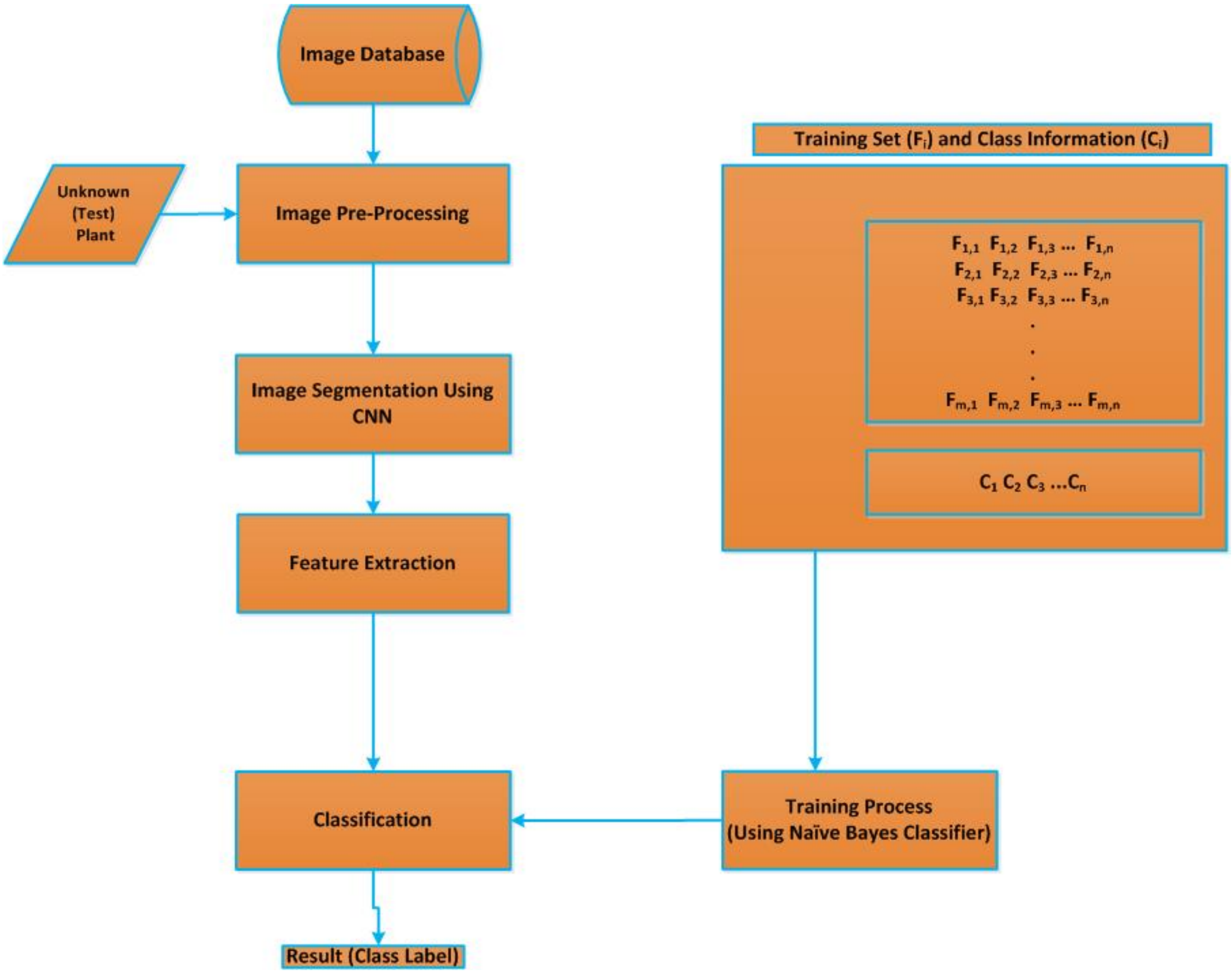


Image Database
Unknown (Test) Plant
Image Pre-Processing
Image Segmentation Using CNN
Feature Extraction
Classification
Result (Class Label)
Training Set ($F_i$) and Class Information ($C_i$)
$F_{1,1}$ $F_{1,2}$ $F_{1,3}$ … $F_{1,n}$
$F_{2,1}$ $F_{2,2}$ $F_{2,3}$ … $F_{2,n}$
$F_{3,1}$ $F_{3,2}$ $F_{3,3}$ … $F_{3,n}$
$F_{m,1}$ $F_{m,2}$ $F_{m,3}$ … $F_{m,n}$
$C_1$ $C_2$ $C_3$ …$C_n$
Training Process (Using Naïve Bayes Classifier)

Figure 3: Learning system based on Naive Bayes Classifier

### *Experimental validation*

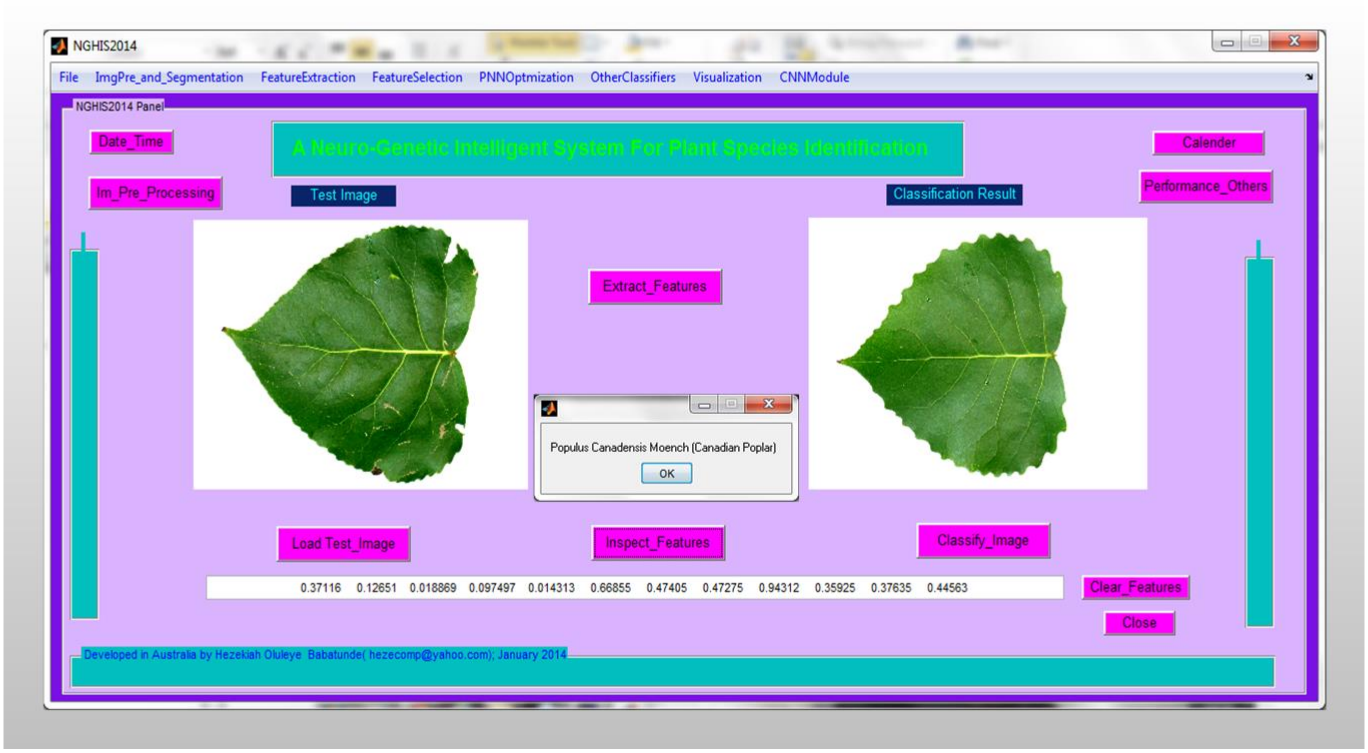
The approach used in validating the naive Bayes Classifier herein is the k-Fold Cross Validation (k-Fold CV) with k = 10. Generally, a cross validation (CV) is a method of partitioning the feature space into training and testing sets as shown in (Babatunde et al., 2014). Herein, the naive Bayes Classifier was fitted using training set, while the fitted model was validated through testing set by measuring the error predicted. The training set and testing set were both disjoint to ensure that the testing set for evaluating the naive Bayes Classifier are not used in fitting the model. Without the loss of any generality, and suppose the information in section  are true, the dataset (feature space) $X$ is then partitioned into two sets viz $X = X_1 \cup X_2$, such that $k$ elements are in $X_1$ and $D-k$ elements in $X_2$. The naive Bayes classifier was then trained or fitted using the set $X_2$. The historical pattern of $X_2$ was used to produce classifications (predictions) results for observations $X_{X_1} \in X_1$ given $X_2$.

### *Results and Discussion*

A Naive Bayes classifier assigns a new observation (unseen test data) to the most probable class, assuming the features are conditionally independent given the class value. The classifier herein has three methods viz $f1(.)$, $f2(.)$, & $f3(.)$ . The $f1(.)$ fits a NBC to the training data, $f2(.)$ predicts the class label for test data and $f3(.)$ assigns posterior probability for each class of the test data. The posterior probability is already defined in section . The prior property for NBC is a vector of length NClasses containing the class priors. An empty class is assigned a prior value of zero. Gaussian distribution was used as the functional for pdf estimation. This classifier is one of the simplest classifiers available but it works incredibly well. The results of this work are shown in Table 2. Several image segmentation techniques including canny, prewitt, LoG were compared with the CNN used in this work. The CNN was solved using runge-kutta method. The whole system was unbiasedly evaluated using CV with k = 10. Accuracy and computational time (in seconds) were used as performance metric for the developed system. It's shown in the table that the system performed better than the conventional image segmentation techniques (edge detectors).

Table 2: Classification accuracy and computational time

| **Edge Detector** | **Time (seconds)** | **Accuracy (%)** |
|---|---|---|
| CNN | 6.01 | 70.45 |
| Sobel | 7.25 | 70.12 |
| Canny | 7.10 | 70.28 |
| LoG | 7.68 | 69.25 |
| Robert | 7.66 | 70 |



Populus Canadensis Moench (Canadian Poplar)

```
0.37116   0.12651   0.018869   0.097497   0.014313   0.66855   0.47405   0.47275   0.94312   0.35925   0.37635   0.44563
```

Developed in Australia by Hezekiah Oluleye  Babatunde( hezecomp@yahoo.com); January 2014

Figure 4: Image processing tool

## *Conclusion*

A demonstration of classification capability of Naive Bayes classifier (NBC) has been described in this paper . The system was found to perform better when compared to conventional image segmentation techniques (edge detectors) as evidenced by reduction in computational time and increased classification accuracy (see Table 2). In comparison to the classifier used in the paper (Wu et al., 2007), NBC has less classification accuracy but faster in computation. This work shows that NBC works well in image classification.

## *Acknowledgements*

This work was supported by Edith Cowan University International Post Graduate Research Scholarship (ECUIPRS, Australia) and Tertiary Education Trust Fund (TETF, Nigeria).

# References

Babatunde, O., Armstrong, L., Leng, J., & Diepeveen, D. (2014). Zernike moments and genetic algorithm: Tutorial and application. *British Journal of Mathematics and Computer Science*, *4(15)*, 2217-2236.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

Charles, T. Z., & Ralph, Z. R. (1972). Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, *C-21*(3), 269-281.

Chomtip, P., Chawin, K., Pitchayuk, S., & Nititat, S. (2011). Leaf and flower recognition system (e-botanist). *IACSIT International Journal of Engineering and Technology*, *3(4)*, 10-15.

Chua, L. O., & Roska, T. (2002). Cellular neural networks and vision computing. *Cambridge University Press*.

Chua, L. O., & Yang, L. (1988). Cellular neural networks: theory and applications. *IEEE Trans on Circuits and System*, *35*(10), 1257-1272.

Cope, J. S., Corney, D., Clark, J. Y., & Remagnino, P. W. (2011). Plant species identification using digital morphometrics: A review. *Digital Imaging Research Centre, Kingston University,London,UK and Department of Computing, University of Surrey, Guildford Surrey, UK*, 1-21.

Dengsheng, Z., & Guojun, L. (2000). A comparative study on shape retrieval using fourier descriptors with different shape signatures. *Gippsland School of Computing and Information Technology,Monash University,Australia*.

Fourier, J. B. (1878). *The analytical theory of heat.* The Universal Press.

Hezekiah, B., Akinwale, A. T., & Folorunso, O. (2010). A cellular neural networks- based model for edge detection. *Journal of Information and Computing Science*, *5*(1), 003-010.

MathWorks. (2007). Matlab neural network toolbox documentation. *MathWorks. Inc. [Online]. Available:.*

Mathworks. (2009). Signal processing toobox (discrete fourier transform (dft)). *MathWorks. Inc*.

Meeta, K., Mrunali, K., Shubhada, P., Prajakta, P., & Neha, B. (2012). Survey on techniques for plant leaf classification. *International Journal of Modern Engineering Research (IJMER)*, *1*(2), 538-544.

Pahalawatta, K. K. (2008). A plant identification system using both global and local features of plant leaves. *MSc Thesis at the department of Computer Science and Software Engineering, University of Canterbury, New Zealand*, 1-127.

Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, *Vol 33, Issue 3*, 1065-1076.

Tyler, K. (2006). Fourier descriptors: Properties and utility in leaf classification. *ECE 533 Fall 2006*.

Wu, S. G., Bao, F. S., Xu, E. Y., Wang, Y.-X., Chang, Y.-F., & Xiang, Q.-L. (2007). A leaf recognition algorithm for plant classification using probabilistic neural network. *IEEE 7th International Symposium on Signal Processing and Information Technology, Cario, Egypt; ArXiv 0707.4289 v1 [ CS.AI]*.