

Support Vector Machine Vs an Optimized Neural Network for Diagnosing Plant Diseases

Mohammed Sammany

Dep. of Mathematics
Faculty of Science
Cairo University, Egypt
sammanyddr1@hotmail.com

Khaled Saeed Saad Zaghloul

Dep. of Computer Engineering
Faculty of Engineering
Cairo University, Egypt
khaled.saeed84@gmail.com

ABSTRACT —Vegetable crops suffer from many leaf batches, which differ in color, shape, and size according to the cause. Leaf batches happen as a result of plant pathogens. In agriculture mass production, it is needed to discover the beginning of plant disease batches early to be ready for appropriate timing control. In this regard, Support Vector Machine (SVM) has been used to classify the plants symptoms according to their appropriate categories, these categories are Yellow Spotted (YS) category, White Spotted (WS) category, Red Spotted (RS) category, and discolored category (D). The results obtained using SVM have been compared to the results obtained by an optimized Multi-layered Perceptron (MLP).

Key Words— Support Vector Machine (SVM), Multi Layer Perceptron (MLP)

I. INTRODUCTION

Vegetable crops suffer from many leaf batches, which differ in color, shape, and size according to the cause. Leaf batches happen as a result of plant pathogens (fungi, Bacteria, and Virus diseases), insect feeding (sucking insect pests), and plant nutrition (lack of micro elements) [1]. In agricultural mass production, it is needed to discover the beginning of plant diseases' batches early to be ready for appropriate timing control to reduce the damage, minimize production costs, and increase the income. Plant leaves are considered the first station for the rest and germination of bacterial and fungal capsules due to the suitable macro environment [2]. Leaf batch characteristics play a curial role in differentiating between the different causes. The diagnosis of leaf batches may cause some confusion due to the similarities in batch shape, size, and color. Only an expert could identify it. The first step in fighting against these leaf batches is the adequate recognition of their presence (i.e.) correct diagnosis. An abnormal symptom is an indication to the presence of diseases, and hence, can be regarded as an aid in diagnosis.

Leaf batches (spots) are considered the important units indicating the existence of diseases. In order to assign those leaf spots to their cause, we first need to extract their features such as color, shape, and size. Second, we need a classifier capable to learn from those features and then differentiate between them. In this paper, we used SVM as a tool to classify the plant symptoms according to their appropriate categories, which are the yellow spotted (YS), white spotted (WS), red spotted (RS), and discolored (D) categories. The results obtained using SVM have been compared to those of MLP. This paper is organized as follows: Feature extraction will be presented in Section (II). Sections (III) describes MLP as a classifier. Using genetic algorithms as a tool for optimizing the network architecture and parameters is presented in section (IV). Section (V) describes the classifier objective from SVM's point of view. Finally, we introduced the experimental results with a comparative study between the two methods in section (VI). The paper ends with a conclusion and future work.

II. FEATURE EXTRACTION

In order to recognize the spot category, a number of features are extracted from a segmented image to be later used for classification. These features correspond to color characteristics of the spots such as the mean of the gray level, of the red, green, and blue channel of the spots. Other features correspond to morphological characteristics of the spots [3, 4] such as:

- **The length of the principal axes:**
Major and minor axes length of a spot
- **The diameter of a spot d** is measured as:

$$d = \sqrt{\frac{4 \times Area}{\pi}} \quad (1)$$

- **Eccentricity Measure**, also called circularity ratio (CR), its value between $[0,1]$ the spot whose circularity ratio is zero is actually a circle, while the spot whose circularity ratio is one is actually a line. The circularity ratio is computed as:

$$CR = \frac{\sqrt{(\text{major})^2 - (\text{minor})^2}}{\text{major}} \quad (2)$$

- **Compactness Measure**, also called solidity ratio, has a value between $[0,1]$. If the spot has a solidity value equal to one, this means that it is fully compacted. It is the ratio between the spot area to the convex hull. The formula is computed as:

$$\text{Ratio} = \frac{\text{SpotArea}}{\text{ConvexHull}} \quad (3)$$

- **Extent Measure**, also called rectangularity ratio, has a value between $[0,1]$, when this ratio has the value one, then the shape is perfectly rectangle, extent measure is computed as:

$$EM = \frac{\text{SpotArea}}{\text{BoundingBo xArea}} \quad (4)$$

- **Euler's Number Measure**: This measure describes a simple topologically invariant property of the spot. It is computed as the number of objects in the region minus the number of holes in those objects.
- **Orientation Measure**: Is the angle in degrees between the x - axis, and the major axis length of the spot.

Since our main objective, is to build a classifier capable of classifying leaves symptoms. We distributed the disorders in each category as follows: The first category (YS) contains 11 disorders, which are Leaf Blight, Leaf Spot, Downey, High Temp, Jassid, Magnesium Def, Potassium Def, Salt Injury, Scab, Spider, and Zinc Def. The second category WS contains 9 disorders, which are Aphids, Leaf Miner, Magnesium Def., Manganese Def, Powdery, Slat Injury, White Fly, Leaf Spots, and Tobacco Virus. The third category RS contains 9 disorders, which are Leaf Blight, Anthracnose, Downey, Gummy Stem Blight, Leaf Spot, Pesticide Injury, Phosphorus Def, Spider, and Toxicity. The fourth category D contains 9 disorders, which are Downey, Iron Def., Manganese Def., Mosaic, Nitrogen Def., Pesticide Injury, Potassium, Salt injury, and Trips (see Appendix) [5]. Images for each class have been collected as follows, 20 images for each of the WS , YS , and RS categories, 32 images for the D category, and 25 images for normal category. The analysis of these images showed that

there is a relationship between those categories. Venn diagrams have been used to represent the overlap between each of these categories as shown in figure (1).

- The discolored category is overlapped by 44%-55% with other categories.
- The Yellow Spotted category is overlapped by 33%-55% with other categories.
- The Red Spotted category is overlapped by 33%-55% with other categories.
- The White Spotted category is overlapped by 33%-55% with other categories.

Table (1) summaries the total number of spots for each category that constitute our training set.

Category Name	Number of Images	Total Number of Spots
Red Spotted (RS)	20	259
White Spotted (WS)	20	514
Yellow Spotted (YS)	20	185
Discolored (D)	32	337
Normal (N)	25	573
Total	117	1868

Table (1) Total number of images and spots for the five categories, which has been used in the training process.



Figure (1) Venn diagrams representing the overlap between categories

III. CLASSIFIER OBJECTIVE USING MLP

The objective of MLP is to assign the input patterns to one of the categories that are represented in terms of neural networks' outputs, so that they represent the probability of class membership. Figure (2.a) illustrates a symbolic column before and after the symbolic translation process. A "1" in an expanded column indicates the occurrence of the column's corresponding string and a "0" indicates a non-occurrence. For the construction of MLP to manage the classification task of our problem, we put five neurons in the output layer (neuron for each class) with Tanh Sigmoid activation function for every neuron. Figure (2.b) represents a snapshot of the training set used to train our neural network.

	A	B	C	D	E	F		A	B	C	D	E	F
1	Class						1	Class					
2	RS						2	RS	1	0	0	0	0
3	WS						3	WS	0	1	0	0	0
4	YS						4	YS	0	0	1	0	0
5	D						5	D	0	0	0	1	0
6	N						6	N	0	0	0	0	1

(a)

	A	B	C	D	E	F	G	H	I	J	K	RS	WS	YS	D	N
1	210.95	196.66	120.06	0.8	0.56	0.78	20.12	28.05	16.73	-5	133.74	1	0	0	0	0
2	210.95	196.66	120.06	0.8	0.56	0.78	20.12	28.05	16.73	-5	133.74	0	1	0	0	0
3	212.83	198.21	120.55	0.55	0.6	0.76	18.3	21.56	17.95	-6	23.96	0	0	1	0	0
4	212.83	198.21	120.55	0.55	0.6	0.76	18.3	21.56	17.95	-6	23.96	0	0	0	1	0
5	213.92	148.24	100.91	0.89	0.37	0.62	15.18	27.24	12.67	-1	46.29	0	0	0	0	1
6	214.51	161.41	129.25	0.92	0.37	0.63	25.93	50.37	19.96	-7	-30.32	0	0	0	1	0
7	219.43	199.5	118.14	0.94	0.36	0.7	33.87	67.14	22.88	-10	132.08	1	0	0	0	0
8	220.16	198.96	136.03	0.95	0.24	0.55	28.79	63.13	20.32	-12	42.35	0	0	0	0	1
9	220.75	174.74	151.66	0.9	0.52	0.81	17.11	27.16	11.99	0	25.56	0	0	1	0	0
10	221.57	167.65	132.27	0.98	0.24	0.51	14.54	41.81	8.69	0	65.14	0	0	0	1	0
11	221.71	162.05	122.84	0.96	0.31	0.53	27.91	70.09	19.2	-4	-3.34	1	0	0	0	0

(b)

Figure (2) (a) symbolic column before and after the symbolic translation process (b) snap shot of the training set used to train our neural network

IV. INTEGRATING GENETIC ALGORITHMS WITH NEURAL NETWORKS TO OPTIMIZE CLASSIFIER CONFIGURATION

Genetic Algorithms [6] can be used to determine the best network parameters by successively trying and testing different combinations of parameters. Like evolution, good parameter sets are more likely to survive from one population to the next. Genetic optimization can be used to set many of the parameters in a neural network (e.g. number of hidden neurons, learning rates, input selection ... etc). In the field of neural networks, the genetic optimization takes place by repeatedly training the network with various parameters and calculating the best MSE for each network [7]. Since genetic algorithms are searching algorithms based upon the principles of evolution observed in nature, it combines selection, crossover, and mutation operators with the goal of finding the best solution to a problem. Genetic Algorithms search for this optimal solution until a specified termination criterion is met. The criterion used here to evaluate the fitness of each potential solution is the lowest cost achieved during the training run (Cross validation) [8].

The solution to a problem is called a *chromosome*. A chromosome is made up of a collection of genes, which are simply the neural network parameters to be optimized [9]. A genetic algorithm creates an initial population (a collection of chromosomes) and then evaluates this population by training a neural network for each chromosome. It then evolves the population through multiple generations (using the genetic operators) in the search for the best network parameters. In this section, we describe the way we used genetic algorithms in order to search a space of neural network topologies and parameters to select those that match, optimally, our criteria.

- **Encoding**

The neural network is defined by a "genetic encoding" in which the genotype codes for the different characteristics of the MLP and the phenotype is the MLP itself. Therefore, the genotype contains the parameters related to the neural network model, i.e. the number of neurons in each layer l (NH l), learning rate η , momentum constant α , and decay parameter λ . In the genetic algorithm used here, the chromosome structure $y = \{y_1, y_2, \dots, y_t\}$, constituted by $t = 5$ loci, is reported in figure (3).

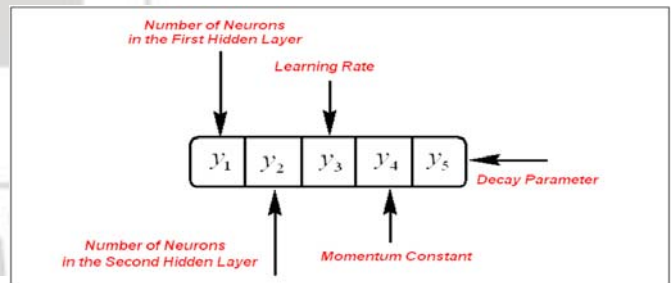


Figure (3) The graphical representation of the genetic encoding

Each gene is defined in the subset $A_i, i = \{1,2,3\}$ reported in the third row of the table (2).

Locus	y_1	y_2	y_3	y_4	y_5
Gene	NH1	NH2	η	α	λ
Set	$A_1 = [1,30]$	$A_2 = [0,30]$	$A_3 = [0,1] \subset R$	$A_4 = [0,1] \subset R$	$A_5 = [0,1] \subset R$

Table (2) Genetic encoding, the maximum value for NH l has been fixed equal to 30. The learning rates are restricted on the interval [0,1]

- **The Fitness Function**

Since the generalization ability of neural network, is computed with reference to validation set. Therefore, to evaluate the goodness of an individual, the parameter

which seems to describe better its goodness is the mean squared error on the validation set ,that is,

$$CVE = \frac{1}{N_{CV}} \sum_{i=0}^{N_{CV}} |E_i|^2 \quad (5)$$

where N_{CV} the number of patterns in the cross validation set, and E_i is the error between the desired and the network output.

• **The Optimization Algorithm**

A population of μ neural networks, which represents potential candidate solutions, is let free to evolve. These solutions are individually evaluated to determine how well they solve the problem. A selection process allows establishing the survivors. They mate by means of genetic operators to create the individuals of the next generation. The process is repeated until the stopping criterion is fulfilled, this scheme is referred to as Simple Genetic Algorithm (SGA) [10]. Let MLP (y_i); $i \in \{1, \dots, \mu\}$ is the algorithm for training a MLP related to the individual y_i (which represents the neural network configuration). The general scheme that has been used in the optimization process is given by the following pseudo-code.

```

Given a training set  $T_s$ ;
begin
randomly initialize a population of  $\mu$  neural network structures;

while (termination criterion not fulfilled) do

    train  $y_i$  by means of MLP( $y_i$ ) on  $T_s$ ;
    evaluate the trained network  $y_i$ ;
    save the best trained network in the new population;
    select the best  $\lambda$  neural network configurations;

    for  $i = 1$  to  $\mu - 1$  do
        randomly select two structures among the  $\lambda$ ;
        recombine them so as to obtain one offspring;
        perform mutation on the offspring;
    od

update variables for termination;
od
end
    
```

Figure (4).The pseudo-code of the genetic algorithm used for optimizing the neural network

The genetic algorithms parameters in addition to the optimized architecture of our neural network and its parameters are all summarized in tables (3) and (4), respectively.

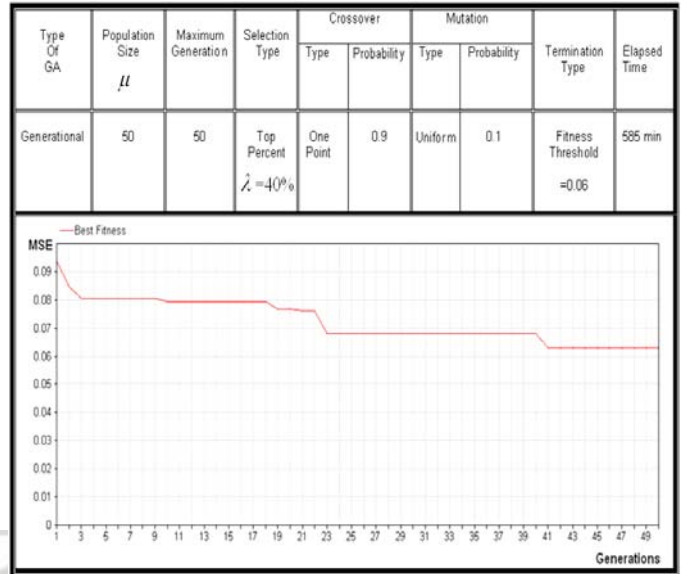


Table (3) The parameters of genetic algorithms used for optimizing the neural network

NN Optimized Architecture Based GA	NN Optimized Learning Rates For Each Layer			NN Optimized Momentum Constants For Each Layer			NN Optimized Decay Parameters For Each Layer		
	η_1	η_2	η_3	α_1	α_2	α_3	λ_1	λ_2	λ_3
11-27-20-5	0.6113	0.4401	0.3485	0.0566	0.1282	0.0449	0.0483	0.04	0.0219

MSE Average	CVE Average	Test Avg	Iterations
0.0328	0.0683	80%	5000

Table (4) The optimized architecture and parameters of neural network with the classification results for the training and the cross validation sets

The results summarized in table (4) reveal that, the optimized neural networks capable of classifying the infected images into their five categories with Mean Squared Error for the Cross Validation set CVE=0.0683, and for the training set MSE=0.0328. The neural network total average classification accuracy is 80%. In order to make a comparative study with Support Vector Machine (SVM), and for more evaluation of the classifier capabilities, additional 400 images (not seen before) have been introduced to the optimized network, (100 images for each category), and the recall percentage has been calculated for each class. Before presenting the results obtained using SVM, we will give a brief overview on how the SVM technique works as a classifier.

V. CLASSIFIER OBJECTIVE USING SVM

The basic idea of SVM is to map a given data set from input space into higher dimension feature space F , called **dot product space**, via a map function ϕ , where

$$\phi: R^N \rightarrow F \tag{6}$$

Then, it performs a linear learning algorithm in F . This requires the evaluation of dot products

$$K(x, y) = (\phi(x), \phi(y)) , \tag{7}$$

where $K(x, y)$ is called the **kernel function**. If F is high dimensional, then the right hand side of equation (7) will be very expensive to compute [11]. Therefore, kernel functions are used to compute the dot product in the feature space using the input parameters.

There are many types of kernels such as:

$$K(x, x_i) = x_i^T x \quad (\text{Linear}) \tag{8}$$

$$K(x, x_i) = (x_i^T x + \tau)^d \quad (\text{Polynomial of degree } d) \tag{9}$$

$$K(x, x_i) = e^{-\|x-x_i\|_2^2/\sigma^2} \quad (\text{RBF}) \tag{10}$$

$$K(x, x_i) = \text{Tanh}(\alpha \cdot x_i^T x + \theta) \quad (\text{MLP Kernel}) \tag{11}$$

In machine learning problems, it is required to classify unseen patterns to more than one class; the function used for that purpose is called the decision function $f(x)$, it is given by:

$$f(x) = \begin{cases} +1 & \text{if } \sum_{i=1}^l \alpha_i y_i K(x, x_i) + b \\ -1 & \text{if } \text{Otherwise} \end{cases} \tag{12}$$

where,

l the number of training patterns

x unseen pattern vector

x_i the i^{th} training pattern vector

y_i label of the i^{th} training pattern

b Constant offset (or threshold)

1 and -1 are the labels of decision classes

The parameters α_i can be computed as the solution of a **quadratic programming** problem of the form:

$$\underset{w \in \mathbb{N}, b \in \mathbb{R}}{\text{minimize}} \quad \tau(w) = \frac{1}{2} \| w \|^2$$

Subject to $y_i (\langle w, x_i \rangle + b) \geq 1$ for all $i = 1, \dots, l$ (13)

where,

w Weight Vector in feature space

\mathbb{N} Feature Space

\mathbb{R} Set of Real

τ Objective Function

where w is the weight vector perpendicular to the decision hyper-plane [11].

The computed non-zero α_i 's correspond to training patterns known as **support vectors**. Finally, substituting the values of α_i in (12) produces the decision function **hyper-plane** in the feature space that corresponds to a nonlinear function in the input space as shown in figure (5). Thus, the classification problem becomes easier to be solved in the higher dimension space than in the lower dimension space.

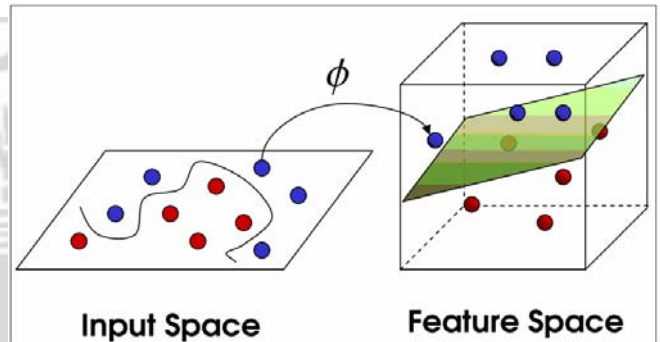


Figure (5) Mapping data to the higher dimensional feature space

To get M-class classifiers, a set of binary classifiers f^1, f^2, \dots, f^5 was constructed. Each trained to separate one class from the rest. Due to the fact that the problem permits a pattern being assigned to more than one class, the binary classifiers were not combined.

VI. EXPERIMENTAL RESULTS

An optimized neural network with two hidden layers has been used (The other parameters are demonstrated in table (4)). This architecture has been trained on a training set consisting of 1468 patterns representing various infected leaves from the five categories. Twenty percent of this training set has been truncated for validation. On the other hand, SVM with a polynomial kernel, where $d = 2$, has been trained using the same data set. Both methods have been tested on the same set using 400 patterns not seen before (100 patterns for each class). The results obtained using MLP versus SVM are shown in table (5). From this table we can see that, the total average accuracy of the SVM over the five classes is **83%**, whereas using an optimized MLP it was **80%**. This means that the performance of SVM outperforms the performance of MLP.

<i>Classes</i>	<i>Accuracy of MLP</i>	<i>Accuracy of SVM</i>
RS	89%	89 %
WS	93%	92%
YS	64%	80%
D	80%	75%
N	74%	79%
Average	80%	83%

Table (5). The performance of SVM Vs MLP, for each class

VII. CONCLUSION AND FUTURE WORK

Due to the extensive overlap among the plant symptoms, the problem of diagnosing plant diseases is regarded as a one of the most complex issues in the field of non-linearly separable problems. In this paper, we used SVM as a tool for classifying the plant diseases according to their appropriate category. The results obtained using SVM has bee compared to the results obtained by MLP, which has been optimized by means of genetic algorithms. Finally, we concluded that SVM is more efficient than MLP in solving such kind of problems.

Although SVM method is a good technique, which is used in many areas, it still suffers from some disadvantages. One of the major disadvantages of SVM is embodied in determining the appropriate kernel and its parameters to optimize the overall performance of the machine.

A most recently discovered approach can be used to perform the same task done in this paper. This new approach depends on the idea of taking objects, attributes, and decision values, and create rules for upper, lower, and boundary approximations of a set. With these rules, a new object can easily be classified into one of the set regions. This new promising approach referred to as **Rough Sets**. Rough sets method was discovered by the *Polish* scientist of mathematics Zidizalaw Pawlak and his co-workers from the *Polish academy of science*. It has a wide range of uses, such as medical and financial data analysis, stock market prediction, voice recognition, and image processing...etc. Rough sets are also helpful in dealing with vagueness and uncertainty in decision situations [12].

In our future work, we are looking forward to use the Rough Sets method in this, and other applications, and to compare our results with SVM.

ACKNOWLEDGMENTS

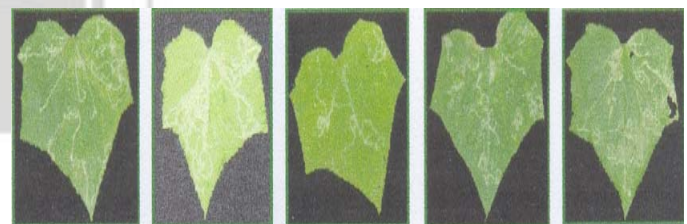
The authors would like to thank:

- The supervisor of this work, Dr.Amir F .Atiya from the Department of Computer Engineering, Cairo University, Egypt, for his scientific and technical support.

- Dr Mohammed Hussein El-Helly, from the Central Lab of Agriculture and Expert Systems (CLAES), Cairo, Egypt, for providing the data set used in our simulations, and his support concerning the part of diagnosing plant diseases.
- Prof Bernard Scholkopf, from Max Plank Institute for Biological Cybernetics, Tübingen, Germany, for his encouragement, and for providing the simulation software package [13].
- Dr Mohammed El-Beltagy, from the Department of Decision Support Systems, faculty of Computers and Information, Cairo University, Egypt, for his lectures concerning SLT, and SVM, from Feb-2006 to May-2006.

APPENDIX

In this appendix, we introduce some images of infected leaves, which constitute our training set [5].



CLAES Samples of defected Images [5]



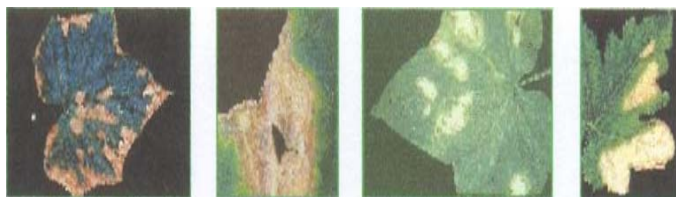
CLAES Images for Leaf Miner [5]



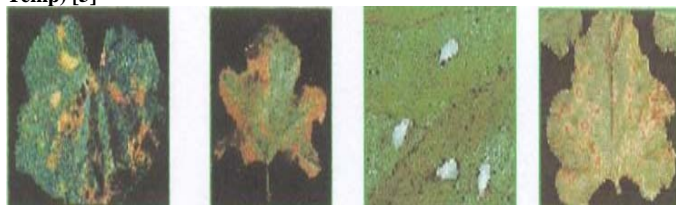
CLAES Images for Downey Mildew [5]



CLAES Images for Powdery Mildew [5]



From Left right (Phosphorus Def – Gummy Stem Blight- Scab- High Temp) [5]



From Left right (Anthracnose – Pesticide Injury- White Fly- Leaf Blight) [5]

[13] T. Joachims, B. Scholkopf and C. Burges and A. Smola (ed.), MIT-Press, SVM-Light Software package Available at <http://svmlight.joachims.org> 1999.

REFERENCES

[1] Agrios G.N., Plant Pathology, 4th Edition. Academic Press, 1997.

[2] Campbell C.L., and Madden L.V., Introduction to Plant Disease Epidemiology. John Wiley & Sons, New York City, 1990.

[3] Sonka M., Halvák V., and Boyle R., "Image Processing Analysis and Machine Vision", Brooks/Cole Publishing Company", USA, 1999.

[4] MATLAB Image Processing Toolbox User's Guide Version 3, Mathworks Inc., www.mathworks.com 2002.

[5] El-Helly.M "Image Analysis Based Interface for Diagnostic Expert Systems". Ph.d dissertation. Faculty of Computer and Information Cairo University, (2004).

[6] Holland, J.H." Adaptation In Natural and Artificial Systems, MIT Press" (1975)

[7] Spišiak M., Kozák Š. "Automatic generation of neural network structures using genetic algorithm" Neural Network world Volume 15, 381-394. (2005)

[8] Lefebvre.C , Fancourt.C " GeneticSolutions Software Package". NeuroDimension. <http://www.nd.com/support/help.htm>. Inc copy right @ (1994-2005).

[9] Lefebvre.C , Fancourt.C " NeuroSolutions Software Package". NeuroDimension. Inc copy right @ 1994-2005. www.nd.com

[10] M. D. Vose and A. H. Wright. "The simple genetic algorithm and the walsh transform": Part I, theory. *Evolutionary Computation*, 6(3):253–273, (1998)

[11] Bernard Scholkopf , Alexander J. Smola Learning with kernels. The MIT Press Cambridge , Massachusetts London, England 2002.

[12] J. Komorowski, L. Polkowski, A. Skowron, "Rough Sets: A Tutorial", Poland, 2004.

AUTHORS

Mohammed Sammany
(M.sc Candidate)

Dep. of Mathematics
Faculty of Science
Cairo University, Giza, Egypt.
sammmanyddr1@hotmail.com



Khaled Saeed Saad Zaghoul
(Undergraduate Student)

Dep. of Computer Engineering
Faculty of Engineering
Cairo University, Giza, Egypt.
khaled.saeed84@gmail.com

