Jarosław Gocławski*, Joanna Sekulska-Nalewajko*

An Image Segmentation Method to Reveal Regions of Stress Manifestation in Stained Cucurbits' Leaves Using LVQ Neural Network

1. Introduction

The severity of foliar diseases and many stress reactions can be quantified by the evaluation of changes in leaf colour. The coloured regions can be naturally visible as primary symptoms in the form of spots or powder, but they often need chemical treatment to be exposed.

Nowadays biologists often register and recognize these changes with the methods of computer image analysis [1, 11, 17]. Popular freeware applications like *NIH*, *Scion Image*, *ImageJ* or commercial applications as *WinFOLIA* and even *Adobe Photoshop* or *Paint Shop Pro* are tools for this purpose. Semi-automatic procedures apply simple grey-level thresholding or image look-up table (LUT) adjustment plus standardization for the detection of stress symptoms [7, 17]. The idea presented in this paper is to replace image LUT adjustments by training an appropriate classifier in colour space of image pixels. Further measurements of coloured areas are based on previously learned parameters of the classifier for a certain population of leaves with similar colour features. The algorithm described in the paper provides the fully automatic detection of leaf blades from scanned images and the recognition of specifically stained regions in *Cucurbitacea* leaves (pumpkin and cucumber). Studied plants were inoculated with the pathogen *Erisiphe cichoracearum* for a series for biological experiments. Among them was the detection of reactive oxygen species (ROS), such as H_2O_2 or O_2^-, which are generated in specific regions of leaves as a response to the infection.

The system of image acquisition, processing and analysis consists of the following modules:

– a standard flatbed scanner of *200/300 dpi* and *24 bit/pixel* resolutions with appropriate firmware software package,

  * Katedra Informatyki Stosowanej, Politechnika Łódzka, Polska

–   laptop with dual core processor *Intel Core (TM) 2 Duo T5750 2 GHz*, *4 GB* RAM and operating system *Windows 7 32 bit*,
–   MATLAB 2008 environment and *Visual Studio Express 2008* C++ compiler plus M/C++ application executing the proposed segmentation method,
–   *Excel* application to preserve the computed results of stressed regions measurements.

The studied leaves have been exposed on a highly saturated colour background, different from colours appearing inside of a leaf lamina (Fig. 1). Red-brownish or blue leaf areas are associated with H_2O_2 or O_2^- generation respectively. The scanned images from these leaves were saved to JPEG files and then loaded into an application executing the segmentation task. This application running in MATLAB environment has been built from the units representing M-language functions or C++ MEX files. In programming some library functions built in *Image Processing Toolbox* and *Neural Network Toolbox* have been used [15, 16].
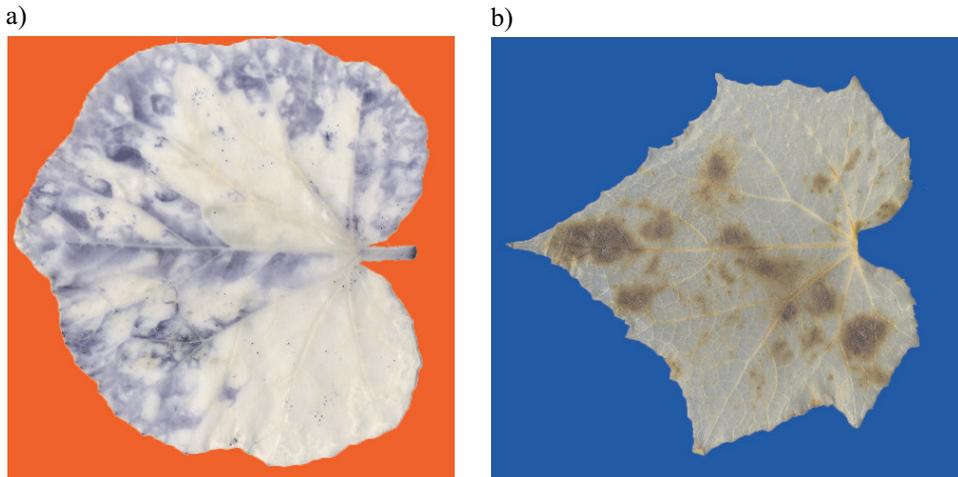
a)   b)

Fig. 1. Example leaf images with stained regions of histochemical reactions:
a) blue regions of O_2^- reaction product; b) red-brown areas of H_2O_2 reaction product

2. Segmentation method

The proposed segmentation method begins with leaf blade detection and pixel features extraction (Fig. 2). In the second stage of the method nonlinear classification is performed by a competitive LVQ artificial neural network (LVQ ANN) [4, 13, 14, 16]. After the segmentation all detected regions with specific staining colours are related to 3 saturation levels to improve the quantification of coloured stress symptoms. The ANN parameters (weights and biases) are estimated by a suitable learning procedure.

2.1. Image preprocessing stage

The preprocessing stage starts with the detection of a leaf blade located on a background with colour close to one of the basic colours $\{R, G, B\}$ and different from colours appearing inside of the leaf. The background colour BC is detected in a square of [50×50] pixels placed in the image upper-left corner, which must be free from leaf tissue. Then the difference of colour component images $I_{BC} - I_G$ can be globally thresholded by Otsu method (Equ. (1)) [10].

$$I_M =\sim T\left(I_{BC} - I_G\right) \tag{1}$$

where:

I_{BC} – image component of background colour $BC \in \{R, B\}$,
I_G – green colour component of the original image I_{RGB},
I_M – binary mask image of a leaf blade,
$\sim$ – the operator of binary image negation,
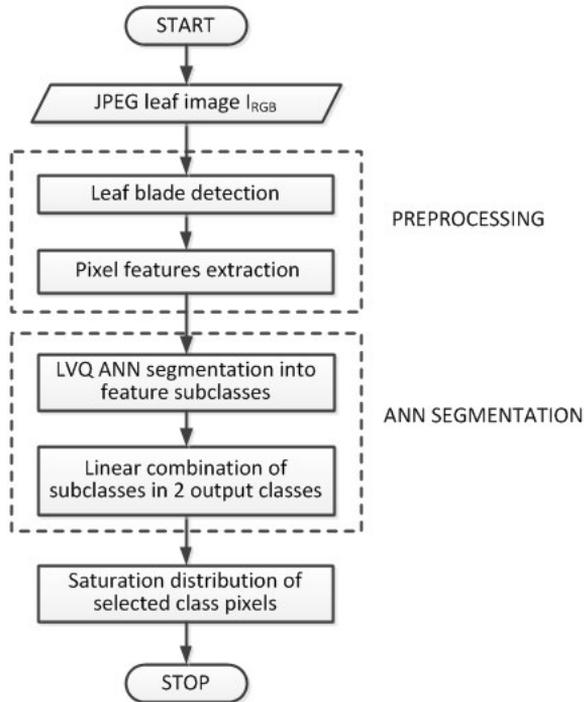T – Otsu thresholding operation.

Fig. 2. The flowchart of leaf image segmentation procedure

The example difference image before thresholding is shown in Figure 3a. The thresholding is typically followed by the elimination of holes (*imfill*) and small residual objects (*bwareaopen*) [15].

The image mask obtained in Equ. (1) and combined with original image is shown in Figure 3b as a white contour overlapped on leaf edges.
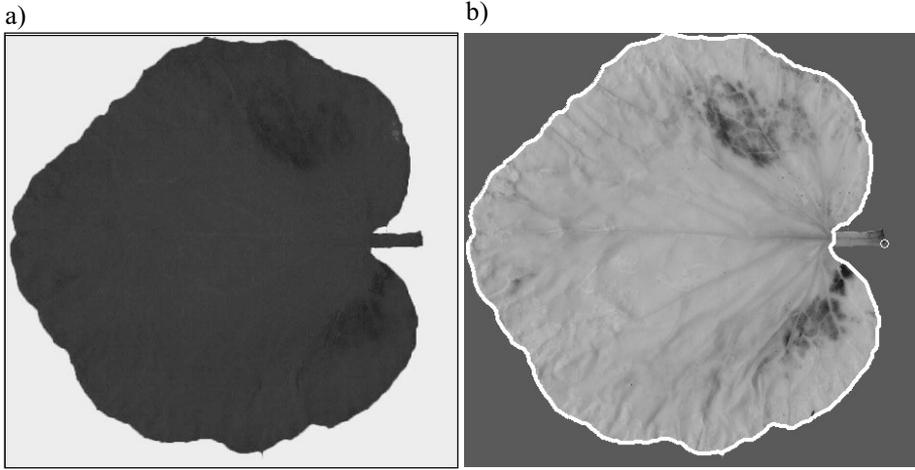
a)   b)

Fig. 3. The example images of: a) $I_{BC} - I_G$ colour component difference expressed in grey levels; b) leaf with the overlapped white contour of detected leaf mask. Small circle in the right image – detected petiole tip

A leaf petiole should be excluded from the leaf mask of I_M, because dye concentration in this part of leaf is not considered by specialists. The method of petiole elimination has been described in details in other paper by the same authors actually prepared for publication.

The visual examination of leaf images leads to the conclusion that the stained regions differ from intact parts of leaves by colour hue H and saturation S. Hence image pixels can be described in the two dimensional feature space of $\mathbf{p} = [p_H, p_S]^T$ vectors of pixel data. The I_H, I_S images are obtained by the MATLAB function [15]:

$$\{I_H, I_S\} = rgb2hsv(I_{RGB})$$ (2)

where:

I_{RGB} – original leaf image,
I_H, I_S – hue and saturation image components.

The $\mathbf{p}$ vector components p_H, p_S are provided to the input of neural network after reading them columnwise (MATLAB specificity) from the images I_H, I_S respectively, under the leaf blade mask of I_M.

2.2. Neural network classifier

The observation of leaf surfaces in the tested population leads to the conclusion, that several pixel groups can be distinguished, each with similar H, S values. The non

specifically stained leaf areas without chlorophyll are whitish and semi–translucent outside of vein system. In these parts of leaf background colours filtered by leaf tissue can be visible. Additionally the uncleaned reminds of chlorophyll appear as greenish spots. The areas of ROS reaction products often have a wide range of saturations or hues, in particular those which are blue (Fig. 1). The groups of pixels with similar H, S features can represent the parts of classes (subclasses) to be identified. They must be finally merged in two output classes: pixels stained in a histochemical reaction and other pixels from intact leaf parts. We cannot be certain that the final classes are linearly separable. The matrices of pixel feature vectors $[p_H, p_S]^T$ can have millions of elements for typical leaf images of 1000×1000 pixels. Therefore only a limited subset of points is taken at time-critical segmentation procedures (as learning). The subset is obtained by dividing the feature images I_H, I_S into squares of $[L×L]$ pixels and the random selection of one pixel from each square. Assuming $L = 8$, the number of vector feature data used for training varies approximately between 6000 and 8000 items. It depends on a leaf blade area in pixels of the trained image and the random selection of these pixels in the $[L×L]$ squares.

Based on the above considerations *Linear Vector Quantization* (LVQ) neural network has been proposed to solve the identification problem of specifically stained leaf regions [11]. The applied network model is shown in Figure 4.
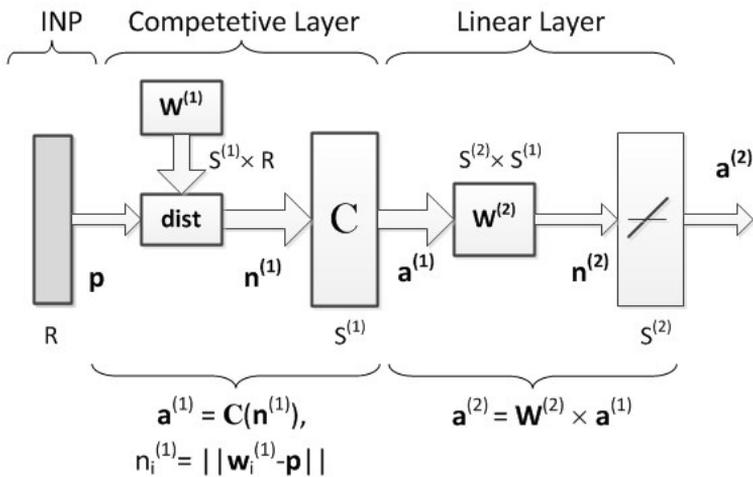

Fig. 4. The model of the neural network proposed for leaf's image segmentation, $\mathbf{W}^{(1)}[S^{(1)}×R]$ – matrix of the first layer, $\mathbf{W}^{(2)}[S^{(2)}× S^{(1)}]$ – matrix of the second layer, C – nonlinear competition module, $\mathbf{p}[R×1]$ – input data vector, $\mathbf{a}^{(1)}[S^{(1)} ×1]$ – output vector of first layer, $\mathbf{a}^{(2)}[S^{(2)} ×1]$ – output vector of second layer, $dist(\mathbf{W}^{(1)},\mathbf{p})$ – the module evaluating distances from the feature vector $\mathbf{p}$ to each vector of neuron weights in $\mathbf{W}^{(1)}$

The distance function *dist* together with the competition function C are active both at a learning process as well as at classification. The function C selects the single neuron

defined by the vector $\mathbf{w}_i^{(1)}[R{\times}1]$, $i \in \{1, ..., S^{(1)}\}$ transposed from the i-th row of matrix $\mathbf{W}^{(1)}$, which is closest to the input feature vector $\mathbf{p}[R{\times}1]$ according to the rule:

$$a_i^{(1)} = C\left(n_i^{(1)}\right) = \begin{cases} 1 & \text{if} \quad i = \arg\min_k \left\| \mathbf{w}_k^{(1)} - \mathbf{p} \right\| \\ 0 & \text{otherwise} \end{cases}, \quad i, k \in \{1, \ldots, S^{(1)}\} \tag{3}$$

where:

$n_i^{(1)}$, $a_i^{(1)}$ – i-th neuron output before and after competition respectively,

$\| \cdot \|$ – the Euclidean norm of the vector difference $\mathbf{w}_k^{(1)} - \mathbf{p}$.

The matrix $\mathbf{W}^{(2)}[S^{(2)}{\times}S^{(1)}]$ of output layer weights $w_{ij}^{(2)} \in \{0,1\}$ assigns the membership of j-th subclass from the first layer to the i-th output class as $w_{ij}^{(2)}{=}1$. It is assumed that the number of subclasses belonging to a certain output class is proportional to the estimated percentage of output class pixels given by some explicit vector $\mathbf{pc}$. Hence, the weights $w_{ij}^{(2)}$ can be determined explicitly, without learning [16]. After learning stage the pixel classification process follows the rule:

$$\mathbf{A}^{(2)} = \mathbf{W}^{(2)} \times C\left(dist(\mathbf{W}^{(1)}, \mathbf{P})\right)$$

$$dist(\mathbf{W}^{(1)}, \mathbf{p}_q) = \left[\left\| \mathbf{w}_1^{(1)} - \mathbf{p}_q \right\|, \cdots, \left\| \mathbf{w}_{S^{(1)}}^{(1)} - \mathbf{p}_q \right\| \right]^T \tag{4}$$

$$\mathbf{P} = [\mathbf{p}_q], \quad q = 1, ..., Q$$

where:

$\mathbf{P}$ – the matrix $[R{\times}Q]$ of Q pixel feature vectors – $\mathbf{p} = [p_H,\ p_S]^T$ (hue and saturation),

$\mathbf{A}^{(2)}$ – the matrix $[S^{(2)}{\times}Q]$ of Q output classes binary vectors – $\mathbf{a}_q^{(2)} = [a_1^{(2)},\ a_2^{(2)}]^T$ (non-specifically and specifically stained pixels),

$\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$ – the weight matrices of LVQ neural network ,

C – the symbol of subclass competition operation,

$dist(\mathbf{W}^{(1)}, \mathbf{p}_q)$ – the vector of Euclidean distances between data vector $\mathbf{p}_q$ and each weight matrix row vector $\mathbf{w}_i^{(1)}$.

Learning vector quantization is a method for training competitive layers in a supervised manner, which needs template targets, here in the form of binary images I_T (Fig. 5). They are manually prepared by specialists by assigning '1' to the pixels of ROS stained colours and '0' otherwise. Having the first layer output vector $\mathbf{a}^{(1)}$ with only one nonzero value and knowing a'priori the weight matrix $\mathbf{W}^{(2)}$ it is possible to map output template vector to the first layer output as given in Equ. (5).

$$\mathbf{T}^{(1)} = \left(\mathbf{W}^{(2)}\right)^T \times \mathbf{T} \tag{5}$$

where:

$\mathbf{T}$ – $[S^{(2)} \times Q]$ the matrix of template binary data vectors from I_T read columnwise, under the mask of I_M,

$\mathbf{T}^{(1)}$ – $[S^{(1)} \times Q]$ the matrix of template binary data vectors mapped to the first layer output.

For a certain feature vector $\mathbf{p}(q)$ (or $\mathbf{p}_q$) the weights of winning neuron $\mathbf{w}_{i*}^{(1)}$ are moved towards $\mathbf{p}$ or away from $\mathbf{p}$ proportionally to the learning rate α, depending on the coincidence of competition output $\mathbf{a}^{(1)}(q)$ with the template vector $\mathbf{t}^{(1)}(q)$ for the first layer (Equ. (6)).

$$\mathbf{w}_{i*}^{(1)}(q) = \begin{cases} \mathbf{w}_{i*}^{(1)}(q-1) + \alpha \cdot \Delta\mathbf{w} & \text{if} \quad a_{i*}^{(1)}(q) = t_{i*}^{(1)}(q) = 1 \\ \mathbf{w}_{i*}^{(1)}(q-1) - \alpha \cdot \Delta\mathbf{w} & \text{if} \quad a_{i*}^{(1)}(q) \neq t_{i*}^{(1)}(q) \end{cases} \tag{6}$$

$$\Delta\mathbf{w} = \mathbf{p}(q) - \mathbf{w}_{i*}^{(1)}(q-1), \quad q = 1, K, \ldots, Q$$

where:

q, Q – the index of pixel feature vector and the number of vectors respectively,

α – learning rate,

$a_{i*}^{(1)}(q)$ – the winner neuron element $i*$ of q-th first layer output vector,

$t_{i*}^{(1)}(q)$ – the winner neuron element $i*$ of q-th first layer template vector.

In the considered algorithm the weights of $\mathbf{W}^{(1)}$ are modified incrementally and the learning rate α after each epoch is slightly decreased by $\Delta\alpha$ to some limit. The proposed learning process has been equipped with two stopping criteria (Equ. (7)):

1) achieving the acceptable level of learning error $e < \varepsilon$ as the averaged maximum norm of the differences between output and template data,
2) exceeding the limited number of epochs ep_{max}.

$$e = \frac{1}{Q} \sum_{q=1}^{Q} \left\| \mathbf{W}^{(2)} \times C\left(dist(\mathbf{W}^{(1)}, \mathbf{p}_q)\right) - \mathbf{t}_q \right\|_{\infty} < \varepsilon \tag{7}$$

$$ep > ep_{max}$$

where:

$C(\mathbf{n}_q)$ – the q-th vector output from the competition function C,

$\mathbf{t}_q$ – the q-th template vector at the network output,

$\mathbf{p}_q$ – the q-th input feature vector $[R \times 1]$,

$\mathbf{W}^{(1)}$ – the weight matrix $[S^{(1)} \times R]$ of the network first layer,

$\mathbf{W}^{(2)}$ – the weight matrix $[S^{(2)} \times S^{(1)}]$ of the network output layer.

Each column of $\mathbf{W}^{(2)}$ represents the mapping of a subclass into the set of output classes, each row shows the membership of subclasses to one output class expressed by the entry value of "1".

When the second stopping condition in Equ. (7) is fulfilled and the learning error e is still greater than the required ε value then LVQ2 correction is applied [5, 16]. It corrects only each two vectors $\mathbf{w}_{i*}^{(1)}$ and $\mathbf{w}_{j*}^{(1)}$ of layer 1 that are closest to the input vector $\mathbf{p}$, if one belongs to the correct class, the other belongs to a wrong class and the input falls into a „window" near the midplane of the two vectors. Such an additional procedure allows a weight row vectors to be moved closer to the input, so the results are more robust.
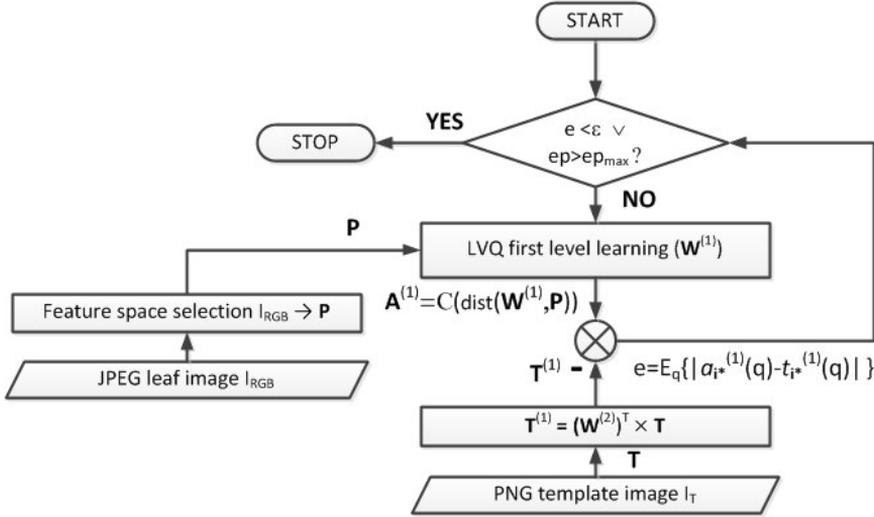

Fig. 5. The flowchart of neural network learning procedure, e, ε – the learning error and its limit, $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$ – weight matrices of the first and second network layers, $\mathbf{P}[R \times Q]$ – feature data matrix, $\mathbf{t}_q^{(1)}$ – data template vector mapped to the first network layer, $dist(\mathbf{W}^{(1)}, \mathbf{p}_q)$ – Euclidean distances between data vector $\mathbf{p}_q$ and each weight matrix row vector $\mathbf{w}_i^{(1)}$, $\mathbf{A}^{(1)}[S^{(1)} \times Q]$ – the array of Q vectors $\mathbf{a}_q^{(1)} = C(dist(\mathbf{W}^{(1)}, \mathbf{p}_q))$, $i*$ – the index of winner neuron element of the vector $\mathbf{a}^{(1)}$

2.3. LVQ Network validation

To estimate network generalisation error 5-fold cross-validation has been performed [2, 12, 16] for the array $\mathbf{P}[R \times Q]$ of feature data vectors from several leaf images with classification templates. The folding routine implements the MATLAB function *crossvalind* from *Bioinformatics Toolbox*, which returns randomly generated indices for the K-fold cross-validation of Q data vectors $\mathbf{p}$ (Equ. (8)).

$$\mathbf{ind} = crossvalind('Kfold', Q, K), \quad ind(i) \in \{1, ..., K\} \tag{8}$$

where $\mathbf{ind}$ – the vector $[Q \times 1]$ of randomly selected $K=5$ indexes.

For the classified data cross-validation confusion matrix has been computed [3], which contains separate counts of all possible classification results for two output classes, the one with the ROS stained leaf pixels and the second with other pixels.

The validation results have been described in the form of 3 different error rates based on the confusion matrix [6, 8]. The matrix is evaluated by counting different outcomes at the comparison of current classification outputs $\mathbf{a}^{(2)}$ with the corresponding template data $\mathbf{t}$ for each pixel selected under the leaf blade mask. The considered error rates: FP – false positive rate, FN – false negative rate, ER – classification error rate, are defined as in Equ. (9).

$$FP = \frac{FPF}{FPF + TNF}, \quad FN = \frac{FNF}{TPF + FNF}, \quad ER = \frac{FNF + FPF}{\sum F} \tag{9}$$

where:

FPF, FNF – the counts of false positive and false negative pixel data fractions respectively,

$\sum F$ – total count of fractions $FPF+FNF+TPF+TNF$ where TPF – true positive fraction and TNF – true negative fraction.

3. Results and discussion

At the preprocessing stage the segmentation algorithm uses built in functions from *Image Processing Toolbox*. The functions of LVQ1 and LVQ2 training have been developed in C++ and stored in MEX files to accelerate their execution. The algorithm has the following parameters adjusted experimentally to obtain good segmentation performance:

– $L = 8$ – the size of pixel data sampling grid applied to the images I_H, I_S,
– $S^{(1)} = 8$ – number of subclasses (output neurons of the first layer) (Fig. 4),
– $\alpha = 0.1$ (0.02) – learning rate at LVQ1 (LVQ2) training (Fig. (5)) modified according to the rule: $\alpha(ep+1) = \max(\alpha_{min}, \alpha(ep) - \Delta\alpha)$, $\alpha_{min} = 0.001$, $\Delta\alpha = 0.02 \cdot \alpha(1)$, where ep – epoch number,
– $ep_{max} = 200$ – maximum number of epochs at each training,
– $\varepsilon = 0.01$ – stopping criterion for training.

The proposed segmentation method was performed on 12 images of single leaves with the stained ROS accumulation regions. The images were read from JPEG files, where they had been stored after scanning. Two images – one with red-brown spots and one with blue spots shown in Figures 1a and 1b respectively were chosen for learning. The intuitive selection criteria of representative images were:

– wider ranges of hue and saturation features in comparison with other images of the same type,
– the colours of ROS reaction product seeming to be easily separable from other colours of stained leaf tissue.
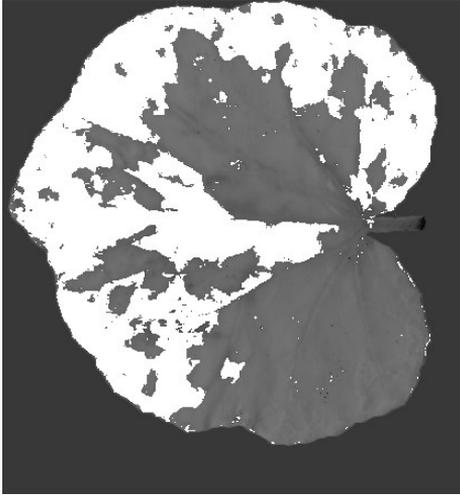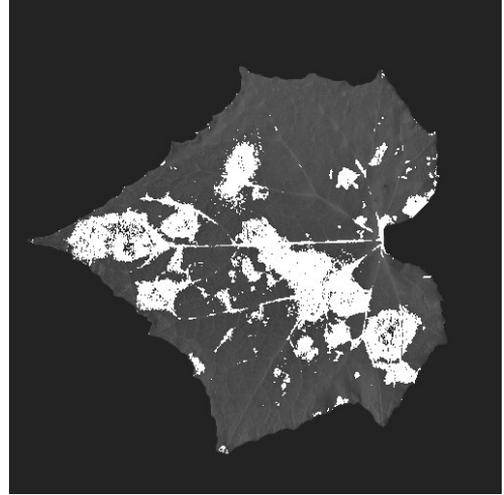
a)   b)

Fig. 6. The example results of image segmentation – overlapped white areas correspond
to anti-stress reaction regions: a) for pumpkin leaf; b) for cucumber leaf

The template images used for network's supervised learning are prepared using colour
masking operations as well as ordinary mask, pencil and eraser tools to original images with
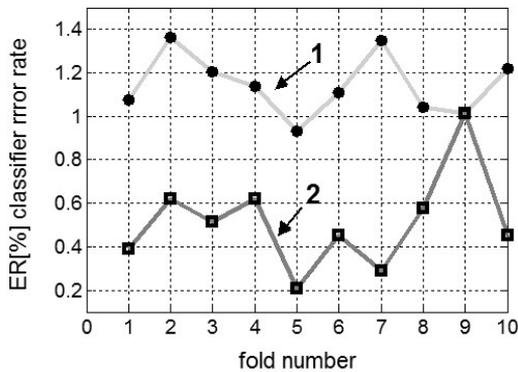the help of *Corel Photo Paint* application.

Fig. 7. The *ER* error plots for two reference images 1 and 2 from
Figures 6a, b during double run of 5-fold validation

The example results of learning and classification for these images are visible as
overlapped regions of white pixels covering original colours from stained ROS reaction
products (Figs. 6a,b). The errors of 5-fold cross-validation executed twice for each of the
selected two images are shown as plots in Figure 7.

Several extra templates were also prepared for testing the classifier with the new pixel data of images not participating in a learning process previously. The misclassification error rates *ER* calculated in this case are collected in Table 1. The traditional segmentation based on luminance thresholding, mentioned previously in section 1, was also developed to compare it with the proposed method. For simplicity it was written in MATLAB environment, without general purpose application like *Scion Image*. The misclassification rates of this method computed for the same images are put in the last column of Table 1. Table 2 lists the execution times at the classification stage for different methods and the images from Table 1.

Table 1
Error rates for an example set of images using the proposed LVQ
classification or standard luminance thresholding

Image number	Classifier method			Standard method
	False Positive Error Rate FP [%]	False Negative Error Rate FN [%]	Total Error Rate ER [%]	Total Error Rate ER' [%]
1	1.96	0.01	1.12	15.61
2	1.84	0.01	1.45	7.54
3	1.66	0.18	1.47	7.65
4	2.48	0.19	2.26	6.15
5	0.00	0.04	0.01	7.77
6	0.37	6.50	0.95	7.89
7	0.65	6.54	0.76	21.59
8	0.35	2.03	0.88	12.52
9	0.15	0.53	0.23	8.19
10	0.15	2.32	0.63	3.71
11[*]	0.80	15.42	1.48	1.89
12[*]	1.40	6.33	1.77	8.38

* image sample grid size $L = 4$, otherwise $L = 8$ (LVQ only)

It can be easily verified in the Table 1 that the error rates of traditional approach are on average 9 times greater than those achieved with the new proposed method based on classification in colour space. Exploiting image colour features and the recognition of nonlinear class boundaries are the main advantages of the new segmentation method. One of the new method's drawbacks is the dependency of execution time and convergence speed

from the starting point to the network weights optimisation [8]. For studied examples the global convergence of classification process was observed in each case, but unless it can be guaranteed, the global optimization methods independent on a starting point need to be applied before, like genetic algorithm or simulated annealing [8, 9]. The other drawback of LVQ segmentation is the requirement to have image patterns (templates) for training the weights of $\mathbf{W}^{(1)}$.

Another "sensitive" point of LVQ method is dividing the number of subclasses $S^{(1)}$ mapped to the output classes ($\mathbf{W}^{(2)}$ matrix weights) in an arbitrary way or proportionally to the output classes percentage. For this reason some low cardinality subclasses can be inaccurately merged with other subclasses at the first layer output.

Table 2
Example execution times of the proposed LVQ classification
and standard luminance thresholding for the images from Table 1

Image number	Classifier method		Standard method
	Learning time [s]	Simulation time [s]	Execution time [s]
1	4.54	1.34	0.02
2	–	1.54	0.05
3	–	1.44	0.05
4	–	1.01	0.05
5	–	1.16	0.07
6	0.02	1.11	0.04
7	–	1.20	0.05
8	–	0.94	0.03
9	–	1.04	0.04
10	–	1.19	0.05
11[*]	10.39	0.89	0.04
12[*]	–	0.96	0.04

* image sample grid size $L = 4$, otherwise $L = 8$ (LVQ only)

PCA transform applied to the LVQ network input reduces data dimensionality just above the 10–15% limit of second component variance and significantly increases classifier

error rate (even up to 20%). For this reason it has not been applied here. The accuracy achieved for the automatic segmentation has been accepted by biologists. In the future the results of image analysis will be compared with biochemical measurement methods of pathological changes in leaf tissue.

Acknowledgements

References

[1] Goodwin P.H., Hsiang T., *Quantification of fungal infection of leaves with digital images and Scion Image software*. Methods in Molecular Biology, vol. 638, 2010, 125–35.

[2] Kohavi R., *A study of cross-validation and bootstrap for accuracy estimation and model selection*. Proc. of International Joint Conference on AI., 1995, 1137–1145.

[3] Kohavi R., Provost C., *Glossary of Terms. Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*. Machine Leanring, vol. 30, No. 2–3, 1998.

[4] Kohonen T., *Self-Organization and Associative Memory*. 2nd ed., Springer-Verlag, Berlin 1987.

[5] Kohonen T., *Improved versions of learning vector quantization*. Proc. of the International Joint Conference on Neural Networks (IJCNN), vol. 1, 1990, 545–550.

[6] Krzyśko M., Wołyński W., Górecki T., Skorzybut M., *Systemy uczące się. Rozpoznawanie wzorców analiza skupień i redukcja wymiarowości*. WNT, Warszawa 2008.

[7] Kwack M.S., Kim E.N., Lee H., Kim J., Chun S., Kim K.D., *Digital image analysis to measure lesion area of cucumber anthracnose by Colletotrichum orbiculare*. Vol. 71, No. 6, 2005, 418–421.

[8] Masters T., *Practical Neural Network Recipies in C++*. Academic Press Inc., San Diego 1993.

[9] Osowski S., *Sieci neuronowe do przetwarzania informacji*. Oficyna Wydawnicza PW, Warszawa 2006.

[10] Otsu N., *A Threshold Selection Method from Gray-Level Histograms*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, No. 1, 1979, 62–66.

[11] Soukupová J., Albrechtová J., *Image Analysis – Tool for Quantification of Histochemical Detections of Phenolic Compounds, Lignin and Peroxidases in Needles of Norway Spruce*. Biologia Plantarum, vol. 46, No. 4, 2003, 595–601.

[12] Stąpor K., *Automatyczna klasyfikacja obiektów*. EXIT, Warszawa 2005.

[13] Tabrizi P.R., Rezatofighi S.H., Yazdanpanah M.J., *Using PCA and LVQ neural network for automatic recognition of five types of white blood cells*. Engineering in Medicine and Biology Society (EMBC), Annual International Conference of the IEEE , Aug. 31 2010 – Sept. 4 2010, Buenos Aires, 5593–5596.

[14] Tang Q., Liu B., Chen Y., Zhou X., Ding J., *Application of LVQ neural network combined with the genetic algorithm in acoustic seafloor classification*. Chinese Journal of Geophysics, vol. 50, No. 1, 2007, 291–298.

[15] The Mathworks Inc., *Image processing toolbox user's guide.* URL http://www.mathworks.com/help/toolbox/images/image_product_page.html.

[16] The Mathworks Inc., *Neural networks toolbox user's guide.* URL http://www.mathworks.com/help/toolbox/nnet/nnet_product_page.html.

[17] Wijekoon C.P., Goodwin P.H., Hsiang T., *Quantifying fungal infection of plant leaves by digital image analysis using Scion Image software*. Journal of Microbiological Methods, vol. 74, 2008, 94–101.