Int. J. Appl. Math. Comput. Sci., 2012, Vol. 22, No. 3, 669–684
DOI: 10.2478/v10006-012-0050-5

NEURAL NETWORK SEGMENTATION OF IMAGES FROM STAINED CUCURBITS LEAVES WITH COLOUR SYMPTOMS OF BIOTIC AND ABIOTIC STRESSES

Jarosław GOCŁAWSKI [*], Joanna SEKULSKA-NALEWAJKO [*], Elżbieta KUŹNIAK [**]

[*] Institute of Applied Computer Science
Łódź University of Technology, Stefanowskiego 18/22, 90-924 Łódź, Poland
e-mail: {jgoclaw,jsekulska}@kis.p.lodz.pl

[**] Department of Plant Physiology and Biochemistry
University of Łódź, Banacha 12/16, 90-237 Łódź, Poland
e-mail: elkuz@biol.uni.lodz.pl

The increased production of Reactive Oxygen Species (ROS) in plant leaf tissues is a hallmark of a plant's reaction to various environmental stresses. This paper describes an automatic segmentation method for scanned images of cucurbits leaves stained to visualise ROS accumulation sites featured by specific colour hues and intensities. The leaves placed separately in the scanner view field on a colour background are extracted by thresholding in the RGB colour space, then cleaned from petioles to obtain a leaf blade mask. The second stage of the method consists in the classification of within mask pixels in a hue-saturation plane using two classes, determined by leaf regions with and without colour products of the ROS reaction. At this stage a two-layer, hybrid artificial neural network is applied with the first layer as a self-organising Kohonen type network and a linear perceptron output layer (counter propagation network type). The WTA-based, fast competitive learning of the first layer was improved to increase clustering reliability. Widrow–Hoff supervised training used at the output layer utilises manually labelled patterns prepared from training images. The generalisation ability of the network model has been verified by K-fold cross-validation. The method significantly accelerates the measurement of leaf regions containing the ROS reaction colour products and improves measurement accuracy.

Keywords: image segmentation, colour space, morphological processing, image thresholding, artificial neural network, WTA learning, Widrow–Hoff learning, *Cucurbita* species, plant stress, ROS detection.

1. Introduction

Recently, leaf image analysis systems have been used for quantifying stress symptoms. They can be alternative accurate methods for subjective visual assessment (James, 1971) or biochemical methods (Cheeseman, 2006), which do not allow obtaining detailed data of stress symptom distribution in leaves. A lot of automated methods of leaf analysis are based on common software packages, e.g., quantifying infection of fungus *Colletotrichum destructivum* in *Nicotiana benthamiana* and other plant species was done by the freely available *Scion Image* application (Wijekoon *et al.*, 2008). This popular, general purpose image analyser requires time-consuming operations or substantial knowledge for appropriate macros building to extract the diseased regions. In that case, the authors considered only grey-level images to identify bright leaf blade profiles and dark diseased areas inside of them by simple, global thresholdings.

Symptoms of infection include leaf discolouration (usually white or dark), thus making it possible to direct quantification. However, a common feature of plant response to biotic and abiotic stresses is the increased production of Reactive Oxygen Species (ROS), which are visible in leaf tissues as specific colour regions only after histochemical detection. These histochemical methods allow obtaining detailed data on *in situ* ROS distribution and accumulation in different leaf parts, thus enabling better comparison of various treatments. However, ROS detected histochemically have been often assessed semi-quantitative by visual estimation (Huang *et al.*, 2010). Recently, image analysis methods have been developed and applied to the quantification of the products of ROS-mediated histochemical reactions in plant tissues (Soukupova and Albrechtova, 2003).

The authors of this paper have been faced with the identification of the sites of ROS generation in pumpkin and cucumber leaves subjected to abiotic stresses (drought and salinity) and infected with a pathogen. The subjects of quantification were regions of accumulation of two ROS species: superoxide anion radical and hydrogen peroxide, visible after leaf staining as blue or red-brown spots, respectively. In histochemically stained and then cleared (chlorophyll free) leaves, these regions differ from the intact leaf tissues by colour hue and saturation values. The colour features and multidimensionality of the feature space suggest using the colour space instead of grey-levels and a formal classifier, e.g., with an ANN (*Artificial Neural Network*) instead of thresholding. So far, LVQ (*Linear Vector Quantization*) type neural networks have been successfully applied to many classification problems like blood cell recognition (Tabrizi *et al.*, 2010) or seafloor acoustic images segmentation (Tang *et al.*, 2007). The authors propose the use of a slightly modified self-clustering WTA (*Winner Takes All*) network (Kohonen 1990; 2001) concatenated with a linear perceptron layer type (Widrow *et al.*, 1988; Hagan *et al.*, 2009). Using a sufficient number of clusters, the network can recognise all visible leaf staining colours and then combine them in two groups: intact blade areas and the concentration regions of the stress reaction. For a network of such a type, these groups do not need to be assumed as linearly separated, which is not guaranteed in the examined populations. The capabilities of using Kohonen networks in image segmentation in the $L^*u^*v^*$ colour space have already been studied (Ong *et al.*, 2002).

2. Plant material and leaf preparing

The material for image analysis consisted of the leaves of cucumber and pumpkin plants cultivated under growth chamber conditions. A set of five-week-old plants was used for abiotic stress treatments. Plants were subjected to water deficit (drought stress) or irrigated with $50\ mM$ NaCl (salt stress) for seven days. The second group of plants were not treated with abiotic factors. Then each group was divided into two subsets: control and inoculated with the pathogenic fungus *Erysiphe cichoracearum*. The plants were analyzed five days after inoculation. Detached leaves were examined according to Unger *et al.* (2005) for superoxide anion radical (O_2^-) visualisation and to Thordal-Christensen *et al.* (1997) for hydrogen peroxide (H_2O_2) detection. After staining and clearing in ethanol, leaves became almost white, as a result of chlorophyll removal, and colour products of histochemical reactions of O_2^- and H_2O_2 were visible as blue and red-brown spots, respectively.

3. Image preprocessing

The stained leaf images have been acquired in a simple computer measurement system consisting of a standard desktop scanner connected to a dual core PC with a 32-bit Windows 7 operating system. The leaf blade images stored in JPEG files are subject to classification designed by the authors to detect stress response regions. Both image preprocessing and pixel classification methods have been developed in the MATLAB environment and implemented in the form of MATLAB functions as well as C++ functions contained in MEX files (The Mathworks Inc., 2011a). The purposes of preprocessing are to extract a leaf blade and to eliminate a leaf petiole. The mechanical cut-off of the petiole is often difficult to do without injuring or even partially damaging the leaf tissue.

3.1. Leaf blade extraction. To simplify the separation of a leaf blade from the image background, it is assumed that the background has a highly saturated uniform colour different from any colour appearing inside of the stained leaf blade. Blue and red backgrounds made of plastic sheets have been applied respectively for leaves with red-brown and blue spots inside. The images are typically scanned at the resolution of $200\ dpi$ and colour depth $24\ bit/px$. In each of them, at least a $50\ px$ background margin must be preserved. The main image processing steps of leaf blade extraction are depicted in Fig. 1 as a flow diagram. The image background colour BC is tested inside of the rectangle window W placed in the upper-left corner of the background margin. With the assumptions above, this colour indicates one of the two possible colour groups of the ROS reaction. It is estimated as

$$BC = \begin{cases} B & \text{if} \quad E\{I_B(W)\} > k \cdot E\{I_R(W)\}, \\ R & \text{otherwise}, \end{cases} \tag{1}$$

where $E\{\cdot\}$ signifies the expected value, $I_R(W),\ I_B(W)$ mean the red and blue components of the true colour image I_{RGB} in the rectangular window $W[50 \times 50]\ px$, and $k = 1.5$ is an arbitrarily chosen constant value. The difference of image colour components I_{BC} and I_G (green) exposes a highly saturated background colour BC and makes the resulting grey-level image independent of potential background intensity variations. The global thresholding of $I_{BC} - I_G$ provides the inversion of a binary leaf blade mask image I_M:

$$I_M = \sim \mathbb{T}(I_{BC} - I_G), \tag{2}$$

where $\mathbb{T}$ denotes the Otsu thresholding operator (Otsu, 1979). After the thresholding (Eqn. (2)), the binary image I_M can be considered the set of white objects $\{O_i\}$ of 8-adjacent pixels on a black background. All potential "holes" in the leaf mask object should be removed by a flood-fill operation on 4-adjacent background pixels. In the MATLAB environment it is represented by the function *imfill* (The Mathworks Inc., 2011b) equivalent to the operation in Eqn. (3):

$$\forall\ (x, y) \in ext(\overline{O}_F), \quad I_M(x, y) = 1, \tag{3}$$

where $ext(\overline{O}_F)$ means the exterior of the 4-adjacent black pixels object $\overline{O}_F \subset\sim I_M$ including the image frame. Only one object O_m from the set $\{O_i\}$ with the greatest area A_m is preserved as a leaf blade mask. Objects with smaller areas are spurious data and must be eliminated as

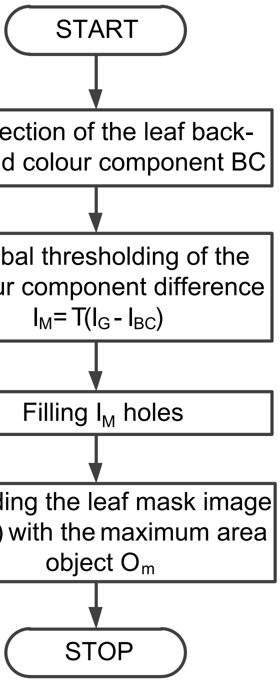

START
Detection of the leaf background colour component BC
Global thresholding of the colour component difference
I_M= T(I_G - I_BC)
Filling I_M holes
Building the leaf mask image I_M(O_m) with the maximum area object O_m
STOP

Fig. 1. Flow diagram of the algorithm providing the binary mask of a leaf image.

$$(x, y) \in O\ \wedge\ A_O < A_m\ \Rightarrow\ I_M(x, y) = 0. \tag{4}$$

3.2. Leaf petiole elimination.

In the tested population of stained leaves a petiole is usually the place of a high concentration of dye, but biologists ignore this part of leaf during visual estimation of a plant's stress. Therefore, in the presented algorithm the petiole is removed from the leaf mask image I_M by the method shown in Fig. 4. Based on the observation of leaf mask contours, the authors noticed that the petiole is always the most protruding part of every leaf. They formulated the hypothesis that the petiole tip can be distinguished as a leaf contour point with the highest curvature value. The hypothesis was successfully verified in the tested population of leaves. The leaf contour $C(O_m)$ has been found by a left-most search for leaf mask edge points in eight directions (Gonzalez and Woods, 2008),

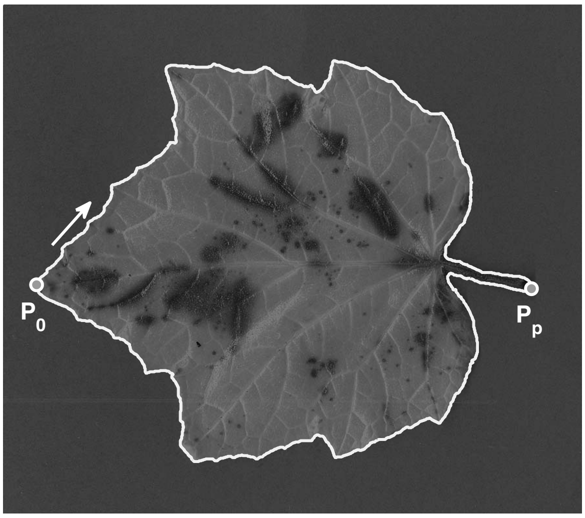


Fig. 2. Example leaf image with the white contour overlapped on the edge of a leaf mask, P_0: the contour starting pixel, P_p: petiole tip pixel.

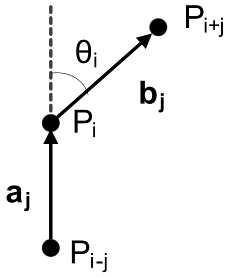


Fig. 3. Description of contour points and vectors for curvature computation. Letter symbols explained in the text.

$$C(O_m) = LML([P_0, \dots, P_{N-1}]). \tag{5}$$

Figure 2 shows an example image with a marked white contour found around the leaf mask according to the rule in Eqn. (5). For each contour point P_i, $i \in [0, \dots, N-1]$, the local curvature is represented as the bending angle θ_i computed by (Du Buf and Bayer, 2002)

$$\theta_i = \frac{1}{M}\sum_{j=1}^{M} \arccos\frac{\mathbf{a}_j \cdot \mathbf{b}_j}{|\mathbf{a}_j| \cdot |\mathbf{b}_j|}, \tag{6}$$

where $\mathbf{a}_j = P_{i-j}P_i$ and $\mathbf{b}_j = P_iP_{i+j}$ are vectors as shown in Fig. 3, $\mathbf{a}_j \cdot \mathbf{b}_j$ is the vector inner product and M is the half size of an averaging mask. This mask represents a built-in low-pass filter smoothing the curvature

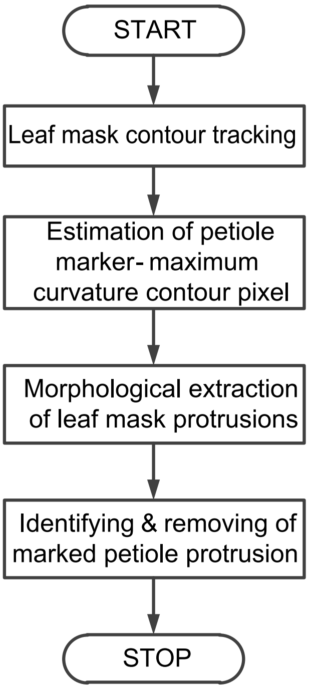

START
Leaf mask contour tracking
Estimation of petiole marker- maximum curvature contour pixel
Morphological extraction of leaf mask protrusions
Identifying & removing of marked petiole protrusion
STOP

Fig. 4. Flow diagram of the algorithm eliminating a leaf petiole.

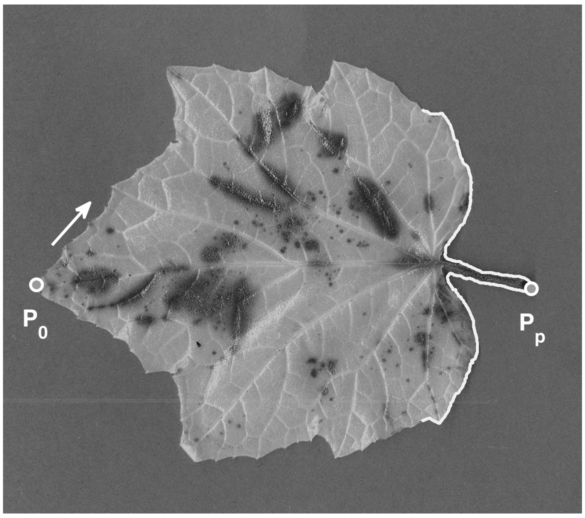


Fig. 5. Example leaf image with the contour part used to calculate the curvature θ_i. P_0: contour starting pixel, P_p: petiole tip pixel.

values θ_i evaluated along a leaf mask contour. This filtering is necessary due to the high sensitivity of θ_i to any contour ripples. The value of M was chosen experimentally in relation to the leaf contour length N as

$$M = 2.5/100 \times N. \tag{7}$$

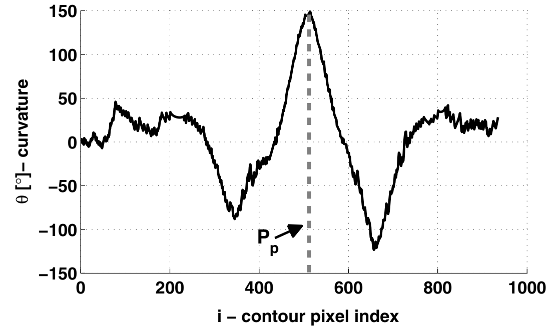


Fig. 6. Example curvature plot $\theta(i)$ along the section of the leaf contour shown in Fig. 5

A proper selection of the value M plays a key role in the detection of petiole protrusion. Each of the scanned leaves is assumed to be placed horizontally in the field of view with its tip on the left side (Fig. 5). Then leaf mask contour tracing begins at the leaf's tip P_0 and runs through the petiole tip P_p located close to half the length of the contour $N/2$. Therefore only the curvature of 30% of the contour pixels around $P_{N/2} \in [P_{i_1}, P_{i_2}]$ is considered. The petiole tip pixel P_p is determined as

$$P_p = \arg\ \max\ \theta_i, \quad i \in [i_1, i_2]. \tag{8}$$

An example curvature plot with the global maximum corresponding to the contour part in Fig. 5 is depicted in Fig. 6. The petiole tip pixel P_p is only used as a marker of the petiole region. This region belongs to the set of leaf mask edge protrusion objects obtained by morphological operations expressed by

$$I'_M = I_M \backslash (I_M \circ S_{R_1}) \ominus S_{R_2}, \tag{9}$$

where S_{R_1} used in opening represents a circular structuring element with the radius R_1, which must be greater than the half of the maximal petiole width. Erosion by S_{R_2} additionally shrinks reminded objects to improve petiole separation from other protrusions. After enumeration (labelling) of these objects (The Mathworks Inc., 2011b), the petiole can be found as the labelled object L_m closest to the point of the maximum curvature P_p (Eqns. (10) and (11)),

$$I_L(\{L_i\}) = LBL(I'_M), \tag{10}$$

$$L_m = \arg\ \min\ \rho_i(L_i, P_p). \tag{11}$$

The leaf mask without a petiole has been evaluated in Eqn. (12) as the logical product of the whole leaf mask I_M and the binary petiole image $\sim \mathbb{T}(I_L(L_m))$ previously dilated by the disk structuring element S_{R_2},

$$I_M = I_M \cap \sim (\mathbb{T}(I_L(L_m)) \oplus S_{R_2}), \tag{12}$$

where $\oplus$ represents a dilation operator.

4. Neural network model design and validation

The second stage of the algorithm involves the classification of leaf blade pixels (inside the mask of image I_M) in the following two groups:

- the sites of ROS generation in leaf tissues visualised after staining,
- the regions of intact leaf tissue with other colours.

4.1. Feature space selection. The observation of stained leaf blade images leads to the conclusion that ROS accumulation areas can be distinguished from other leaf regions by their colour features. Depending on the ROS type (O_2^- or H_2O_2) and chemical agents used in staining, these areas appear as red-brown or blue. The rest of the leaf blade has a background colour mixed with white at the passage through chlorophyll free semi-translucent leaf tissues except for opaque leaf veins. Red-brown stained locations in a tested leaf population have usually high saturation while blue areas can be medium or even low saturated. The low saturated areas are hardly visible as grey levels (Fig. 7). To rate a leaf blade pixel in terms of ROS generation, specialists consider combinations of its colour pigment and vividness. So the image feature space for classification was built from two image colour components matching the above terms: hue H and saturation S. They were obtained by colour space transformation from the RGB to the HSV colour space (Smith, 1978) as

$$\{I_H, I_S, I_V\} = HSV\left(I_{RGB}\right). \tag{13}$$

The traditional hexcone HSV colour model was applied to follow the HSB (HSV) transformation of *Corel Photo Paint 12*, which helps specialists in the manual classification procedure given below (Step 2). This procedure performed by biologists provides pattern results for training and error estimation of the proposed automatic classification. It consists of the following steps listed as commands to a human operator:

1. Load an examined leaf image into the *Corel Photo-Paint* application.
2. Select *Mask → Color mask. . .* and the HSB mode in the popped up colour mask window.
3. Create the mask of stressed colour regions by pointing one or more image pixels with typical ROS colours and by setting their hue and saturation tolerances.
4. Correct the mask with visible floating edges by changing reference pixels and H, S tolerances.
5. Adjust the mask using pencil and eraser tools.
6. Save the binary mask image (*Mask → Paint mask*) to a disk PNG file (*Mask → Save mask to disk. . .*).
7. Calculate the total area of ROS generation sites in physical units using the popular application *ImageJ* and built in image resolutions in X, Y directions.

In the proposed automatic classification of leaf blade regions, manual class labelling is required only for training image data of hue-saturation plane (H, S). To ensure that the expected clusters are closed within one period of the colour hue (not dissected by its limits), both ROS reaction colour types (blue and red-brown) have been located in the lower half of the hue axis H. This has been achieved by applying hue rotations as follows:

$$I'_H(x,y) = \begin{cases} \mathrm{rot}(I_H(x,y), \Delta_1) & \text{if} \quad BC = R, \\ \mathrm{rot}(I_H(x,y), \Delta_2) & \text{if} \quad BC = B, \end{cases} \tag{14}$$

where BC is the background colour as in Eqn. (1), $\Delta_1 = 1/2$ and $\Delta_2 = 1/6$ for $I_H(x,y) \in [0,1]$.

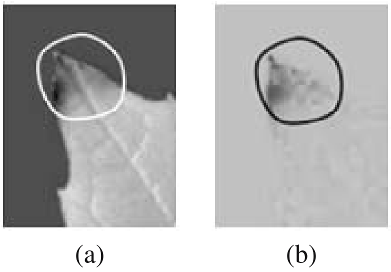


Fig. 7. Example of a blue stained leaf tip in a luminance image (a), a hue image (b). The part of the circled blue region is better distinguished from the background in the hue image.

4.2. Training set preparation. Similarly to other cases of classification, a properly selected training set of input data in the (H, S) space must fulfil two general conditions (Masters, 1993):

- Each of visually distinguished (H, S) colour classes concentrating around particular patterns discussed in Section 4.1 must be included in the training set.
- Within each of these classes, a sufficient statistical variation should also appear and be as large as possible, still mindful of the limited representation.

The whole set of pixel data available for training or classification is naturally divided into subsets building individual images. Additionally, only pixels inside leaf blade masks provide useful data. In the normal mode of classifier activity, manual labelling makes sense only for a single image or at most a few images in the tested population. They must contain most colour hues and saturations existing in the population to classify every image item sufficiently well. Keeping in mind that the network as a whole finally recognises two classes, the very approximate identification of these classes and the statistical deviations of their data have been considered. All examined leaf populations included both strongly and weakly stressed leaf samples (control group), so in all cases a single image with a medium stressed leaf could be found and used as the training set. The training image can be selected intuitively, by a visual assessment of images or using a heuristic formula of training the ability factor f_i proposed in Eqn. (16). The expression for f_i applies the weighted variances of a leaf blade histogram in the (H, S) space. They are computed inside and between two output colour classes c_1 and c_2, coarsely identified before the final classification. It is assumed that the training image with the index

$$i_t = \arg\ \max\ f_i \tag{15}$$

will maximise

$$f_i = k_f \left(\sum_{k=1}^{2} p_i(c_k) s_i(c_k) + k_g\, \sigma_i(c_1, c_2) \right), \tag{16}$$

where

$$s_i(c_k) = \sum_{j=1}^{2} k_j \sum_{q=1}^{Q(i,c_k)} \left(x_{ij}(c_k, q) - \overline{x}_{ij}(c_k) \right)^2 ,$$

$p_i(c_k)$ is the probability of the (H, S) class c_k, $s_i(c_k)$ is a weighted variance inside of the i-th image class c_k, $\sigma_i(c_1, c_2)$ is the i-th image between-class variance, k_f being a scaling constant, k_g is the participation rate of the between-class variance, $x_{ij}(c_k, q)$ is the q-th input data in the j-th dimension of the i-th image for the class c_k, $\overline{x}_{ij}(c_k)$ is the mean data value in the class c_k and the j-th dimension of the image i.

Table 1. Training ability factors f_i for the example population of images with the maximum and minimum values underlined.

Image i	Train ability f_i
1	1.8359
2	2.7222
3	2.5978
4	3.9283
5	1.2117
6	3.6953
7	0.5066
8	2.5903
9	1.5800
10	1.4347

The two classes c_1 and c_2 used in Eqn. (16) were obtained by applying the *kmeans* algorithm (Forgy, 1965) to the input data $\mathbf{x} = [x_1, x_2]$, $x_1 = I_H(p)$, and $x_2 = I_S(p)$ taken from hue and saturation image components I_H, I_S of any pixel p as follows:

$$\{c_1, c_2\} = \text{kmeans}\left(\{\mathbf{x}\}, N_c\right), \tag{17}$$

where $N_c = 2$ is the required number of classes. The computed values f_i for an example population of 10 images are given in Table 1 and two images with extreme f_i are illustrated in Figs. 8(a) and (b). For the data in this table the coefficients in Eqns. (16) and (17) are $k_f = 1/256$, $k_g = 0.1$, $k_1 = 1$, $k_2 = 1.44$. There seems to be no danger of network overfitting in the learning process because of the large number of training feature vectors derived from pixel colours inside of a leaf blade mask. The scanned leaf images typically have a size of about 1000×1000 pixels and a leaf blade occupies about half of the image area. The first network layer proposed in Section 4.3 with two inputs and 8–12 neurons has only 16–24 vector weights. The second layer with one neuron has the same amount of weights as the number of first layer outputs, which gives a total of maximum 36 weights to learn. So the number of data is incomparably larger than four times of the total network weights number recommended by practitioners as the minimum data limit to avoid overfitting (Masters, 1993). Nevertheless the proposed network model is tested for overfitting by K-fold cross-validation described below. When experiments suggest that the selected training image does not represent sufficiently colour subclasses or their variations, giving unaccepted classification errors, further images with the highest f_i factors can be applied to additional training.

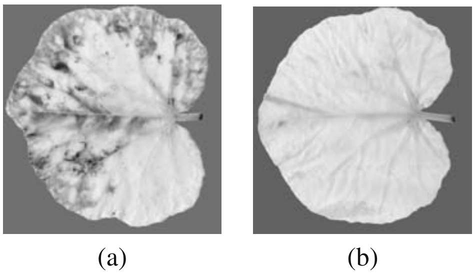


Fig. 8. Selected images corresponding to the minimum and maximum values of f_i from Table 1: the image with $f_{\max}$ featured by the noticeable variances of (H, S) within and between the two basic colour classes c_k, $k = 1, 2$ in Eqn. (17) (a), another leaf image with $f_{\min}$ and the same variances of small values (b). Darker leaf blade regions are blue and bluish-grey, respectively, in the images (a) and (b).

4.3. Structure of a neural network. The proposed model of the classifier applies a two-layer neural network of the counter propagation type (Hecht-Nielsen *et* *al.*, 1987) shown in Fig. 10. The first layer is a self-organising Kohonen network with MWTA (*Modified Winner Takes All*) learning accepting two-dimensional input vector $\mathbf{x} = [x_1, x_2]^T$ of I_H and I_S pixel values masked by I_M. It has been developed because there is no guarantee that the regions of ROS generation and other leaf parts are linearly separable in the (H, S) space and can always be extracted by a single layer network or another linear classifier. Using the modified learning method aims to achieve a more reliable classification result and will be discussed in the following sections. Input image data of hue and saturation physically restricted to the domain $[0, 1]$ are mapped to the range $[-1, 1]$ optimal for the Kohonen network (Kohonen, 1990; Masters, 1993). A prerequisite for network application is to normalise the input data vector to a fixed length. To preserve the original $\mathbf{x}$ data lengths, Z-axis normalisation has been applied as explained in Fig. 9. It expands the original data space to a three-dimensional space with unit length vectors according to the following formulae:

$$x'_i = \frac{x_i}{\sqrt{2}}, \quad i \in \{1, 2\}, \qquad (18)$$
$$x'_3 = \sqrt{1 - \frac{||\mathbf{x}||^2}{2}}.$$

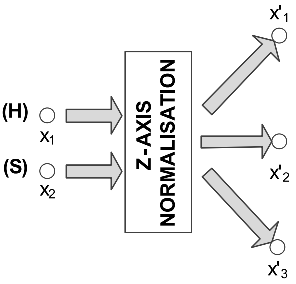


Fig. 9. Z-axis normalisation.

The initial values of weight array $\mathbf{W}^{(1)}$ are rescaled and extended like input data in Eqn. (18). After training the weights are reduced in dimensionality, backscaled and remapped. All components of the input vector are connected to each of the M output neurons $\mathbf{y} = [y_1, \dots, y_M]^T \in [0, 1]^M$ representing the centres of clusters (Fig. 10). The output neuron y_i of the first layer is defined by the weight vector $\mathbf{w}_\mathbf{i}^{(\mathbf{1})} = [w_{i1}^{(1)}, w_{i2}^{(1)}]^T$. The neuron $y_c(\mathbf{w}_c^{(1)})$ whose weights $\mathbf{w}_c^{(1)}$ are closest to the data vector $\mathbf{x}$ becomes the 'winner' (Eqns. (19) and (20)) and is slightly updated towards the input vector by the learning rule given in Eqn. (25),

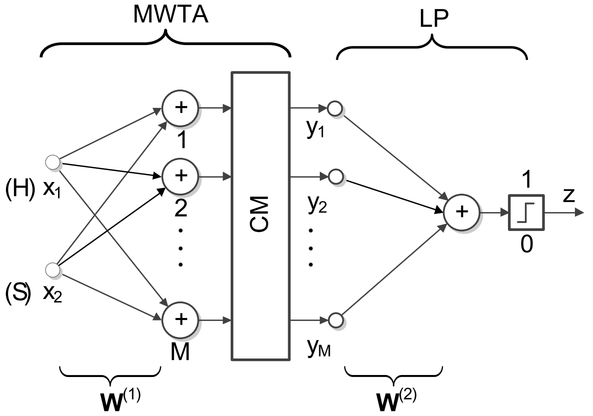


Fig. 10. Structure of a 2-layer neural network applied for the segmentation. M: number of clusters, $\mathbf{W}^{(1)}$,$\mathbf{W}^{(2)}$: weight arrays of the first and second network layer, respectively, CM: data competition module, MWTA: modified WTA layer, LP: linear perceptron layer.

$$c = \arg \min_i \rho(\mathbf{x}, \mathbf{w}_i^{(1)}), \qquad (19)$$

$$y_j = \begin{cases} 1 & \text{if } j = c, \\ 0 & \text{otherwise,} \end{cases} \qquad (20)$$

where ρ is distance between the input $\mathbf{x}$ and the i-th weight vector $\mathbf{w}_\mathbf{i}^{(\mathbf{1})}$, c the index of the 'winner' weight vector closest to the input vector. The second linear perceptron type layer contains one neuron with the stair step activation function at its output (Eqn. (22)) not involved in a learning process. This layer merges the initially clustered pixel data at M-dimensional input in two final classes: with and without ROS colour features, which are expressed respectively as zeros and ones at single output. The weight vectors of the output neuron $\mathbf{w}^{(\mathbf{2})} = [w_1^{(2)}, \dots, w_M^{(2)}]^T$ (Eqn. (21)) must be calculated by matching an output pattern (supervised learning) prepared by specialists as mentioned in Section 4.1,

$$u = \sum_{i=0}^{M} w_i^{(2)} y_i, \qquad (21)$$

$$z = \begin{cases} 1 & \text{if } u > 0.5, \\ 0 & \text{otherwise.} \end{cases} \qquad (22)$$

where y_0 is the offset adjusted in the output layer.

4.4. Training and validation of the network. For the training of a neural network's competitive layer, the MWTA method has been proposed, previously mentioned in Sections 1 and 4.3. The goal of the training modification is to minimise some drawbacks accompanying the standard WTA algorithm (Kohonen, 2001). All changes have been made between the consecutive epochs of standard WTA learning as shown in Fig. 12.

The WTA training procedure begins with random initialisation of the weight vectors $\mathbf{w}_i^{(1)}$ representing initial class centres, collected the weight array $\mathbf{W}^{(1)}$. They are selected by uniform, no-replacement, random sampling from indexes of the Q data samples entered in one epoch,

$$\mathbf{w_i^{(1)}}(0) = \mathrm{rand}(\{1,\dots,Q\}),\ i \in \{1,\dots,M\}. \tag{23}$$

For the proposed MWTA method its start is extended to the set $\{\mathbf{W}^{(1)}\}$ of several weight vector arrays handled independently according to the WTA rule during one epoch. Given the pixel feature input vector $\mathbf{x}(q)$ normalised in three dimensions, the output neurons with weights $\mathbf{w}_i^{(1)}(q)$ from array $\mathbf{W}_i^{(1)}$ compete to match to this vector using the dissimilarity measure (Rubner *et al.*, 2001) as the Euclidean distance:

$$\|\mathbf{x}(q) - \mathbf{w}_c^{(1)}(q)\| = \min_i \|\mathbf{x}(q) - \mathbf{w}_i^{(1)}(q)\|, \tag{24}$$

where $\mathbf{w}_c^{(1)}(q)$ is the weight vector winning competition at the presentation of the q-th datum. The weights of winner neuron y_c are then modified according to the rule

$$\mathbf{w}_i^{(1)}(r) = \begin{cases} \mathbf{w}_i^{(1)}(q) + \alpha^{(1)}\left[\mathbf{x}(q) - \mathbf{w}_i^{(1)}(q)\right] \\ \quad \text{if } i = c, \\ \mathbf{w}_i^{(1)}(q) \\ \quad \text{if } i \neq c, \end{cases} \tag{25}$$

where $r = q + 1$. The aim of WTA learning is to find the minimum quantisation error $E(\mathbf{W^{(1)}})$ at the approximation of all input data vectors $\mathbf{x}$ by M weight vectors (or neurons) shown in Fig. 10.

Using the Euclidean metric, this error can be expressed as

$$E\left(\mathbf{W}^{(1)}\right) = \max_c \frac{1}{Q}\sum_{q=1}^{Q} \|\mathbf{x}(q) - \mathbf{w}_c^{(1)}(q)\|^2, \tag{26}$$

where Q is the number of data vectors in one epoch, $\mathbf{w}_c^{(1)}(q)$, $c \in \{1,\dots,M\}$ is the weight of neuron winning at the presentation of the $\mathbf{x}(q)$ vector. The learning process described in Eqn. (25) ensures only the convergence to a local minimum, when the learning rate $\alpha^{(1)}$ is small enough. The learning rate has been experimentally chosen as a constant value $\alpha^{(1)} = 0.002$ to ensure good and fast convergence of the algorithm. The stopping criterion at the local minimum of $E(\mathbf{W^{(1)}})$ exploits the effect of neuron position (weight) stabilisation around the minimum and is expressed by Eqn. (27):

$$\max_i \|\Delta\mathbf{w}_i^{(1)}(ep)\| < \epsilon, \tag{27}$$

$$\Delta\mathbf{w}_i^{(1)}(ep) = \mathbf{w}_i^{(1)}(ep) - \mathbf{w}_i^{(1)}(ep-1), \tag{28}$$

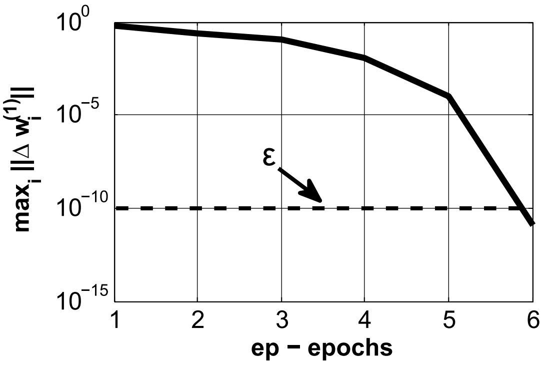


Fig. 11. Example plot of $\mathbf{W^{(1)}}(ep)$ weight stabilisation during MWTA training for validation using (H,S) pixel data of the image from Fig. 8(a). $\mathbf{W}^{(1)}(ep)$ represents the selected sequence of weights convergent to the minimum quantisation error $E(\mathbf{W}^{(1)})$ (Eqn. (26)).

where $i \in \{1,\dots,M\}$, ϵ represents the weights improvement limit and ep the epoch number (Fig. 11). The number of convergence steps to fulfil the above criterion is varying and depends on the randomly selected starting weights (Eqn. (23)). To limit the time of learning, the following additional condition of maximum allowable epochs EP is imposed on the convergence limit:

$$ep < EP. \tag{29}$$

During each epoch of WTA training all valid sets (arrays) of weight vectors $\mathbf{W}_k^{(1)} = [\mathbf{w}_i^{(1)}]_k$, $k = 1,\dots,N_k$, $i = 1,\dots,M$, are independently moved towards the nearest class centres minimizing the error function in Eqn. (26). Each of them also comes under the stopping criterion given above. After stopping, the currently fixed weight sets are stored for final solution analysis (arrays $\mathbf{V}_k$ in Fig. 12). All of still shifted sets are tested for their possible concentrations in the input data space. Sufficiently concentrated groups of weight arrays are replaced by the concentration centres before the next training epoch. It is expected that with the successive training epochs the weight arrays $\mathbf{W}_k^{(1)}$ will concentrate around the different local minima of $E(\mathbf{W}_k^{(1)})$ existing in the optimisation task considered. The measure of this concentration is taken as the maximum deviation d of $\mathbf{w_{ik}}$ set elements from weight set centres $\overline{\mathbf{w}}_\mathbf{i}$ as given in Eqn. (30):

$$d = \max_{k,i} \|\mathbf{w}_{ik}^{(1)} - \overline{\mathbf{w}}_i^{(1)}\|, \tag{30}$$

$$\overline{\mathbf{w}}_i^{(1)} = \frac{1}{N_k}\sum_{k=1}^{N_k} \mathbf{w}_{ik}^{(1)}, \quad i = 1\dots M,$$

where d is the deviation of the weight set, $-\overline{\mathbf{w}}_i^{(1)}$ is the mean weight vector of i-th neuron, $\mathbf{w}_{ik}^{(1)}$ stands for the weights of i-th neuron from the weight set $\mathbf{W}_k^{(1)}$.

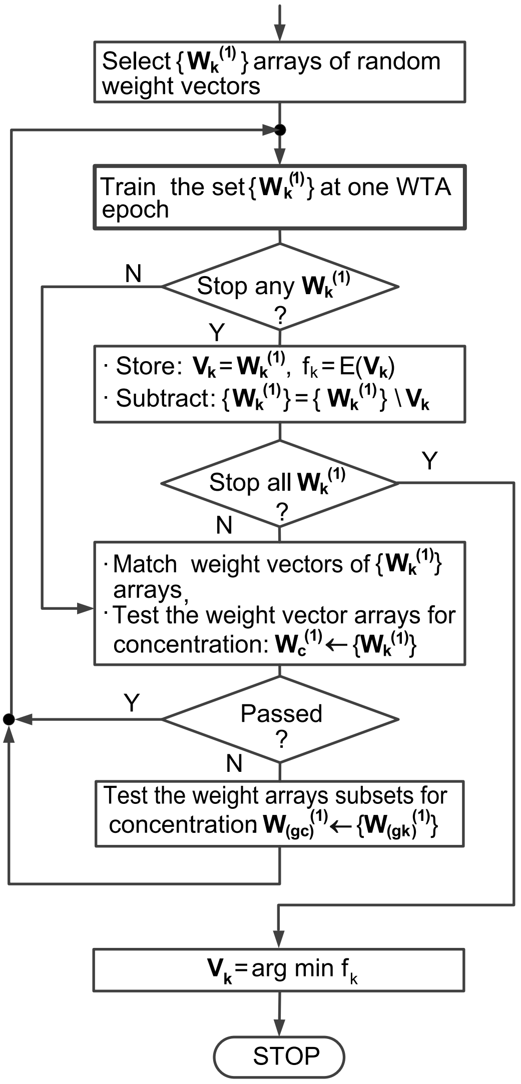

Select {W_k^(1)} arrays of random weight vectors
Train the set {W_k^(1)} at one WTA epoch
Stop any W_k^(1) ?
N
Y
· Store: V_k = W_k^(1), f_k = E(V_k)
· Subtract: {W_k^(1)} = { W_k^(1)} \ V_k
Stop all W_k^(1) ?
Y
N
· Match weight vectors of {W_k^(1)} arrays,
· Test the weight vector arrays for concentration: W_c^(1) ← {W_k^(1)}
Passed ?
Y
N
Test the weight arrays subsets for concentration W_(gc)^(1) ← {W_(gk)^(1)}
V_k = arg min f_k
STOP

Fig. 12. Flow diagram of the proposed MWTA algorithm for training the first layer of the neural network. $\mathbf{W}_k^{(1)}, k = 1, \ldots, N_k$: k-th set of neuron weights, $\mathbf{V}_k$, f_k: k-th vector set and its error function value stored when WTA learning is stopped, $\mathbf{W}_c^{(1)}$: centroids array of $\mathbf{W}_k^{(1)}$ set, $\mathbf{W}_{(gk)}^{(1)}$, $gk = 1, \ldots, N_{gk}$: k-th set of neuron weights in the group g, $\mathbf{W}_{(gc)}^{(1)}$: centroids array of $\mathbf{W}_{(gk)}^{(1)}$ set in the group g.

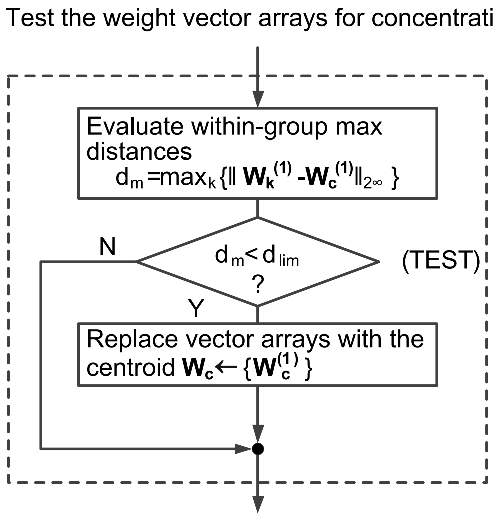

Test the weight vector arrays for concentration
Evaluate within-group max distances
d_m = max_k{‖ W_k^(1) - W_c^(1)‖_2∞ }
N
d_m < d_lim ?
(TEST)
Y
Replace vector arrays with the centroid W_c ← {W_c^(1)}

Fig. 13. Detailed flow diagram of testing the vector weight arrays $\mathbf{W}_k^{(1)}$ for concentration. d_{lim}: limit of weight vectors concentration, d_m: maximum deviation in the set $\mathbf{W}_k^{(1)}$.

Many weight sets help to increase the chance of detection and storage of more than one local minimum. The global minimum is then selected from them by the comparison of minimal values $E(\mathbf{W}_k^{(1)})$. The weight sets highly concentrated in groups are considered to be redundant. The weights in the same or almost the same positions come under the same attraction of data vectors at successive training epochs. The feature space distance limit $d_{\lim}$ in

$$d < d_{\lim} \Rightarrow \{\, \mathbf{w}_{ik}^{(1)}\} = \{\overline{\mathbf{w}}_i^{(1)}\}, \tag{31}$$

serves to recognise all weight sets $\mathbf{W}_k^{(1)}$ as the same and can be taken arbitrarily or can be better estimated from the original image resolution in a data space.

Here $\overline{\mathbf{w}}_i^{(1)}$ and d are as in Eqn. (30). When the weight set does not fulfil the condition as a whole, it is divided into subsets (subgroups) using the built-in *mean-shift* algorithm (Fukunaga and Hostetler, 1975).

$$\left\{\left\{\mathbf{W}_{(gk)}^{(1)}\right\}_g\right\} = \mathrm{mean\_shift}\left(\{\mathbf{W}_k^{(1)}\}, K, B\right), \tag{32}$$

where $gk \in \{1, \ldots, N_{gk}\}$ is the within-subset weight array index, $g \in \{1, \ldots, N_g\}$ is the subset index, K is the flat circle kernel with the radius B in $\mathrm{nrows}(\mathbf{W}_k^{(1)}) \times \mathrm{ncols}(\mathbf{W}_k^{(1)})$ dimensional space, B is a scalar bandwidth value. Then the process of deviation validating is repeated for each subset (Fig. 13).

After each epoch the vectors of any weight array relating to the same neurons may be located at random row numbers and have to be matched before the analysis of concentration. This has been done as follows:

$$\{\mathbf{w}_{ik}^{(1)}\} = \arg\ \mathrm{first}(\mathrm{sort}(\mathbf{R}_k), M)), \tag{33}$$
$$\mathbf{R}_k = \left[\rho_{ij}(\mathbf{w}_{ik}^{(1)}, \mathbf{w}_{j1}^{(1)})\right],$$

where the function $\mathrm{first}(\cdot, M)$ selects the sequence of M shortest distances $\rho_{ij}(\cdot,\cdot)$ between the i and j components of the first and k-th weight vector sets previously sorted in ascending order.

To train the output layer MWTA self-clustering must be first finished and its vector weights be fixed.

The network's linear layer is to minimise the error (Kohonen, 2001; Osowski, 2006)

$$E(\mathbf{W^{(2)}}) = \frac{1}{Q}\sum_{q=1}^{Q}\|\boldsymbol{\delta}(q)\|^2, \tag{34}$$

$$\boldsymbol{\delta}(q) = \mathbf{t}(q) - \mathbf{W}^{(2)}\mathbf{y}(q),$$

as a quadratic function of weight matrix $\mathbf{W}^{(2)}$, where $\mathbf{t}(q)$ signifies the q-th vector of the image pattern, $\mathbf{y}(q)$ is the q-th output vector of the self-organised layer, $\boldsymbol{\delta}(q)$ is the error between the q-th pattern and output vectors, $q \in \{1,\dots,Q\}, Q$ is the number of data vectors per one epoch.

An example behaviour of $E(\mathbf{W}^{(2)})$ is shown in Fig. 14. To minimise this error, the Widrow–Hoff method has been used for learning (Eqn. (35)). It applies a gradient-like adjustment for each example from the training set,

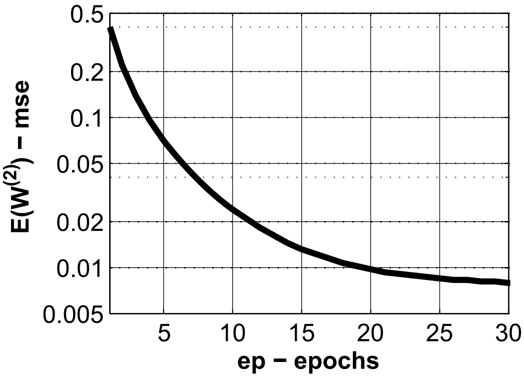


Fig. 14. Example plot of the mean square error vs. the epoch number in supervised learning of the network linear layer.

$$w_{ij}^{(2)}(q+1) = w_{ij}^{(2)}(q) + 2\alpha^{(2)}\cdot\delta_i(q)\cdot y_j(q), \tag{35}$$

where $w_{ij}^{(2)}$ is the weight between the j-th input and the i-th output of the second network layer, $y_j(q)$ is the j-th component of the M-dimensional output vector of the WTA layer. The symbol $\alpha^{(2)}$ means the maximum stable learning rate (Fukunaga and Hostetler, 1975):

$$\alpha^{(2)} = \frac{1}{\lambda_{\max}\left(\mathbf{YY^T}\right)}, \tag{36}$$

where $\lambda_{\max}(\cdot)$ is the maximum eigenvalue, $\mathbf{Y}$ is the horizontal array $[(M+1)\times Q]$ of Q input vectors provided to the neural network from Fig. 10 in one epoch.

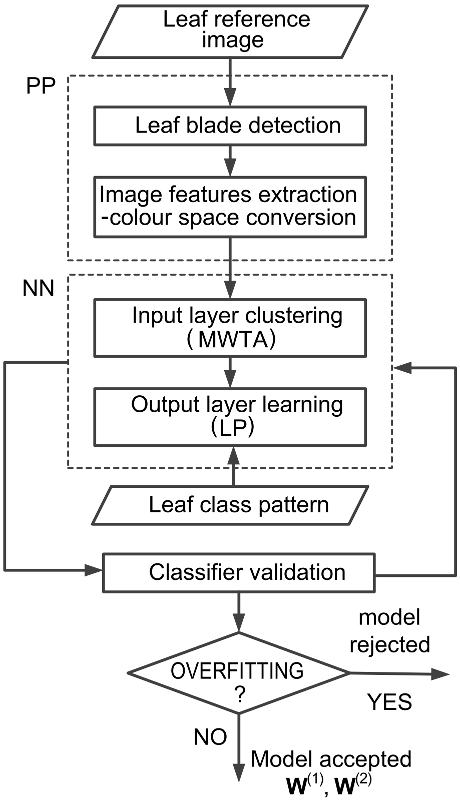


Fig. 15. Components of the neural network classifier model. PP: preprocessing module, NN: neural network module, LP: linear perceptron.

To verify the network generalisation ability, quintuple cross-validation has been carried out on the vectorised hue, saturation of leaf blade data $\mathbf{x} = [x_1, x_2]^T$ in several training images with manually labelled patterns (Fig. 15). The folding routine implements the MATLAB function *crossvalind*, which returns randomly generated indices for a K-fold cross-validation of Q data items,

$$\{\mathbf{ix_1},\ \mathbf{ix_2}\}_k = \mathit{crossvalind}('\mathit{Kfold}', Q, K), \tag{37}$$

where $k = 1,\dots,K$ is the fold number, $\mathbf{ix}_1, \mathbf{ix}_2$ are respectively training and testing index vectors with values in $\{1,\dots,Q\}$ for cross-validation of Q data. The presented algorithm divides leaf blade pixels into the class of ROS coloured ones and others, which gives four possible results in comparison with pattern data. The results are counted in separate elements of confusion matrices (Masters, 1993), whose model is shown in Fig. 16.

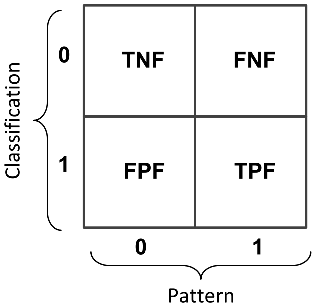
In K-fold cross-validation, only, K estimates of classification errors can obtained. To achieve better estimation performance, K-fold cross-validation can be executed several times with the data or their indices reshuffled before each round (Du Buf and Bayer, 2002; Refaeilzadeh *et al.*, 2009). As a result of the cross-validation, confusion matrices are computed for the classified data from trained images. The matrices include separate counts of all possible classification results. The confusion matrix based factors selected to estimate classification method quality are

$$PE = \frac{FPF}{TNF + FPF}, \tag{38}$$

$$NE = \frac{FNF}{FNF + TPF}, \tag{39}$$

$$ER = \frac{FNF + FPF}{\sum F}, \tag{40}$$

where $\sum F$ denotes the sum of all fraction counts, PE is the false positive error rate, NE is the false negative error rate, ER is the classifier error rate. The last factor is used to validate the proposed classifier. The process of preparing the classifier model is shown as the flowchart in Fig. 15.



		0	1
Classification	0	TNF	FNF
	1	FPF	TPF

Fig. 16. Structure of the confusion matrix for classification into two classes. TNF: true negative fraction, FNF: false negative fraction, FPF: false positive fraction, TPF: true positive fraction.

5. Experimental results

5.1. Experiment framework. The proposed segmentation algorithm with neural network classification was developed in the MATLAB 2008a environment as mentioned in Section 3. The code of the preprocessing stage was written as MATLAB scripts with the intensive use of vectorisation techniques. The colour space transformation, image thresholding, edge detection, tracing and the morphological extraction of the leaf blade mask (Figs. 1 and 4) apply appropriate functions built in the *Image Processing Toolbox* (The Mathworks Inc., 2011b). To increase the network training speed, the clustering of the first layer as well as the supervised learning of the second layer were written as C++ functions in MEX files compiled with *Visual Studio Express 2008* (The Mathworks Inc., 2011a). The prepared algorithm was executed on a PC with a dual core processor *Intel Core (TM)2 Duo T5750 2 GHz, 4 GB RAM* and the operating system *Windows 7*.

Segmentation by the presented method was performed on 12 images of single leaves with visible ROS accumulation regions. The images were read from JPEG files, where they had been stored after scanning. Three leaf images, with the best training ability values f_i (Eqn. (16)), one with blue and two with red-brown regions, were chosen as training data. They represent leaves of different plant species (cucumber, pumpkin) affected by the combinations of pathogen and additional stress factors (e.g., drought). The execution of the proposed segmentation algorithm is independent of the type of stress, as long as the colour symptoms in leaves are similar in hue and saturation. At the preprocessing stage of the segmentation only the leaves from two groups are distinguished, determined by the ROS types (O_2^- or $\mathrm{H}_2\mathrm{O}_2$). The binary pattern images of regions with stress response were manually labelled to enable supervised learning of the classifier output layer (Fig. 15).

To initially assess the classifier quality 5-fold cross-validation was performed for the two different training images from Figs. 17 and 20, representing different ROS types. Then the cardinalities of training vector sets were taken respectively as $4/5$ of the values 498473 and 387693 equal to the pixel number of each leaf blade.

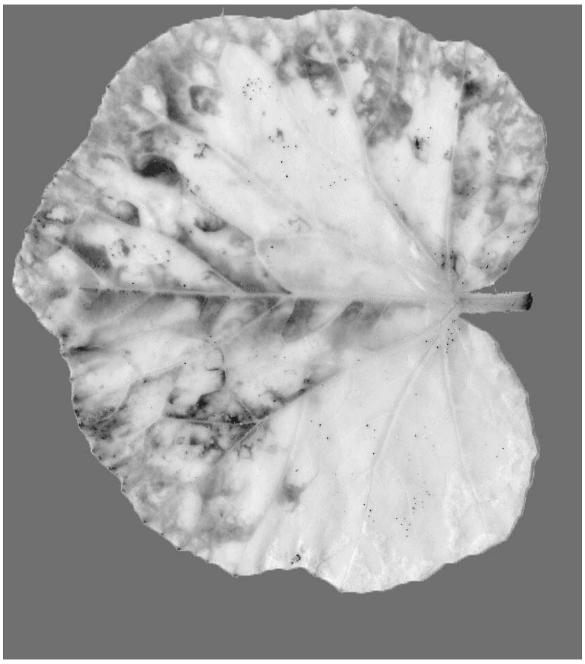


Fig. 17. Example of a pumpkin leaf image undergoing segmentation, with blue stained regions of ROS reaction products visible as darker pixels. Image size: 844×952 px, depth: 24 bit/px, leaf blade area: 498473 px.

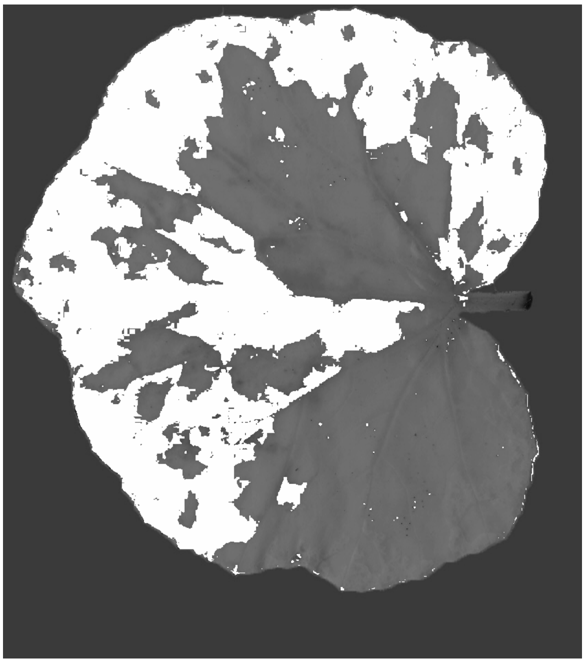


Fig. 18. Segmentation result for the image from Fig. 17 with the class of ROS stained pixels shown as white regions.

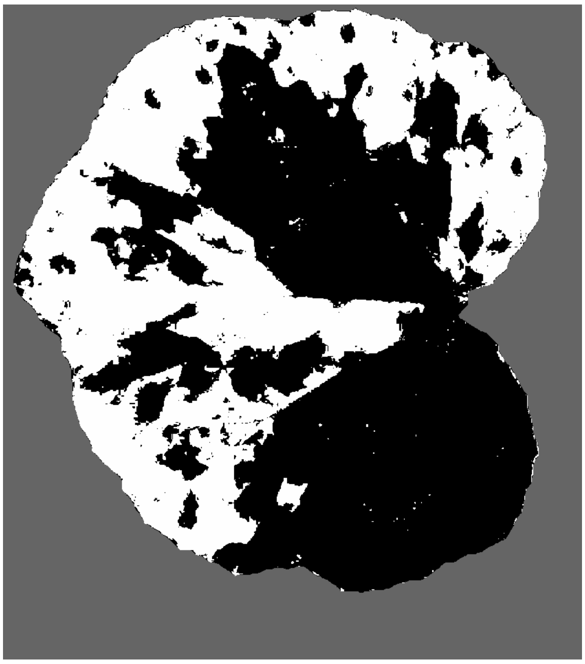


Fig. 19. Manually labelled pattern for the image from Fig. 17 with the class of ROS stained pixels shown as white regions and a grey image background.

5.2. Discussion of the results. The classifier error rates ER at five-fold cross-validation for the training images from Fig. 17 (Fig. 18) and Fig. 20 (Fig. 21) are visualised in Fig. 23. Small values of the errors in the range

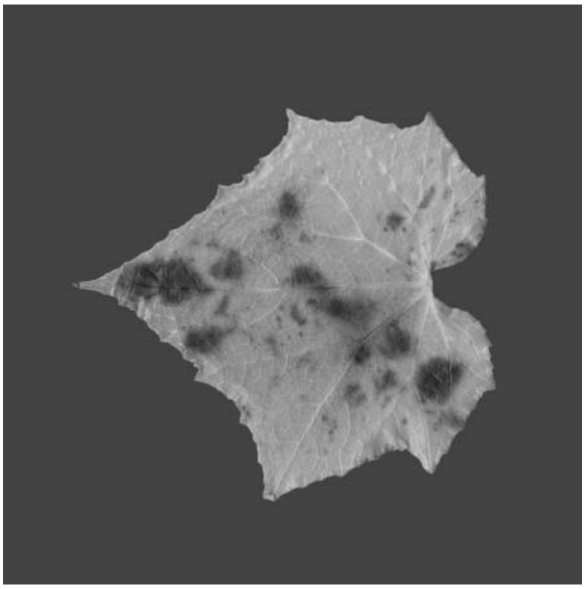


Fig. 20. Example of a cucumber leaf image before segmentation with red-brown stained regions of ROS reaction products visible as darker pixels. Image size: 1180×1182 px, depth: 24 bit/px, leaf blade area: 387693 px.

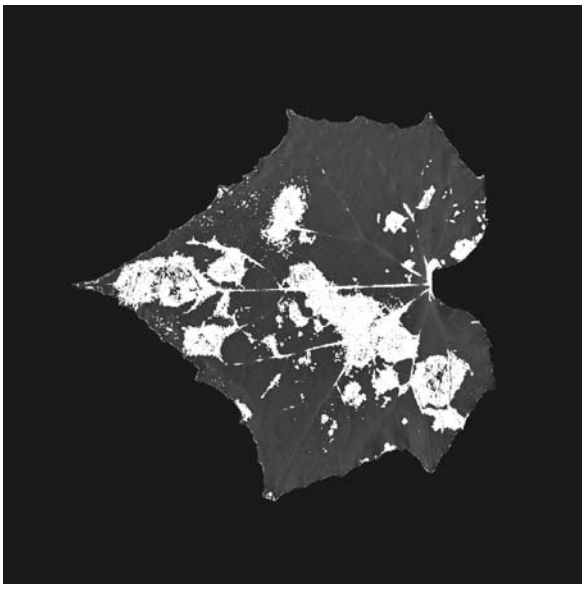


Fig. 21. Segmentation result for the image from Fig. 20 with the separated class of ROS stained pixels shown as white regions.

$[0.9,\ 1.6]\%$ confirm the proper training of the classifier, free of the overfitting phenomenon. The classification errors in the segmentation of 12 example images mentioned in Section 5.1 are listed in Table 2. The classification was preceded by three network trainings with images representing differently stained leaf groups. Each of the trainings involved the H, S feature data of all leaf blade pixels.

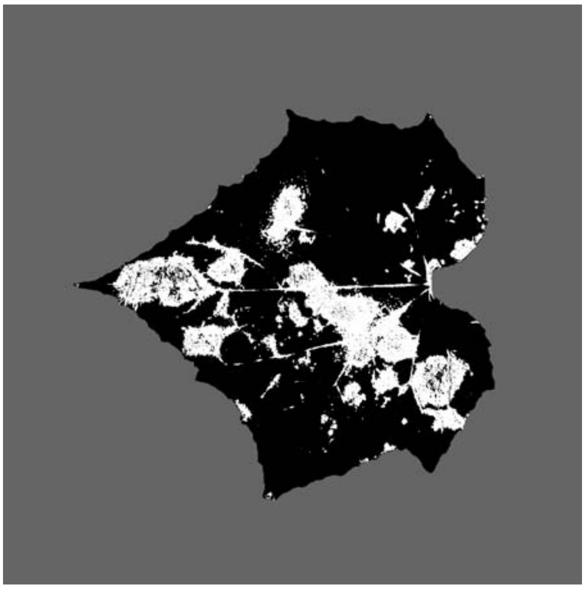


Fig. 22. Manually labelled pattern for the image from Fig. 20 with the class of ROS stained pixels shown as white regions and a grey image background.

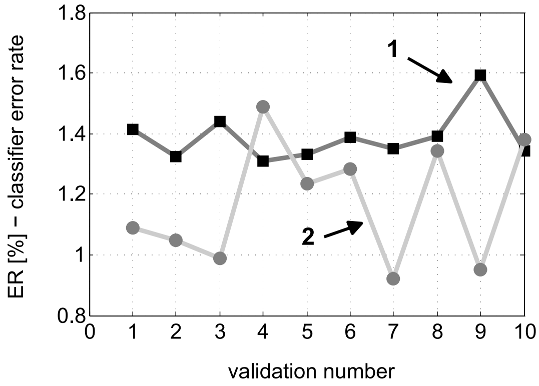


Fig. 23. Example plots of the error rates ER computed according to Eqn. (40) at five-fold cross-validation for images in Fig. 18 (plot 1) and Fig. 21 (plot 2).

Table 2. List of classification errors derived from the confusion matrix.

Image number	False positive error rate PE [%]	False negative error rate NE [%]	Classifier error rate ER [%]
1	2.39	0.01	1.36
2	2.25	0.01	1.77
3	1.79	0.18	1.59
4	3.04	0.19	2.77
5	0.08	8.64	0.24
6	0.75	0.03	0.56
7	1.22	0.00	0.95
8	1.45	1.83	1.52
9	0.10	3.47	0.97
10	1.60	6.22	1.94
11	1.20	5.30	1.39
12	1.87	3.78	1.92

Table 3. Example learning times for the images from Figs. 18 and 21, respectively. The learning times of layer 1 are calculated for $N_k = 10$ initial neuron weight sets.

Image	NN learning time	
number	layer 1	layer 2
(validation)	[s]	[s]
1(1)	29.95	0.97
1(2)	43.18	0.76
1(3)	27.63	0.61
1(4)	33.61	0.55
1(5)	30.08	0.63
2(1)	18.90	3.94
2(2)	17.69	2.70
2(3)	17.58	3.44
2(4)	19.84	4.23
2(5)	15.86	4.46

Classification errors were evaluated for all these images, whose binary patterns were manually labelled for this purpose (Figs. 19 and 22). The computed error rates ER vary from 0.24% to 2.77% (mean 1.42%), which means good compliance with the patterns. False positive errors (mean 1.49%) are regarded as less important than false negative errors (mean 2.30%), because further studies of dye concentration in the stained areas enable detection of this type of errors.

It should be emphasised that sometimes manual identification of stained foliar areas can be ambiguous, which gives different possible patterns for one image. In such cases the pattern variant closest to the automatic classification results was taken into account. The accuracy of final classification results was accepted by specialists identifying the sites of ROS accumulation in stained leaves. The execution times of MWTA self-clustering and linear perceptron supervised learning registered in the example cross-validation are listed in Table 3 for $M = 8$ MWTA layer neurons and $N_k = 10$ weight vector sets. The average clustering times computed from time data shown in Table 3 are about 33 s and 18 s respectively, for the training images from Figs. 17 and 20. The times can be different because of different leaf areas and (H, S) distributions as well as the random starting values of MWTA initial weight vectors. The MWTA training with 10 initial weight sets lasts on average 4 to 5 times longer than the classic WTA learning of the same tested image population. This is the cost of increasing the chance to achieve the global minimum of the clustering error in the first layer.

Because MWTA consists of partially independent tasks of weight arrays correction, they can be executed as parallel threads with GPU computing applying CUDA or

Open CL technology (Sanders and Kandrot, 2011). This will be the subject of future research. The MWTA $(N_k = 10)$ classification errors shown in Table 2 are of the same order as in the case of applying classic WTA $(N_k = 1)$ when each of these methods stops at the same weight array $\mathbf{W}^{(1)}$ indicating the global minimum of $E(\mathbf{W}^{(1)})$ as given in Eqn. (26). If the WTA method reaches only a local, but not the global, minimum of $E(\mathbf{W}^{(1)})$, the error rates of classification can be relatively high.

It may happen for some types of error functions with different local and global minima, whose shapes are explicitly unknown and a single initial weight array in the WTA algorithm is randomly selected. The high error values can also appear when this array has been accidentally localised far from the nearest minimum of $E(\mathbf{W}^{(1)})$ and the minimisation process has been broken by the limited number of allowable iterations. In the series of 100 trainings with the classic WTA method applied to four images with high ability factors f_i (Eqn. (16)), increased classification errors up to $ER \approx 6 - 7\%$ appeared on average in 1 per 12 times. Two of the four tested images are shown in Figs. 17 and 20. All trainings were limited to 60 epochs at the clustering stage and had the convergence condition ϵ (Section 5.3).

The training of the linear perceptron layer with the maximum learning rate (Eqn. (36)) is much faster (0.7 s and 3.8 s on average). It should be remembered that the learning process refers only to the small number of training images and the rest of each image population is intended for the classification, typically 10 to 20 times faster than the learning.

5.3. Algorithm parameters.

The developed algorithm has a series of parameters, which can be tuned if necessary to achieve good adaptation to different image classes. The preprocessing involves two parameters:

- $R_1 = 20$: the radius of the structuring element used at the image I_M opening (Eqn. (9)) to cut off a leaf's petiole, above half an expected petiole width,
- $R_2 = 4$: the radius of the structuring element used at the auxiliary erosion/dilation (Eqns. (9) and (12)) for the isolation of leaf edge protrusions.

The most important parameters of NN-classification are

- $M = 8$: the number of clusters (output neurons) in the first layer,
- $N_k = 10$: the number of initial weight vector sets for MWTA training,
- $\alpha^{(1)} = 0.002$: the learning rate of the MWTA layer,
- $\epsilon = 10^{-10}$: the limit of the weight vector location error $\max_i \|\Delta \mathbf{w}_i^{(1)}(ep)\|$ in the clustering layer,
- $EP = 60$: the allowed number of epochs at clustering,
- $\epsilon = 0.01$: the limit of the mean square learning error $E(\mathbf{W}^{(2)})$ in the output layer,
- $EP = 100$: the allowed number of epochs at output learning,
- $B = 0.002$: mean-shift bandwidth,
- $d_{\lim} = 0.004$: distance limit for MWTA weight concentration.

The number of first layer clusters $M = 8$ was selected as the lowest number of neurons for which the average classification error $\overline{ER}$ of 12 tested images from Table 2 stops to decrease at the value near 1.4% (Fig. 24). A further enlargement of M is then useless and only increases the computational effort. In this paper only the correctness of the image segmentation method has been verified. To take full advantage of the results in biological research, the comparison with classic biochemical methods is necessary. The quantitative segmentation results expressed as the areas of ROS reaction products in plant leaves will be statistically compared with the results of biochemical analysis in the future.

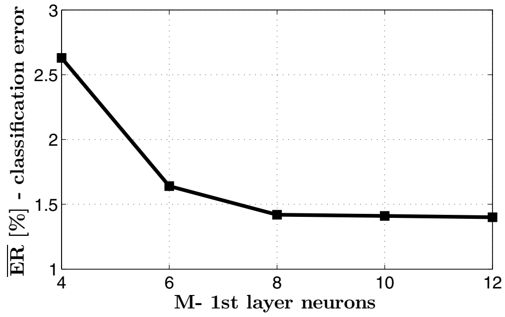


Fig. 24. Plot of a mean classification error $\overline{ER}$ vs. the number M of neurons in the neural network's first layer.

6. Conclusions

A method for automatic segmentation of cucurbits leaves was presented. It aims to detect the blade regions containing the colour products of ROS the reaction. Counting and measurements of these areas are of basic importance for biologists defining the degree of the plant response to biotic and abiotic stress, and it is a terminal stage of the complex biological experiment. In this case, ROS measurement can verify the influence (positive or negative) of abiotic factors (salinity and drought) on pathogen infection. The results of experiments can give a suggestion for modification of cultivation conditions, and thus provide a natural protection of cucurbits against fungal disease. All leaf pixel classification errors collected in Table 2 fall in the range acceptable by biologists. The execution is also faster than manual labelling. This indicates the usefulness of the proposed method for the extraction and area measurement of ROS accumulation sites in stained leaves.

The proposed algorithm consists of preprocessing and classification stages. The preprocessing includes leaf blade detection and leaf petiole elimination using procedures shown in the flow diagrams in Figs. 1 and 4. The second stage of the algorithm applies a two-layer neural network of the counter propagation type to classify hue and saturation data of leaf blade pixels. The first network layer has been trained using the new MWTA (modified WTA) method (Fig. 12), which needs more time to execute than original WTA learning, but instead gives a much higher certainty of correct classification due to the use of several initial neuron weight arrays. At the training phase of the neural network's first layer the neuron weight arrays are systematically replaced by their centres when they sufficiently concentrate around the local minima of the quantisation error function (Eqn. 25). Finally, only the solution providing the global minimum is selected.

The accuracy of leaf pixels classification obtained with the proposed method is sufficient to estimate the level of plant reaction stress in the examined populations. In the paper only a counter propagation network type was considered to replace manual classification of leaf blade pixels. The possible comparison of the proposed solution with other neural network types to find the optimal architecture and training methods is out of the scope of this paper and will be the subject of future research. The proposed segmentation method of leaf images with ROS reaction colour products is fully automatic except for the neural network training phase, which requires manual labelling of single pattern images. Therefore it is faster than any manual labelling of every image using general purpose graphic applications and also more free of human errors, which improves overall segmentation accuracy. Implementation of the segmentation algorithm requires only the MATLAB environment with the *Image Processing Toolbox* and low cost hardware (a personal computer and a desktop scanner), which should be accessible in each research laboratory.

References

Cheeseman, J.M. (2006). Hydrogen peroxide concentrations in leaves under natural conditions, *Journal of Experimental Botany* **57**(10): 2435–2444.

Du Buf, H. and Bayer, M. (Eds.) (2002). *Automatic Diatom Identification*, World Scientific Publishing, New York, NY/London/Singapore/Hong Kong.

Forgy, E. (1965). Cluster analysis of multivariate data: Efficiency vs. interpretability of classification, *Biometrics* **21**: 768–769.

Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Transactions on Information Theory* **21**(1): 32–40.

Gonzalez, R.C. and Woods, R.E. (2008). *Digital Image Processing,* 3rd Edn., Prentice Hall, Upper Saddle River, NJ.

Hagan, M.T., Demuth, H. B. and Beale, M.H. (2009). *Neural Network Design*, University of Colorado, Denver, CO, Chapter 10, http://www.personeel.unimaas.nl/westra/Education/ANO/10widrow_hoff.pdf

Hecht-Nielsen, R. (1987). Counterpropagation networks, *Applied Optics* **26**(23): 4979–4984.

Huang, CX-S., Liu, J-H. and Chen, X-J. (2010). Overexpression of PtrABF gene, a bZIP transcription factor isolated from *Poncirus trifoliata*, enhances dehydration and drought tolerance in tobacco via scavenging ROS and modulating expression of stress-responsive genes, *BMC Plant Biology* **10**: 1–18, paper 230, http://www.biomedcentral.com/1471-2229/10/230.

James, W.C. (1971). An illustrated series of assessment keys for plant diseases, their preparation and usage, *Canadian Plant Disease Survey* **51**(2): 39–65.

Kohonen, T. (1990). The self-organising map, *Proceedings of IEEE* **78**(9): 1464–1479.

Kohonen, T. (2001). *Self-organizing Maps,* 3rd Edn., Springer-Verlag, Berlin/ Heidelberg/New York, NY.

Masters, T. (1993) *Practical Neural Network Recipes in C++*, Academic Press Inc., San Diego, CA.

Ong, S., Yeo, N., Lee, K., Venkatesh, Y. and Cao, D. (2002). Segmentation of color images using a two-stage self-organizing network, *Image and Vision Computing* **20**(4): 279–289.

Osowski, S. (2006). *Neural Networks for Information Processing*, Warsaw University of Technology Press, Warsaw, (in Polish).

Otsu, N. (1979). A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man, and Cybernetics* **9**(1): 62–66.

Refaeilzadeh, P., Tang, L. and Liu, H., (2009). Cross-validation, *Encyclopedia of Database Systems*, Springer Publishing Company, New York, NY, pp. 532–538,

Rubner, Y., Puzicha, J., Tomasi, C. and Buhmann J.M. (2001). Empirical evaluation of dissimilarity measures for color and texture, *Computer Vision and Image Understanding* **84**(1): 25–43.

Sanders, J. and Kandrot, E. (2011). *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley, New York, NY/London.

Smith, A.R. (1978). Color gamut transform pairs, *Computer Graphics* **12**(3): 12–19.

Soukupova, J. and Albrechtova, J. (2003). Image analysis—Tool for quantification of histochemical detection of phenolic compounds, lignin and peroxidases in needles of Norway spruce, *Biologia Plantarum* **46**(4): 595–601.

Tabrizi, P.R., Rezatofighi, S.H. and Yazdanpanah, M.J. (2010). Using PCA and LVQ neural network for automatic recognition of five types of white blood cells, *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, Teheran, Iran*, pp. 5593–5596.

Tang, Q-H., Liu, B-H., Chen, Y-Q., Zhou, X-H. and Ding, J-S. (2007). Application of LVQ neural network combined with the genetic algorithm in acoustic seafloor classification, *Chinese Journal Geophysics* **50**(1): 313–319.

The Mathworks Inc. (2011a). MEX-files guide, `http://www.mathworks.com/support/tech-notes/1600/1605.html`

The Mathworks Inc. (2011b). Image processing toolbox user's guide, `http://www.mathworks.com//help/toolbox/images/`

Thordal-Christensen, H., Zhang, Z., Wei, Y.D. and Collinge, D.B. (1997). Subcellular localization of H_2O_2 in plants. H_2O_2 accumulation in papillae and hypersensitive response during the barley-powdery mildew interaction, *Plant Journal* **11**(6): 1187–1194.

Unger, Ch., Kleta, S., Jandl, G. and Tiedemann, A. (2005). Suppression of the defence-related oxidative burst in bean leaf tissue and bean suspension cells by the necrotrophic pathogen *Botrytus cinerea*, *Journal of Phytopathology* **153**(1): 15–26.

Wijekoon, C.P., Goodwin, P.H. and Hsiang, T. (2008). Quantifying fungal infection of plant leaves by digital image analysis using Scion Image software, *Journal of Microbiological Methods* **74**(2–3): 94–101.

Widrow, B., Winter, R.G. and Baxter, R.A. (1988). Layered neural nets for pattern recognition, *IEEE Transactions on Acoustics, Speech and Signal Processing* **36**(7): 1109–1118.

Jarosław Gocławski received the M.S.E. degree in electronics in 1977 and then the Ph.D. degree in 1986, both from the Electrical Faculty of the Technical University of Łódź, Poland. From 1977 to 1989 he worked on image processing, analysis and tomographic reconstruction as a lecturer at the Institute of Electronics of the Technical University of Łódź. From 1989 to 2005 he was employed at the Informatics Department in the IMAL company in Łódź, where he worked on the design of specialised software and hardware image processing solutions for medicine and biology. Since 2006 he has been a lecturer at the Computer Engineering Department of the Technical University of Łódź. His research interests include image processing and analysis for the purpose of measurements in textile industry and the biology of plants. In the last three years he has co-authored several journal and conference papers in these areas.

Joanna Sekulska-Nalewajko received the M.Sc. degree in environmental biology in 1996 and then the Ph.D. degree in phycology and hydrobiology in 2001, both at the Faculty of Biology and Environmental Protection, University of Łódź, Poland. In 2001 she was employed at the Computer Engineering Department of the Technical University of Łódź. Since 2005 she has been an assistant professor there. She works on diatom ecology in modern and past inland water environments. Another research interest of hers is concerned with biological image analysis for diagnostic purposes. She is an author of several papers on this topic.

Elżbieta Kuźniak received M.Sc. degree in biology in 1986 and the Ph.D. in plant physiology in 1994 from the Faculty of Biology and Environmental Protection, University of Łódź, Poland. In 2005, she was promoted to the present position of an assistant professor in the Department of Plant Physiology and Biochemistry, University of Łódź. Her current research interest focuses on the roles of reactive oxygen species and antioxidants in plant-pathogen interactions.

Received: 12 September 2011
Revised: 18 March 2012