

# Comparison of different classification algorithms for weed detection from images based on shape parameters

Martin Weis <sup>\*</sup>, Till Rumpf <sup>†</sup>, Roland Gerhards,  
Lutz Plümer

Variability of weed infestation needs to be assessed for site-specific weed management. Since manual weed sampling is too time consuming for practical applications, a system for automatic weed sampling was developed. The system uses bi-spectral images, which are processed to derive shape features of the plants. The shape features are used for the discrimination of weed and crop species by using a classification step.

In this paper we evaluate different classification algorithms with main focus on  $k$ -nearest neighbours, decision tree learning and Support Vector Machine classifiers. Data mining techniques were applied to select an optimal subset of the shape features, which then were used for the classification. Since the classification is a crucial step for the weed detection, three different classification algorithms are tested and their influence on the results is assessed. The plant shape varies between different species and also within one species at different growth stages. The training of the classifiers is run by using prototype information which is selected manually from the images.

Performance measures for classification accuracy are evaluated by using cross validation techniques and by comparing the results with manually assessed weed infestation.

**Keywords:** weed mapping, weed detection, digital image analysis, shape analysis, feature selection, classification, supervised learning

---

<sup>\*</sup>Dep. of Weed Science, Univ. of Hohenheim, Germany, [Martin.Weis@uni-hohenheim.de](mailto:Martin.Weis@uni-hohenheim.de)

<sup>†</sup>Dep. of Geoinformation, Inst. of Geodesy and Geoinformation, Univ. of Bonn, Germany, [Rumpf@igg.uni-bonn.de](mailto:Rumpf@igg.uni-bonn.de)

# 1. Introduction

The automated identification of weed species in the field is close to be realized as an operational product. Information about weed species, weed density, and weed coverage distribution can be used to manage weeds on a sub-field level either by using chemical or mechanical control. Sampling (using a regular or irregular grid) is necessary to manage weeds site-specifically. Manual weed scouting is time consuming and gets expensive with increasing number of sampling points. Brown and Noble (2005) reviewed the approaches for weed identification, Slaughter et al. (2008) identified the robust weed detection as a primary obstacle for robotic weed control technology and reviewed the approaches for weed detection as well as actuator technology. Non-imaging photodiode sensors have been used without and with artificial light sources [Sui et al. (2008)] to detect plants and assess weed infestations, but these sensors cannot distinguish between plant species.

Image processing methods were successfully used to distinguish different weeds and crops using shape parameters: Pérez et al. (2000) computed different compactness features and crop row location information to detect broadleaved weeds in cereals. They reached similar classification performances of about 89% (cereals) and 74% (broadleaved weeds) using a *k*NN and Bayes rule classifier.

A system was developed by Sökefeld et al. (2007) and Oebel and Gerhards (2005) to discriminate weeds and crop from bi-spectral images by using shape features. Field tests led to average herbicide savings of 35-70%. This system was reimplemented and refined. The system aims at detecting weeds and distinguishing them from the crop to make decisions about the best management strategy. In order to derive a suitable classification with the shape features three different algorithms were compared. The objective of this study is the selection of a suitable classification algorithm, therefore properties of different algorithms are contrasted and discussed after applying them to a data set.

## 2. Materials and methods

Images were taken in the field and processed by using image processing techniques to derive shape features for the objects in the image. A process chain for an automatic classification of different weeds based on the shape parameters was defined in the software Rapid Miner [Mierswa et al. (2006)]. Two models, one with a base division with regard to the species and another with regard to subclasses for the species were learned on the basis of training data for each of the three classifiers, especially focusing on Support Vector Machines (SVMs). After normalisation with mean zero and standard deviation one the inner evaluation of the classifiers' performances was carried out by cross-validation. Finally, we applied the learned model to unseen data samples and evaluated the prediction performance by using a manual classified set of images.

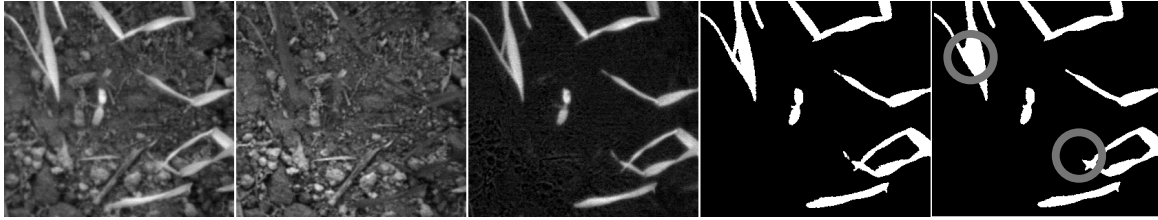


Figure 1: Image processing (from left to right): infrared (IR) image, red (R) image, difference image, thresholded image, morphological operators and size criterion applied. Circles: small region filtered and hole closed.

## 2.1. Study site

We have used data sets from a field located on the experimental field station of the University of Hohenheim (Ihinger Hof), which is southwest of Stuttgart. The measurements took place in November 2008 in an oil seed rape field (*Brassica napus* L., BRSNN) with some volunteer barley (*Hordeum vulgare* L., HORVS). The latter is the weed species, which can be controlled by a post-emergence herbicide for grass weeds. The crop was in a cotyledon stage and the weed in a one- or two-leaf stage. The images had a ground resolution of  $0.25 \text{ mm}^2$  per pixel and the field was sampled along tracks with 3 m distance.

## 2.2. Image processing

The system uses difference images of the near infrared and red light spectra. The images were taken in the field from a distance of 1 m and were associated with a DGPS (differential GPS) position. Since plants absorb red light (620–680 nm) for their photosynthesis and have a high reflection in the near infrared spectrum ( $> 720 \text{ nm}$ ), they appear dark in the red image (R) and bright in the near infrared (IR) as shown in Fig. 1 (the example in Fig. 1 was taken from another series, in which red and infrared images were available, and thus not showing oil seed rape). Other materials (soil, mulch, stones) have a similar reflection in these bands. In the difference of the two images (IR-R) the plants appear bright and the background is dark. This observation principle allows for differentiating between plants and background with a grey level threshold. Some preprocessing steps to reduce noise follow: a size criterion suppresses smaller regions than the cotyledon leaves of oil seed rape (right circle in Fig. 1) and morphological operations (opening/closing) smooth the contour and connect nearby regions as well as fill small holes (left circle in Fig. 1).

Shape features are derived for each remaining segment, which are defined as connected components in the binary image. Three types of features are computed: region-based features, that are derived from the pixels of each segment, e.g. size, statistical moments and Hu features [Hu (1962)], which are well known for a long time. The second type of features, the contour-based features, are derived from the border pixels of the segments. Fourier descriptors and curvature scale space features fall into this category. The third type of features are derived from the skeleton of the segments, which is the centre line. Combined with a

distance transform, which assigns each pixel of the region a value for the distance to the border, statistical measures resemble the thickness of the segments (mean, max, variance) and are useful to distinguish broad leaves from narrow leaves.

The shapes of different species vary between the different growth stages. Training data for the shape based classification uses classes which describe the species (EPPO-Code), their growth stage (BBCH-Code) and special situations which occur during segmentation (i.e. single leaves L, overlapping O). The following classes were defined for the subclasses training set: BRSNN10L, BRSNN10N (single leaf and whole plant in cotyledon stage); HORVS10N, HORVS12N, HORVS12O, HORVS12L (N means ‘normal’ segmentation) and two additional noise classes for elongated and compact noise in the images: NOISE-00L, NOISE00X. No other species were found in the images.

All information about the images, positions, classes, the training data and results of the image processing and classification are stored in a relational database. A process chain for an automatic classification in the software Rapid Miner [Mierswa et al. (2006)] was defined.

By using a genetic feature selection algorithm the dataset was reduced to the following relevant ten shape features: *areasize*, *compactness*, *Drear*, *Drigh*, *Hu2*, *Hu3*, *Ja*, *Jb*, *Rmin*, *skelmax*. These are nine region-based features (J\*: inertia of main axes, D\*: distances to mainaxes, Hu\*: Hu features, Rmin: minimum distance to border) and one skeleton feature (*skelmax*, maximum distance of the skeleton to the border).

## 2.3. Classification

Classification algorithms aim at finding regularities in patterns of empirical data (training data). If two classes of objects are given the problem is also known as dichotom. Then the classifier is faced with a new object which has to be assigned to one of the two classes. The training data can be formulated as

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n) \quad (1)$$

where the bold  $\mathbf{x}$  describes a feature vector of the form  $x_1, \dots, x_m$ , and where  $y_k = 1$  if  $\mathbf{x}_k$  belongs to class one and  $y_k = -1$  if  $\mathbf{x}_k$  is in class two. A labelled object  $y$  has an assignment to one of the classes and its feature vector  $\mathbf{x}$  is used for the learning of supervised classifiers. This results in finding a function  $f$  which approximates the training data in the best way possible, i.e.

$$y_i = f(\mathbf{x}_i) \quad \forall \quad i \in \{1, \dots, n\}. \quad (2)$$

In this case the classes are different weeds at varying stages of development and the features are shape parameters. However it is not enough to find a function  $f$  which fits the training data. Additionally the ability to classify unseen data, which is often called as generalisation ability, is important, too. Finding a function which fits the training data is an ‘ill-posed problem’ since there is an infinite number of functions having this property. It is a basic assumption of machine learning that the class of functions to be learned has to be restricted.

The most simple case is a set of linear functions which have a fixed number of parameters given by the number of features.

Learning a function  $f$  starts with a set of labelled training data, which in our case was generated from the segmented images by defining prototypes. Afterwards the learned function is also evaluated on labelled test data. In the following we introduce the three classification algorithms. First, a short overview on the  $k$ -nearest neighbour algorithm and the decision tree learning is given. Afterwards, the support vector machine (SVM) classifier is explained in more detail.

### 2.3.1. $k$ -Nearest Neighbour

One example for an instance-based learning algorithm is the  $k$ -Nearest Neighbour ( $k$ NN) algorithm. It uses the  $k$  nearest neighbours to make the decision of class attribution directly from the training instances themselves. Usually, Euclidean distance is used as distance metric. The decision for attaching the sample in question to one of the several classes is based on the majority vote of its  $k$  nearest neighbours. An odd number should be chosen for  $k$  to allow for a definite majority vote.

### 2.3.2. Decision tree

Decision tree learning is one of the most widely used and practical methods for inductive inference. There are many algorithms for constructing decision trees. We use the most popular one: i.e. C4.5 [Quinlan (1993)]. A decision tree can be seen as a data structure in form of a tree. Every interior node contains a decision criteria depending only on one feature. The features' relevance for classification is determined by entropy reduction which describes the (im)purity of the samples. For the first split into two parts the feature with the highest relevance is used. After such a decision the next feature is determined which splits the data optimally into two parts. Since always one feature is considered at a time, a boundary of axis parallel parts is formed. This is recursively repeated on each derived subset. If followed from root to a leaf node the decision tree corresponds to a rule based classifier. An advantage of decision tree classifiers is their simple structure, which allows for interpretation (most important features are near the root node) and visualisation.

### 2.3.3. Support vector machines

Support Vector Machines (SVM) were invented by Vladimir Vapnik in 1979 [Vapnik (1982)]. Basically the SVMs separate two different classes through hyperplanes. If the classes are separable by hyperplanes an optimal function can be determined from the empirical data. The hyperplane is expressed by its normal vector  $\mathbf{w}$  and a bias  $b$ . The class of hyperplanes can be specified in the scalar product space  $\mathbb{H}$  (feature space) as follows

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \quad \text{where} \quad \mathbf{w} \in \mathbb{H}, b \in \mathbb{R} \quad (3)$$

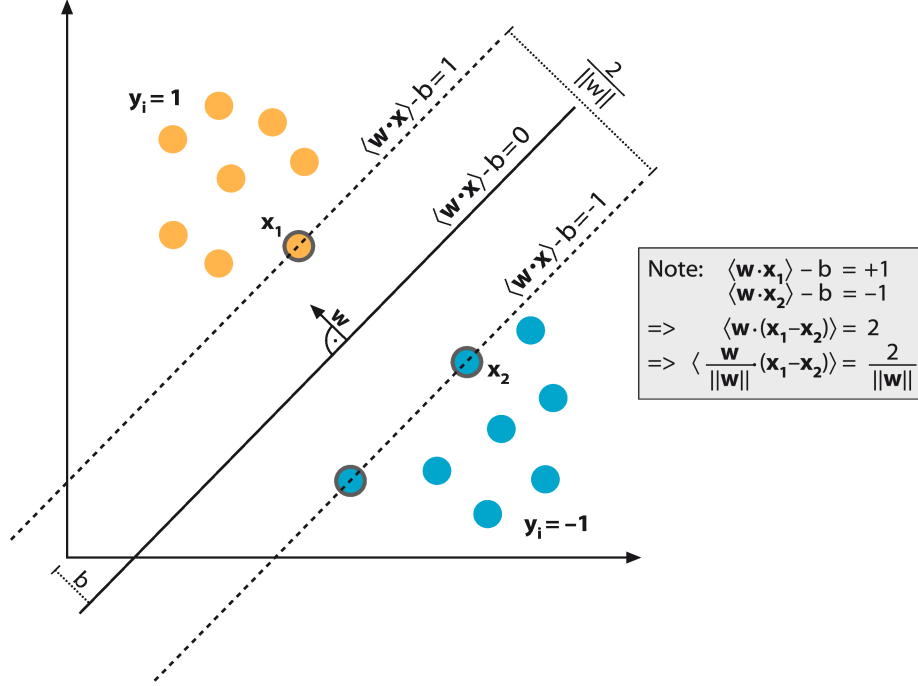


Figure 2: The optimal separating hyperplane is shown as a solid line.

where  $\langle \mathbf{w}, \mathbf{x} \rangle$  means  $x_1 \cdot w_1 + \dots + x_n \cdot w_n$ . This yields the corresponding decision function

$$f(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) \quad (4)$$

where the sign function extracts the sign of a real number. It is defined as  $-1$  if  $f(\mathbf{x}) < 0$  and  $1$  if  $f(\mathbf{x}) > 0$  which denotes the two different class labels  $+1$  and  $-1$ . Usually there exist many hyperplanes which separate the two classes. The basic idea behind SVMs is that the optimal hyperplane maximises the margin between data sets of opposite classes. In order to construct the optimal hyperplane, the following equation has to be solved

$$\min_{\mathbf{w} \in \mathbb{H}, b \in \mathbb{R}} \tau(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 \quad (5)$$

$$\text{subject to } y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \quad \forall i \in \{1, \dots, n\}. \quad (6)$$

The constraint (6) ensures that  $f(\mathbf{x}_i)$  yield  $+1$  for  $y_i = +1$  and  $-1$  for  $y_i = -1$ , and that the two classes are separated correctly. If  $\|\mathbf{w}\| = 1$  the left hand side of (6) is the distance of the training sample  $\mathbf{x}_i$  to the hyperplane. This is the Hessian normal form representation of the hyperplane. The distance of each training sample to the hyperplane can be computed by dividing  $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)$  by  $\|\mathbf{w}\|$ . The overall margin is maximised if the constraint (6) is satisfied for all  $i \in \{1, \dots, n\}$  with  $\mathbf{w}$  of minimal length, as given in (5). The distance of the closest point to the hyperplane is  $1/\|\mathbf{w}\|$ . This can be illustrated by considering two

training samples, one of each class respectively, and by projecting them onto the hyperplane normal vector  $\mathbf{w}/\|\mathbf{w}\|$  (see Note in Fig. 2).

The formulas (5) and (6) specify the constrained optimisation problem. It can be transformed to a ‘dual’ problem, where  $\mathbf{w}$  and  $b$  are eliminated by introducing Lagrange multipliers  $\alpha_i$ .

$$\max_{\alpha \in \mathbb{R}^n} W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \quad (7)$$

$$\text{subject to} \quad \alpha_i \geq 0 \quad \forall \quad i \in \{1, \dots, n\} \quad \text{and} \quad \sum_{i=1}^n \alpha_i y_i = 0. \quad (8)$$

This leads to the following decision function

$$f(\mathbf{x}) = \text{sgn} \left( \sum_{i=1}^n y_i \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b \right) \quad (9)$$

where  $b$  can be computed by the Lagrange multipliers which do not equal zero. These are called support vectors. All other samples with  $\alpha_i = 0$  are discarded.

Up to now only linearly separable classes were considered, but SVMs are able to classify samples with a non-linear discriminant. The basic idea of SVMs is to map the data into a new feature space and then solve the constrained optimisation problem. Obviously it seems to be very expensive to compute the mapping into a high-dimensional space. For this reason a kernel function is introduced to make the computation very simple [Boser (1992)]. This is referred to as the ‘kernel trick’, which causes an implicit mapping in the feature space without explicitly knowing the mapping function  $\Phi$ . Accordingly the scalar product  $\langle \mathbf{x}, \mathbf{x}_i \rangle$  can be substituted by

$$k(\mathbf{x}, \mathbf{x}_i) := \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \rangle = \langle \mathbf{x}, \mathbf{x}_i \rangle. \quad (10)$$

So far we have made the implicit assumption that the datasets are free of noise and may be classified perfectly. In this case SVMs give ‘hard margins’. In practice, this assumption does not hold true in most cases. This problem, however, may be handled by ‘soft margins’ which allow and penalize classification errors. Accordingly, in 1995 a modification was introduced where slack-variables  $\xi_i$  are used to relax the so-called hard-margin constraints (6) [Cortes and Vapnik (1995)], so that some classification errors depending on  $\xi_i$  are allowed. The influence of the classification errors are parametrised with the parameter  $C$ . A larger  $C$  penalizes a wrong classification more strongly.

#### 2.3.4. Multi-class Classification

In contrast to decision tree and  $k$ -nearest neighbour classifiers, support vector machines handle binary problems. There are several methods to extend them for multi-class classification effectively by combining different binary classifiers. Some methods consider multiclass problems at the expense of much larger optimisation problems with expensive computation. In the used LIBSVM algorithm [Chang and Lin (2007)] the ‘one against one’

approach [Knerr et al. (1990)] is applied in which  $k(k-1)/2$  classifiers are constructed and each one trained with data of two classes. The classification decision is based on a majority vote of the class assignments. If classes have identical votes the one with the smallest index is selected.

### 3. Results and discussion

Two models, one with three classes according to the species (plus noise) and another with subclasses for the species, were trained for each of the three classifiers based on the training data. The inner evaluation of the classifiers performances was carried out by cross-validation. Finally, we apply the learned model on unseen data samples and evaluated the prediction performance using a manually classified set of images.

The classifiers were initially trained with three classes, one for each species and noise (BRSNN, HORVS and NOISE). A cross-validation was used to assess the performance of each classifier. All three classifiers show a good performance (cross-validation: more than 98% correctly classified objects) to separate the three species from each other.

Table 1: Comparison of the classifiers for the three classes (species) case: the  $R^2$  values for each class are used as performance measure.

Classifier\Class	BRSNN	HORVS	NOISE
Decision Tree	0.96	0.97	0.97
SVM	0.95	0.96	0.97
kNN	0.97	0.96	0.98

#### 3.1. Comparison to manual classification

A subset of the images (68) was selected and the number of weed and crop plants in each image were manually counted. This is the reference data set, as the number of plants is used to generate a management decision (spray or not). The classification result was assembled for these images to get the number of weed and crop plants per image. These numbers can then be compared to the manually assessed reference data.

The comparison between the classification results and the manually counted planes yields a systematic overestimation of the BRSNN and HORVU classes. This is due to the fact that after the image processing (2.2) the plants are split into parts, especially the oil seed rape plants (BRSNN) were often split into two separate germination leaf objects. On the other hand, in images with a very high number of oil seed rape the number of plants is underestimated due to overlaps of plants which leads to segments which contain more than one plant in the image processing step. These problems can be taken into account by using a more detailed model for classification by introducing new classes for single leaves and overlaps of plants, as it is done in the following section.

## 3.2. Classification with subclasses

In order to account for the error of the split objects, training data was created with separated classes (subclasses) for each species. Two growth stages and single leaf classes were added as subclasses.

The classification using the three classifiers led to three confusion matrices, which can be compared in Table 2. The submatrices for the species are marked and allow for a comparison of intra- and interclass errors. All three classifiers reach high correct classification rates for the inter-class distinction of the different species (and noise): the submatrices (B\*–H\*, lower left and upper right) contain a maximum of one wrongly classified object. By weighting the subclasses (0.5 for dictyledonous leaves) there is no overestimation any more, since the automatically number of plants correspond to the manually counted ones.

The introduced subclasses can also be used to show differences between the classifiers. They differ in the classification of intra-species classes (H\*–H\*, lower right submatrix). The  $k$ -NN classifier and decision trees cannot adequately separate the subclasses of HORVS, because the model complexity is not sufficient. The reason why decision tree learning cannot adequately differentiate between the subclasses is caused by regarding only one feature at a time during division into two subsets. Thereby the aforementioned boundary of axis parallel parts is obtained and samples which are not differentiable by this way are not adequately split.

In such cases SVMs have advantages, since they can construct nonlinear separation functions in the feature space. Using a radial basis function kernel to build non-linear class boundaries, a separation of the different subclasses of HORVS is possible. The SVM classifier still provides good classification rates for the difficult-to-separate subclasses, which is an important property and qualifies this algorithm for other datasets containing more weed species.

## 4. Conclusion

Image processing of bi-spectral images was used for the detection of weed and crop densities based on shape features. Three classification algorithms were compared by using manually derived densities for oil seed rape and barley. A simple class scheme, one for each species and noise, can be classified correctly by all compared classifiers ( $k$ -nearest neighbours, decision tree learning and support vector machines).

Due to oversegmentation or undersegmentation more classes were introduced (here: subclasses of the species) to take into account single leaf image segmentation and overlaps of plants. In this case the performance of the three classifiers varies. The model complexity of  $k$ -NN and decision tree classifiers is not sufficient to separate the subclasses of HORVS. SVMs comprise a complex function class and construct an adequate model. Thus SVMs can adopt better to the more complex situations with different species, which is of practical relevance for site-specific herbicide variation. In future, this type of classifier will be applied to image series with more different species of weeds and crops.

Table 2: Confusion matrix for three classifiers, the overall accuracy values are 86% (SVM), 83% (*k*NN) and 78% (decision tree). The class abbreviations are: BL (BRSNN10L), BN (BRSNN10N), NL (NOISE00L) NX (NOISE00X), H2L (HORVS12L), H0N (HORVS10N), H2N (HORVS12N), H2O (HORVS12O),

↓predicted	true class								
<b>SVM</b>	BL	BN	NL	NX	H2L	H0N	H2N	H2O	precision %
BL	65	2	0	1	0	0	0	1	94
BN	0	44	0	0	0	0	0	0	100
NL	0	0	52	2	0	0	0	0	96
NX	0	0	1	31	0	0	0	0	97
H2L	0	0	0	0	31	11	1	5	65
H0N	0	0	0	0	9	33	6	0	69
H2N	0	0	0	0	0	4	17	0	81
H2O	0	0	0	0	1	0	0	8	89
%recall	100	96	98	91	76	69	71	57	
<b><i>k</i>NN</b>									
BL	65	2	0	1	0	0	0	0	96
BN	0	44	0	0	1	0	1	0	96
NL	0	0	51	2	0	0	0	0	96
NX	0	0	2	31	0	0	0	0	94
H2L	0	0	0	0	28	17	2	5	54
H0N	0	0	0	0	9	24	3	0	67
H2N	0	0	0	0	0	6	18	0	75
H2O	0	0	0	0	3	1	0	9	69
%recall	100	96	96	91	68	50	75	64	
<b>Decision tree</b>									
BL	64	4	0	1	0	0	0	1	91
BN	0	41	0	0	2	0	0	0	95
NL	0	0	51	2	1	0	0	0	94
NX	1	0	1	28	0	0	0	1	90
H2L	0	1	1	1	19	10	0	5	51
H0N	0	0	0	0	14	30	8	1	57
H2N	0	0	0	0	0	7	16	0	70
H2O	0	0	0	2	5	1	0	6	43
%recall	98	89	96	82	46	63	67	43	

## References

- Boser, E. B. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152. ACM Press.
- Brown, R. and S. Noble (2005). Site-specific weed management: sensing requirements – what do we need to see? *Weed Science* 53(2), 252–258.
- Chang, C.-C. and C.-J. Lin (2007). Libsvm: a library for support vector machines.
- Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning* 20(3), 273–297.
- Hu, M. K. (1962, Feb.). Visual pattern recognition by moment invariants. *IRE Transactions Information Theory* 8(2), 179–187.
- Knerr, S., L. Personnaz, and G. G. Dreyfus (1990). Single-layer learning revisited: a step-wise procedure for building and training a neural network. In J. Fogelman (Ed.), *Neuro-computing: Algorithms, Architectures and Applications*. Springer-Verlag.
- Mierswa, I., M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler (2006, August). Yale: Rapid prototyping for complex data mining tasks. In L. Ungar, M. Craven, D. Gunopulos, and T. Eliassi-Rad (Eds.), *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*, New York, NY, USA, pp. 935–940. ACM.
- Oebel, H. and R. Gerhards (2005). Site-specific weed control using digital image analysis and georeferenced application maps – first on-farm experiences. In *5th ECPA*, Uppsala, pp. 131–138.
- Pérez, A., F. López, J. Benlloch, and S. Christensen (2000, Feb). Colour and shape analysis techniques for weed detection in cereal fields. *Computers and Electronics in Agriculture* 25, 197–212.
- Quinlan, R. J. (1993). *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Sökefeld, M., R. Gerhards, H. Oebel, and R.-D. Therburg (2007, June). Image acquisition for weed detection and identification by digital image analysis. In J. Stafford (Ed.), *Precision agriculture '07*, Volume 6, The Netherlands, pp. 523–529. 6th European Conference on Precision Agriculture (ECPA): Wageningen Academic Publishers.
- Slaughter, D. C., D. K. Giles, and D. Downey (2008). Autonomous robotic weed control systems: A review. *Comput. Electron. Agric.* 61(1), 63–78.

Sui, R., J. A. Thomasson, J. Hanks, and J. Wooten (2008, Jan). Ground-based sensing system for weed mapping in cotton. *Computers and Electronics in Agriculture* 60, 31–38.

Vapnik, N. V. (1982). *Estimation of dependences based on empirical data*. New York, NY: Springer.

## A. Erratum

The confusion table in the published version had errors (Table 2), the correct version can be found in this version. The result and discussion still holds, although the difference between the performance of  $k$ NN and SVM classifiers is small. SVMs reach the best recall values (true positive rate), indicating that they have the best ability of the three classifiers to discriminate the subclasses.