Pixel-Based Classification Method for Detecting Unhealthy Regions in Leaf Images

Satish Madhogaria[1,2], Marek Schikora[1,3], Wolfgang Koch[1,2], Daniel Cremers[3]

[1] Dept. Sensor Data and Information Fusion, Fraunhofer FKIE, Wachtberg, Germany
{satish.madhogaria, marek.schikora, wolfgang.koch}@fkie.fraunhofer.de

[2] Dept. of Computer Science, University of Bonn, Germany

[3] Dept. of Computer Science, Technical University of Munich, Germany
daniel.cremers@in.tum.de

Abstract: In this paper, we present a pixel-based, discriminative classification algorithm for automatic detection of unhealthy regions in leaf images. The algorithm is designed to distinguish image pixels as belonging to one of the two classes: healthy and unhealthy. The task is solved in three steps. First, we perform segmentation to divide the image into foreground and background. In the second step, support vector machine (SVM) is applied to predict the class of each pixel belonging to the foreground. And finally, we do further refinement by neighborhood-check to omit all falsely-classified pixels from second step. The results presented in this work are based on a model plant (*Arabidobsis thaliana*), which forms the ideal basis for the usage of the proposed algorithm in biological researches concerning plant disease control mechanisms.

1 Introduction

During last years, image classification tasks have found tremendous appreciation in biological researches where a number of tasks are being simplified with the help of automated image classification [WKJK98, CZE03, KSC+95]. Plant diseases need to be controlled not only to maintain the quality of food produced by growers around the world but also to reduce food-borne illnesses from infected plants [SCCH08]. Thus, automatic extraction of unhealthy regions in leaf images is useful for various biological research based on disease control mechanisms [HHOJ10, DRBM09]. There is a wide variety of plant diseases caused by environmental factors (nutrition, moisture, temperature, *etc.*) or by organisms (fungi, bacteria, viruses) that attack plants but in most cases one common symptom is changes in the color intensities in the infected regions of leaves. A good color variation model can be employed to distinguish healthy and unhealthy regions in leaf images. A probabilistic algorithm, employing a Gaussian mixture model (GMM) and a Bayesian classifier for classifying disease symptoms in *Arabidopsis* plants was presented in [SSK+10]. The results from Bayes-like classifiers can be inaccurate, because the estimation of a robust GMM is not always possible from real data. To overcome these limitations we propose

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

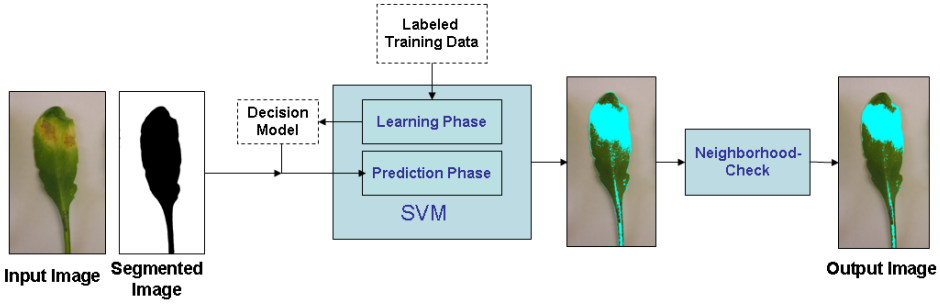


Figure 1: Overview of the proposed algorithm. Input image is the (Arabidopsis Thaliana) leaf with monochromatic background. First, segmentation method is applied to obtain the pixels belonging to leaf. Second, each pixel belonging to leaf is classified using linear SVM classifier. Finally, the output from classifier is further refined through neighborhood-check method to obtain the output image.

here a different classification strategy. The algorithm described in this paper uses color feature space as input to a well-known machine learning algorithm (Support Vector Machine (SVM)) to classify the pixels of a leaf. Figure 1 presents an overview of the steps described in this paper. First a segmentation method, described in section 2, is applied to obtain a binary image with only foreground and background information. Each pixel belonging to the foreground region is then given as an input to a linear SVM classifier (described in section 3) for predicting the class to which it belongs. After all the pixels belonging to the foreground are classified, the neighborhood information is used to alter the result of pixels classified as unhealthy, but does not match with the visual perception of an human observer. The neighborhood-check method is described in section 4.

2 Segmentation

The input is a leaf image with almost monochromatic background. First, we need to separate the pixels belonging to a leaf (foreground) and not belonging to the leaf (background) in the input image. Besides reducing the computational cost in the next step, a good segmentation method can also improve the overall result by eliminating any misclassification outside the leaf boundary. Therefore, we divide the image into foreground and background so that only the pixels belonging to the foreground are considered for classification in the next step.

The binary segmentation of an image $I : \Omega \to [0,1]^3 \subset \mathbb{R}^3_1$ with $\Omega \subseteq \mathbb{R}^2_1$ can be seen as separation of the image plane Ω into disjoint regions Ω_{obj} and Ω_{bgd}, with $\Omega = \Omega_{\text{obj}} \cup \Omega_{\text{bgd}} \cup \Gamma$, where Γ denotes the contour of the segmentation. So we are looking for a binary image $u : \Omega \to \{0,1\}$. The most influential region based image segmentation model was introduced by Mumford and Shah in 1989 [MS89]. Many models based on this functional and its derivatives have been proposed, e.g. [CV01, UPCB08]. In this work we use the segmentation method proposed in [SSK+10]. The method uses a convex energy functional

[SHRW09] but with the I1I2I3 color space [Haf99] instead of HSV. Following [SHRW09] a convex energy functional in the I1I2I3 color space can be written as:

$$E(u, \boldsymbol{\mu}_{\text{obj}}, \boldsymbol{\mu}_{\text{bgd}}) = \int_{\Omega} \left(f(I_{123}(\mathbf{x}), \boldsymbol{\mu}_{\text{obj}}) - f(I_{123}(\mathbf{x}), \boldsymbol{\mu}_{\text{bgd}}) \right) u(\mathbf{x}) \mathrm{d}\mathbf{x} + \lambda \int_{\Omega} |\nabla u(\mathbf{x})| \mathrm{d}\mathbf{x}, \tag{1}$$

with

$$f(I_{123}(\mathbf{x}), \boldsymbol{\mu}) = w_1([I_{123}(\mathbf{x})]_{\text{I1}} - \boldsymbol{\mu}_{\text{I1}})^2 + w_2([I_{123}(\mathbf{x})]_{\text{I2}} - \boldsymbol{\mu}_{\text{I2}})^2 + w_3([I_{123}(\mathbf{x})]_{\text{I3}} - \boldsymbol{\mu}_{\text{I3}})^2 \tag{2}$$

denoting a weighted squared sum of the individual channels. For the results presented in this paper we use $w_{\text{I1}} = 0.1$ and $w_{\text{I2}} = w_{\text{I3}} = 0.45$. As an additional input we use mean values for the foreground $\boldsymbol{\mu}_{\text{obj}}$ and background $\boldsymbol{\mu}_{\text{bgd}}$ and a smoothing parameter $\lambda \in \mathbb{R}$. $[I_{123}(\mathbf{x})]_{\text{I}n}$ denotes the value of pixel $\mathbf{x}$ for the color channel In. The desired segmentation is a binary image $u : \Omega \subseteq \mathbb{R}^2 \rightarrow \{0, 1\}$. We minimize (1) using successive over-relaxation (SOR), as in [SHRW09] to estimate the optimal u.

3 SVM classification

Having obtained a binary image $u : \Omega \subseteq \mathbb{R}^2 \rightarrow \{0, 1\}$, we now want to classify each pixel belonging to Ω_{obj} into unhealthy and healthy regions. For this purpose we use a state-of-the-art machine-learning algorithm, support vector machine (SVM), that have found wide acceptance in recent years due to its ability to classify linear and non-linear data. SVMs have been applied with great success in many challenging classification problems processing large data sets. The basic concept was introduced in [CV95]. It is based on learning from examples, which means, it requires a separate set of training and testing data. The training algorithm builds a model that predicts the class of unknown input data.

We need a labeled training data, which serves as an input to the learning function. Like many other pixel-based classification methods, we exploit the color variation property of image co-ordinates in order to form a decision model. Since the components of I1I2I3 color space [Haf99] are uncorrelated, statistically it is the best way to detect color variations. While I1 contains the illumination information, I2 and I3 mainly contains color information. Hence, we use only I2 and I3 in order to provide invariance to illumination changes. Thus the training data comprises of 2D color values, selected from healthy and unhealthy leaf images and labeled into two different classes.

3.1 Training phase - offline

Suppose we have L number of training vectors belonging to two different classes, $(\mathbf{x}_i, y_i)$ where $i = 1, ..., L$ and y_i is either 1 (healthy) or -1 (unhealthy), indicating the class to which $\mathbf{x}_i$ belongs. SVM is based on the concept of finding a hyperplane which can be

ISBN 978-3-88579-286-4

described by a set of points satisfying the equation:

$$\mathbf{w} \cdot \mathbf{x} + b = 0, \quad \mathbf{w} \in \mathbb{R}^n,\ \mathbf{x} \in \mathbb{R}^n,\ b \in \mathbb{R} \tag{3}$$

where $\mathbf{w}$ is normal to the hyperplane and $b/||\mathbf{w}||$ is the perpendicular distance from the hyperplane to the origin. The goal here is to choose $\mathbf{w}$ and b so as to maximize the margin

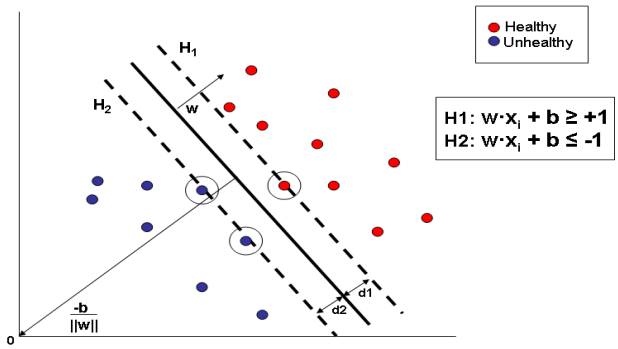


Figure 2: Hyperplane through two linearly separable classes. Points on the hyperplanes are called support vectors and forms the basis for predicting the class of unlabeled data

between two parallel hyperplanes H1 and H2 (see Figure 2). Thus, our training data can be described by equation:

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1 \geq 0\ \forall_i \tag{4}$$

The training part of SVM algorithm finds a $\mathbf{w}$ that leads to the largest b. It can be solved by finding the solution of following optimization problem:

$$\min \frac{1}{2}||\mathbf{w}||^2 \quad \text{such that} \ \ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0\ \forall_i \tag{5}$$

It is transformed into its dual form by using lagrangian formalisation:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{L} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] \tag{6}$$

where α_i are non-negative lagrange multipliers. According to [Bur98], the final dual optimization problem can be written as:

$$\begin{aligned} \text{maximize } L_D &= \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_i \\ \text{subject to} \sum_i \alpha_i y_i &= 0 \text{ and } \alpha_i \geq 0 \ \ \forall_i \end{aligned} \tag{7}$$

Note that the dual form requires only the dot product of each input vector $\mathbf{x}_i$ to be calculated, which is important for the kernel trick. Equation 7 is a convex optimization problem

ISBN 978-3-88579-286-4

and QP (Quadratic Programming) solver is run on it to obtain α, from which we can get $\mathbf{w}$:

$$\mathbf{w} = \sum_{i=1}^{L} \alpha_i y_i \mathbf{x}_i \tag{8}$$

The training cases with $\alpha_i > 0$ are called support vectors which are situated in the support hyperplane and they determine the solution. Any data point which is a support vector will have the following form:

$$y_s(\mathbf{w} \cdot \mathbf{x_s} + b) = 1 \tag{9}$$

Using any support vector, b can be derived from equations 8 and 9 (see [Bur98, JST04] for detailed derivation):

$$b = \sum_{s \in S} (y_s - \sum_{m \in S} \alpha_m y_m \mathbf{x}_m . \mathbf{x}_s) \tag{10}$$

Where S denotes the set of indices of the support vectors. S is determined by finding the indices i where $\alpha_i > 0$. Instead of using an arbitrary support vector $\mathbf{x}_s$, it is better to take an average of the support vectors in S. Thus, the training phase of SVM gives $\mathbf{w}$ and $\mathbf{b}$ which is used later to compute the class of unknown vectors. Since the training phase is time consuming, it is done offline.

3.2 Prediction phase - online

In the prediction phase, all pixels labeled as foreground pixel in the segmentation step (Section 2) are classified into one of the two classes - healthy or unhealthy. Each new pixel, $\mathbf{x}'$ is classified by evaluating:

$$y' = \text{sign}(\mathbf{w} \cdot \mathbf{x}' + b) \tag{11}$$

where $\mathbf{w}$ and b are obtained from the training part of the SVM algorithm.

Although, using binary SVM gives good performance in most of the cases but it still relies on a good segmentation method in step 2, which means that if there are pixels labeled as foreground outside the boundary of the leaf then the SVM should also classify them into one of the two classes. As an example in figure 3, we can see that due to error in segmentation, there are pixels outside the leaf region marked as unhealthy. Segmentation error occurs when there is a prominent shadow of the leaf in the image due to which the proposed segmentation method labels pixels inside the shadow region as foreground. To make the SVM classifier more efficient we can classify each pixel into one of the three classes: healthy, unhealthy and background. Though, inherently, SVMs are binary classifiers but it is easily possible to do a multi-class classification with SVMs by building a set of one-verses-one classifiers and choose the class that is selected by most classifiers. Figure 3 compares the result with two-class and three-class SVM.

(a) Binary SVM

(b) 3-class SVM

Figure 3: Top image shows output from binary SVM classifier, where unhealthy pixels outside the leaf boundary are noticeable. This is due to prominent shadow near the leaf boundary which is labeled as foreground pixels in the segmentation step. We can overcome this problem by using a multi-class SVM, where each pixel is classified into three classes - healthy, unhealthy and background.

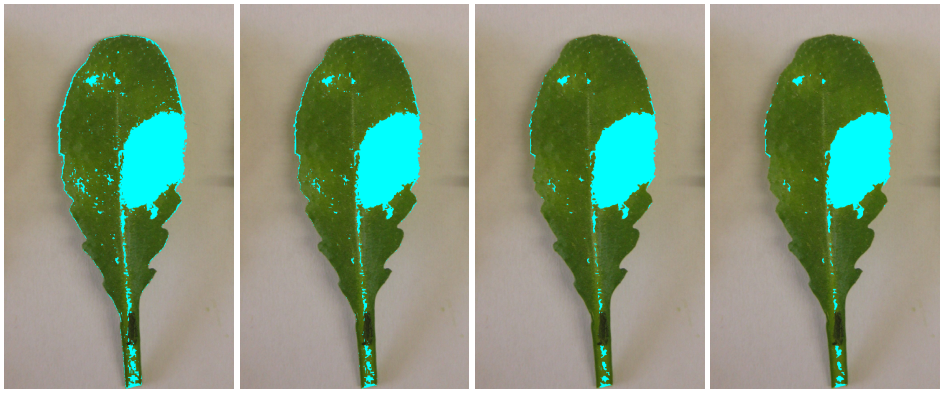
4 Neighborhood-Check

Output from the classification step (Section 3) shows a high number of isolated pixels labeled as unhealthy, which maybe be perceived by human eye as healthy. This is due to the fact that each each pixel is too small for a human eye to see and usually they see the combinations of pixels. There could be a pixel within an healthy region that have similar color values as the one from infected region which makes the classifier to mark it as unhealthy. Here, we exploit the fact that usually the infected regions should be densely populated with infected pixels. We can, therefore, use the neighborhood classification information to alter the result of isolated pixels, classified as unhealthy. This step works as follows: For each $(\mathbf{x}_i, y_i)$ with $y_i = -1$ (unhealthy), define the number of pixels which are

classified as unhealthy in the neighborhood radius $n \in \mathbb{Z}$ as c_i. We perform the following:

$$\text{if } c_i < \frac{(2n+1)^2 - 1}{2}, \text{ then set } y_i = +1 \text{ (healthy)} \tag{12}$$

We used $n = 2$ to obtain the results presented in this paper. In Figure 4, we can see the effect of using n = 1, 2 and 3. It further strengthen the fact for choosing n = 2 since using neighborhood radius of 1 slightly improves the result from SVM classifier but not as good as using 2 or 3. Although neighborhood radius of 2 or 3 shows almost the same effect but we choose n = 2 to reduce the computational cost. Figure 5 shows another example where the result from step 2 could be improved remarkably with the help of neighborhood-check.

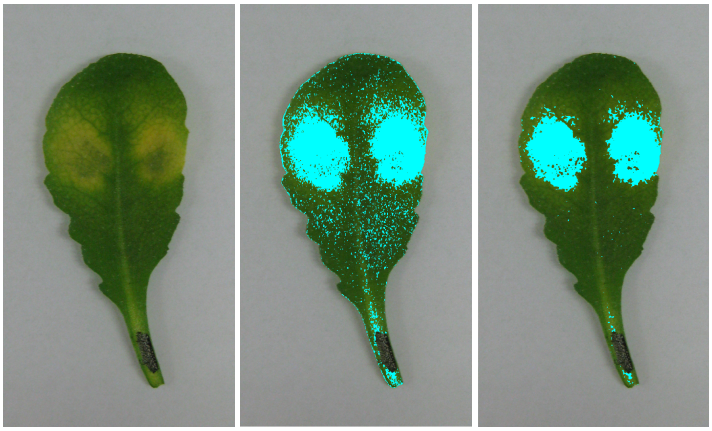


(a) Result from SVM classifier (b) n=1 (c) n=2 (d) n=3

Figure 4: Neighborhood radius could be varied to obtain better result. We can see from the figure that neighborhood radius, n = 2 and n = 3 yields almost the same result. Using n=1 improves the result from SVM classifier, (a) but not as good as (b) and (c).

5 Results

The algorithm have been tested extensively on more than 500 infected leaf images. The testing images were obtained from biological experiments done in University of Gießen. The plants were infected with *Salmonella* bacteria and observed through time to investigate how the plant reacts and defends itself against the disease. The input images are the infected leaf images with monochromatic background and the output is the classified image with marked unhealthy region. It also provides an objective measurement for the disease rate. Figure 6 shows some outputs from the classification algorithm described above. The results obtained from this algorithm were convincing and could be easily used for such biological experiments. Figure 7 shows a comparison between the proposed and a probabilistic method [SSK+10]. We extended the probabilistic algorithm with the



(a) Input image (b) Before (c) After

Figure 5: (b) is the output from SVM classifier. It shows high number of pixels marked as unhealthy while the human eye perceive them as healthy. In an attempt to alter the result of those isolated pixels, neighborhood-check method is applied. (c) is the result from neighborhood-check and matches well with the visual perception of human observer.

proposed neighborhood-check to have a fair comparison. It can be easily seen that the proposed algorithm, which combines the accuracy of SVM with a neighborhood-check method, outperforms the probabilistic method. From the figure we can see that some unhealthy region in leaf are left unmarked by Bayesian classifier. Besides there are some marks near the boundary of the leaf which are wrongly classified as unhealthy. This problem is overcome by using multi-class SVM. SVMs are more robust in separating the data whereas with Bayes-like classifiers the estimation of robust Gaussian Mixture Model is not always possible with real data. Experiments prove that higher accuracy could be achieved with SVM. Here, we use linear SVM because it is computationally efficient and avoids the complexities of tuning several parameters, which is the case with non-linear kernels.

6 Conclusion

An automatic pixel-based classification method for detecting unhealthy regions in leaf images is presented in this paper. The method has been tested extensively and promising results were obtained. Linear SVM has been used to classify each pixel. We have also shown how the results from SVM could be improved remarkably using the neighborhood-check technique. The proposed method is compared to the existing method and it is concluded that higher accuracy can be achieved with this method. The algorithm can be used in biological researches based on plant disease epidemiology and could well be extended for other detection tasks which also mainly rely on color information, but extension to other features is easily possible.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

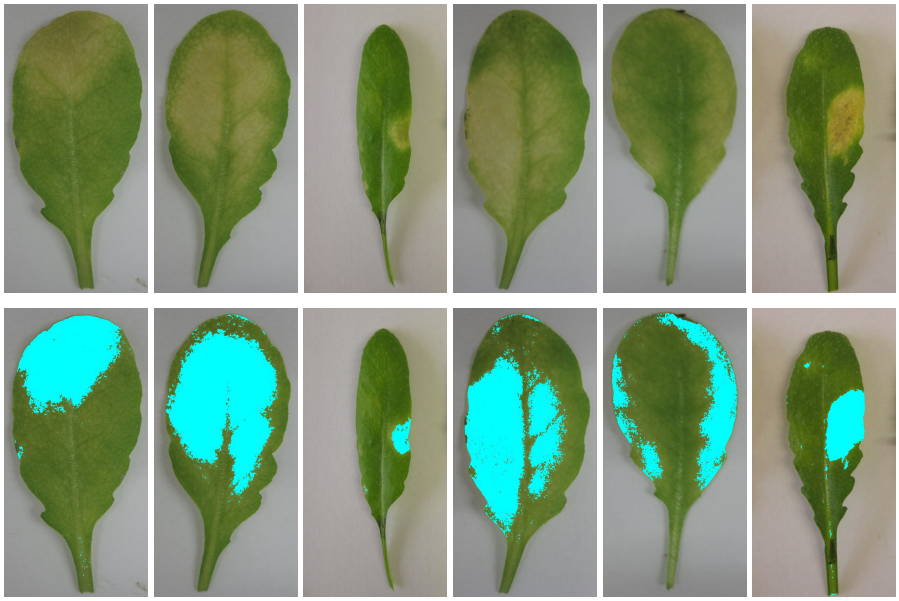


Figure 6: Classification Results - Top row shows input images and the bottom row shows outputs from the proposed classification algorithm

Acknowledgement: We would like to thank Dr. Adam Schikora and his team in University of Gießen for providing us with the leaf images and helping us understand the problem.

References

[Bur98] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[CV95] C. Cortes and V. Vapnik. Support Vector Networks. In *Machine Learning*, volume 20, pages 273–297, 1995.

[CV01] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[CZE03] Chris A. Cocosco, Alex P. Zijdenbos, and Alan C. Evans. A Fully Automatic and Robust Brain MRI Tissue Classification Method. *MEDICAL IMAGE ANALYSIS*, 7(4):513–527, 2003.

[DRBM09] Gloria Díaz, Eduardo Romero, Juan R. Boyero, and Norberto Malpica. Recognition and Quantification of Area Damaged by Oligonychus Perseae in Avocado Leaves. In *Proceedings of the 14th Iberoamerican Conference on Pattern Recognition: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, CIARP '09, pages 677–684, Guadalajara, Jalisco, Mexico, 2009. Springer-Verlag.

[Haf99] W. Hafner. *Segmentierung von Video-Bildfolgen durch Adaptive Farbklassifikation*. PhD thesis, Computer Science Department, TU Munich, 1999.

[HHOJ10] H. Hashim, M. A. Haron, F. N. Osman, and S. A. M. Al Junid. Classification of Rubber Tree Leaf Disease Using Spectrometer. In *Proceedings of the 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, pages 302–306, Washington, DC, USA, 2010. IEEE Computer Society.

[JST04] Nello Cristianini John Shawe-Taylor. Kernel Methods for Pattern Analysis. New York, NY, USA, 2004. Cambridge University Press.

[KSC+95] M. Kamber, R. Shinghal, D.L. Collins, G.S. Francis, and A.C. Evans. Model-based 3-D segmentation of multiple sclerosis lesions in magnetic resonance brain images. *Medical Imaging, IEEE Transactions on*, 14(3):442 –453, sep 1995.

[MS89] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[SCCH08] A. Schikora, A. Carreri, E. Charpentier, and H. Hirt. The dark side of salad: Salmonella typhimurium overcomes the innate immune response of Arabidopsis thaliana and shows an endophatogenic lifestyle. *PLoS ONE*, 3(5):e2279, 2008.

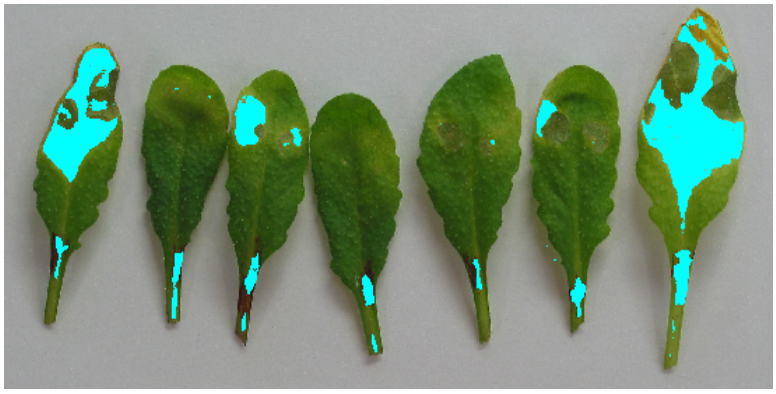
[SHRW09] M. Schikora, M. Häge, E. Ruthotto, and K. Wild. A convex formulation for color image segmentation in the context of passive emitter localization. In *12th International Conference of Information Fusion*, pages 1424–1431, July 2009.

[SSK+10] M. Schikora, A. Schikora, K.-H. Kogel, W. Koch, and D. Cremers. Probabilistic Classification of Disease symptoms caused by Salmonella on Arabidopsis Plants. In *GI Jahrestagung (2)'10*, pages 874–879, 2010.

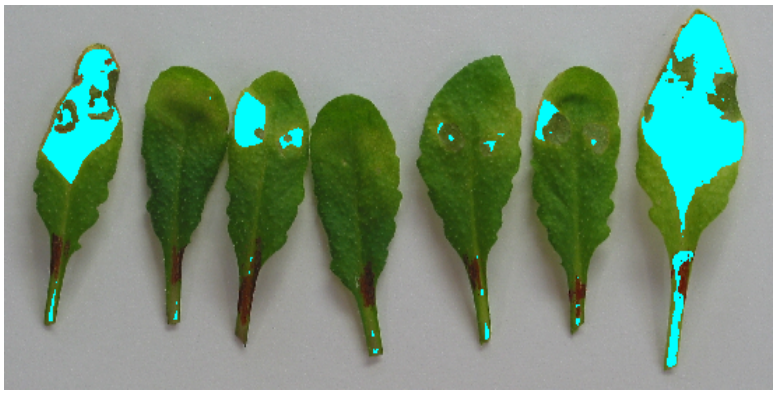
[UPCB08] M. Unger, T. Pock, D. Cremers, and H. Bischof. TVSeg - Interactive Total Variation Based Image Segmentation. In *British Machine Vision Conference (BMVC)*, September 2008.

[WKJK98] Simon K. Warfield, Michael Kaus, Ferenc A. Jolesz, and Ron Kikinis. Adaptive, Template Moderated, Spatially Varying Statistical Classification. In *Medical Image Analysis*, pages 43–55. Springer, 1998.

(a) Input image



(b) Bayesian Classifier



(c) SVM Classifier

Figure 7: Example image showing result from probabilistic and SVM classification. Difference is clearly noticeable in the right-most leaf in the image, where portions in the upper half is left unmarked by Bayesian classifier. Also, the pixels outside the leaf boundary (see second from right in (b)) are marked which can be avoided by using multi-class SVM classifier. Higher accuracy can be achieved by using SVM in the second step.