Application of complex networks in the automatic classification of damaging agents in soybean leaflets

Eduardo Severino Mapa, Kayran dos Santos, Thiago Luis Guimarães de Souza e David Menotti
Departamento de Computação - Universidade Federal de Ouro Preto
Campus Universitário, Morro do Cruzeiro, CEP 35400-000, Ouro Preto, Minas Gerais, Brasil
eduardomapa@gmail.com,kayransantos2@gmail.com,thiagoluis13@gmail.com, menottid@gmail.com

Abstract

Many of the difficulties in managing soybean crops are related to the identification of insect/pests harmful to the cultivars, since crops can be attacked by a wide range of such agents. By identifying the most common agents that cause damage to the leaves (e.g., beetles, caterpillars, etc.), we get more knowledge about appropriate strategies of control. The proposed work develops an automatic classification method of the main agents that cause damage to leaves. Acquired images are preprocessed and the contours of the damage taken. Each one of the contours will be modeled by complex networks. The network connectivity is measured to compose the feature vectors which discriminates these damages. The results of automatic classification using LDA and SVM will be reviewed after obtaining the characteristics.

1. Introduction

Pest control in crops soybean is a decision-making system which result of action by differents types of insect/pests on cultivars, as well as the frequency that occurs each one of these agents.

In the most cases, the choice of control occurs in unplanned way, usually by the excessive use of the chemicals products where, according with Picano & Guedes *et al.* [11], it results in damage, environmental pollution and toxicity to humans

The identification of the most common occurrences (*e.g.* beetles, caterpillars, *etc.*) together with determination of the frequency which each agent occurs in leaflets, it help the choice of an efficient control strategy. According [10], the current methods for identification and sampling frequency based on the "Cloth beat" technique which a cloth is extended between two parallel rows of cultivars, then plants are bent over the cloth and the insects fall in. For purposes of sampling, these insects will be identified and counted where it will be estimate the rate of the each one these agents occurs. This technique shows a method of sampling low rate and costly over time.

Under the guidance of experts in plant of the UFV shown in [12], there is a possibility of distinguishing between two main classes of agents (*i.e.*, beetles, caterpillars), through of shap (contour, area, *etc.*) of the damage caused by each one on the leaflets. It can be seen in the leaflet as exemplified by Figures 1(a) and 1(b).

(a) (b)

Figure 1. Clippings in soybean leaflets: (a) damage attributed to a beetle, (b) damage attributed to a caterpillar.

In the current literature, there are automatic classification techniques of diseases related to cultivars in both soybean and several cultures, it cited by the authors [3] and [4]. But there is no method on the characterization of the insects/pests that act on these crops.

This paper describes the development of a method to automatic classification of these agents by means digital images, and will complement other method proposed in [8] detection and automatic quantification of leaf area damaged soybeans. This will make possible the choice an efficient control strategy, which will reduce financial losses and the environment caused by the excessive use of pesticides.

The following section will detail the methodology, since acquisition and preprocessing of the samples (2.1), through modeling of contours and feature extraction using complex networks (2.2) to the stage of comparasion between to classifier LDA(*Linear Discriminant Analysis*) and SVM (*Support Vector Machine*).

2. Methodology

2.1. Aquision and preprocessing of the images

Researchers from Experimental Field Bacuri (UFV) help us to obtain nearly 180 samples of various leaflets soybean cultivars, which were digitized by scanner and stored in the file format *bitmap* (bmp) with resolution of 200 dpi (*"Dots per inch"*).

A preprocessing including two adapted techniques presented by Nazaré-Jr [8] was applied in the leaflets to raise theirs regions corresponding to the damage on the image.

The first technique consists in remove the shadow of the image, through a conversion from RGB color space (R-*red*, G-*green*, B-*blue*) for the HSV (H-*hue*, S-*saturation*, V-*value*) color system. According to Nazareth-Jr *et al.* [8], to convert the image to the HSV color space, only the value of the channel H(*hue*) is sufficient for detection and elimination of the region that represents the leaflet shadow in the image.

The second technique consists in eliminate the waste outside the leaflets acquired while scanning the image. This technique can be subdivided into three steps:

1) *Image Segmentation* This segmentation consists in detection of the appropriate threshold using the Otsu [7] algorithm, which define the region of interest (actual leaflet). The remaining samples will be considered as background, so that the result of this segmentation will be an image, where all region considered as background will have white ($R = G = B$ =255) *pixels*, and the region of interest will have black ($R = G = B = 0$) *pixels*. But the image still contains waste outside the leaflet, as shown in Figure 2.

Figure 2. Result of step 1 - Image Segmentation

2) *Waste Removal:* After the segmentation, next step is to apply a labeling algorithm [9]. It will determine the largest connected region in segmented image. Then, we reject all other regions and this way all wastes outside the leaflet are removed. The result is presented in Figure 3.

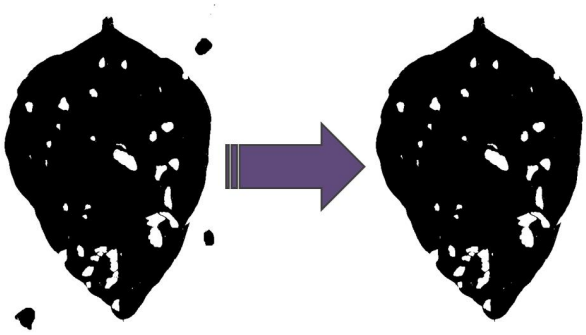

Figure 3. Result of step 2 - Waste Removal

3) *Damage Segmentation:*

With the image 4 that represent the reconstituted leaflet, in other words, without damage, we will target damages through intersection between the image obtained in step 2 and the image 4 with their levels inverted. As shown in 5:

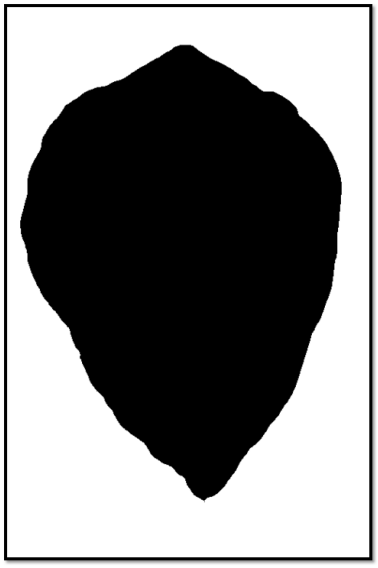

Figure 4. Image reconstituted leaflet

4) *Discard minor damage to 0.02% of the leaflet:*

In order to reduce the amount of damage that we believe are less significant during the evaluation, we decided that minor damage to 0.02% of the image it will not be considered. The method used to eliminate such damage consists in identify objects (damage) that have

Figure 5. Result of step 3 - Damage Segmentation

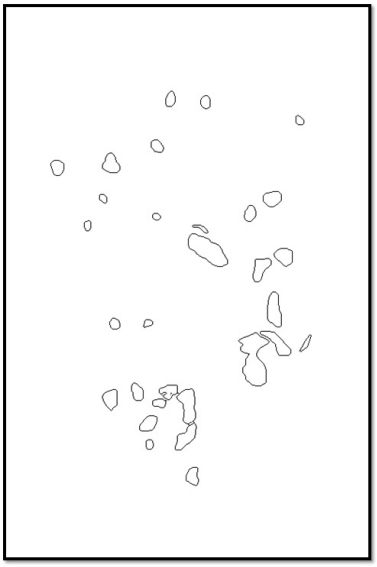

Figure 6. Result of step 5: Contour Detection of damage

less than 0.02% in relation to the size of image and apply to their *pixels* white (R=G=B=255) color which will be part of the background.

5) *Detect edge damage:*

According to Canny [5], the contour detection process is used to simplify the analysis of images, dramatically reducing the amount of data to be processed, while at the same time preserving useful information about limits of structure of the object. After the step 4, we used the contour detector algorithm proposed by Canny [5] in the samples of the leaflets. This result is presented in Figure 6, which will be used in the preprocessing of damage separation.

6) *Separate damage*

At this stage of preprocessing, each localized damage in Figure 6, will be separated so that each individual is an image representing an injury. After the step *Detect edge damage*, each object (damage) is selected. Then, the method is applied to eliminate the area outside the object, this method detects the minimum and maximum horizontal and vertical features that the object have and subtract the original image, thus a new image is cut originated with the dimensions of the damage.

2.2. Modeling contours by complex networks

2.2.1 Complex Networks

According to [2] the area of complex networks can be viewed as an intersection of two other important areas, graph theory and statistics.

In the actual literature we can find applications of complex networks in several areas of the science computer, such as display [1] and [6] that shape texts and texture images through complex networks. This study used this approach during modeling of the forms in contours that will be classified, as apply by [2] who also models in this way contours for classification of images.

The modeling presented in [2] is based in a model of Watts-Strogatz network [13]. This model has two interesting properties, first all vertices can be reached by any other with a small number of edges, second is the high number of cycles minimum (ie size 3) that they are formed. These properties are defined as small-world properties.

Feature extraction is carried out a development momentum through a growing network limited by a threshold sequence.

2.2.2 Construction of Complex Network

In order to model the outline of an image through a complex network, we consider the contour of the image as a set of points $C = [p_1, p_2, \ldots, p_n]$ where p_i is a vector of components $[x_i y_i]$ representing each pixel belonging to the contour, where x_i and y_i are their coordinates. So the network will be built as a graph where each pixel (p_i) is a vertex and each edge will have weight determined by Euclidean distance between their vertex, *i.e.*,

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \qquad (1)$$

Then we obtain a matrix W of size $N \times N$ where N represents the number of vertex of the network. The matrix will be filled with weights of the edges calculated, *i.e.*:

$$w_{ij} = W([w_i, w_j]) = d(p_i, p_j) \quad (2)$$

Afterwards, the matrix values will be normalized by equation 3 in the interval $[0, 1]$, *i.e.*.

$$W = \frac{W}{\max w_{ij} \in W} \quad (3)$$

This way the network obtained is a regular network, since each vertex has an edge to every other (complete graph). A regular network has not interesting feature about our approach, then, we should realize a transformation to make this complex network, this transformation is described next step for feature extraction.

2.2.3 Dynamic Evolution

As shown in the previous section, the first step to the process of feature extraction is transform the regular network obtained in a complex network, then it will set a process of dynamic evolution according to a threshold T_l. Transformation δ will proceed as follows way, with each iteration l will be removed from all edges of the regular network whose weight is larger than T_l.

A adjacency matrix A will be obtained which will determine for each edge of the regular, if it exists ($a_{ij} = 1$) or not ($a_{ij} = 0$) in the new generated network, *i.e.*,

4:

$$A_{T_l} = \delta_{T_l}(W) = \begin{cases} a_{ij} = 0, & \text{se } w_{ij} \geq T_l \\ a_{ij} = 1, & \text{se } w_{ij} \leq T_l \end{cases} \quad (4)$$

Evolution of the network is ensured by use of several functions δ where the threshold T_l will be increased sequentially from a value of T_{inc}, as shown in the function 5 $f : T \rightarrow T$, where

$$\begin{aligned} &T_0 = T_{ini} | 0 < T_{ini} < 1, \\ &T_{l+1} = f(T_l) \text{se } T_l < T_Q < 1, \\ &f(x) = x + T_{inc} \end{aligned} \quad (5)$$

with T_{ini} and t_Q, thresholds initial and final respectively. This two functions define the dynamic evolution, with a finite number of variations in the complex network where it will be extracted features that we will describe below.

2.2.4 Connectivity

Connectivity k_i of a vertex i corresponds to the number of edges directly connected to this vertex, and can be obtained through of adjacency matrix A, *i.e.*,

$$k_i = \sum_{j=1}^{N} a_{ij} \quad (6)$$

First feature vector extracted for subsequent classification will be compound by connectivity descriptors presents in the several transformations of the network T_l. For each A_{T_l} obtained two values are calculated that describe connectivity of each network,*i.e.*, the average degree (k_μ) and the maximum degree (k_K) of their vertex, using respectively equations 7 and 12.

$$k_K = \max_i k_i \quad (7)$$

$$k_\mu = \frac{1}{N} \sum_{j=1}^{N} a_{ij} \quad (8)$$

As demonstrates by [2], although these features seem simple, they are rotation invariance and scale invariance through minor adjustments. The rotation invariance is obtained by normalization implemented in the matrix W in the interval [0,1]. On the other hand, scale invariance can be achieved for a normalization of k_i by the number of vertex (N) that compose the network, as shown in 9.

$$\forall k_i = \frac{k_i}{N} \quad (9)$$

The feature vector φ displayed below, it will be obtained by concatenating the values of average degree (k_μ) and maximum degree (k_K) obtained for each stage of network evolution, with the threshold at interval $[T_0, T_Q]$.

$$\varphi = [k_\mu(T_0), k_K(T_0), k_\mu(T_1), k_K(T_1), \ldots, k_\mu(T_Q), k_K(T_Q)]$$

2.2.5 Joint Degree

Besides connectivity, it is possible to examine some other characteristics on the complex networks. We can use Joint Degree that determining measures of correlation between the degrees of the vertices.

To determine the measures that compose the feature vector of Joint Degree, we must know the distribution of the probability $P(k_i, k')_i$, but in our approach we consider $(k_i = k')$, so as described in [2]. So, the distribution $P(k_i, k')_i$ represents the probability of a vertex i of degree k_i to be connected to another vertex of the same degree.

Thus, the features extracted of the Joint Degree will be: entropy, energy, and joint degree average. Which are described below:

1) *Entropy:* According to [2] *et al.*, entropy is directly related to the degree of order or disorder in a system. The calculation of entropy can be defined by the expression below:

$$H = -\sum_{i=1}^{N} P(k_i, k')_i log_2 P(k_i, k')_i. \quad (10)$$

2) *Energy:* Energy can be defined by the following expression:

$$E = -\sum_{i=1}^{N}(P(k_i, k')_i)^2. \quad (11)$$

3) *joint degree average:* This measure consists in discover two arbitrary nodes of the same degree in a network:

$$P = \frac{1}{N}\sum_{i=1}^{N} P(k_i, k')_i. \quad (12)$$

2.3. Classification

We used the linear classifier LDA to classify the damage as beetles damages and caterpillars damages .

The LDA research the best line of data separation in order to increase the distance between classes and decrease the distance intra class.

LDA calculates the centroids of the classes (μ_i) and the global centroid (μ). The calculation of centroids is done by averaging over the feature vectors. Over a threshold the feature vectors are the maximum and average degrees of network connectivity. With the centroids local, the datas are normalized subtracting them. After the covariance matrix of each class is calculated for the normalized data. The covariance matrix is calculated using the covariance matrix of the classes and their probabilities priori. With this matrix we can calculated the inverse of the covariance (C^{-1}). Then we can calculate the discriminant function for each class. This function is calculated by:

$$f_i = \mu_i C^{-1} x_k^T - \frac{1}{2}\mu_i C^{-1}\mu_k^T - ln(p_i) \quad (13)$$

Class of the damage that we want to classify is one where the function discriminant is maximized when the feature vector (x_k^T) of damage is given.

3. Experiments

To carry out experiments was used a base of 180 images of the leafs, preprocessed using the methods shown in Section 2.1. All damage taken images were saved in generating a base of 1700 damage.

The damage found were manually classified by the authors supervised learning as damage of beetles or caterpillars. This classification led to a 1000 base damage caused by caterpillars and 700 damage by beetles.

The threshold T_l used to generate the features was varied as shown in table 1.

T	t_0	t_{inc}	t_q	Number of threshold
T1	0.01	0.015	0.150	10
T2	0.01	0.035	0.350	10
T3	0.01	0.050	0.500	10
T4	0.01	0.065	0.650	10
T5	0.01	0.085	0.850	10
T6	0.025	0.075	0.875	12

Table 1. Used threshold

We used the Cross Validation to validade the tests where our base was separated in 10 partitions. Each partition contained 170 images of damage which 100 images of caterpillar and 70 images of Coleoptera. We made ten iterations. Each iteration a different partition was separated. The separated partition was used in tests and other partition was used as train of the system. These partitions were automatically generated using the software Matlab.

For each threshold were extracted features based in the degree connectivity and joint degree, and the amount of features for each threshold shown in Table 2

T	Connectivity	Joint Degree
T1	20	30
T2	20	30
T3	20	30
T4	20	30
T5	20	30
T6	24	36

Table 2. Amount of features for each threshold and different feature

For classification we used the LDA and SVM classifiers analyzed by cross-validation. The results obtained with LDA are presented in tables3 e 4

t0	tinc	ta	n Feature	hit test	% sucess (test)	hit train	%sucess(train)
0.01	0.0150	0.150	30	1520	89.41%	13771	90.01%
0.01	0.0150	0.150	20	1518	89.29%	13768	89.99%
0.01	0.0250	0.250	30	1522	89.53%	13765	89.97%
0.01	0.0250	0.250	20	1520	89.41%	13757	89.92%
0.01	0.0350	0.350	30	1517	89.24%	13762	89.95%
0.01	0.0350	0.350	20	1521	89.47%	13767	89.98%
0.01	0.0450	0.450	30	1521	89.47%	13765	89.97%
0.01	0.0450	0.450	20	1516	89.18%	13766	89.97%
0.01	0.0500	0.500	30	1521	89.47%	13757	89.92%
0.01	0.0500	0.500	20	1525	89.71%	13762	89.95%
0.01	0.0550	0.550	30	1516	89.18%	13760	89.93%
0.01	0.0550	0.550	20	1515	89.12%	13772	90.01%
0.01	0.0650	0.650	30	1525	89.71%	13766	89.97%
0.01	0.0650	0.650	20	1517	89.24%	13768	89.99%
0.01	0.0750	0.750	30	1519	89.35%	13774	90.03%
0.01	0.0750	0.750	20	1516	89.18%	13762	89.95%
0.01	0.0850	0.850	30	1525	89.71%	13768	89.99%
0.01	0.0850	0.850	20	1523	89.59%	13765	89.97%
0.01	0.0875	0.875	30	1518	89.29%	13774	90.03%
0.01	0.0875	0.875	20	1524	89.65%	13761	89.94%

Table 3. Experiment with LDA - separeted features

The sensitivity and specificity obtained by the methods are shown in the tables 5, 6, 7 and 8.

Results obtained by SVM are shown in Table 9.

t0	tinc	ta	n Feature	hit test	% sucess (test)	hit train	%sucess(train)
0.01	0.0150	0.150	50	1554	91.41%	14088	92.08%
0.01	0.0250	0.250	50	1557	91.59%	14092	92.10%
0.01	0.0350	0.350	50	1554	91.41%	14051	91.84%
0.01	0.0450	0.450	50	1551	91.24%	14042	91.78%
0.01	0.0500	0.500	50	1554	91.41%	14106	92.20%
0.01	0.0550	0.550	50	1546	90.94%	14064	91.92%
0.01	0.0650	0.650	50	1547	91.00%	14068	91.95%
0.01	0.0750	0.750	50	1553	91.35%	14040	91.76%
0.01	0.0850	0.850	50	1553	91.35%	14050	91.83%
0.01	0.0875	0.875	50	1540	90.59%	14094	92.12%

Table 4. Experiment with both LDA feature vectors

T	Sens(Bee)	Spec(Bee)	Spec(Bee)	Esp(Bee)
T1	0,777	0,978	0,784	0,98
T1	0,77	0,979	0,783	0,981
T2	0,771	0,979	0,783	0,981
T2	0,774	0,981	0,784	0,981
T3	0,778	0,978	0,783	0,981
T3	0,779	0,977	0,784	0,98
T4	0,771	0,98	0,783	0,981
T4	0,776	0,98	0,785	0,981
T5	0,773	0,977	0,785	0,981
T5	0,773	0,979	0,784	0,98
T6	0,741	0,787	0,751	0,799
T6	0,746	0,789	0,751	0,799

Table 5. Sensitivity and specificity to beetles in LDA with separeted features

T	Sens(Cat)	Spec(Cat)	Sens(Cat)	Spec(Cat)
T1	0,978	0,777	0,98	0,784
T1	0,979	0,77	0,981	0,784
T2	0,979	0,771	0,981	0,784
T2	0,981	0,774	0,981	0,785
T3	0,978	0,779	0,981	0,783
T3	0,977	0,779	0,98	0,784
T4	0,98	0,771	0,981	0,783
T4	0,98	0,776	0,981	0,785
T5	0,977	0,773	0,981	0,785
T5	0,979	0,773	0,98	0,784
T6	0,787	0,741	0,800	0,753
T6	0,789	0,746	0,799	0,751

Table 6. Sensitivity and specificity to caterpillars in LDA with separeted features

	Test		Train	
T	Sens(Bee)	Spec(Bee)	Sens(Bee)	Spec(Bee)
T1	0,871	0,945	0,879	0,949
T2	0,836	0,964	0,846	0,971
T3	0,863	0,944	0,874	0,951
T4	0,866	0,942	0,875	0,951
T5	0,869	0,942	0,88	0,947
T6	0,86	0,95	0,87	0,95

Table 7. Sensitivity and specificity to beetles in LDA with both features

4. Conclusion

Use of complex networks allows to model several applications that use classification by shapes contour, but in the current literature, there is no approach that classify the damage agents in soybean leaflets. Then, the approach described consists in adaptation of modeling complex networks, on the contours of the damage, in order to extract

	Test		Train	
T	Sens(Cat)	Spec(Cat)	Sens(Cat)	Spec(Cat)
T1	0,945	0,871	0,949	0,879
T2	0,964	0,836	0,971	0,846
T3	0,944	0,863	0,951	0,874
T4	0,942	0,866	0,951	0,875
T5	0,942	0,869	0,947	0,88
T6	0,917	0,9	0,923	0,903

Table 8. Sensitivity and specificity to caterpillars in LDA with both features

t0	tinc	ta	n Feature	hit test	% sucess (test)	hit train	%sucess(train)
0,01	0,015	0,15	30	1457	85.71%	15297	99.90%
0,01	0,015	0,15	20	1000	58.82%	9000	58.80%
0,01	0,025	0,25	30	1457	85.71%	15300	100.00%
0,01	0,025	0,25	20	1000	58.82%	9000	58.80%
0,01	0,035	0,35	30	1445	85.00%	15257	99.70%
0,01	0,035	0,35	20	1000	58.82%	9000	58.80%
0,01	0,045	0,45	30	1463	86.06%	15042	98.30%
0,01	0,045	0,45	20	1007	59.24%	9124	59.60%
Average (Joint Degree)				1455,5	85.62%	15224	99.48%
Average (Degree)				1001,75	58.93%	9031	59.03%

Table 9. Accuracy of SVM to separate features

enough features to discriminate the occurrence of agents.

The experiments executed illustrate the results with the variation of the parameters that define the dynamic evolution of networks through the threshold T_l.

Thus, results were obtained using this model. The extracted features can discriminate most of the patterns found. However, improvements will be necessary to the process of classification using new parameters of SVM, or combination of other classifiers.

References

[1] L. Antiqueira, M. d. G. V. Nunes, O. Oliveira, and L. d. F. Costa. Strong correlations between text quality and complex networks features, 2005. `http://arxiv.org/abs/physics/0504033`.

[2] A. R. Backes, D. Casanova, and O. M. Bruno. A complex network-based approach for boundary shape analysis. *Pattern Recognition*, 2009.

[3] A. Camargo and J. S. Smith. Image pattern classification for the identification of disease causing agents in plants. *Computers and Electronics in Agriculture*, 66:121–125, 2009.

[4] A. Camargo and J. S. Smith. An image-processing based algorithm to automatically identify plant disease visual symptoms. *Biosystems Engineering*, 102:9–21, 2009.

[5] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[6] T. Chalumeau, L. d. F. Costa, O. Laligant, and F. Meriaudeau. Texture discrimination using hierarchical complex networks. In *Proceedings of the Second International Conference on Signal-Image Technology and Internet-Based Systems*, pages 543–550, 2006.

[7] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 3a. edição edition, 2008.

[8] A. C. Nazaré-Jr, D. G. Menotti, J. M. R. Neves, and T. Sediyama. Detecção automática da Área foliar danificada da soja através de imagens digitais. In *WUW-SIBGRABI 2009*, pages 1–8, 2009.

[9] H. Pedrini and W. R. Schwartz. *Análise de Imagens Digitais*. Thomson, São Paulo, 1a. edição edition, 2008.

[10] M. C. Picanço, F. L. Fernandes, E. G. F. Morais, M. R. Campos, and V. M. Xavier. *Tecnologias de Produção e Usos da Soja*, chapter Manejo Integrado das Pragas, pages 119–132. Mecenas, Londrina, 2009.

[11] M. C. Picanço and R. N. C. Guedes. Manejo integrado de pragas no brasil: situação atual, problemas e perspectivas. *Ação Ambiental*, 2(4):23–26, 1999.

[12] T. L. G. Souza, D. G. Menotti, J. M. R. Neves, and T. Sediyama. Desenvolvimento de uma interface online de avaliação manual em rumo a um método automático de caracterização dos agentes causadores de lesões em folíolos de cultivares de soja. In *WUW-SIBGRABI 2010*, pages 1–5, 2010.

[13] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.