Automatic Detection of Ceratocystis Wilt in Eucalyptus Crops from Aerial Images

Jefferson R. Souza[1], Caio C. T. Mendes[2], Vitor Guizilini[3], Kelen C. T. Vivaldini[2], Adimara Colturato[4], Fabio Ramos[3] and Denis F. Wolf[2]

***Abstract*— One of the challenges in precision agriculture is the detection of diseased crops in agricultural environments. This paper presents a methodology to detect the *Ceratocystis* wilt disease in Eucalyptus crops. An unmanned aerial vehicle is used to obtain high-resolution RGB images of a predefined area. The methodology enables the extraction of visual features from image regions and uses several supervised machine learning (ML) techniques to classify regions into three classes: ground, healthy and diseased plants. Several learning techniques were compared using data obtained from a commercial Eucalyptus plantation. Experimental results show that the GP learning model is more reliable than the other learning methods for accurately identifying diseased trees.**

I. INTRODUCTION

In Brazil, Eucalyptus is the main forest species and its planted area comprises approximately 4.9 million hectares [1]. The harmful effect of diseased crop areas is a known problem and its damage to eucalypt plantations represents an estimated loss of $400 million per year, totaling $2.8 billion dollars in seven years [2]. Diseases can occur in all stages of the eucalyptus development and can attack and kill most of the crops in several locations and times of the year.

One of these diseases is *Ceratocystis fimbriata* [3], a fast spreading fungus that affects plants and is difficult to control. Its symptoms arise as a consequence of vessel blockage through mycelial growth, which prevents the water absorbed by the root system from appropriately supplementing the aerial part of the plant. Most of the current methods for detecting diseased areas are inefficient. Typically, humans must physically traverse large areas of the agricultural environment to inspect tree foliage, therefore, leaf inspection can be very time consuming and costly.

Plenty of research has been devoted to the development of applications for detection and classification of images captured by UAVs (Unmanned Aerial Vehicles).

Reid et al. [4] classified the vegetation of natural environments for geostatistical studies. Color and texture features were acquired from an image to generate the feature vectors used to classify vegetation by a multi-class generalization of the Gaussian Process (GP). In the identication of trees, the test data showed accuracies of up to 88% among four classes of trees. The GP framework was successfully applied for the learning of a non-parametric image-vegetation model.

Hung et al. [5] shows an algorithm to detect and classify tree crown species. An image is segmented using color and texture information, which is grouped into a feature vector of three colour channels and thirty texture channels. SVM is used to predict the labels of each feature vector. The classification stage uses supervised learning in the feature collection from the regions of interest. For each class (object, shadow and background), 100 training instances were collected and 95.6% accuracy was obtained in the optimized segmentation. In comparison with the tree crown hand labelled in each ground truth image, the accuracy was 80%.

Ghiasi and Amirfattahi [6] proposed a segmentation framework to provide a safe landing using an aerial image. They manually extracted color and texture features from images, labeled each area semantically and employed two K-NN classifiers. Local Binary Pattern histogram Fourier [7] and RGB color histograms were used to represent the texture and color features. The images were segmented into superpixels, and for each superpixel features were computed separately and then classified by KNN. The combined classifiers were applied to a large set of aerial images and resulted in an acurracy rate above 95%.

Moranduzzo and Melgani [8] developed a method to detect cars using high resolution images (2 cm/pixel) and applied a screening step of asphalted zones obtained from road maps of a Geographic Information System. Feature extraction was performed based on Scalar Invariant Feature Transform to identify a set of consistent keypoints. Keypoints were classified between the points that belonged to the same car and all the others using SVM. Data were used in the grouping of keypoints belonging to the same car. The efficiency of the method was confirmed by accuracy results of above 65%.

Diaz-Varela et al. [9] proposed a method for the automatic identification of terraces. The authors used computer vision techniques to generate orthoimagery and digital surface models (DSMs) with low-user intervention. A multi-scale object-oriented classification method was used to identify the terraces. The results showed a root mean square error lower than 0.5 when the height of the terraces was evaluated in relation to GPS data by DSM. The automated terrace classification achieved 90% of accuracy. The authors concluded that the

[1]Jefferson R. Souza is with the Department of Information Systems, Faculty of Computing, Federal University of Uberlândia (UFU), Monte Carmelo, MG, Brazil, jeffersonrodrigo.s@gmail.com.

[2]Caio C. T. Mendes, Kelen C. T. Vivaldini and Denis F. Wolf are with the Mobile Robotics Lab, Institute of Mathematics and Computer Science (ICMC), University of São Paulo (USP), SP, Brazil, {caiom; denis}@icmc.usp.br; kteixeira@sc.usp.br.

[3]Vitor Guizilini and Fabio Ramos are with the Australian Centre for Field Robotics, School of Information Technologies, University of Sydney, Australia, v.guizilini@acfr.usyd.edu.au; fabio.ramos@sydney.edu.au.

[4]Adimara Colturato is with the Laboratory of Critical Embedded Systems, ICMC, USP, Brazil, adimara@gmail.com.

use of UAVs to verify areas of several hectares is quick, efficient and lowers costs for remote sensing in vegetation.

We propose a detection approach to identify three different features, namely: (1) the ground; (2) healthy plants (eucalyptus trees); and (3) diseased plants, using UAV images. The proposed methodology consists of three steps: feature extraction, training and testing set generation and supervised learning. First, using a block-based approach, features are extracted from an aerial image dataset. These images are hand-labeled and partitioned into training and testing sets. Finally, four ML methods are trained using the training set and evaluated using the testing set.

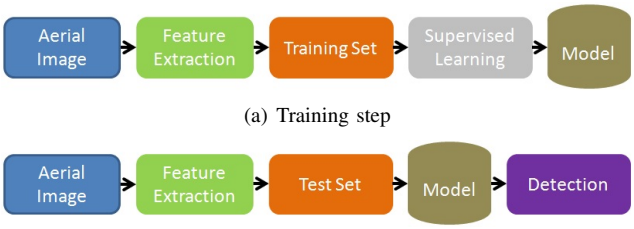
The main contributions of this research are:

1) A block-based approach, including the use and evaluation of a "contextual block" to extract visual features;
2) A *Ceratocystis* wilt detection model from aerial images, which was selected based on performance-wise evaluation of four distinct ML methods.

The remainder of this paper is organized as follows: Section II presents the proposed methodology, detailing the feature extraction and ML techniques used for the *Ceratocystis* wilt detection; the experimental setup, results and analysis are shown in Section III; finally, section IV draws the conclusions and suggests directions for future work.

II. Proposed Methodology

This paper proposes a method for distinguishing and identifying the ground, healthy plants and diseased plants. Fig. 1 shows the methodology divided into three sub-sections.



(a) Training step

(b) Detection step

Fig. 1. Diagram of the methodology. Data flow is represented by arrows.

A block-based approach is used in which features are extracted from square areas (e.g. 10×10) and fed into a model. The result is used for the classification of square areas (ground, healthy or diseased). The images were captured in 4cm/pixel using the UAV. Therefore, the area covered by a 10×10 block is $0.16m^2$. Instead of one block, we apply a pair of blocks of different sizes: a smaller non-overlapping classification block (*ClaB*), whose area is classified by the model, and a larger contextual block (*ConB*), whose features are expected to yield contextual cues. The same features are extracted from both blocks and concatenated as the input to a model. The step is defined by the size of the classification block (if its size is 10×10, the step is 10 for each dimension). Since the contextual blocks are larger and the classification block is at their center, they will overlap (Fig. 2). The *ClaB* is defined as 10 and the *ConB* is modified on the results section. To find better classification results for the three classes.

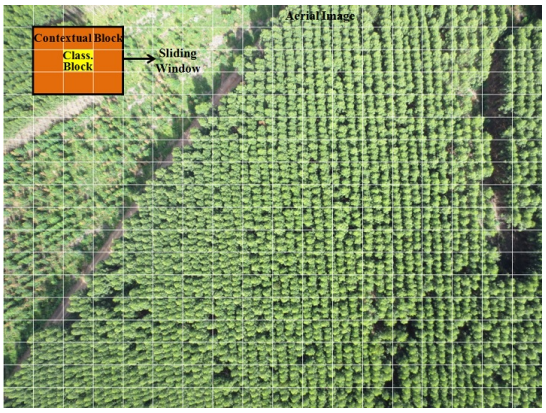


Fig. 2. Block-based approach: the features are extracted from square areas.

A. Feature Extraction

In terms of features, the RGB color space is converted into CIELab and gray-scale, therefore each pixel yields a four-element vector $\mathbf{p}_{u,v} = \{cl, ca, cb, g\}^T$, where (u, v) are the coordinate indexes. The mean and variance of each element are used. Formally, for a block B, the mean (M) and variance (V) are defined according to

$$M = \frac{1}{|B|} \sum_{(u,v)\in B} e_{u,v} \tag{1}$$

$$V = \frac{1}{|B|} \sum_{(u,v)\in B} (e_{u,v} - M)^2 \tag{2}$$

where $|B|$ is the number of pixels belonging to a block and $e_{u,v}$ is an element value of the $\mathbf{p}$ vector.

The standard Local Binary Patterns (LBP) [10] is employed to encode texture features in the gray-scale image $\mathbf{p}_{u,v} = \{g\}$. An LBP image is created according to

$$LBP^P_{u,v} = \sum_{n=0}^{P-1} s(\mathbf{n}_p - \mathbf{p}_{u,v})2^p \tag{3}$$

where P is the number of neighbors, $\mathbf{n}_0, \mathbf{n}_1, ..., \mathbf{n}_{P-1}$ is the neighbour set (their gray-scale value) of pixel $\mathbf{p}_{u,v}$ and s is given by Eq. 4.

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We use four neighbors, i.e. the four connected pixels, and one pixel distance. The histogram of each block in the LBP image is also used as features, and in the case of four neighbors it can be defined as

$$H_i = \sum_{(u,v)\in B} \delta(LBP_{u,v} = i)\,,\ i = 0, 1, \ldots, 15 \tag{5}$$

where H_i is the i-th bin of the histogram and

$$\delta(x) = \begin{cases} 1 & \text{if x is true} \\ 0 & \text{if x is false.} \end{cases} \tag{6}$$

LBP was employed as a general texture descriptor, and although the fine details are not present in the images, the

LBP is able to generate distinguishable features on different illuminations situations as the road and trees regions.

Only 4 instead of 8 (the usual number) neighbors were used, because they yield a 16-bin histogram instead of 256. The use of 4 neighbors avoids a high dimensional feature vector, which may imply a large Vapnik Chervonenkis (VC) [11] dimension, as the case of Neural Networks. A low VC-dimension is important when a small number of samples are used, as in diseased plants. Further, as we are dealing with high-resolution images, a low dimensional feature vector allows for faster training and evaluation times. We also included an additional feature that is the difference between the full gray-scale image mean and the block mean, this should create illumination invariance between images. Therefore, we have a total of 50 features, 25 features for each block.

B. Training and Testing Set

To create a sample database, we manually classify the images and only features of the pairs of blocks in which all pixels of the classification block belong to the same class (i.e. using a 10×10 classification block, all of the 100 pixels should be equally classified). The training and testing division is performed based on image indexes: a set of images is used for training and another for testing purposes. Such a scheme reinforces a significant difference between training and testing sets. All samples are feature-wise normalized according to

$$f_{norm} = \frac{f - m(\mathbf{f})}{std(\mathbf{f})} \tag{7}$$

where f_{norm} was the normalized value, f is the feature value, $\mathbf{f}$ is the vector containing all training values of a certain feature and m and std return the mean and standard deviation of a vector, respectively. The testing set is normalized by the same parameters m and std calculated in the training set.

C. Supervised Learning

The normalized vector is used to train ML methods. Classification methods were selected based on their well-known accurate generalization performances: K-Nearest Neighbors (K-NN) [12], Random Forest (RF) [13], Artificial Neural Networks (ANN) [14] and Gaussian Processes (GP) [15].

KNN is an instance-based classification method; it classifies a test sample according to the class of its K nearest training samples (Fig. 3), where K nearest is defined according to a distance metric (in our case, the Euclidean distance).

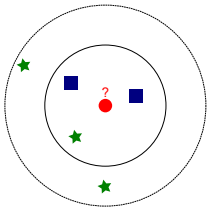


Fig. 3. K-NN with two classes, square and star. The red circle is a new query, which is classified as a square with $K = 3$ and a star with $K = 5$.

RF is a method that combines the prediction of multiple decision trees. Each tree is trained by a random subset of all available data and the output is the combination of the results of all trees. Fig. 4 shows the RF for multi-class classification.

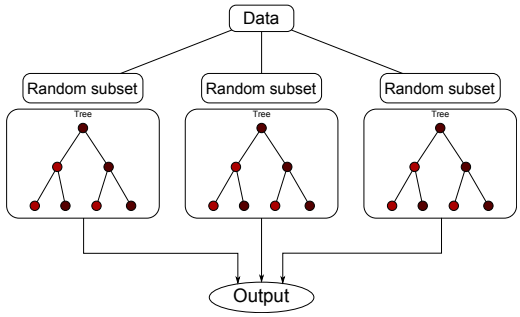


Fig. 4. RF method. A decision tree is created, for each random subset of features, and the result is the combination of the outputs of all trees.

ANN is a biologically inspired method, whose model can be viewed as a graph where each node is called "neuron". We use the Multilayer Perceptron (MLP) ANN, that has multiple layers of neurons. Its connections do not form a directed cycle and the information flows in only one direction (i.e. from input to output layers, Fig. 5). We use the backpropagation [16] algorithm to estimate the connection weights, a sigmoidal activation function and one hidden layer.

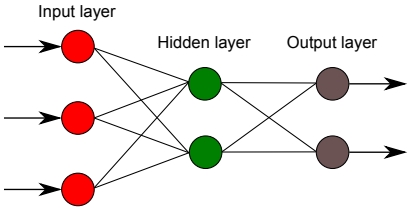
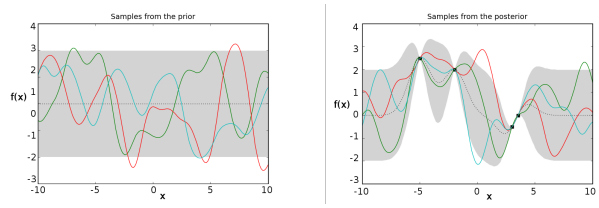


Fig. 5. MLP architecture with three layers: input, hidden and output layers.

GP is a Bayesian nonparametric tool that learns the input-output transformation function based on training data. A covariance function $k(x, x')$ is defined as distance (correlation) between points in the input space and the training information is extrapolated to estimate the output of new unobserved data (Fig. 6). A sigmoid function $sig_{logit}(x) = {}^{1}/_{(1+e^{-x})}$



(a) Untrained model (b) Trained model

Fig. 6. Priori and posteriori examples of a GP function. In (a) there is still no information about the underlying phenomenon, so the space function is large (as is the variance, given by the grey area). In (b), as data is incorporated into the model, this function space shrinks to accomodate only those that correspond to the observations (black dots).

is used to squeeze the regression values between $[0, 1]$, forming a binary classifier. Because this is a probabilistic framework, a measure of variance is provided and indicates the accuracy of each estimate. Due to the large number of training points, we chose to use a sparse GP, known as FITC (Fully-Independent Training Conditional) [17]. A number of support points is selected. The training data is projected into this small space during matrix inversion. This helps to decrease the computational cost of the GP, which is cubic to the number of training points.

III. Experimental Results

We implemented the described system and tested its performance using aerial images from an eBee robot, SenseFly[1] to evaluate the capabilities and performance of the methodology. The robot is a small fixed-wing UAV equipped with an RGB camera and an onboard GPS and powered by a lithium polymer battery. It has an embedded system for navigation control, flight maneuvers, sensor activaction, actuator controls and critical situation management (for example, the eBee alerts the base station when the battery is too low to perform a specific route).

Aerial images were captured by an IXUS 127 HS Canon camera of 4608×3456 pixel resolution flying at an average altitude of 890 meters. The frame rate was approximately one image per six seconds. For the image acquisition, the eBee software can be used to upload a map with a predefined route that must be covered by the UAV. The eBee flies over the route and takes pictures, which are uploaded to a computer after the flight and merged into an image containing the whole area observed (Fig. 7).

A. Training and Model Selection

The UAV followed a predefined route on a farm containing eucalyptus (Fig. 7) to collect data. The dataset consists of 154 images, from which we used only 15 due to the presence of diseased plants. In theory, each classification block produces a sample, however, a classification block may contain pixels with different classes. When that is the case, we ignore the block and do not produce a sample. The dataset was partitioned into 10 images for training (6442 unambiguous samples, where all pixels belong to the same class). For testing, 5 images are employed, where features are extracted from every possible block, each sample is classified according a given classifier and the results are used to generate a classified image. The evaluation is performed pixel-wise using the classified image and the ground truth. Sample training and testing databases were also generated for each combination of block size. Several ML algorithms were then applied. Experiments with the same random seeds were conducted for a comparison among the results regarding the F-Measure of the diseased plants.

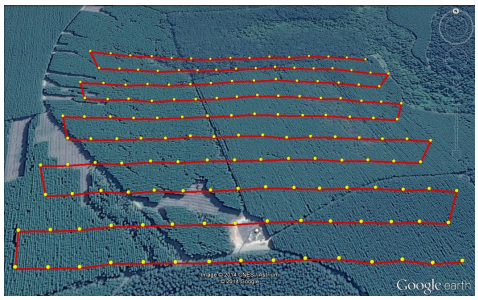


Fig. 7. The red line shows a predefined route performed by a UAV (eBee). The aerial images were captured for each GPS position (yellow circles).

[1] www.sensefly.com

The first experiment focused on finding a suitable model to represent the diseased, healthy and ground areas. The predictive performances of the models (Section II-C) were compared and results are shown in Tables I, II, III and IV.

Table I refers to the K-NN for each K neighbor according to the block size. Hyperparameters $K = 9$ and 90×10 block size provided the best F-Measure result over the testing set.

TABLE I
F-Measure of K-NN for each block size (BS).

	K neighbors				
BS	1	3	5	7	9
30×10	0.3019	0.4281	0.4667	0.4849	0.4957
50×10	0.3775	0.5246	0.5787	0.5999	0.6190
70×10	0.4139	0.5748	0.6429	0.6661	0.6780
90×10	0.4580	0.6143	0.6638	0.6862	**0.6865**

Table II shows the RF results for each number of trees according to the *BS*. Hyperparameters $NT = 25$ and 30×10 *BS* provided the best F-Measure result in the testing set.

TABLE II
F-Measure of Random Forest for each block size (BS).

	Number of Trees (NT)				
BS	5	10	25	50	100
30×10	0.3160	0.3835	**0.6875**	0.6523	0.6721
50×10	0.5447	0.5185	0.6811	0.5903	0.4451
70×10	0.5247	0.4336	0.4404	0.5696	0.4506
90×10	0.4040	0.6652	0.6588	0.5723	0.5544

Table III provides the ANN results for each number of *HN* according to the block size. Hyperparameters $HN = 50$ and 30×10 *BS* achieved the best F-Measure in the testing set.

TABLE III
F-Measure of ANN for each block size (BS).

	ANN - Hidden Neurons (HN)				
BS	50x200x3	50x100x3	50x50x3	50x25x3	50x12x3
30×10	0.6151	0.7582	**0.7940**	0.6772	0.7505
50×10	0.6662	0.6932	0.6238	0.6201	0.7180
70×10	0.7156	0.6543	0.6333	0.6018	0.6721
90×10	0.7124	0.6987	0.6964	0.6249	0.6514

Table IV shows the GP results for each kernel function according to the block size. In all experiments, 200 input coordinates were selected as support points according to the k-means algorithm (the dense GP implementation was unable to cope with this amount of training data). The Rational Quadratic covariance function with 70×10 *BS* as contextual block achieved the best F-Measure over the testing set.

Table V shows a comparison of the best models with their respective optimized hyperparameters. The GP showed

TABLE IV

F-MEASURE OF GP FOR EACH BLOCK SIZE (BS).

BS	GP - Kernel Function (KF)					
	Squared Exp.	Linear	Exp.	Matern 3/2	Matern 5/2	Rational Quadratic
30×10	0.7741	0.7034	0.7460	0.7802	0.7626	0.7543
50×10	0.7289	0.6726	0.7292	0.7662	0.7408	0.7412
70×10	0.7491	0.6679	0.7333	0.7488	0.5703	**0.8040**
90×10	0.7279	0.6763	0.7308	0.7513	0.7496	0.7056

higher F-Measure than the K-NN, RF and ANN methods. The hyperparameters of the best methods found are K-NN ($K = 9$), RF ($NT = 25$), ANN ($HN = 50$) and GP (Rational Quadratic kernel).

TABLE V

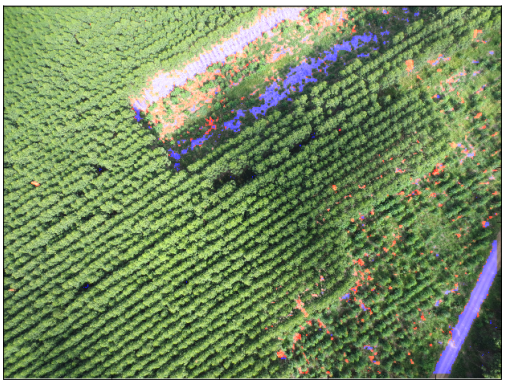
RESULTS OF THE BEST SUPERVISED ML MODELS.

Best ML Models	F-Measure
K-Nearest Neighbors	0.6865
Random Forest	0.6875
Artificial Neural Networks	0.7940
Gaussian Processes	**0.8040**

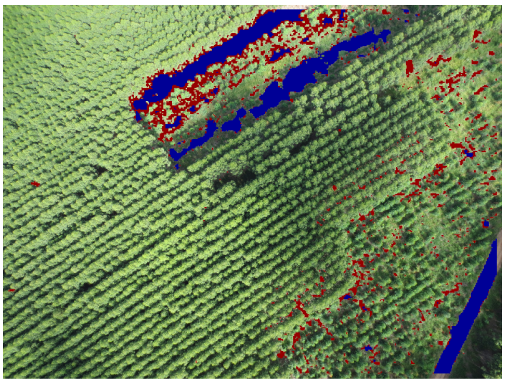
B. Validating the Two Best Models

In the second experiment, our goal was to validate the two best ML methods. We sought a model to detect samples of ground, healthy and diseased plants using the best ML methods (Table V), since they provided similar performance levels. Results of ANN and GP for the second experiment are depicted in Figs. 8(a) and 8(b)). The road was quite well segmented, therefore most of the diseased trees are presented in the image. The noise in the classification was due to the presence of ambiguous areas that did not belong to a single class and generated mixed feature vectors.

The predictive performances of the two best ML models were also numerically compared in the second experiment. The GP (Table VI) achieved higher F-measure scores than the ANN model. Ground, healthy and diseased plants were identified by GP with a larger amount of correctly classified pixels, especially in relation to road/tree segmentation.



(a) ANN



(b) GP

Fig. 8. Classification results of the two best models. Red areas show diseased plants and blue areas show roads.

C. Validating the Block-Based and Without Block Approach

The third experiment was to validate the block-based and without block approach for the best ML method. We sought an approach that achieves an improvement in the results of the detection of samples for classes: ground, healthy and diseased. We can see that the Table VII shows a higher F-measure for the diseased plants using the block-based approach. Three classes were identified with a larger amount of correctly classified pixels using the block-based approach.

TABLE VI

CONFUSION MATRIX OF THE BEST ML MODELS PERFORMED IN ONE TEST IMAGE, WHERE G - GROUND, H - HEALTHY AND D - DISEASED.

(a) ANN (Precision = 0.8827, Recall = 0.8825 and F-Measure = 0.8826) for the diseased plants.

	Ground truth		
Predicted	G	H	D
G	137251	2000	**88**
H	3298	7258572	**2916**
D	**2498**	**500**	**22573**

(b) GP (Precision = 0.9178, Recall = 0.9184 and F-Measure = 0.9181) for the diseased plants.

	Ground truth		
Predicted	G	H	D
G	140917	0	**247**
H	1316	7259783	**1841**
D	**814**	**1289**	**23489**

TABLE VII

CONFUSION MATRIX OF THE GP MODEL, WITH AND WITHOUT CONTEXTUAL BLOCK FOR ONE TEST IMAGE (P - PRECISION, R - RECALL AND F - F-MEASURE).

(a) GP without block (P = 0.7888, R = 0.9193 and F = 0.8490) for the diseased plants.

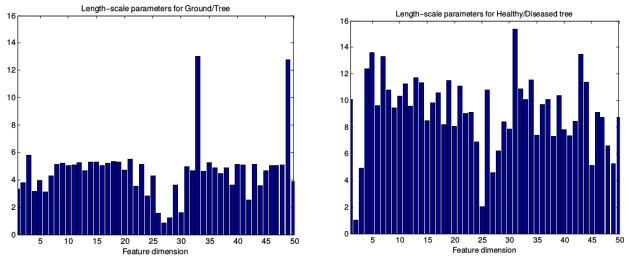
	Ground truth		
Predicted	G	H	D
G	135367	100	**223**
H	2572	7259783	**1841**
D	**5108**	**1189**	**23513**

(b) GP with block (P = 0.9178, R = 0.9184 and F = 0.9181) for the diseased plants.

	Ground truth		
Predicted	G	H	D
G	140917	0	**247**
H	1316	7259783	**1841**
D	**814**	**1289**	**23489**

D. Analysing the Best Model

Here, the focuses is on the analysis of the best model selected in the previous sections: the GP framework with 70×10 contextual blocks and a rational quadratic covariance function. Initially, we observe the length-scales obtained during the hyperparameter optimization stage for each feature dimension (Fig. 9). These length-scales essentially determine how sensitive each dimension is for classification purposes (smaller length-scales indicate less connected dimensions). This information could be employed to detect and remove dimensions that are not particularly useful in classification, thus reducing the dimensionality of the problem and, its computational complexity.

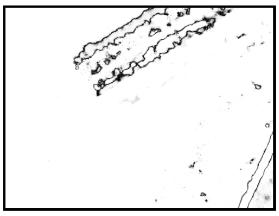


(a) Ground/Tree classification ($\alpha = 6.55$, $\sigma_v^2 = 4.41$)

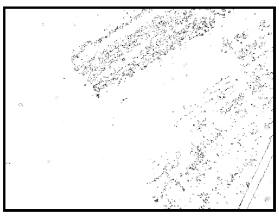
(b) Healthy/Diseased classification ($\alpha = 31.45$, $\sigma_v^2 = 3.02$)

Fig. 9. Trained length-scales for each feature dimension.

Variance values provided by the GP framework were also analysed, and are an estimate of the classification confidence. This information enables the detection of areas whose classification is particularly imprecise, therefore the results should not be used without careful consideration. Fig. 10(a) shows the Ground/Tree variance and, as expected, the transitional areas are more uncertain, since some pixels belong to both classes in the same contextual block. Fig. 10(b) shows the Healthy/Diseased variance, which accounts for most of the noise in the final classification results (Fig. 8(b)).



(a) Ground/Tree



(b) Healthy/Diseased

Fig. 10. Classification variance in the GP for both binary classifications. Darker areas indicate larger variances (lower certainty).

IV. Conclusion

A method for the detection and learning of visual features representing ground, healthy and diseased plants has been proposed, using RGB images from an UAV in outdoor environments. These features were processed by four distinct machine learning techniques (K-NN, RF, ANN and GP), and the classification output was compared for the selection of the best model within the proposed methodology and according to the F-measure metric. The aim is the detection of *Ceratocystis* wilt, a common Eucalyptus disease in Brazil, however the same technique can be readily applied to any visually observable condition.

We believe these results are an important step towards the advancement of precision agriculture as a reliable way of monitoring large crops autonomously. Further work will focus on improving the feature vector, both by removing irrelevant features and also introducing new descriptors (e.g. textons and HOG). The variance estimate provided by the GP framework will also be a valuable tool for the access to the local accuracy of classifications, which could be used to guide the UAV as to further survey areas of high uncertainty.

Acknowledgment

The authors would like to acknowledge the support granted by FAPEMIG, CNPq and Federal University of Uberlândia.

References

[1] ABRAF, "Associao BRAsileira de produtores de Florestas plantadas. Anurio Estatstico da ABRAF, year = 2012," Tech. Rep.
[2] D. R. Negro, T. A. F. S. Jnior, J. R. S. Passos, C. A. Sansgolo, M. T. A. Minhoni, and E. L. Furtado, "Bio. of Eucalyptus Urograndis Wood by Fungi," *Int. Bio. & Bio.*, vol. 89, pp. 95–102, 2014.
[3] I. P. Bedendo, *Doenças vasculares*, 3rd ed., ser. Manual de Fitopatologia: Princípios e Conceitos. São Paulo: Agronômica Ceres, 1995.
[4] A. Reid, F. Ramos, and S. Sukkarieh, "Multi-class Classification of Vegetation in Natural Environments Using an Unmanned Aerial System," in *IEEE ICRA*, 2011.
[5] C. Hung, M. Bryson, and S. Sukkarieh, "Vision-based Shadow-aided Tree Crown Detection and Classification Algorithm using Imagery from an Unmanned Airborne Vehicle," in *34th International Symposium for Remote Sensing of the Environment*, 2011.
[6] M. Ghiasi and R. Amirfattahi, "Fast Semantic Segmentation of Aerial Images based on Color and Texture," in *IEEE ICRA*, 2013.
[7] T. Ahonen, J. Matas, C. He, and M. Pietikainen, "Rotation Invariant Image Des. with LBP Hist. Fourier Features," in *Image Analysis*, 2009.
[8] T. Moranduzzo and F. Melgani, "Automatic Car Counting Method for Unmanned Aerial Vehicle Images," *Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1635–1647, 2014.
[9] R. A. Diaz-Varela, Z.-T. P. J., V. Angileria, and P. Loudjania, "Automatic identification of agricultural terraces through object-oriented analysis of very high resolution {DSMs} and multispectral imagery obtained from an unmanned aerial vehicle," *Journal of Environmental Management*, vol. 134, pp. 117–126, 2014.
[10] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with LBP," *TPAMI*, 2002.
[11] V. Vapnik and A. Chervonenkis, "On the uniform conv. of relative freq. of events to their prob." *Theory of Prob. & Its Applications*, 1971.
[12] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. The MIT Press, 2006.
[13] L. Breiman, "Random forests," *Machine Learning*, vol. 45, 2001.
[14] C. M. Bishop, *Neural Networks for Pattern Recognition*, USA, 1995.
[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Comp. and Machine Learning)*, 2005.
[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, pp. 533–536, 1986.
[17] A. Naish-Guzman and S. B. Holden, "The generalized fitc approximation." in *Neural Inf. Proc. Systems*. Curran Associates, Inc., 2007.