2017 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
April 26-28, Coimbra, Portugal

Low-cost Multi-spectral Vegetation Classification using an Unmanned Aerial Vehicle

João Natividade, José Prado and Lino Marques
Institute of Systems and Robotics
Department of Electrical Engineer and Computers
University of Coimbra, 3030–290, Coimbra, Portugal
Email: {jaugusto, lino}@isr.uc.pt

***Abstract*—In Precision Agriculture (PA), the decision support system is expected to be able to assist determining the needs of the farm. One of the crucial needs is to specify the amount of different fertilizers to be used, and also distinguished hydration levels of crops. The amount of necessary fertilizers to apply is directly related with the area occupied by a certain type of plant, and with the hydration levels of such area. In order to achieve such a goal, Unmanned Aerial Vehicle (UAV) can be used to scan over an area, while detecting and classifying the type of vegetation in herbaceous crops. The majority of current monitoring technologies are very expensive, or the low cost systems use cameras that will gather information only in the visible spectrum. Therefore, we propose a low-cost multi-spectral system, where an Unmanned Aerial Vehicle (UAV) was equipped with a set of exchangeable filters over a camera, connected to a Raspberry Pi (RPi). Two classifiers were implemented and optimized in order to maximize the true positive rate (TPR) while minimize the false positive rate (FPR). The entire system is automated and the classification output is provided from the RPi to a ground station in real-time, by a Wi-Fi socket connection. The classifiers have shown to be able to distinguish, based on our sensor data, two types of vineyard and tree species of plants. For comparison purposes, we present results showing the performance of both classifiers while using data gathered by our system. The Region Of Interest (ROI) was identified by a thresholding algorithm based on Normalized Difference Vegetation Index (NDVI) measurements.**

I. Introduction

Nowadays, most PA research is oriented towards the implementation of new sensors and instruments, able to detect crop patterns in real or *quasi* real-time. Discrimination between crop species can be extremely important when one wants to differentiate crop treatments, such as the use of fertilizers, herbicides or even the way crops should be hydrated. Monitoring crops with an UAV can strongly help farmers to improve their ability to manage vegetation treatments, once the identification of different features can be computed on a daily basis and with results provided in real-time. As a first step, the UAV should understand which is the species that it is looking at. The main goal of this work is focused on this subject, with the creation of a Pattern Recognition System (PRS) with these data. From the moment when vegetation is classified among different species, the final output can be used not only for territorial mapping, but also to compute different VIs which can then be used to compare crop vigour between vegetation that belongs to the same species. Since different species reflect distinct radiation on the V (Visible) and NIR wavelengths of the electromagnetic spectrum, it was developed a low-cost "product" (see figure [1]), composed by one RPi NIR camera connected to a RPi and mounted on an UAV.

978-1-5090-6234-8/17/$31.00 ©2017 IEEE

Fig. 1: Multispectral configuration (view from different sides).

The process of pattern recognition is fully-automated and controlled by the ground station which defines the exact moment when samples should be captured and afterwards processed. The user can choose between PRS execution during flight or after UAV's landing. Before classification comes into play, the PRS takes care of the identification of crops (set of pixels which neither belong to the bare-soil nor to non-organic materials). It was assumed that a particular sample (set of {V, NIR} images, see figures [2b] and [2c], respectively) entirely belongs to the same crop species. Therefore, the UAV flew over several farmland of a single crop species (vineyards and forests of either eucalyptus, pine trees, orange trees, olive trees or magnolias).

(a) UAV during flight. (b) V image. (c) NIR image.

Fig. 2: Set of images related to UAV's flight.

As a new contribution to crop monitoring, a particular segmentation algorithm for low Depth Of Field (DOF) aerial images was tested offline. Furthermore, a Wi-Fi configuration

was tested between the RPi located in an UAV and a ground station fixed on the ground, which allowed real-time classification for the crop vegetation being sampled, based on two classifiers.

II. Related Work

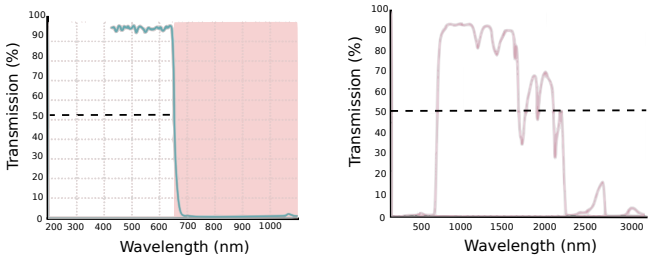
Distinct studies demonstrate that it is possible to make crop classifications from UAV images [5], [6]. In fact, each of the aforementioned studies uses either a "professional" multi-spectral camera (*e.g.*, Tetracam [16]), or a pair of digital cameras for the capture of multi-spectral data, which significantly increases the overall cost of the sensory rig[1]. Furthermore, [8] argues that it is possible to classify the average yield (*e.g.*, from sugarcane) according to a Decision Tree (DT) and using values provided by the NDVI. Moreover, either K-means or SVMs classification was tested for vineyard detection for non-aerial images [14], [4], [13]. Later in [13], a custom-made rig sensory that integrates a CCD camera and a servo-controlled filter wheel was proposed, for the identification of grapevine elements, which is composed by a high-cost system not suitable to bit fit into an UAV due to its size and its imperative laboratory terms of use. As mentioned in [11], there is currently a need for standardization of hyper-spectral image processing techniques, and the development of affordable platforms will endow more people to have access to it, thus being an important step in this direction. Thus, in order to move forward and overcome some of the issues found in these recent studies, we propose a low-cost hardware system, where our UAV was equipped with a set of sensors connected to a Raspberry Pi, as described in detail in the next section.

III. Hardware Description

The UAV model (Sky Hero Little Spyder SK00-104-RTF) is manually controlled by a Radio Control (RC) (FrSky Taranis X9D Plus 16CH RC Transmitter). The flight controller (3DR Pixhawk, aided by a 3DR GPS module) has got an external compass, and uses a barometer which measures air pressure as the primary means for determining altitude. The quad-rotor is equipped with 4 motors (Sky Hero 2806, 950 kV) and it is power supplied by a 4S (four cells) (3700 mAh, 14.8 V, Zippy Compact 25C Series) Li-Po battery. Depending on the farmland to be classified (vineyards or crop forests), the UAV has flown up to an height of either 3 or 20 meters, respectively, above the ground level and during a 5 minutes period. The UAV's takeoff occurred in *Stabilize* flight mode and so it remains until the moment when the target altitude is reached. Here, the flight mode is changed to *Altitude Hold*.

The multispectral design was based on the concept of capturing different radiation (differentiated wavelengths) with the use of distinct optical filters (see figure [3]) laid in front of the lens of a 8-MegaPixel RPi NIR camera board (Raspberry Pi NoIR Camera Module v2). Samples were captured by the use of the combination of V with NIR optical filters. The Wireless Local Area Network (WLAN) is established by the use of both an access point (Tube 2H, Alfa Network Inc., 150 Mbps, 2.4 GHz) and a USB Wi-Fi (150 Mbps) adapter connected to the RPi. Finally, a servo (TowerPro SG90 Mini Gear Micro Servo 9g) allows the filters' wheel to rotate, according to Pulse-Width Modulation (PWM) signals sent from the RPi.

[1]When compared to the multi-spectral approach exposed during this article.



(a) V optical filter. (b) NIR optical filter.

Fig. 3: Spectral responses for different optical filters.

IV. Pattern Recognition System

In describing a crop classification system, distinctions were made among pre processing (identification of the ROI), feature extraction, classification and post processing (assessing of the final class according to the output provided by the two classifiers). Figure [4] shows a slightly more elaborate diagram of the components considered in a PRS.

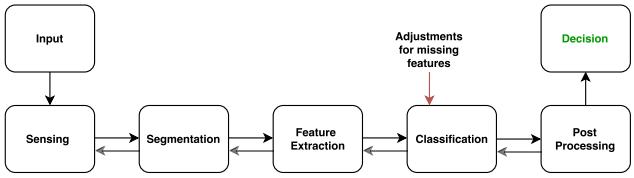


Fig. 4: The PRS.

A. Sensing and Data Collection

A non-snapshot technique based on filter shifting was implemented, providing two images per sample. The implemented architecture was a client-server approach, which consists in a Raspberry Pi acting as client, receiving commands to take photos and broadcast the images; while the ground station server decides when to take the photos and stores samples in disk. The ground station establishes a connection to the RPi (via SSH) which is previously configured for booting with a specific IP address. A client-server interprocess communication is used between both the RPi and the ground station, respectively. The client establishes a TCP socket connection to the server, sending an image stream from the RPi to the ground station (socket A). Streaming is done according to a resolution reducing factor of $\frac{1}{16}$ (decreasing the amount of data to be transmitted which allows real-time image transmission). At the same time and from another socket, a message is continuously sent from the RPi towards the ground station, saying whether the streaming is in high or low resolution (socket B). Through this way, the ground station can choose

in real-time between "streaming" or capturing the desired image. Every time the ground station decides to capture a sample, the filter which is attached in front of the camera's lens changes its position, allowing the capture of the same sample in different bands. For each sample, it is considered that all images are overlapped with each other. After data collection, two segmentation approaches were considered.

B. Segmentation based on NDVI thresholding

It is a multi-spectral remote sensing data technique, which finds the NDVI (described on [IV-D]) for each sample, and "thresholds" the final data according to a specific value. This technique can be extremely accurate for farmland which hasn't vegetation beyond the one that belongs to the ROI.

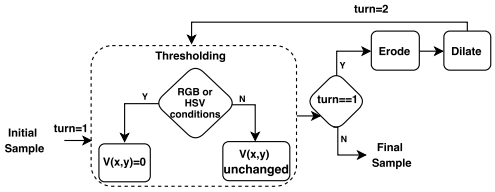


Fig. 5: Block diagram for the segmentation based on NDVI thresholding.

C. Segmentation based on low DOF V image

When moving the sensor in a digital camera in relation to its lens, the plane in object that is sharply focused moves as well. Through this way of thinking, the sensor can be moved by a specific range, until the feature goes out of focus [*i.e.*, when the width of the blurred image of the feature has become larger on the sensor than the maximum allowed Circle of Confusion (COC)]. Therefore, when striking the sensor, objects which come to a focus too far in front of or behind the COC (C1) will spread out to a size larger than the circle. Examining the lines that connect the edges of the lens aperture to each side of the COC (see figure [6]), one can see that an object gets out of focus, when positioned outside D1. Repeating on the object space an analogous procedure to the one made on the image space of the lens, another COC is achieved (C2) and D2 is created. Scene features inside D2 will focus to positions inside D1, leading to a blur which will not be greater than one COC (*i.e.*, features will appear "in focus" on the sensor). The width of the object space D2 is called DOF [9]. Sharply focused objects have more details within the object

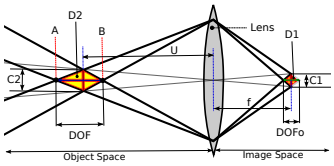


Fig. 6: DOF illustration.

than the ones which are not "in focus", leading to higher values of wavelet coefficients in the high frequency bands of the transform [7]. The high frequency energy is then measured by the standard deviation and variance of wavelet coefficients for those bands. Discrete Wavelet Transform (DWT) represents an image as a sum of wavelet coefficients, known as *wavelets*, with different location and scale. It represents the data into a set of high pass and low pass coefficients [12]. The algorithm for this segmentation was developed with the idea of being part of a target recognition process, which has as input images with low DOF. Therefore, desired vegetation would be "in focus" whereas background and/or foreground would be "out of focus". This segmentation has its pseudo-code as an adapted version of [7]. The image is partitioned into blocks, which are "classified" as background or OOI, according to their average, μ, intensity (feature 1) and their standard deviation, σ, and variance, σ^2, of wavelet coefficients of high frequencies (features 2 and 3, respectively). The

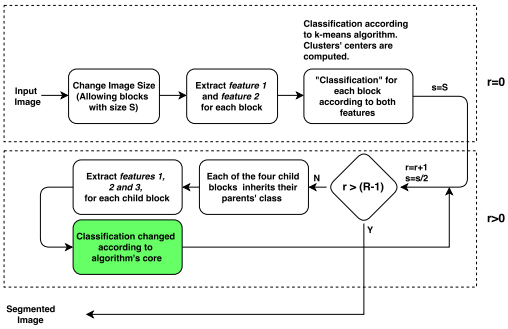


Fig. 7: Block diagram for the segmentation based on low DOF V image.

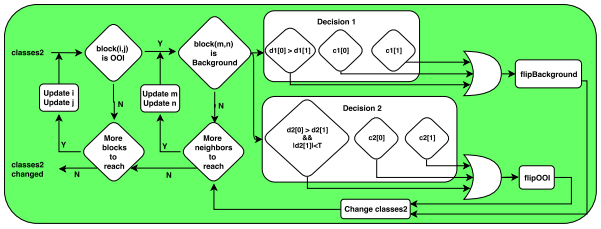


Fig. 8: Block diagram for the core of the segmentation based on low DOF V image.

TABLE I: Conditions used to assess the change of class labels.

Condition	**Description (when condition verified)**
$d1[0] > d1[1]$	Block (m, n) has its μ closer to OOI than to Background
$c1[0]$	Block (i, j) has its σ^2 closer to OOI cluster than to Background cluster
$c1[1]$	Block (i, j) has its σ closer to OOI cluster than to Background cluster
$d2[0] > d2[1]$	Block (i, j) has its μ closer to Background than to OOI
$d2[1] < \mathrm{T}$	Bounding condition of changing block (i, j) to Background
$c2[0]$	Block (i, j) has its σ^2 closer to Background cluster than to OOI cluster
$c2[1]$	Block (i, j) has its σ closer to Background cluster than to OOI cluster

Multi Resolution Segmentation Algorithm MRSA comprises 2 sequential phases: $r = 0$ (Phase 1) and $r > 0$ (Phase 2) (see figure [7]). During Phase 1, the image is partitioned into blocks of size S, and their grayscale mean as well as their wavelet coefficients are computed. Therefore, each block is classified as being an OOI or background, through K-means algorithm, according to features 2 and 3. At the end of this stage, both clusters' centers for each "classification" are computed and afterwards used in Phase 2. Throughout Phase 2, objects are subdivided by a factor of 2, in each iteration. Typically, $S = 32$ for $r = 0$, leading to 4 iterations in Phase 2 ($r = 5$ is despised as at this iteration blocks shrink to the size of a pixel and so are likely to be smooth). Defining R as the number of iterations required to have objects with a size of a pixel, for each iteration $r \in [0, R-1]$, features are computed for every object. Depending on conditions specified on the algorithm's core (see figure [8] and table [I]), each object class is maintained or changed.

D. Feature Extraction

Spectral VIs reduce the multiple-waveband data at each pixel to a single numerical value. Equations for each VI were considered as follows (for more detail, see [1], [15] and [2]):

$$\text{NDVI} = \frac{\text{IR} - \text{V}}{\text{IR} + \text{V}}, \qquad \text{GNDVI} = \frac{\text{IR} - \text{G}}{\text{IR} + \text{G}},$$
$$\text{GRVI} = \frac{\text{IR}}{\text{G}}, \qquad \text{SAVI} = \frac{\text{IR} - \text{R}}{\text{IR} + \text{R} + L} \times (1 + L),$$
$$\text{SR} = \frac{\text{IR}}{\text{R}}, \qquad \text{GVI} = \frac{\text{G} - \text{R}}{\text{G} + \text{R}},$$
$$\text{RNDVI} = \frac{\text{IR} - \text{R}}{\text{IR} + \text{R}}, \qquad \text{ExG} = 2 \times g - r - b.$$

For each VI, its mean is computed and afterwards stored as future input data for the classification method.

E. Classifiers

1) Support Vector Machines: According to the data that were captured, *C-classification* guarantees better performances when compared to *nu-classification* (see figures [11] and [12], for Receiver Operating Characteristic (ROC) curves according to each tested dataset). For this reason, only the mathematical model for C-classification was described. Each instance in the training set contains one "target value" (*i.e.*, the class labels) and several "attributes" (*i.e.*, the features). The training set is composed by l pairs of *(feature, label)*, $(x_1, y_1), ..., (x_l, y_l)$, where $x_i \in R^n, i = \{1, ..., l\}$ and $y \in \{1, ..., k\}$ ($k = \{5, 3\}$, depending on the related dataset) is the class of x_i. For training data from the i^{th} and the j^{th} classes, SVMs require the solution of the following binary (optimization) problem:

$$\min_{w^{ij}, b^{ij}, \xi^{ij}} \quad \frac{1}{2}(w^{ij})^T w^{ij} + C \sum_{i=1}^{l} \xi_t^{ij}$$
$$\text{subject to} \quad (w^{ij})^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \text{ if } y_t = i,$$
$$(w^{ij})^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \text{ if } y_t = j,$$
$$\xi_t^{ij} \geq 0.$$

The training vectors (features x_i) are mapped into a higher dimensional space by the function ϕ. Minimizing $\frac{1}{2} w^T w$ is equivalent to maximize $\frac{2}{||w||}$ (maximize the margin between two groups of data). When data are not linear separable, there is a penalty term $C \sum_{i=1}^{l} \xi_i$ which can reduce the number of training errors. The kernels, K ($K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$), were tested according to the list mentioned in table [II].

TABLE II: Different kernels used on the PRS.

Type of Kernel	**Formula**[2]
Linear	$K(x_i, x_j) = x_i^T x_j$
Polynomial	$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
Radial Basis Function (RBF)	$K(x_i, x_j) = e^{-\gamma \|\|x_i - x_j\|\|^2}, \gamma > 0$

2) Decision Trees: Considering a training set composed by l pairs of *(feature, label)* ($x_i \in R^n, i = \{1, ..., l\}$ and $y \in \{1, ..., k\}$), a decision tree recursively partitions the space such that the samples with the same labels are grouped together. Let the data at node m be represented by Q. For each split candidate, $\theta = (j, t_m)$, consisting of a feature j and a threshold t_m, the data are partitioned into $Q_{left}(\theta)$ and $Q_{right}(\theta)$ subsets:

$$Q_{left}(\theta) = (x, y) \mid (x_j <= t_m), \; Q_{right}(\theta) = Q \setminus Q_{left}(\theta).$$

The *impurity* G at m is computed[3] using an impurity function H:

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta)),$$

and the parameters (j^*, t_m^*) are then selected, according to the ones that minimizes the impurity:

$$\theta^* = \text{argmin}_\theta G(Q, \theta).$$

Until the maximum allowable depth is reached (or N_m reaches a state where its value is lower than the minimum allowable samples at a leaf node) $Q_{left}(\theta^*)$ and $Q_{right}(\theta^*)$ are recursively computed. Taking on values $\{0, ..., k-1\}$ for the different classes at node m and representing a region R_m with N_m observations, let

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

be the proportion of class k in node m. Finally, the two *criterion* chosen for measuring the impurity were the *gini* and *entropy* defined, respectively, as follows:

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk}), \; H(X_m) = -\sum_k p_{mk} log(p_{mk}).$$

F. Finding the Best Classifier

Learning the parameters of a prediction function and testing it on the same data is a methodological mistake. A model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data (this is called *overfitting* [10]). To avoid it, it is common practice when performing a

[2] γ, r and d are kernel parameters (hyperparameters).

[3] n_{left}, n_{right} and N_m are, respectively, the number of nodes in the left and right subsets, and the sum of the number of left and right nodes, according to the data at a particular node m.

supervised machine learning experiment, to hold out part of the available dataset as a test set. Different estimators (such as SVMs and DTs) require an evaluation of several settings (so-called *hyperparameters* [10]). For the *k-fold* Cross Validation (CV) approach, the training set is split into *k* smaller sets. For each of the *k folds*, the following procedure is ensued: 1) A model is trained using $k-1$ of the folds as training data; 2) The resulting model is validated on the remaining fold. In conclusion, the performance measure reported by *k-fold* cross-validation is then the average of the values computed in the loop. After achieving the "best" hyperparameters for the model, test data comes then into play. SVMs CV took into account the following specifications: 1) *Type*: C-SVM or nu-SVM; 2) *Kernel Type*: Linear, Polynomial or RBF; 3) *Degree* (for polynomial kernel); 4) $\mathbf{\Gamma}$ (for polynomial or RBF kernels); 5) *Cost* (for C-SVM type) and nu (for nu-SVM type). The following specifications were taken into account for DTs CV: 1) *Criterion*: gini or entropy; 2) *Maximum Depth*: 5, 7, 9, 11 or None; 3) *Minimum Samples to Split*:1, 2, 3, 4 or 5; 4) *Minimum Samples per Leaf*: 1 or 2. The PRS for the two datasets was evaluated according to the following approaches: 1) Scoring parameter (model-evaluation tool using CV [assessing of *accuracy* for CV], where *k* results from the folds are averaged to produce a single estimation for *accuracy*); 2) Metric functions (set of functions [*precision*, *recall*, *accuracy* (*Acc*) and ROC curve - relation between True Positive Rate (*TPR*) and False Positive Rate (*FPR*)] assessing prediction errors):

$$TPR = Recall = \frac{Tp}{Tp+Fn}, \quad FPR = \frac{Fp}{Fp+Tn},$$
$$Precision = \frac{Tp}{Tp+Fp}, \quad Acc = \frac{Tp+Tn}{Tp+Tn+Fp+Fn}.$$

In conclusion, the final "optimized" models for the PRS according to each dataset were achieved having in account the following enumerated items (for item 1. it was assigned the highest priority when choosing classifier's configuration, whereas item 3. has the lowest priority): 1) Choice of the minimum value for the Euclidean distance between the correspondent point on the ROC curve and the *sweet spot* ($TPR = 1, FPR = 0$); 2) Choice of the configuration which results on the best CV accuracy; 3) Choice of the configuration which results on the best precision. Finally, the accuracy was used to compute the probability associated to the output's classification (see section [IV-G]) during test predictions.

G. Post Processing

When performing classification, achieving the final decision should be done according to the correlation between prediction of both classifiers. Here, this decision (final prediction) is computed as an *accuracy* function related to the two classifiers. Each classifier works on a testing dataset, with a total of *N* (*N* differs between datasets #1 and #2) different unseen and randomly selected samples. According to the achieved ratio between correct and incorrect decisions, accuracies for both classifiers are computed: $accSVM$ and $accDT$ (considering SVMs and DTs, respectively). Therefore, the final prediction [probability (*prob*) for crop class] is computed as the probability of either both or just one classifier hit correctly the unseen test (see the equation mentioned below).

$$prob = accSVM \times accDT + accSVM \times (1 - accDT) + accDT \times (1 - accSVM) \quad (1)$$

This procedure is only considered for the case where the two classifiers agree on the same class. Otherwise, the final result is disregarded and a new test should be done.

V. Experimental Results

A. Software (SW) Implementation

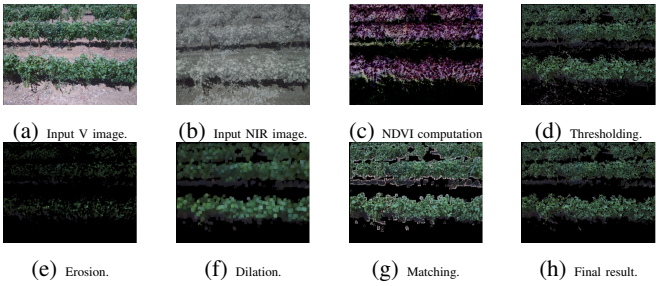
The pattern classification algorithm was implemented and tested in *C++* programming language and the following libraries were used to manipulate data: `OpenCV`, SVMs library [3] (`libsvm`) and DTs library [10] (`scikit-learn`).

B. Segmentation and Classification

The PRS was tested according to the following datasets:

- Pattern Recognition System for **Trees Classification**: distinguishing from *Pine Trees*, *Orange Trees*, *Olive Trees*, *Eucalyptus* and *Magnolias* $\rightarrow$ PRS for dataset $\#1$[4];
- Pattern Recognition System for **Vineyards Classification**: believing that grapevines from different "qualities" generate different VIs, this PRS has the goal of distinguishing from 3 different vineyards $\rightarrow$ PRS for dataset $\#2$[5].

A set of 50 samples was used for the test dataset #1, concerning the classification of 5 different trees (5 subsets of 10 images for each species). On the other hand, 175 samples were taken for the classification test when considering dataset #2 (71 samples for Vineyard $\#1$, 57 for Vineyard $\#2$ and 47 for Vineyard $\#3$). For the considered metric functions on the PRS, the entire data were split which led to N ($N = \{5, 3\}$, depending on the dataset) binary classifications. For the two datasets, 82 configurations were tested for SVMs and 80 for DTs (by the change of specific hyperparameters) according to the PRS. Considering both SVMs CV and DTs CV, data

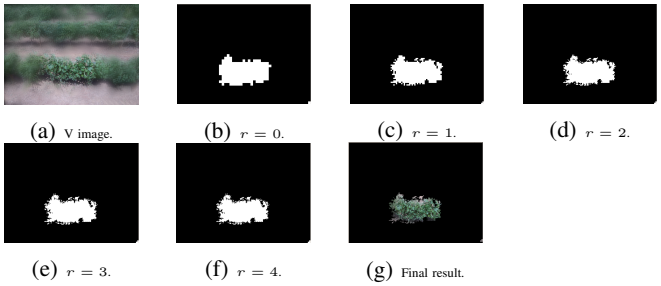


(a) Input V image. (b) Input NIR image. (c) NDVI computation (d) Thresholding.
(e) Erosion. (f) Dilation. (g) Matching. (h) Final result.

Fig. 9: Example of the segmentation based on NDVI thresholding for the "entire" vineyard.

[4]489 samples: 121 Pine Trees, 114 Orange Trees, 93 Olive Trees, 91 Eucalyptus and 70 Magnolias.

[5]635 samples: 215 for Vineyard $\#1$, 256 for Vineyard $\#2$ and 164 for Vineyard $\#3$.



(a) V image. (b) $r = 0$. (c) $r = 1$. (d) $r = 2$.

(e) $r = 3$. (f) $r = 4$. (g) Final result.

Fig. 10: Example of the segmentation based on low DOF V image for a "particular region" of the vineyard.

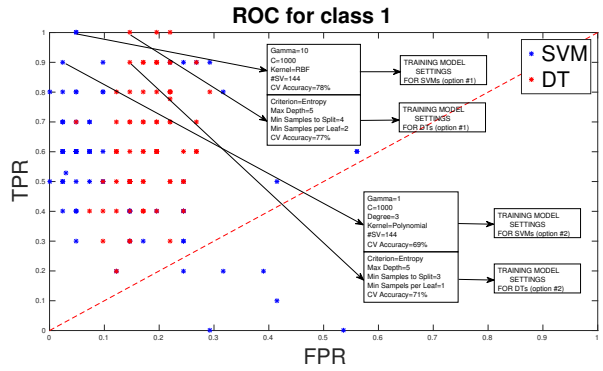


Fig. 11: TPR FPR relationship for class 1 (related to test set from dataset #1).

were cross validated with 10 folds. From figure [11], one can see that SVM's kernel can be defined by a RBF ($\Gamma = 10$) of type C-SVM ($C = 1000$), offering a precision of 83% and a CV accuracy of 78% (where 144 support vectors were identified) (option #1). Indeed, option #1 provide the best ROC point for the binary classifications of classes 1, 2 and 5. Another possibility might be defining the kernel by a polynomial function of 3^{rd} degree ($\Gamma = 1$) of type C-SVM ($C = 1000$) (option #2, which coincidentally computes the same number of support vectors that option #1 does), since this configuration offers the best position on the ROC curve, for the binary classifications of classes 3 and 4. Options #1 and #2 led to testing accuracies of 72% (36/50) and 68% (34/50), respectively. With regard to DTs, the training model can be configured with two different sets of hyperparameters. In fact, as mentioned for SVMs during PRS training stage for dataset #1, a specific configuration for DTs better suits data for classes 1, 2 and 5 (option #1), whereas another configuration achieves better performances for classes 3 and 4 (option #2). These two options are defined, respectively, as follows: *Criterion=entropy*, *Maximum Depth=5*, *Minimum Samples Split=4* and *Minimum Samples per Leaf=2* (option #1, also defined by a precision of 87% and CV accuracy of 77%) and *Criterion=entropy*, *Maximum Depth=5*, *Minimum Samples Split=3* and *Minimum Samples per Leaf=1* (option #2, also defined by a precision of 80% and CV accuracy of 71%). These models led to testing accuracies of 74% (37/50) and 70% (35/50), respectively.

Considering PRS for dataset #2 and from figure [12], one can see that SVM's kernel can be defined by a RBF ($\Gamma = 1000$) of type C-SVM ($C = 1000$), offering a precision of 97% and a CV accuracy of 83% (where 199 support vectors were identified). This model led to a testing accuracy of (approximately) 73% (127/175). With regard to DTs, the training model that better describes the dataset should be the following set of hyperparameters: *Criterion=entropy*, *Maximum Depth=None*, *Minimum Samples Split=2* and *Minimum Samples per Leaf=1*, defined by a precision of 87% and CV accuracy of 77%. This model led to a testing accuracy of (approximately) 79% (138/175). In fact, for each binary classification among classes 1, 2 and 3, the same configuration has achieved the better performance for the PRS. The

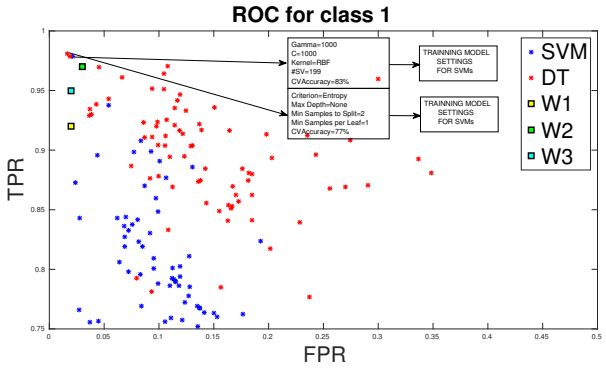


Fig. 12: Zoomed TPR FPR relationship for class 1 (related to test set from dataset #2).

feature space for the set of considered VIs is similar between Eucalyptus and Olive trees. The radiation reflected by these two tree species cannot be well differentiated for the achieved electromagnetic radiation. For this reason, either SVMs or DTs achieved better performances (for binary classifications) considering particular configurations for the two tree species mentioned above. Nevertheless, both configurations for each classifier led to satisfactory predictions for the entire set of data (see both CV and testing stage accuracy results for the PRS according to the dataset #1 as mentioned above), when multi-class was considered.

On the other hand, the PRS for the dataset #2 was trained for one particular configuration for each classifier. The same set of hyperparameters was located near to the sweet spot for all the three binary classifications. For this reason, it was chosen a specific configuration for training each learner, knowing that the closer a point follows the left-hand border and then the top border of the ROC (sweet spot), the more accurate the test is. It is also important to clarify that each precision and recall results were computed as being an average among binary classes. This means that their values are greater than any other metric. Indeed, these metrics are increased by the correct distinction among samples from belonging or not to a specific class (binary classification), whereas CV accuracy's rate is only increased if the algorithm hits the correct class, from the entire set of possibilities (multi-class classification).

Hence, precision was only considered as a measure to untie from different configurations, after concluding the assessment of both the ROC curves and CV accuracies for a particular configuration (see section [IV-F]). In conclusion, the correct choice of kernel parameters was crucial for obtaining good results, which practically means that an extensive search should be conducted on the parameter space before results can be trusted.

The performance for the classification of the PRS according to dataset # 2 was compared[6] to the relationship between TPR and FPR of three different studies. One can see such binary classifications from figure [12]: 1) [14] classifies pixels which belong to grapes, according to different color spaces and bin sizes during the "segmentation" process (performance defined as W2 in figure [12]); 2) [4] differentiates vines prior to ripening from grapevines during ripening (performance defined as W1 in figure [12]); 3) [13] sustains that a classification of image pixels into five clusters (leaves, stems, branches, fruit and background) can be accurately measured, and achieves its best results for the identification of stems (performance defined as W3 in figure [12]).

TABLE III: Final results obtained for the PRS related to datasets $\#1$ and $\#2$, according to both SVMs and DTs.

SVMs configuration	**CV Accuracy**	**Precision**	**Recall**	**Accuracy**
Dataset $\#1$ (option $\#1$)	78%	93%	86%	72%
Dataset $\#1$ (option $\#2$)	69%	82%	78%	68%
Dataset $\#2$	83%	97%	94%	73%
DTs configuration	**CV Accuracy**	**Precision**	**Recall**	**Accuracy**
Dataset $\#1$ (option $\#1$)	77%	87%	88%	74%
Dataset $\#1$ (option $\#2$)	71%	80%	82%	70%
Dataset $\#2$	77%	87%	90%	79%

VI. Conclusion

Results for the PRS considering both datasets showed that one can classify farmland with a low-cost camera and an UAV. The algorithm has achieved better performances on the distinction between vineyards species (PRS for dataset $\#2$). Indeed, learning which is the type of grape presented on a vineyard could provide insights for a farmer in order to distinguish which should be the best way to treat these crops. Moreover, during the harvest process farmers might desire to differentiate from grapevine varieties to increase vine's quality. Here, the segmentation based on NDVI thresholding can be extremely accurate. In fact, datasets for each vineyard were collected for farmland which is only composed by one variety. Therefore, there's no need to identify the ROI with low DOF samples. The PRS for dataset $\#1$ has shown serious difficulties when distinguishing between Olive trees and Eucalyptus. Notwithstanding, the overall classification task has resulted in satisfactory results.

[6]Performances labeled as W1, W2 and W3 in figure [12] are related to approximated values presented on articles [4], [14] and [13], respectively.

Acknowledgment

This research was partially supported by the FCT (the Portuguese Foundation for Science and Technology) under grant SFRH/BPD/94307/2013 and by EU project FP7-TIRAMISU, grant FP7/SEC/284747.

References

[1] A. A. K. Bhandari and G. K. Singh, "Feature extraction using normalized difference vegetation index (ndvi): a case study of jabalpur city," *2nd International Conference on Communication, Computing And Security*, vol. 6, no. 1, pp. 612–621, June 2012.

[2] D. L. M. S. Angela Kross, Heather McNairn and C. Champagne, "Assessment of rapideye vegetation indices for estimation of leaf area index and biomass in corn and soybean crops," *International Journal of Appiled Earth Observation and Geoinformation*, vol. 34, no. 1, pp. 235–248, February 2015.

[3] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[4] R. S. Debadeepta Dey, Lily Mummert, "Classification of plant structures from uncalibrated image sequences," the Robotics Institute, Carnegie Mellon University, Intel Labs Pittsburgh.

[5] L. P. D. P. D. M. b. B. O. A. F. Giovanna Sonaa, Daniele Passonia, "Uav multispectral survey to map soil and crop for precision farming applications," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B1, July 2016.

[6] E. F. L. G. E. L. A. M. F. P. V. Jacopo Primicerio, Salvatore Filippo Di Gennaro, "A flexible unmanned aerial vehicle for precision agriculture," *Precision Agric*, January 2012.

[7] R. M. G. James Z.Wang, Jia Li and G. Wiederhold, "Unsupervised multiresolution segmentation for images with low depth of field," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 1, pp. 85–90, January 2001.

[8] R. A. C. L. Jeferson Lobato Fernandes, Jansle Vieira Rocha 2, "Sugarcane yield estimates using time series analysis of spot vegetation images," *Scientia Agricola*, vol. 68, no. 2, pp. 139–146, March/April 2011, programa de Pós-Graduaão em Planejamento e Desenvolvimento Rural Sustentável.

[9] A. A. Marc Levoy and N. Willett. (2011, March) Depth of field. [Online]. Available: http://graphics.stanford.edu/courses/cs178-11/applets/dof.html

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, see http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html\#sklearn.tree.DecisionTreeClassifier.

[11] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri *et al.*, "Recent advances in techniques for hyperspectral image processing," *Remote sensing of environment*, vol. 113, pp. S110–S122, 2009.

[12] M. Rathee and A. Vij, "Image compression using discrete haar wavelet transforms," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 3, no. 12, pp. 47–51, June 2014.

[13] C. S. J. S. Roemi Fernndez, Hctor Montes and M. Armada, "Combination of rgb and multispectral imagery for discrimination of cabernet sauvignon grapevine elements," *Sensors*, vol. 13, no. 6, pp. 7838–7859, June 2013.

[14] M. W. Scarlett Liu, Samuel Marden, "Towards automated yield estimation in viticulture," in *Australasian Conference on Robotics and Automation*. University of New South Wales, Australia, December 2013.

[15] H. O. Takeshi Motohka, Kenlo Nishida Nasahara and S. Tsuchida, "Applicability of green-red vegetation index for remote sensing of vegetation phenology," *Remote Sensing*, vol. 2, no. 1, pp. 2369–2387, October 2010.

[16] Tetracam. (2016) Tetracam Inc. 21601Devonshire Street #310, Chatsworth, CA 91311. [Online]. Available: http://www.tetracam.com/index.html