SVM texture classification for tropical vegetation mapping

Sebastien Chabrier[a], Benoit Stoll[a], Jean-Baptiste Goujon[b]

[a]Assistant professor,Gepasud, UPF, FAA'A Tahiti, French Polynesia;
[b]Student, ESEO, France

ABSTRACT

Nowadays, remote sensing is an essential science in French Polynesia because of its extended territory and the remoteness of its 120 islands. There is a strong need to study the vegetation cover and its evolution (biodiversity threat, invasive species, etc.).
A growing satellite images database has been acquired throughout, giving access to very high resolution optical images such as Quickbird data. These data allow accessing the vegetation canopy spectral and contextual information, texture classification has proved to be an efficient tool to map the complex vegetation found in tropical regions.
The main goal of this paper is to propose an optimized SVM multispectral-texture classification method for tropical vegetation mapping.
One of the texture computation drawbacks is the window treatment size, which is related to the largest texture element size. In complex tropical vegetation cover, this parameter leads to very small ground truth learning database, inducing a significant degradation of the classifications accuracy. We propose to increase the thumbnail numbers using an under-sampling method, optimizing the size and the number of the thumbnails.
The other drawback is the high dimensionality of the problem when dealing with multispectral textures. We thus propose to rank and select the most pertinent textures attributes in order to reduce the dimensionality without reducing the classification accuracy.
We first introduce the study context, before exposing preliminary studies on tuning the SVM learning method. The adapted method is then accurately exposed and the interesting experimental results as well as a sample of applications are presented before to conclude.

Keywords: S VM, texture, classification, texture attributes, multispectral.

1. INTRODUCTION

Nowadays, remote sensing is an essential science in French Polynesia because of its territory extent which is as large as Europe with 5 millions km^2 and because of the isolation of its 120 islands summing only 4000km^2.

There is a strong need to study the vegetation cover and its evolution (invasive species, deforestation, forest fire, hurricane impacts, soil degradation, ...). A growing satellite images database has been acquired throughout, giving access mainly to very high resolution Quickbird and Ikonos optical and infrared data.

Using optical remote sensing data allows accessing the vegetation canopy spectral and contextual informations, texture classification has proved to be an efficient tool to map the complex vegetation found in tropical regions.

One of the texture computation drawbacks is the window treatment size which has to be as large as the largest texture element to emphasize. In tropical forests, the size of the trees canopy would need a 20 meters window size on each side, leading to a degradation of the classification accuracy.

The main goal of this paper is to propose an optimized texture classification method for tropical vegetation using a smaller window for the texture attributes computation.

In the first section we introduce the context of this work, while in the next section we expose the results of preliminary studies used to tune our learning method. In the third section the experimental results and a sample of applications are exposed before to conclude.

Further author information: (Send correspondence to Sebastien Chabrier)
Sebastien Chabrier: E-mail: sebastien.chabrier@upf.pf, Telephone: +689 803 805

2. PRESENTATION

Before dwelling on the proposed method, we present the context of this study in terms of tools and data. We first introduce the SVM learning method chosen, then the texture attributes and programming tools are presented. Finally, we introduce the study site along with the available data and the set of classes.

2.1 SVM

In recent years, Support Vector Machine (SVM) is considered as one of the most effective method by many peers when dealing with learning methods.[1]

SVM is introduced by Vapnik[2] and extensively described in,[3,4] and.[5]

The basic principle is simple: suppose we have a training set $\{\mathbf{x}_i, \mathbf{y}_i\}$ where $\mathbf{x}_i$ is the texture attributes vector describing a region or an image represented by its class $\mathbf{y}_i$ in the learning database. The goal of supervised classification is to identify the class of a local image area. For two classes problems, $y_i \in \{-1, 1\}$, the Support Vector Machines implement the following algorithm. First of all, the training points $\{\mathbf{x}_i\}$, are projected in a space $\mathcal{H}$ (of possibly infinite dimension) by means of a function $\Phi(\cdot)$. Then, the goal is to find, in this space, an optimal separation hyperplane, in the sense of a criterion that we will define shortly. Note that for the same training set, different transformations $\Phi(\cdot)$ lead to different decision functions. A transformation is achieved in an implicit manner using a kernel $K(\cdot,\cdot)$ and, consequently, the decision function can be defined as :

$$f(\mathbf{x}) = \langle w, \Phi(\mathbf{x})\rangle + b = \sum_{i=1}^{\ell} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{1}$$

with $\alpha_i^* \in \mathbb{R}$. The value w and b are the parameters defining the linear separation hyperplane. In SVMs, the optimality criterion to maximize is the margin, that is the distance between the hyperplane and the nearest point $\Phi(\mathbf{x}_i)$ of the training set. The α_i^* allowing the optimization of this criterion are defined by solving the following problem:

$$\begin{cases} \max_{\alpha_i} \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i K(\mathbf{x}_i, \mathbf{x}_j y_j) \\ \text{with constraints,} \\ 0 \ \leq \ \alpha_i \ \leq \ C \ , \\ \sum_{i=1}^{\ell} \alpha_i y_i \ = \ 0 \ . \end{cases} \tag{2}$$

where C is a penalization coefficient for data points located in or beyond the margin and provides a compromise between their quantity and the width of the margin. Originally, SVMs have essentially been developed for two classes problems ($y_i \in \{-1, 1\}$). However, several approaches can be used for extending SVMs to multiclass problems. The method we choose in this communication, is called *one against one*. Instead of learning N decision functions, each class is discriminated here from another one. Thus, $\frac{N(N-1)}{2}$ decision functions are learned and each one is considered as a vote for the affectation of a new point $\mathbf{x}$. The class of this point $\mathbf{x}$ is the class obtaining the majority votes.

We provide a determination of the kernel and parameters of such SVM for this study in the next section.

2.2 Texture attributes

In order to recognize some vegetation classes, we use the texture present in each available spectral bands.

To characterize these textures, we use the 15 classical texture attributes of Haralick[6,7] which are computed from the cooccurence matrix (see Table 1).

In order to check the ability of these textures attributes to identify the vegetation classes within a region, we study the evolution of these attributes on several image database, introduced therafter.

A1	Angular Second Moment	A9	Entropy
A2	Contrast	A10	Difference Variance
A3	Correlation	A11	Difference Entropy
A4	Sum of Squares: Variance	A12	Information Measures of Correlation 1
A5	Inverse Difference Moment	A13	Information Measures of Correlation 2
A6	Sum Average		
A7	Sum Variance	A14	Information homogeneity
A8	Sum Entropy	A15	Contour information

Table 1. Texture attributes.

2.3 Tools

The software tool used for this study is Orfeo Toolbox* 2.6[8,9] for all the computation of feature extraction or learning and recognizing processes.
The other software used is R† which is a free software environment for statistical computing and graphics.
The computer used is an I7 960 with 6Gb of RAM.

2.4 Data

This study takes place in the Moorea biocode project (http://mooreabiocode.org/) which aims to create the first comprehensive inventory of all non-microbial life in a complex tropical ecosystem. In this project, the GePaSud laboratory has been charged to map the natural vegetation cover of Moorea island using optical, radar and ancillary data.[10] For this study, which focus on the optical texture classification, we use a 2002 Quickbird scene of Moorea (Fig. 1).

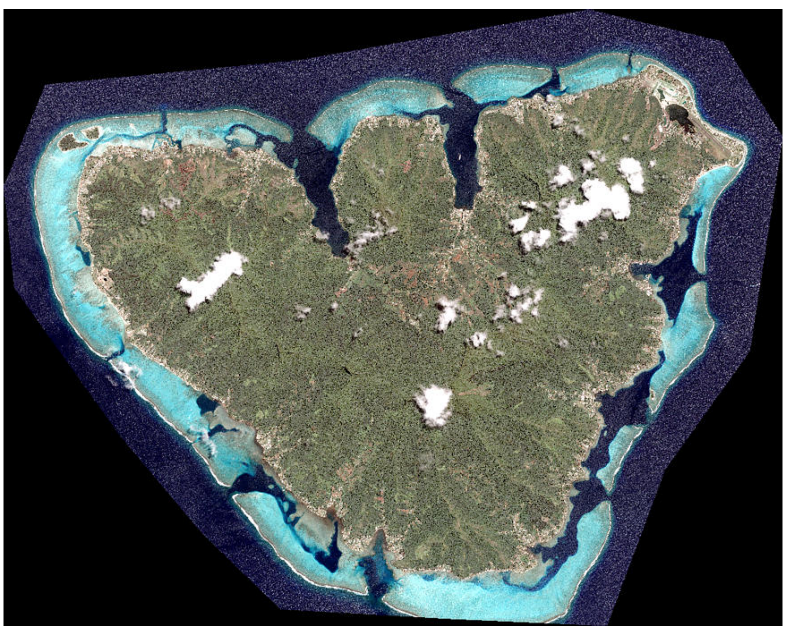


Figure 1. 2002 Moorea Quickbird RGB image.

This Quickbird image has a resolution of 61 cm for its 4 pansharpened bands (RGB and IR) and has been projected in the WGS-84, Zone 6-South system.

*http://blog.orfeo-toolbox.org/
†http://www.r-project.org/

2.5 Classes choice

An extensive number of ground truth campaigns have been conducted all over Moorea from 2009 to 2011. A detailed ground truth database has been crated using a GPS GeoXH Trimble as localization system (reaching a sub-meter precision).

Consequently, 15 main classes of vegetation has been proposed in[11] to represent the Moorea tropical vegetation, but as some of these classes are lacking in our study site, the class set has been downscaled to 6 classes. An example of each one of these 6 classes is presented in Tab. 2.5. Those classes have different texture pattern sizes, some have very fine textures such as *Leucaena* and others have large scale texture such as the *Falcata* trees which crown can be 20meters wide.

The largest texture pattern of the *Falcata* class leads to the use of a 32 per 32 pixel window (400sqm) for accurate texture computation. Finding such large areas of monotypical vegetation is very tricky in polynesian islands complex vegetation cover.

The initial ground truth database is created by extracting 10 Thumbnails (32 per 32 pixels) from the Quickbird image for each one of the 6 classes (Tab. 2.5). This database is further referenced as *B1*.

SVM algorithm needs a very small quantity of support vectors to find the most accurate separation hyperplane, but a sufficient enough amount of learning vectors is needed to statistically find the best support vectors. The previous *B1* ground truth database (60 thumbnails) is too small to find efficiently accurate support vectors for the SVM learning step.

Thus, the main goal of this paper is to determinate a robust method to build an efficient ground truth database considering limitations due to the vegetation cover. This database should be big enough to discriminate and classify accuratly the vegetation classes using the texture attributes.

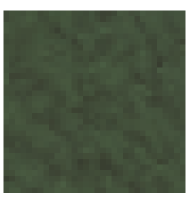
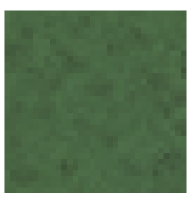
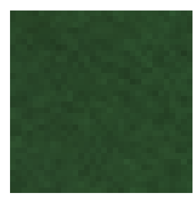
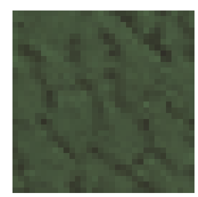
					
C1. *Casuarina*	C2. *Dicranopteris*	C3. *Falcata*	C4. *Leucaena*	C5. Swamp	C6. *Pinus*

Table 2. Examples of the 6 vegetation classes (RGB bands).

3. PRELIMINARY STUDIES

Many studies have been conducted on the optimization of the learning step parameters when using SVM as in.[12] Despite some certainty on a few parameters, such as the kernel choice (Hsu advocates the RBF kernel in[5] for example), the context of learning highly condition the others parameters. Tuning the SVM parameters is thus considered as a preliminary step to our study.

3.1 SVM autoconfiguration and kernel choice

The basic tool used to achieve this study is the OTB library that provides a SVM algorithm and proposes a parameter optimization method given a kernel. This method is iterative and requires two parameters: the maximum number of iterations and the type of kernel which has to be tuned.

The learning step is based on the 60 thumbnails (32 by 32 pixels *B1* database) corresponding to the 10 sub-images per texture class. The verification step is computed on the same 60 thumbnails of the *B1* database,

leading to a percentage of correct recognition called GRP (Good Recognition Percentage). The experiment is conduced using the various Kernel availables.

The Fig. 2 gives an overview of the results. The number of iterations used to optimize the learning step is represented on the abscissa . On the ordinate the results are represented as the GRP (Good Recognition Percentage). Thus, the vertical bars represent the GRP obtained for the various kernels.

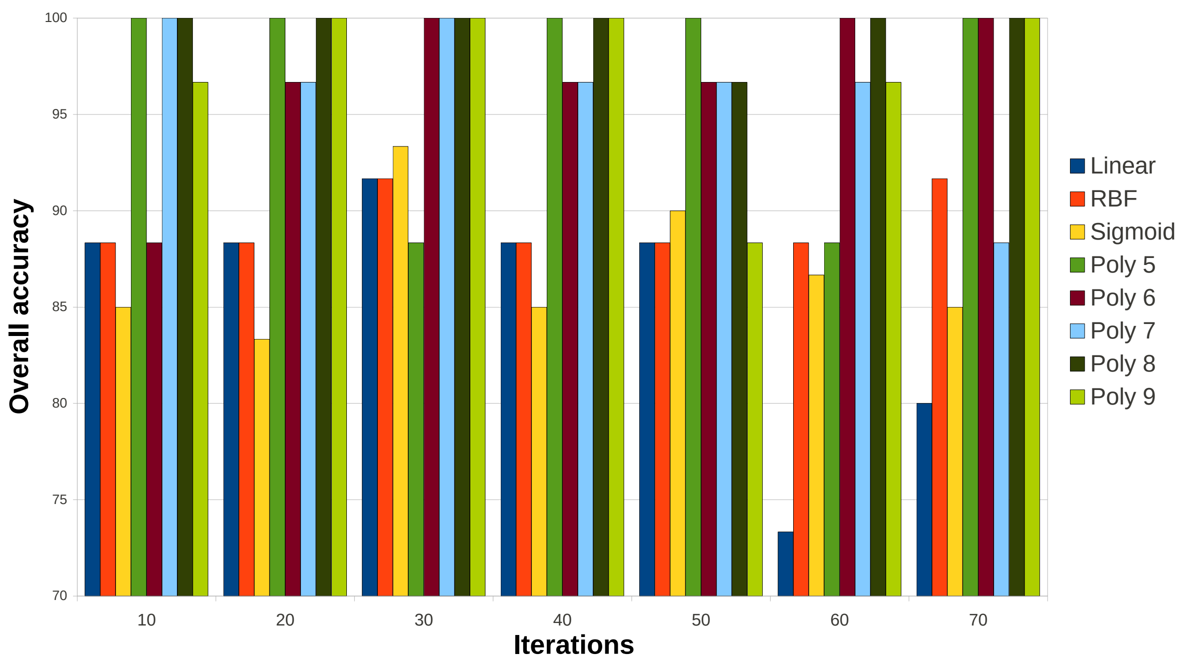


Figure 2. GRP for the set of 60 images of 32 pixels by 32 pixels depending on the kernel and the number of SVM optimization iterations.

By looking at this table, we can first note that the optimal quantity of iterations for the Linear, the RBF and the Sigmoid kernels if 30 iterations. For the polynomial kernels 30 iterations if a good compromise: 4 of them give 100% OA. We can also note another interesting point, if we use too many optimization iterations, it may degrade the quality of the SVM learning as for example in the cases of the linear and sigmoid kernels with 70 iterations. Finally, we decided to use a compromise of 30 optimization iterations which is giving the best overall results.

	5 It	70It	30It			
			8F		60F	
Kernel	60F	60Th	30Th	1920Th	30Th	1920Th
Linear	1”	21”	1”	27”	1”	45”
RBF	3”	1’ 36”	1”	13’ 9”	7”	28’
Sigmoid	40”	3 h 12’	49”	12h	7’	28h 48’
Polynomial	1’	5 h 44’	10’ 48”	20h 40’	8’	31h 45’

Table 3. Computation time of learning and its autoconfiguration based on the kernel used (columns), the quantity of iterations for the SVM optimisation (It), the quantity of features used in each descriptor’s vectors (F) and the quantity of thumbnails used for the SVM training (Th).

Tab. 3.1 reports the overall minimum and maximum computing time for both SVM learning and optimisation, calculated from Fig. 2 experiment.

This table enlight the fact that increasing the quantity of iterations also increases very significantly the computation time for the learning step. In particular, the sigmoid and polynomial kernels require a huge amount of tuning time for 70 iterations considering such a small training set of only 60 images.
If we now consider the quantity of training images for a quantity of iterations set up to 30, we can also notice that it increases drastically the amount of computing time. Once again, the sigmoid and polynomial kernels require a huge amount of time such as around 30 hours for 1920 training images while the linear and the RBF kernels require only less than an hour.

The polynomial kernel is the only one reaching 100% good recognition in Fig. 2, however, as shown in Tab. 3.1, it requires, by far, the longest tuning computation time. Thus, a good compromise seems to be the RBF kernel with a reasonable computation time and decent results such as suggested in.[5]

3.2 SVM fixed-parameters strategy

As seen in the previous chapter, the SVM learning step computation can be time consuming with automatic parameters optimisation, and it increases even more with a bigger training data set.

A method to speed up the computation consists in tuning the SVM's parameters under certain conditions. These conditions are determined thereafter. Then those tuned SVM parameters are used in all the learning steps when the conditions are the same.
Based on the previous results, a SVM RBF kernel with 30 iterations of parameters optimization is used in the following.

3.2.1 impact of the optimization strategy when reducing the thumbnails quantity

In order to get enough thumbnails for this experiment, a sub-sampling of each 32x32 thumbnail is operated giving a set of artificially built 25x25 pixels thumbnails. All the 25x25 pixels thumbnails are extracted giving 64 thumbnails for each 32x32 pixels image.

The artificial database is named *B2* (64 * 6 * 10 = 3840 thumbnails of 25x25 pixels).

Initially, we optimize the parameters of SVM with the 15 Haralick texture attributes on each of the 4 bands with half of the thumbnails (15 * 4 * 64 *10 / 2 = 19200 thumbnails of 25x25 pixels per class), then the recognition is tested on the other half. The very same parameters optimization is operated with less and less of learning thumbnails, always taking the remaining images for the recognition test.

In the same time, the set of SVM parameters previously optimised with the 1920 thumbnails and 60 attributes is used, without re-optimisation for the training and recognition steps on the same batches of thumbnails as above.

The results of these computations are shown in Fig. 3 where the abscissa represents the quantity of thumbnails used and the ordinate represents the rate of correct classification on the remaining images (not used for the training step). We can notice that the results are equivalent and that there is no need to re-compute the parameters optimization whenever one wishes to operate the learning step with fewer thumbnails.

This result is important because, regarding the learning step of the *B2* 3840 thumbnails (25x25 pixels) data base, the SVM training steps takes from 1 minute to 6 hours with automatic optimization and from 1 second to 20 minutes by using the fixed parameters previously determined on half of the original thumbnails.

We showed that a set of parameters can be optimized on the overall thumbnails database, and used efficiently for the learning step on a part of this database.

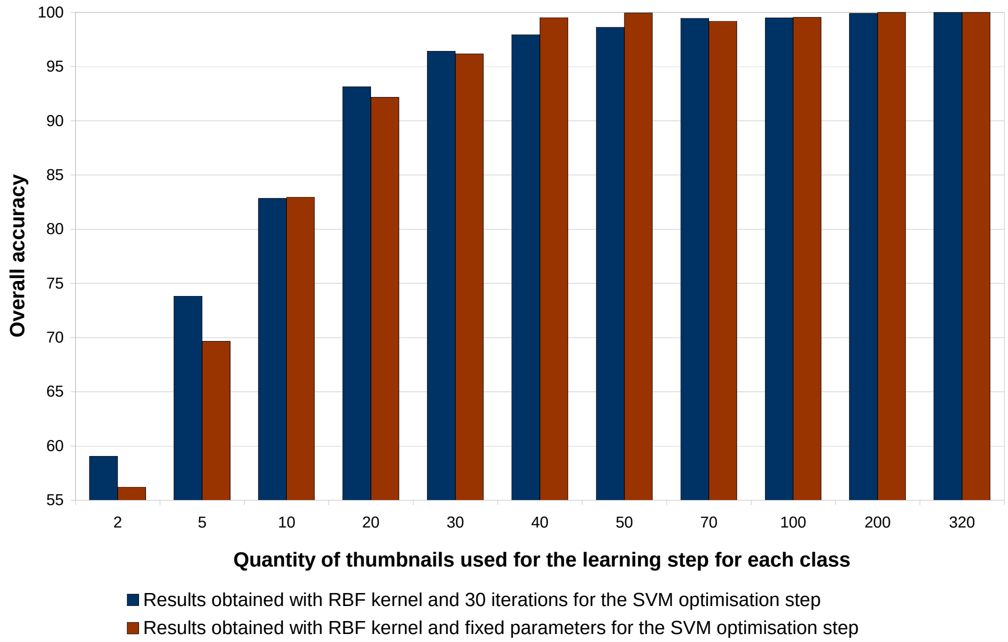


Figure 3. Comparison of fixed parameters for the SVM compared to an optimization each time with a fixed set of attributes in learning step.

3.2.2 impact of the optimization strategy when reducing the texture attributes quantity

The next question we'd like to answer is whether this fixed set of parameters can be used if some textures attributes are removed from the learning step.

As mentioned earlier, we still use here the RBF kernel with 30 iterations for the automatic optimization step of the SVM parameters.

In this second experiment, we take into account the computation results of the texture attributes on the whole set of the *B2* database, i.e. 3840 thumbnails and 15 attributes for each of the four bands resulting in 60 vectors of 3840 attributes.

A Principal Component Analysis (PCA) of these attributes is computed, which helps us to determine the ranking of the most important texture attributes considering the 60 attributes spread over the four bands.

An automatic optimization of the SVM parameters is processed using the 10 and 8 most important textures attributes, and the learning step is operated. Then we compare, in the following graph, with the results obtained by learning using the previously determined parameters (based on the 60 attributes).

The results of these computations are shown in Fig. 4 where the abscissa represents the quantity of thumbnails used and the ordinate the overall accuracy on the remaining images (not used for the SVM training step). We can notice that results obtained by the learning step using the parameters optimized using the complete set of 60 attributes gives worse results compared to a re-optimization of the SVM parameters for each subset of texture attributes.

Thus, according to both tests sets, we can conclude, so far, that for a set of attributes and a given data set, a single set of optimized parameters can be used for SVM training regardless the amount of thumbnails used for training. Whereas, if we need to vary the number of attributes, it is necessary to re-process an automatic optimization step each time.

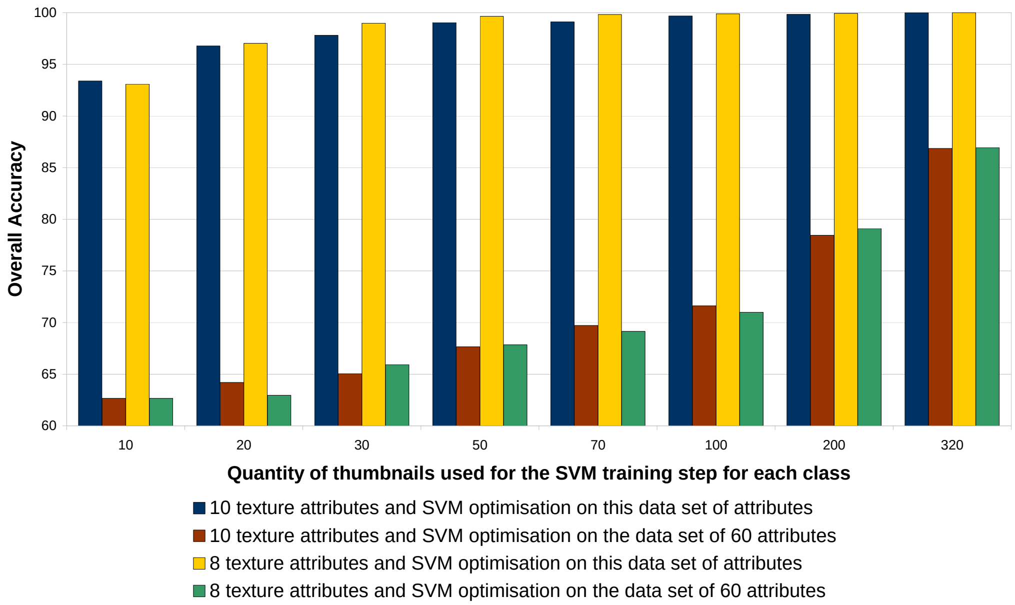


Figure 4. Comparison of fixed parameters for the SVM learning between 3 different sets of texture attributes.

4. EXPERIMENTAL RESULTS

Two issues for learning classes in remote sensing in tropical environment are generally encountered, among other things: The lack of relevant data for the learning step and the need of a wide window to compute the texture attributes. We therefore propose two methods, exposed in the next section, to overcome these problems.

Then, within the two sections thereafter, the amount of information necessary to operate a relevant learning is determined, as well as the texture attributes necessary for an efficient recognition of our vegetation classes.

4.1 Decreasing the size of the shifting windows and augmenting the quantity of learning data

The first issue we wish to solve here is related to the size of the window for computing the texture attributes that is supposed to contain at least the largest pattern to recognize. In our case, a window size of 32 pixels by 32 pixels is needed because of the Falcata's crown which diameter may be as large as 20 meters. The largest window size drawback is that the bigger the texture attributes computing window, the less accurate the classification results are. It would thus be interesting to reduce the thumbnails size for the learning step.

The second issue is that it is often difficult to locate enough thumbnails for the training step: the more we have, the better the detection is because we have statistically more chances to find the best support vectors.

But finding large enough area of monotypical vegetation is often complicated in tropical islands environment because of the heterogeneity and the complexity of the vegetation cover.

The idea suggested here is to enlarge the original ground truth images database by processing sub-images, thus obtaining a composite database of lower spatial size but more numerous sub-images. In a 32 by 32 pixels thumbnail, 64 different sub-images can be extracted and used for the training or the recognition steps of the SVM.

In this case, with 10 extracted and validated images per class, we can compute 64 shifting windows of 25x25 pixels * 10 images per vegetation class * 6 classes = 3840 sub-images.

This procedure responds to the two above-mentioned problems at once: the accuracy of the classification is increased by reducing the size of the window used for the texture attributes computation and the amount of

data used for the learning step in increased in the same time.

To check the efficiency of this methodology, all the texture attributes are computed on the 4 bands of the 60 images of 32x32 pixels (*B1* database) by using the SVM RBF kernel as well as an automatic optimization of its parameters over 30 iterations.

The result is that 4 of the 60 pictures of our learning database are not recognized correctly giving an overall accuracy of 93 %.

We then compute all the 15 texture attributes on the 3840 4-bands sub-images of 25x25pixels (*B2*), the RBF kernel for the SVM training is used as well as an automatic parameters optimization over 30 iterations. A SVM learning step of half of the 25x25 pixels thumbnails is computed followed by a recognition on the other half. It appears that all these thumbnails have been well classified.

We can then conclude that in our case the reduction of image size for learning and increasing their quantity doesn't reduce the learning ability of the SVM algorithm.

4.2 Reduction of the training set

However, despite the good recognition rates obtained previously, the numbers of training thumbnails is important. For computation time issues, we'd like to reduce the size of the learning database, but we need to know what is the incidence of the size reduction on the SVM recognition rate.

In this purpose, we extract randomly a number of 25x25 pixels thumbnails in each class out of the 32x32 pixels original *B1* database. From 5 to 320 thumbnails are extracted per class. The training step is performed using the same parameters set as above.

Then, in each case, a SVM recognition step is processed on the remaining thumbnails. All the computations are performed 10 times and the results are averaged. The Fig. 5 presents the overall results.

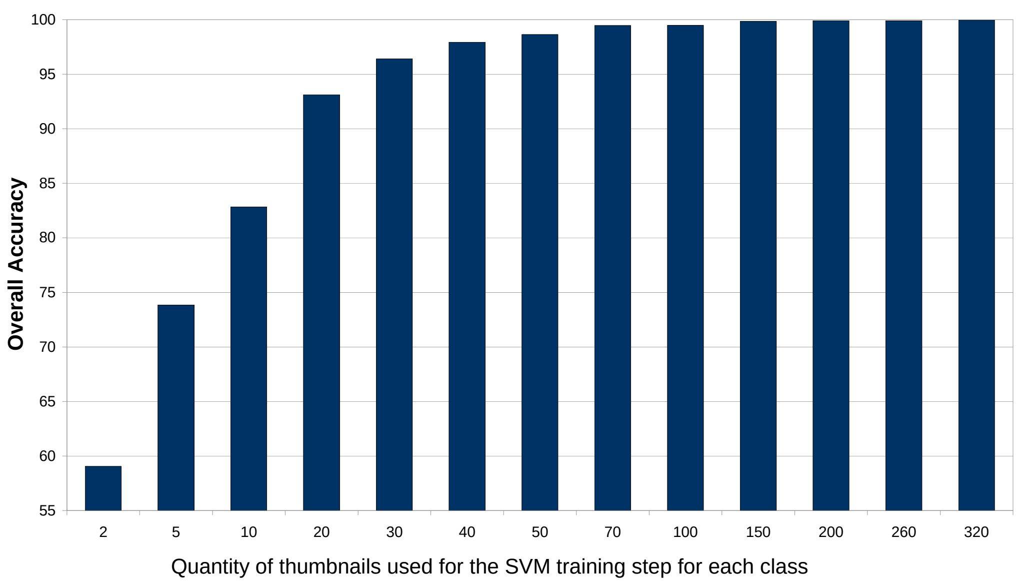


Figure 5. Impact of the training-set reduction.

In Fig. 5, the x-axis represents the quantity of thumbnails randomly taken from each vegetation class, while the y-axis shows the overall recognition accuracy for all the remaining thumbnails of 25x25 pixels (not used for learning). The overall accuracy is at least 99.5 % from 320 to 70 thumbnails per class (i.e. 1920 to 420 thumbnails

of 25x25 pixels for the overall learning). We keep a decent recognition rate down to 30 thumbnails per class for the training step with an overall accuracy of 96.5 %. With less learning thumbnails, the rate of good recognition drops relatively quickly. In conclusion on this chapter, we can considerably reduce the number of thumbnails used for the SVM learning step without decreasing the recognition rate of the algorithm.

4.3 Reduction of the number of learning texture attributes

This section proposes to further improve the previous results by reducing the quantity of textures attributes necessary for the same results.
In order to reduce the amount of texture attributes for the learning step, we need to rank them in order to select a set of the most significant ones. To achieve this goal, we use the R software to compute PCA on the set of attribute vectors for the four bands of each image of 25x25 pixels. This way, we made a reduction of the 60 vectors into 5.

From the obtained results, the importance of each attribute is considered over each band proportionally to the importance of each one of the five generated bands in order to rank the 60 attributes.

Tab. 4.3 shows the first 10 attributes sorted this way.

ACP rank	Texture feature	Band
1	Sum Entropy	blue
2	Entropy	blue
3	Entropy	red
4	Variance	green
5	Sum Entropy	red
6	Contrast	green
7	Contrast	infrared
8	Sum Entropy	infrared
9	Contrast	red
10	Variance	red

Table 4. Texture features rankins according to PCA computation over the 60 original features on the 4 bands RGB/IR.

The six classes of our data set are then learned with half of the thumbnails at our disposal (1920) for 4, 6, 8, 10 top attributes classified by PCA. The same procedure is applied on the 30 attributes of both infrared and green bands as well as on the 60 attributes. Then, by keeping the same SVM parameters, the training step is processed over less and less images for each class. Fig. 6 shows the recognition results of these computations on the remaining images.

The first thing that we can state in the Fig. 6 is that, over 100 images training set, the recognition results by using all the 60 texture attributes or only the 8 or 10 first attributes of the ACP, are very similar within 1% close.

Under 100 images in the training set, we can notice that using only 8 or 10 PCA texture attributes gives better results than using all the attributes. That could be explained if we consider that using the 60 attributes values brings a lot of overinformation which could degrades the results.

Using only the first 6 or 4 PCA texture attributes, gives, in all cases, worst results than the 8 or 10 first ones.

Noticing that the infrared and the green bands seems to contain more information that the two other ones, the OA is computed using the 15 texture attributes only on these two bands. The results are the worst ones in all the cases.

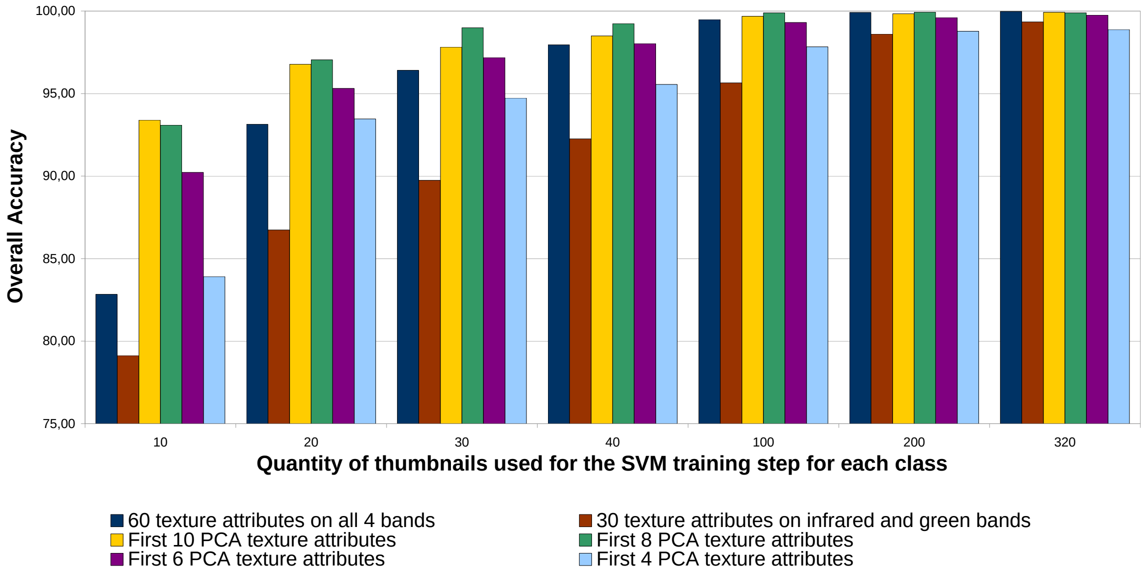


Figure 6. Impact of the texture attributes quantity reduction

Finally, using the first 8 PCA texture attributes and 30 learning images seems to be the best compromise, and brings an OA of more than 98% when training on a part of the images database and recognising on the rest of the images.

To validate this study, we keep the same protocol as above, except that we did not perform recognition on the remaining images but on a new set of thumbnails of 25 pixels per 25 pixels extracted from 30 new images of 32x32 pixels (5 per class) validated on the field and non-overlapping with the *B1* database. Fig. 7 shows the recognition results of these computations on the new thumbnails set.

The results obtained in the previous section are confirmed even on a new set of data for recognition. Namely that 30 images of 25x25 pixels per class are enough to obtain a high recognition rate.

In addition, we can see that using the first 8 attributes of the ACP is sufficient to obtain convincing results (more than 95% OA). Using 100 learning images could lead to a slightly better OA of more than 97%.

5. APPLICATION

For each class, a thumbnail corresponding to an area of size 100 pixels by 100 pixels is extracted from our Quickbird image. Each Thumbnail contains nearly a single type of vegetation class.

The parameters proposed in the previous sections are used, namely the RBF kernel, the previously determined eight texture attributes and 30 learning thumbnails par class in order to learn the six classes of the section I.

Images classification are then computed by shifting a window of 25x25 pixels wide inside them.

For each window shifting, the 8 selected texture features are computed and the resulting class is assigned to the central pixel of the window. Thus a 75x75 pixels wide results is obtained. Tab. 5 displays the six original images and the six classification results obtained.

As displayed in the table, the original images are represented with a lighter border. They are shown this way to differentiate the classification results from the complete original image.

As one can see, the results are mostly homogeneous with the class corresponding to the original image content.

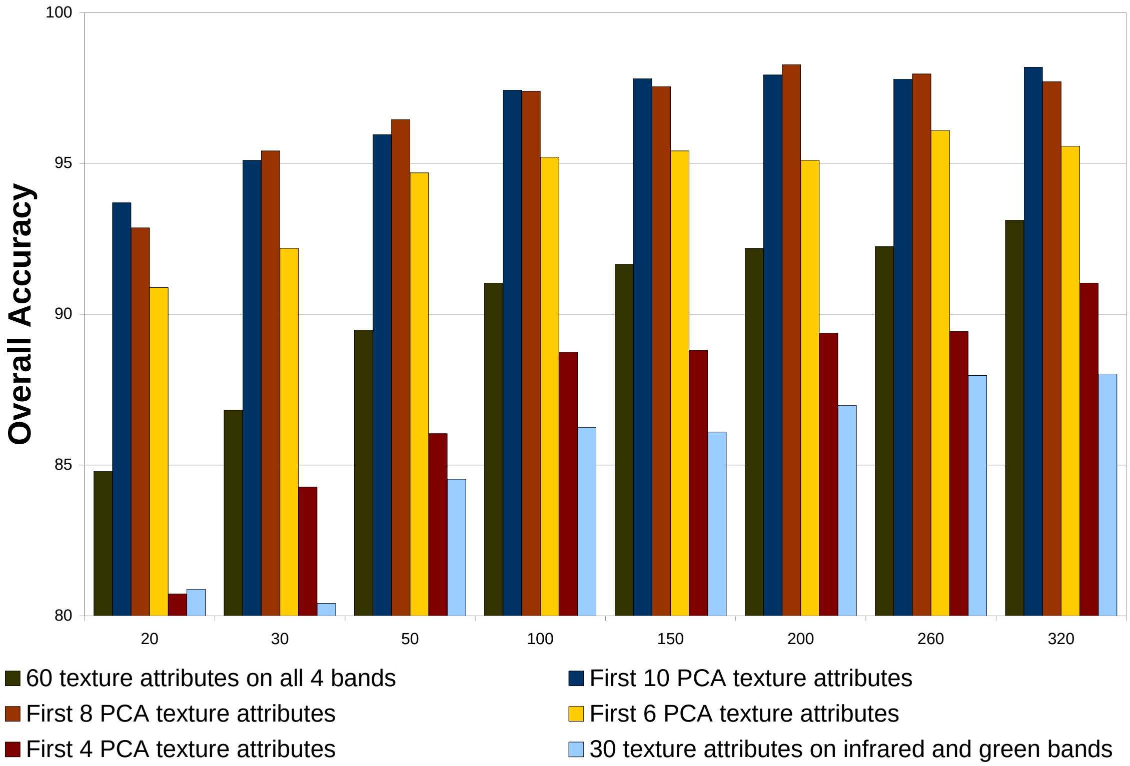


Figure 7. Validation of the texture attribute reduction methodology.

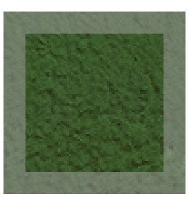
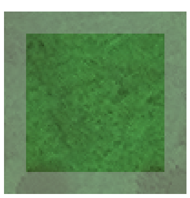
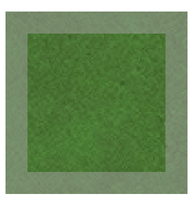
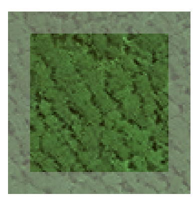
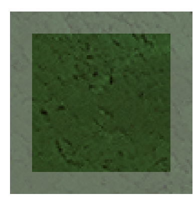
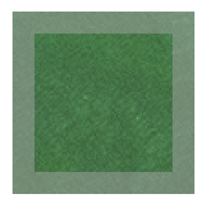
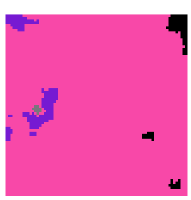
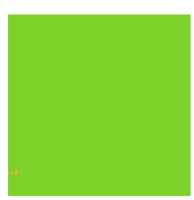
 A. *Casuarina*	 B. *Dicranopteris*	 C. *Leucaena*	 G. *Pinus*	 H. *Falcata*	 I. *Swamp vegetation*
 D.	 E.	 F.	 J.	 K.	 L.

Table 5. Example of images classification results. original images: A. B. C. G. H. I., respectively classification results: D. E. F. J. K. L. .

6. CONCLUSION

The main goal of this paper is to propose an optimized SVM multispectral-texture classification method for tropical vegetation mapping. To achieve this goal, five important results were drawn.

The first and the second ones deal with the optimisation of the SVM learning step. First, with the objective to optimise the SVM parameters on a complete image database for a given features set, the same parameters can be used for the SVM learning step on a subset of these images. Second, with the aim to reduce the quantity of texture features used for the SVM learning step, it is necessary to operate the SVM parameters optimisation for each new set of texture attributes.

The third and the fourth results shown in this paper solve two problems: First of all, using sub-images of the learning images set for the learning step increases the quantity of data available for the SVM training step. There's statistically more chances to find the best support vectors for the SVM separation and thus it gives better results than using the original bigger images. Second of all, there's no longer need to find plenty of large image areas containing a full pattern of the texture to be learned. With our method, only a few of them is enough to ensure a good learning and recognition.

The last important result deals with the previous experiments conclusions on learning protocol and tuning parameters strategy applied on a real case. As shown in the experimental results, using only 30 sub-pictures for each class to be learned and the ten first PCA texture attributes, we obtain more than 95% OA on a completely new data set to be classified.

Acknowledgment

The authors would like to thank the french polynesia government SRD delegation and research delegation and the BIOCODE program for their support. They also would like to thank Robin Pouteau for its useful discussions and the ground truth campaign data.

REFERENCES

[1] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, pp. 247–259, 2011.

[2] V. Vapnik, "Statistical learning theory," in *New York: Whiley*, 1998.

[3] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Disc.*, vol. 2, pp. 121–167, 1998.

[4] B. Schölkopf and A. Smola, "Learning with kernels," Cambridge, MA: MIT press, Tech. Rep., 2002.

[5] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification department of computer science and information engineering," NAtional Taiwan Univ., Taiwan, Tech. Rep., 2009.

[6] K. S. R. Haralick and I. Dinstein, "Textural features for image classification," *IEEE Trans. Systems Man Cybernet*, vol. 3, pp. 610–621, 1973.

[7] R. H. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision Graphics Image Processing*, vol. 29, pp. 100–132, 1985.

[8] J. I. E. Christophe and A. Giros, "Orfeo toolbox: a complete solution for mapping from high resolution satellite images," *The XXI Congress, The International Society for Photogrammetry and Remote Sensing, ISPRS'08, Beijing, China*, pp. 1263–1268, 2008.

[9] J. Inglada and E. Christophe, "The orfeo toolbox remote sensing image processing software," *IEEE International Geoscience and Remote Sensing Symposium, IGARSS'09, Cape Town, South Africa*, 2009.

[10] R. Pouteau and B. Stoll, "Svm selective fusion (self) for multi-source classification of structurally complex tropical rainforest," *IEEE Journal of Selected Topics in Earth Observations and Remote Sensing*, vol. 5, pp. 1203–1212, 2012.

[11] R. Pouteau, "Etude de la phytogéographie des îles hautes de polynésie française par classification svm d'images multi-sources," Ph.D. dissertation, Université de la polynésie Française, 2011.

[12] S. Liu, "Ieee international conference on mechatronics and automation," pp. 509 – 513, 2008.