SVM method

RAFIK AREZKI

16/04/2020

The segmentation of multispectral and thermal images with the SVM method

La segmentation des images multispectrales et thermiques avec la méthode SVM

SVM Method :

We are interested in the method a classic SVM class method of uni and multivariate classification. She's from the class of supervised machine learning methods.

This class of SVM methods has been shown to be powerful and efficient for performing segmentation or object detection tasks in an RGB image. They are also widely used for multivariate classification tasks.

The principle of these methods is to construct a separating hyperplane which effectively separates two sets of individuals (pixels) given. Once trained and configured, it can be used to make predictions on other data of the same type.

These methods work as follows:
1. First, they look for all the hyperplanes in the sets. These hyperplanes are called border supports of the sets.
2. Then, in a second step, they choose the optimal margin hyperplane, ie the hyperplane separating the sets at a maximum margin (see, Next figure).

The objective of the study

In this part, we used the SVM method for image segmentation on multi-spectral and thermal images (with seven wavelengths, see the folder of images and masks in tif). The goal was to separate the leaves of trees (apple trees) from the rest in a vegetarian environment on multi-spectral and thermalimages.

We were able to load only a small part of our dataset. The data annotation was done on the ERDAS@IMAGE software. This annotation was done manually, we selected the pixels corresponding to the leaves of the trees.

B1. . . B6 are the different wavelengths

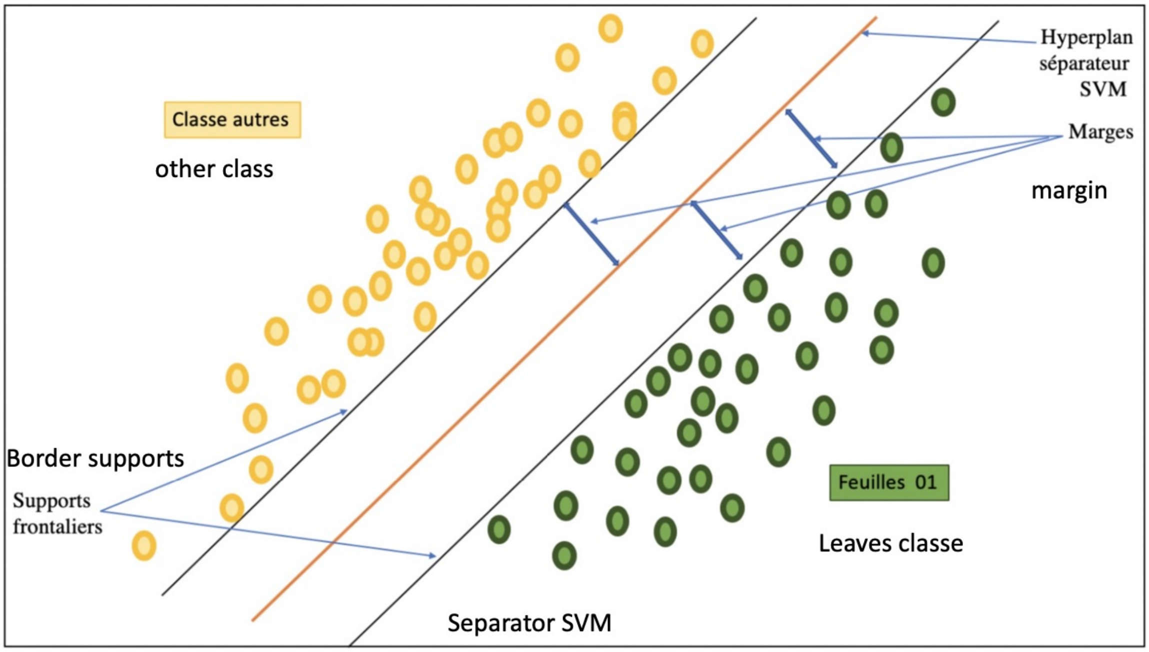

Figure 1: SVM method

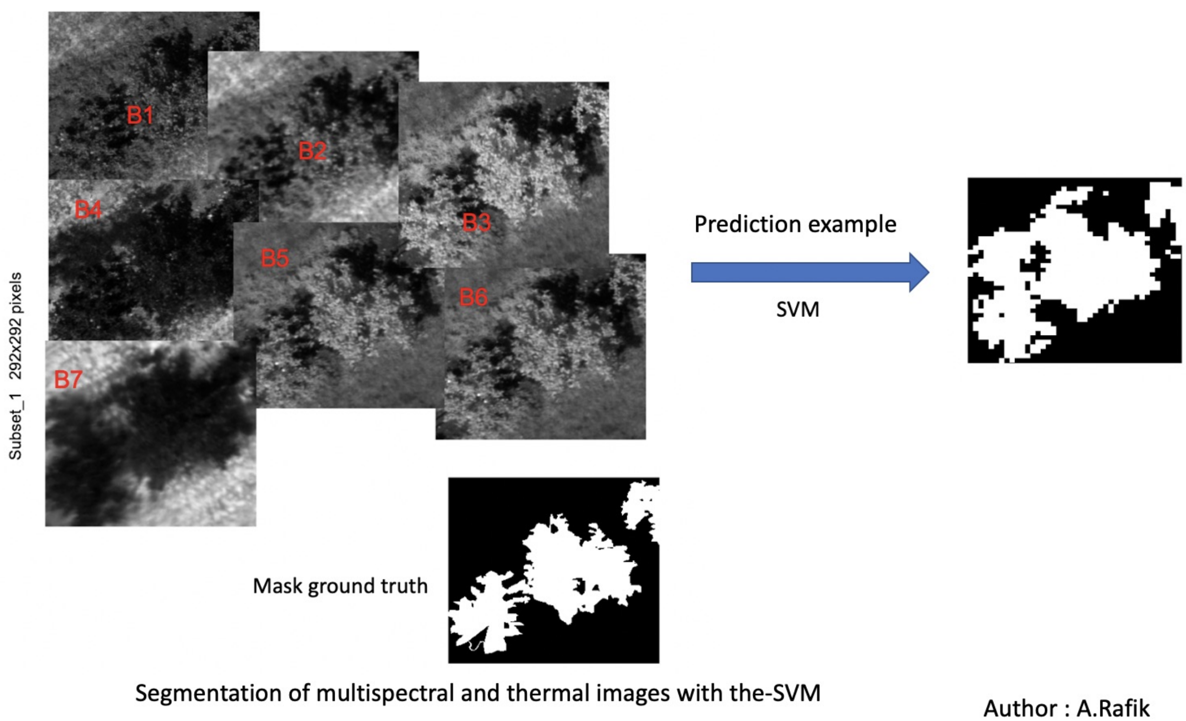

Figure 2: visual segmentation diagram with SVM

implementation of the method SVM

This implementation of the SVM method was done under python using the *scikit-images* and *scikit-learn*Libraries. We used the program written by B. Mathieu10 for locating objects in an RGB image. We have improved the model written by used By [Bérengère Mathieu - published 10/31/2017] ("https://makina-corpus.com/@@search?Creator=bmt") for locating an object within an image. We adapted this model to our dataset.

In this algorithm, the SLIC (Simple Linear Iterative Clustering) function is used. This function is used to split an image into super-pixels to reduce the size of the data training, from the pixel scale to the super-pixel scale, while keeping as much information as possible about these images.

It works as follows: First, it groups the closest pixels (similar in color, similar in texture...) Into rectangular and regular super-pixels. These super-pixels will be described by their average color
and the location of its barycenter.

To train the SVM model, we provided 80% of normalized images as inputs and their masks as outputs. We applied the SLIC method on all images and their masks to build two kinds of matrices: a matrix that we have named X of barycenters of super-pixels (multispectral and thermal vectors) and a vector Y of classes "sheets" and "others" corresponding to the super-pixels. In the end, the SVM model was trained on the *X* matrix of the barycenters as input and the vector *Y* a output.

the test was done on 20% of data.

The different manipulations :

we have stored part of our dataset in the data -> data train and data test or validation folder.

Training model SVM :

To train the SVM model, simply add $Train = True$ for the train_test function in the *main.py* script. This script uses the $Construct\_XY$ function of the training vectors of the script $train\_SVM$. This script simply loads the masks, it preserves the most interesting superpixels, then saves them in NumPy.npy format in two vector X inputs and in Y output.

Prediction with SVM:

For the prediction we use *pred.py*, the masks will be saved in the same files of the tif images.

Accuracy of trained model SVM:

To measure the accuracy or the prediction accuracy of the trained model, we use the measurement function in the *measure_Functions.py* script.

We used the script $train\_test\_SVM$*.ipynb* to apply the different manipulations and display them on jupyter with the command “jupyter notebook”.

I would like to thank everyone who contributed to this study.

Author: rafik AREZKI

Email: arezkirafik@gmail.com