CILAMCE 2017
Florianópolis - Santa Catarina - Brazil

Study of pattern recognition techniques for identification and counting of individual seedlings in eucalyptus spp plantations from high definition aerial images

Guilherme P. J. Franzé

Luiz C. S. Góes

Éder A. Moura

guipjf@ita.br

goes@ita.br

edermoura@gmail.com

Instituto Tecnológico de Aeronáutica

50 Praca Marechal Eduardo Gomes, 12228-900, Sao Paulo, Sao José dos Campos, Brazil

Emanuel R. Woiski

Douglas. D. Bueno

woiski@dem.feis.unesp.br

douglas@mat.feis.unesp.br

Universidade Estadual Paulista "Júlio de Mesquita Filho"

56 Avenida Brasil Sul, 15385-000, São Paulo, Ilha Solteira, Brazil

Abstract. *An actual business case of seedling detection and counting from a high-definition picture taken by a spectral sensor onboard an UAV from an Eucalyptus spp plantation stand of approximately 62 acres (25 hectares) was provided by Eldorado Brasil. With the help of Scientific Python Libraries from the literature, an appropriate methodology was developed for the automatic counting of individual seedlings applying HSV color space conversion and extracting the corresponding histogram as feature. The algorithm was properly validated on the training data, which is a significant area from the whole plantation stand and segmented it into counting manageable 12 tiles in order to study the difference of reflectance for each segment. Then, in order to test the model, we chose a completely new tile, in any way as part of our training data, taken from some region of the mosaic. We decided to use the same samples as for training data in order to investigate their representative power. The results were considered very encouraging, stimulating future works in this line of research.*

Keywords: *Eucalyptus spp, Python, Image processing, UAVs, Binarized classification*

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are rapidly becoming one of the most important resources available to scientists to collect data that is hard to gather using other more accessible platforms. There are a number of reasons for that. UAVs can flight over risky areas because there is no pilot or scientist on board. They can be designed for long range and high payload flights, carrying a variety of interchangeable high resolution imaging instruments. Since technology is becoming cheaper and widely available − in many cases open-sourced hardware and software − they are recoverable for maintenance and upgrades in its sensors and communication systems. Comparing UAVs to satellites, they fly at much lower altitudes allowing for higher resolutions and finer details in the collected images, at reduced costs [Wegener et al., 2004].

A large amount of images are collected daily by UAVs and other platforms. However, like any data, images alone represent nothing without the proper use of pattern recognition, reconstruction, and interpretation techniques in order to extract business intelligence (BI) from them, allowing the decision maker to take the best action in timely manner.

With image processing techniques, it is possible to extract global as well as local quantitative and qualitative information from pictures in general. In order to cover larger areas using limited-range UAVs, numerous georeferenced aerial images are recorded during different periods of the day and merged into one extensive picture, also called mosaic. This should be the main goal of the image processing in precision agriculture (PA): to be applied in order to evaluate large areas at a reasonably short period of time and at low cost.

1.1 Motivation and objective

The high demand for proper use of pattern recognition techniques motivated us to work in one of the problems proposed by Eldorado Brasil[1], which is a large and important company of the pulp industry in Brazil. Therefore, the objective of the present work is to study and evaluate methods of image pattern recognition to discriminate and count objects in a mosaic aerial image containing individual seedlings within an eucalyptus plantation stand.

1.2 Paper organization

In Section 2, we describe the problem to be explored and discuss its implications. In Section 3 we explain the development of the algorithm and discuss its application to a properly selected *training* data set. Defining some statistical parameters, we show how to choose an optimum value for the sole hyper-parameter. Section 4 contains a report on the choice of a completely new tile that was taken from some region of the mosaic, present in Fig. 10, our estimatives and the time spent to do it. In Section 5 we conclude describing some findings and outline ideas for future work.

2 THE SPECIFIC PROBLEM TO BE APPROACHED

Imagine the following scenario: a large company makes its profits from extensive eucalyptus plantation. Business information on the quality of the crop regarding seedling uniformity,

[1] http://www.eldoradobrasil.com.br

distribution, and quantity data should be acquired by the enterprise in order to take the best decisions in space and time, mandatory to increase its profit margin. Since the plantation stands are very large and numerous, fast and accurate seedling counting represent a permanent issue for those companies.

The idea of this work is to develop an algorithm to automatize the seedling counting stated in the scenario described above using high-definition images taken by UAVs.

To complete this purpose, a mosaic aerial image assembled from a collection of pictures taken by UAV flights covering an Eucalyptus stand was provided by a large company with its headquarter close to Ilha Solteira - Sao Paulo. This image has the composition of the near infrared (NIR), green, and blue bands, but represented in the RGB, that is, a fake RGB color space.

3 THE ALGORITHM AND APPLICATION TO TRAINING DATA SET

3.1 HSV color space approach

HSV stands for Hue-Saturation-Value and it is a more natural and linear color space to describe the transition of colors than the conventional RGB. Thus, with help of the OpenCV library [Itseez, 2015], a code was implemented to read any image within a directory and convert it to the HSV color space. Notice that OpenCV reads images in BGR color space rather than RGB.

As we can see in Fig. 1-(b), using the HSV color space, distortions and variations in reflectance and radiance of the soil and everything on it become quite visible. Note that small displacements in any of the HSV bands don't change colors abruptly as it would be for any RGB band. Even though the range of intensity in the Hue band is short, it is possible to depict in Fig. 1-(d) two major peaks that will be discriminated in order to classify seedlings from other things within the image.

In Fig. 1 we compare the same image and respective histogram in the RGB (a),(c) and HSV (b),(d) color spaces. In (d), there is an intensity range overlap making it somewhat imprecise the definition of what seedlings are and what are not, making it harder to decide about a threshold.

The block diagram of Fig. 2 represents all the steps the script needs in order to generate a binary mask out of a mosaic image and some representative samples. For that, a methodology was implemented using some of the Scientific Python libraries. Basically, a script was developed to convert an image from the fake RGB to HSV color space. Following that, a code was defined to read the image from which the user wants to obtain the number of seedlings. In addition, the algorithm is ready to be trained with a representative number of seedlings sampled by the user from the mosaic aerial image, as it will be detailed within Subsection 3.4.

Both image and samples were converted into the HSV color space, as it is depicted in the block diagram of Fig. 2. A mean value and standard deviation were found for each band for each sample, and then, an average of those values is calculated considering the whole sample set. Thus, we ended up with three mean and three standard deviation values out of this process as important features to generate the binary mask present in Fig. 2. For that, a range of maximum and minimum values of each intensity of the color space was defined. An arithmetic operation, Eq. 1 and 2, were applied in order to determine the threshold interval, that is, the pixel intensities

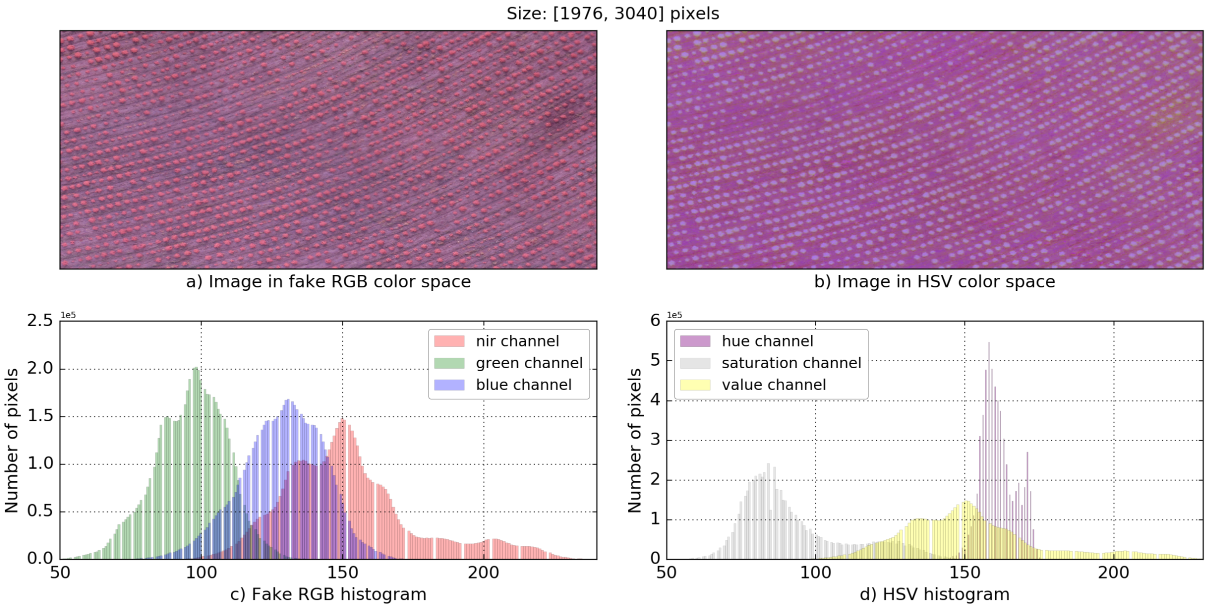

Figure 1: Training data in fake RGB (a) and HSV (b) color spaces with their respectives color intensity histograms (c) and (d)

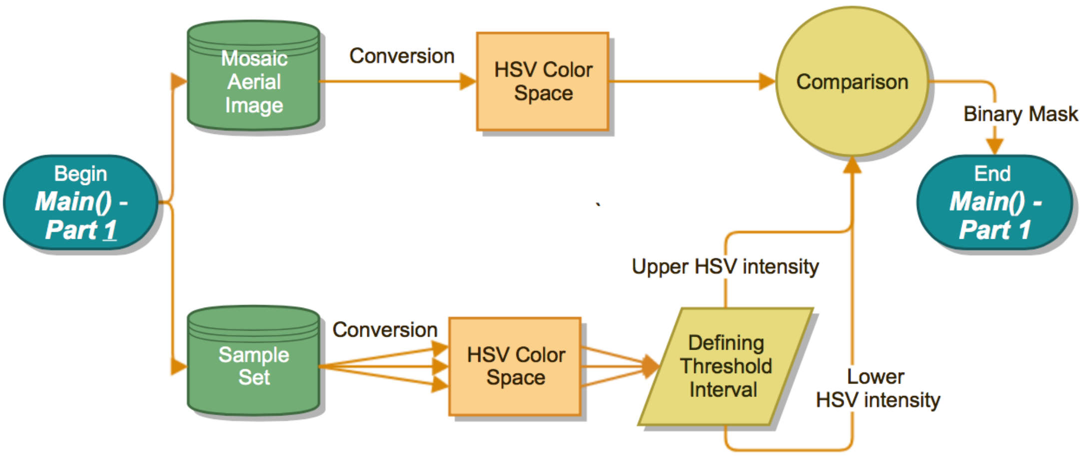

Figure 2: Block diagram of the HSV color space approach to main function part 1.

within the established range are turned into 255, and the remaining are converted into 0. The pixels in the resulting binary mask will be white to the possible seedlings and black to background (everything else).

$$L^{upper} = \bar{X} + f\sigma \quad (1)$$

$$L^{lower} = \bar{X} - f\sigma \quad (2)$$

where $\bar{X}$ is global mean, f is our sole hyper-parameter to be covered in Subsection 3.4, and σ represents the standard deviation for all samples.

The units of the sampling process have to be representative of the whole area covered within the mosaic in order to take into consideration the variations in reflectance and radiance of the soil and everything else during the process of aerial picture recording.

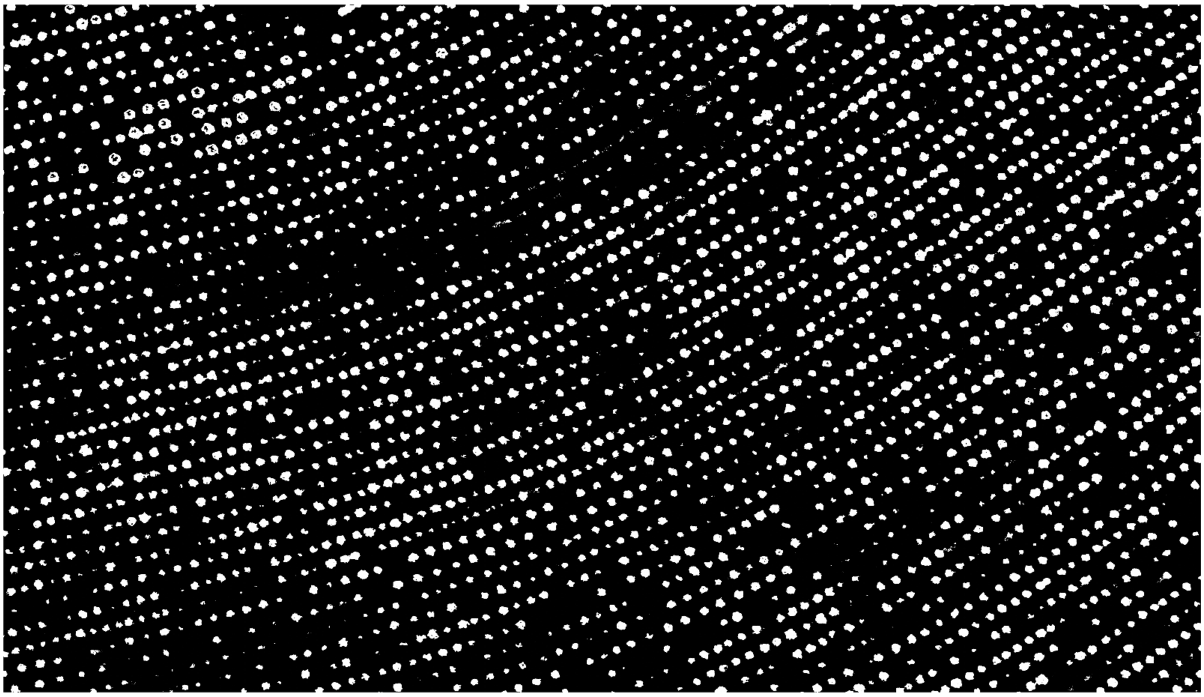

Figure 3: Binary mask generated from HSV color space image.

In Fig. 3 a binary mask, generated from Fig. 1-(b), is presented. The white blobs were assumed to be seedlings representations and they are quite noisy. To proceed further, the mask should suffer some morphological treatment - blurring, opening, and closing, sequentially - in order to obtain more symmetrical blobs as well as to suppress noise, as it will be described in the next Subsection.

3.2 Block diagram

The block diagram in Fig. 4 depicts the logical structure of the algorithm developed. Starting the code, a folder is created and labeled using a figure name as well as the timestamp and the hyper parameter value in order to make that folder uniquely identifiable at a glance. In sequence, the user has to make a choice about splitting or not the image. If affirmative, a number of tiles (image partitions) must be proposed by the user, leaving up to the system to evaluate an optimum number considering that the images are always split into exactly equal parts, even if this would mean creating more than the requested number [Dobson, 2013]. In the main directory, a folder will be created for each tile containing the tile image itself together with another folder called ”Samples” that needs to be filled out with a representative number of seedlings. Otherwise, should the user opt for not dividing the image, a ”Samples” folder will be created and, together with a copy of the image, placed inside the main directory. Regardless the choice, notice that the main function *Main()* of the algorithm, responsible for the actual counting process, will run as many times as the number of tiles.

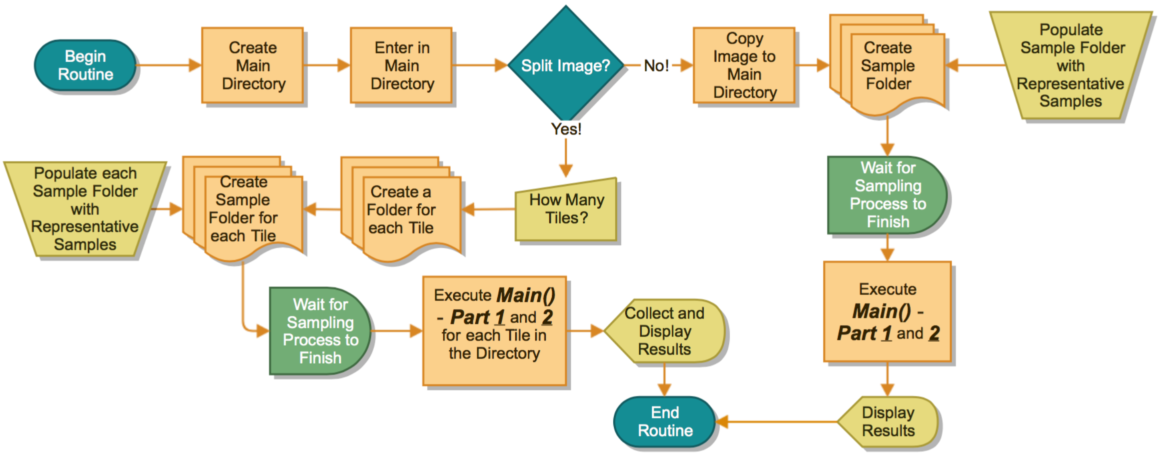

Figure 4: Overview of the logical interface structure.

The *"Execute Main() - Part 1 and 2"* block in the logical structure of Fig. 4 is divided into two parts. The first part, corresponding to the binary mask generation process and presented in Fig. 2, was already described and discussed in Subsection 3.1. The second part, depicted in Fig. 5, starts with the binary mask generated previously as an input. The resulting binary mask is full of noise, therefore, it has to be cleaned up (*denoised*). This process involves the application of *closing*, *blurring* (a type of linear spatial filtering), and *opening*, sequentially. Finishing denoising, the process of seedling identification begins with two functions: *distance_transform_edt* [Jones et al., 01] and *peak_local_max* [van der Walt et al., 2014]. The first is responsible to determine the shortest euclidian distance from every white pixel to the black background, what generates an array whose components values are a nonlinear function of the vegetation biomass [Haboudane et al., 2004]. Euclidean distance (E_D) is the arithmetic operation represented by Eq. 3.

$$E_D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3}$$

where (x_i, y_i) is the pixel i coordinate.

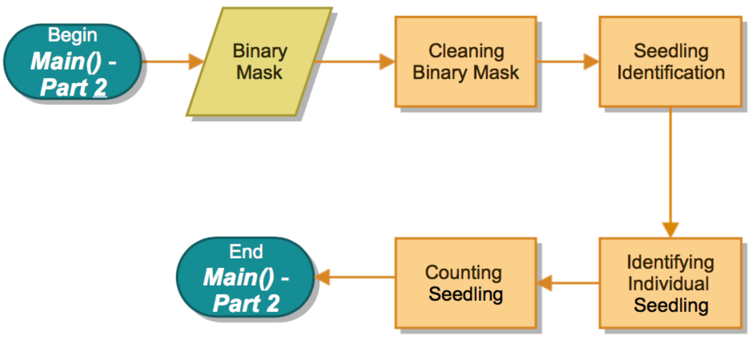

Figure 5: Representing the counting process in blocks of the main function part 2.

With *peak_local_max*, we expected to aggregate anything less than *minimum* E_D in one single peak but it was not what happened. Therefore, we had to develop our own procedure. We generated an array with the euclidean distance between every two prospective peaks and evaluated its corresponding histogram. Notice in Fig. 6 that there is a big gap between 13 and 24 pixels with no distance at all. Consequently, we chose 20 pixels for the minimum distance in such a way that peaks separated less than that were colapsed into just one peak, marking each of those with the help of Matplotlib [Hunter, 2007]. The result will be seen in Fig. 11 in Section 4.

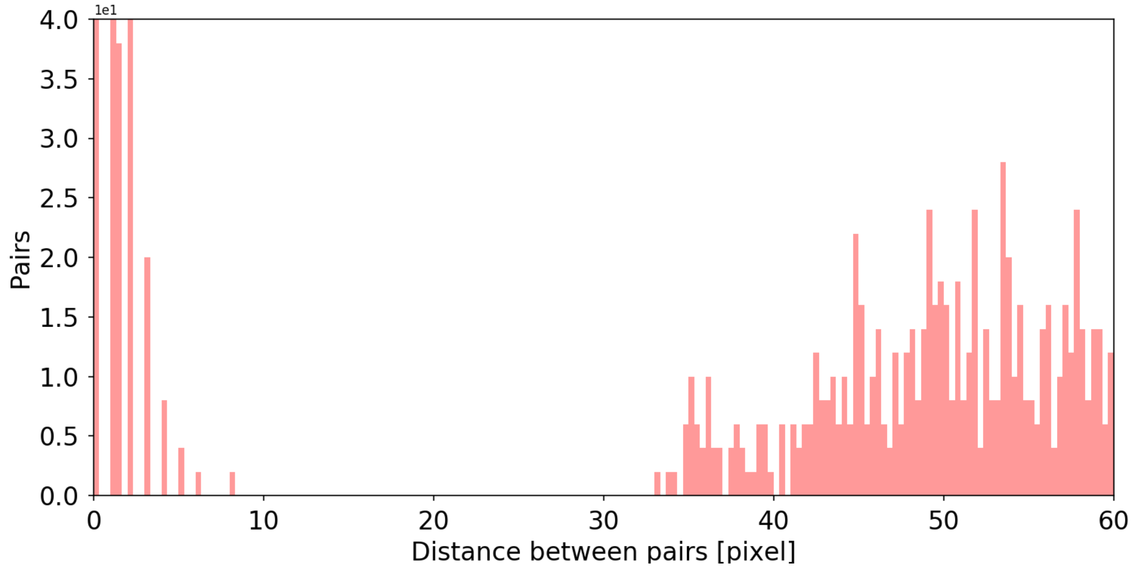

Figure 6: Euclidian distance between prospective peaks.

3.3 Statistical and scoring parameters

For convenience, the situation described in Section 2 was turned into a binary classification problem in the sense of one-against-all. Seedlings detected by the algorithm should be considered positive occurrences, whereas everything else is negative. Detected seedlings, which are real, are *true positives* (T_P). Detected seedling not real are *false positives* (F_P). Non-detected seedlings, which are real, are *false negatives* (F_N). The sum of all real seedlings within the image is the *ground truth* (G_T). The number of seedlings detected by the algorithm (V) is expressed in the Eq. 4.

$$V = T_P + F_P \tag{4}$$

Equation 5 is the ground truth.

$$G_T = T_P + F_N \tag{5}$$

Precision (P), Eq. 6, is defined as the number of true positives (T_P) over the number of true positives plus the number of false positives (F_P). Note that P is a measure of result relevancy, that is, from the detected seedlings, how many are *relevant* [Pedregosa et al., 2011].

$$P = \frac{T_P}{T_P + F_P} \tag{6}$$

Recall (R), Eq. 7, is defined as the number of true positives (T_P) over the number of true positives plus the number of false negatives (F_N). R is a measure of how many truly *relevant* results are returned [Pedregosa et al., 2011].

$$R = \frac{T_P}{T_P + F_N} \tag{7}$$

An ideal algorithm with high Precision and high Recall will return many seedlings, with almost all resulting true seedlings. These quantities are also related to the (F_1-$score$), which is defined in Eq. 8 as the harmonic mean of Precision and Recall [Pedregosa et al., 2011]. It is really common in real applications a compromise between P and R in the sense of an increase in Precision implies a decrease in R and vice-versa. Therefore, the harmonic mean of those values plays a key role in the evaluation of this compromise.

$$F_1 = 2\frac{P \times R}{P + R} \tag{8}$$

3.4 Hyper-parameter tuning and model validation

Going back to Eq. 1 and 2, in order to determine L^{upper} and L^{lower}, we need to choose a value for f, our only hyper-parameter. To investigate the f behavior, we selected a significant area from the whole plantation stand as our training data, shown in Fig. 1-(a), provided by Eldorado Brasil and segmented it into counting manageable 12 tiles in order to study the difference of reflectance for each segment. For each one of them and for f equals to 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, we obtained V, F_P, F_N, G_T and, from Eq. 5, T_P. These results were obtained with the same 35 hand-picked samples from representative parts of the training data.

In order to do a better assessment out of the results obtained for the range of f, we normalized F_{P_i} and F_{N_i} by G_{T_i}, for each tile i, as following:

$$\hat{F}_{P_i} = \frac{F_{P_i}}{G_{T_i}} \times 100\% \tag{9}$$

$$\hat{F}_{N_i} = \frac{F_{N_i}}{G_{T_i}} \times 100\% \tag{10}$$

Also we calculated the total values for F_P according to Eq. 11, and similarly defining $\hat{F}_N$, considering all tiles together as a big one.

$$\hat{F}_P = \frac{\sum_{i=0}^{n} F_{P_i}}{\sum_{i=0}^{n} G_{T_i}} \tag{11}$$

$$\hat{F}_N = \frac{\sum_{i=0}^{n} F_{N_i}}{\sum_{i=0}^{n} G_{T_i}} \tag{12}$$

With the help of the previous definitions, we drew the bar charts (a) to (f) presented in Fig. 7, in which for every f, we registered the values of $\hat{F}_{P_i}$ and $\hat{F}_{N_i}$, for each tile i.

Still in Fig. 7, it is possible to notice the importance of the hand-picked sample set to train the algorithm. Notably, the percentages of $\hat{F}_{P_i}$ and $\hat{F}_{N_i}$ change along the tiles, which is a sign of distortions and variations in reflectance and radiance of the soil and everything on it. Therefore, the importance of the sample representativeness is stressed.

Let's consider the behavior of $\hat{F}_P$ and $\hat{F}_N$ with f in Fig. 8. Notice that $\hat{F}_N$ does not change much for f larger than 6.0, whereas $\hat{F}_P$ increases significantly with f, as expected. On the one hand, it appears to be a kind of lower limit of about 3.5% of G_T for $\hat{F}_N$ in this model. On the other hand, from the joint behavior of $\hat{F}_P$ and $\hat{F}_N$, there seems to be an optimum value of f around 6.0%.

Figure 8 shows the $\hat{F}_P$ is more sensible to ascendant variances in f than $\hat{F}_N$. Between $f = 6.0$ and $f = 7.0$, it is possible to depict $\bar{\hat{F}}_N$ and $S_{\hat{F}_N}$ are practically constants, while $\bar{\hat{F}}_P$

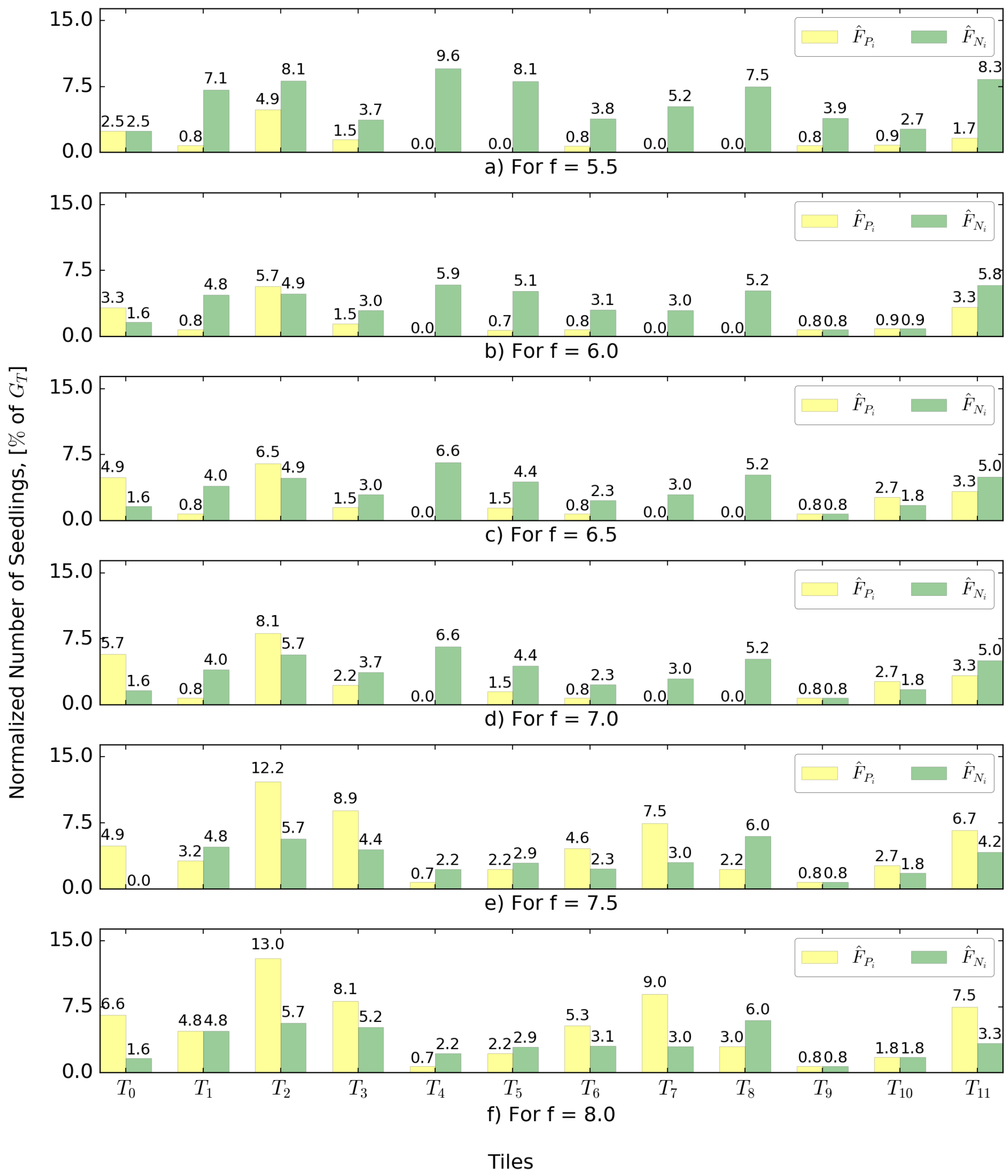

Figure 7: Values of $\hat{F}_{P_i}$ and $\hat{F}_{N_i}$ for each tile i.

and $S_{\hat{F}_P}$ are not. If f is too great, the code is prone to produce large values for F_P, whereas setting this hyper-parameter too small, we end up with a large F_N, what means according to Eq. 5, small values of T_P.

In order to summarize and better understand those charts, we calculated, based on Eq. 6 and 7 , the mean values of Precision ($\bar{P}$), for each value of f, according to Eq. 13, and similarly to Recall ($\bar{R}$).

$$\bar{P} = \frac{\sum_{i=0}^{11} T_{P_i}}{\sum_{i=0}^{11}(T_{P_i} + F_{P_i})} \tag{13}$$

$$\bar{R} = \frac{\sum_{i=0}^{11} T_{P_i}}{\sum_{i=0}^{11}(T_{P_i} + F_{N_i})} \tag{14}$$

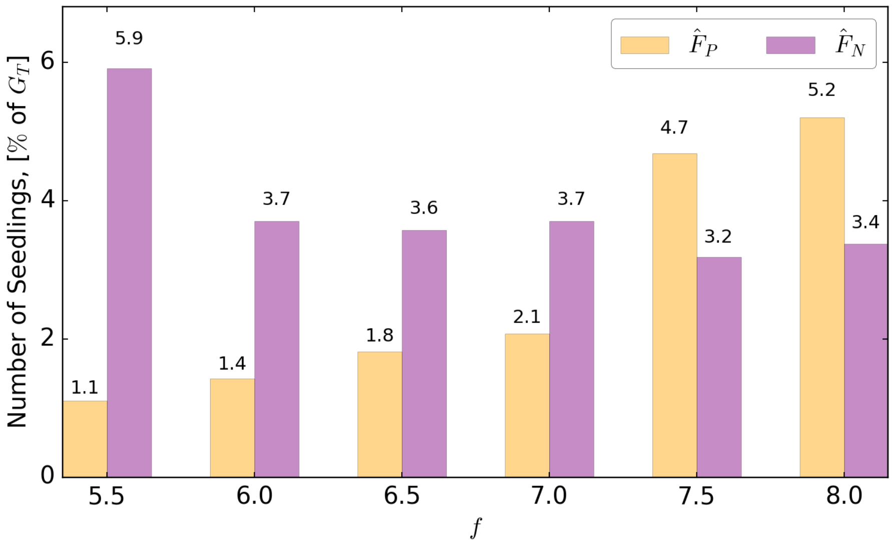

Figure 8: Values of $\hat{F}_P$ and $\hat{F}_N$ for each f.

Next, $\bar{P}$ and $\bar{R}$ were deployed into Eq. 8 to obtain $\bar{F}_1 - score$.

Notice that in Fig. 9 that $\bar{P}$ drops steadily, while $\bar{R}$ - after $f = 6.0$ stays mostly stagnant. That behavior is reflected in $\bar{F}_1 - score$. According to $\bar{F}_1 - score$ metrics, $f = 6.0$ is clearly the best value for this setting. Additionally, observe that from values of f after 7, $\bar{P}$ and $\bar{R}$ exchange places, exactly when $\bar{R} = \bar{P}$, and consequently same as $\bar{F}_1 - score$ as well.

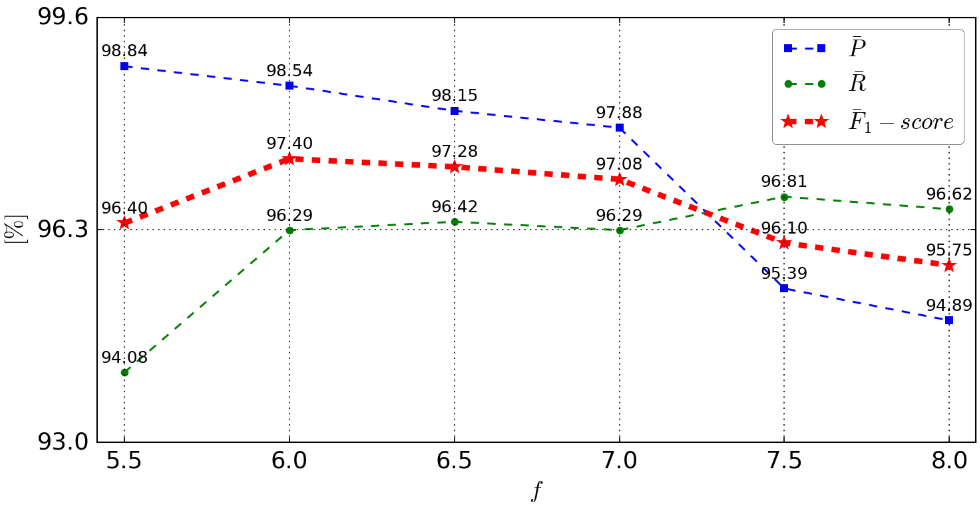

Figure 9: $\bar{P}$, $\bar{R}$, and $\bar{F}_1 - score$ as a function of f.

4 APPLICATION TO TESTING DATA SET

In order to test the model, we chose a completely new tile taken from some region of the mosaic, present in Fig. 10. This tile is not part of our training data and it was selected keeping a seedling hand-counting manageable size of approximately 0.2 hectares. We decided to use the same samples as for training data in order to investigate their representative power, and $f = 6.0$, which is the optimum value we have discovered, according to $\bar{F}_1 - score$ metrics.

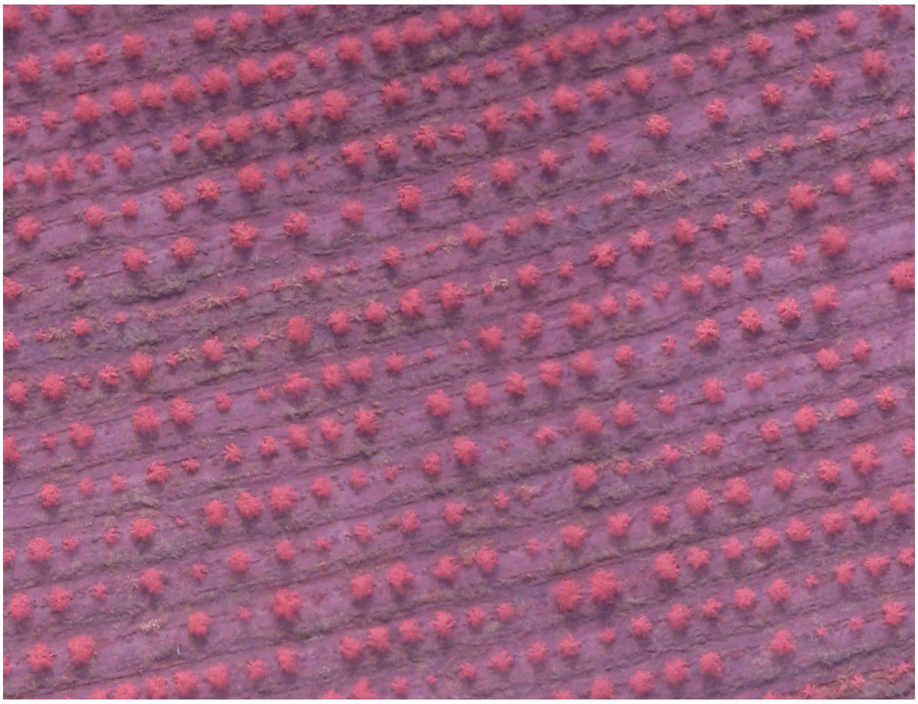

Figure 10: Tile for test represented in fake RGB color space.

Figure 11 depicts the algorithm output image. As described in Subsection 3.1, each red mark represents a unique seedling. Table 1 shows all the results evaluated by the algorithm. It took approximately 6 seconds wall clock time in a *conventional laptop*[2] to execute the whole process and print the data out, which is a considerable short time.

Table 1: Algorithm output for *tile* test

$[INFO]$	Start training the model with $f = 6$...
$[DONE]$	It took 0.00868 seconds to train the model
$[INFO]$	Upper limit is: $[175.8 \quad 169.48 \quad 295.74]$ Lower limit is: $[168.26 \quad 92.0 \quad 92.77]$
$[INFO]$	Start scanning the image...
$[DONE]$	It took 5.518 seconds to scan the model
$[RESULT]$	The number of trees scanned is: 314

Analysing the Fig. 11 it is possible to spot that the algorithm was able to capture most of the true seedlings. However, there are much more false positives and much less false negatives than expected considering the training tiles results.

The next step was to obtain $\hat{F}_{P_{T_{test}}}$, $\hat{F}_{N_{T_{test}}}$, and put those values in the bar chart of Fig. 7-(b). The result shows $\hat{F}_{P_{T_{test}}}$, $\hat{F}_{N_{T_{test}}}$ did not fit very well along with the training data since $\hat{F}_P$ is larger than $\hat{F}_N$, as it can be seen in Fig. 12. One possible reason for that behavior lies on the use of a sample set from training data, which is not representative of this new tile.

[2]MacBook Pro Intel I5, 8Gb of RAM

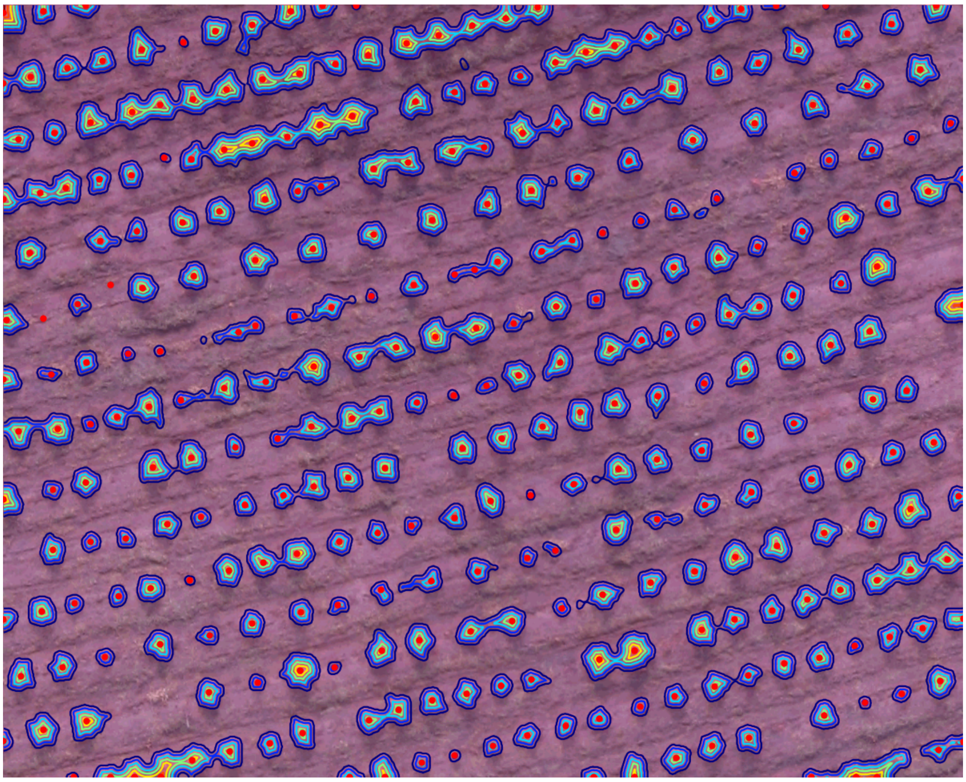

Figure 11: Output of the counting and identification analysis.

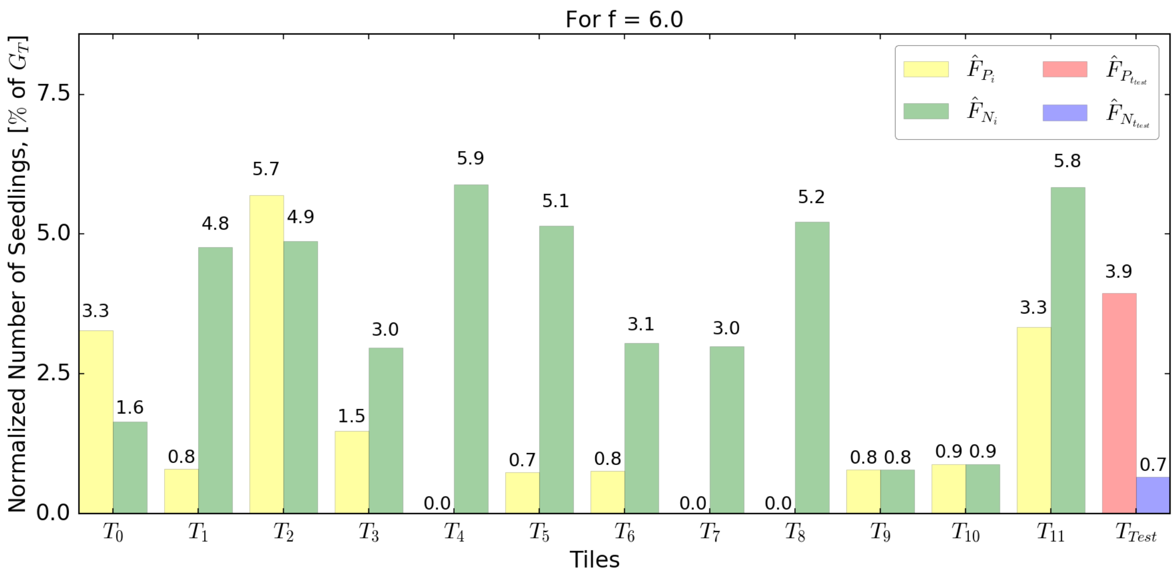

Figure 12: Values of $\hat{F}_{P_i}$ and $\hat{F}_{N_i}$ for each training tile i compared to tile test.

5 CONCLUSIONS AND FUTURE WORK

We summarize here some of our findings with respect to the present work. The key success of the algorithm is the choice of HSV, once it guarantees a nicely linear color variance. However, the distribution of the color intensity histogram has moderate discriminative power, as it can be seen in Fig. 1. Noise reduction in the binary mask is a critical factor and the combination of *opening*, *blurring* and *closing* was determined to be the best for this specific problem. The mosaic aerial image quality plays a fundamental role in the behavior of the algorithm in the sense of the degradation of the discriminative ability. The algorithm has f as its sole hyper-parameter, making it easier to be tuned than most of machine learning approaches [Pedregosa et al., 2011]. The representativity of the sample set is really important to the quality of the results.

Regarding the *training* data, we structured the solution to a binarized classification problem in the sense of one (seedlings) against all (everything else). In other words, it is impossible to obtain true negatives (T_n). Since the Precision ($\hat{P}$) and Recall ($\hat{R}$) are both values close to 1, the number of False Positives ($\hat{F}_P$) and False Negatives ($\hat{F}_N$) is very small in comparison with G_T, the Ground Truth. For the range of ascending f, $\hat{P}$ drops and $\hat{R}$, that is, $\hat{T}_P$, becomes stagnated at around 96$\%$. It seems that the model is incapable of lower this limit. $\hat{F}_1 - score$ played a key role in choosing an optimal value for f.

Regarding the *test* data, we already mentioned the inversion of $\hat{P}$ and $\hat{R}$, or $\hat{F}_P$ and $\hat{F}_N$. A larger $\hat{F}_P$ is either or a symptom of a too large value of f or misrepresentative sampling set or a combination of both effects. That remains to be tested.

The present work opens up a number of possibilities that can be explored. we addressed only solutions for the HSV color space. In the literature there are several indices that can be applied to enhance the discriminative power of the algorithm. The variability of the results must be evaluated by applying algorithm over several *training* data sets. Variability caused by the sampling process has to be assessed. Representative sampling seems to be a key step of the procedure. Finally, we are very interested on how the algorithm behaves on areas with competitive vegetation.

ACKNOWLEDGEMENTS

We acknowledge Eldorado Brasil, which provided the mosaic aerial image utilized here for evaluation and validation of the developed algorithm; the InnoVANT club[3], which contributed with intellectual and strategical insights; CNPq for the research sponsorship; and UNESP and ITA for the institutional supports.

REFERENCES

Dobson, S. (2013). Image slicer.

Haboudane, D., Miller, J. R., Pattey, E., Zarco-Tejada, P. J., and Strachan, I. B. (2004). Hyperspectral vegetation indices and novel algorithms for predicting green {LAI} of crop canopies: Modeling and validation in the context of precision agriculture. *Remote Sensing of Environment*, 90(3):337 – 352.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95.

Itseez (2015). Open source computer vision library.

Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: Open source scientific tools for Python. [Online; accessed 2017-02-09].

[3]innovant@gmail.com

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors (2014). scikit-image: image processing in Python. *PeerJ*, 2:e453.

Wegener, S., Schoenung, S., Totah, J., Sullivan, D., Frank, J., Enomoto, F., Frost, C., and Theodore, C. (2004). Uav autonomous operations for airborne science missions.