Mathematics and Computer Science
2019; 4(6): 112-129
http://www.sciencepublishinggroup.com/j/mcs
doi: 10.11648/j.mcs.20190406.13
ISSN: 2575-6036 (Print); ISSN: 2575-6028 (Online)

Determination of Forest Reserves Area Using Images Processed by Drones, Neural Networks and Monte Carlo Method

Paulo Marcelo Tasinaffo[1, *], Afonso Henriques Moreira Santos[4], Elias Cavalcante Junior[1], Carlos Henrique Quartucci Forster[1], Rafael Augusto Lopes Shigemura[1], Rafael Jacomel[2], Victor Ulisses Pugliese[1], Bruno Koshin Vazquez Iha[3], Adilson Marques da Cunha[1], Gildarcio Sousa Goncalves[1], Luiz Alberto Vieira Dias[1]

[1]Computer Science Department, Brazilian Aeronautics Institute of Technology (ITA), São Jose dos Campos, Brazil
[2]IBM Brazil, São Paulo, Brazil
[3]Mathematics and Statistics Institute, Sao Paulo University (USP), Sao Paulo, Brazil
[4]Electrical Engineering Institute, Federal University of Itajuba (UNIFEI), Itajuba, Brazil

Email address:

tasinaffo@ita.br (P. M. Tasinaffo), afonso@unifei.edu.br (A. H. M. Santos), eliasjuniorbiologo@gmail.com (E. C. Junior), forster@ita.br (C. H. Q. Forster), rafael@ita.br (R. A. L. Shigemura), rafaelmj@br.ibm.com (R. Jacomel), pugliese@ita.br (V. U. Pugliese), brunoiha@ime.usp.br (B. K. V. Iha), cunha@ita.br (A. M. da Cunha), gildarcio@ita.br (G. S. Goncalves)
[*]Corresponding author

To cite this article:

Paulo Marcelo Tasinaffo, Afonso Henriques Moreira Santos, Elias Cavalcante Junior, Carlos Henrique Quartucci Forster, Rafael Augusto Lopes Shigemura, Rafael Jacomel, Victor Ulisses Pugliese, Bruno Koshin Vazquez Iha, Adilson Marques da Cunha, Gildarcio Sousa Goncalves, Luiz Alberto Vieira Dias. Determination of Forest Reserves Area Using Images Processed by Drones, Neural Networks and Monte Carlo Method. *Mathematics and Computer Science.* Vol. 4, No. 6, 2019, pp. 112-129. doi: 10.11648/j.mcs.20190406.13

Received: June 26, 2019; **Accepted**: July 30, 2019; **Published**: December 24, 2019

Abstract: Land cover classification analysis from satellite imagery methods are important because they are the basis for characterizing surface conditions and evolution, supporting the management and optimization of land resources, evaluating global climate and environmental changes, and facilitating sustainable regional economic and social development. In order to address these necessities, artificial neural networks have been used extensively. In addition, other methods based on computer vision are very useful to solve this task. In this paper, the authors propose an approach based on Monte Carlo method and artificial neural networks in order to classify regions of small forest reserves from drones' images and calculate their respective areas. Next to the small forest reserve will be extended a standard rectangular tarpaulin of 250 square meters and based on this reference it will be possible to calculate the area of the forest reserve if the ground is relatively flat. The proposed approach will be compared with a method based on watershed algorithm. The automatic calculation of the forest area through images generated by drones has much practical application for environmental engineers, for example, for the calculation of environmental impact and determination of carbon loss if such forests are consequently deforested.

Keywords: Remote Sensing, Artificial Neural Networks, Monte Carlo Methods, Watershed Algorithm, Unmanned Aerial Vehicles (UAVs), Drone-based Imagery

1. Introduction

Land cover can be defined as a combination of natural and artificial surface structures occupying a given territory [1]. Satellite remote sensing has the characteristics of analyzing vast soil coverings, producing and analyzing large amount of information [2].

Methods to rapidly and accurately perform the extraction of land cover information using satellite remote sensing are important part of this research area [3, 4]. Such methods are important because they are the basis for characterizing surface conditions and evolution, base for supporting management and optimization of land resources, assessing global climate and environmental changes and facilitating sustainable regional economic and social development [5-7].

The work described in [8, 9] contributed immensely to the consolidation of methods mentioned below.

The Russian mathematician A. N. Kolmogorov solved in 1957 an important mathematical problem for the theory of estimation. He proved that it is possible to represent any function of n-dimensional space through a linear combination of limited and one-dimensional nonlinear functions. Hornik et al. [8] adapts Kolmogorov's work to the particular case of artificial neural networks with feedforward architecture, giving rise to the universal approximation of functions. Today the existence of this universal approximation of functions is well established [8, 9].

Most of methods mentioned below are based on deep neural networks, especially in convolutional neural networks. In [10], to address the need of large training dataset, complex network structure and training time, the authors proposed a framework that combines information from several sources and pooling layer, as well as a neural network for test data generation.

The authors of [11] compared one method based on a pixel-to-pixel to other method which used convolutional neural networks and a U-net model in the task to identify urban and forest areas. In [12] was proposed a classification method using photos of local social networks and convolutional neural network. In [13] the watershed algorithm was used together with the Markov Random Fields (MRV) method for satellite image segmentation.

Although neural networks have been extensively used, there is also available in literature some computer vision methods, such watershed algorithm, which is used for information extraction tasks. An approach using the watershed algorithm with markers and filters was used to delineate tree contours in [14].

In this work, it will be compared an approach based on multilayer perceptron (MLP) and Monte Carlo methods with another one based on the watershed algorithm, when used to calculate forest reserve areas from photos captured by drones.

The remainder of this paper is organized as follows: In the next section there will be described in more detail the techniques used in this work. There will have a section dedicated to discussions related to environmental impacts in 3. In section 4 there will elaborate about the proposed approach. There will describe about the results obtained in section 5. Finally, there will conclude the work in section 6, talk about future work in 7 and recognize some people and institutions in 8.

2. Theoretical Development

In this section there will be described the main techniques used in this work. We will start with a brief explanation about Artificial Neural Networks; then we will summarize some concepts about Monte Carlo methods, and will finalize the section describing variations of Watershed algorithm.

2.1. Artificial Neural Networks

According what was stated in [15], the nervous system is formed by an extremely complex set of cells, the neurons. Each neuron is composed of three sections, they are: body, dendrites and axon. The dendrites (input terminals) are responsible for the aggregation and transmission of stimuli to cell body. When a specific threshold of electrical potential is reached, new impulses are transmitted from the axons (output terminals) to the neighboring cells, through synapses [16]. Synapse is the region where two neurons come into contact and through which the nerve impulses are transmitted between them [15]. This process can be repeated in several layers of neurons. Thus, a network of neurons is formed. Each of these networks has a responsibility in the brain, namely sensitive, motor and interneuron.

The effort to understand the brain became easier after the pioneering publication of Ramón y Cajál (1911), presenting the idea of neurons as structural constituents of the brain [17]. However, the search for a computational model that simulates the functioning of brain cells occurred in 1943, with the work described by Minsky e Papert (Minsky e Papert, 1969) [16]. In the following years, the field of research in this area evolved significantly.

In 1958, Frank Rosenblatt proposed a network topology called Perceptron, which was able to classify linearly separable patterns, thus causing the volume of works related to neural network to increase substantially. However, in 1969, a very influential paper published by Minsky and Papert (Minsky and Papert, 1969) showed limitations of the known learning algorithm and provoked a new period of lethargy among scientific community [16-18].

During the 1980s, enthusiasm resurfaced thanks to the increase in available computing resources and important methodological advances [16], such as the publication of Hopfield's work (Hopfield, 1982) and the Back-propagation (BP) algorithm (Rumelhart et al. al., 1986) allowing the training of feedforward networks with multiple layers - commonly called multilayer perceptrons [18, 19].

Thus, Neural Networks or Artificial Neural Networks can be classified as a very simplified model that try to barely emulate the biological function of neural networks found in living organisms.

2.1.1. The Multiplayer Perceptron

The Multilayer Perceptron (MLP) architecture, published in 1982 are similar to Perceptron network, with supervised learning, but this is composed of more than one layer of neurons, interconnected by synaptic weights.

Thus, consider an input vector X in the initial layer of the network, the output values of that layer are connected to a new layer and this process repeats itself to the N layer, where N is the layer whose output values belong to the last layer from the Web.

The learning of this neural network was proposed with the publication of the backpropagation algorithm, for the training of the Feedforward networks. Thus, the training is done calculating whether network’s output is correct. And in negative cases, all the values of the synaptic weights are recalculated, from the last to the first layer, in order to reduce the error verified.

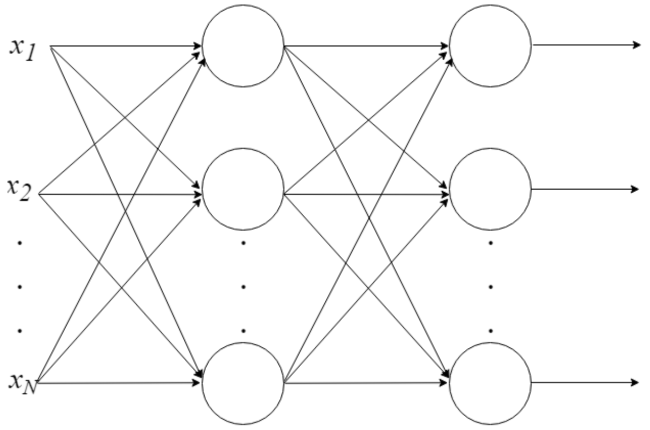


Figure 1. *Example of a Multilayer Perceptron network.*

2.1.2. Levenbergh-Marquart Method

The work to plan the design of an artificial neural network has seven primary steps: 1. Data collection, 2. Creation of the Neural Network, 3. Configuration of the Neural Network, 4. Initialization of weights and bias, 5. Training of the Neural Network, 6. Validation and Testing of the Neural Network and 7. Use of the Neural Network.

The MATLAB neural network toolbox uses the network object to store all the information that defines the neural network. A network object is a MATLAB object variable that is extensive enough to describe the entire architecture of an MLP (MultiLayer Perceptron) network.

After the neural network has been created, it needs to be configured and then trained. Configuration involves arranging the network in such a way that it is compatible with the problem that is to be solved and as defined through the numerical training data sample. After the neural network has been configured, the network tuning parameters (named weights and biases) need to be refined in such a way that network performance is optimized. The process of weight refinement is called network training. For the neural network to be configured and trained, it requires the use of the data sample. Given this small introduction with respect to the design and training process of a neural network, the following describe briefly the Levenberg-Marquardt algorithm, which was the algorithm effectively used in this article to generate all practical results.

This method is an optimization method, proposed by K. Levenberg [1944] and D. Marquardt [1963], and its algorithm is able to find the local minimum in a multivariate function, expressed by the sum of real non-linear squared functions [18, 19], and using the formula below:

$$\phi(x) = \frac{1}{2}\sum_{i=1}^{m} f_i^2\,(\mathrm{x}) = \frac{1}{2}\,||F(x)||^2 \tag{1}$$

Exactly like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed with the second-order training velocity approach, but without having to compute the Hessian matrix. When the performance function takes the form of the square sum (as is typical in MLP training), then the Hessian matrix can be approximated as:

$$H = J^T J \tag{2}$$

and the gradient can be computed as

$$g = J^T e \tag{3}$$

where J is the Jacobian matrix containing the first derivatives of the network error function with respect to weights and bias, and e is a vector of network errors. The Jacobian matrix can be computed using the standard backpropagation technique which is much less complex than directly computing the Hessian matrix. The Levenberg-Marquardt algorithm uses this approximation for the Hessian matrix in the following update:

$$x_{k+!} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{4}$$

When the scalar μ is zero, this is precisely Newton's method, using the approximation for the Hessian matrix. When μ is large, this becomes the descent of the gradient with a small step size. Although this method has converged rapidly in only a few iterations, each iteration takes a much longer time than that required, for example, by the simple backpropagation method. Also, for very large networks with hundreds of weights this method will require a lot of memory and a lot of processing time. Thus, this method seems to be the fastest method to train neural networks with moderate size MLP architecture.

2.2. Monte Carlo Method

The Monte Carlo method is a way of solving problems using random numbers. This method is widely used in computational model simulations. In this context, simulation is defined as a technique that emulates the operation of a real-world system as that system evolves over time. A suitable way of simulating the behavior of the type of variables to be analyzed is through the development of a simulation model using discrete event probability distributions known as the Monte Carlo method.

As Metropolis (1949) shows in [23], it was found to be possible to give up of the complex analytical mathematical models that modeled the interactions between nuclear particles in favor of modeling only the rules and statistics that govern each stage of the process. Starting from a uniform statistical distribution and then mapping it to distributions of interest to the given problem, it was possible to process multiple decision chains (simulations) and then extract relevant results with precision compatible with analytical methods. The Monte Carlo methods are able to establish precise geometric considerations of length and area of complex figures from purely probabilistic considerations. This is due the large random number generation capability inherent in computers.

A curious question is the origin of the name Monte Carlo. The method exploits statistical properties of random numbers to ensure that the correct result is computed in the same way as in a casino game to make sure that the house will always make a profit, and one of the most famous cities because of its casinos is Monte Carlo.

Generation of Random Numbers: Any computer simulation of a physical system involving randomness includes a method for generating random number sequences. For example, the simulation of queuing systems involves generation of interval between arrivals of people and time of service of each client. Random numbers must always satisfy the properties of the physical processes they are simulating and the numerical simulation in computers involves the generation of long sequences of random numbers, thus, it involves the generation of random variables with predefined distributions, for example, uniform and Gaussian distributions. It should be quite clear that in this article only uniform distribution was used.

A vast category of random number generators is based on simple algorithms that can be easily implemented on a computer. These algorithms can usually be represented (see [28]) by a tuple (S, f, μ, U, g), where:

S is a finite sets of states;

f is a function of S to S;

μ is a distribution of probability over S;

U is an output space, for example, for a uniform random generator, U is the interval $(0,1)$;

g is a function from S to U.

Thus, computationally, a random number generator must have the following structure (Kroese et al, 2011):

Algorithm 1.1 (Generic random Generaton)

```
1. Initialization: generate a seed $S_0$ of probability distribution $\mu$ over $S$. Make $t=1$.
2. Transition: Make $S_t = f(S_{t-1})$.
3. Output: Make $U_t = g(S_t)$.
4. Repeat: Make $t = t+1$ and back to step 2.
```

The algorithm produces the sequence $U_1, U_2, U_3, \ldots$ of pseudo-random numbers. They can be referenced only by random numbers. One quite important definition is the length-period of a random number generator. Length period is, by definition, the smallest number of steps an algorithm takes before entering an already visited state again. The concept of length period is important to realize if the random values generated are not flawed.

A known example similar to the application explained above is the estimation of the value of the number π. Below is showed an example of the Figure 2 of a circle with radius R:

In order to simplify the calculations, consider only the section marked by the yellow square with R side in the lower right corner of the image. In this way the area of the circle within this square is:

$$area_{circle} = 2\pi R^2 \tag{5}$$

and the area of the square is:

$$area_{square} = R^2 \tag{6}$$

The fraction of those two areas is:

$$\frac{area_{circle}}{area_{square}} = \pi \tag{7}$$

Using Monte Carlo, is obtained the following image:

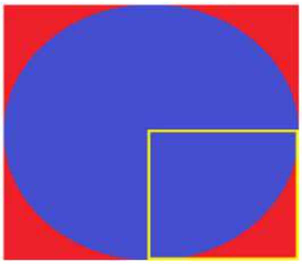


Figure 2. Image of a circle with a section bounded by a yellow square used to obtain the estimate of π.

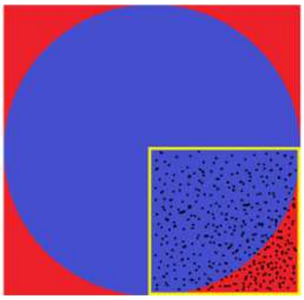


Figure 3. Use of Monte Carlo in the area bounded by the yellow square. The black dots represent the samples generated with Monte Carlo.

Continuing the generation of random points to the total of 18500 points within the square were, there was obtained 14597 points inside the circle's area. Using the equation X permits to verify that the estimate of the value of π to 18500 points is approximately 3.15611, very close to the expected value of 3.14159.

The determination of the number of sufficient random samples for the estimation of a measure is subjective and inextricably related to the tolerated uncertainty. But to obtain a more accurate value a viable approach is the generation of several results using the Monte Carlo method and with these results calculate the mean and variance of the same. In this approach the Central Limit Theorem states that the results tend to follow a normal distribution and that the mean of the results approaches the actual value of the measure [32].

Computationally these samples are generated using pseudorandom numbers generators, which are actually deterministic generators, but still generate sequences of numbers very close to real-time sequences being sufficient for use in the Monte Carlo method.

2.3. Image Segmentation Using Watershed Algorithm

This subsection elaborates about the algorithm of watershed and its application for image segmentation. The idea of the watershed technique is relatively old and was introduced in 1979 [33]. The basic idea was inspired in geography, more specifically, in the idea of geographic basins.

In geography, watershed or geographic basin is the extension or flowing surface of a central river and its tributaries. They are located in areas of higher altitudes of the relief by water partitions, in which rainwater is either drained superficially generating rivers and streams, or infiltrate in the soil for the formation of springs and the water table [34].

The levels of a geographic basin can be segmented as boundary topographic maps as the one in Figure 4 extracted from [35]:

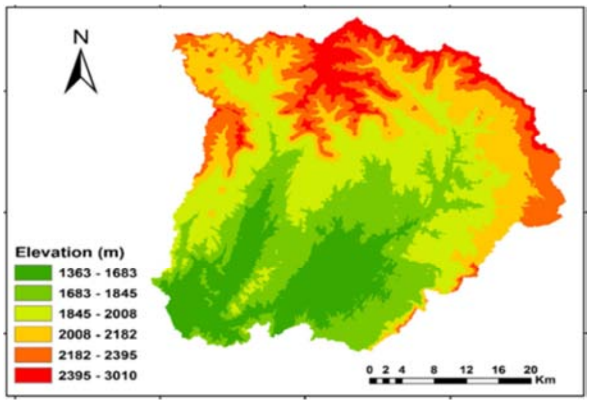


Figure 4. *Example of topographic map.*

In a simplified way, it can be said that the watershed algorithm works by segmenting an image using the assumption that higher brightness means greater heights and minor brightness mean lower heights as in a process of performing the topography of a geographic basin.

There are variations of this algorithm. In [36] the author mentions four variations of it: Marker-based Watershed Segmentation, Adaptive Marker-based segmentation, K-means Clustering, and Enhanced Region-based Segmentation.

3. Environmental Impact

One main goal of proposed approach is in helping to assess long-term environmental impacts, more specifically, the effects of deforestation. As the algorithms aid the mapping of forested areas, the gathered information permits automated and manual analyses about forest covering profile along time.

Ecosystems are dynamic and complex systems in which their diverse elements (fauna, flora, microbes, etc.) influence each other in a delicate balance. As those elements are not isolated systems, they also affect the surroundings. Therefore, changes in ecosystems can generate direct and indirect consequences not only to the ecosystem itself, but also to nearby regions.

One of the changes that generate the largest changes in the regions where it occurs is deforestation [37]. It is possible to mention some consequences of deforestation such as: alteration of biodiversity, increased soil erosion, less rainfall infiltration by plants; decrease in rainfall, increase in regional temperatures; greater possibility of landslides.

Deforestation reduces the hydrological services provided by trees, which are fundamental.

In Brazil, part of the water vapor emanating from the forests is transported by wind to the Central-South regions, where most of the country's agricultural activity is located.

The value of Brazil's annual crop is about US $ 65 billion - or about R$ 120 billion in 2009 values. If even a small fraction of this amount depends on rainfall originating from the water vapor of the Amazonas forest, the lack of rain would bring considerable losses for the country.

When rainfall reduction is added to the natural variability that characterizes rainfall in the region, the resulting drought can have a major environmental impact.

In Brazil there is a huge concern for address the problem of environmental impact since the creation of the 1934 Forest Code [38], its updates [39] and the Brazilian Constitution of 1988 [40].

4. Methodology

In this section there will be presented the proposed approach and a variation of the watershed algorithm whose results will be used to compare each other.

4.1. Proposed Algorithm: Hybrid Algorithm Using Neural Network Combined with the Monte Carlo Method

The proposed algorithm is very simple. It is based on the calculation of the value of the constant π and as explained in section 2.2. The major difficulty is to determine an algorithm that says if a random point is inside or outside an irregular contour. This is possible through the use of an artificial neural network. In our case, a neural network with MLP architecture will be used. The proposed algorithm has the following functional steps:

A picture of the forest is taken (this can be easily obtained through a drone) with the standardized tarpaulin whose area is 250 m^2 (Figure 5).

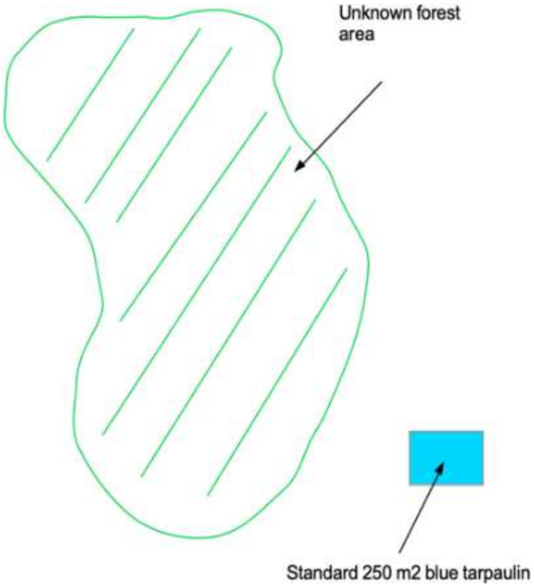


Figure 5. *Picture of the forest and blue tarpaulin.*

The contour points of the forest are selected using the mouse on the standard x and y coordinate axis ranging between 0 and 1 (Figure 6).

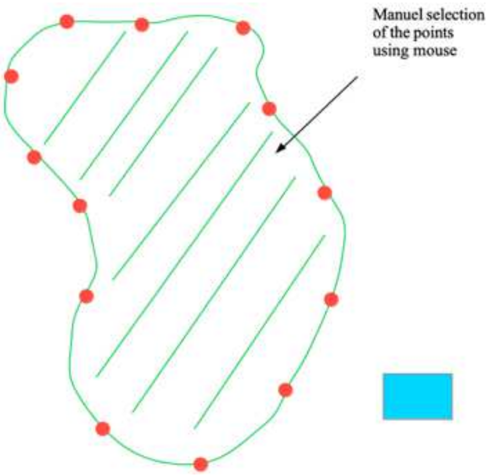


Figure 6. *Manual Selection of the forest border.*

The neural network is trained to decide if any randomly generated point, generated on the photograph taken by the Drone, is inside or outside the forest (Figure 7).

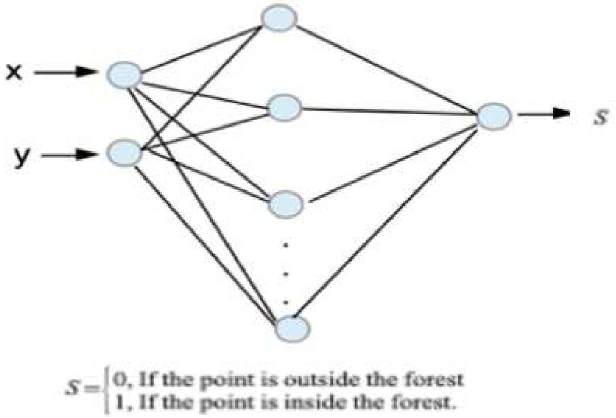

x
y
S
S = 0, If the point is outside the forest
1, If the point is inside the forest.

Figure 7. *Neural Network used to determine if a point is inside or outside the forest.*

One million random points (or more) were generated over the photograph of the forest. The total number n_1 of points within the forest is counted, and then the total number n_2 of points within the standardized canvas is counted and the area is calculated using the rule described in Figure 8. To do this, simply divide n_1 by n_2 and multiply this value by the area of the canvas.

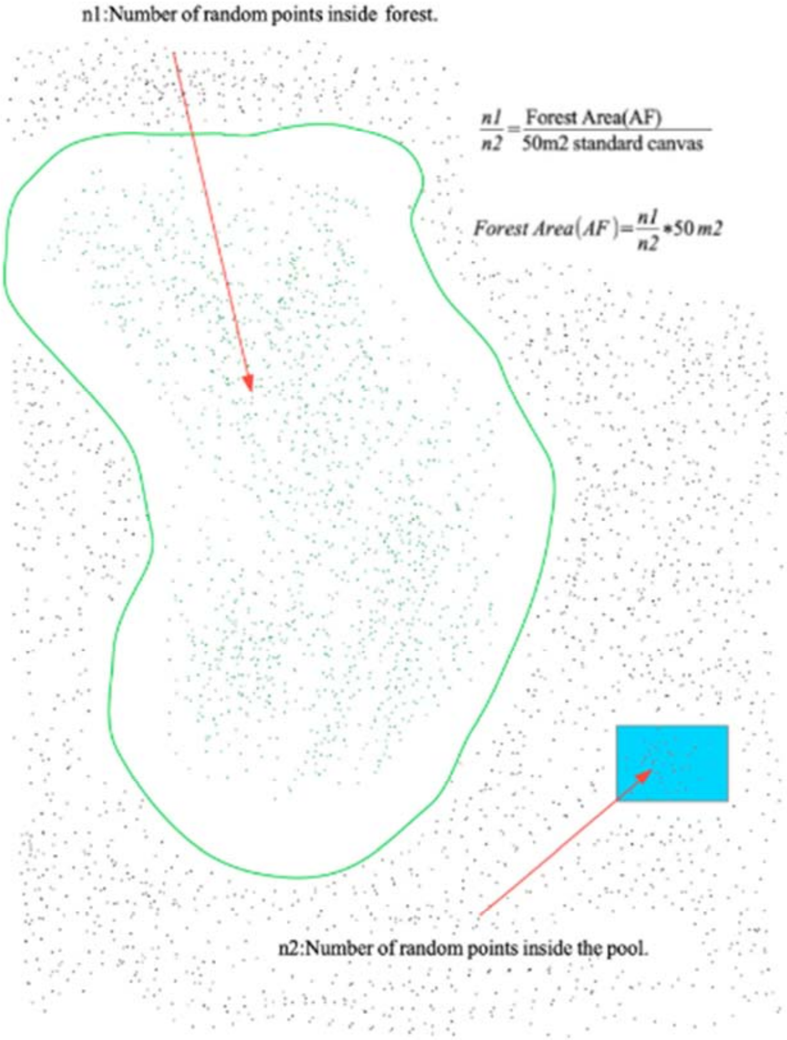

n1:Number of random points inside forest.
n1/n2 = Forest Area(AF) / 50m2 standard canvas
Forest Area(AF) = n1/n2 *50 m2
n2:Number of random points inside the pool.

Figure 8. *Billions of random points and the rule used to calculate forest area.*

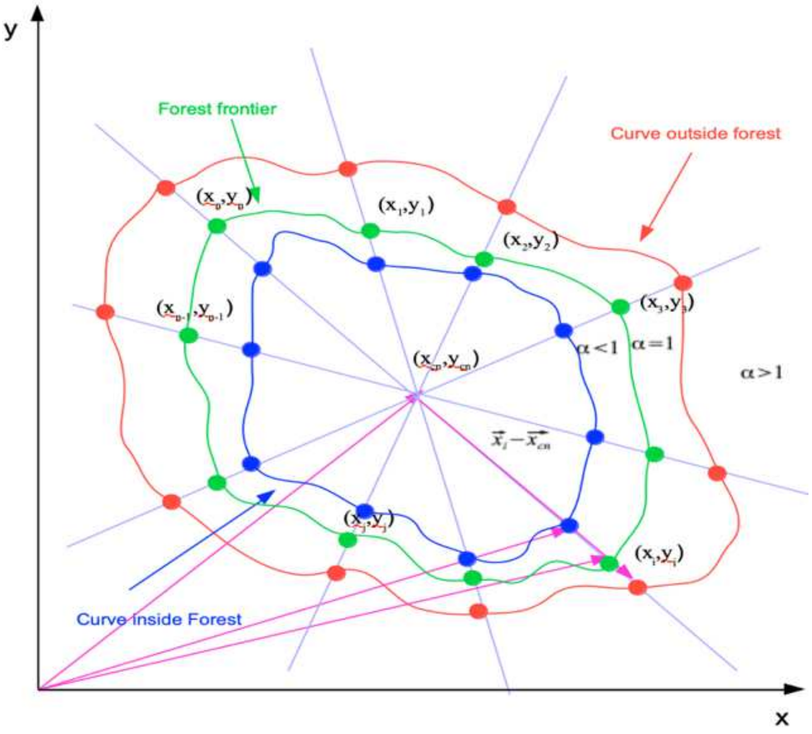


Figure 9. *Mathematical details of the proposed algorithm.*

In Figure 9 and formulas (8) and (9) below there is explained the mathematical detail of step 3 of the proposed algorithm:

In Figure 9 the green curve represents the real contour (partially convex). Some significant points of this outline are selected using the mouse. After this, the center of mass of these points is determined by equation (8).

$$\vec{X}_{cm} = \frac{X_1+X_2+\cdots+X_n}{n}, \vec{Y}_{cm} = \frac{Y_1+Y_2+\cdots+Y_n}{n} \quad (8)$$

Then reduced and enlarged copies of the green outline are performed by the following equation:

$$\vec{X}_i^{\alpha} = (X_i^{\alpha}, Y_i^{\alpha}) = \alpha(\vec{X}_i - \vec{X}_{cm}) + \vec{X}_{cm}$$

$$= \alpha\vec{X}_i + (1-\alpha)\overrightarrow{X}_{cm} \quad (9)$$

$$for\ \alpha = 0, \Delta\alpha, 2\Delta\alpha, \ldots, k\Delta\alpha\ e\ i = 1, 2, \ldots, n-1, n$$

For the extended points alpha values greater than 1 will be chosen. For the reduced points, alpha values smaller than one will be used. The alpha values will be the output of the neural network. The two inputs of the neural network will simply be the coordinates of the schematic points in Figure 9. Details of the alpha values can be seen as follows:

$$\alpha = 1\ (training\ point\ in\ the\ forest\ frontier)$$

$$0 < \alpha < 1\ (Training\ point\ inside\ the\ forest)$$

$$\alpha > 1\ (Training\ point\ outside\ the\ forest)$$

Note:

$$\vec{X}_{cm}\ is\ in\ \alpha = 0, consequently\ \vec{X}_{cm}$$

is the minimum point of the valley with the shape of the forest map (see Figure 10)

4.1.2. Geometric Interpretation of the Neural Network

The function is a two-dimensional surface (independent variables $X_i^{\alpha}\ and\ Y_i^{\alpha}$ in the form of a valley whose contour is exactly in the map format of the forest reserve and $\vec{X}_{cm}$ is the minimum point of the valley where $\alpha = 0$. Since the neural network is a universal approximation of functions, $\hat{f}_{nn}[\ldots]$, it can learn any kind of contour (see Figure 10), including the irregular contour of forest reserves.

$$\alpha = \hat{f}_{nn}(X_i^{\alpha}, Y_i^{\alpha}, \widehat{\mathcal{W}}) \quad (10)$$

Input of neural network: $X_i^{\alpha}\ and\ Y_i^{\alpha}$
Output of neural network: α
Function weights: $\widehat{\mathcal{W}}$
Nonlinear interpolation function: $\hat{f}_{nn}[\ldots]$

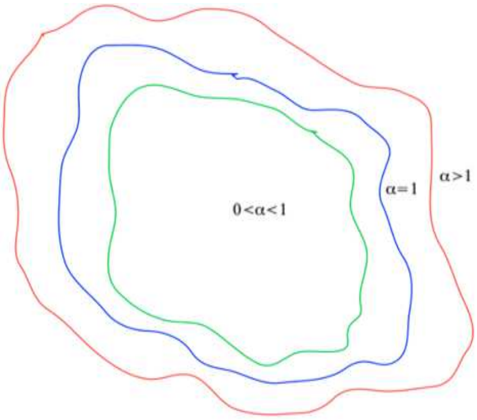


Figure 10. *Level contours in the forest reserve format.*

4.1.3. Limitations of Proposed Approach

The proposed method has the following limitations:

i. Works only with convex shapes. Non-convex shapes will not be allowed. When a shape is nonconvex, there is a good probability of dividing this undesirable shape into $\mathcal{M}$ convex regions;

ii. Applicable only to non-large forest reserves. The reason for this is that the drone cannot fly at high altitudes and also the rectangle canvas area cannot be negligible in relation to the total forest area;

iii. The forest reserve should be approximately flat, for example, the forest reserve cannot be located in hills, because the photo of the drone only takes the horizontal projection of the forest, that is, the photo of the drone does not take the view in true greatness of forests on sloping terrain.

In order to explain why the proposed methodology would not work in non-convex regions it is possible to use Figure 11:

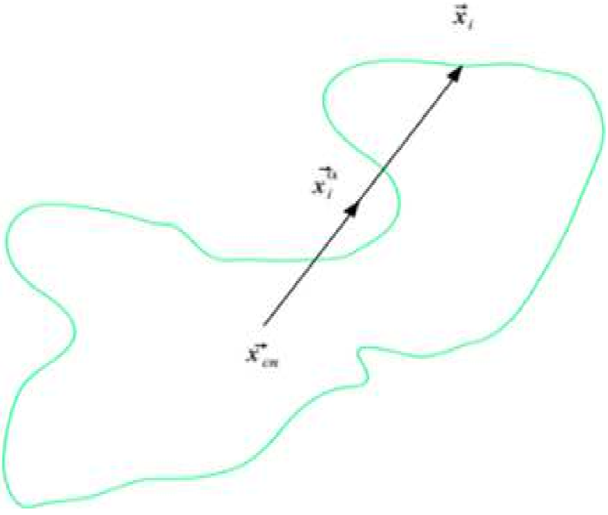


Figure 11. *Example of situation where proposed approach would not work.*

In Figure 11 the external point of the forest would be classified as internal causing an error in the classification of the points, so our method would work with some margin of error in non-convex surfaces.

4.2. Forest Reserve Area Calculation Using a Watershed Algorithm Variation

In this subsection there will be described a variation of the watershed algorithm used at this work in order to compare it with the proposed approach. The algorithm was chosen because it is widely referenced in literature for image segmentation. The variation described was based on [4] and basically works as follows:

1. First the image was segmented using the watershed algorithm:
   a. A few points were scored in the areas of interest (in our case of the urban, forest and tarpaulin area) for the watershed algorithm understand where to segment the image.
   b. The algorithm segmented the image by generating a new image, segmenting the areas of interest and coloring them with different colors.
2. There was used the pixels identified as forest area (PF) and the tarpaulin (PP) to calculate the area of the forest (FA) using the following rule (It is known know that the tarpaulin area is 250 m^2):

$$\frac{PF}{PP} = \frac{FA}{250\ m^2} \tag{11}$$

The image processing used here was a drone photograph taken by a drone in a condominium in the city of Ribeirão Preto and located in Brazil (see Figure 12).

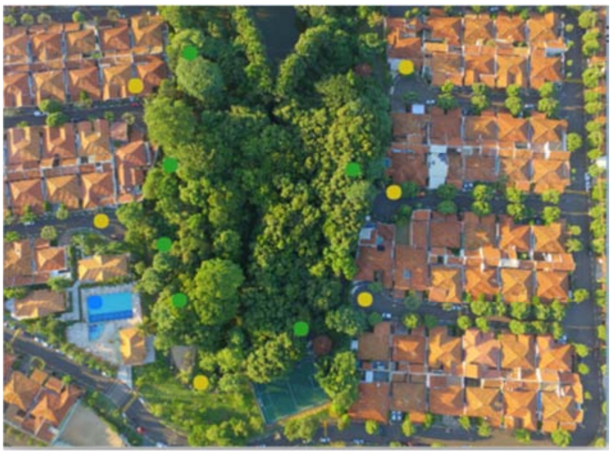


Figure 12. *Figure marked with six points in urban and forest area and one point in pool area.*

The images in Figure 12 shows examples of the segmented areas using six, twelve and sixteen points marking the forest area and the non-forest area. One point was used to mark the pool area. Figure 12 shows the forest and urban areas marked with six points each. The tarpaulin area was marked with one point. Figure 13 shows the segmentation generated:

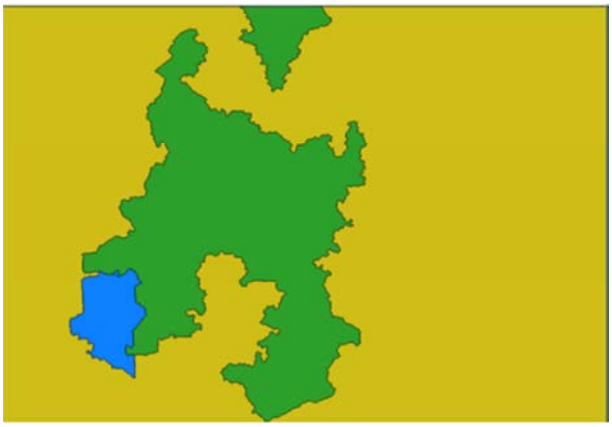


Figure 13. *Segmentation generated using six points in urban and forest area and one point in pool area.*

Figure 14 shows the forest and urban areas marked with twelve points each. The pool area was marked with one point. Figure 15 shows the segmentation generated:

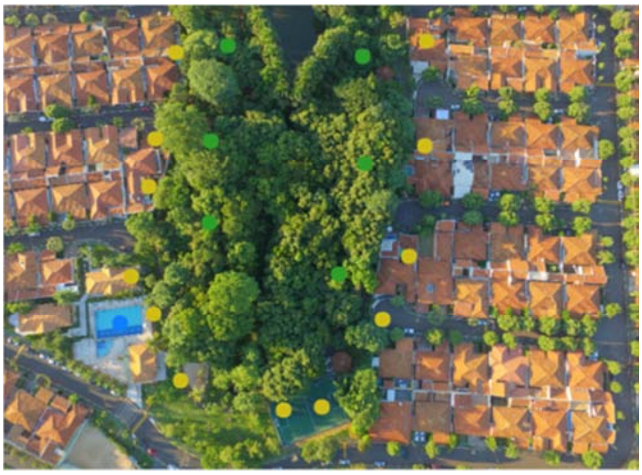


Figure 14. *Figure marked with twelve points in urban and forest area and one point in pool area.*

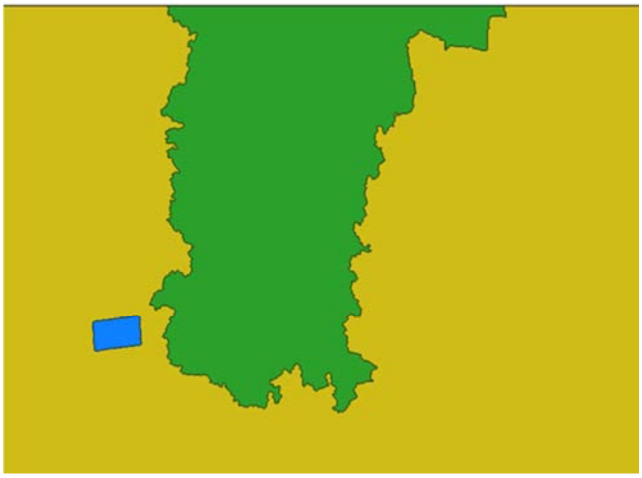


Figure 15. *Segmentation generated using twelve points in urban and forest area and one point in pool area.*

Figure 16 shows the forest and urban areas marked with sixteen points each. The pool area was marked with one point. Figure 17 shows the segmentation generated.

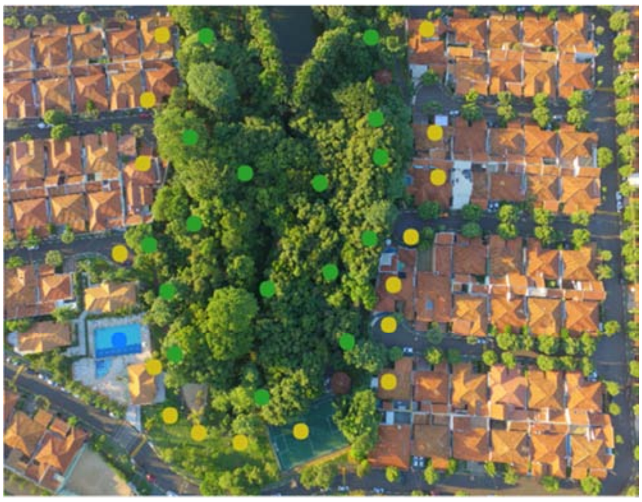


Figure 16. *Figure marked with sixteen points in urban and forest area and one point in pool area.*

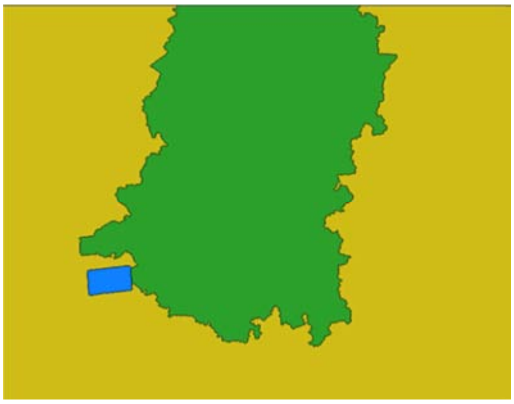


Figure 17. *Segmentation generated using sixteen points in urban and forest area and one point in pool area.*

Figure 18 shows the overlap of the original image with our best segmentation result (using 16 markers):

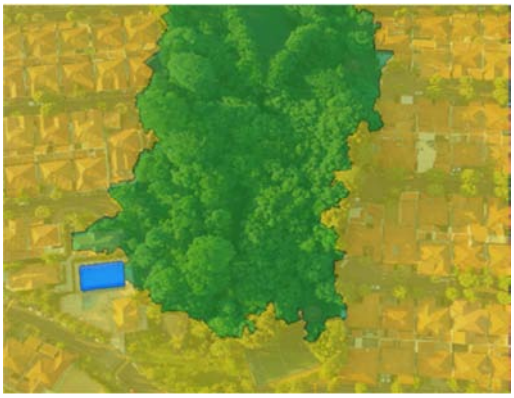


Figure 18. *Overlapping the best result in the original image.*

4.3. Forest Reserve Area Calculation Using the Proposed Approach

In this subsection there will be described the forest reserve calculation process using Artificial Neural Networks and Monte Carlo method, rigorously based on the algorithm proposed in section 4.1. The following images reflects the step-by-step results and basically works as follows:

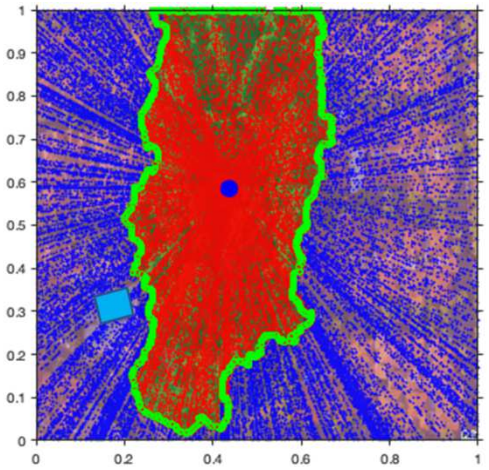


Figure 19. *Geometrically classified points provided for the training of the Neural Network on the map of Forest Reserve.*

In Figure 19 is possible to see, in green, the 1042 manually selected points representing the forest reserve boundaries. The tarpaulin is also visible (as the blue rectangle) and is quite important given is the only known area, 250 square meters.

The centered blue circle shows the calculated centroid, or mass center of the forest reserve, using the equation (8). The very importance of this point is to serve as referential for the algorithm determine if a randomly generated point is inside or outside the forest reserve, based on equation (9).

That said, the software generated and plotted 100.489 points, classifying each of them as inside (red) or outside (blue) based on its distance from centroid and its calculated alpha (as detailed in section 4.1).

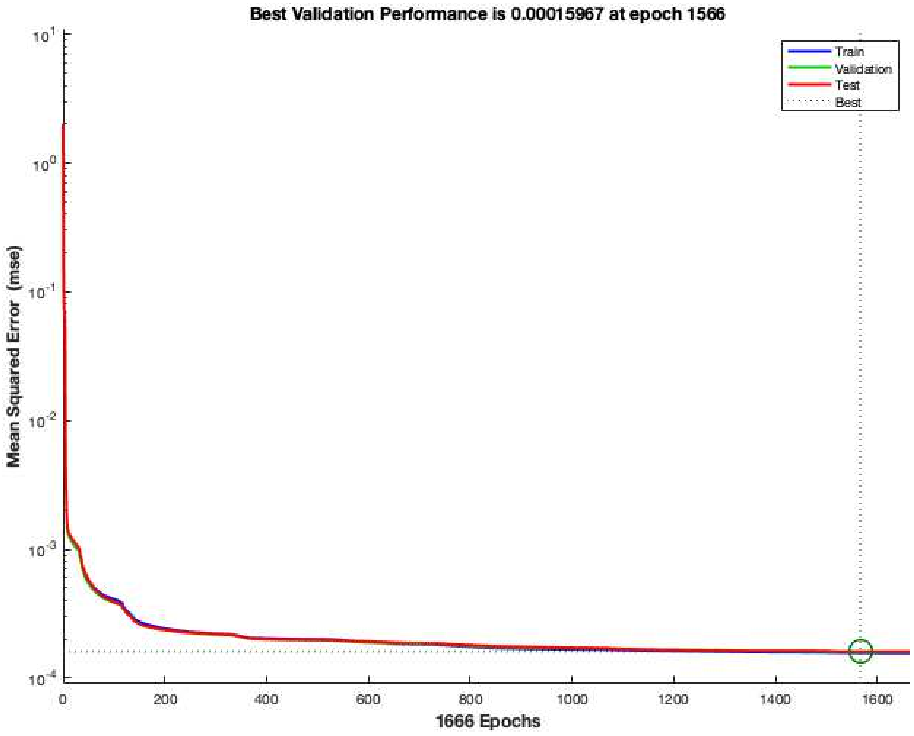


Figure 20. *Neural Network training performance on the map of Forest Reserve.*

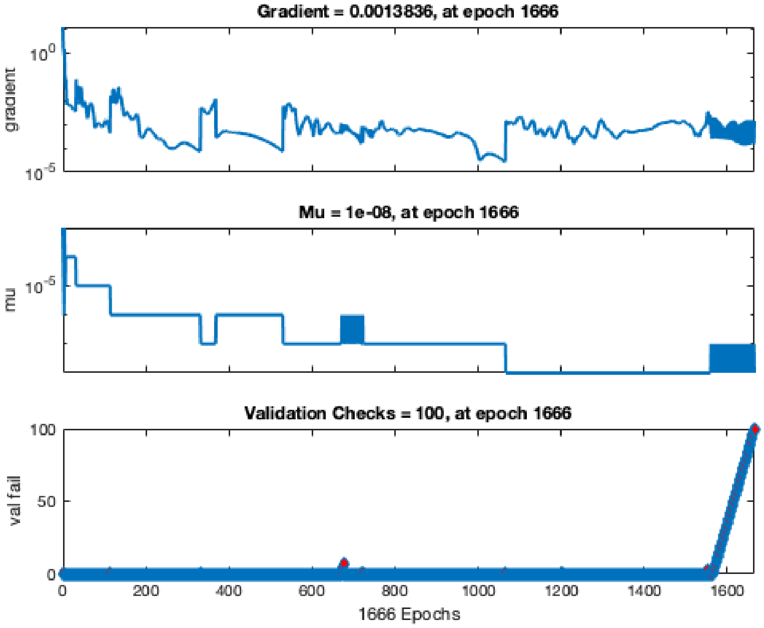


Figure 21. *Neural Network training states on the map of Forest Reserve.*

Right after generated, the 100,459 points are fed into artificial neural network, so that it can learn how to do it on its own for whatever point generated hereafter. Figure 20 shows the performance history and it is interesting to note how fast the mean squared error reduces, thanks to Levenberg-Marquart algorithm.

On figure 21 is possible to see the logging of stopping conditions during training. It took about 1580 epochs to have the neural network optimally trained. Below, in Figure 22, the reached statistical correlation, the R index, was quite high: .9996 for all sets, training, validation and test.

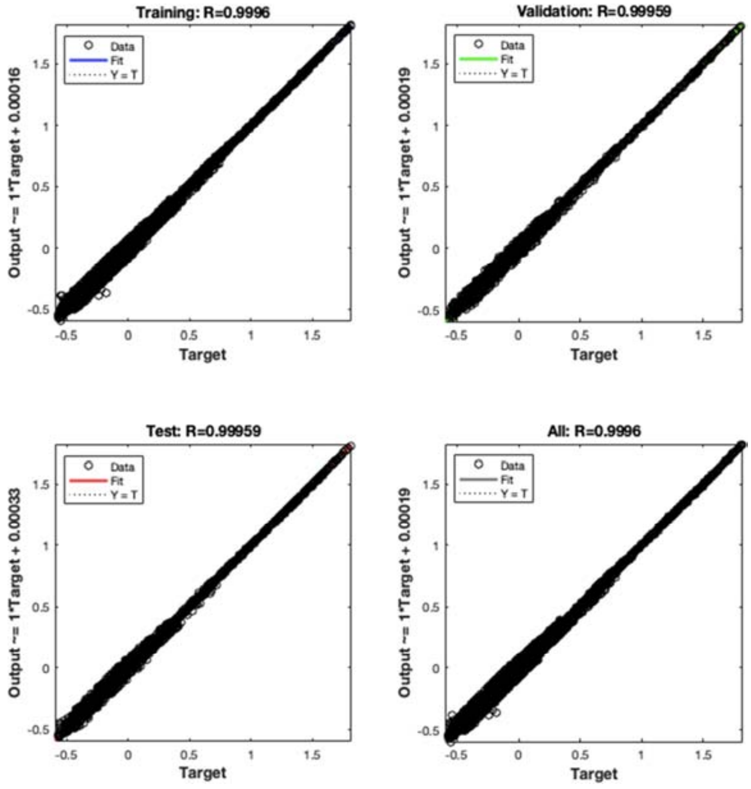


Figure 22. *Statistical regression on the training / test / validation and global Neural Network data on the map of Forest Reserve.*

As result, the overall learning performed well enough to expect good generalization. After the learning process, there were randomly generated another dataset of 100,459 points, this time not classified, and fed again to network assess its learning. Below, in figures 23 and 24 the neural network learning prism demonstrates the generalization for 5 levels of forest contour in 2D and 3D projections.

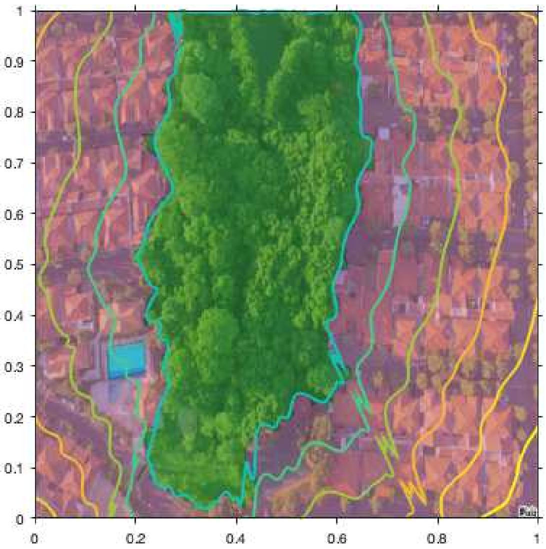


Figure 23. *2D Neural Network Learning Prism on the forest reserve in Ribeirão Preto / SP – Brazil.*

Specially for 3D projection, showed by figure 26, is possible to see that the artificial neural network learned not only to amplify the forest contour, but also to reduce it proportionally, exactly as predicted on figure 9.

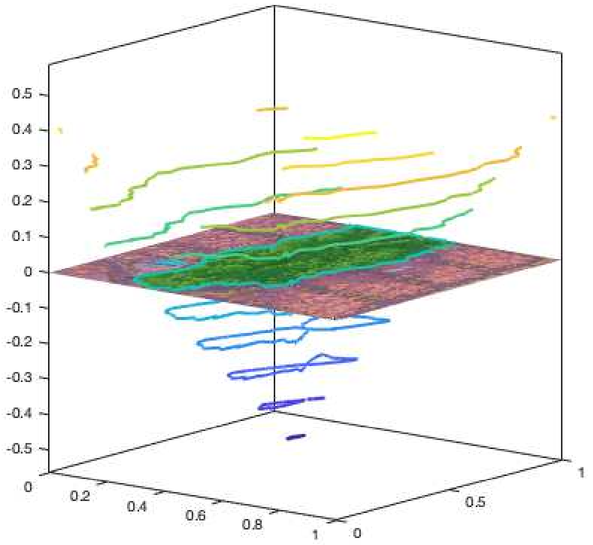


Figure 24. *3D Prism of Neural Network Learning in forest reserve in Ribeirão Preto / SP – Brazil.*

Finally, figure 25 shows a mesh grid layering the manually selected contour (green), the points classified as inside forest (red) and the points classified as outside forest (blue). Even considering the complexity of underlying function, is possible to see that the network achieved considerable precision, with very few visible blue areas representing misclassification.

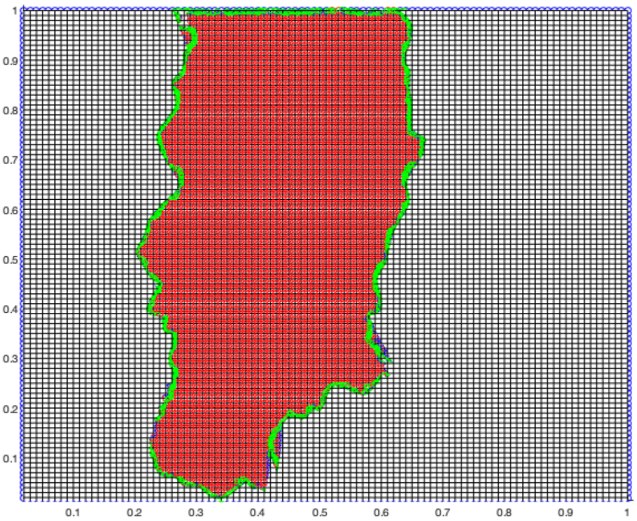


Figure 25. Representation of Learning of the Neural Network over the forest reserve.

5. Discussion of Results

In this section there will be discussed the numerical results or proposed approach. First, we will use the watershed methodology and the proposed method in order to calculate the forest area. The watershed method was chosen because it has already been widely used in the literature with good results. After this first analysis we will use the proposed approach to calculate the area of France and Brazil.

5.1. Proposed Methodology and Watershed Algorithm

In this section is showed the Watershed algorithm versus the proposed approach to calculate the area of the forest and the results are compared. In Table 1 we have the forest area calculated using the proposed approach and using the Watershed algorithm.

Table 1. Forest area calculation using watershed algorithm and proposed methodology.

Forest Area Calculation	
Forest Ares Using Watershed algorithm (m^2)	17.187, 22 m^2
Forest area using proposed methodology (m^2)	17.030, 38 m^2

It is worth mentioning that in order to calculate the area of the forest using watershed algorithm was used the formula 3.4 with PF = 307445, PP = 4472. Is possible to notice that the convergence between the methods is greater than 99% for the estimated area values, which can be considered satisfactory when it comes to areas that cannot be estimated by direct means.

5.2. Proposed Methodology Used in the Calculation of the France and Brazil Areas

Experimentally, 100.489 points were generated for Neural Network training and the same number of points for Neural Network Test, with the objective of estimating the area of 3 different places: for original forest reserve (unknown area), for Brazil and for France, each of which is shown below.

The first step in the area calculation is to generate the training patterns for each map, from the geometric rule described in equations (8) and (9). Starting at the manually selected points in the boundaries, the algorithm finds the centroid of the maps and thus determines whether a point is inside or outside the selected area.

Right after, the training patterns were provided to Artificial Neural Network responsible for learning how to classify each point according to the coordinates and geometry of the shape in question. In the images below is possible to see the best validation performances for France and Brazil, respectively.

The difference in performance between the maps is remarkable, with RNA trained for France reaching 2 orders of magnitude more precision than the RNA trained for Brazil. In Figure 28 is possible to see that for the case of France 10E-5 was reached while in the case of Brazil (Figure 29) 10E-3 was reached.

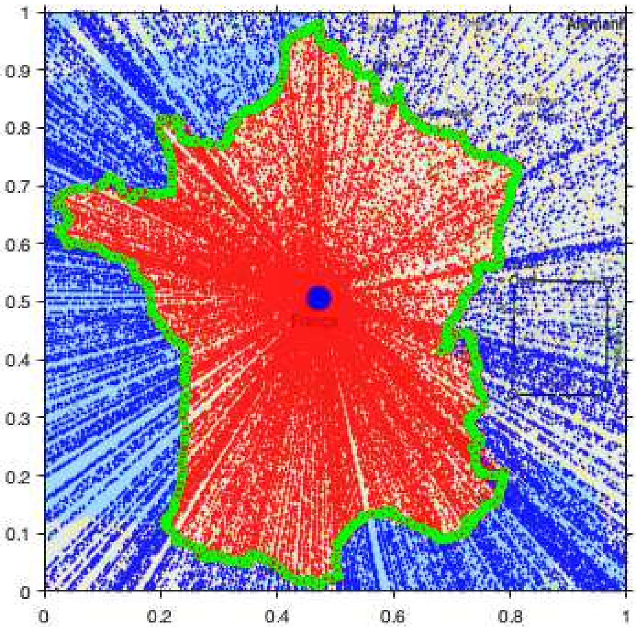


Figure 26. Geometrically classified points provided for the training of the Neural Network on the map of France.

The main hypothesis for this difference is the fact that the Brazilian map contains non-convex regions, causing a reduction of efficiency in the generation of training standards

Figures 30 and 31 show very similar behavior among RNAs in terms of training states. Interesting fact is the similarity amount of times required to stabilize learning: at least 450.

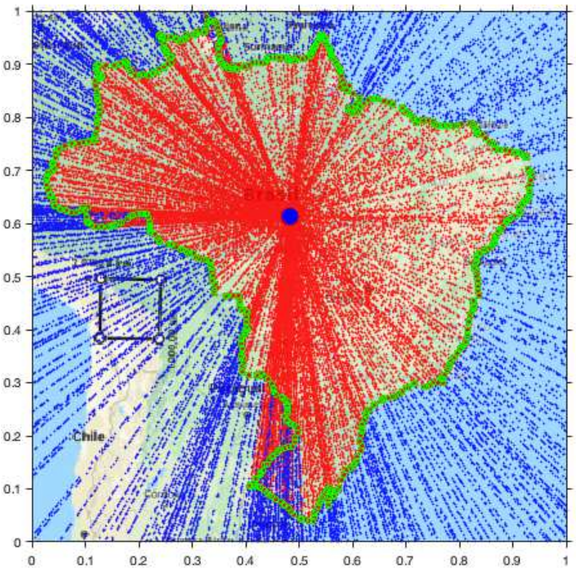
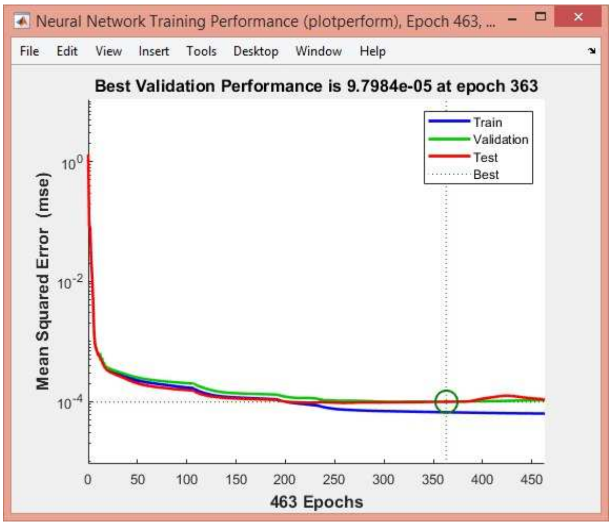
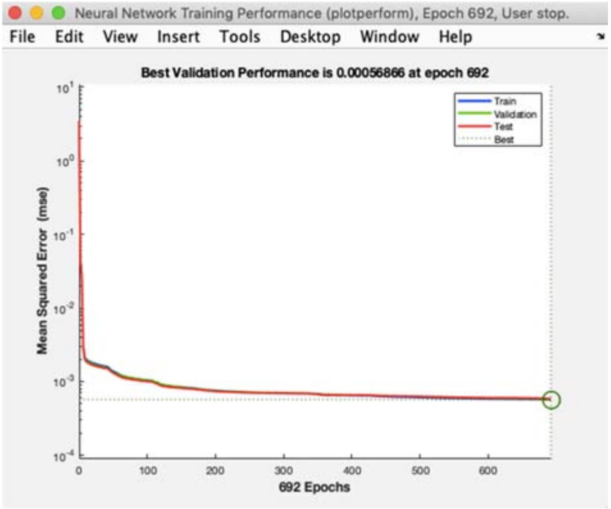


Figure 27. *Geometrically classified points provided for the training of the Neural Network on the map of Brazil.*



Best Validation Performance is 9.7984e-05 at epoch 363

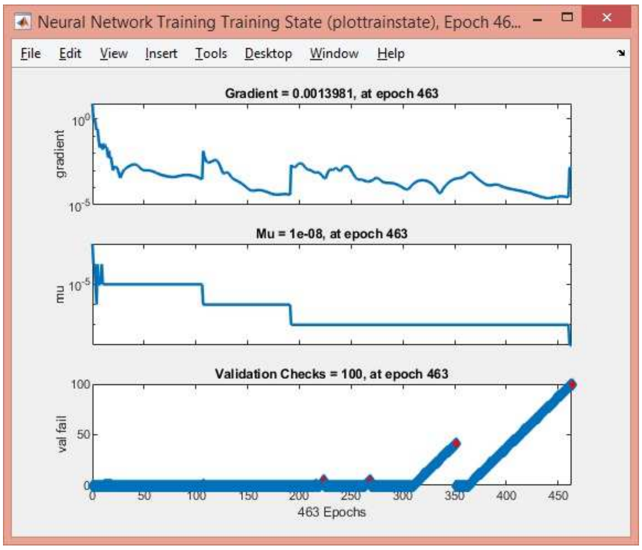
Figure 28. *Neural Network training performance on the map of France.*



Best Validation Performance is 0.00056866 at epoch 692

Figure 29. *Performance of the training of the Neural Network on the map of Brazil.*

In Figure 30, referring to the states of learning in France, is possible to perceive a softer variation of the gradient, indicating greater stability during the learning process.

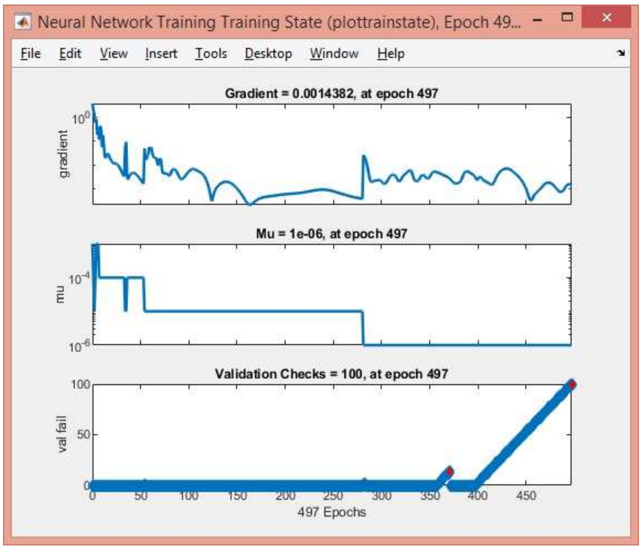


Gradient = 0.0013981, at epoch 463

Mu = 1e-08, at epoch 463

Validation Checks = 100, at epoch 463

Figure 30. *Neural Network training states on the map of France.*



Gradient = 0.0014382, at epoch 497

Mu = 1e-06, at epoch 497

Validation Checks = 100, at epoch 497

Figure 31. *Neural Network training states on the map of Brazil.*

Regarding the generalization capacity developed by ANNs, there is a higher quality of learning in figures 32 (France) in relation to 33 (Brazil). Again, there happens the impact of the non-convex areas on the prism shown in Figure 33, especially in the southern region of Brazil, on the borders of the state of Rio Grande do Sul.

The Figures 34 and 35 graphically demonstrate how RNA represents the knowledge for classification of points, according to the edges that were learned. Is possible to see clearly (and again) distortions happening in southern Brazil. The authors believe that this problem can be reduced by using more training patterns or less precise selection of border points so that non-convex contours are rectified and reduce this undesired effect.

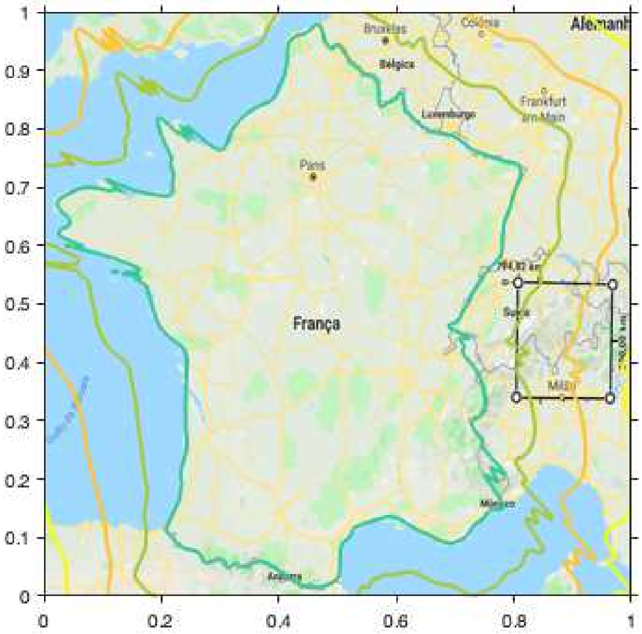


Figure 32. 2D Neural Network Learning Prism on the map of France.

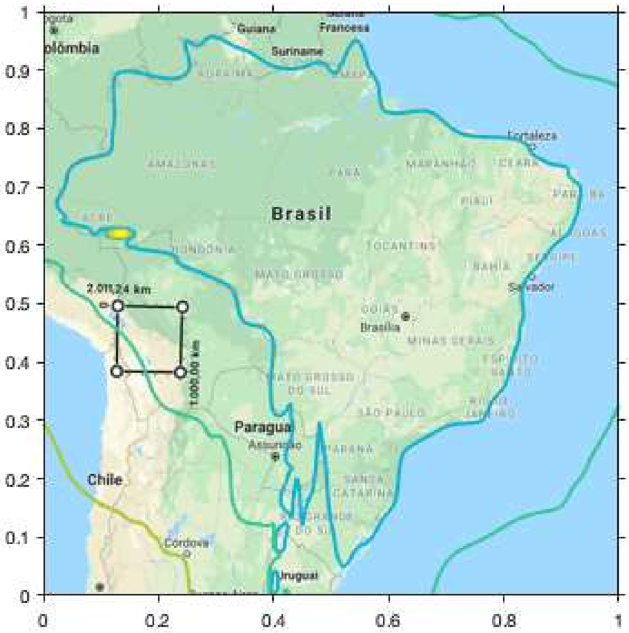


Figure 33. 2D Prism of Neural Network Learning on the map of Brazil.

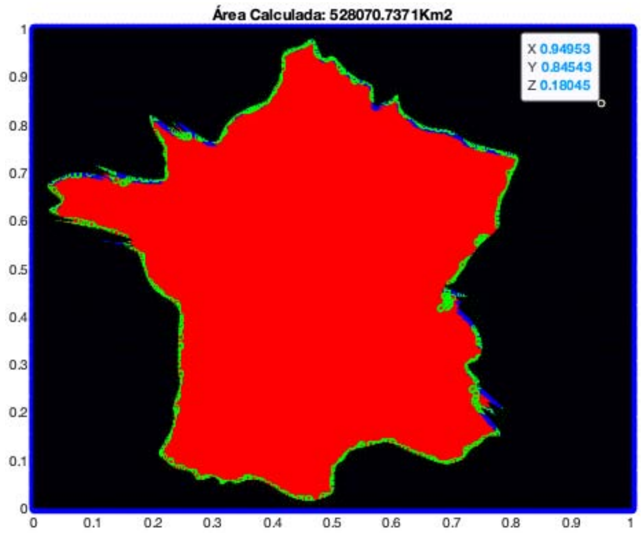


Figure 34. Representation of Neural Network Learning on the map of France.

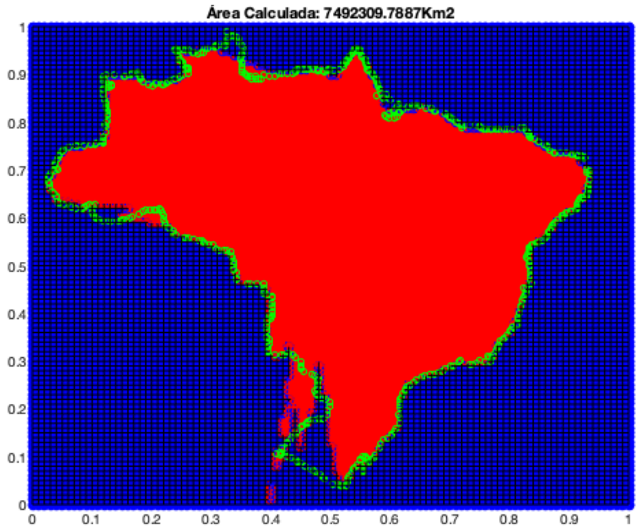


Figure 35. Representation of Learning of the Neural Network on the map of Brazil.

Also, when analyzed statistically using regression, it is clear that even with the problems of non-convex areas the network learns robustly. This is compiled by the very high R index, with both maps reaching value greater than 0.99 for all categories of data (training, testing, and validation).

Finally, the Table 2 synthetizes the estimates for the values of areas of each country, together with the official values (Km^2).

Table 2. Comparison between the area calculation of France and Brazil using the proposed approach and the official values.

Country	Area (Method)	Area (Official)	Hit (%)
France	528,071	544,000	97,07
Brazil	7,492,310	8,516,000	87,98

As comparison with above results, the algorithm was tested against a normalized circle inside a 4x4 cm square, and achieved the result of 3,1412 cm2, achieving 99,99% of pi accuracy.

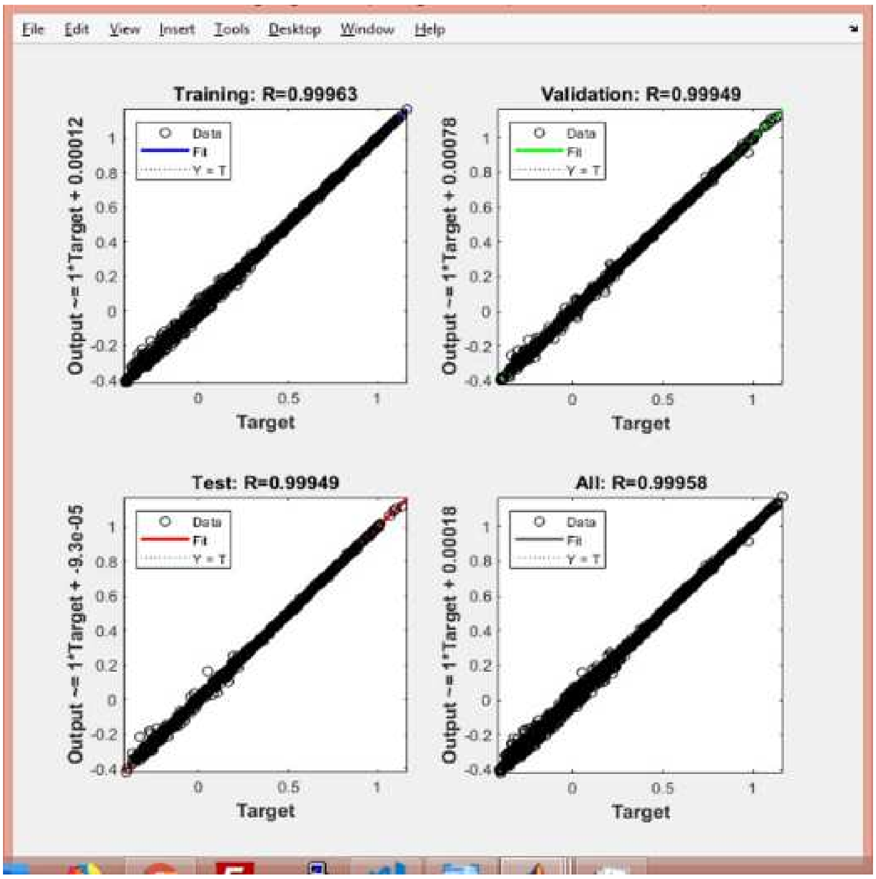


Figure 36. *Statistical regression on the training / test / validation and global Neural Network data on the map of France.*

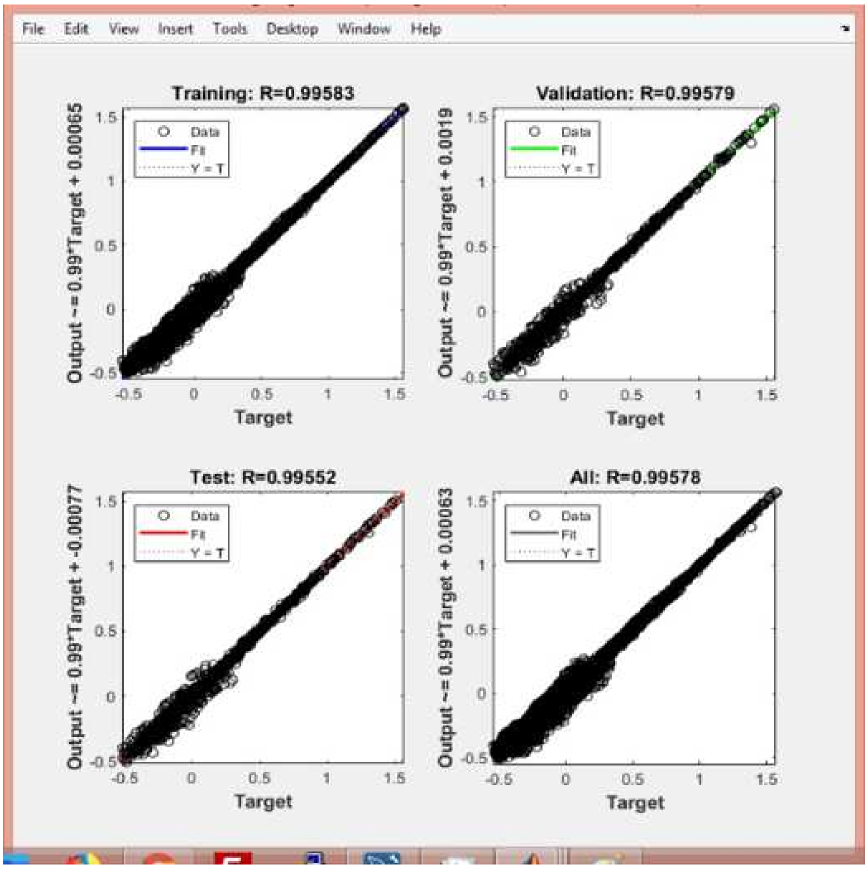


Figure 37. *Statistical regression on the training / test / validation and global Neural Network data on the map of Brazil.*

6. Conclusion

The objective of this article was to report the main results of a research work involving Monte Carlo Methods and Artificial Neural Networks, for the estimation of forest reserve areas from photos taken by drones.

The proposed method allowed an estimation of areas of a forest reserve, as well as the area of Brazil and France. Thus, we show that: it is possible to train an artificial neural network to classify randomly generated points and to signal them as inside and outside arbitrarily defined shapes, as long as they are as convex as possible; and it is also possible to calculate the area of the shapes using the amount of inside / outside points and the Monte Carlo methods.

The following challenges and requirements were successfully addressed in this research: Monte Carlo methods for area calculation, image edge detection and binary classification using artificial neural networks. However, there is not a bed of roses. The training of the neural network to determine the frontier of near convex topographic regions is quite complex and requires a lot of computational processing.

The Monte Carlo method, used to determine areas of irregular figures, is interesting because it is an alternative method that avoids using the Riemann integral. However, it is not trivial, since the determination of mathematical functions of arbitrary and nonconvex figures is also a very complex problem. For example, in hydroelectric dams you need to determine the area that will be flooded before closing the floodgates. The reason for this is diverse, since to accurately determine the potential energy of the waterfall that will be generated, as possible compensation for the expropriation of rural areas.

The most used method for the determination of this area, in this case, is through the triangularization method. This method has its origins in ancient Greece when mathematicians of that time were able to precisely determine the area of the perfect circle through triangularization using infinite triangles to fill the entire empty space of a circle. The automatic triangulation of non-convex complex figures through computational algorithms is also quite complex. Perfectly filling a figure of these with triangles will also require a multitude of them.

In this way, is possible to conclude that there is no easy way to determine the area of real-world non-convex geometric figures with automated way. Finally, a proof of concept applied to a real set of maps showed the partial effectiveness of the proposed method. For example, in determining the area of the United States of America we chose not to perform this experiment because there is a rather remarkable non-convexity in this map located in the Florida peninsula. However, this problem can be more easily solved if the map of the United States of America is divided in two: one part formed by Florida and one part by the rest of the country. These two problems are solved separately, and the two areas are summed together to obtain the total area.

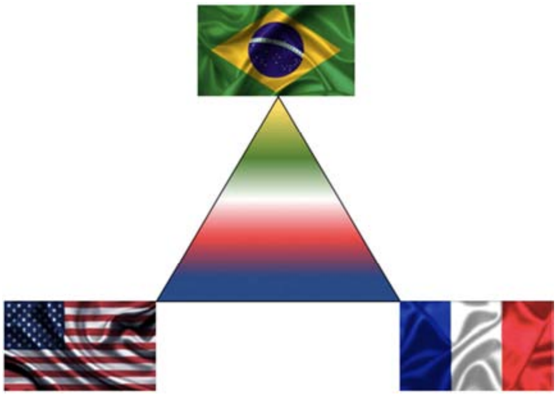
7. Future Work

As a natural continuation of this research and due to its importance in the environmental context, the authors of this article suggest the following works for future research, as the following extension of the proposed concept:

a) Obtain an algorithm that can automatically determine the boundary points of a forest reserve, without the need for human intervention (this will avoid the manual use of the mouse to generate these points);
b) To obtain an efficient algorithm to properly generate internal and external points of non-convex irregular figures that will later be used in the proper training of the neural network (this will allow to calculate with more precision the area of countries like Brazil and USA);
c) Its use to learn deforestation patterns gathered from image databases, whether satellite or other sources;
d) The consequent application in the estimation of the amount of carbon dioxide not extracted from the atmosphere, if there is deforestation of the forest reserve photographed by the drone;
e) In the not-so-distant future NASA could provide us with a complete multiple neural network of Earth's three-dimensional relief from satellite imagery;
f) Obtaining a neural network from Earth's relief has some interesting applications, such as accurately determining the volume of Earth's mountains and the volume of water in the oceans;
g) Our model of area calculation does not consider the curvature of the Earth and a possible improvement of this model would be to take this into account;
h) Performing the tasks proposed in this article using the Multilayer Perceptrons networks is very laborious, and, therefore, is suggested trying to use other more modern neural network architectures;
i) Finally, a study of the applicability of the proposed concept to other scenarios where the area estimation could not be done directly, and thus to diffuse the commercial use of the application developed here.

Acknowledgements

The authors would like to thank: the Brazilian Aeronautics Institute of Technology (*Instituto Tecnológico de Aeronáutica - ITA*) and the Casimiro Montenegro Filho Foundation (*Fundação Casimiro Montenegro Filho - FCMF*) for its infrastructure; to Mechanical Engineering student Leandro Tasinaffo de Melo for collecting, via drone, and granting the rights to the images used in this article; and the company Digital Business Ecosystem for the financial support of this research project.



"USA, France and Brazil: Three powers united by Science!"

References

[1] Q. Weng, D. Lu, A sub-pixel analysis of urbanization effect on land surface temperature and its interplay with impervious surface and vegetation coverage in Indianapolis, United States, International Journal of Applied Earth Observation and Geoinformation, 2008, 10 (1), pp. 68–83.

[2] J. Cihlar, Land cover mapping of large areas from satellites: Status and research priorities, International, Journal of Remote Sensing, 2000, 21 (6-7), pp. 1093–1114.

[3] Y. Hu, B. Nacun, An analysis of land-use changes and grassland degradation from a policy perspective in inner Mongolia, Sustainability, 2018, 10 (11), pp. 4048.

[4] B. Claas, N. Yunfeng, H. Tobia, Land-use change and land degradation on the mongolian plateau from 1975 to 2015—A case study from Xilingol, China, Land Degradation & Development, 2018, 29 (6), pp 1595–1606.

[5] J. Patino, J. Duque, A review of regional science applications of satellite remote sensing in urban settings, Computer Environments and Urban Systems, 2013, 37, pp. 1–17.

[6] J. Liu, W. Kuang, Z. Zhang, X. Xu, Y. Qin, Y, J. Ning, W. Zhou, S. Zhang, R. Li, C. Yan, Spatiotemporal characteristics, patterns, and causes of land-use changes in China since the late 1980s, Journal of Geographical Sciences, 24 (10), pp. 195–210.

[7] B. Quesada, A. Arneth, N. Noblet-Ducoudré, Atmospheric, radiative, and hydrologic effects of future land use and land cover changes: A global and multimodel climate Picture, JGR Atmospheres, 2017, 122 (10), pp. 5113–5131.

[8] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks, 1989, 2 (5), pp. 359–366.

[9] N. Cotter, The Stone-Weierstrass and its application to neurais networks, IEEE Transactions on Neural Networks, 1990, 1 (4), pp. 290–295.

[10] Y. Hu, Q. Zhang, Y. Zhang, H. Yan, A Deep Convolution Neural Network Method for Land Cover Mapping: A Case Study of Qinhuangdao, China, remote sensing, 2018, 10 (12), pp. 322–337.

[11] C. Cleve, M. Kelly, F. Kearns, M. Moritz, Classification of the wildland–urban interface: A comparison of pixel- and object-based classifications using high-resolution aerial photography, Computers Environment and Urban Systems, 2008, 32, pp. 317–326.

[12] Y. Qin, M. Chi, X. Liu, Y. Zhang, Y. Zeng and X. Zhao, Classification of high resolution urban remote sensing images using deep networks by integration of social media photos, IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, Netherlands, 2018, pp. 7243–7246.

[13] M. Omati, M. Sahebi, Change Detection of Polarimetric SAR Images Based on the Integration of Improved Watershed and MRF Segmentation Approaches, IEEE Journal of selected topics in applied earth observation and remote sensing, 2018, 11 (11), pp. 4170–4179.

[14] G. Matasci, N. Coops, D. Willians, N. Page, Mapping tree canopies in urban environments using airborne laser scanning (ALS): a Vancouver case study, Forest Ecosystems, 2018, 5 (31), pp. 1–9.

[15] Artificial neural networks <http://conteudo.icmc.usp.br/pessoas/andre/research/neural/> (accessed 26.05.2019).

[16] Neural Network and appllications in information retrieval system <http://www.scelo.br/pdf/%0D/ci/v35n1/v35n1a03.pdf> (accessed 26.05.2019).

[17] S, Haykin, Neural Networks, Bookman Publisher, 2007.

[18] Training of Artificial Neural Networks Based on Variable Structure Systems with Adaptative Learning Rate. < https://www.ppgee.ufmg.br/documentos/Defesas/669/Tese_Ademir_Nied.pdf> (accessed 26.05.2019).

[19] L. Fleck et al., Neural Network Basic Principle, Magazine of Eletronic Science Innovation and Technology, 2016, 7 (13) pp. 47–57.

[20] J, More, The Levenberg-Marquardt algorithm: implementation and theory, Numerical analysis, 1978, pp. 105–116.

[21] M. Loukrakis, A brief description of the Levenberg-Marquardt algorithm implemented by levmar, Foundation of Research and Technology, 2005, 4 (1), pp. 1–6.

[22] P. Tasinaffo, G. Gonçalves, A. Cunha, L. Dias, Using Monte Carlo method to estimate the behaviour of neural training between balanced and unbalanced data in classification of patterns, Artificial Intelligence Reasearch, 2018, 7 (2), pp. 1-25.

[23] N. Metropolis, S. Ulam, The Monte Carlo method, Journal of the American statistical association, 1949, 44 (247), pp. 335–341.

[24] C. Graham C., D. Talay, Strong Law of Large Numbers and Monte Carlo Methods, Stochastic Simulation and Monte Carlo Methods, Stochastic Modelling and Applied Probability, 2013, 68, pp. 13–35.

[25] J. M. Hammersley, Monte Carlo methods, Monographs on Applied Statistcs and Probability, 1964, pp. 178.

[26] Y. A. Shreider, The Monte Carlo method, Pergamon Press, 1966.

[27] P. Glasserman, Monte Carlo methods in financial engineering: stochastic modeling and applied probability, Srping Publlisher, 2003, 53, pp. 53.

[28] D. P. Kroese, T. Taimre and Z. I. Botev, Handbook of Monte Carlo methods. Wiley Series in probability and statistics", 2011.

[29] A. Papoulis, Probability, random variables, and stochastic processes, McGraw-Hill Europe Publisher, 2002, 4, pp. 55–57.

[30] A. Jazwinski, Stochastic processes and filtering theory, Dover Publisher, 1970, 12, pp. 104–106.

[31] J. Vuolo, Fundamentos da teoria de erros, Blucher Publisher, 1996, 2, pp. 32–34.

[32] Monte Carlo Methods - a special topics course. <https://www.math.arizona.edu/~tgk/mc/book.pdf> (acessed 26.05.2019).

[33] S. Beucher, C. Lantuejoul, Use of Watersheds in Contour Detection, International Workshop on image processing: Real-time Edge and motion detection/estimation, 1979.

[34] A. Schiavetti and A. Camargo, Basin Concepts: Theories and Applications, UFSC Publisher, 2002, pp. 17.

[35] V. Kalpana, Analysis of rain fall and the temperature of Coimbatore District using land use and land cover change detection by image segmentation, Multimedia Tools and Applications, 2018, 77 (23), pp. 30487–30504.

[36] S. Madhumitha, M. Manikandan, Quantitative analysis of marker-based watershed image segmentation, Current Science, 2018, 114 (5), pp. 1007–31013.

[37] A. Goudie and D. Cuff, Encyclopedia of Global Change. Environmental Change and Human Society, Oxford University Press, 2001.

[38] Decree n°23.793 <http://www.planalto.gov.br/ccivil_03/decreto/1930-1949/D23793.htm> (accessed 01.06.2019).

[39] Decree-law nº 4.471 <https://www2.camara.leg.br/legin/fed/lei/1960-1969/lei-4771-15-setembro-1965-369026-publicacaooriginal-1-pl.html> (accessed 01.06.2019).

[40] 1988 Brazilian Constitution, Art. 23 VI <http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm> (accessed 01.06.2019).