ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

Weightless neural systems for deforestation surveillance and image-based navigation of UAVs in the Amazon forest

Eduardo S. Ribeiro[1,2], Vitor A.M.F. Torres[1], Brayan R.A. James[1],
Mateus T. Braga[1], Elcio H. Shiguemori[3], Haroldo F. C. Velho[4],
Luiz C.B. Torres[1,2] and Antônio P. Braga[1] *

1- Federal University of Minas Gerais,
Department of Electronics Engineering, Brazil

2- Federal University of Ouro Preto,
Department of Computing and Systems, Brazil

3- Institute of Advanced Studies - IEAv,
Ministry of Defense, São José dos Campos-SP, Brazil

4- National Institute of Space Research - INPE,
São José dos Campos-SP, Brazil

Abstract. This work proposes a novel methodology for the recognition of deforestation areas in tropical forests using weightless neural systems in UAVs. The weightless neural systems embedded in hardware brings a considerable improvement in the speed of processing of image-based navigation of UAVs. In our approach the UAV navigates at the frontier of the deforestation area by means of previously trained descriptors, being able to monitor the increase of deforestation area. Experiments using images of the Amazon rainforest have been performed to validate the proposed approach.

1 Introduction

GPS has become a major aid to aerial navigation and a widely adopted solution by small and unmanned aerial vehicles (UAVs) localization. Although it can provide a relatively precise positioning information, signal quality depends on weather conditions and may fail or become unavailable [1]. Due to recent advances in pattern recognition systems and lower cost of hardware, visual navigation of UAVs has become a viable alternative to sole GPS navigation [2]. Most UAV aerial surveillance and inspection systems are based on small vehicles, which have load capacity limitations and require a careful selection of lightweight sensors and on-board computer. It is, therefore, particularly important to adopt pattern recognition systems that do not demand a high computational power and that are feasible to hardware implementation.

Weightless Neural Systems (WNS) are an attractive solution for implementing embedded pattern recognition systems, since they can be directly implemented in hardware as look-up-tables representing Boolean functions. This paper evaluates the use of WNS in visual navigation of UAVs and for deforestation

*Corresponding Author: apbraga@ufmg.br. This work has been supported by the Brazilian agencies CNPq, FAPEMIG and CAPES.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

surveillance. For visual positioning, the UAV is required to estimate its position according to its current view of the earth and a georeferenced image (a map) of the flight area that is provided in advance. By locating itself in the larger image, geographical coordinates can be estimated. In the approach presented in this paper, images captured from a real UAV flight are pre-processed with a WNS edge detection filter prior to recognition with a spatial correlation template matching approach.

Autonomous visual navigation is also evaluated for detecting deforestation in large green areas, such as in the Amazon forest. UAV surveillance in the region is particularly important, since it goes through long cloudy periods over the year, which restricts the use of satellite images in many applications. Also, ionosphere phenomena are more common in the southern hemisphere, which degrades GPS signal reliability and reinforces the need to search for alternative surveillance systems. Deforestation detection is formulated as a binary image classification problem with WISARD Discriminators [3]. This module converts a color image to 1-bit per pixel and this is used as input for the edge detection, which is also implemented with WISARD discriminators.

2 RAM Discriminator

A RAM discriminator [4] is basically formed by a single layer of p n-inputs RAMs, which are connected to the N-dimensional input field according to a pre-established connection criteria. Random connections are usually adopted, although the connection pattern may change according to the recognition task. The n-tuples corresponding to each RAM address are read directly from each input field to which the address lines are connected. Since connections are usually sparse ($n \ll N$) and randomly chosen from the input field, each RAM learns a (different) randomly selected Boolean function $h_j(\mathbf{x})$. Their outputs are then summed-up in order to produce discriminator's response, so the output of a RAM discriminator $r_i(\mathbf{x})$ to a given input pattern $\mathbf{x}$ can be generally represented $r_i(\mathbf{x}) = \sum_{j=1}^{p} h_j(\mathbf{x})$, where $h_j(\mathbf{x})$ is the Boolean function performed by RAM neuron j.

Training a RAM discriminator is accomplished with a one-against-all strategy, so each RAM is trained only with examples of the class associated to the discriminator it belongs to. Recognition occurs by presenting the input pattern $\mathbf{x}$ to all discriminators, which respond with the number of matches with previously trained examples. Final classification is then accomplished with a Winner-Takes-All (WTA) strategy, that assigns to $\mathbf{x}$ the class of the discriminator with the largest response $r_i(\mathbf{x})$.

3 UAV Position Estimation through images

Images captured during a UAV flight over the city of São José dos Campos, Brazil, are used in the following experiments. The goal is to estimate the UAV position having as input in-flight captured images and a previously georeferenced image of the whole navigation area. Position estimation is accomplished by locating the current image captured by the UAV, with a Template Matching [2, 5] approach, in the geo-referenced image. Captured images may present distortions due to camera resolution, illumination and perspective effects due to UAV axes of rotation, that may also affect position estimation [6]. So, edge detection is used prior to Template Matching in order to reduce discrepancies between images, which were captured at different time slots, under different conditions and with different image sensors.

Experiments were carried out with the following techniques for image distortion correction: perspective correction, rotation axes adjustment and WNS edge extraction. Perspective correction considers UAV's axes angles (Yaw, Pitch and Roll) and the intrinsic camera parameters in order to obtain the homographic matrix H, which is applied to the image in order to obtain the projection transformation. For rotation adjustment, only Yaw angle is considered [6]. Edge detection is accomplished with a 3×3 kernel filter implemented with a single layer WNS, that learns from pre-established edge patterns. Figure 1 presents a general overview of the methodology adopted for position estimation. Maximization of the spatial correlation between images, after perspective correction and edge detection, is used to estimate position. In the experiments that follow, the Canny operator [7] was also used in order extract edges and compare the WNS results with another edge detection approach.

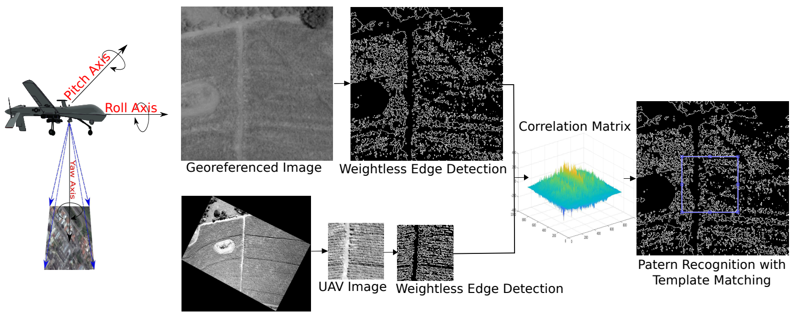


Fig. 1: General overview of UAV position estimation processing flow with edge detection accomplished by a weightless neural function.

The resolution of the georeferenced embedded image was 0.5 $m/pixel$ and its size 9872×10386 $pixels$. So the covered flight area had aproximately 5×5 kms. The size of the original UAV in-flight captured images was reduced to approximately 13% of its original resolution resulting in 220×220 $pixels$. As can be observed in Fig. 2, the planned flight was accomplished in a countryside region with predominance of green vegetation. The figure also shows the planned route and position estimation results for the three methods. Table 1 summarizes the performance of all methods considering distance estimation error, standard deviation, False Positive ($FP_{img} = 1$ if $Dist.$ $Error_{img} \geq 100$ m) and True Positive ($TP_{img} = 1$ if $Dist.$ $Error_{img} < 100$ m) errors. As can be observed, the best performance was achieved with perspective correction followed by edge detection with Canny operator [7]. WNS approach, however, had a competitive

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

performance, particularly considering the standard deviation of the errors and the simplicity of implementation, as will be discussed in the next sections.

Fig. 2: Position estimation for the three methods. The squares in the image represent the real position, circles represent the estimation with perspective correction, triangles represent the estimation with rotation correction and the red stars the WNS approach with perspective adjustment.

Table 1: Performance considering the distance to the real UAV position. (FP and TP, when the position estimation of the compared algorithms exceeds or not an error greater than 100 meters)

Case	Av. Dist. Error	Std Dev.	FP(%)	TP(%)
Perspective+Canny	**36.04**	20.27	**0 %**	**100 %**
Rotation+Canny	50.01	25.70	2.86 %	97.14 %
Perspective+WNS	44.94	**17.50**	**0 %**	**100 %**

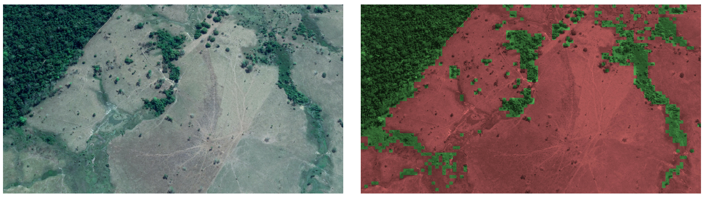
4 Deforestation Surveillance

Google Earth RGB images of the Amazon forest, representing forest green areas and deforestation, were selected for the experiments. Classification was accomplished on pixel level, so each image channel was discretized and represented with 4 bits, resulting on 12 bits per pixel. Since the two classes are well defined, discretization levels was not too critical, so 4 bits per channel was enough to properly represent the two classes. Two discriminators were then trained with randomly sampled examples of pixels extracted from forest and deforested training images. A total of 169 pixel examples were selected for training, with 56 examples of deforestation and 113 of forest. Each discriminator was constructed with 5 12-inputs RAMs, which were randomly connected to the pixel depth representation.

Although discriminators were built as classifiers on the pixel level, images were not scanned on a pixel-by-pixel basis. Since the objective was to identify the presence or not of larger deforested areas, some degree of discretization in the detected regions is acceptable. So, in order to reduce image scanning costs, square regions of the input images were classified instead of individual pixels. For each square region, 50% of the pixels were randomly selected and classified.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Each square region was classified according to a majority voting amongst all pixel classifications, resulting on a certain degree of discretization, depending on the square size, as can be observed in Fig. 3. It is interesting to notice in the figure that the two regions were correctly identified and that, even some trees in the deforested area were spot by the classifier. Although the deforestation region has some green tones, it was correctly classified, which is particularly interesting in this context, since some of them can be used for cattle farming and are actually green. This outcome is explained by the fact that training samples were selected from dense forest areas, which are quite characteristic and differ from the pale green tones in the deforested area.



(a) Original image. (b) Detected regions.

Fig. 3: Examples of deforestation identification with WISARD discriminator. Deforestation region is shown in red. In this example, 50×50 were classified according to the majority voting of pixel classification within each region.

5 FPGA Implementation

The edge detector and deforestation discriminators were synthesized considering the Xilinx Zynq-7000 family, which integrates ARM processor cores with a FPGA fabric. This architecture is particularly appropriate for this problem, since the CPU can be dedicated to higher level tasks, while accelerating specific procedures in the FPGA, such as edge detection and image classification. The considered model was the xc7z020clg484, which contains a dual-core ARM Cortex$^{\text{TM}}$-A9 CPU and a FPGA Artix-7 with 53200 Look-Up Tables (LUTs).

The synthesis of the edge detection circuit in the FPGA for a single RGB pixel required the space presented in the first row of Table 2. As can be observed, very little of the whole FPGA space available was required, which suggests that a large number of edge detectors could be implemented in the same FPGA in order to process kernels in parallel. For instance, the parallel processing of three rows of a 900×900 image would require around 3.9% of the LUT space, considering that each kernel is composed by 9 pixels.

The combinational circuit of the deforestation discriminators was also synthesized for the same IC, which resulted in the LUT space required presented in the second row of the Table 2. Similarly to the edge detection circuit, the discriminators also required very little space of the FPGA, which suggests also that several discriminators could be implemented in parallel. For instance, processing a 50×50 region in parallel, considering 50% of the pixels, would require 33% of the LUTs. In the final application, since the deforestation detection circuit

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

output is in fact a binarization of the input image, it could be used as the input of the edge detection module, simplifying the full implementation.

Table 2: LUTs required for the implementation of the edge and deforestation detection circuits for a single RGB pixel.

Resource	**Estimation**	**Available**	**Utilization %**
LUT (Edge Detection)	7	53200	0.01
LUT (Deforestation)	14	53200	0.03

6 Conclusions

We proposed a new methodology for the detection of deforestation regions using UAVs and Weightless neural systems. Experiments with real images of the Amazon rainforest showed that the WNS were able to identify deforestation areas accurately, and the extension of the work to other types of surveillance such as burnt areas and invasion of environmentally protected areas becomes highly feasible. It can be observed that the WNS in hardware makes UAV navigation processing time very fast. The complete implementation certainly would not use IC pins for input and output, but the combinational logic would be the same. It is very likely that an architecture which pipelines the output of the deforestation discrimination to the edge detection would require less space than the conservative evaluation: $(14 \times 9) + 7 = 133$ LUTs, for each kernel. Even this pessimistic estimation represents only 0.25% of the LUT space available in the selected component, which is not among the most complex ones from the same family. From this analysis we conclude that it is possible to implement combinational blocks to handle several kernels in parallel, while still retaining enough capacity to implement the high speed bus communication between the CPU and the FPGA and also add the sequential circuits which will scan the image rows.

References

[1] PM Kintner, BM Ledvina, and ER De Paula. Gps and ionospheric scintillations. *Space weather*, 5(9), 2007.

[2] Gianpaolo Conte and Patrick Doherty. An integrated uav navigation system based on aerial image matching. In *Aerospace Conference, 2008 IEEE*, pages 1–10. IEEE, 2008.

[3] I. Aleksander, W.V. Thomas, and P.A. Bowden. Wisard: as radical step forward in image recognition. *Sensor Review*, 4:120–124, 12 1984.

[4] I. Aleksander. Self-adaptive universal logic circuits. *Electronics Letters*, 2(8):321–322, August 1966.

[5] GAM Goltz, EH Shiguemori, and HFC Velho. Position estimation of uav by image processing with neural networks. In *X CBIC*, pages 1–6, 2011.

[6] Brayan Rene Acevedo Jaimes. Estratégias Para Aumentar a Robustez De Estimação De Posição Geográfica Em Vants Através De Imagens. Master's thesis, UFMG, 12 2016.

[7] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.