10th Brazilian Congress on Computational Intelligence (CBIC'2011), November 8 to 11, 2011, Fortaleza, Ceará Brazil
© Brazilian Society on Computational Intelligence (SBIC)

POSITION ESTIMATION OF UAV BY IMAGE PROCESSING WITH NEURAL NETWORKS

Abstract – This paper presents a study of three artificial neural networks with supervised training and different architectures: network with radial basis function, multilayer perceptron and cellular neural network. These networks were applied to edge detection in aerial and satellite images, for later correlation calculation in spatial domain between these images to simulate the estimation of the geographical position of a unmanned aerial vehicle - UAV. The neural networks results were compared with Sobel and Canny operators.

Keywords – Artificial Neural Networks, Image Processing and Air Navigation by Images.

1 Introduction

The application of UAV stepped up lately due to low operational and manufacture costs compared to conventional aircrafts, don't have crew, among other factors. UAVs are employed in remote sensing, with synthetic aperture radar (SAR) or optical sensors on board, for monitoring cities, human activities or areas of risk such as conflict zones or natural disasters. Radio controlled UAVs usually use antennas with linited reach. In addition, when flying they are subject to possible interferences in the communication process with the control base. Interference in the signal can also occur with UAVs use Global Positioning System (GPS) signals combined with the INS (Inertial Navigation System) for navigation. Possible blocking of GPS signals may also become a limitation to its use. An alternative to overcome these problems is to make use of air navigation systems based real time captured ground information.

The use of image processing, computer vision, geoprocessing, and pattern recognition techniques applied to autonomous navigation of UAVs is seen in some studies as in [1], using digital elevation model, in [2] using edge information for the registration of aerial images from satellite image, stereo vision in [3], matching between images in [4], among many.

Many papers use ANNs in the navigation system of UAVs, research by [5] shows it. In the literature there are few studies using ANNs in image processing and / or pattern recognition in images in order to help the navigation of UAVs. But the use of ANNs for this purpose can be seen in some studies as in [6], that developed an ANN classify surface in candidate regions for crash in the landing phase of a UAV after the segmentation of the terrain flown. The landing area is chosen using *fuzzy* rules that use data of the terrain available in the database of the covered areas. The work of [7] also developed an aplication in which ANNs are used in the processing of captured images of the area flown by the UAV. Initially, the texture of the image is extracted by a Gabor filter and an ANN searches the area between some landmarks used for navigation of the UAV, derived from a satellite image. An MLP and a self-organizing map of Kohonen - SOM - were developed for this purpose and their results were compared.

Many tasks performed by the conventional algorithms in the areas of image processing, computer vision and pattern recognition, of interest for autonomous air navigation for images of UAVs may be performed by ANNs, for example: separation of texture images in real time [8], matching in stereo vision [9], detection of edges [10], segmentation [11], among many.

In recent decades ANNs of different architectures have been developed for edge detection in images. For example, cellular neural networks - CNNs - in [12]. In [10] several architectures for this task: MLP, RBF, Bidirectional Associative Memory, Adaptive Resonance Theory, Learning for Vector Quantization, Maxnet and Fuzzy Associative Memory. In [13] networks based on the primary visual cortex. In [14] an adaptive neural networks based on edge patterns.

The objective of this study was to develop three ANNs for detecting edges. The following networks architectures were tested: RBF, MLP and CNN. They are applied them to estimate of the geographic position of a UAV through the correlation in spatial domain between the edges of aerial images - captured by the UAV - and georeferenced satellite image. Initially this work shows the development of the three ANNs and after shows the estimation of the geographic position of the UAV through the correlation in spatial domain between the edges of aerial and satellite images, with use of ANNs, Sobel operator [19] and Canny operator [20]. This work uses Sobel operator because it was efficient in [2] for the same task and it uses Canny operator because it was used in [10] witch a ground truth. Following, an explanation of each ANN used at work.

2 Neural Network with Radial Basis Functions

We applied the method of *self-organized selection of centers* [15] to define the centers and weights of the RBF. These centers were defined as Gaussian function. Although the original model of the RBF network output functions considers the network as a linear model, this study chose to use the hyperbolic tangent function in the output layer.

The architecture of the proposed network has nine entries, which correspond to each element of a window 3 x 3 that moves on the image pixel-to-pixel, either vertically and horizontally. The hidden layer has a number of neurons corresponding to the number of centers found in the clustering process [15]. The output layer has one neuron for the estimated edge information.

The training of the RBF was based in [10], which generated fourteen and twenty six edge patterns for training the ANNs. These patterns were described in a 3 x 3 window with binary values. In this work we used eight edge patterns equals some to [10] and more two additional patterns were proposed to train the networks to identify the absence of edge as Figure 1. This Figure also shows the output for each pattern.

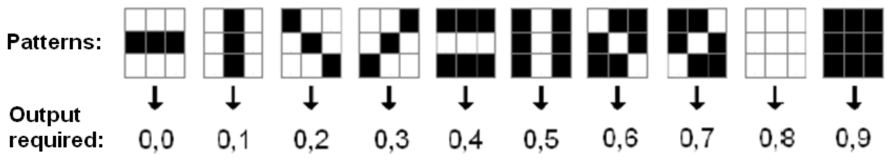

Figure 1 – Eight patterns edge and two without edge of the training set.

After training, the RBF network was activated with an image as follows: a 3 x 3 window moves over the image in a pixel-to-pixel basis (horizontal and vertical) and captures this region, which is normalized and sent to the RBF in the form of a 1 x 9 vector for activation of each layer.

Here we developed a new way of interpreting the output of the network. The output of the network (a scalar) is compared with all the desired outputs of the network. The pixel is assigned with the edge information corresponding to the central value of the pattern that is closer to the network output. However, in order to cope with the edge detection task, this core value of the 3 x 3 window turns into 0 or 1, depending on the pattern. Thus, the output image contains only binary values.

3 Neural Network Multilayer Perceptron

The MLP is a network based on the operation of the perceptron of Rosenblatt [15], having one or more hidden layers and one or more neurons per layer. The MLP of this work has nine neurons in the input layer, corresponding to each element of a 3 x 3 window and it works similarly to the RBF network. The hyperbolic tangent function was used as the activation function of all neurons of the MLP.

We used the *error backpropagation algorithm* [15] to train the MLP. The set of input training patterns were the same used for the RBF, as well as the desired output.

The MLP was activated in the same way of the RBF, i.e., extraction of a 3 x 3 window of the image. This window is transformed into a normalized 1 x 9 vector. These elements active the network and the output of the network, a scalar, is compared with the values of output for each pattern, as in the RBF. The central pixel - 0 or 1 - of the pattern that is closer to the output of the network is assigned to the output image.

4 Cellular Neural Network

The CNNs were proposed by [16]. Each neuron connects with its adjacent neighbors. The neighborhood of a neuron can be represented by (1):

$$V(i,j) = c(k,l) : \max(|k-i|,|l-j|) \leq r \tag{1}$$

where *V(i,j)* represents the set of neighboring cells *c* with indices *k,l* of the cell with index *i,j,* where *r* is the radius of the neighborhood, and $1 \leq k,i \leq N, 1 \leq l,j \leq M$, *N* and *M* the size of the CNN. The CNN built in this work has dimensions 3 x 3 with neighborhood between cells with radius *r=1.*

Mathematically, a CNN operates according to (2) and with rule of (3):

$$\hat{x} = sat(x + Tx + B) \tag{2}$$

$$sat(x_{i,j}) = \begin{cases} -1 \rightarrow x_{i,j} < -1 \\ x_{i,j} \rightarrow -1 \leq x_{i,j} \leq 1 \\ 1 \rightarrow x_{i,j} > 1 \end{cases} \tag{3}$$

where $\hat{x}$ is the output of the network, *x* is the input of network, *T* is the sparse matrix that contains the weights of the connections between neurons, *B* is the vector of bias, $sat(x_{i,j})$ is the equation of saturation of neuron *i,j* of the network and $x_{i,j}$ is the input *x* corresponding to neuron *i,j*. During activation of the network, (2) is fed with its own output until *x* become stable, that is, until $\hat{x} \approx x$.

In the training of the CNN we used thirty two edge patterns and two patterns without edge as Figure 3. The patterns were composed of bipolar values, -1 and 1. These patterns were the desired output of the network and the entries for training are the same patterns with a Gaussian noise of 20%. [17] reproduced and compared five different training algorithms for

CNNs, among them a method based on perceptrons was developed by [18]. This method was used here. If *T* is a sparse matrix containing the connections between neurons and *B* is the vector of bias, we have:

$$W^i = [w_1^i, w_2^i, \ldots, w_i^i + \mu, \ldots, w_n^i, B_i] \text{ and } \bar{y}^p = [y^p, 1] \tag{4}$$

where $w_{(i,j)} = T_{(i,j)}$ if *j*-th cell belonging to the neighborhood of *i*-th cell and $w_{(i,j)} = 0$, otherwise and y^p o *p*-th pattern. According to [18] using the constant $\mu > 1$, and y_i^p the *i*-th element of *p*-th pattern, we have:

$$\begin{aligned} W^i \bar{y}^p \geq 0 \rightarrow y^p = 1 \\ W^i \bar{y}^p < 0 \rightarrow y^p = -1 \end{aligned} \tag{5}$$

The weights W^i of (5) are found by the algorithm of perceptron training [15]. The objective of the training of a CNN is to obtain *T* and *B* using the perceptron training algorithm and evaluate the stability of *x* in (2). If *x* is not stable, it is estimated again *T* and *B*, using *x* as input in the training of the perceptron, and so on until *x* is stable, reaching a minimum error between consecutive values of *x*.

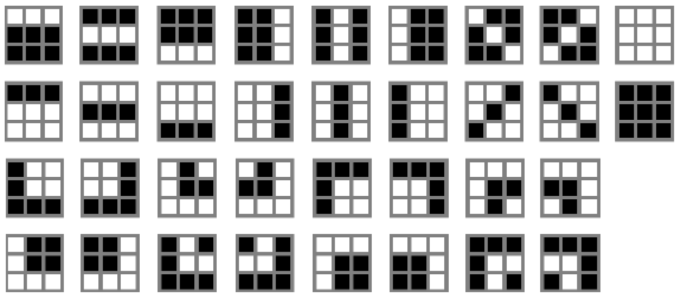

Figure 3 – Polar edge patterns of the CNN training set.

The CNN is activated with an image the same way as the other networks. The 3 x 3 window is transformed into a 1 x 9 vector. The elements of this vector are set to values between -1 and 1. This vector activates (2). The output of (2) is used as input activation in the new iteration until *x* is stable. The central value of the vector *x* is assigned to the stable image output.

5 Position Estimation of UAV

The geographical position of UAV can be estimated by matching images of the land, captured in real time and by georeferenced satellite images on board of the UAV according to the region flown. There is a high computational cost for finding a small region flown in a satellite image of the entire region containing the flight plan. This cost can be reduced by the search of the flown region around a point estimated by the inertial sensor of the UAV, which is fixed by periodically matching the images.

Factors such as scale, rotation, difference in brightness and difference in spectral response of the scene caught between the two systems (camera imaging of the UAV and the satellite imaging system) mean that the same scene is represented differently in the real time captured image and the satellite image used, onboard the UAV.

One common factor is that both images were captured on target nadir, eliminating effects of perspective. The rotation can be corrected by information from the compass of the UAV. The difference in scale between the images can be estimated by information from the altimeter of the UAV. Details of the land in both images can be eliminated applying a low-pass filter to these images. This means that only the more relevant objects and features of the land are kept in the images. Factors related to differences in brightness and spectral response are eliminated using only the edges extracted from images.

The correspondence between the aerial and satellite is made as follows: initially the image captured in real time by the UAV is corrected in scale and rotation according to the region of the satellite image loaded onto the UAV. This region is defined around the point where the inertial sensor is estimated to be the geographical position of the UAV. A median filter of a 3 x 3 convolution mask is applied to the two images in order to eliminate small details in images. Then the edges are extracted. For this purpose, in this work we used the ANNs presented, Sobel operator and Canny operator. With the edges of two images, the matching is done by the calculation of the correlation [19] in spatial domain, given by (6):

$$c(s,t) = \sum_x \sum_y f(x,y) w(x-s, y-t) \tag{6}$$

where *c(s,t)* is the correlation *c* indices *s,t*, with *s=1,...,M-1*, e *t=1,...,N-1*, *M* and *N* are the dimensions of the matrix *f*, *M* x *N*, that contains the edges of the satellite image, and *w* matrix, with dimensions *J* x *K* with *J* < *M* and *K* < *N*, containing the edges of the aerial image. Given the higher correlation *c* found between the matrices *f* and *w*, the point on the satellite image *f* corresponding to the central point of the aerial image *w* corresponds to the location of UAV at the time that such region of the image was flown. Knowing this point, its coordinates correct the inertial sensor and the process of matching resumes. Figure 2 summarizes the process of matching described. To accelerate the calculation of the correlation, the aerial image has its area reduced around its central point, after the correction of scale and rotation.

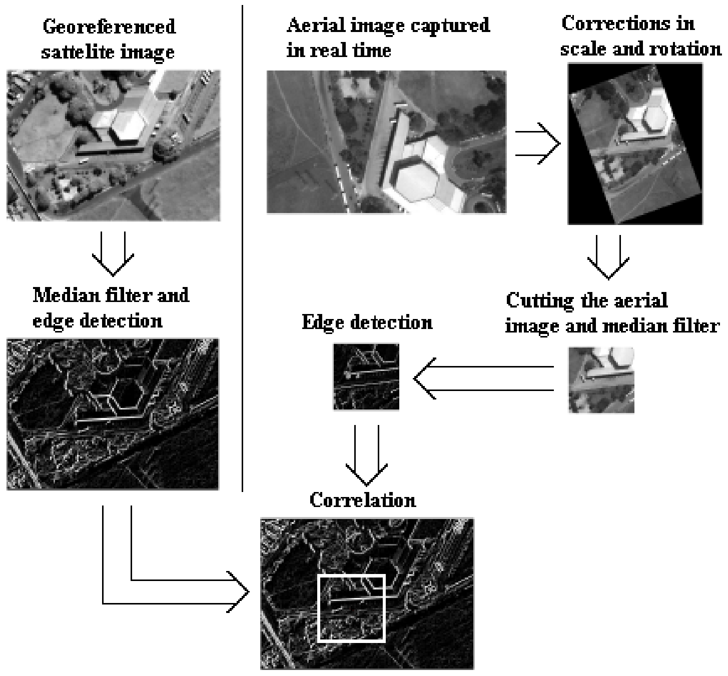
Georeferenced sattelite image
Aerial image captured in real time
Corrections in scale and rotation
Median filter and edge detection
Edge detection
Cutting the aerial image and median filter
Correlation

Figure 2 - Steps of the process of matching.

To simulate the images captured by a UAV in real time, images of an aerial videography held at a constant altitude were used. The satellite image used has a high spatial resolution of 0.6 meters. The calculation of correlation was performed on ten pairs of images (aerial and satellite images) that simulate ten different regions flown by the UAV. Each satellite image contains the region corresponding to the aerial image. Noises were included in the values of scale and rotation to assess the robustness of the method to estimate the UAVs position is how a table should be prepared for this conference.

6 Results

For training the RBF the following parameters were considered: $\sigma = 0.3$ of Gaussian function; learning rate $\eta = 0.05$, and the moment constant $\alpha = 0.5$. The last two mentioned parameters are used in the error backpropagation algorithm [15] used for training the weights of the output layer. As stopping criterion for this algorithm we used a threshold for the least mean squared of the network equal to 10^{-6} and the maximum number of epochs of 10000.

The MLP architecture was composed of two hidden layers and seven neurons in each hidden layer. The training parameters used were $\eta = 0.05, \alpha = 0.5$, for the criterion to stop the iterations we used least mean squared error threshold of 10^{-4} and maximum number of epochs equal to 50000.

The perceptron of the CNN was trained with $\eta = 0.05$ and maximum number of epochs equal to 100. To evaluate *x* stability in (2), we considered the error reached between two consecutive values of *x* to be less than or equal to 0.05 and a maximum limit of 100 of iterations for the *x* stability. The error limit for the stability of *x* in activation was 0.09 and the maximum number of iterations in (2) was 20.

The parameters chosen for the three ANNs reached the minimum error established in the training. The range and the variation of these parameters are shown in Table 1.

Table 1 – ANN Parameters, values tested for each parameter and best parameter for each ANN.

ANN Parameters	Values	RBF	MLP	CNN
Max. N of epochs RBF/MLP	1000; 5000; 10000; 20000; 30000; 40000; 50000 e 60000	50000	50000	-
Max. N of epochs CNN	50; 100; 200; 500	-	-	100
N of centers	10; 34	10	-	-
Distância of centers	1; 1.5; 2, 2.5; 3	1.5	-	-
η	0.001; 0.002; 0.003; 0.004; 0.005; 0.006; 0.007; 0.008; 0.009; 0.01; 0.02; 0.03; 0.04; 0.05; 0.06; 0.07; 0.08; 0.09; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9	0.05	0.05	0.03
α	0.01; 0.02; 0.03; 0.04; 0.05; 0.06; 0.07; 0.08; 0.09; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9	0.5	0.5	-
Min. error of training	$10^{-4};10^{-5};10^{-6}$	10^{-6}	10^{-4}	-
Max. N of iterations por *x* estabilization	20; 50; 100; 200; 500; 1000; 1500; 2000	-	-	1000 (training) 50 (activation)
***x* estabilization**	0.001; 0.005; 0.01; 0.02; 0.03; 0.04; 0.05; 0.06; 0.07; 0.08; 0.09; 0.1	-	-	0.01 (training) 0.02 (activation)

Noises in scale and rotation were entered for the UAV position estimation. These noises were: 0%, $\pm 5\%$ and $\pm 10\%$. The correlation was calculated between 10 pairs of images for each situation above. The average error, in meters, was calculated for each noise condition set and for each edge extractor used. The error in meters was obtained by the Euclidean distance between the point estimate in the correlation and the point is taken as truth by multiplying the distance in pixels for 0.6, which corresponds to the spatial resolution in meters of each pixel.

Considering as *hitting* a point estimation with an error less than or equal to 10 meters, one can evaluate the performance of different edge extractors used in this work considering the amount of hitting for each noise situation of scale and rotation. Table 2 shows the average error in meters and the hitting for each operator and for each noise situation.

Table 2 – Estimation mean error and hitting for each operator and for noise situation of scale and rotation.

Escale Noise	**Mean Error**					**Hitting**				
	Rotation Noise									
	0%	**+5%**	**-5%**	**+10%**	**-10%**	**0%**	**+5%**	**-5%**	**+10%**	**-10%**
	CNN									
0%	7.9	9.3	10.7	16.4	11.9	8	7	8	4	8
+5%	20.0	17.6	18.5	20.8	19.4	5	6	5	3	6
-5%	14.5	17.5	13.9	32.6	14.5	8	6	9	4	9
+10%	13.8	17.9	19.5	17.5	19.8	5	4	4	3	5
-10%	23.6	43.4	22.6	38.5	23.9	7	4	8	3	8
Sobel										
0%	17.6	26.1	24.0	37.0	22.4	8	5	7	3	8
+5%	27.5	25.8	21.7	33.6	22.1	5	5	7	4	6
-5%	18.8	22.4	14.6	29.3	14.4	7	5	8	4	9
+10%	32.2	37.8	27.7	52.7	34.2	4	3	7	1	3
-10%	33.0	41.4	25.8	47.4	26.4	6	3	6	3	7
MLP										
0%	35.6	21.7	29.2	31.8	28.0	7	7	6	5	6
+5%	33.1	27.8	26.7	41.2	23.6	4	4	6	2	5
-5%	27.0	40.8	30.9	47.0	25.3	5	4	4	3	6

+10%	28.6	37.6	38.0	30.2	31.8	3	3	4	3	5
-10%	33.9	33.5	26.1	41.8	33.0	4	4	5	4	4
RBF										
0%	11.0	29.4	38.7	50.8	47.8	9	5	4	2	2
+5%	30.2	41.5	29.1	51.5	36.4	4	2	4	2	3
-5%	45.1	57.5	39.4	47.9	51.6	4	2	5	2	3
+10%	49.7	48.9	32.6	49.8	50.6	2	2	4	1	2
-10%	55.4	36.4	54.8	46.2	63.2	2	4	2	2	2
Canny										
0%	44.1	73.4	45.5	63.5	54.5	5	1	3	1	2
+5%	66.1	62.1	55.1	74.6	56.4	2	2	2	0	2
-5%	44.5	61.7	49.5	68.9	54.9	3	1	3	1	3
+10%	70.8	68.3	57.9	72.0	66.4	1	1	1	0	0
-10%	51.3	60.0	53.8	76.3	52.7	3	2	2	0	2

In Table 3 are the average processing times, in seconds, for the operation of extracting edge held by the operators studied for the aerial and satellite images. Figure 4 shows some edge examples extracted by edge operators this work in an aerial image and a satellite image.

Table 3 – Average processing time in seconds of each operator for the aerial and satellite images.

Operator	**Aerial Image**	**Satellite Image**
CNN	2	13
Sobel	<1	<1
MLP	0.5	2
RBF	2	10
Canny	<1	<1

5 Discussion and Conclusions

The output image containing the edges generated by the three ANNs have binary values, does not require the use of thresholds, this process is already inserted into the ANN in how their outputs are interpreted. The Canny operator uses the operation of hysteresis and thresholding to binary its output, which requires initial parameters. The CNN obtained better results compared with the other operators used in average error and in hitting. The second best was the Sobel operator. The other operators did not demonstrate satisfactory results for the application, and the Canny operator results less impressive. By analyzing the tests without inserting error in scale and rotation, the RBF had the best result in the number of hitting. Taking into account the average error in estimating the position of the UAV to this situation, CNN missed less. CNN has a processing time bigger than the other operators, although it provided the best results for the application. The processing time for the satellite image for this application is not relevant as it can be processed before the flight of the UAV. Only the aerial image requires real time processing, and in this case, the processing times are similar.

For the learning of the ANNs in this work we used the idea of [10]: learning with edge patterns with binary values. But the training and activation of these ANNs are done differently from [10]: in the RBF and MLP, the output of the network is associated with a scalar number representing the edge pattern detected. On activation of the CNN, the pattern edge is calculated by a recurring operation, as in [17]. The training is based on the perceptron learning, using edge polar patterns with noise associated with the same patterns without noise.

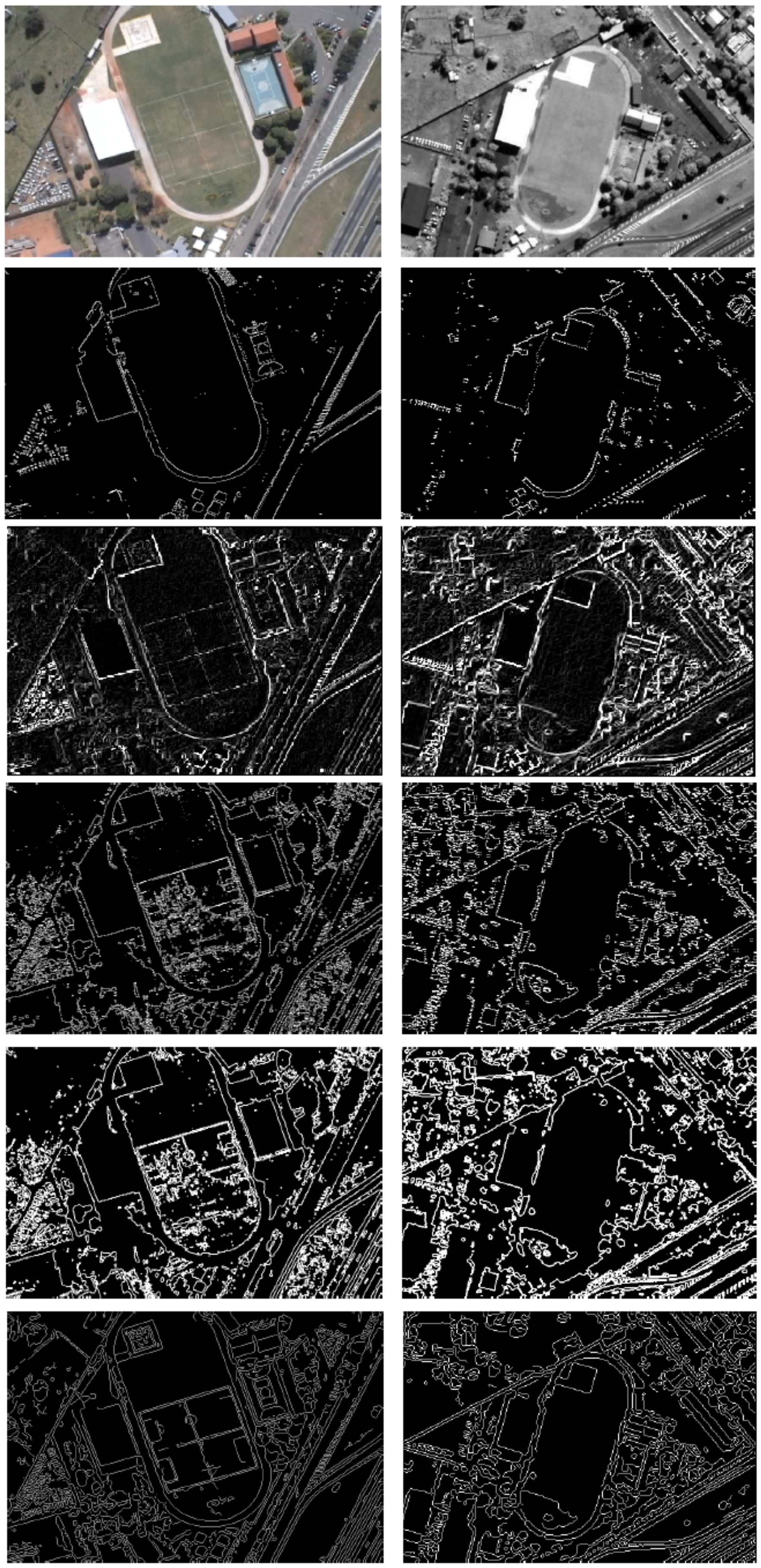

Figure 4 – Up to down: Edges extracted by: CNN, Sobel, MLP, RBF and Canny. Right: aerial image and left satellite image.

Following are the highlights of the work presented in this paper:

- The use of patterns without edges for training and a new form of association between patterns and desired output during training, as well as the activation of RBF and MLP. Associate a scalar to each edge pattern is the simple idea that allows that the output image of these ANNs is already in binary form, eliminating the process of thresholding (which is done by ANNs) which is necessary in classical edge detectors;
- Development of a methodology based on images, with the use of ANNs, for automatic position estimation of UAV. This methodology used the edge information for correlation and achieved better results with the use of CNN instead of classical edge detectors (Canny and Sobel).

Constant changes in ANNs with the aim of improving their results, such as: training parameters and activation, architecture, form and amount of training patterns, among others, are some future work as also research for the use of other ways to use information from the imagery of the area flown by the UAV to be able to estimate their geographical position. The implementation of the methodology in high-speed devices for processing, to meet applications that require real-time response also can be development.

6 Acknowledgments

This work was funded by CNPq and project PITER-IEAv.

7 References

[1] J. Zhang and W. Liu, Vision-based uav navigation using 3d gis data, B. Vasudev, T. R. Hsing, A. G. Tescher, and T. Ebrahimi, Eds., vol.5022, no. 1. **SPIE**, (2003), pp. 910–920.

[2] G. Conte and P. Doherty, An integrated uav navigation system based on aerial image matching, **in Aerospace Conference. IEEE**, (2008), pp. 1–10.

[3] S. Lee, S. Yoon, H. J. Kim, and Y. Kim, Wireless stereo vision system development for rotary-wing uav guidance and control, **3rd International Conference on Sensing Technology**, pp. 168–173, (2008).

[4] A. Nemra and N. Aouf, Robust feature extraction and correspondence for uav map building, **Mediterranean Conference on Control and Automation**, vol. 0, pp. 922–927, (2009).

[5] H. Chao, Y. Cao, and Y. Chen, Autopilots for small fixed-wing unmanned air vehicles: A survey, **in Mechatronics and Automation, 2007**. ICMA 2007. International Conference on, (2007), pp. 3144–3149.

[6] D. L. Fitzgerald, Landing site selection for uav forced landings using machine vision, Ph.D. dissertation, **Queensland University of Technology**, Brisbane, Australia, (2007).

[7] E. H. Shiguemori, M. V. T. Monteiro, and M. P. Martins, Landmarks recognition for autonomous aerial navigation by neural networks and gabor transform, in Image Processing: Algorithms and Systems V, vol. 6497. San Jose, California, EUA: **IS&T/SPIE 19th Annual Symposium Electronic Imaging Science and Technology,** (2007).

[8] K. Shimonomura and T. Yagi, Neuromorphic vlsi vision system for real-time texture segregation, **Neural Networks**, vol. 21, no. 8, pp. 1197 – 1204, (2008), neuroinformatics.

[9] N. M. Nasrabadi and C. Y. Choo, Hopfield network for stereo vision correspondence, **IEEE Transactions on Neural Networks**, vol. 3, no. 1, pp. 5–13, Janeiro (1992).

[10] A. P. A. de Castro, Detecção de bordas e navegação autônoma utilizando redes neurais artificiais, Master's thesis, **INPE**, São José dos Campos, Brazil, (2003).

[11] A. Schultz, C. Rekeczky, I. Szatmari, T. Roskaf, and L. . Chua, Spatio-temporal cnn algorithm for object segmentation and object recognition, **Fifth IEEE International Workshop on Cellular Neural Networks and their Applications**, London, United King, pp. 347–352, (1998).

[12] I. N.Aizenberg, N. N.Aizenberg, and J. Vandewalle, Precise edge detection: Representation by boolean functions, implementation on the cnn, **Fifth IEEE International Workshop on Cellular Neural Networks and their Applications**, London, United King, pp. 301–306, (1998).

[13] M. Ursino and G. E. La Cara, A model of contextual interactions and contour detection in primary visual cortex, **Neural Networks**, vol. 17, no. 5-6, pp. 719–735, (2004).

[14] T. Kubota, Massively parallel networks for edge localization and contour integration - adaptable relaxation approach, in **Neural Networks**, vol. 17, no. 3. Oxford, United King: Elsevier, (2004), pp. 411–425.

[15] S. Haykin, Neural Networks: A Comprehensive Foundation. Upper Saddle River, NJ, USA: **Prentice Hall PTR**, (1998).

[16] L. O. Chua and L. Yang, Cellular neural networks: Theory, **IEEE Transactions on Systems and Circuits**, vol. 35, no. 10, pp. 1257–1272, (1988).

[17] L. Corrêa, A. Delbem, and Z. Liang, Associative memories using cellular neural networks, **in Intelligent Systems Design and Applications**, 2007. ISDA 2007. Seventh International Conference on, (2007), pp. 539–544.

[18] D. Liu and Z. Lu, A new synthesis approach for feedback neural networks based on the perceptron training algorithm, **IEEE transactions on Neural Networks**, vol. 8, no. 6, pp. 1468–1492, (1997).

[19] R. C. Gonzalez and R. E. Woods, Digital Image Processing. Boston, MA, USA: **Addison-Wesley Longman Publishing Co., Inc.**, (1992).

[20] J. Canny, A computational approach to edge detection, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, vol. 8, no. 6, pp. 679–698, (1986).