

Real Time Runway Detection in Satellite Images Using Multi-Channel PCNN

Hualiang Zhuang

College of Information Engineering,

Zhejiang University of Technology, Hangzhou, China 310023

zhuanghl@zjut.edu.cn

Kay Soon Low

School of EEE, Nanyang Technological University

Singapore 639798

ekslow@ntu.edu.sg

Abstract—This paper proposes a pulse coupled neural network with multi-channel (MPCNN) linking and feeding fields for multispectral image processing. Different from the conventional PCNN, pulse based RBF units are introduced into the model neurons of PCNN to determine the fast links among neurons with respect to their spectral feature vectors and spatial proximity. This MPCNN can be implemented in parallel on a FPGA chip to perform real-time image segmentation and edge detection. Based on the output of the neural circuits, modified Hough Transform and landmark feature extraction algorithm is designed to perform airport runway detection in satellite images. Experimental results show that the proposed parallel MPCNN circuits drastically improve the processing speed over the popular seeded region growing (SRG) algorithm for segmentation of RGB satellite images. Furthermore the detection accuracy of the proposed scheme remains competitive.

Keywords — *Pulse coupled neural network; satellite image; runway detection; seeded region growing; radial basis function neural network; field programmable gate array*

I. INTRODUCTION

State-of-the-art satellite remote sensing system, such as IKONOS, can produce images in 1-metre panchromatic and 4-meter multispectral resolution or even higher resolution. This provides valuable data for ground feature analysis. In recent years, research works based on satellite images have been reported in various applications [1-7]. Amongst these applications, airfield runway detection is a popular topic [2-7] due to its importance for modern aviation system and challenging difficulty in processing method. In [4], Adaboost learning algorithm is employed to detect airport runways in satellite images gathered from GoogleEarth™. Textural features of data samples are extracted for training of Adaboost so that targeting runway can be detected by the classifier. It is a tedious work to manually label large amount of training data for offline training. In [5], the Kernel Matching Pursuits is used for segmentation to detect rectangle shape of runway in regions of interest using a texture based prescreening. Different from this offline data-training methodology, some researchers manage to speed-up the processing. In [6], a classical segmentation based on image intensity threshold is performed before shape analysis so that high-resolution images can be processed faster. In [7], real-time runway detection is implemented by using simplified Hough Transform for collinear feature extraction based on the result of edge detection. Nonetheless its sequential processing based edge tracking is still time-consuming. In recent years,

high-resolution terrain data has become available for commercial use. To utilize this new resource for automation of aviation systems such as auto targeting and auto landing, the bottleneck of processing time needs to be overcome. In this paper, we propose a new approach that enhances a pulse coupled neural network (PCNN) model and implemented it in parallel circuits for real time runway detection based on commercially available satellite images.

Conventional PCNN is a bio-inspired neural computing model [8]. Typically, it is a two dimensional (2-D) array of neurons, where each neuron maps to a pixel in the 2-D image under processing. It has been successfully applied to region-based image segmentation and related edge detection [8, 9]. In this research, there are two issues to be resolved. Firstly, although the conventional PCNN is a parallel algorithm, its implementation is difficult due to its complex dynamics and multiple inter-neuron couplings [8]. Secondly, the typical PCNN model is confined to grayscale images. This is because the essence of its inter-neuron coupling is defined in terms of products of scalars. In contrast, the pixels of multispectral images are featured by vectors.

In this research, we propose a multi-channel PCNN that overcomes the aforementioned issues. Firstly, the dynamics and complex couplings in the original PCNN are simplified to the level suitable for digital platform such as FPGA. This is possible as most of the original dynamics and complex couplings are related to PCNN's functionality on feature analysis only. For the segmentation-oriented algorithm, they are not required and can be simplified [9]. Secondly, the proposed approach is designed as a vector-wise model to deal with multispectral images. Different from the conventional PCNN, it introduces a pulse based radial basis function (RBF) into the model neurons of PCNN to work as an internal computation unit. It uses the timing and competitive effect of pulses produced by the neurons to compute the distances between pixel's feature vectors and their rank order respectively. Then the proposed algorithm will utilize this information to determine the fast links among neurons with respect to the spectral feature vectors and spatial proximity of mapped pixels. The RBF unit's pulsed behavior is designed to be compatible with PCNN's pulsed behavior so that the algorithm can naturally deal with multispectral images whose pixels are presented in vectors instead of scalars. We name this PCNN as multi-channel PCNN (MPCNN). This MPCNN performs segmentation and edge detection so that collinear features of the potential regions can be extracted. As MPCNN algorithm is parallel-oriented, the proposed algorithm can be implemented in parallel on VLSI (e.g. FPGA) to improve the processing speed. In our experiment, it is observed that the

processing speed of segmentation phase is improved by hundreds times as compared to sequential-code-based approaches.

The MPCNN outputs the resultant grouped edges of segmented regions. This provides the convenience for detecting collinear feature of each piece of edge individually. A modified Hough Transform [10] is employed to detect the parallel straight runway boundaries which are selected as key feature of a runway. A simplified searching scheme is designed to reduce the computation of Hough Transform. Other runway features, such as centerline strips are also used for recognition in the final confirmation. Experimental results show that the detection accuracy of the proposed scheme has an improvement of 10% as compared with the SRG scheme in both single and multiple runways.

II. Runway features

A. Boundaries of Runway

For runway detection, their straight boundaries can be selected as the primary indicators. Most of the boundaries of runways are explicitly painted to direct the airplanes. For those runways that have no painted boundary, the distinct long and straight edges of a runway are not difficult to be detected. Nonetheless, a long and straight strip may not be a runway as it could be a highway or a long dam. Hence, additional landmarks are required for runway identification.

B. Landmark Feature

For most runways, pairs of parallel bars and dashed centerline stripes are usually painted on the runway to guide airplanes to locate the touchdown point and to determine the gliding direction. Some examples are showed in Fig. 1. Amongst these landmarks, centerline stripes are more reliable as critical feature of runway because they cover almost the entire length of the runway that decreases the probability of misleading detection. In contrast, those pairs of parallel bars are apt to be missed due to ambient disturbance. Examples in Fig. 1(b) & (c) illustrate this point. Therefore the centerline stripes are used as the secondary landmark feature of the runway.

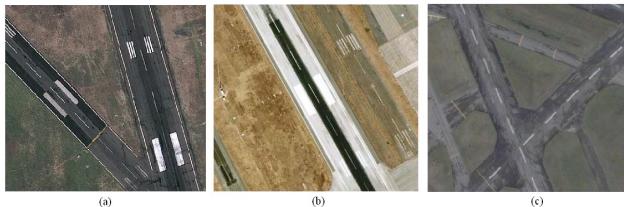


Fig. 1. Landmark features of runways.

III. MPCNN FOR MULTISPECTRAL IMAGE PROCESSING

A. Multi-Channel PCNN

Similar to conventional PCNN, the proposed network is a 2D array of neurons as shown in Fig. 2. Each neuron maps one pixel of the corresponding image. The bottom box in Fig. 2 illustrates the model of a single neuron. To be consistent with the conventional PCNN, the input fields are classified as linking and feeding fields. Here the feeding is defined as the

spectral feature vector of the current pixel. To establish coupling with the neighborhoods, each neuron has linking field for receiving spectral linking and pulse linking signals from its neighboring neurons. One of the key differences between the proposed and conventional PCNN is that the later utilizes the pixel intensity to implement the coupling of pulses while the proposed PCNN uses the feature vectors.

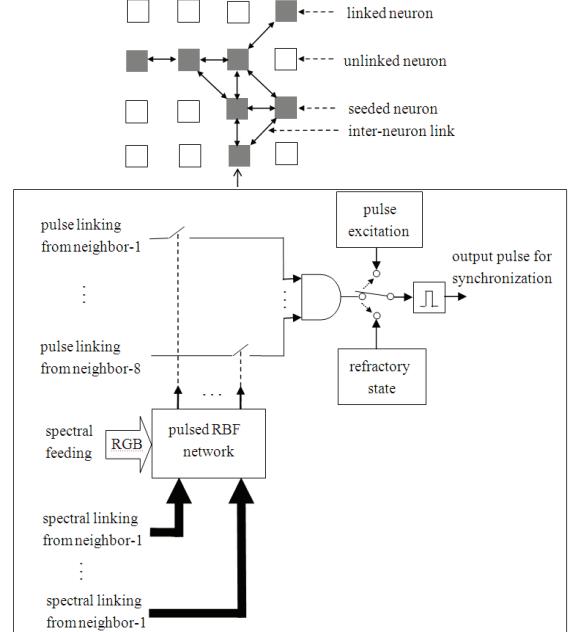


Fig. 2. Multi-channel PCNN with 2D array of neurons.

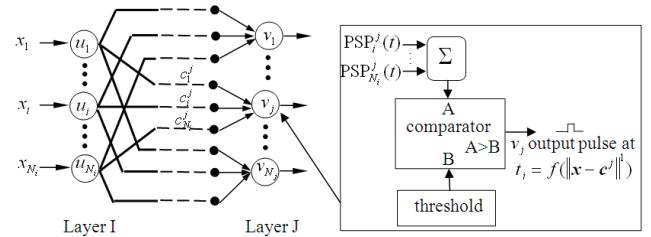


Fig. 3. Pulsed RBF neural model.

To establish vector-oriented coupling, pulse-based RBF models are introduced as shown in Fig. 3 to compute and rank the distances between the current pixel's feature vector, $\mathbf{x} = [x_1, \dots, x_i, \dots, x_{N_i}]^T$, and the linking spectral vectors, $\mathbf{c}^j = [c_1^j, \dots, c_i^j, \dots, c_{N_j}^j]^T$, $j = 1, \dots, N_j$. The linking vectors are from the neighboring pixels and they are stored in the synapses of the RBF units. Here j is the index of neighboring pixels. As we only consider the coupling between the current and its 8 neighborhoods' pixels, $N_j = 8$. N_i denotes the dimension of

the feature vector. Moreover, we can specify $\mathbf{x} = [R, G, B]^T$ as we only use RGB color images in this study. These distances determines the starting time an individual postsynaptic potential (PSP) is excited. More specifically, the PSP activated between neuron u_i and v_j is defined as a function of time t as follows,

$$PSP_i^j(t) = t - |x_i - c_i^j|, \quad PSP_i^j \geq 0. \quad (1)$$

The integrated potential of neuron v_j is computed as

$$s_j(t) = \sum_{i=1}^{N_i} (t - |x_i - c_i^j|). \quad (2)$$

Assume the neuron v_j emits an output pulse at time t_j when $s_j(t)$ reaches a predefined threshold θ , then t_j can be derived by substituting θ into (2) as

$$t_j = \frac{\theta}{N_i} + \frac{1}{N_i} \sum_{i=1}^{N_i} |x_i - c_i^j|. \quad (3)$$

In (3), t_j reflects a RBF of the Manhattan distance between x and c^j . The output neuron v_j is of the integrate-and-fire (IAF) type [11]. When its integrated potential $s_j(t)$ reaches the threshold θ , an output pulse emits as illustrated in the inset of Fig 3. The latency of a pulse quantifies the distance between related synaptic vector and the input vector. These temporal output pulses also indicate the rank order of respective distances, and will be utilized in the segmentation algorithm later.

As shown in Fig. 2, we simplify the conventional *fast links* as hard-wired connections of pulse linking fields. These *fast links* can be established among neurons according to a criterion related to inter-pixel distance operated by the pulsed RBF, which reflects the similarity of their spectral feature. After all the links are established, the representative neuron of each linked region will be excited sequentially. This will cause all neurons in the linked region to emit output pulses synchronously. A neuron will be forced into refractory state right after it emits an output pulse. It will be recovered to normal state after all neurons of the network get into the refractory state. This mechanism provides the basis for image segmentation by linking homogenous regions. An algorithm will be proposed for this PCNN to perform image segmentation in the ensuing subsection.

A. Algorithm for Multispectral Image Segmentation

In this algorithm, a pixel will interact with its 8 neighbor pixels to establish inter-pixel coupling according to their spectral similarity. In the spectral feature space, the pixel is named a *seeded pixel* if the second minimum distance to its 8 neighbors is less than a predefined threshold. The corresponding neuron is named a *seeded neuron*. To avoid equivocality, we assume that the neighbor pixel with minimum distance is unique. Then pixel s is a *seeded pixel* if

$$\mu_s \leq \theta_\mu \quad (4)$$

where $\mu_s \in \{\|x^s - c^j\| \mid j \neq \arg(\min_{j=1,\dots,8} (\|x^s - c^j\|))\}$, and $\|\cdot\|$ is an operator of Manhattan distance, and θ_μ is a predefined threshold value, and x^s is the color vector of pixel s . This definition means a *seeded pixel* has at least two similar-feature neighborhoods. θ_μ determines the degree of the similarity. Utilizing the timing of output pulse, the RBF circuits described in Fig. 3 provide a convenient computation for rank order of

distances between a pixel and its neighborhoods. That means the minimum value implied in (4) can be quantified by the timing of the earliest output pulse of the current RBF. Any output pulse later (non-earlier) than the earliest pulse can be utilized to calculate the inequality (4). To perform image segmentation, any neighbor pixel fulfills the inequality (4) will be linked to the seed so that seeded regions can be expanded by capturing pixels with similar features. This prevents the edges from blurring so that detailed edge can be kept while the capturing effect smoothes the local area despite the presence of noise. Finally, resultant segmentation can be achieved. The segmentation algorithms are listed as follows:

ALGORITHM

Step 1.

For all enabled pixels, determine the *seeded pixels* using (4). The *fast links* will be established between the *seeded neuron* and its neighborhoods as illustrated in Fig. 1. Neurons that link with a seeded neuron are labeled as *linked neurons*. In this step, some groups of linked neurons are determined.

Step 2.

Compute a mean feature vector $\sigma^g = [\sigma_1^g, \dots, \sigma_n^g]^T$ for each group of linked neurons as follows.

$$\sigma_i^g = \frac{1}{M} \sum_{j=1}^M x_{ij}^g \quad (5)$$

where g is the group index, j is the pixel index, and M is the total number of pixels in the group. Replace the feature vectors of the linked neurons using the mean feature vector of its group.

Step 3.

For any unlinked neuron u that has *linked neighbor neurons*, a link will be established with the linked neighborhood, neuron j_u whose group's mean feature vector is the nearest to this unlinked neuron, i.e.

$$j_u = \arg(\min_j \|x^u - c^j\|) \quad (6)$$

The resulting neuron in (6) can be determined by the pulsed RBF since its earliest output pulse indicates the corresponding neighbor neuron j as shown in Fig. 3. It is termed as a *linked neuron*.

Step 4.

If there is any unlinked neuron, go to Step 2. Otherwise, the resultant groups of linked neurons represent all the segments of this image.

Groups of edges of the segmented regions are then extracted for further processing.

IV. RUNWAY RECOGNITION BASED ON RESULTANT POTENTIAL REGIONS

The segmentation results in section III need to be analyzed to verify whether the potential regions are real runways or just misleading segments. Here the patterns of the straight boundaries and the centerline stripes of the runway are selected as the critical features for the judgment.

A. Size of Window

The dimensions of most runways are less than 5000-meter long and 80-meter wide. In this study, the size of processing

window is selected as $400\text{m} \times 400\text{m}$. When a larger area is scanned, 100m overlap margin should be given between successive windows to guarantee the coverage of 80m wide section.

B . Collinear Edge Detection and Landmark Recognition

In general, the surrounding of runways is spatial and flat for flying safety. In the top-bird-view, it appears in large segments. This reduces the number of segments in the region of interest. Based on the edges of the obtained segments, collinear edges of remaining segments will be extracted for recognition of runway boundaries. In this subsection, we will simplify the well-known Hough transform (HT) so that the computation is fast enough for real time collinear edge detection.

The original HT has to scan all the lines ranging from 0 to 180-degree orientations, which pass through each significant pixel. This is because it needs to rank the accumulated number of pixels on each labeled line to determine the existence of collinear feature. Here, the resource-consuming computation can be reduced as follows.

In section III, the edges of all segmented regions are grouped distinctly. That means each piece of edge is a set of connected pixels. This provides the convenience for performing HT to each piece of edge individually. This is the first factor for the computation reduction. Consider the geometric relationship of a pixel with its neighborhoods on the connected edge of a region. If the pixel and its nearest neighbor pixels are on a straight line, the angular range they spanned is confined in a narrow range. In this case, we can derive that the orientation range of HT can be confined to a range of $45^\circ (\pi/4)$ instead of 180° . This is the second factor for the computation reduction. To further reduce the computation, we utilize hierarchical searching scheme, where coarse and fine scans are performed hierarchically. The coarse scan uses $\pi/20$ as angular incremental while the fine scan is conducted by angular incremental of $\pi/360$ within a range of $\pi/20$ confined by the coarse scan.

Once a pair of straight lines is identified as the boundaries of a potential runway, the strip between this pair of lines will be analyzed to detect the runway centerline strips based on their simplex geometric feature. If this feature is detected, it will be confirmed as a runway.

V. EXPERIMENTAL RESULTS

As compared with the conventional methods, the salient feature of MPCNN is its ease of implementation in digital circuits. For this study, it has been implemented using a Spatan-3A FPGA (XC3SD3400A -4FGG676C) with 3.4 millions of logic gates running at a clock rate of 250MHz. For this investigation, it is designed for a 400×400 -pixel image. It is found that 87% of the FPGA resources, i.e. about 2.958 millions of logic gates have been used for the segmentation and edge detection phase. For the Hough Transform and the post feature analysis, they are implemented on a host PC connecting the FPGA through a USB link.

In the following experiment, the performance and processing speed of the segmentation based on the MPCNN are benchmarked with the conventional SRG method [12].

Moreover, the overall runway detection scheme is evaluated using various images. The effect of auxiliary landmark analysis is also discussed. For brevity, the pre-processing is not in the scope of this discussion.

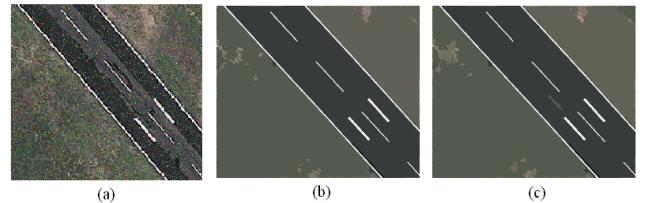


Fig. 4. Segmentation of noisy image. (a) noisy image; (b) result of SRG; (c) result of MPCNN.

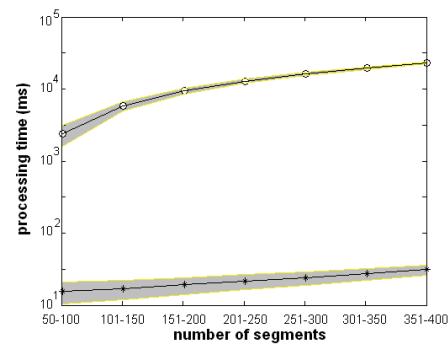


Fig. 5. Comparison of processing time between SRG and MPCNN with respect to various numbers of resultant segments. 'o' for average time of SRG; '*' for average time of MPCNN. Gray ribbons cover the span of all testing cases.

A. Evaluation on Segmentation Phase

In section III-B, it is mentioned that the segmentation based on MPCNN belong to the methodology of seeded region growing. Results of an example image are given in Fig. 4 to compare with the performance of conventional SRG for processing an image with salt-and-pepper noise with a density of 0.05. It is observed that the MPCNN successfully eliminates the noise similar to what the conventional SRG does.

Fig. 5 compares their processing speed with respect to the resultant number of segments. The conventional SRG algorithm is coded in Visual C++ and implemented on a Pentium4 computer at 3GHz, 12 times of the clock rate of the proposed FPGA. From Fig. 5, we can observe that both methods have linear-like trend. Moreover, the MPCNN shows a far swifter processing speed. As compared with SRG, the increase in processing time of the proposed method is far slower with the increase in number of image segments. For example, the average processing time of SRG is 2.37 seconds for the image with a range of 50 to 100 segments. This is increased by nearly 10 times to 23.2 seconds for the case of 351 to 400 segments. In contrast, the processing time of MPCNN is doubled from 15.6 ms to 31.1 ms.

B. Performance of Overall Scheme

Most of the literatures consider the whole airport as the ultimate objective of the detection scheme except [3, 4]. Thus they are selected as benchmarking counterparts to evaluate the accuracy of the runway detection algorithm. In the experiment, 50 images with single runway and 50 images with multiple

runways are gathered from the database of GoogleEarth™. They are acquired from the top-bird-view of various airports all over the world. The sizes of the processing windows are 400×400 with 1 meter spatial resolution. The spectral bands are RGB. Table 1 summarizes the average accuracy of the benchmarks and the proposed method for the 100 images. It is observed that the detection accuracy of the proposed method is more competitive than the methods in [3]. As compared with the method in [4], although its true negative accuracy is a little less competitive, the true positive accuracy of the proposed method is still competitive. Fig. 6 shows the results of various examples, including single and multiple runways as well as runway with entrance from airplane parks. In all cases, the runways are successfully detected based on the clearly extracted features, i.e. parallel straight boundaries with a series of centerline stripes in between.

Table 1. Accuracies of runway detection of various methods.

	True positive accuracy	True negative accuracy
method in [3]	76%	80%
method in [4]	83%	91%
proposed method	88%	90%

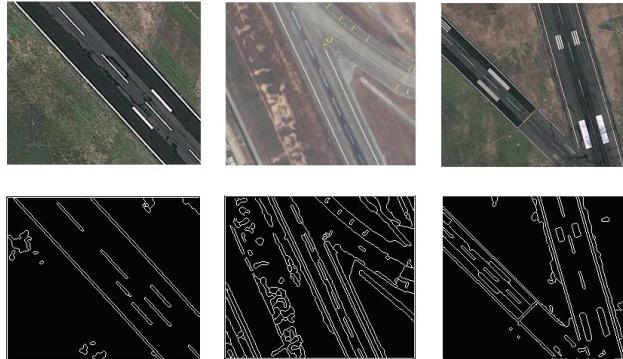


Fig. 6. Runway detection in various cases. Upper row for original images; lower row for results.

overall runway detection system also outperforms recent reported literatures.

REFERENCES

- [1] T. Celik, “Unsupervised change detection in satellite images using principal component analysis and k-means clustering”, *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 4, pp. 772-776, 2009.
- [2] C. Tao, Y. Tan, H. Cai and J. Tian, “Airport detection from large IKONOS images using clustered SIFT keypoints and region information”, *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 1, pp. 128-132, 2011.
- [3] J. H. Horne, “Fully automated airfield feature extraction and mapping”, *Proceedings of ASPRS 2004 Conference*, pp. 23-28, Denver, Colorado, May, 2004.
- [4] U.Zongur, U. Halici O. Aytekin and I. Ulusoy, “Airport runway detection in satellite images by Adaboost learning”, *Image and Signal Processing for Remote Sensing, Proceedings of SPIE*, vol. 7477,747708, 2009.
- [5] D. Liu and L. Carin, “Airport detection in large aerial optical imagery”, *Acoustics Speech and Signal Processing 2004 Proceedings (ICASSP'04) IEEE International Conference*, pp. 761-764, 2004.
- [6] P. Gupta, and A. Agrawal, “Airport detection in high-resolution panchromatic satellite images”, *Journal of Institution of Engineers*, vol. 88, no. 5, pp. 3-9, 2007.
- [7] N. Di, M. Zhu and Y. Wang, “Real time method for airport runway detection in images”, in *Proceedings 2008 International Conference on Audio, Language and Image Processing*, pp. 563-567, July, 2008.
- [8] J. L. Johnson, “PCNN Models and Applications”, *IEEE Trans. on Neural Networks*, vol.10, no. 3, pp. 480-498, 1999.
- [9] G. Kuntimad, and H. S. Ranganath “Perfect Image Segmentation Using Pulse Coupled Neural Networks”, *IEEE Trans. on Neural Networks*, vol.10, no. 3, pp.591-598, 1999.
- [10] T. F. Chan, J. Shen, “Mathematical models for local non-texture inpaintings”, *Journal of Applied Mathematics*, vol. 62, pp. 1019-1043, 2002.
- [11] H.L. Zhuang, K. S. Low and W. Y. Yau, “A pulsed neural network with on-chip learning and its practical applications”, *IEEE Trans. on Industrial Electronics*, vol 54, no. 1, pp. 34-42, 2007.
- [12] F.Y. Shih and S. Cheng, “Automatic seeded region growing for color image segmentation,” *Image Vision Computing*, vol. 23, no. 10, pp. 877-886, 2005.

VI. CONCLUSION

In this paper, a PCNN with multi-channel linking and feeding fields is proposed for multispectral image processing. Different from the conventional PCNN, pulse based RBF units are introduced into the model neurons of PCNN to determine the fast links among neurons with respect to their spectral feature vectors and spatial proximity. These IAF (integrate-and-fire) RBF units utilize the timing of individual pulse produced by the neurons to reflect the rank order of distance between pixels’ feature vector so that seeded pixels are determined and homogenous pixels are linked. Experimental results show that the performance of the proposed MPCNN is comparable to SRG algorithm for segmentation of noisy images. Implemented in a FPGA with parallelism, the MPCNN improve the speed of segmentation drastically as compared with its sequential counterpart. This allows its use for real-time runway detection, which is a critical technique for aviation automation. The