Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2013, Article ID 617618, 7 pages
http://dx.doi.org/10.1155/2013/617618

Research Article

Recursive Neural Networks Based on PSO for Image Parsing

Guo-Rong Cai and Shui-Li Chen

School of Sciences, Jimei University, Xiamen, China

Correspondence should be addressed to Shui-Li Chen; sgzx@jmu.edu.cn

Received 24 February 2013; Accepted 3 March 2013

Academic Editor: Zhenkun Huang

This paper presents an image parsing algorithm which is based on Particle Swarm Optimization (PSO) and Recursive Neural Networks (RNNs). State-of-the-art method such as traditional RNN-based parsing strategy uses L-BFGS over the complete data for learning the parameters. However, this could cause problems due to the nondifferentiable objective function. In order to solve this problem, the PSO algorithm has been employed to tune the weights of RNN for minimizing the objective. Experimental results obtained on the Stanford background dataset show that our PSO-based training algorithm outperforms traditional RNN, Pixel CRF, region-based energy, simultaneous MRF, and superpixel MRF.

1. Introduction

Image parsing is an important step towards understanding an image, which is to perform a full-scene labeling. The task of image parsing consists in labeling every pixel in the image with the category of the object it belongs to. After a perfect image parsing, every region and every object are delineated and tagged [1]. Image parsing is frequently used in a wide variety of tasks including parsing scene [2, 3], aerial image [4], and facade [5].

During the past decade, the image parsing technique has undergone rapid development. Some methods for this task such as [6] rely on a global descriptor which can do very well for classifying scenes into broad categories. However, these approaches fail to gain a deeper understanding of the objects in the scene. Many other methods rely on CRFs [7], MRFs [8], or other types of graphical models [9, 10] to ensure the consistency of the labeling and to account for context. Also, there are many approaches for image annotation and semantic segmentation of objects into regions [11]. Note that most of the graphical-based methods rely on a presegmentation into superpixels or other segment candidates and extract features and categories from individual segments and from various combinations of neighboring segments. The graphical model inference pulls out the most consistent set of segments which covers the image [1]. Recently, these ideas have been combined to provide more detailed scene understanding [12–15].

It is well known that many graphical methods are based on neural networks. The main reason is that neural networks have promising potential for tasks of classification, associative memory, parallel computation, and solving optimization problems [16]. In 2011, Socher et al. proposed a RNN-based parsing algorithm that aggregates segments in a greedy strategy using a trained scoring function [17]. It recursively merges pairs of segments into supersegments in a semantically and structurally coherent way. The main contribution of the approach is that the feature vector of the combination of two segments is computed from the feature vectors of the individual segments through a trainable function. Experimental results on Stanford background dataset revealed that RNN-based method outperforms state-of-the-art approaches in segmentation, annotation, and scene classification. That being said, it is worth noting that the objective function is nondifferentiable due to the hinge loss. This could cause problems since one of the principles of L-BFGS, which is employed as the training algorithm in RNN, is that the objective should be differentiable.

Since Particle Swarm Optimization (PSO) [18] has proven to be an efficient and powerful problem-solving strategy, we use a novel nonlinear PSO [19] to tune the weights of RNN. The main idea is to use particle swarm for searching good

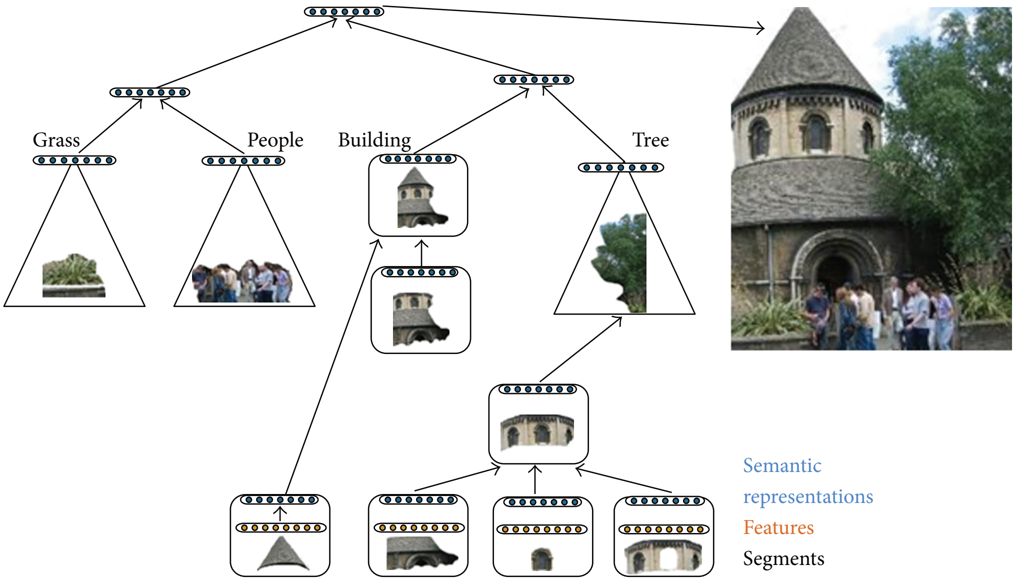


Figure 1: Hierarchical architecture of image parsing based on recursive neural network.

combination of weights to minimize the objective function. The experimental results show that the proposed algorithm has better performance than traditional RNN on Stanford background dataset.

The rest of the paper is organized as follows: Section 2 provides a brief description of the RNN-based image parsing algorithm. Section 3 describes how PSO and the proposed algorithm work. Section 4 presents the dataset and the experimental results. Section 5 draws conclusions.

2. Image Parsing Based on Recursive Neural Networks

The main idea behind recursive neural networks for image parsing lies in that images are oversegmented into small regions and each segment has a vision feature. These features are then mapped into a "semantic" space using a recursive neural network. Figure 1 outlines the approach for RNN-based image parsing method. Note that the RNN computes (i) a score that is higher when neighboring regions should be merged into a larger region, (ii) a new semantic feature representation for this larger region, and (iii) its class label. After regions with the same object label are merged, neighboring objects are merged to form the full scene image. These merging decisions implicitly define a tree structure in which each node has associated with the RNN outputs (i)–(iii), and higher nodes represent increasingly larger elements of the image. Details of the algorithm are given from Sections 2.1 to 2.3.

2.1. Input Representation of Scene Images. Firstly, an image x is oversegmented into superpixels (also called segments) using the algorithm from [20]. Secondly, for each segment, compute 119 features via [10]. These features include color and texture features, boosted pixel classifier scores (trained on the labeled training data), and appearance and shape features. Thirdly, a simple neural network layer has been used to map these features into the "semantic" n-dimensional space in which the RNN operates, given as follows.

Let F_i be the features described previously for each segment, where $i = 1, \ldots, N_{\text{segs}}$ and N_{segs} denotes the number of segments in an image. Then the representation is given as

$$a_i = f\left(W^{\text{sem}} F_i + b^{\text{sem}}\right), \quad (1)$$

where $W^{\text{sem}} \in R^{n \times 119}$ is the matrix of parameters we want to learn, b^{sem} is the bias, and f is applied element wise and can be any sigmoid-like function. In [17], the original sigmoid is function $f(x) = 1/(1 + e^{-x})$ (Figure 2).

2.2. Greedy Structure Predicting. Since there are more than exponentially many possible parse trees and no efficient dynamic programming algorithms for RNN setting, therefore, Socher recommended a greedy strategy. The algorithm finds the pairs of neighboring segments and adds their activations to a set of potential child node pairs. Then the network computes the potential parent representation for these possible child nodes:

$$p(i, j) = f\left(W\left[c_i; c_j\right] + b\right). \quad (2)$$

With this representation, a local score can be determined by using a simple inner product with a row vector $W^{\text{score}} \in R^{1 \times n}$:

$$s(i, j) = W^{\text{score}} p(i, j). \quad (3)$$

As illustrated in Figure 3, the recursive neural network is different from the original RNN in that it predicts a score for being a correct merging decision. The process repeats until all pairs are merged and only one parent activation is left, as

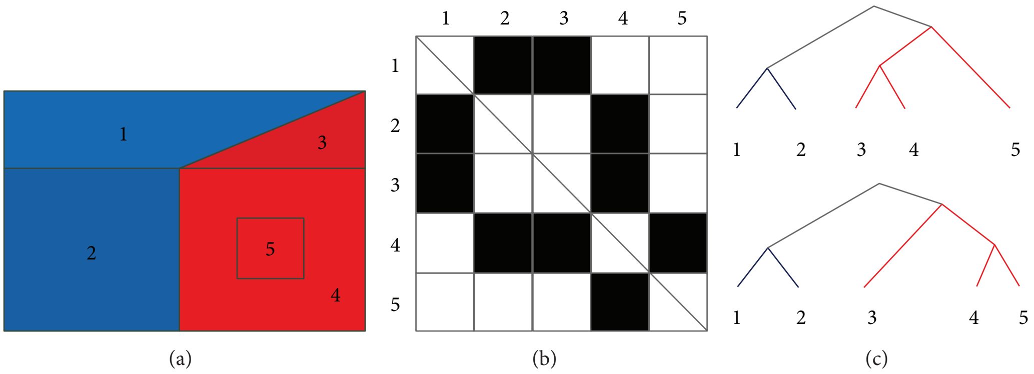


Figure 2: Illustration of the RNN training inputs: (a) a training image (red and blue are differently labeled regions). (b) An adjacency matrix of image segments. (c) A set of correct trees which is oblivious to the order in which segments with the same label are merged.

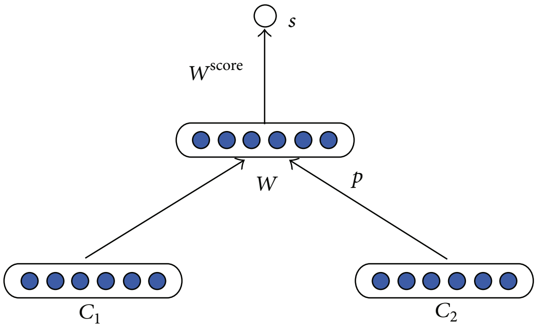


Figure 3: Recursive neural network which is replicated for each pair of input vectors.

shown in Figure 1. The final score that we need for structure prediction is simply the sum of all the local decisions:

$$s\left(\text{RNN}\left(\theta, x_i, \hat{y}\right)\right) = \sum_{d \in N(\hat{y})} s_d, \tag{4}$$

where θ are all the parameters needed to compute a score s with an RNN, $\hat{y}$ is a parse for input x_i, and $N(\hat{y})$ is the set of nonterminal nodes.

2.3. Category Classifiers in the Tree. The main advantage of the algorithm is that each node of the tree built by the RNN has associated with it a distributed feature representation. To predict class labels, a simple softmax layer is added to each RNN parent node, as shown later:

$$\text{label}_p = \text{softmax}\left(W^{\text{label}} p\right). \tag{5}$$

When minimizing the cross-entropy error of this softmax layer, the error will backpropagate and influence the RNN parameters.

3. Nonlinear Particle Swarm Optimization for Training FNN

As for traditional RNN-based method, the objective J of (5) is not differentiable due to the hinge loss. For training RNN, Socher used L-BFGS over the complete training data to minimize the objective, where the iteration of the swarm relates to the update of the parameters of RNN. That being said, it is worth noting that the basic principle of L-BFGS is that the objective function should be differentiable. Since the objective function for RNN is nondifferentiable, L-BFGS could cause problems for computing the weights of RNN. To solve this problem, a novel nonlinear PSO (NPSO) has been used to tune the parameters of RNN.

3.1. Nonlinear Particle Swarm Optimization. As a population-based evolutionary algorithm, PSO is initialized with a population of candidate solutions. The activities of the population are guided by some behavior rules. For example, let $X_i(t) = (x_{i1}(t), x_{i2}(t), \ldots, x_{iD}(t))$ $(x_{id}(t) \in [-x_{d\max}, x_{d\max}])$ be the location of the ith particle in the tth generation, where $x_{d\max}$ is the boundary of the dth search space for a given problem and $d = 1, \ldots, D$. The location of the best fitness achieved so far by the ith particle is denoted as $p_i(t)$ and the index of the global best fitness by the whole population as $p_g(t)$. The velocity of ith particle is $V_i(t) = (v_{i1}(t), v_{i2}(t), \ldots, v_{iD}(t))$, where v_{id} is in $[-v_{d\max}, v_{d\max}]$ and $v_{d\max}$ is the maximal speed of dth dimension. The velocity and position update equations of the ith particle are given as follows:

$$\begin{aligned} v_{id}(t+1) &= w \cdot v_{id}(t) + c_1 r_1 \left(p_{id} - x_{id}(t)\right) \\ &\quad + c_2 r_2 \left(p_{gd} - x_{id}(t)\right), \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1), \end{aligned} \tag{6}$$

where $i = 1, \ldots, n$ and $d = 1, \ldots, D$. $w, c_1, c_2 \geq 0$. w is the inertia weight, c_1 and c_2 denote the acceleration coefficients, and r_1 and r_2 are random numbers, generated uniformly in the range $[0, 1]$.

Note that a suitable value for the inertia weight provides a balance between the global and local exploration abilities of the swarm. Based on the concept of decrease strategy, our nonlinear inertia weight strategy [19] chooses a lower value of w during the early iterations and maintains higher value

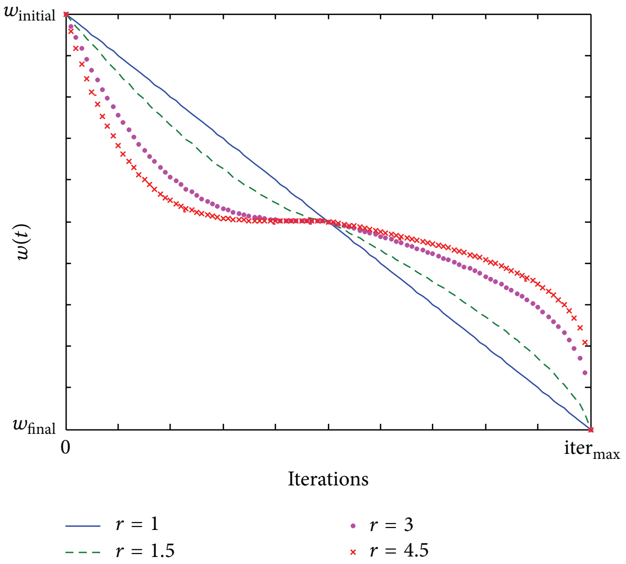


Figure 4: Nonlinear strategy of inertia weight.

of w than linear model [21]. This strategy enables particles to search the solution space more aggressively to look for "better areas", thus will avoid local optimum effectively.

The proposed update scheme of $w(t)$ is given as follows:

$$w(t) = \begin{cases} \left(1 - \dfrac{2t}{\text{iter}_{\max}}\right)^{r} \dfrac{(w_{\text{initial}} + w_{\text{final}})}{2} \\ \quad + \dfrac{(w_{\text{initial}} - w_{\text{final}})}{2}, & t \leq \dfrac{\text{iter}_{\max}}{2}, \\ \left(1 - \dfrac{2\left(t - (\text{iter}_{\max}/2)\right)}{\text{iter}_{\max}}\right)^{1/r} \dfrac{(w_{\text{initial}} - w_{\text{final}})}{2} \\ \quad + w_{\text{final}}, & t > \dfrac{\text{iter}_{\max}}{2}, \end{cases} \tag{7}$$

where $\text{iter}_{\max}$ is the maximum number of iterations, t denotes the iteration generation, and $r > 1$ is the nonlinear modulation index.

Figure 4 illustrates the variations of nonlinear inertia weight for different values of r. Note that $r = 1$ is equal to the linear model. In [19], we showed that a choice of r within [2-3] is normally satisfactory.

3.2. Encoding Strategy and Fitness Evaluation. Let $\theta = (W^{\text{sem}}; W; W^{\text{score}}; W^{\text{label}})$ be the set of RNN parameters; then each particle can be the expressed as the combination of all parameters, as shown later:

$$\begin{array}{|c|c|c|c|} \hline W^{\text{sem}} & W & W^{\text{score}} & W^{\text{label}} \\ \hline \end{array} \tag{8}$$

During the iteration, each particle relates to a combination of weights of neural networks. The goal is to minimize a fitness function, given as

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} r_i(\theta) + \frac{\lambda}{2} \|\theta\|^2, \tag{9}$$

where $r_i(\theta) = s(\text{RNN}(\theta, x_i, y^*)) + \Delta(x_i, l_i, y^*) - \max_{y_i \in Y(x_i, l_i)}(s(\text{RNN}(\theta, x_i, y_i)))$ and y^* denote the parse tree generated by the greedy strategy according to parameter θ. Minimizing this objective means minimize the error between the parsing results, which is generated by the best particle and the labeled training images (ground truth).

3.3. Summary of PSO-Based Training Algorithm.

Input includes a set of labeled images, the size of the hidden layer n, the value of penalization term for incorrect parsing decisions κ, the regularization parameter λ, the population of particles m, the values of nonlinear parameter r and the number of iterations $\text{iter}_{\max}$.

Output includes the set of model parameters $\theta = (W^{\text{sem}}, W, W^{\text{score}}, \text{and } W^{\text{label}})$, each with respect to weights of a recursive neural network.

(1) Randomly initialize m particles and randomize the positions and velocities for entire population. Record the global best location p_g of the population and the local best locations p_i of the ith particle according to (9), where $i = 1, 2, \ldots, m$.

(2) For each iteration, evaluate the fitness value of the ith particle through (9). If $(f(x_i)) < (f(p_i))$, set $p_i = x_i$ as the so far best position of the ith particle. If $(f(x_i)) < (f(p_g))$, set $p_g = x_i$ as the so far best position of the population.

(3) Calculate the inertia weight through (7). Update the position and velocity of particles according to (6).

(4) Repeat Step 2 and Step 3 until maximum number of generation.

(5) Compute the weights of RNN according to the best particle.

4. Experimental Results and Discussion

4.1. Description of the Experiments. In this section, PSO-based RNN method is compared with traditional RNN [17], pixel CRF [10], region-based energy [10], simultaneous MRF [8], and superpixel MRF [8], by using images from Stanford background dataset. All the experiments have been conducted on a computer with Intel sixteen-core processor 2.67 GHz processor and 32 GB RAM.

As for RNN, Socher recommends that the size of the hidden layer $n = 100$, the penalization term for incorrect parsing decisions $\kappa = 0.05$, and the regularization parameter $\lambda = 0.001$. As for the particle swarm optimization, we set

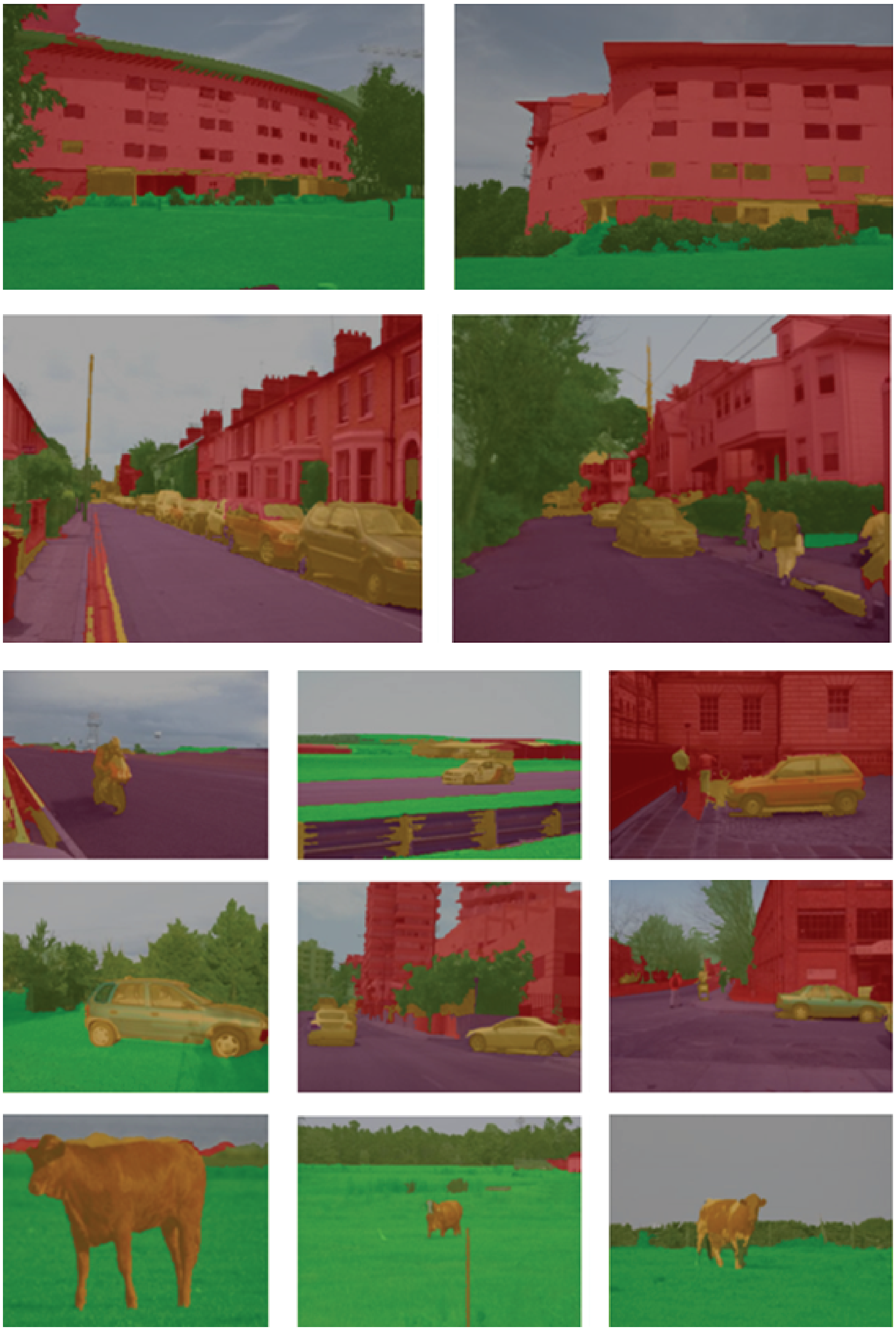


Figure 5: Typical results of multiclass image segmentation and pixel-wise labeling with PSO-based recursive neural networks.

the population of particles $m = 100$, the number of iterations $\text{iter}_{\max} = 500$, $c_1 = c_2 = 2$, $w_{\text{initial}} = 0.95$, $w_{\text{final}} = 0.4$, and $r = 2.5$.

4.2. Scene Annotation. The first experiment aims at evaluating the accuracy of scene annotation on the Stanford background dataset. Like [17], we run fivefold cross-validation and report pixel level accuracy in Table 1. Note that the traditional RNN model influences the leaf embeddings through backpropagation, while we use PSO to tune the weights of RNN. As for traditional RNN model, we label the superpixels by their most likely class based on the multinomial distribution from the softmax layer at the leaf nodes. One can see that in Table 1, our approach outperforms previous methods that report results on this data, which means that the PSO-based RNN constructs a promising strategy for scene annotation. Some typical parsing results are illustrated in Figure 5.

4.3. Scene Classification. As described in [17], the Stanford background dataset can be roughly categorized into three

Table 1: Accuracy of pixel accuracy of state-of-the-art methods on Stanford background dataset.

Method and semantic pixel accuracy in %
Pixel CRF, Gould et al. (2009) 74.3
Log. Regr. on superpixel features 75.9
Region-based energy, Gould et al. (2009) 76.4
Local labeling, Tighe and Lazebnik (2010) 76.9
Superpixel MRF, Tighe and Lazebnik (2010) 77.5
Simultaneous MRF, Tighe and Lazebnik (2010) 77.5
Traditional RNN, Socher and Fei-Fei (2011) 78.1
PSO-based RNN (our method) 78.3

scene types: city, countryside, and sea side. Therefore, like traditional RNN, we trained SVM that using the average over all nodes' activations in the tree as features. That means the entire parse tree and the learned feature representations of the RNN are taken into account. As a result, the accuracy has been promoted to 88.4%, which is better than traditional RNN (88.1%) and Gist descriptors (84%) [6]. If only the top node of the scene parse tree is considered, we will get 72%. The results reveal that it does lose some information that is captured by averaging all tree nodes.

5. Conclusions

In this paper, we have proposed an image parsing algorithm that is based on PSO and Recursive Neural Networks (RNNs). The algorithm is an incremental version of RNN. The basic idea is to solve the problem of nondifferentiable objective function of traditional training algorithm such as L-BFGS. Hence, PSO has been employed as an optimization tool to tune the weights of RNN. The experimental results reveal that the proposed algorithm has better performance than state-of-the-art methods on Stanford background dataset. That being said, the iteration of swarms dramatically increases the runtime of the training process. Our future work may focus on reducing the time complexity of the algorithm.

Acknowledgments

This work was supported by the National Nature Science Foundation of China (No. 61202143, No. 61103052, and No. 11101187); Major Project of Industry University of Fujian Province (2011H6020); Doctoral Program Foundation of Institutions of Higher Education of China (No. 20090121110032); Shenzhen Science and Technology Research Foundation (Nos. JC200903180630A, ZYB200907110169A); The Natural Science Foundation of Xiamen City, China (3502Z20123022).

References

[1] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Learning*, no. 99, pp. 1–15, 2012.

[2] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 33, no. 12, pp. 2368–2382, 2011.

[3] Z. Tu and S. C. Zhu, "Parsing images into regions, curves, and curve groups," *International Journal of Computer Vision*, vol. 69, no. 2, pp. 223–249, 2006.

[4] J. Porway, Q. Wang, and S. C. Zhu, "A hierarchical and contextual model for aerial image parsing," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 254–283, 2010.

[5] O. Teboul, L. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios, "Parsing facades with shape grammars and reinforcement learning," *IEEE Transactions on Pattern Analysis and Machine Learning*. In press.

[6] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[7] S. Nowozin, P. V. Gehler, and C. H. Lampert, "On parameter learning in CRF-based approaches to object class image segmentation," *Lecture Notes in Computer Science*, vol. 6316, no. 6, pp. 98–111, 2010.

[8] J. Tighe and S. Lazebnik, "Superparsing: scalable nonparametric image parsing with superpixels," *Lecture Notes in Computer Science*, vol. 6315, no. 5, pp. 352–365, 2010.

[9] X. He and R. Zemel, "Learning hybrid models for image annotation with partially labeled data," in *Proceeding of Advances in Neural Information Processing Systems 21, Proceedings of the 22nd Annual Conference on Neural Information Processing Systems*, pp. 625–632, Vancouver, British Columbia, Canada, December 2008.

[10] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 1–8, Kyoto, Japan, October 2009.

[11] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille, "Recursive segmentation and recognition templates for image parsing," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 34, no. 2, pp. 359–371, 2012.

[12] L. Zhu, Y. Chen, and A. Yuille, "Learning a hierarchical deformable template for rapid deformable object parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1029–1043, 2010.

[13] F. Han and S. C. Zhu, "Bottom-up/top-down image parsing with attribute grammar," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 59–73, 2009.

[14] L. Ladický, P. Sturgess, C. Russell et al., "Joint optimization for object class segmentation and dense stereo reconstruction," *International Journal of Computer Vision*, vol. 100, no. 2, pp. 122–133, 2012.

[15] R. Socher and L. Fei-Fei, "Connecting modalities: semi-supervised segmentation and annotation of images using unaligned text corpora," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 966–973, San Francisco, Calif, USA, June 2010.

[16] Z. K. Huang, C. H. Feng, and S. Mohamad, "Multistability analysis for a general class of delayed Cohen-Grossberg neural networks," *Information Sciences*, vol. 187, pp. 233–244, 2012.

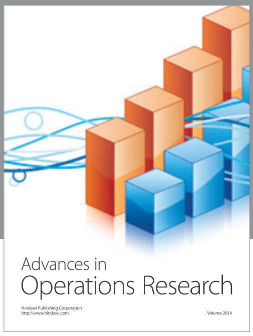
[17] R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the International Conference on Machine Learning*, pp. 129–136, 2011.

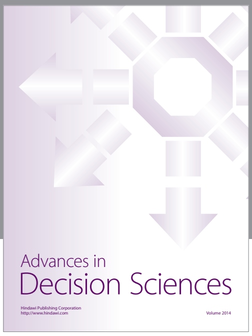
[18] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.

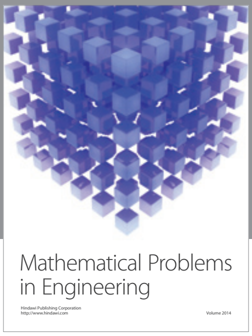
[19] G. R. Cai, S. Z. Li, S. L. Chen, and Y. D. Wu, “A fuzzy neural network model of linguistic dynamic systems based on computing with words,” *Journal of Donghua University*, vol. 27, no. 6, pp. 813–818, 2010.

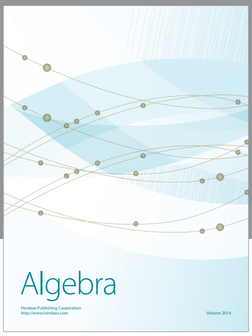
[20] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 24, no. 5, pp. 603–619, 2002.

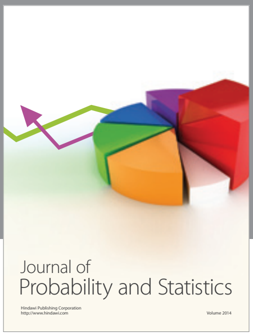
[21] Y. Shi and R. Eberhart, “Parameter selection in particle swarm optimization,” in *Proceeding of the 7th International Conference on Evolutionary Programming VII (EP ’89)*, pp. 591–600, 1998.


Advances in
Operations Research

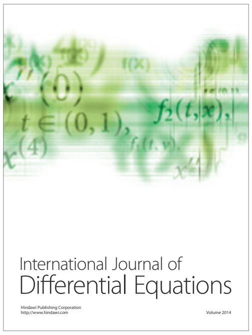

Advances in
Decision Sciences

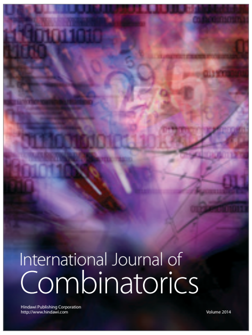

Mathematical Problems
in Engineering

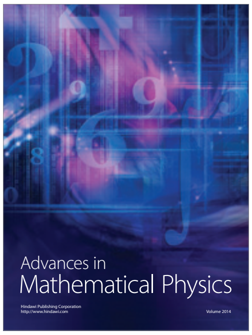

Algebra

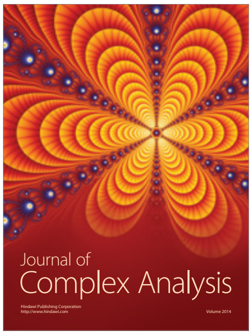

Journal of
Probability and Statistics

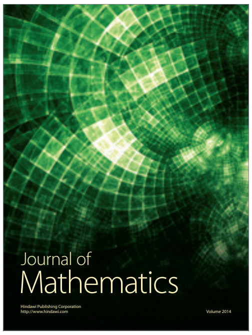
The Scientific
World Journal

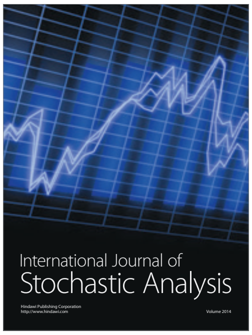

International Journal of
Differential Equations

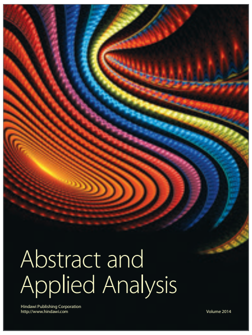

International Journal of
Combinatorics

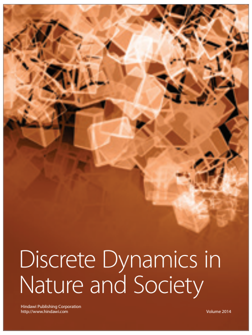

Advances in
Mathematical Physics

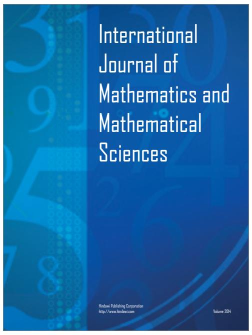

Journal of
Complex Analysis


Journal of
Mathematics


International Journal of
Stochastic Analysis

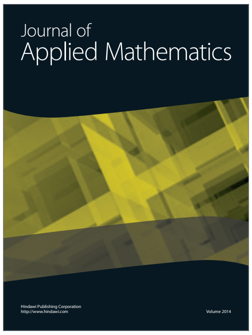

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


Journal of
Applied Mathematics

Journal of
Optimization