Automatic Classification of Sunspot Groups for Space Weather Analysis

Rudy Adipranata[1], Gregorius Satia Budhi[1] and Bambang Setiahadi[2]

[1]*Informatics Dept., Petra Christian University, Surabaya, Indonesia*
[2]*Indonesian National Institute of Aeronautics and Space (LAPAN), Watukosek, Indonesia*
rudya@peter.petra.ac.id), greg@peter.petra.ac.id,
bambangsetiahadi@rocketmail.com

Abstract

The sun is the unlimited energy source for life on the earth. However, besides as the energy source, the sun also gives disruptions to the universe around the earth and also to the life on the earth. Sources of the disruptions from the sun are flares and Coronal Mass Ejection/CME. Both of those disruptions in general come from group of sunspots. With the growing ~~of~~ dependency of human life with modern technology, either facility on the surface of the earth or in universe around the earth, the disruptions from the sun should be anticipated. In order to know the complexity level of sunspot groups and their activity, Modified-Zurich sunspot classification is used. Image of sunspots can be taken using the Michelson Doppler Imager instrument (MDI) Continuum / SOHO (Solar and Heliospheric Observatory).

This research was conducted on the automatic classification of sunspot group that can be used to analyze the space weather conditions and provide information to the public. There are two stages to classify sunspot groups namely feature extraction and pattern recognition. For feature extraction, we used digital image processing to get features of sunspot group, and for pattern recognition, we used artificial neural network. We compared 3 methods of artificial neural networks to get the best result of classification namely backpropagation, probabilistic and combination between self-organizing map and k-nearest neighbor. Among three of them, probabilistic neural network gave the best classification result.

Keywords: *sunspot groups classification, artificial neural network, pattern recognition*

1. Introduction

Life on the earth depends on the sun as the source of unlimited energy. However, besides as the energy source, the sun also gives disruptions to the universe around the earth and also to the life on the earth. There are two kinds of disruptions from the sun namely flares and Coronal Mass Ejection/CME. Both of those disruptions in general come from group of sunspots. The sunspots phenomenon is formed as a result of the magnetic flux tube emerged from the sun to the photosphere and corona [1]. The intersection of the magnetic flux tube with the photosphere forms a sunspot, which appears black because the magnetic flux has cooling effect so that the sunspot temperature is lower than the surrounding photosphere [2].

Sunspots evolve from a tiny spot with low activity into a very complex configuration with the possibility of having high activity, which issues an explosion and hurls corona mass. In order to determine the complexity level of sunspot groups and their activity, Modified-Zurich sunspot classification is used [3]. Modified-Zurich sunspot

classification has class A, B, C, D, E, F (growing level of complexity) and then is gradually declined until becomes a class H, which can be seen in Figure 1 [3]. This classification is very important to analyze the space weather.

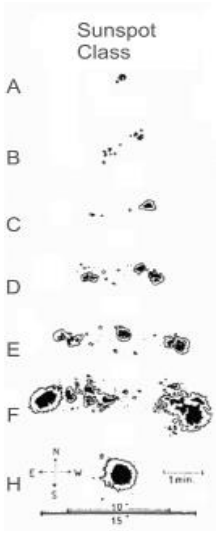

Figure 1. Modified-Zurich sunspot classification

2. Feature Extraction

Feature extraction is a stage to obtain features of sunspot groups in an image. Image of sunspots can be taken using the Michelson Doppler Imager instrument (MDI) Continuum / SOHO (Solar and Heliospheric Observatory). These features are area of sunspot group, perimeter of sunspot group, diameter of sunspot group and the number of sunspots in group. To extract the features, the image of sunspots should be segmented. Watershed method will be used for segmentation process. To get the features of sunspot group, sunspots, after being segmented, will be grouped using DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) clustering algorithm [4].

2.1. Watershed Segmentation

Watershed is one of the methods used to perform digital image segmentation. The concept of watershed is to assume that an image has three-dimensional shape that is the position of x and y with its color pixel level. The x and y are the positions of color pixel level, in which in this case, the gray level is the height of the assumption that the value closer to white has a higher altitude. Assuming the form of topography, there are three different points: (a) a point which is the regional minimum, (b) a point that if there is a drop of water, then the water will fall down to a certain minimum position, and (c) a point that if there is a drop of water, the water has the possibility to fall into one minimum position (not always fall to a minimum point, but it may fall into a certain minimum point or another). To a certain minimum regional, a set of points that satisfy the condition (b) is called a catchment basin, while the set of points that satisfy the condition (c) is called the watershed line [5].

From the explanation above, this watershed segmentation method has the objective to perform a search watershed line. The basic idea for the working of this segmentation is assumed that there is a hole made on the regional minimum, and then the whole topography is flooded by water from the hole with a constant velocity. When the water that rises from the

two catchment basins is about to join, a dam is built to prevent the merger. The flow of water will reach the desired level and stops flowing when only the upper part of the dam is visible. The edge of the dam is called the watershed line, and the watershed line is the result of segmentation, assuming that line the edge of the watershed is about to segmented objects. For more details, the drawing can be seen in Figure 2 [5].

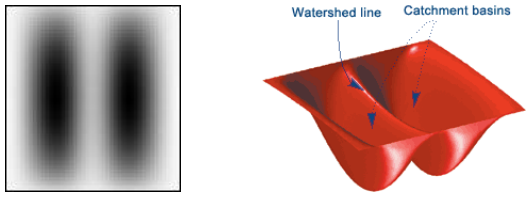

Figure 2. Watershed concept

Figure 2(a) displays a two-dimensional image of the watershed concept in which two dark colored parts are the catchment basins and an area between two catchment basins is an area where the watershed line will be, whereas Figure 2(b) displays three-dimensional images of the watershed concept.

3. Artificial Neural Network

Artificial neural network is a thinking model based on the working of the human brain [6]. Just like the human brain, neural networks are composed of interconnected neurons. Each neuron receives input signals, and they are processed to produce an output signal. Neural networks can learn like the human brain by giving weight to each input on neuron. By using weights, the neural network can learn a given input.

3.1. Backpropagation Neural Network

Backpropagation is a method of neural networks with multilayer feedforward neural network structure [6]. Neurons are divided into several layers, namely input layer, output layer and several hidden layers. Neurons in the hidden layer store the patterns of the input in the form of weights from the connections between neurons. The structure of backpropagation neural network can be seen in Figure 3 [6].

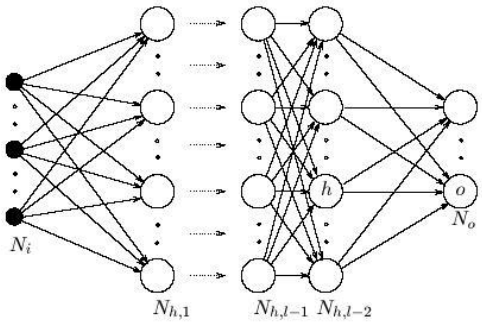

Figure 3. Backpropagation structure

The steps in the backpropagation algorithm are as follows [6]:

1. Initialization

Determine all the weights and thresholds of the neural network with random numbers that are in the range (1).

$$\left(-\frac{2.4}{F_i},+\frac{2.4}{F_i}\right) \tag{1}$$

F_i is the total input of neuron i in the neural network

2. Activation

Calculate output of neurons in the hidden layer and output layer using equation (2).

$$y(p)=sigmoid\left[\sum_{i=1}^{n} x_i(p)\times w_i(p)-\theta\right] \tag{2}$$

n is the number of input neuron, x is the input signal, w is the weight, θ is a threshold, p indicates p^{th} iteration, and *sigmoid* is the sigmoid activation function that follows the equation (3).

$$Y^{sigmoid}=\frac{1}{1+e^{-X}} \tag{3}$$

3. Weight Training

Calculate the error signal to the neurons in the output layer using the equation (4).

$$e(p)=y_d(p)-y(p) \tag{4}$$

y_d is the desired output results.

Calculate the error gradient for the neurons in the output layer using equation (5).

$$\delta(p)=y(p)\times[1-y(p)]\times e(p) \tag{5}$$

Calculate the weight corrections for neurons in the output layer using equation (6).

$$\Delta w(p)=\alpha\times y(p)\times\delta(p) \tag{6}$$

α is the learning rate

Change weight to the neurons in the output layer using equation (7).

$$w(p+1)=w(p)+\Delta w(p) \tag{7}$$

Calculate the error gradient for the neurons in the hidden layer using equation (8).

$$\delta_j(p)=y_j(p)\times[1-y_j(p)]\times\sum_{k=1}^{l}\delta_k(p)\times w_{jk}(p) \tag{8}$$

l is the number of neurons in the previous layer.

Calculate the weight correction for the neurons in the hidden layer using equation (9).

$$\Delta w(p) = \alpha \times x(p) \times \delta(p) \tag{9}$$

α is the learning rate

Change weight to neurons in the hidden layer using the following equation (10).

$$w(p+1) = w(p) + \Delta w(p) \tag{10}$$

4. Iteration

Repeat the above process starting from step 2 to obtain the value of sum of squared errors below 0.001.

3.2. Probabilistic Neural Network

Probabilistic Neural Network (PNN) is a method of artificial neural networks using the principles of Bayesian statistical theory which is to replace classification heuristic principle used by backpropagation algorithm [7]. Because of this, PNN is used to perform pattern classification [8].

Architecture of the PNN consists of four layers, namely input layer, pattern layer, summation layer and decision layer / output layer as shown in Figure 4 [7]. The input layer does not perform any calculations, just transferring the input data to each neuron in the pattern layer. Each neuron in the pattern layer will calculate probability (distance) between input data to the data stored in the neurons pattern layer. Furthermore, the summation layer will receive input from each neurons pattern layer and calculate the summation of them, so it will get the probability an input *x* which is the member of a group *t*. Lastly, output layer will produce classification results based on the results of the summation neuron that have the greatest value.

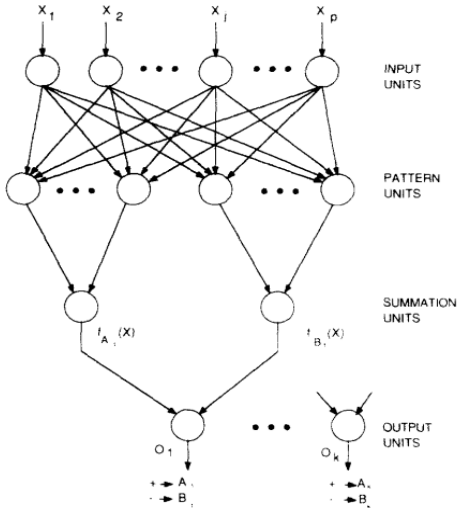

Figure 4. Probabilistic neural network architecture

3.3. Combination of Self-Organizing Map and K-Nearest Neighbor Neural Network

Self-organizing map is a method of artificial neural network that implements competitive learning [8]. In general, self-organizing map is used to perform the classification or grouping (clustering). Self-organizing map consists of *m* cluster units and *n* input units. Cluster units and input units are connected to the weight vector.

In general, the principle of self-organizing map is selecting one of the *m* cluster units with weight vector that is the most similar to the input pattern using minimum squared euclidean distance. Weight vector of the cluster unit and several other adjacent cluster units are updated. This method can be used to group multiple input vectors into *m* groups. The structure of self-organizing maps can be seen in Figure 5 [8].

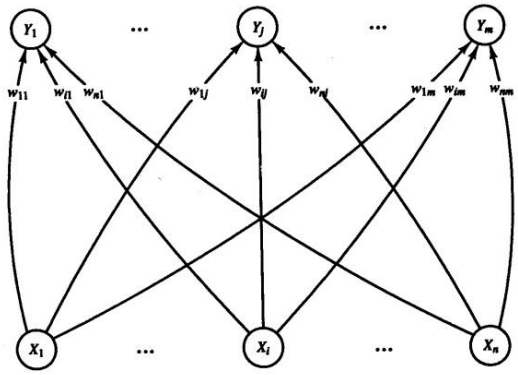

Figure 5. Self-organizing map structure

K-Nearest Neighbor (KNN) is a method to classify an object based on the data that are located closest to the object. This method is the most common method used for the estimation and prediction [9]. The steps in the KNN algorithm are as follows [10]:

Input: a set of pre-classified training instances, a query instances q, and a parameter k, defining the number of nearest neighbors to use

Output: a label indicating the class of the query instance q

Step 1: Find the k closest training instances to q according to a distance measure

Step 2: Select the class of q to be the class held by the majority of the k nearest training instances

To calculate the distance measure in Step 1, euclidean distance equation (11) is used. To count majority vote in Step 2, weighted voting equation (12) is used [10].

$$d_{Euclidean}(x,y) = \sqrt{\sum_i (x_i - y_i)^2} \quad (11)$$

$$vote = \frac{1}{d(x,y)^2} \quad (12)$$

4. System Design and Experimental Result

Overall system design for automatic classification of sunspot groups can be seen in Figure 6, and the example of digital image of sunspots can be seen in Figure 7.

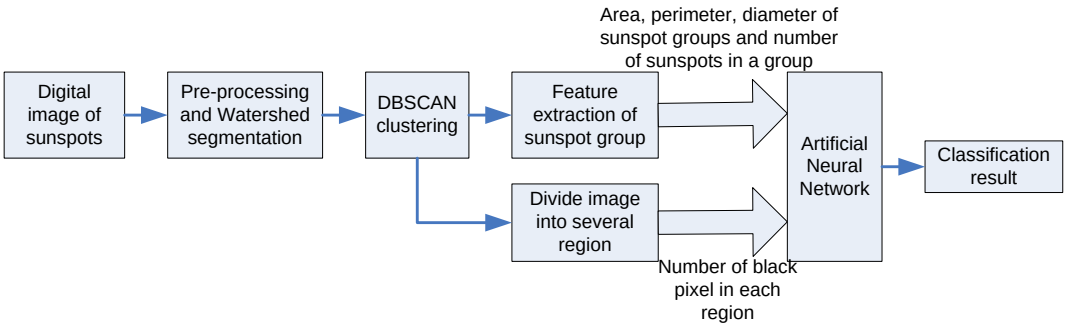

Figure 6. System Design

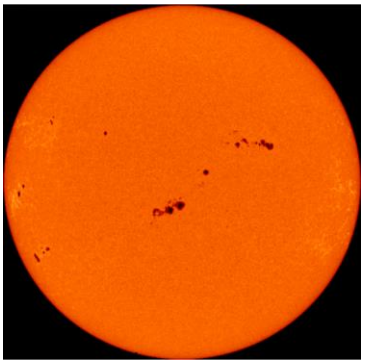

Figure 7. Digital image of sunspots

The watershed segmentation processing produced over segmentation, so that the desired object could not be segmented properly. It can be seen in Figure 8 [11]. To overcome the over segmentation, we should apply pre-processing before watershed segmentation.

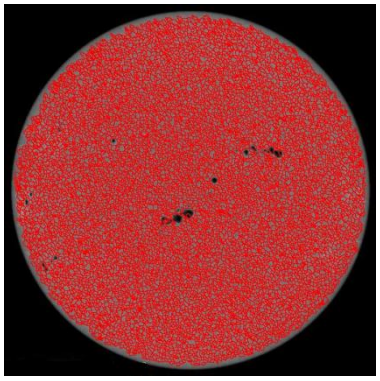

Figure 8. Result of watershed segmentation

The pre-processing implemented are opening, closing, erosion, dilation, canny and sobel [5]. We conducted research to find the best combination of those pre-processing to obtain the best segmentation results. From experiment, the combination of opening, erosion, sobel, canny, dilation and closing was the best combination to obtain the best segmentation result. It can be seen in Figure 9 [11].

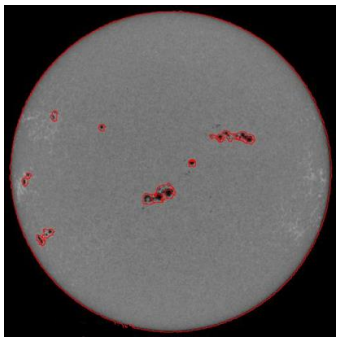

Figure 9. Result of watershed segmentation using pre-processing

After being segmented, the sunspot should be grouped. When viewed manually, sunspots form groups with irregular curve. This irregular curve is difficult to be detected automatically by using common clustering methods such as k-means or k-medoids. Because using those methods, the number of clusters, cluster central point should be determined, and will have oval or elliptical cluster result. Meanwhile, the curve of sunspot groups has irregular shapes. Therefore, to classify sunspots, we used DBSCAN (Density Based Spatial Clustering of Applications with Noise) clustering method [4].

Using DBSCAN, number of clusters generated is not limited and depends on the distance of each sunspot against other spots. DBSCAN clustering method is moving from one point to another and produces irregular clusters curved, according to their positions. This corresponds to the shape of the sunspot group. The results of clustering/grouping of sunspots with DBSCAN can be seen in Figure 10 [12]. After being clustered by using DBSCAN method, the next step was to search and mark the edges of each sunspot group, so that later, the image could be cropped easily for each group to extract the features. The features being extracted for each group are area of sunspot group, perimeter of sunspot group, diameter of sunspot group and the number of sunspots in group. Edge marking results of each group of sunspots can be seen in Figure 11 [12].

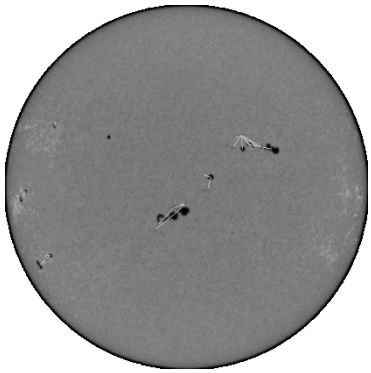

Figure 10. DBSCAN clustering result

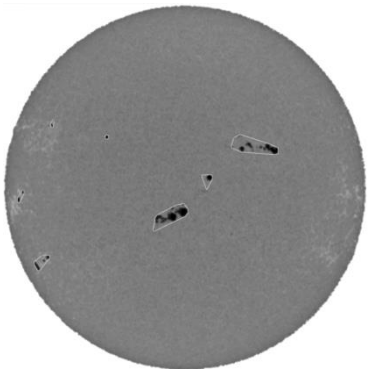

Figure 11. Edge marking of sunspot groups

Furthermore, the digital image of sunspot groups was converted into data that could be received by the input of the artificial neural network. To convert, the original color image was processed into black and white image by using grayscaling and thresholding methods. After being a black-and-white image, we divided the image into several regions. Examples of the process can be seen in Figure 12 [13]. After dividing into several regions, the number of black pixels in each region was counted.

Figure 12. Dividing sunspot groups image into several regions

Next was the process of normalization of the four group features and number of black pixels of each region by using the min-max normalization [14]. Min-max normalization is a method of normalization that transforms the data in linear fashion into a new range. Formula for min-max normalization can be seen in equation (13). The normalized values are used as neuron input of artificial neural network.

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A \quad (13)$$

We used and compared three artificial neural network methods to get the best result of classification, namely backpropagation, probabilistic and combination between self-organizing map and k-nearest neighbor.

In backpropagation, all input data was processed, and we updated all the weight value to achieve a convergent condition. The mapping of sunspot class to backpropagation output layer can be seen in Table 1.

Table 1. Mapping of sunspot class to backpropagation output layer

Output Layer			Sunspot Class
Neuron 1	Neuron 2	Neuron 3	
0	0	0	A
0	0	1	B
0	1	0	C
0	1	1	D
1	0	0	E
1	0	1	F
1	1	0	Unknown
1	1	1	H

In this backpropagation, there are three types of experiments that have been done. For the first one, we tested on variation in the number of hidden layer. We had 214 data, where 164 data was used for training and 50 data was used for testing. We used learning rate parameter at 0.2, the number of region = 5x5. Testing results can be seen in Table 2. From the results, it can be seen that 5 hidden layers result on the highest accuracy.

Table 2. Backpropagation testing results based on variations of the number of hidden layers

Hidden Layer	Accuracy	% accuracy
3 layers	32/50	64%
5 layers	33/50	66%
7 layers	15/50	30%

The second was testing on variation of learning rate. We used same parameters as the first one, but we used 5 hidden layers. Testing result can be seen in Table 3. From the results, it can be seen that learning rate = 0.2 produced the highest accuracy.

Table 3. Backpropagation testing results based on variation of learning rate

Learning Rate	Accuracy	% accuracy
0,2	33/50	66%
0,5	10/50	20%
0,8	15/50	30%

The last test was testing on variation of the number of image region. We used same parameter as testing before and 5 hidden layers. Testing result can be seen in Table 4. From the results, it can be seen that 5x5 regions resulted on the highest accuracy.

Table 4. Backpropagation testing results based on variation of the number of image region

The number of regions	Accuracy	% accuracy
3 x 3	22/50	44%
5 x 5	33/50	66%
7 x 7	26/50	52%

In probabilistic neural network, the number of input units was obtained from dimension size of the input data (*R*), consisting of diameter, area, perimeter of sunspot group, number of sunspots for each group, and the number of black pixel for each image region. After that, each input unit was connected to *Q* pattern units, where *Q* is the amount of training data. Thus, the size of initial weight matrix was *Q* x *R*. Each pattern unit generated a distance value between the input and weight. Also, each pattern unit was connected with all summation units through the last weight, which was *K* x *Q* matrix, where *K* is the number of the classification results, class A, B, C, D, E, F, or H. Then, each summation unit was connected to an output unit whose function is to seek the largest value of summation unit and take the index of the summation unit as a result of classification.

We have done two types of experiments using 214 data in which 164 data were for training and 50 data for testing. The first one was done by varying the spread of probabilistic neural network, and the input image was divided by 5x5 regions. Testing result can be seen in Table 5. When the spread is higher, the accuracy of classification will decrease.

Table 5. Probabilistic neural network testing results based on spread variation

Spread	Accuracy	% accuracy
0.2	47/50	94%
0.5	45/50	90%
0.9	43/50	86%

The second experiment was done on variation of the number of image region. We used 0.2 for spread value. Testing result can be seen in Table 6.

Table 6. Probabilistic neural network testing results based on variation of the number of image region

Number of regions	Accuracy	% accuracy
3 x 3	46/50	92%
5 x 5	47/50	94%
7 x 7	45/50	90%

In self-organizing maps, all sample data were trained until all the weight values that connected all input neurons to all self-organizing maps became convergent. After converging, the weight was stored and later could be used in the classification process. For the classification process, inputs required were a set of weights that have been converging, self-organizing maps architecture, sample of data used for training process, and sunspot group data that would be classified. The first step was to cluster of all sample data that would form clusters on available maps. Then, we looked for the winning map that best fits the data to be classified. Once obtained, then all sample data that belonged to a cluster with winning map were processed using the k-NN method to get the classification result.

For testing, the same as two methods before, we used 164 data for training and 50 data for testing. We have done testing on variation of the learning rate and used 5x5 regions size. The result can be seen in Table 7.

Table 7. Self-organizing maps testing results based on variation of learning rate

Learning Rate	Accuracy	% accuracy
0.2	36/50	72%
0.5	37/50	74%
0.9	36/50	72%

Another testing based on variation of region size has been done also. The result can be seen in Table 8.

Table 8. Self-organizing maps testing results based on variation of number of region

Region size	Accuracy	% accuracy
3 x 3	39/50	78%
5 x 5	38/50	76%
7 x 7	37/50	74%

After the completion of the experiments on each of the three artificial neural network methods, we compared the result of those methods. The comparisons that have been done between the three methods are comparison of the accuracy with a varying number of training data and comparison of the accuracy with a varying number of image regions. For comparison of the accuracy with a varying amount of training data, probabilistic neural network has performed better than the others. The result can be seen in Figure 13.

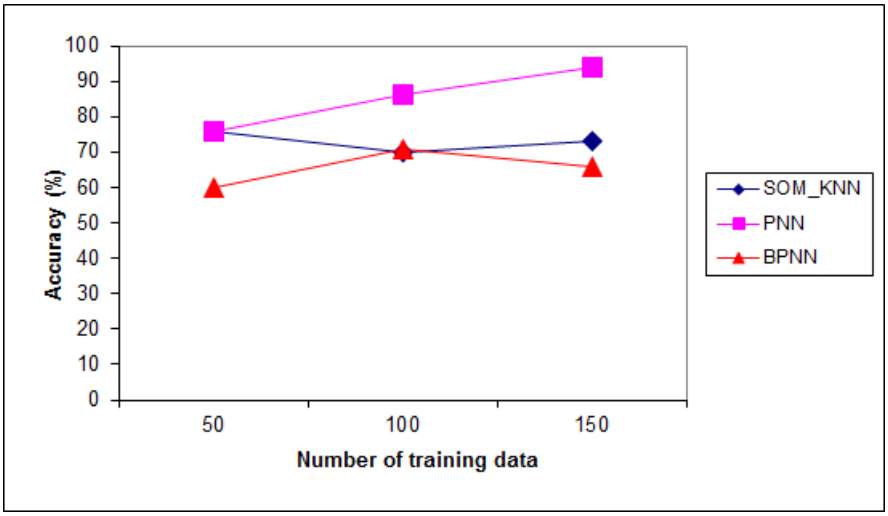

Figure 13. Comparison result based on varying number of training data

For comparison of the accuracy with a varying number of image regions, probabilistic neural network has performed better than the others again. The result can be seen in Figure 14.

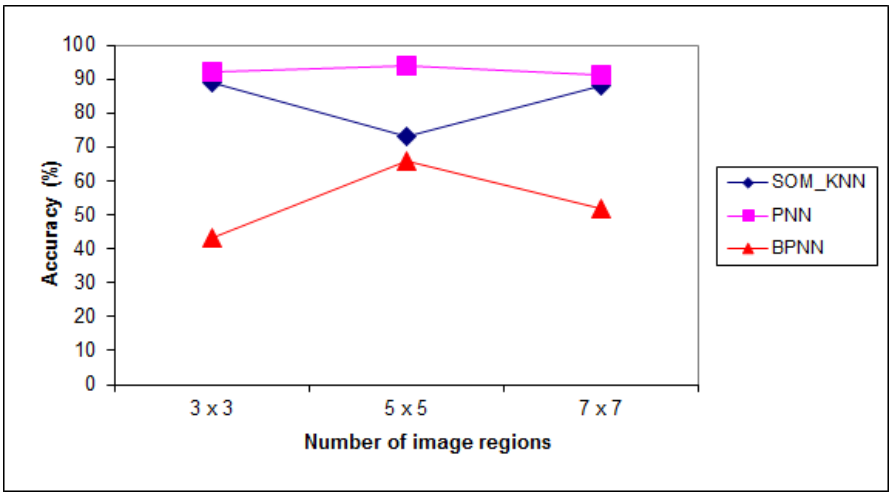

Figure 14. Comparison result based on varying number of image regions

From the comparison result, it can be concluded that probabilistic neural network performed better for classification of sunspot groups than the other two methods. Thus, in our system, we used probabilistic neural network. The screen view of our system can be seen in Figure 15. By using our system, the image of sunspot can be classified automatically and the classification result can be used for space weather analysis.

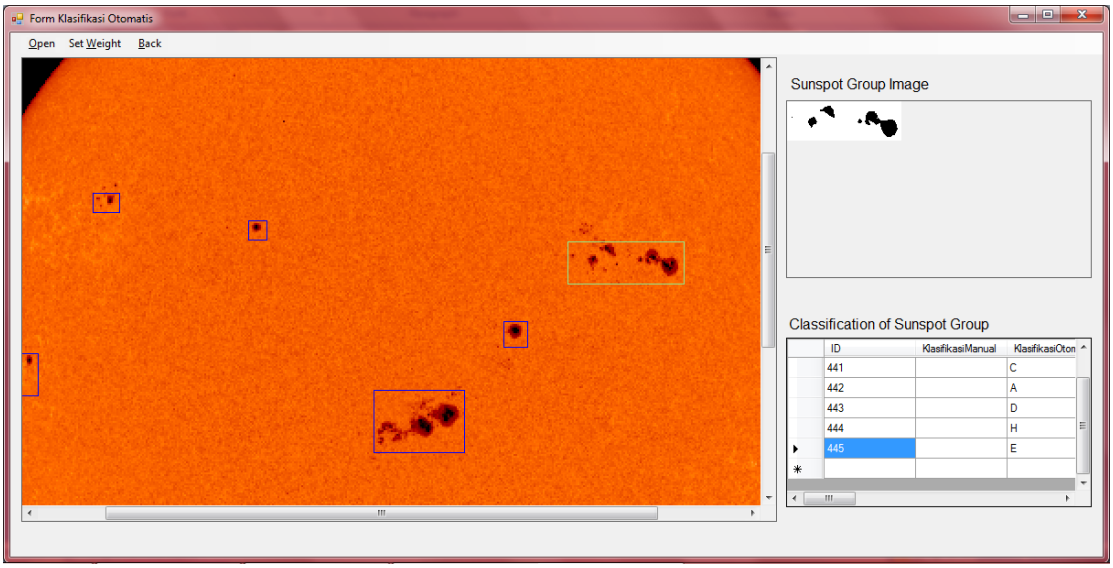

Figure 15. Screenshot of Automatic Classification System

5. Conclusion

This paper has discussed system development for automatic classification of sunspot groups from digital solar image. The classification was done using artificial neural network. From the experimental result, it can be concluded that backpropagation method was less appropriate to be used on the system, because the maximal accuracy was only 71%. Probabilistic neural network method was more appropriate for our automatic classification system because from the comparison result, it appeared that the performance of a probabilistic neural network was always better than the other two methods.

Acknowledgments

This research was funded by Research Competitive Grant DP2M Directorate General of Higher Education (101/SP2H/PP/DP2M/3/2010), National Education Ministry, Indonesia, fiscal year 2010, Research Competitive Grant DIPA-PT Coordination of Private Higher Education Region VII, East Java, Indonesia (0082/SP2H/PP/K7/KL/IV/2011), fiscal year 2011 and Research Center, Petra Christian University, Surabaya, Indonesia (12/Sugas-Pen/LPPM-UKP/2012), fiscal year 2012, entitled "Automated Sunspot Group Classification or Analyzing Space Weather Conditions". We also thank to Adrian Hartanto N. for his work on system coding.

References

[1] B. Setiahadi, "Automatic Determination of the Relative Sunspot Number from White-Light Full Disk Solar Digital Data Using Cluster Method and Turtle Algorithm", Proceeding of International Conference on Mathematics and Natural Sciences (ICMNS), Bandung, Indonesia, **(2006)**.
[2] V. Bothmer and I. A. Daglis, "Space Weather, Physics and Effects", Springer-Praxis Publishing, **(2007)**.
[3] Patrick S. McIntosh, "The Classification of Sunspot Groups", Solar Physics, vol. 125, no. 2, **(1990)**, pp. 251-267.
[4] M. Ester, H. P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, **(1996)**.

[5] R. C. Gonzalez and R. E. Woods, "Digital Image Processing 3rd Edition", Upper Saddle River, New Jersey, **(2008)**.
[6] M. Negnevitsky, "Artificial intelligence: A Guide to Intelligence Systems (2nd ed.)", Addison Wesley, New York, **(2005)**.
[7] D. F. Specht, "Probabilistic Neural Network", Neural Networks, vol. 3, no. 1, **(1990)**, pp. 109-118.
[8] L. Fausett, "Fundamentals of Neural Networks: Architectures, Algorithms, and Applications", Prentice-Hall International, Inc., **(1994)**.
[9] D. T. Larose, "Discovering Knowledge in Data: An Introduction to Data Mining", John Wiley & Sons, Inc., **(2005)**.
[10] A. Schenker, H. Bunke, M. Last and A. Kandel, "Graph-Theoretic Techniques For Web Content Mining", World Scientific, **(2005)**.
[11] R. Adipranata, G. S. Budhi, B. Setiahadi and B. Anwar, "Segmentation of Sunspot Group using Watershed Method", Proceeding of National Conference of System and Informatics, Bali, Indonesia, **(2010)**.
[12] G. S. Budhi, R. Adipranata, M. Sugiarto, B. Anwar and B. Setiahadi, "Sunspot Grouping of Solar Digital Image using DBSCAN Clustering Method", Proceeding of National Seminar of Information Technolgy Application, Yogyakarta, Indonesia, **(2011)**.
[13] G. S. Budhi, R. Adipranata, B. Setiahadi, B. Anwar, A. Hartanto and A. N. Tjondrowiguno, "Combination of Self-Organizing Map Neural Network and K-Nearest Neighbor for Automatic Classification of Sunspot Group Image", Proceeding of National Conference of System and Informatics, Bali, Indonesia, **(2011)**.
[14] J. Han, M. Kamber and J. Pei, "Data Mining: Concept and Techniques 3rd Edition", Morgan Kaufmann Publishers, Waltham USA, **(2006)**.

Authors

Rudy Adipranata is currently a senior lecturer in Informatics Department, Petra Christrian University, Surabaya, Indonesia. He received his bachelor degree in Electrical Engineering from Petra Christian University, Surabaya, Indonesia, and master degree in Software Engineering from Graduate School of Software, Dongseo University, Busan, South Korea. His research interests are image processing and computer vision.

Gregorius Satia Budhi is currently a senior lecturer in Informatics Department, Petra Christian University, Surabaya, He received his bachelor degree in Electrical Engineering, minority on computer science from Adhi Tama Institute of Technology Surabaya, Indonesia and master degree in Informatics from Sepuluh November Institute of Technology, Surabaya, Indonesia. His research interests are artificial intelligence and data mining.

Bambang Setiahadi is a senior researcher at LAPAN (Indonesian National Institute of Aeronautics and Space). He received his degree in ITB-Astronomy for B.Sc and Drs, and his M.Sc and D.Sc at NAOJ (National Astronomical Observatory of Japan). His permanent office is at Watukosek Solar-Terrestrial Observatory, East Java, Indonesia. His specialization are solar magnetohydrodynamics and solar activity early warnings.