Prediction of HSV color model parameter values of cloud movement picture based on artificial neural networks

Aleksander Radovan and Željko Ban
Faculty of Electrical Engineering and Computing, Zagreb, Croatia
aleksander.radovan@fer.hr
zeljko.ban@fer.hr

Abstract – In order to predict the exact moment of Sun shading by clouds and Sun cover duration to optimize the energy flow in the microgrid with solar photo electric system, it is essential to transform cloud images from RGB color model into HSV color model to be able to precisely detect cloud edges and determine the position of centroids for prediction of cloud movements. Parameters that define the quality of the image depend on the range of values for Hue, Saturation and Value (HSV) components. The dynamics of clouds and changing their shapes, sizes and colors require constant adjustments of those parameters by a human to get the best results. This paper deals with prediction and automatic setting of the HSV parameters by using artificial neural network and supervised learning. The image processing and parameters prediction was performed by an application developed in Java programming language based on JavaCV library and Encog framework for implementation of the artificial neural network.

Keywords – prediction of image parameters, HSV color model, neural networks, Java, Encog

I. INTRODUCTION

In order to track clouds on the sky and detect the moment of Sun shading by clouds, specific image transformations should be performed. These transformations include conversion from RGB to HSV color model for edge detection and performing the morphological operations erosion and dilatation [1]. For every of H, S and V components the minimum and maximum values should be determined to form a "filter" for image parameters in order to determine optimal cloud shape to get the most precise measurements. For the purposes of this paper a Java application that uses JavaCV [2] library has been developed. The Java application is used to perform described image transformations and to display the transformed image which consist of detected edges of the clouds in black, approximation of the shapes of the clouds with rectangle shape (in white) and ellipsis shape (in green), as shown on Figure 1. Also, the Java application draws the presumed location of the Sun (in yellow), detects the nearest cloud to the Sun (the rectangle and ellipsis of the nearest cloud are in red color), determines covered surface of the Sun (in black) and shows the current movement direction of the clouds based on detected centroid point and the location change of this point within 12 seconds (line in red that points from the centroid point to the image border). The clouds with the area smaller than 1000 pixels are ignored because they do not affect the Sun coverage. Similarly, in cases of merging several smaller clouds to one larger cloud, or disintegration of one larger cloud into several smaller clouds, generate a centroid position change that is ignored since it does not represent the movement of the clouds. For example, Figure 1 contains two objects without the red direction line from their centroids because they were forming a bigger cloud 12 seconds ago and this large cloud split into two smaller clouds. All other clouds contain this line because 12 seconds ago they got the similar shape.

The determination of centroid position is calculated every 12 seconds and the red direction line connects the last two calculated centroids, starting from next-to-last to the last, all through to the border of the image.

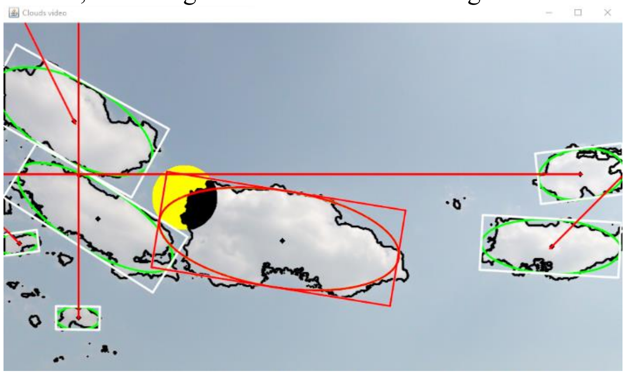


Figure 1. Java application screen with transformed image of the clouds

In addition to the Java application screen that shows the transformed image of the clouds, within Java application two additional screens are implemented: one with the histogram of the image (Figure 2) and one with slider components for fine tuning of H, S and V parameters of the image (minimum and maximum values) and the size of the structure element for erosion and dilatation operation purposes (Figure 3). Parameters show on those two figures were used as input or output data of the artificial neural network. For the generation of the histogram JFreeChart library [3] is used. R, G and B values from the histogram will be used as one the inputs for the artificial neural network used for automatically adjust the H, S and V parameters to get the best results of cloud edges detection. The artificial neural network automatically adjusts the parameters shown on the Figure 3 to keep the quality of the edge detection high and, at the same time, to maintain the precision of determining the position of the centroid based on closed curves.

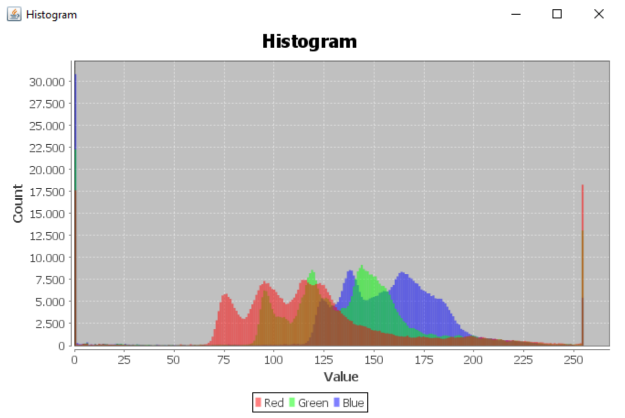
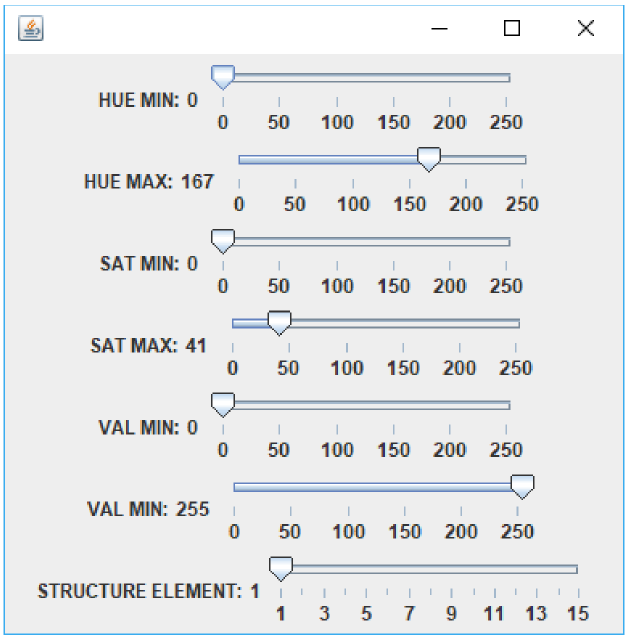


Figure 2. Histogram screen

Every of H, S and V components has the minimum and maximum value (for example, "HUE MIN" and "HUE MAX", as shown on the Figure 3). The size of the structural element defines a number of pixels used to precisely define the cloud edges using morphological operations erosion and dilatation. [1][4] The size of the structural element defines a precision of the cloud edge detection: the lower the size, the black line follows the shape of the cloud more precisely. If the structural element is too big, the precision of the cloud edge detection is not on the satisfactory level.

Figure 1 shows the case in which H, S and V parameters are optimally set, but if the values are not properly fine-tuned, the edges of the clouds could not be precisely defined. Figure 4 shows the case in which the S parameter is too high and Figure 5 show the case in which the S element is too low.



HUE MIN: 0

HUE MAX: 167

SAT MIN: 0

SAT MAX: 41

VAL MIN: 0

VAL MIN: 255

STRUCTURE ELEMENT: 1

Figure 3. Screen for H, S and V and structure element size fine-tuning

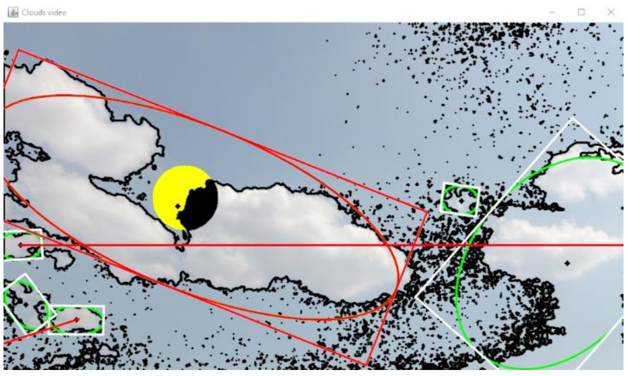


Figure 4. Image with too high value for the S parameter

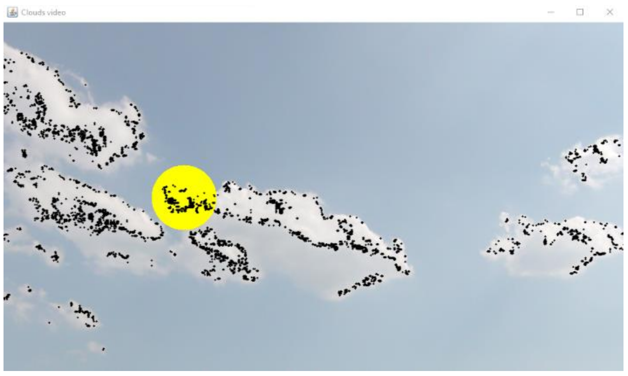


Figure 5. Image with too low value for the S parameter

II. AUTOMATIC DETERMINATION OF H, S AND V PARAMETER VALUES BY USING ARTIFICIAL NEURAL NETWORK

For the purpose of automatically determining the optimal values for the H, S and V components, the neural network and the Encog framework implemented in Java programming language were used.

The neural network has 5 layers of neurons, input layer, three hidden layers and one output layer as shown on Figure 6. The first input layer has 6 neurons and every neuron is used for one of the following parameters used in learning phase: the average R value from the histogram (Figure 2) from the previous image processed 12 seconds ago, the average value of the G component from the histogram from the previous image, the average value of the B component from the histogram from the previous image, the number of very large clouds (with the area greater than 20.000 pixels), the number of clouds with an area greater than zero pixels, but less than 100 pixels, and as the last parameter the number of clouds with the surface equal to zero (this can happen when one of the values that represent H, S and V components, like shown on Figure 4 and 5, has a value that was not optimized for cloud edges detection). All those values are based on the previous image and affects the parameters on the next image. Structure element size wasn't taken into calculations since the best result were generated with the value of 1.

Output layer has six neurons each of which is assigned one of the following normalized values for the following parameters: "HUE MIN", "HUE MAX", "SAT MIN", "SAT MAX", "VAL MIN" and "VAL MAX". All these parameters match the values used on image configuration

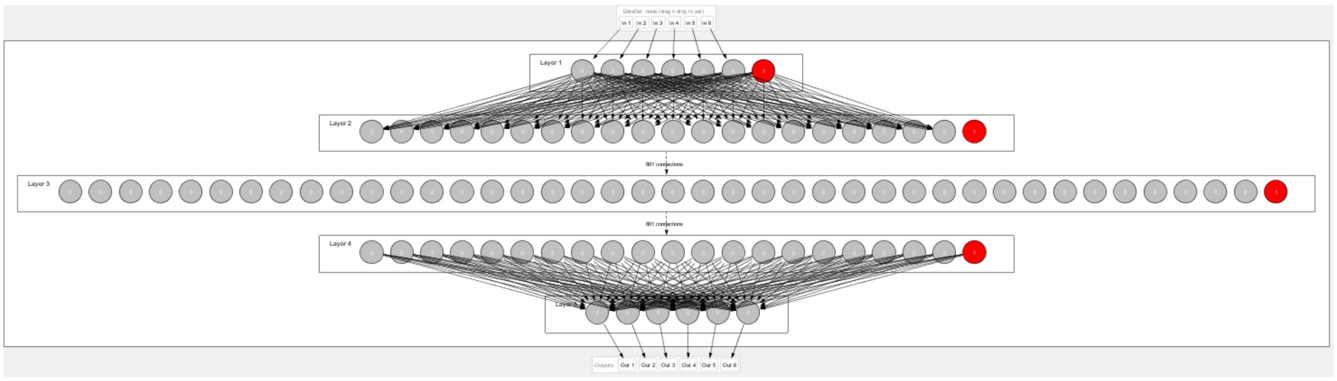


Figure 6. Neural network structure

screen shown on Figure 3 and define the boundary values for “filtering” the set of H, S and V values to optimally detect the cloud shapes on the image.

Three hidden layers between the input and the output of the neural network had the following number of neurons: 20 neurons in second layer, 40 neurons in third layer and 20 neurons in the fourth layer. All hidden layers and input used sigmoid activation function and bias neurons. Only the output layer didn’t use a bias neuron. The neural network structure is shown on the Figure 6 that was generated using the Neuroph, Java neural network framework. [5] Bias neurons are marked in red, while other neurons are in gray.

During the learning phase, input values were normalized so that the values were divided by the number of 256 in order to convert their values to the range between 0 and 1. These values were set as input values for the first three neurons of the input layer of the neural network. Depending on the number of clouds and elements in the image, if the number of elements with the area equal to 0 (represented by the small black points in Figures 4 and 5) was more than 200, then the input value for the fourth neuron was 1, and otherwise it was 0. If the number of clouds with the area greater than 0 but less than 100 was above 50, the value given to the fifth neuron was 1, otherwise it was 0. The last neuron was set to 1 when there was at least one a large cloud of 20,000, and if there were not so large clouds in the picture, then the sixth neuron set the value 0. These values are defined by the experts and are based on experience.

Supervised learning was performed by setting the image parameters (every 300th image that was processed, sampled every 12 seconds from the video) as input values for the neural network, where the application user, as the expert for setting the image parameters, could adjust the optimal parameters for the H, S and V components values through the whole video duration. After a certain number of processed images, the input and output data were submitted for learning as long as the error did not fall below the threshold of 0.15%. Collected neural network knowledge is stored to the external file and is loaded after every restart of the application during the autonomous application phase. [6]

III. EXPERIMENTAL RESULTS

Using the prepared neural networks to determine the optimal H, S and V parameter values on images other than those used in supervised learning phase gave the results shown in the next tables.

Firstly, the neural network was taught on 100 images of clouds and was used to automatically process 471 images for which the optimal parameters were defined by the image processing expert. After that, the amount of images for learning increased to 200, 300 and at the end of 400 images.

For supervised learning on 100 images to minimize errors below 0.15%, 701 learning iteration were needed. In the case of learning on 200 images, only 30 learning iterations were needed to minimize the error below 0,15%. For learning on the sample of 300 images, only 14 learning iterations were required, and on a sample of 400 images the network error fell below 0.15%, after only 6 learning iterations. In Table 1 the relative errors in prediction of all six output values in comparison to the image processing expert for different learning periods were shown. Since the changes within a time frame of 12 seconds were not significant, the relative errors were also small.

Table 1. Relative error values of predicted values for H, S and V image components

Learning period (number of images)	H MIN	H MAX	S MIN	S MAX	V MIN	V MAX
100	0%	0,0043%	0%	0,0017%	0%	0,0001%
200	0%	0,00225%	0%	0,0004%	0%	0,0001%
300	0%	0,00173%	0%	0,00056%	0%	0,00013%
400	0%	0,001525%	0%	0,0002%	0%	0,00005%

Given that the minimum values of H, S and V components during the entire supervised learning phase were set to 0, the neural network outputs were also identical to that value. Increasing the number of patterns (images) for learning, the error was getting smaller and it was possible to reduce acceptable the error level before using the neural network in the autonomous mode. For example, taking the maximum number of 10,000 iterations into consideration that were allowed to process before the learning process is interrupted, a lower maximum error rate for larger learning patterns was achieved, up to 0.007% for 400 learning patterns (in comparison to 0,15% in the case of lower number of processed iterations), as shown in Table 2.

Table 2. Minimum error rates that were achieved by using up to 10.000 learning iterations

Learning period (number of images)	Number of iterations	Minimum error rate
100	2693	0,14%
200	338	0,08%
300	769	0,08%
400	127	0,07%

The highest error rates that occurred when using a neural network for a particular H, S or V parameter are shown in Table 3. The highest error rates are related to the H and S parameters which were most often changed during the whole cloud tracking time.

Table 3. Maximum error rates of predicted values for H, S and V image components

Learning period (number of images)	H MIN	H MAX	S MIN	S MAX	V MIN	V MAX
100	0%	26,34%	0%	29,27%	0%	1,96%
200	0%	22,16%	0%	17,07%	0%	0,78%
300	0%	14,37%	0%	31,71%	0%	1,18%
400	0%	25,75%	0%	21,95%	0%	2,35%

Table 1 shows the relative error rates of the neural network when predicting the H, S, and V parameters of the image including the recognition of the same input parameters patterns used by the neural network at learning. If these values are excluded and the network is only used on those input values that were not used to learn, the average and maximum error values are shown in Tables 4 and 5.

Table 4. Relative error values of predicted values for H, S and V image components without using the learning patterns

Learning period (number of images)	H MIN	H MAX	S MIN	S MAX	V MIN	V MAX
100	0%	0,0002%	0%	0,0003%	0%	0,0003%
200	0%	0,00005%	0%	0,00005%	0%	0,00005%
300	0%	0,000033%	0%	0,000166%	0%	0,000033%
400	0%	0,00%	0%	0,000075%	0%	0,000025%

Table 5. Maximum error values of predicted values for H, S and V image components without using the learning patterns

Learning period (number of images)	H MIN	H MAX	S MIN	S MAX	V MIN	V MAX
100	0%	0,85%	0%	27,27%	0%	1,26%
200	0%	0,39%	0%	2,17%	0%	0,39%
300	0%	0,39%	0%	35,71%	0%	0,39%
400	0%	0,00%	0%	17,65%	0%	0,39%

IV. CONCLUSION

This paper presents one of the ways in which automated determination of optimal values of H, S and V components can be automated in order to adequately detect the edges of the clouds and to track their movement. Because of the unpredictable nature of clouds, it was not possible to precisely determine an algorithm to predict the H, S and V values, the artificial neural network was implemented using Java programming language and the Encog framework. Experimental results have shown that the error rates are smaller if the neural network had learned on a large number of images. Although it cannot be determined how exactly the neural network because of its "black box nature", and other disadvantages like greater computational burden, proneness to overfitting and the empirical nature of model development [7], the advantages like requiring less format statistical training, ability to implicitly detect complex nonlinear relationships between dependent and independent variables, ability to detect all possible interactions between predictor variables and the availability of multiple training algorithm can be very effectively used, like in this case.

The motivation for this work resulted of the need for constant presence of an expert who would need to adjust the H, S and V parameters for good image processing results. Instead, an artificial neural network can make a good substitute for that and is able to be further developed, on the basis of an expert system where one can intervene if necessary and correct the output results of the neural network if necessary and to create the new learning patterns. Additional new samples collected after the initial learning period of the neural network can be used to further improve accuracy of the future predictions, should be used in combination with the samples that were used before in order to maintain the present knowledge and enrich it with newer elements.

Other methods for solar radiation prediction can be based on geographical and meteorological data as inputs for the artificial neural network [8], but use different input parameters also show that by using the soft computing methods the prediction results can be used creating an autonomous photovoltaic system.

V. FUTURE WORK

The result in this paper will be used to further develop and automate the algorithm for predicting the clouds movement and calculate the exact moment and duration of the shadow over the Sun. These information can be directly used to predict the amount of direct solar irradiance and the amount of generated energy within the short period of times.

VI. REFERENCES

[1] A. Radovan, Z. Ban: „Prediction of cloud movements and the Sun cover duration", Faculty of Electrical Engineering and Computing, Zagreb, Croatia, IEEE Xplore Digital Library, Published 2014.

[2] JavaCV library, GitHub repository, https://github.com/bytedeco/javacv

[3] JFreeChart library, http://www.jfree.org/jfreechart/index.html

[4] R.M. Haralick, S.R. Sternberg, X. Zhuang: Image Analysis Using Mathematical Morphology, IEEE Transactions and Pattern Analysis and Machine Intelligence, vol. PAM1-9, No. 4, July 1987.

[5] Neuroph, Java neural network framework, http://neuroph.sourceforge.net/

[6] Saving and loading the neural network data to a file, http://4dev.tech/2016/02/saving-your-neural-network-training-progress/

[7] Jack V. Tu: Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes, Elsevier, 1996.

[8] Jane Oktavia Kamadinata, Tan Lit Ken, Torhu Suwa: Global Solar Radiation Prediction Methodology using Artificial Neural Network for Photovoltaic Power Generation System, SmartGreens 2017., 6th International Conference on Smart Cities and Green ICT System