

ARTIFICIAL LIGHT INTENSITY DETECTION IN GREENHOUSE HORTICULTURE BASED ON ARTIFICIAL INTELLIGENCE TECHNIQUE

Prayoth Kumsawat^{*}, Thanakorn Hoprasertwong, Arthit Srikaew, and Kitti Attakitmongcol

Received: June 26, 2020; Revised: October 31, 2020; Accepted: November 04, 2020

Abstract

Accurate detection of LED light intensity is important in many applications especially in the analysis of plant growth in greenhouse horticulture which has now become essential due to the need to control many parameters such as temperature, humidity and light intensity to grow specific plant. In this paper, we propose artificial LED light intensity detection in greenhouse horticulture by applying the Shallow Neural Network. We simulate various light intensity scenarios and use camera to capture the pictures. Then, these pictures will be used to train the Neural Network. The experimental results show that the algorithm can detect the LED light intensity accurately.

Keywords: Neural network, Artificial light, Backpropagation neural network, Light intensity

Introduction

Nowadays, weather and environment are not always favorable for agriculture. In addition, people have turned their attention to eat more fruits and vegetables than before. Therefore, various methods have been invented to grow plants efficiently. One of those is growing vegetables using artificial LED light in the greenhouse, which has widely been studied and developed, such as in (Olle and Viršile, 2013; Gómez and Izzo, 2018). However, the intensity of light has direct effect to the photosynthesis and growth rate of the plant. Thus, it is necessary to provide the artificial light with suitable intensity in the greenhouse.

In order to apply artificial intelligence technique to detect light intensity, it is necessary to have a camera to capture images of the light source from the LEDs. Then, these images will be used to

train And test algorithm based on the Shallow Neural network. Normally, Shallow Neural Network is the definition of a qualified Neural Network consisting of one hidden layer and trained only in supervised way (backpropagation) whereas the Neural Network with many hidden layer is known as the Deep Neural Network. Thus, depth of the Neural Network describes the number of hidden layers. The Shallow Neural Network is named to compare with the Deep Neural Network (Judd, 1988). However, if we use the Shallow Neural Network directly, this might be an overfitting problem. We can solve this problem by studying the improving Neural Network in (Rosin and Fierens, 1995).

In this paper, we propose algorithm to detect the LED light intensity and compare the effectiveness

School of Telecommunication Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand. E-mail: prayoth@sut.ac.th

^{} Corresponding author*

of our method with the other work in (Banik *et al.*, 2018b).

Materials and Methods

Our proposed method starts from backpropagation algorithm (Rojas, 1996). First, we simulate various events with red and blue color channels. We snap pictures of LED from simulation using Logitech C310 HD Webcam. Next, we convert the images from RGB to HSV color spaces using image processing algorithm (Gonzalez and Woods, 2002). Then, we define the rectangular box to extract HSV sub-image from the original HSV image. This process is based on RGB color channel intensity. Then, we calculate the average values of the intensity and store them in a table which will be later used as a condition with the light measurement using the Photosynthesis Photon Flux Density (PPFD).

In our method, there are four main feature values: Red, blue, green intensity and PPFD value from the MQ-200 Quantum sensor metering instrument. This method is classified as small data sets and complexity at low to medium levels which is appropriate to implement the Shallow Neural Network with this dataset. In comparison to the Deep Neural Network, the Deep Neural Network may yield slightly more accurate results but it consumes much more computational resources and processing time than the Shallow Neural Network which requires much smaller number of hidden layers.

Finally, we employ the scaled conjugate gradient algorithm (Moller, 1993) to train LED color intensity and classify pattern by the Shallow Neural Network toolbox. Figure 1 shows the block diagram of our proposed method.

Experimental Setup

Our algorithm collects sample data to create a new data set using Principle Component Analysis (PCA). This data set will be later used in the Shallow Neural Network algorithm. Our design uses 640 data sets, which have already been tested and found to be the most appropriate amount for training, validating and testing the Neural Network. We designed a total of 10 events that maintain blue intensities at 0%, 25%, 50%, 75%, and 100%, maintain red intensities at 0%, 25%, 50%, 75%, and 100% respectively. Then, in each of the above events, we maintain the color saturation value, which increases the color intensity (red) from events where we maintain the blue intensity. We will increase the red color by 25% = >50% = >75% = >100% respectively. In the event of maintaining the

red intensity, we will increase it the same way. Initially, we were interested in 1 event with 4 LEDs (1 LED image is equal to 4 red LEDs and 4 blue LEDs). For example, the event that keeps the blue intensity at 25% and then increases the red by 25%, indicates that we will be interested in 4 red LEDs or the event that, keeps the red intensity to 50% and increase the blue by 75%, indicates that we are interested in 4 blue LEDs).

We have designed the events in accordance with the random variable set at 640 random variables. In summary, we collect 4 LED images per event and consequently one event contains 16 data set. Finally, there will be a total of 40 events and 640 data set.

RGB Color Space to HSV Color Space

First, we snap pictures with the blue intensity of 25 % and various red intensity of 25, 50, 75 and 100 % in RGB as shown in Figure 2. Then, the pictures on Figure 2 are converted to HSV color space in Figure 3.

The RGB color space is composed of red, green, and blue dimensions. RGB color space is an image data obtained from the Logitech C310 HD Webcam. It is converted to the HSV color space,

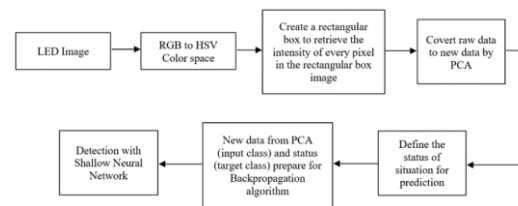


Figure 1. Block diagram for our operating process

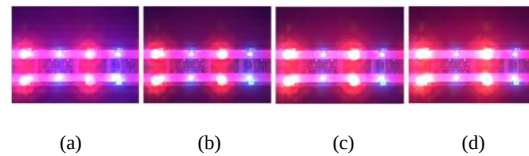


Figure 2. Simulation of the blue intensity 25% in RGB color space. (a) red intensity 25%, (b) red intensity 50%, (c) red intensity 75%, and (d) red intensity 100%

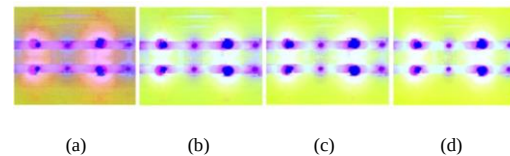


Figure 3. Simulation of the blue intensity 25% in HSV color space. (a) red intensity 25%, (b) red intensity 50%, (c) red intensity 75%, and (d) red intensity 100%

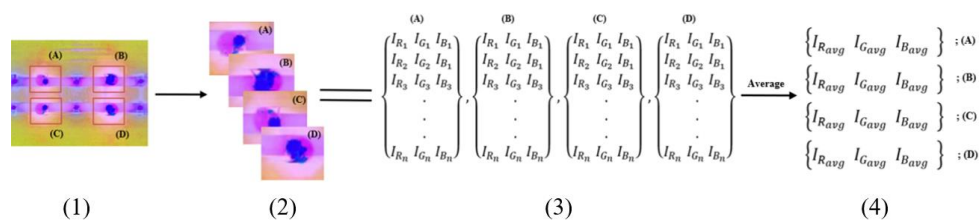


Figure 4. Schematic sketch of proposed method

which consists of light frequency difference (Hue), hue saturation (Saturation) and the brightness of the color (Value). When the image data is converted to HSV color space, the color saturation level is applied to the Neural Network based on the Red, Green, and Blue dimensions. This makes the range of every pixel of the image 0 to 255. To demonstrate the effectiveness of data preparation, this reduces the computation time and complexity for data preparation using Principal Component Analysis (PCA) in further step.

In our method, HSV color space is a better way for extracting important eigenvalues of the images as compared to the RGB color space. In the simulation event of the LED intensity detection, it is difficult to distinguish between 2 images in RGB color space. For example, in an event where the blue intensity is 0% and then the red LED intensity is increased by 25% = >50% = >75% = >100%. It is very difficult to observe a difference in the intensity of red LEDs in RGB color space. Thus, it is also difficult to find the distinctive features in the images. This makes the image extraction less effective. However, after the image is converted to HSV color space, it is easier to notice the differences between 2 images with different intensities. The images of the same light intensities in both color spaces are illustrated in Figures 2 and 3.

Data Management

This section describes the process of extracting information from images. First, we consider Figure 3(a) which shows the picture with blue and red intensity of 25% in HSV color space. Next, we determine 4 rectangular boxes as in Figure 4(1). Only the pixel values in these boxes will be extracted in the next step. In Figure 4(2), the picture in Figure 4(1) will be extracted into 4 sub-images (A)-(D). Then, for each sub-image, we extract the intensity in each pixel where $I_{R_1} I_{G_1} I_{B_1}$ represent the color intensity of red, green, and blue at pixel#1. We continue this process to every pixel in each sub-image where n is the number of total pixels. Finally, we calculate the average values of the color intensity of red, green, and blue for each sub-image as in Figure 4(4). This process will be repeated to

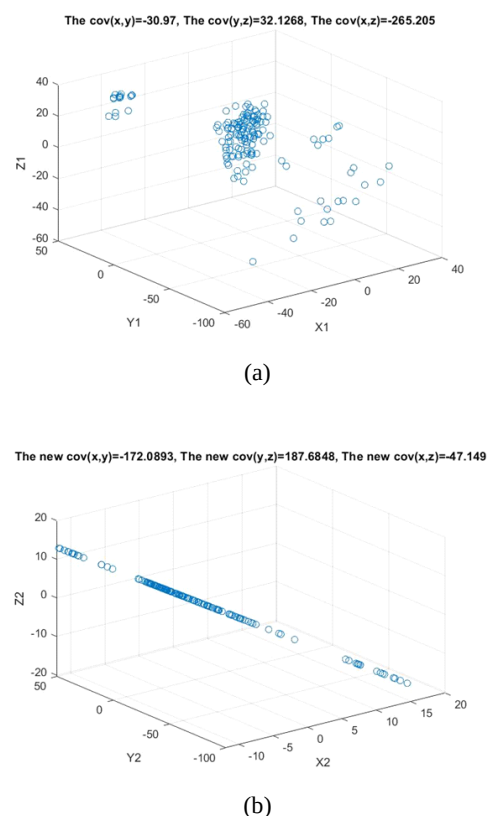


Figure 5. (a) 3D graph of Raw data, and (b) 3D graph of New data

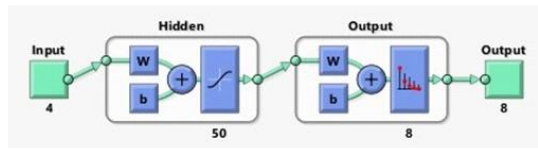
Figures 3(b), 3(c), 3(d), and all 40 events that we create.

Once we find all average values from all 40 events from pictures in HSV domain, we have the data of size 640×3 which is defined as raw data. Then, we apply the Principal Component Analysis (PCA) (Jolliffe, 1986) to the raw data to reduce redundancy as in (Ouivirach, 2011).

After, we exploit the PCA to transform our raw data to new data set, we compare the results in 3D axis as in Figures 5(a) and 5(b). We can see that after using the PCA in Figure 5(b), data is better organized than the raw one in Figure 5(a). Well-organized data normally yields good result in analysis and classification.

Table 1. Accuracy of color intensity for each color channel

Name of the method	Environmental light ($\mu\text{mol/s/m}^2$)	Color channel	Accuracy (%)
Multivariate regression model (Banik <i>et al.</i> , 2018)	13 (Low)	Red	90.93
		Blue	87.33
	57 (Strong)	Red	89.76
		Blue	87.20
BNN model (Banik <i>et al.</i> , 2018)	13 (Low)	Red	93.20
		Blue	92.27
	57 (Strong)	Red	93.23
		Blue	91.45
Our proposed method	13 (Low)	Red	99.99
		Blue	-
	57 (Strong)	Red	99.99
		Blue	99.99

**Figure 6. The network diagram**

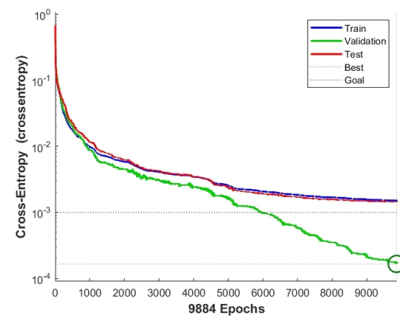
Shallow Neural Network

In this section, we describe our method of pattern classification using the Shallow Neural Network. As part of MATLAB R2018b, we use MATLAB and choose backpropagation algorithm (Moller, 1993) to train the new data from using the PCA. We also use 3 color values from the conversion by PCA and use PPFD value (LEDGardener, 2017), another important data obtained from the light measuring instrument, MQ- 200 quantum sensor with a handheld meter. Figure 6 shows our final network setting which defines all 4 inputs as red, green, blue, and PPFD intensities. The number of hidden layer for this algorithm is 50 and since there are 8 target values (categories) associated with each vector input, the number of neuron output is 8.

Results and Discussion

In our experiment, we use our dataset in three sections, which are the training section, the testing section, and the validating section. We use 70% of dataset in the training process, 15% in testing the network and 15% for algorithm validation. The target is to detect the percentage of red and blue intensity. If the detection is correct (the algorithm can classify the number of percentage of red and blue intensity), the output is 1. If not, the output is 0.

Figure 7 shows and compares the plot of the training, testing and validating efficiency. Figure 8 shows the confusion matrix, which both output and target class are in percentage. Table 1 shows the accuracy for each color channel.

**Figure 7. Plot the Training, the Validation, and the Test performance (Left)**

		Confusion Matrix									
Output Class	1	76 11.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	80 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	80 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	80 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	5	4 0.6%	0 0.0%	0 0.0%	0 0.0%	80 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.2% 4.8%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80 12.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80 12.5%	0 0.0%	0 0.0%	100% 0.0%
	8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80 12.5%	0 0.0%	100% 0.0%
		95.0% 5.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	99.4% 0.6%
		Target Class									

Figure 8. The confusion matrix (Right)

In Figure 7, we noticed that the cross-entropy of validating section yields the best result. It gives cross-entropy of 0.0015 at 9,884 epochs. This demonstrates that data set prepared for the neural network is suitable and the neural network has moderate to excellent performance. When we observe an important aspect of training and testing, the results of training and testing Neural Network have positive trend or the trend has identical direction with the one that shows superb performance. We consider to implement further hardware development

for LED light intensity detection in greenhouses. Finally, we consider very low cross entropy validation performance. The validation efficiency is very low because the Neural Network learning dataset is very difficult or the data set for validating the neural network is uncomplicated.

In Figure 8, a confusion matrix of size 8×8, with a total of 640 datasets is demonstrated. We can see that 80 sets of red are 25% intensity, 80 sets of red are 50% intensity, 80 sets of red are 75% intensity, 80 sets of red are 100% intensity, 80 sets of blue are 25% intensity, 80 sets of blue are 50% intensity, 80 sets of blue are 75% intensity, and 80 sets of blue are 100% intensity. In this confusion matrix, the outputs class is similar to the detection class. It can be seen that high accuracy of detection relates to the target class or actual class. However, there is an error at position 1 (80 sets of red are 25% intensity). The algorithm can detect 76 sets from 80 sets. The other 4 cannot be detected correctly. The overall performance of the algorithm has an efficiency of 99.4%.

Table 1 shows the comparison between our work with the previous work in (Banik *et al.*, 2018b). The comparison is made at the environmental light according to (Banik *et al.*, 2018b). Therefore, we have to convert the light measurement unit from lux to PPFD by the method described in (Waveform lighting, 2020). When the results in (Banik *et al.*, 2018b) is converted to PPFD, it matches one of our design event. However, there is one situation at low blue color intensity which does not match to our design. So the comparison cannot be made. From the results, it has been shown that our method yields higher accuracy than previous work in Multivariate regression model (Banik *et al.*, 2018a) and Boosting neural network (Banik *et al.*, 2018b).

From the results of our method, detection in the red and blue components has accuracy up to 98.75% and 100 %, respectively. This shows that the Shallow Neural Network is an effective tool.

Conclusions

In this paper, we propose a method to detect LED light intensity using the Shallow Neural Network. First, we use camera to capture pictures of light source with different intensity and then convert them from RGB color space to HSV color space. Then, we rearrange the pixel values of the image into a matrix, find the average values of three color components and use PCA to transform the data. We finally use this dataset to train, test and validate the Shallow Neural Network.

Our result has been shown that the proposed algorithm can detect the LED light in red and blue components accurately and yields the detection better than the result in (Banik *et al.*, 2018b).

Acknowledgement

This work was supported by a grant from Suranaree University of Technology, Nakhon Ratchasima, Thailand. The authors would like to thank the Thailand Research Fund and Commission on Higher Education for all great supports.

References

- Banik, P.P., Saha, R., and Kim, K.D. (2018a). Regression analysis for LED color detection of visual-MIMO system. *Opt. Commun.*, 413:121-130.
- Banik, P.P., Saha, R., Kwon, T.H. and Kim, K.D. (2018b). LED color detection of visual-mimo system using boosting neural network algorithm. *Proceedings of the 10th International Conference on Ubiquitous and Future Networks (ICUFN)*; July 3-6, 2018; Prague, Czech Republic, p. 685-688.
- Gómez, C. and Izzo, L. G. (2018). Increasing efficiency of crop production with LEDs. *AIMS Agri. and Food.*, 3(2):135-153.
- Gonzalez, R.C. and Woods, R.E. (2002). *Digital Image Processing Using MATLAB*. 2nd ed. Prentice Hall, New Jersey, 793p.
- Judd, S. (1988). On the complexity of loading shallow neural networks. *J. Complex.*, 4(3):177-192.
- Jolliffe, I.T. (1986). *Principal Component Analysis*. 2nd ed. Springer. NY, 518p.
- LEDGardener. (2017). Lumens, PAR, PPF, and PPFD: Measuring COB Grow Light Output.: LEDGardener. Available from: <https://ledgardener.com/lumens-par-ppf-and-ppfd-measuring-cob-grow-light-output/>. Accessed date: Jan 24, 2020.
- Moller, M. F. (1993). A Scaled Conjugate Gradient Algorithm For Fast Supervised Learning. *Neural Networks.*, 6:525-533.
- Olle, M. and Viršile, A. (2013). The effects of light-emitting diode lighting on greenhouse plant growth and quality. *Agri. and Food Sci.*, 22(2):223-234.
- Ouivirach, K. (2011). Writing PCA on MATLAB.: Ouivirach, K. Available from: <https://www.kanouivirach.com/2011/02/develop-pca-in-matlab-by-yourself/>. Accessed date: Jan 24, 2020.
- Rojas, R. (1996). The Backpropagation Algorithm. In: *Neural Networks - A Systematic Introduction*. Varga, P., (eds). Springer-Verlag, Berlin, p. 149-188.
- Rosin, P.L. and Fierens, F. (1995). Improving neural network generalization. *Proceedings of International Geoscience and Remote Sensing Symposium (IGARSS)*; July 10-14, 1995; Firenze, Italy, p. 1255-1257.
- Waveform lighting. (2020). Convert Lux to PPFD. Waveform lighting. Available from: <https://www.waveformlighting.com/horticulture/convert-lux-to-ppfd-online-calculator>. Accessed date: Jan 24, 2020.