

LED Color Detection of Visual-MIMO System Using Boosting Neural Network Algorithm

Partha Pratim Banik, Rappy Saha, Tae-Ho Kwon, and Ki-Doo Kim*

School of Electrical Engineering

Kookmin University, Seoul, Korea

*Corresponding author: kdk@kookmin.ac.kr

Abstract— LED color detection is a vital part in visual-MIMO system. For deciding transmitted symbols from an LED array image, it is important to detect the color of LED on receiver side. In this paper, we propose a training algorithm, called boosting neural network (BNN) to predict the color of LED on receiver side. First, we take the image of LED array and segment the LED image by using LED detection algorithm. After segmenting the LED image, the LED image is resized in 10 by 10 dimension that means 100 pixels. Each pixel is the input to the BNN model for each RGB color channel. For studying the behavior of each color LED image in low (565 lux) and strong (2450 lux) environmental light intensity, we train our BNN model for low and strong environmental light intensity. Finally, we compare the performance of our BNN model with the regression analysis model at low and strong environmental light intensity. We obtain greater closeness accuracy for each color channel at both environmental light intensities.

Keywords—visual-MIMO, boosting, neural network, LED array, symbol decision.

I. INTRODUCTION

Visual-MIMO system implies visual communication between LED array and camera. MIMO stands for multiple-input multiple-output and it refers “multiplexing” to send the stream of bits using multiple elements of light like LED array [1, 2]. LED array transmits the symbols and as a receiver camera captures the image of LED array to decide the transmitted symbols. The symbols are the colors emitted from the LEDs. But the spectrum of visible light is affected by high environmental noise such as environmental light intensity and other heat sources of IR [3]. As a result, receiver cannot receive the actual colors of symbols. For this reason, we need a processing method in the receiver side to process this distorted symbols in such a way that the distorted symbols color get closed to the actual symbols color.

To overcome this environmental light interference problem, we need a color detection algorithm. Machine learning algorithms are widely used for different color detection applications [4, 5]. A phenomena of color distortion in receiver side is shown in Fig. 1. Fig. 1 shows the distortion of the transmitted color in receiver side. For this reason, any image processing algorithm gets stuck to process it for detecting the color of the symbol in receiver side. From this sense, we can say that if we cannot process it to detect the actual color by image processing algorithm, we can predict it as precisely as

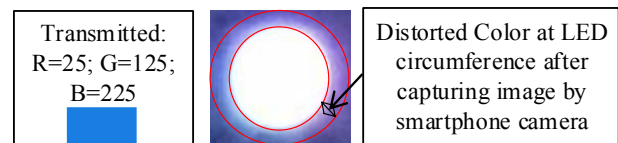


Fig. 1. An example of distorted color in receiver side.

possible by designing a learning algorithm.

In this paper, we propose boosting neural network (BNN) learning algorithm to predict the color of symbol. We use the dataset and LED detection algorithm of paper [6]. We compare the performance of our method with that of paper [6].

We describe our research works in the following sections. The related works are described in section II. In section III, the proposed method is described. We describe the experimental setup in section IV. Results are described in Section V and the paper is concluded in section VI.

II. RELATED WORKS

J.-E. Kim et al. described a visual-MIMO scheme with original color and brightness during the performance of seamless communication [2] and also explained the applicability of Visual-MIMO in V2X communication [7]. P. Hu et al. tried to increase the data rate without detecting the color of LED [8]. In addition to this, they used tri-color RGB LED to produce color bars but the distance between transmitter and receiver was about 3 cm which was unrealistic short distance [8]. M. Varga et al. showed a demo application of visual-MIMO in automobile safety [9]. They described that in the application of visual-MIMO, computer vision based image analysis techniques are needed to demodulate the transmitted signals [9].

There are some researches for LED color detection. J.-E. Kim et al. described an idea of dynamic palette to detect the color of the transmitted symbol more efficiently than conventional method [10]. But they evaluated their method for some cases of symbol decision. In [6], the multiple linear regression was used to predict the color. They described elaborately the performance of their multiple linear regression for detecting LED color of visual-MIMO system. In this paper, we propose a more efficient NN based algorithm that performs better than [6] and we validate our proposed algorithm by evaluating it by the metrics of [6].

This research was supported by the Basic Science Research Program through the National Research Foundation (NRF) of Korea funded by the Ministry of Education [2015R1D1A1A01061396] and was also supported by the National Research Foundation of Korea Grant funded by the Ministry of Science, ICT, Future Planning [2015R1A5A7037615].

III. PROPOSED METHOD

Our proposed method is based on the combination of boosting [11] and neural networks (NN). After segmenting LED array image, the LED image is resized in 10 by 10 dimension by using the `imresize()` function of MATLAB. This function resizes any image by using bi-cubic interpolation algorithm with antialiasing. We have 100 intensity values for every color channel. In [11], boosting is defined as the combination of some weak learners to build a strong learner. For this reason, we first train every pixel by using back propagation algorithm [12] through an NN network which consists of one input, one hidden neuron and one output neuron. Every trained pixel can predict the color of symbols but their predictions are so erroneous. They act like weak learners. We can use these weak learners to establish a strong learner by training them with backpropagation algorithm in a fully connected neural network. In the fully connected NN model, we use three layers, input layer, hidden layer with six neurons and output layer. The 1st and 2nd training flowcharts are given in Fig. 2 and 3, respectively.

The differences in Fig. 2 and 3 are the input and termination condition. In Fig.2, we use $(\text{previous RMSE} - \text{current RMSE}) < 0.001$ because after every iteration the root mean square error (RMSE) value of training dataset will be decreased. So, previous RMSE will always be greater than current RMSE. After updating weights properly, the current RMSE will not decrease largely and we check the change in RMSE less than 0.001. We set the value 0.001 because it

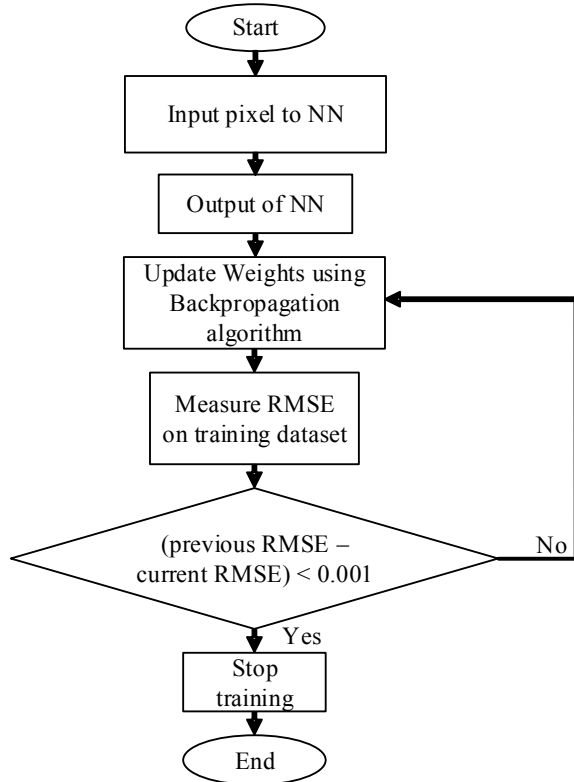


Fig. 2. Training flowchart for input pixel.

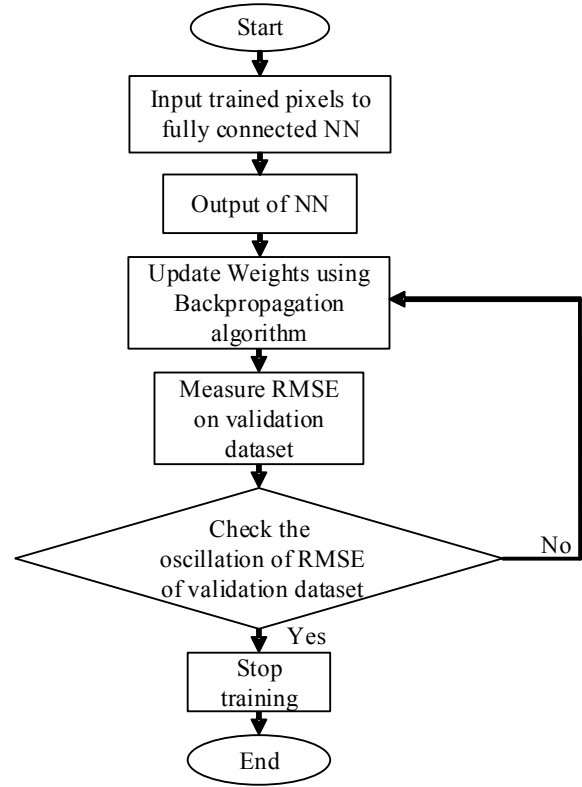


Fig. 3. Training flowchart for input trained pixels.

indicates very small change in RMSE that means this small change of RMSE will not update the weight vector largely. For this reason, we can terminate the training after satisfying this change. In Fig.3, the input is 100 trained pixels to fully connected NN. The fully connected NN is shown in Fig. 5. In this stage, the termination condition is the checking of oscillation of RMSE on validation set. Generally, in every iteration, the RMSE of training dataset and validation dataset will be decreased. But at a point, the RMSE on validation set will start increasing or oscillation. This indicates that if the training is going on after that point, the weight vector will become over-fit on training dataset. It also indicates that the weight vectors will work properly only on training dataset. The validation set is very useful to prevent this overfitting. So, we check the oscillation for terminating the training session of fully connected NN. We do not use it in Fig. 2 because the training is performed only for one input pixel. So, we do not need to use it. Overfitting issue will arise when there are multiple inputs to any machine learning model.

After training, the architecture of each stage is shown in Fig. 4 and 5. x_j is the input intensity of any color channel. w_j is the weight from input to hidden neuron, K. Here, 'j' stands for the input number. w_k is the weight from hidden to output neuron, L. L is the final neuron which is called weighted input. Here, the Leaky Rectified Linear Unit (ReLU) activation function is used for the neuron of hidden layer. The mathematical expression of the activation function of hidden neuron, K is shown in (1).

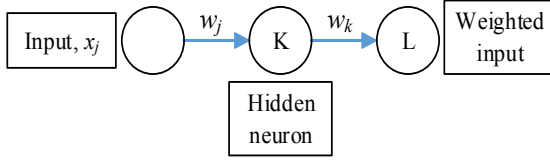


Fig. 4. The architecture of NN from input to weighted input of 1st stage.

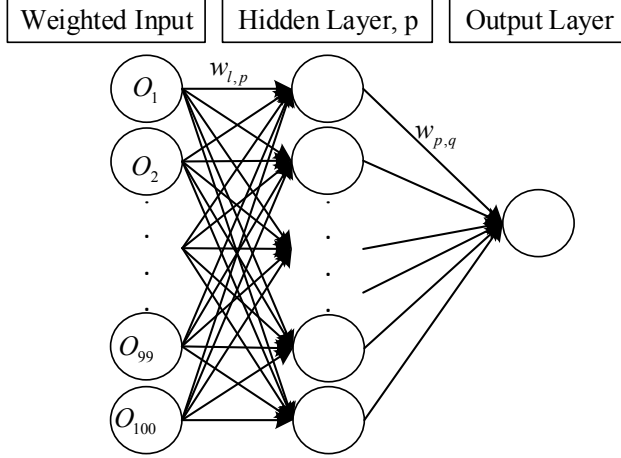


Fig. 5. The architecture of fully connected NN from weighted input to output of 2nd stage.

$$O_K = \max(aw_jx_j, w_jx_j), \quad a \neq 1 \text{ and } 0; \quad a > 1 \text{ or } a < 1 \quad (1)$$

O_K is the output of neuron K. The activation function of neuron L is given below:

$$O_L = \max(aw_Kx_K, w_Kx_K), \quad a \neq 1 \text{ and } 0; \quad a > 1 \text{ or } a < 1 \quad (2)$$

O_L is the output of neuron L. This is the weighted input. Then, the 100 weighted input is given to the fully connected NN with a hidden layer of six neurons. These six neurons are connected to a neuron in output layer. Leaky ReLU function is also used as activation function for six neurons of hidden layer and output layer neuron. Fig. 5 shows the fully connected network with weighted input. O_1 to O_{100} are the weighted inputs. $w_{l,p}$ is the weight from weighted input to hidden layer neuron. ' p ' means the weighted input and ' p ' means the hidden layer neuron. $w_{p,q}$ is the weight from hidden layer neuron to output layer neuron. ' q ' means the output layer neuron. In output layer, there is one neuron as output. It is designed for every color channel. The Leaky ReLU activation function is used in every neuron.

IV. EXPERIMENTAL SETUP

Fig. 6 shows the detail experimental setup with necessary instruments. The experiment was done in a dark room with two light sources. The light sources can vary up to different intensity (lux). The transmitter LED and receiver camera are fixed at distance of 30 cm by tripod 1 and tripod 2. So, both transmitter and receiver are in the same level. For this reason, shooting angle is 0° with respect to object direction because they are placed face to face. Light meter (CL-200A) was used

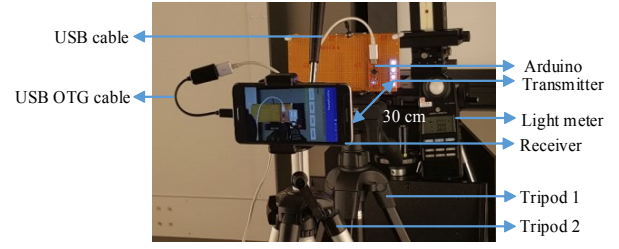


Fig. 6. Experimental setup.

to measure the environmental light. We used WS2812B, 1×4 LED array as transmitter which is programmed by Arduino Nano MEGA328P based microcontroller then, we take pictures using smartphone SAMSUNG GALAXY GRAND PRIME device by setting USB communication between camera and Arduino by a mobile app. We take our image data in two light intensities, 565 lux and 2450 lux and named this intensity as low and strong environmental light, respectively. In each intensity, we captured images at low exposure. So, we take total two sets of image data and each set contains 84 images. Each image contains 4 symbols.

V. RESULT AND DISCUSSION

In this paper, we divide the total dataset into three portions, training, validation and test dataset. We use 65% dataset as training dataset, 15% dataset as validation dataset and 20% dataset for test dataset. We compare the result of BNN algorithm with the multiple linear regression algorithm of paper [6]. We compare the predicted color distortion percentage from transmitted color in CIE1931 color space and also measure the closeness accuracy for each color channel. The results are calculated on test dataset. Table 1 and 2 show the closeness accuracy for each color channel and the number of test symbols on different range of distortion percentage at low and strong environmental light intensity.

TABLE I. CLOSENESS ACCURACY FOR EACH COLOR CHANNEL

Name of the method	Closeness Accuracy (%) for Each Color Channel		
	Environmental light	Color channel	Closeness Accuracy (%)
Multiple linear regression [6]	Low (565 lux)	Red	90.93
		Green	88.68
		Blue	87.33
	Strong (2450 lux)	Red	89.76
		Green	85.20
		Blue	87.20
Proposed method	Low (565 lux)	Red	93.20
		Green	94.96
		Blue	92.27
	Strong (2450 lux)	Red	93.23
		Green	95.44
		Blue	91.45

TABLE II. NUMBER OF TEST SYMBOLS FOR DIFFERENT RANGES OF DISTORTION

Name of the method	Number of Test Symbols And Distortion Ranges				
	Environmental light	<15 %	15% to 50%	50% to 70%	>70%
Multiple linear regression [6]	Low (565 lux)	45	19	3	1
	Strong (2450 lux)	38	24	5	1
Proposed method	Low (565 lux)	60	8	0	0
	Strong (2450 lux)	64	4	0	0

We get more improved closeness accuracy for each color channel at low and strong environmental light intensity than paper [6], as shown in Table 1 by bold font. Table 2 shows the different range of distortion percentage at CIE1931 color space. Paper [6] achieved 66.17% and 55.88% of test symbols having less than 15% distortion. In our proposed method, we get this value more, 88.25% and 94.11% of test symbols having less than 15% distortion at low and strong environmental light, respectively. We achieve more test symbols having small distortion than paper [6].

VI. CONCLUSION

This paper considers validation set during training that prevents overfitting tendency. But paper [6] did not consider any validation set to get rid of the problem of overfitting tendency in their trained model. For this reason, BNN's prediction achieves more closeness accuracy in each color channel than paper [6]. The idea of using pre-trained pixels as input to the proposed BNN model contributes to achieve better performance in LED color detection than paper [6]. There is another reason for better performance of BNN. Paper [6] used very small number of input parameters, about fifteen, but BNN uses 100 input parameters. Generally, NN model establishes more complex non-linear relationship between input and output than multiple linear regression model. These are the reasons to achieve better result in LED color detection by BNN model than the algorithm of paper [6] on same dataset. Although it is not experimented in this paper, from the improvement of closeness accuracy in each color channel by BNN model, we can say that BNN model has the possibility to perform better in symbol decision than paper [6] for visual-MIMO system.

REFERENCES

- [1] A. Ashok, M. Gruteser, N. Mandayam, and K. Dana, "Characterizing multiplexing and diversity in visual MIMO," 45th Annual Conference on Information Sciences and Systems (CISS), pp. 1-6, March 2011.
- [2] J.-E. Kim, J.-W. Kim, Y. Park, and K.-D. Kim, "Applicability of color-independent visual-MIMO for v2x communication," 7th International conference on ubiquitous and future networks (ICUFN), pp. 898-901, August 2015.
- [3] A. Ashok, M. Gruteser, N. Mandayam, J. Silva, M. Varga, and K. Dana, "Challenge: Mobile optical networks through visual MIMO," MobiCom'10: Proceedings of the 16th Annual International Conference on Mobile Computing and Networking, New York, NY, USA, pp. 105-112, 2010.
- [4] J. L. Binangkit and D. H. Widyantoro, "Increasing accuracy of traffic light color detection and recognition using machine learning," 10th International Conference on Telecommunication Systems Services and Applications (TSSA), pp. 1-5, October 2016.
- [5] T. Kawanabe, N. D. Kamarudin, C. Y. Ooi, F. Kobayashi, X. Mi, M. Sekine, A. Wakasugi, H. Odaguchi, and T. Hanawa, "Quantification of tongue colour using machine learning in Kampo medicine," European Journal of Integrative Medicine, vol. 8, no. 6, pp. 932-941, December 2016.
- [6] P. P. Banik, R. Saha, and K.-D. Kim, "Regression analysis for LED color detection of visual-MIMO system," Optics Communications, vol. 413, pp. 121-130, April 2018.
- [7] J.-E. Kim, J.-W. Kim, Y. Park, and K.-D. Kim, "Color-space-based visual-mimo for v2x communication," Sensors, vol. 16, issue 4, 591, April 2016.
- [8] P. Hu, P. H. Pathak, X. Feng, H. Fu, and P. Mohapatra, "ColorBars: increasing data rate of led-to-camera communication using color shift keying," proceedings of the 11th ACM conference on Emerging Networking experiments and technologies, article no. 12, December 2015.
- [9] M. Varga, A. Ashok, M. Gruteser, N. Mandayam, W. Yuan, and K. Dana, "Demo: visual MIMO based led - camera communication applied to automobile safety," proceedings of the 9th international conference on Mobile systems, applications, and services, Bethesda, Maryland, USA, pp. 383-384, 2011.
- [10] J.-E. Kim and K.-D. Kim, "Symbol decision method of color-independent visual-MIMO system using a dynamic palette," 2017 23rd Asia-Pacific Conference on Communications (APCC), pp. 1-4, December 2017.
- [11] A. J. Ferreira and M. A. T. Figueiredo, "Boosting Algorithms: A Review of Methods, Theory, and Applications," in Zhang C. Ma Y. (eds) Ensemble Machine Learning, Chapter 3, Springer, Boston, MA, pp. 35-85, January 2012.
- [12] R. Rojas, "Neural Networks - A Systematic Introduction," Springer-Verlag, Berlin, Heidelberg, Chapter 7, pp. 149-182, 1996.