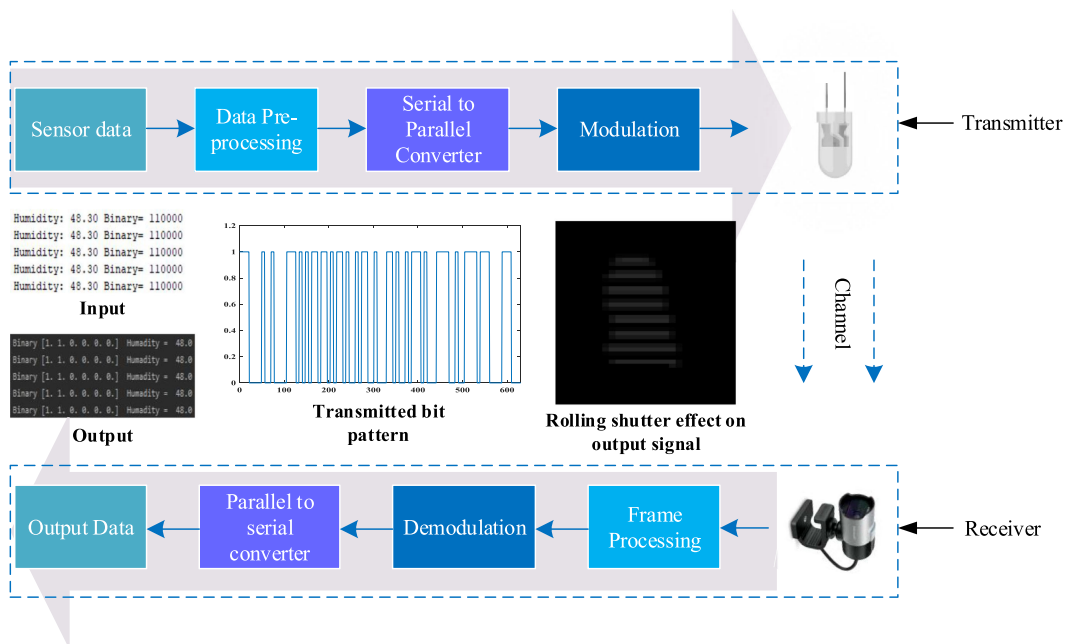


# Experimental Demonstration of Continuous Sensor Data Monitoring Using Neural Network-Based Optical Camera Communications




Volume 12, Number 5, October 2020

Md. Faisal Ahmed, *Student Member, IEEE*  
 Moh. Khalid Hasan, *Member, IEEE*  
 Md. Shahjalal, *Student Member, IEEE*  
 Md. Morshed Alam, *Student Member, IEEE*  
 Yeong Min Jang, *Member, IEEE*



DOI: 10.1109/JPHOT.2020.3017642

# Experimental Demonstration of Continuous Sensor Data Monitoring Using Neural Network-Based Optical Camera Communications

Md. Faisal Ahmed , *Student Member, IEEE*,  
Moh. Khalid Hasan , *Member, IEEE*,  
Md. Shahjalal , *Student Member, IEEE*,  
Md. Morshed Alam , *Student Member, IEEE*,  
and Yeong Min Jang , *Member, IEEE*

Department of Electronics Engineering, Kookmin University, Seoul 02707, Korea

DOI:10.1109/JPHOT.2020.3017642

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see <https://creativecommons.org/licenses/by/4.0/>

Manuscript received August 8, 2020; accepted August 14, 2020. Date of publication August 18, 2020; date of current version September 15, 2020. This work was supported in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-0-01396) supervised by the IITP (Institute for Information & communications Technology Promotion), in part by the Institute for Information and Communications Technology Promotion (IITP) Grant through the Korea Government (MSIT) under Grant 2017-0-00824, and in part by the Development of Intelligent and Hybrid OCC-LiFi Systems for Next Generation Optical Wireless Communications. Corresponding author: Yeong Min Jang (email: yjang@kookmin.ac.kr).

**Abstract:** This technical paper presents an optical camera communication system by using data as input collected from a sensor that continuously garners humidity and temperature information from the environment. A small light-emitting diode (LED) and an LED array are used as a transmitter and a low-speed rolling-shutter camera as a receiver. A modulation scheme standardized in IEEE 802.15.7, referred to as variable pulse-width modulation, is used to encode the data bits. The proposed scheme is flicker-free in different frequencies and applicable in both static and mobile scenarios. A neural network is designed for LED detection and improving the bit-error-rate in mobile scenarios. A new method based on the region-wise comparison and an existing method is used to remove the interference and noise generated from the neighboring light sources, respectively. The decoding procedure is performed and analyzed in Python 3.7. A data rate of 1.02 kbps is achieved using the single LED, which is further augmented to 64 kbps by using an  $8 \times 8$  LED dot matrix.

**Index Terms:** Optical camera communications (OCC), variable pulse width modulation (VPWM), image processing, neural network (NN), rolling-shutter effect, flicker-free communication.

## 1. Introduction

Optical camera communications (OCC) operates in infrared or visible bands and exploit an image sensor as the receiver and light-emitting diodes (LEDs) as the transmitter. OCC offers several advantages such as lower power consumption, low cost, and high security. The worldwide availability of camera-mounted devices makes OCC immensely potential. In addition, OCC is less susceptible to noise and interferences [1]. However, the major limitation of OCC is that it operates in low data rate. The main reason is the low sampling rates of current commercial cameras along with out of

focus effect, unsteady frame rate, and random block [2]. The maximum data rate using OCC is circumscribed within a few kbps; consequently, OCC is suitable for low-rate applications such as sensing, health care, localization, and device-to-device communications [3], [4]. The transmission range using OCC depends on the camera parameters along with LED size. The most significant things to be considered while using OCC are the channel characteristics, synchronization between the camera and LED, data transmission pattern, motion capture, modulation, and proper applications.

Several OCC systems are developed by researchers over the last decade. The systems are mainly implemented on a smartphone or a computer. A smartphone camera-based OCC system is implemented in [5] that achieve a communication distance of 7.5 m with a large LED (several cm in diameter). However, the work didn't consider motion capture and achieved a data rate of several bps. The motion-capture issue is addressed in [6] using simultaneous camera tracking. The literature explained camera tracking for localization and used a high-speed fish-eye camera, also received the signal at the sampling rate of 48 kHz. However, in lower frequencies, flickering problem appeared. Most OCC implementations in the literature used low frequencies, and mitigated the flickering problem using under-sampling [7] and multiple-level modulation [8], [9] techniques. It is worth noting here that flicker-free communication offers the advantage of integrating the OCC system with the conventional light sources [10], [11]. On the other hand, until now, researchers have only used large LEDs and LED arrays. Literatures [1], [2], [12] showed the challenges using multiple LED array and large LEDs. However, many applications require small LEDs (several mm of diameter) such as health monitoring [13] and valve position monitoring in industries.

Very few works considered small LEDs to implement communication using camera. The detection and recognition of the small LEDs of an LED array in both mobile and static environments are challenging. On the other hand, the BER will be substantial in mobile conditions. Therefore, considering these limitations, in this paper, we propose an OCC system using an LED (3 mm outside diameter) and an LED array (comprising  $8 \times 8$  dot matrix) as transmitter and a low-frame rate camera as receiver. To prove the feasibility of our proposed system in a real-world scenario, humidity and temperature data, unceasingly collected from an AM2302 Sensor as the sensing information, are transmitted using the LED, and the camera receives the data concurrently. A standardized modulation scheme, called variable pulse-width modulation (VPWM), is adopted and updated to mitigate the flickering problem while using offset pulse width.

We develop a neural network (NN) at the receiver for accurately detecting the LED in both static and mobile conditions. In existing literature, an LED array is detected as a single LED, however, each LED of the LED array can be detected using the proposed NN technique. The NN can classify large image datasets with remarkable characteristics in several spatial layers and automatically learn from data through backpropagation. Different NNs have been presented in the literature [14], [15] for the detection of the LED array and to support mobility. However, they simulated the mobile scenario by moving a finger continuously, blocking the particular LEDs [16], [17]. The error rate was high using those methods. In our proposed scheme, we have simulated the mobile scenario by moving the camera itself and achieved an excellent BER. The whole system is implemented, and its performance is analyzed in Python 3.7. In brief, the main contributions of this paper are listed as follows:

- To efficiently detect the LED, and each specific LED in an  $8 \times 8$  dot matrix in both static and mobile scenarios, an NN architecture is developed.
- In mobile condition, the LED images experience deformation and vector displacement, therefore, increasing the BER. Consequently, a technique based on NN, referred to as feature matching, is proposed to improve the BER performance.
- A new method based on region-wise comparison is proposed and implemented to mitigate the interference generated from the neighboring LEDs.

The rest of the paper is organized as follows: Section 2 describes the proposed system. Section 3 provides experimental results and a discussion of the research findings. Finally, our work is concluded in Section 4.



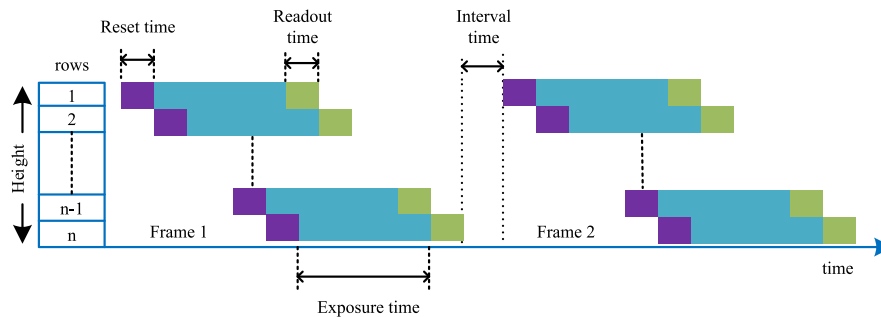


Fig. 4. The read-out architecture of the image sensor using the rolling-shutter effect.

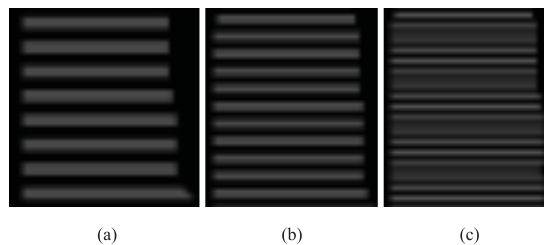


Fig. 5. Stripe pattern using different frequencies due to the rolling-shutter effect in the camera: (a) 500 Hz; (b) 1,000 Hz; and (c) 2,000 Hz.

transitions to generate a pulse. The acknowledged pulse of each bit is separated using the time signal. If the difference of the indices is more than four, which means that the temporal separation is longer than a certain period, then the corresponding bit is set for the relevant data byte. Once some bits are decoded, the process of calculating the checksum, which is the sum of the first four data bytes, is truncated to eight bits. If this equals the fifth byte, the transmission is correct and the function returns data as the humidity and temperature. If the checksum is incorrect, the function returns values that are wrong to allow checking in the calling program. The sensor operation is controlled with a microcontroller to assure the correct value with minimum error. Finally, the sensor data is continuously transmitted using the OCC system.

## 2.2 Principle

We have designed the OCC system using a small LED and an LED array as transmitter and a low-frame rate camera (30 fps) as receiver. We have developed an NN to effectively detect the LED(s) in static and mobile scenarios. We have simulated the mobile environment by moving the camera manually. In the existing literature, the BER is high while considering mobility in OCC [14]. We have applied a technique, called Feature Matching, to reduce the BER. The NN employment for the detection of LED(s) and performance enhancement in mobile scenarios are explained in Section 2.4. In this section, we will explain the basic operating principle of the developed OCC system.

Fig. 1 shows the overall structure of the implemented OCC system, where the input data collected from the sensor and output data retrieved from the receiver are depicted. Moreover, the data pattern of transmitted signals is also demonstrated in binary form. The string and character data produce 1 byte of the binary, which is decoded in the receiver and converted using ASCII code for each output data. Fig. 1 also shows the dark and bright stripes generated in the image sensor due to the rolling-shutter effect.

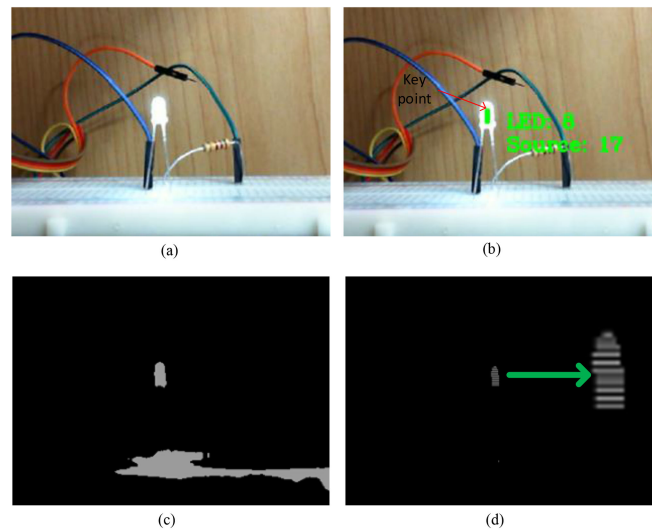


Fig. 6. (a) Exact LED image during data transmission. (b) detection and classification of LED using NN. (c) LED image with noise. (d) stripe pattern after filtering out the noise.

**2.2.1 Transmitter** The transmitter comprises two primary parts such as a AM2302 sensor and a microcontroller with the LED(s). Each data structure (e.g., integer, string, character, etc.) is defined with a symbol, and all types of data are converted into binary. This preprocessing is done before the modulation process. While implementing the LED array, the serial data is converted to parallel for assigning data to each LED before modulation.

The synchronization between the transmitted symbol and frame is necessary for error-free transmission. We have used VPWM to receive data from the LED. In conventional PWM, 4 duty cycles are considered, and the pulse train of data bits in the continuous 0% and 100% duty cycles cause flickering. For this reason, we have omitted the two duty cycles and varied the width of the 'ON' part of the third duty cycle while keeping the 'OFF' part unchanged. As shown in Fig. 2, the two duty cycles represent the digital '0' and '1,' respectively, consequently mitigating any flickering generated due to a long string of '0' and '1'. Each data packet includes three parts: header, data bit, and footer. Both header and footer are defined as 11100 and 10100, respectively; therefore, the data packets can be separated easily from each other. The LED blinking frequency is set to 0.5, 1, and 2 kHz to transmit the information.

**2.2.2 Receiver** A Microsoft 6CH-00001 LifeCam Cinema webcam is used as receiver to receive data from the LED(s). The whole reception process is depicted in Fig. 3. The blocks 2-4 in Fig. 3 are described in Sections of 2.3 and 2.4. The image sensor applies a row-by-row capturing mechanism, and resultantly, bright and dark stripes are generated. Fig. 4 depicts how the image sensor reads every row of pixels using the rolling-shutter effect. The exposure time controls the intensity of the bright and dark stripes, in addition to background image removal, to retrieve the original transmitted signal. Python 3.7 is used for data processing taking the rolling-shutter effect into account.

The stripes are captured in  $640 \times 480$  resolution and as discussed earlier, the bright stripes have different widths due to applying VPWM. In our experiment, three different exposure times are used such as 0.65 ms, 1.25 ms, and 2.5 ms. By changing the exposure time, the intensity of the bright stripes is controlled. Initially, the bright and dark regions are measured by the number of rows in the image sensor. The width of the stripes varies with frequency, as shown in Fig. 5. It can be observed that the width of the stripes increases when the frequency is low. Fig. 5 also shows that the width of the bright stripes is different in a specific frame. The binary bits are extracted by estimating the number of rows of the bright stripes and variation of width.

After the image sensor receives the optical signals, each image is processed in several steps. First, a Gaussian blur function is used to identify the bright pixels using a  $5 \times 5$  matrix concentration. Then, the image is converted into a grayscale format to remove the background. Next, a threshold is used based on the brightness of the stripes to recognize the image as binary patterns. Afterward, a dilatation process is performed with kernel 10 iteration and  $3 \times 3$  matrix concentration. Thenceforth, to recognize the stripe pattern more precisely, the resolution of the image is increased using a function. Then, the sequence of the average grayscale value for each stripe is generated. After that, a certain threshold is applied to decode the original sequence of bits. In this instance, the threshold is defined after taking an average of the maximum and minimum value of the normalized intensity. Finally, data is serially shifted in an array and every 8 bits of the binary value is assigned to represent the character and string in the text. When a symbol is detected, the integer value representing the value of humidity and temperature is decoded.

**2.2.3 Channel Capacity** The received signal could be affected by noise, such as ambient light sources and interferences from other LEDs. The received signal at the receiver can be represented as

$$R = (1 - \chi)A\xi h_{l,c}T_l + \eta, \quad (1)$$

where  $T_l$  is the transmitted signal,  $\xi$  denotes the optical to electrical conversion efficiency,  $\chi$  represents the distortion factor of the image at the camera,  $A$  is the area of the LED object in the received image, and  $\eta$  represents the white Gaussian noise matrix with zero-mean. The term  $h_{l,c}$  is the visible light channel response that can be represented as [18]

$$h_{l,c} = g_r L_m d_{c,l}^{-2} \cos(\delta_i), \text{ for } \delta_i < \delta_{FOV}, \quad (2)$$

where  $g_r$  is defined as the optical filter gain of the camera,  $d_{c,l}$  represents the Euclidean distance between the LED and the camera, and  $\delta_i$  denotes the incidence angle with respect to the camera axis. This incidence angle should be less than the field-of-view (FOV) of the LED ( $\delta_{FOV}$ ) to successfully detect the LED.  $L_m = [(m + 1)/2\pi] \cos^m(\psi_{ir})$  signifies the Lambertian radiant intensity of the optical channel, where  $m$  is the Lambertian emission index and  $\psi_{ir}$  represents the irradiance angle of the light source. Signal-to-interference-noise ratio (SINR) at the receiver can be expressed using the following Equation

$$\text{SNIR} = \frac{(1 - \chi)A\xi h_{l,c}T_l}{\sum_{i=1}^{N-1} A\xi h_{l,c}T_l + \eta}, \quad (3)$$

where  $N$  represents the number of surrounding LEDs within the camera FOV. The capacity of such an OCC system can be derived from Shannon's capacity formula. According to Ashok *et al* [19], the capacity of a camera-based communication depends on the deterministic nature of the perspective distortions and the Additive white Gaussian noise characteristics of the communication channel and is expressed as

$$C = F_r(W_s \log_2(1 + \text{SNIR})), \quad (4)$$

where  $F_r$  denotes the sampling rate of the camera and  $W_s$  is the spatial bandwidth that represents the number of achievable bits per image frame.

### 2.3 Noise and Interference Subtraction

Background lights and neighboring LEDs can create noise and interference in the image sensor, respectively. We have developed a function in Python to find the areas in which the source LED and the interfering light sources appear. The decoding process becomes challenging if the intensity of the background lights is higher or equal to that of the LED. Otherwise, in reasonable conditions, the reconstruction of the signal is possible using a proper combination of a filter and equalizer. A testbed platform is illustrated in Fig. 6(a). As shown in Fig. 6(b), the detection process is performed by selecting a key point of the light source and subsequently, the width and height of the LED are included constructing a boundary region. As shown in Fig. 6(c), the noise generated by the

reflection of the surrounding light source is illustrated in the binary image. The LED stripe pattern, after filtering out the noise, is shown in Fig. 6(d).

On the other hand, interference occurs if a LED light source unexpectedly appears inside the image sensor FOV. In particular, after detection, several frames are captured and analyzed. If the stripe pattern of a specific area are changed in the subsequent frames, it is recognized as the source LED. Otherwise, it is categorized as the interfering LED and is filtered out subsequently from the image frame.

### 2.4 Neural Network Model

An NN model is designed to detect the LED and to support mobility during the detection process. When a frame is processed, the LED position in the image frame is compared with the position measured in the previous frames. If the position is changed, then the scenario is defined as mobile. For the LED detection in this scenario, we have trained 70% of total acquired images using `darknet_no_gpu.exe` and OpenCV in Python. In the training process, we used an LED type called `coco_obj.name` where `yolov3.weights` was used as the weight configuration. Another configuration, called `yolov3-tiny_obj.cfg` is used for labeling of the detected LED image to find its exact position. After detection in the moving scenario, the LED image can experience two issues,

- the image can be deformed by inclining the LED at any angle and
- some vectors of the LED image can be displaced.

As a solution, we have used a technique, referred to as feature matching. In this technique, first, each  $5 \times 5$  kernel matrix in the image patches are checked using NN regression and compared with reference stripes. The angle yielded due to the deformation is updated using the original inverse deformation, and the displaced vector field is partially reconstructed by filtering. Afterward, every point of the stripes is resampled using spatial transformation to produce a warped image. Using backpropagation connected with NN regression, the steps are repeated in a loop until the original stripes are completely reconstructed. The whole procedure of feature matching is depicted in Fig. 7.

Due to the movement, the stripes can overlap with each other, consequently increasing the BER. To remediate the issue, we have trained 60% of the total transmitted bits. The trained sets of bits are combinations of character, string, integer, and symbol. If any stripe of the testing image experiences an overlap, it is resolved using the trained dataset. For example, if the NN encounters a sequence with an altered bit, it uses probability to predict the sequence is how much similar to the pre-trained sequence. If the probability exceeds a pre-defined threshold, the sequence is recognized as identical to the pre-trained sequence and the altered bit is replaced accordingly. Besides, in terms of decoding numerical data specifically, the symbol defined before the data is utilized to predict the probability. It is worth noting that the higher the probability (defined in percentage), the faster the recovery from the error.

## 3. Experimental Results

In this section, we have assessed the performance of the proposed OCC system in both static and moving scenarios. In a static scenario, the variation in detected bits is shown against distance. The BER is measured in moving scenarios, and how it is improved with the employment of the NN is illustrated. Finally, the achieved data rate is compared with the existing implemented OCC systems that also used PWM. Table 1 lists the parameters used for the implementation.

As discussed in Section 2.2.1, each data packet consists of header, footer, and payload. The payload sequence can be long, so that a single frame may not contain a whole packet. When the header is detected, the bits are received as payload until the footer is detected. In a certain frame, the width of a bright stripe is measured in terms of pixel rows to detect the frequency. Fig. 8(a) represents the variation of widths of the bright stripes as normalized intensity levels in a specific frame. Afterward, the total number of bits received in that frame is estimated. We have used three frequencies to demonstrate how the number of detected bits varies with distance in a



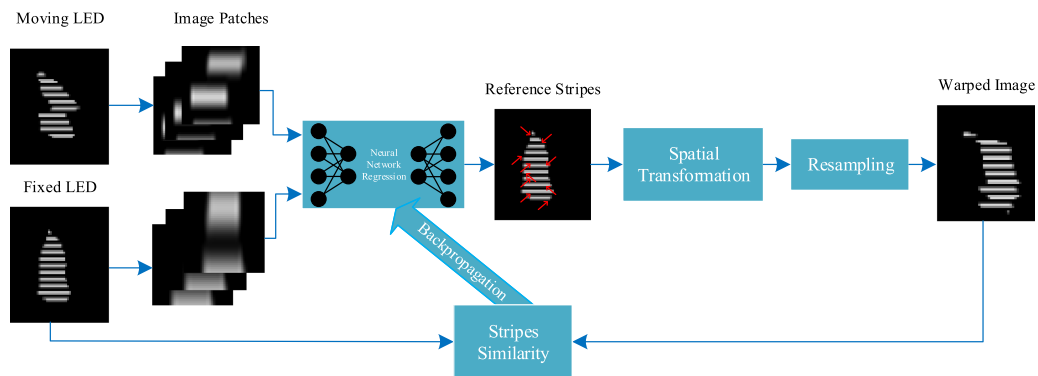


Fig. 7. Neural network-based feature matching for reconstructing the original stripe pattern.

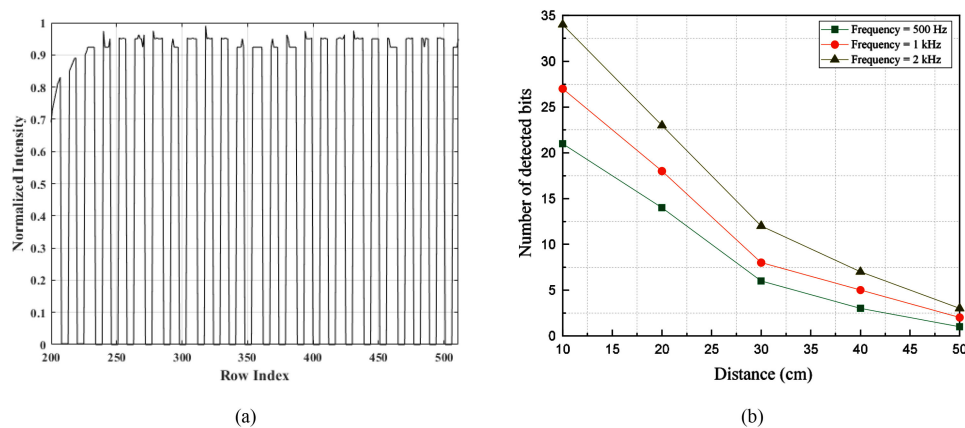


Fig. 8. (a) The normalized intensity concerning the row index and (b) the effect of frequency variation in bit detection with changing distance.

TABLE 1  
Experimental Parameters

Parameters	Values
Maximum distance	50 cm (single LED) & 1 m (LED matrix)
Camera frame rate	30 fps
Exposure time	2.5, 1.25, and 0.65 ms
Image resolution	640 × 480
LED type	3 mm Warm White LED & 8 × 8 dot matrix
Sensor type	AM2302/DHT22
Frequency	0.5, 1, and 2 kHz
Number of bits per frame	10-35

static scenario. As shown in Fig. 8(b), the maximum number of detected bits is found as 34 when the frequency is 2 kHz using a single LED and VPWM. The data rate estimated in this case is 1.02 kbps, which can be augmented using higher frequencies.

The exposure time has an important impact on data decoding. If the exposure time is high, then the effect of noise and interference will be considerably increased. In this case, the BER will be high, which can be resolved using the NN. When the NN is integrated with the system, existing image

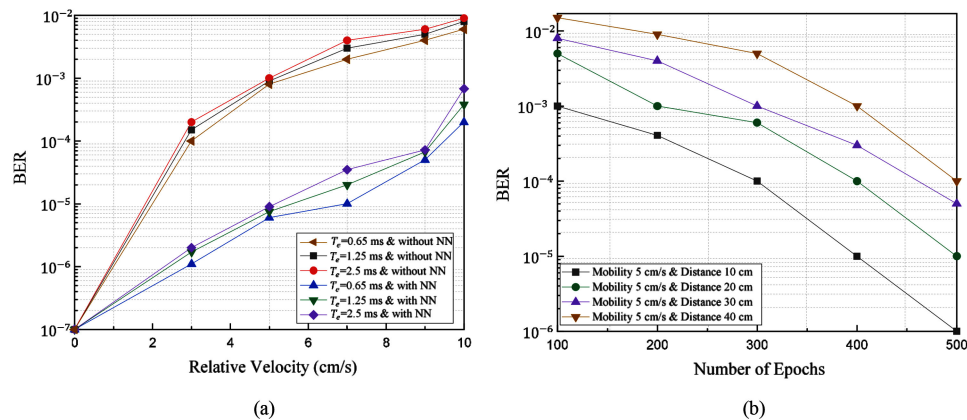


Fig. 9. BER performance with (a) changing receiver velocity and exposure time and (b) integration of NN and variation of the number of iteration to reform the original image.

TABLE 2  
Performance Comparison With Existing OCC Implementations Based on PWM

Data rates	Reference	Distance (m)	Platform	Scheme	LED type
9 kbps	[8]	0.1	Smartphone	PWM	$3 \times 3$ matrix
0.084 kbps	[20]	4	PC, MATLAB	PWM	Large LED
0.62-1.35 kbps	[21]	1.5-5.5	Smartphone, java	Hybrid OOK-PWM	Large LED
0.3-1.5 kbps	[22]	0.5	PC, LabVIEW	PWM	Large LED
7.5-64 kbps	Our work	0.1-0.5	PC, Python	VPWM	$8 \times 8$ matrix

data is trained using darknet in Python for both the static and moving cases. Alongside the training algorithm, we have used feature matching considering the image deformation and displacement vector field in each iteration. In Fig. 9(a), it can be observed how the BER is changed with velocity before and after applying the NN. It can also be seen that the BER declines when the exposure time is decreased. A BER below  $10^{-7}$  is found for the static condition. However, it is increased to around  $8 \times 10^{-3}$  for a velocity of 10 cm/s. Using the NN system, the BER is decreased to  $2 \times 10^{-4}$  when the exposure time is 0.65 ms. Furthermore, when mobility is constant, distorted stripes can be renovated by increasing the number of iterations. As shown in Fig. 9(b), a BER of around  $5 \times 10^{-5}$  can be achieved when the distance is 30 cm, and the number of iterations is 500, which can be increased to around  $8 \times 10^{-3}$  when the number of iterations is reduced.

The data rate can be augmented dramatically by using an LED array instead of a single LED. If we increased the number of LEDs, the number of stripes in each frame would also be multiplied by the same number. As a result, the data rate will be a multiple of the number of LEDs. Fig. 10(a) shows the transmitter consisting of an  $8 \times 8$  array of LEDs. The LED dot matrix after detecting using the NN and the corresponding stripe patterns are also illustrated in Fig. 10(a). The interference due to the neighboring LED and the noise generated due to the reflection of the LED matrix in the plain surface are removed using the NN. The data-rate performance with increasing distance is depicted in Fig. 10(b), where it can be observed that the data rate is multiplied by 64 when the LED array is used. The results are verified for both static and moving cases.

To the best of authors' knowledge, the proposed continuous sensor data monitoring system using OCC has not been considered in the existing literature. We have used VPWM to implement the OCC scheme using a low-speed camera (30 fps) and achieved a high data rate compared to the previous schemes that used the same type of camera. In Table 2, we compared our findings with existing OCC systems that used the similar modulation scheme.

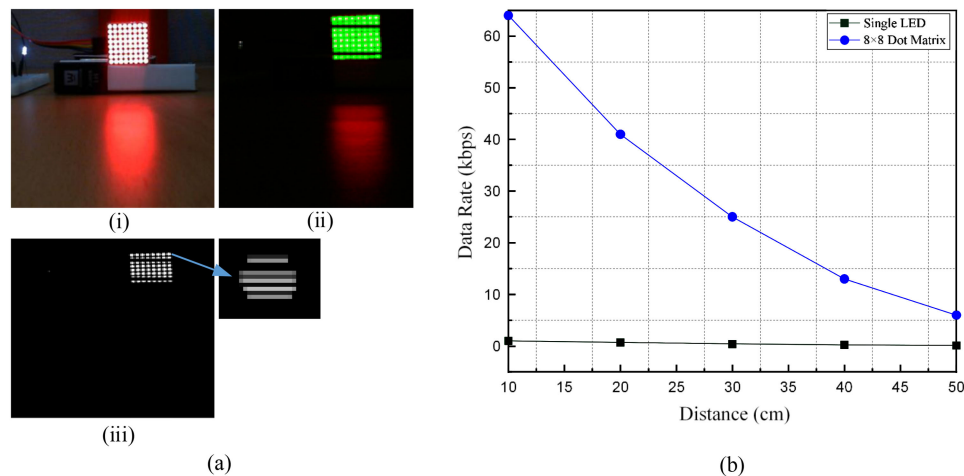


Fig. 10. (a) Demonstration with an  $8 \times 8$  LED matrix where (i) shows the transmitter, (ii) shows the detection of the LEDs using NN, and (iii) shows the stripe pattern, and (b) data-rate performance with changing distance.

## 4. Conclusions

In this paper, an OCC system comprised of a small LED or an LED array as a transmitter and a low-speed rolling-shutter camera as receiver is demonstrated. The data transmitted to the camera are garnered from an AM2302 sensor that continuously collects humidity and temperature data from the environment. VPWM is used to modulate the LED(s) and to achieve flicker-free transmission at different frequencies. The scheme is implemented in mobile and static scenarios, and an NN model is developed to effectively detect the LED in both conditions. However, a substantial amount of BER is observed in the mobile scenario, which is further improved by employing the NN based on feature matching. A method based on region-wise comparison and an existing method are also employed to mitigate the interference and noise generated from the neighboring light sources, respectively. The whole decoding procedure is performed in Python 3.7. The proposed system achieved 1.02 kbps using the single LED and 64 kbps using the  $8 \times 8$  LED array, which outmeasure the previous similar works. However, the transmission range was small compared to some of the existing systems, and its augmentation will be considered in our future works.

## Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable suggestions.

## References

- [1] W. Liu and Z. Xu, "Some practical constraints and solutions for optical camera communication," *Philos. Trans. Roy. Soc.*, vol. 378, no. 2169, 2020.
- [2] N. Saeed, S. Guo, K. H. Park, T. Y. Al-Naffouri, and M. S. Alouini, "Optical camera communications: Survey, use cases, challenges, and future trends," *Physical Commun.*, vol. 37, pp. 1–17, 2019.
- [3] M. Yoshino, S. Haruyama, and M. Nakagawa, "High-accuracy positioning system using visible LED lights and image sensor," in *Proc. IEEE Radio Wireless Symp.*, 2008, pp. 439–442.
- [4] M. Shahjalal, M. T. Hossan, M. K. Hasan, M. Z. Chowdhury, N. T. Le, and Y. M. Jang, "An implementation approach and performance analysis of image sensor based multilateral indoor localization and navigation system," *Wireless Commun. Mobile Comput.*, vol. 2018, no. 8, 2018.
- [5] M. Shahjalal, M. K. Hasan, M. Z. Chowdhury, and Y. M. Jang, "Smartphone camera-based optical wireless communication system: Requirements and implementation challenges," *Electronics*, vol. 8, no. 8, 2019, Art. no. 913.
- [6] Y. Nakazawa, H. Makino, K. Nishimori, D. Wakatsuki, and H. Komagata, "Indoor positioning using a high-speed, fish-eye lens-equipped camera in visible light communication," in *Proc. IEEE Int. Conf. Indoor Positioning Indoor Navigation*, 2013, pp. 1–8.

- [7] P. Luo, Z. Ghassemlooy, H. L. Minh, H. Tsai, and X. Tang, "Undersampled-PAM with subcarrier modulation for camera communications," in *Proc. Opto-Electron. Commun. Conf.*, 2015, pp. 1–3.
- [8] J. Lee, S. Yang, and S. Han, "Optical pulse width modulated multilevel transmission in CIS-based VLC," *IEEE Photon. Technol. Lett.*, vol. 29, no. 15, pp. 1257–1260, Aug. 2017.
- [9] V. P. Rachim and W. Chung, "Multilevel intensity-modulation for rolling shutter-based optical camera communication," *IEEE Photon. Technol. Lett.*, vol. 30, no. 10, pp. 903–906, May 2018.
- [10] B. W. Kim and S. Jung, "Novel flicker-free optical camera communications based on compressed sensing," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1104–1107, Jun. 2016.
- [11] R. Zhang, W.D. Zhong, D. Wu, and K. Qian, "A novel sensor fusion based indoor visible light positioning system," in *Proc. IEEE GC Workshops*, 2016, pp. 1–6.
- [12] M. Z. Chowdhury, M. T. Hossan, A. Islam, and Y. M. Jang, "A comparative survey of optical wireless technologies: Architectures and applications," *IEEE Access*, vol. 6, pp. 9819–9840, 2018.
- [13] M. K. Hasan, M. Z. Chowdhury, M. Shahjalal, and Y. M. Jang, "Real-time healthcare data transmission for remote patient monitoring in patch-based hybrid OCC/BLE networks," *Sensors*, vol. 19, no. 5, 2019, Art. no. 1208.
- [14] T. S. Rajendra, Z. Stanislav, G. Zabih, "Performance evaluation of neural network assisted motion detection schemes implemented within indoor optical camera based communications," *Opt. Express*, vol. 27, no. 17, pp. 24082–24092, 2019.
- [15] S. R. Teli, W. A. Cahyadi, and Y. H. Chung, "Performance evaluation of neural network assisted motion detection schemes implemented within indoor optical camera based communications," *Opt. Express*, vol. 27, no. 17, 2019, Art. no. 24082.
- [16] S. R. Teli, W. A. Cahyadi, and Y. H. Chung, "Optical camera communication: Motion over camera," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 156–162, Aug. 2017.
- [17] A. Sewaiwar, S. V. Tiwari, and Y. H. Chung, "Visible light communication based motion detection," *Opt. Express*, vol. 23, pp. 18769–18776, 2015.
- [18] M. K. Hasan, M. Z. Chowdhury, M. Shahjalal, V. T. Nguyen, and Y. M. Jang, "Performance analysis and improvement of optical camera communication," *Appl. Sci.*, vol. 8, no. 12, 2018, Art. no. 2527.
- [19] A. Ashok, S. Jain, M. Gruteser, N. Mandayam, W. Yuan, and K. Dana, "Capacity of screen-camera communications under perspective distortions," *Pervasive Mobile Comput.*, vol. 16, pp. 239–250, 2015.
- [20] Y. Imai, T. Ebihara, K. Mizutani, and N. Wakatsuki, "Performance evaluation of high-speed visible light communication combining low-speed image sensor and polygon mirror in an outdoor environment," in *Proc. Int. Conf. Ubiquitous Future Netw.*, 2016, pp. 51–55.
- [21] J. Hao, Y. Yang, and J. Luo, "CeilingCast: Energy efficient and location-bound broadcast through LED-camera communication," in *Proc. IEEE Int. Conf. on Computer Commun.*, 2016, pp. 1–9.
- [22] T. Nguyen, C. H. Hong, N. T. Le, and Y. M. Jang, "High-speed asynchronous Optical Camera Communication using LED and rolling shutter camera," in *Proc. Int. Conf. Ubiquitous Future Netw.*, 2015, pp. 214–219.