applied sciences

MDPI

Article

DeepCCB-OCC: Deep Learning-Driven Complementary Color Barcode-Based Optical Camera Communications

Min Tae Kim [1] and Byung Wook Kim [2,*]

[1] Department of Smart Manufacturing Engineering, Changwon National University, Changwon 51140, Korea
[2] Department of Information and Communication Engineering, Changwon National University, Changwon 51140, Korea
* Correspondence: bwkim@changwon.ac.kr

Abstract: Display-to-camera (D2C) communications has emerged as a key method for next-generation videos that offer side information to camera-equipped devices during normal viewing. This paper presents Deep learning-driven Complementary Color Barcode-based Optical Camera Communications (DeepCCB-OCC), a D2C system using multiple deep neural networks built for imperceptible transmission and reliable communication in a D2C link. DeepCCB-OCC takes advantage of a the You Only Look Once (YOLO) model to provide seamless detection of a color barcode area in electronic displays. To identify transmitted color barcode symbols in the received image, we define various color barcode patterns caused by the synchronization jitter between the camera and the display. Then, DeepCCB-OCC incorporates convolutional neural network (CNN) models to accurately detect the pilot and data symbols in the transmission packets, regardless of the various D2C environments. Experiments with a commercial monitor and a smartphone demonstrate that DeepCCB-OCC outperforms the conventional CCB-OCC system from various distances and angles of a D2C link. The experiment results prove that, when the alignment angle was 20 degrees at a distance of 90 cm between the display and the camera, the proposed scheme achieved approximately 79.1 bps, which showed a performance improvement of 14.1% compared to the existing technique.

Keywords: display-to-camera communications; Complementary Color Barcode-based Optical Camera Communications; deep neural networks

check for updates

Citation: Kim, M.T.; Kim, B.W. DeepCCB-OCC: Deep Learning-Driven Complementary Color Barcode-Based Optical Camera Communications. *Appl. Sci.* **2022**, *12*, 11239. https://doi.org/10.3390/app122111239

Academic Editor: Sungyoon Jung

Received: 18 September 2022
Accepted: 3 November 2022
Published: 6 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, video content has become more important than ever for a wide range of businesses, advertisements, and cultures. Unlike traditional marketing methods such as billboards or direct mail, video content can offer curated information with a more personalized and immersive feel. As the demand for such video content increases, there is demand for a new paradigm that simultaneously conveys additional information while watching a video with carry-on devices, e.g., smartphones and mobile tablets. One intuitive solution for offering this service is to use traditional radio frequency (RF)-based communications. However, wireless connections require explicit RF devices, and the broadcasting characteristics of RF signals cause information spamming or leakage to undesired devices. As a viable candidate that can solve this issue, display-to-camera communications (D2C) [1,2], which encodes information in screen imagery that can be decoded with a camera receiver, is attracting attention. Thanks to the high refresh rates ($\geq$120 Hz) in off-the-shelf electronic displays, it is possible to hide data in high-frame-rate videos to cheat the human eye, which has much lower perception rates (40–50 Hz). D2C technology is a type of visible light communications [3–5] that can be applied anywhere in the environment where there are electronic displays and camera sensors, mainly at short communication distances.

The easiest application for the D2C paradigm is quick response (QR) codes [6]. In general, QR codes occupy a small area in the corner of the screen. However, spatial

inclusion of these visual patterns can cause problems; they are unsightly to some viewers, or they impair the perception of key visual information. InFrame and InFrame++ [7,8] were developed to provide D2C communications with imperceptible video artifacts by designing complementary frame compositions and hierarchical frame structures. HiLight [9] uses the orthogonal transparency (alpha) channel to transmit data, and DisCo [10] uses a temporal sequence in a spatial pattern. RainBar [11] and RainBar+ [12] were designed to support flexible frame synchronization and code extraction. SoftLight [13,14] uses a channel coding approach, which adapts the transmission data rate to the link quality. In [15], the Robust and Aesthetic (RA) Code method inserts 2D barcodes into a display to minimize channel loss due to downsampling error and intersymbol interference. In [16], the signal-rich-art code image was proposed, presenting related techniques for counting the number of pattern blocks and for recognition of code patterns. Abe et. al. [17] proposed a technology to insert barcodes using vibrations of colors that cannot be recognized by the human eye. UnseenCode [18] is a barcode insertion and extraction method using cross-correlation of color images. In [19], ChromaCode was proposed, which is an adaptive embedding method that changes according to the brightness and texture of the display. Unlike conventional schemes, Complementary Color Barcode-based Optical Camera Communications (CCB-OCC) [20] is a method to send symbols with carefully designed complementary color pairs that are perceptible to the human eye as a white color bar, but the individual color pattern is detectable by a camera. However, the conventional CCB-OCC system detects the color barcode area using a differential image, but it may fail to accurately detect the area of the barcode because it is greatly affected by the camera distance and angle. In addition, color value extraction using heuristic histogram analysis cannot guarantee robust performance in packet synchronization and data decoding.

In this study, we propose Deep learning-driven CCB-OCC (DeepCCB-OCC), an imperceptible D2C communications system that supports reliable data transmission. To detect seamless barcode image regions in D2C links, a You Only Look Once (YOLO)-based barcode area detection model is implemented. In addition, convolutional neural network (CNN) models can extract data embedded in color barcodes to perform synchronization of packets and data decoding. DeepCCB-OCC can provide reliable communications, owing to the generalization performance of inherent deep neural network (DNN) properties, and thus, the root cause of the unreliability in the conventional scheme can be solved. Experiments with a commercial display and a smartphone demonstrate that the proposed DeepCCB-OCC system boosts the data rate performance in various D2C scenarios.

In summary, the main contributions are as follows:

- We introduce deep learning-aided color barcode extraction and data decoding methods for CCB-OCC systems. We achieve reliable data transmission by a YOLO-based color barcode extraction mechanism together with CNN classification models for packet synchronization and data decoding.
- We design a novel color barcode extraction scheme with the YOLO-based structure and the post-processing process, which accounts for real-time detection and generalization performance, and ensures robust color barcode extraction over the full frame.
- In our experiments, we confirm that various types of color barcodes can be produced due to the time offset problem between the display and the camera. By analyzing these color barcodes, the form of color barcodes collected when shooting a display with a 60 Hz refresh rate at 30 fps are defined as four patterns and used for packet synchronization and signal detection.
- We design CNN models to classify color barcodes for packet synchronization and data decoding. Through this, the problem of instability and generalization of the existing histogram analysis method is solved, and proposed models achieve higher data rates than the previous work in various distance and alignment angle scenarios.

In the remainder of this paper, related research on deep learning-based approaches are introduced in Section 2. The DeepCCB-OCC system model is described in Section 3. The color barcode extraction using deep learning and post-processing methods are presented in

Section 4. Then, the various types of color barcode patterns, and deep learning models for packet synchronization and data decoding are given in Section 5 and Section 6, respectively. In Section 7, the models' training results and experimental results are presented to show the feasibility of the proposed method, followed by the conclusion in Section 8.

2. Related Works

In recent years, tremendous interest in deep learning technology has been increasing in many fields [21]. Among the various deep learning models, the most established algorithm is the CNN, which has shown surprising results in ImageNet object recognition competitions and has been able to replace traditional computer vision methods [22]. From a deep learning perspective, the CNN model can learn image features from raw image pixels through multiple convolutional layers to construct complex nonlinear mappings between inputs and outputs. The lower convolutional layer captures general expressive features such as edges and patters, etc., while the upper layer can learn specific and complex structures. The CNN demonstrates superior performance in various vision problems because it discovers and integrates low/middle/high-level features from the input image by itself and uses them to perform object classification and detection tasks [23]. Therefore, CNN-based deep learning models have been widely utilized in public security, self-driving systems, medical applications, and remote sensing, as it is a field that generates a lot of visual data [24].

Since deep learning technology efficiently learns the representation of data information, recently, deep learning technology has been introduced into a visible light communication (VLC) system that uses a photodetector as a receiver, and many problems of VLC have been successfully solved. In [25], the CNN model is used to simultaneously discovery atmospheric turbulence strength and orbital angular momentum modes in free-space optical (FSO) communication. In [26], the post-equalizer consisting of the long short-term memory (LSTM) neural network is proposed to mitigate impairments in pulse amplitude modulation-based VLC. In [27], the deep learning network for the collaborative constellation design in VLC systems was proposed. In [28,29], a deterministic binarization technique is developed for the design of a deep learning-based transceiver of the on–off-keying (OOK)-modulated VLC system. These studies have validated the power of using deep learning techniques in VLC systems and motivated the construction of a deep learning-based detection framework for OCC systems that need to detect color barcodes.

3. System Model

The system model of the DeepCCB-OCC system is shown in Figure 1. At the transmitting end, the binary data are encoded into a color barcode, and consecutive packets are sent through the display. In order for data to be encoded in a color form, color mapping using a pre-designed color constellation diagram is required. The pilot and data symbols are then mapped to designated color values and consist of data packets. All symbols in the packet are composed of complementary color pairs, and the display shows the symbols composed of complementary color pairs in the designated color barcode area in successive image frames. The receiving end captures the display area using a rolling shutter camera, and the captured image is stored in the image frame buffer. At this time, the color barcode area appears white to the human eye, but continuous color change is observed in the color barcode area. When a color barcode area is detected by using a deep learning model on the stored images, color barcode images are stored in an image list. Then, a deep learning-based pilot symbol detection process for continuous packet synchronization is performed, identifying the start point of each packet. When synchronization is complete, data within each packet are consecutively decoded through the data symbol detector using deep learning.

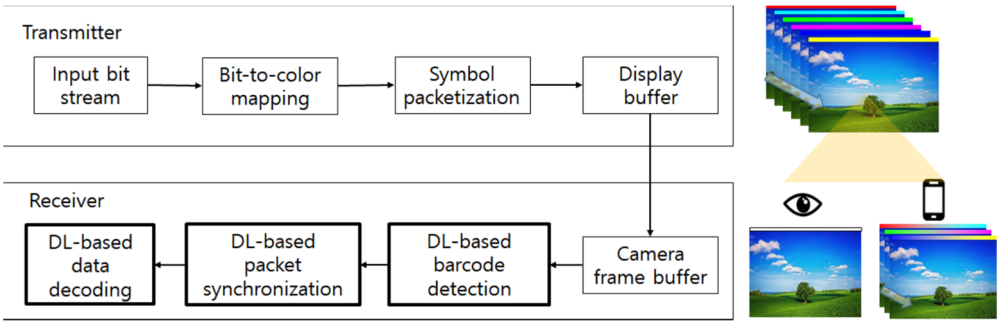


Figure 1. The system model for DeepCCB-OCC.

To transmit information using the DeepCCB-OCC system, it is necessary to match the transmission symbol and the color value. The complementary color relationship of the DeepCCB-OCC system was constructed using the hue circle shown in Figure 2. Primary and complementary colors are located 180 degrees from each other on the hue circle. For primary colors (red, green, and blue), the complementary colors are cyan, yellow, and magenta. When primary and complementary colors are added, the R, G, and B channel values are 255, resulting in white. When the primary and complementary colors appear in succession through the display's fast refresh rate, the human eye sees the combination of the two colors as white. This allows additional information that is unobtrusive to the human eye to be transmitted through the display. As can be seen in the figure, eight symbols constitute the hue circle, and each symbol consists of three bits. Grey coding is used to minimize the bit error rate in the decoding process. Each symbol is expressed as the intensity value of the red (R), green (G), and blue (B) channels, and the color value for each channel for the eight symbols used in CCB-OCC is shown in Figure 2.

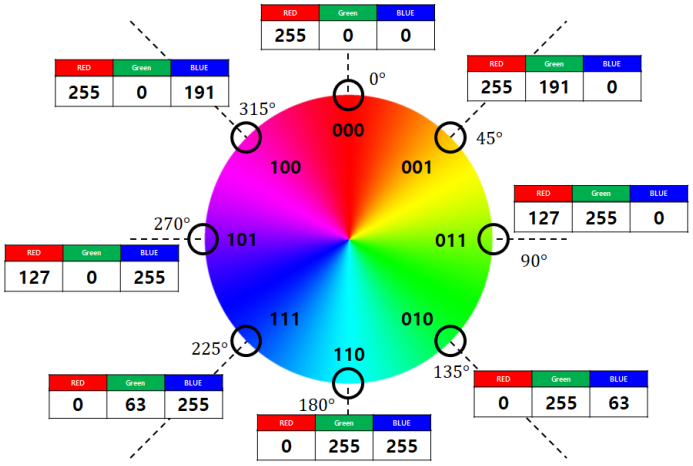


Figure 2. The color constellation for Complementary Color Barcode-based OCC.

When the proposed technique transmits data through the display, color barcodes must be continuously configured so that the primary color and the complementary color are paired. If the R, G, B-encoded symbols are transmitted, the actual data stream is RR'GG'BB', where R, G, and B denote the encoded symbols and R', G', and B' denote the complementary colors. The receiver uses the image captured while filming the display using a rolling shutter camera. Theoretically, when capturing the image at half the speed of the display's output, a color barcode formed by pairs of primary and complementary colors appears as a single captured image. As shown in Figure 1, for example, if the transmitter's screen refresh rate is 60 Hz, and the receiving end captures it at 30 fps, the six transmitted frames (R, R', G, G', B, B') appear at the receiver as three incoming images, RR', GG', and BB', but are perceived as a white bar to the human eye.

Data-encoded color barcode symbols are transmitted in the form of packets, as shown in Figure 3. Transmission packets used in CCB-OCC consist of pilot symbols and data symbols, with each symbol composed of complementary color pairs. The pilot symbol, used for synchronization and channel estimation, is designed as six image frames in the form RR'GG'BB'. Following the pilot symbol, the data are designed as frame pairs consisting of encoded color symbols and their complementary color pairs. Data-encoded image frames are captured by the rolling shutter camera, and the color barcode area in the captured image is detected through a YOLO object detector in the image-processing stage. After that, synchronization and decoding are performed using a CNN model that has been trained on the symbols of the packet.

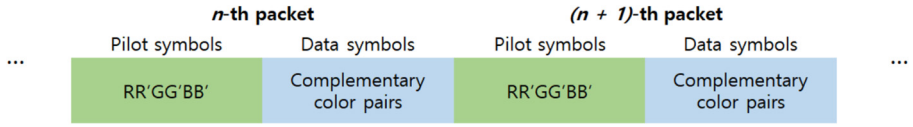


Figure 3. Transmit packet structure of CCB-OCC.

4. Deep Learning-Based Color Barcode Extraction

For the data decoding process in the D2C system, it is necessary to extract only the color barcode area within the captured image on the monitor. In a previous study [20], the color barcode area, extracted from a differential image, resulted in inaccurate results due to the effect of ambient light and the image capture jitter problem. This problem is exacerbated by poor alignment between the display and the camera. Therefore, in this study, we propose a method for color barcode area extraction using YOLO, a real-time object detection model.

Among the various YOLO versions, YOLOv4 was applied in this study for color barcode area extraction. As shown in Figure 4, the YOLOv4 model uses CSPDarknet53 as the backbone structure, while using the YOLOv3 model almost as is, and uses spatial pyramid pooling (SPP) and a path aggregation network (PAN) as the neck structure. In addition, to improve the existing YOLOv3 model, YOLOv4 uses techniques called Bag of Freebies and Bag of Specials. Bag of Freebies is used to improve learning accuracy by increasing training costs. A typical example of this is data augmentation, which uses methods such as Random Erase to add noise to some areas of an image, CutOut to remove some areas, MixUp to mix two images with different labels, and CutMix to cut and paste two images with labels. Bag of Specials is a method to increase prediction accuracy, and various methods are used to enhance specific features, such as expanding the receptive field, introducing an attention mechanism, strengthening the feature integration capability, and selecting the bounding box for the prediction results.

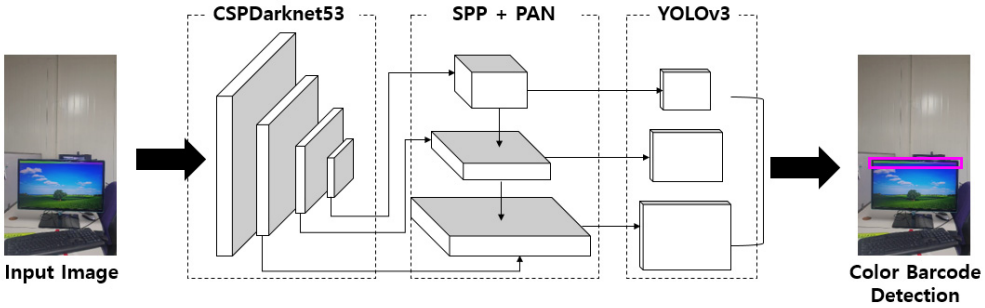


Figure 4. Architecture of YOLO-based color barcode detection.

The input for YOLOv4-based color barcode area extraction is the image captured from the display device that transmits the color barcode. The input image is converted to 512 × 512 pixels, which is the input size for CSPDarknet53. Here, the color barcode area to be extracted in the converted image is significantly reduced. However, CSPDarkent53 used

as a backbone of YOLOv4 has a large receptive field and a large number of parameters, so it shows excellent detection performance even for small objects. In addition, we were able to successfully extract color barcodes of various sizes through the SPP block, which increases the receptive field without affecting the operating speed of the model, and through the PAN, which receives information from different backbone levels. This helps to robustly extract color barcode areas at various distances between the display and the camera.

The ideal form of the received color barcode would be one in which the data-encoded primary and complementary colors are smoothly connected within the color barcode area. However, it is common for actual extracted color barcodes to appear as trapezoids rather than rectangles. This could be due to the shape of and the alignment angle between the display and the camera, causing a large distortion in the captured image. This happened in most capture results, and the distance between the display and the camera makes the distortion even greater, which also affects the size of the color barcode.

The area extracted using YOLO contains ambient noise, in addition to the color barcode, causing performance degradation during synchronization and decoding. Therefore, a process to remove noise is required, but simply reducing the area extracted by YOLO is not effective because the packet area extracted using YOLO is not constant in consecutive frames. For effective noise removal, methods such as differential images and morphological operation are performed in the overall process shown in Figure 5.

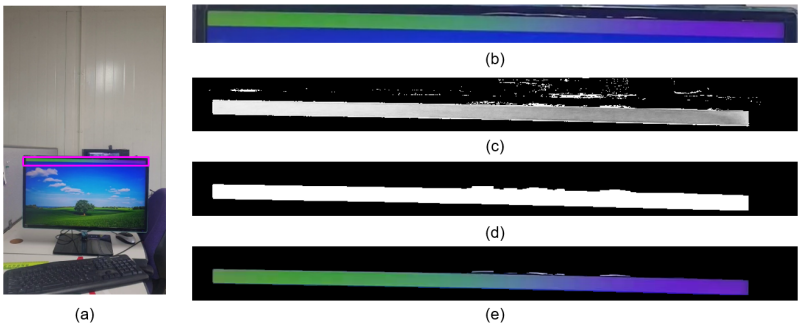


Figure 5. The extraction process of a color barcode area: (**a**) the YOLO result; (**b**) YOLO-based extraction of the color barcode area; (**c**) a color barcode mask using differential images; (**d**) the color barcode mask using a morphology operation; and (**e**) a color barcode with noise removal.

As shown in Figure 5a, the display where packet information is transmitted through the color barcode area is captured, and the YOLO model is trained using data containing annotations of the color barcode area to extract the approximate area of the color barcode from the received display image. The data extracted through YOLO include the x- and y-axis coordinates of the approximate rectangular area of the color barcode for the height and width of the area. As shown in Figure 5b, the extracted color barcode area varies in each incoming video frame and contains ambient noise. To minimize the ambient noise that affects decoding performance, it is necessary to obtain an image containing only the color barcode area without the surrounding area. For this, the coordinates extracted from the color barcode area are stored in the continuous image frame; the minimum values of the x- and y-axis coordinates and the maximum values for the height and width are found from the saved coordinates. Through this process, an area with the smallest possible size can be obtained without significant loss of color barcode information, but it still contains a little ambient noise. In the captured image, the process of removing noise uses the fact that, to transmit data, the color barcode continuously changes, but not the area around it.

Figure 5c shows the color barcode mask image obtained by differential and morphological operations. To create a mask image, the color barcode image is binarized. Here, values above a certain grey-scale threshold of 20 are displayed in white with a value of 255, and the surrounding area is set to 0 (i.e., displayed as black). By obtaining differential images of three successive images and adding them to the previous differential images, a background-removed color barcode mask can be generated. Since noise outside the color

bar code area remains, even in the result from using differential images, multiple morphological operations of erosion and dilation are used to remove it. As a result, extraction of the color barcode area without noise is possible. As shown in Figure 5d, by multiplying the finally obtained color barcode mask image with the image extracted through YOLO, the area of a pure color barcode with noise removed can be extracted.

5. Annotation of Color Barcode Patterns

In CCB-OCC, color barcodes appear in various patterns in the image received from the same transmitted signal due to synchronization mismatch and alignment problems between the transmitting and receiving devices. A previous study [20] used a histogram analysis approach for a limited number of patterns that could appear in the captured images to identify the colors in the barcodes. The RGB values of color barcodes could be extracted by finding the maximum and minimum values of the histogram for each channel, and this was used to determine pilot symbols and data symbols. This method has limitations in that it is difficult to accurately detect RGB values for various histogram patterns, and a heuristic approach was required to extract RGB values in the worst cases. In particular, the histogram distribution of the RGB color values constituting the color barcode change significantly when the capturing environment changes, resulting in inaccurate color value extraction. Unlike the CCB-OCC scheme relying on heuristic histogram analysis [20], we propose a method that can improve the reliability of symbol detection performance by using a CNN model (a well-known technique showing robust performance in object classification fields).

To train the CNN model, annotation of color barcodes is required. The color barcode classes in the proposed DeepCCB-OCC consist of three pilot symbols (R, G, B) and eight data symbols (0–7) in the color constellation, a total of 11 classes. In the color barcode extracted at the receiving end, the primary and complementary colors encoded with the data do not appear exactly as the color barcode is halved, but appear in various shift patterns due to the asynchronous problem between the sending and receiving devices, so additional labeling is required. Labeling of the color barcode pattern for each symbol class should be performed before training the CNN model.

In this paper, as shown in Figure 6, four shift patterns of received color barcodes are defined. The first pattern is an ideal type of color barcode in which the pattern of primary and complementary colors is halved. Here, the shift pattern of the received color barcode is demonstrated with pilot symbol G. The first pattern, which is an ideal shape, is represented by GG′ in Figure 6a. In the second pattern shown in Figure 6b, the primary color is located in the center of the color barcode, while the complementary color of the previous color barcode is displayed on the left side, and the complementary color of the primary color is on the right side. As a result, it appears as R′GG′. In the third pattern, complementary colors of the previous color barcode are captured on the left, and the color barcode of the primary color is displayed on the right: R′G is shown in Figure 6c. The fourth pattern shows the color barcode in the form GG′B, as seen in Figure 6d, so the color barcode for a specific color signal has a total of four patterns. Each pattern consists of three pilot symbols and eight data symbols, so data can be received from a total of 44 classes.

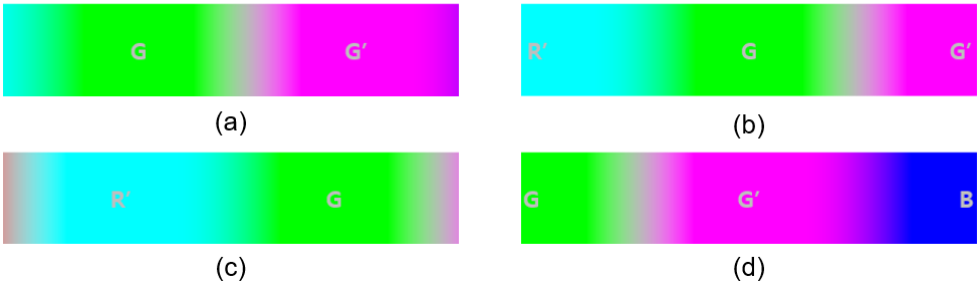


Figure 6. Four different patterns in color barcodes: (**a**) GG′, (**b**) R′GG′, (**c**) R′G, (**d**) GG′B.

As mentioned above, continuous color patterns can be divided into four types, but the color barcode pattern may change depending on the synchronization timing between the transmitting end and the receiving end. In such a situation, the approach to classify 44 classes using a single deep learning model limits color pattern classification accuracy, and ultimately may adversely affect communication performance from the DeepCCB-OCC system. Therefore, to maximize classification performance while maintaining the generalization performance of the DeepCCB-OCC system for various color patterns, CNN models for pilot symbol classification and data symbol classification were independently developed.

6. CNN-Based Synchronization and Color Pattern Classification

For synchronization in the DeepCCB-OCC system, a pilot symbol classification model is used, and only pilot symbols must be accurately identified from the input of several color barcode patterns. This model should be able to distinguish a pilot symbol from a data symbol and should also be able to recognize what kind of pilot symbol it is. For this, the patterns of R, G, and B pilot symbols are designated as individual classes, and various patterns appearing in data symbols are grouped into a single class and are trained. Therefore, the pilot symbol classification model is trained to classify a total of four classes.

In DeepCCB-OCC, color barcodes appear in various patterns in the image received from the same transmitted signal due to synchronization mismatch and alignment problems between the transmitting and receiving devices. In a real-world CCB-OCC system, the received color barcode appears in four patterns for a specific symbol, as shown in Figure 7. As explained in the previous section, 11 classes of data (R, G, B, and 0–7) are encoded in color barcodes, including pilot symbols and data symbols. In addition, since images taken at five different angles were used, the total of the types of color barcodes collected is the number of image capture angles (5) times the number of color barcode patterns (4) times the number of types of encoded data (11); thus, there are 220 types. To build a data set for CNN model training, we collected 20 images for each color barcode type—a total of 4400.

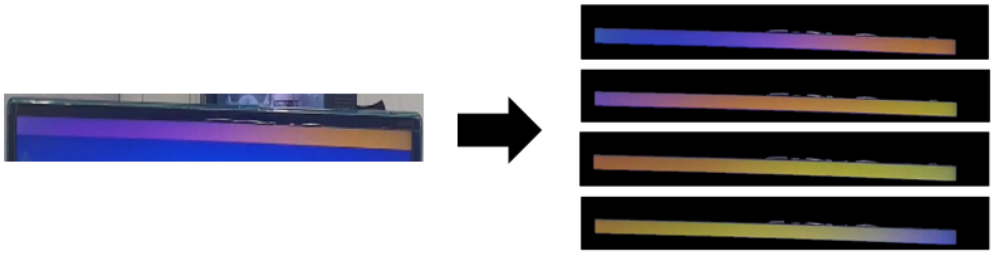


Figure 7. Four different patterns for received color barcode symbol 001 (R, G, B = 255, 191.5, 0).

When the DeepCCB-OCC technique is actually used, the size and shape of the color barcode may appear in various forms depending on the various distances and angles between the display and the camera. To improve model generalization, data augmentation, which has become a standard component of deep neural network training, is used. Figure 8 shows the synthetic data used to train the DeepCCB-OCC model. As can be seen, synthetic data were generated using data augmentation techniques such as enlargement/reduction, rotation, position transformation, and vertical inversion. To support the diversity of the synthetic data, 80 synthetic color barcodes for each color barcode type were created, resulting in a dataset of 22,000. Of those 22,000 color barcode images, 6000 with R, G, and B pilot signals encoded were used to train the CNN model for pilot symbol classification. The remaining 16,000 color barcodes with 0–7 data symbols encoded were used to train the CNN model for data symbol classification.

The structure of the CNN model is shown in Figure 9. Since the size of the color barcode images used in the pilot symbol classification model and the data symbol classification model were the same, regardless of the encoded information, the two structures were configured identically. As input for the model, an RGB three-channel image at 100 × 900 was used, composed of three 2D convolution layers, two maxpooling layers, a fully con-

nected layer, and a dropout layer. Since the training data for each classification model were configured differently, each model was trained independently.

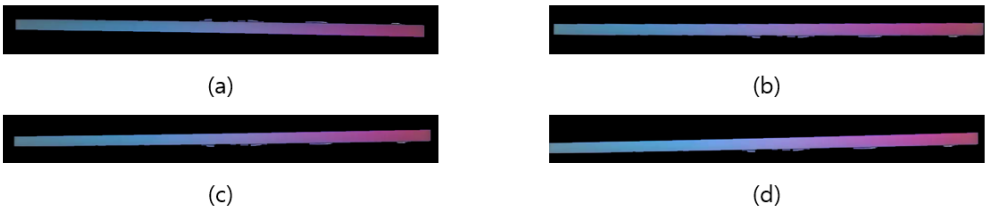


Figure 8. Synthetic data using data augmentation techniques such as enlargement/reduction, rotation, position transformation, and vertical inversion. (**a**) Original data, (**b**–**d**) Synthetic data.

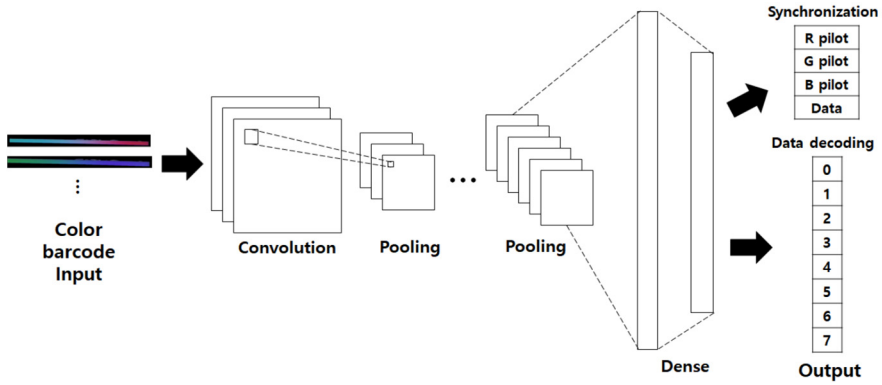


Figure 9. The CNN model's structure for packet synchronization and data decoding.

As shown in the figure, the pilot symbol classification model has four classes, and the data symbol classification model has eight classes. When consecutive color barcode patterns of RR'GG'BB' are obtained using the CNN model for pilot symbol detection, the CNN model for data detection is used as much as the data symbol length of the packet, and the transmitted signals are sequentially decoded. The entire process for decoding data using DeepCCB-OCC is described in Algorithm 1.

Algorithm 1: The entire process of DeepCCB−OCC.

Input: Captured Video with color barcodes
Synchronization ← *False*
while Captured Video **do**
  if Color barcode is detected **then**
    Area ← Color barcode area detected by YOLO
    Mask ← Differential current *Area* with previous *Area*
    Mask ← Morphological operation for *Mask*
    Barcode ← Extract Barcode using *Mask*
    if *Synchronization* **then**
      Data ← Barcode classification using CNN model
      Result.append(*Data*)
    else
      if Pilot symbols R, G, B are detected consecutively **then**
        Synchronization ← *True*
      end if
    end if
  end if
end while
Return *Result*

7. Experimental Results

To evaluate the performance of the DeepCCB-OCC system proposed in this study, communication experiments with a commercial display and a smartphone were performed in an indoor space. The experimental environment simulating an actual environment is shown in Figure 10. When capturing a display using a camera, the formation of the color barcode image is affected by various yaw angles of the camera. In fact, the shape of the collected color barcode is deformed under the influence of distance changes and rotation changes according to the position of the handheld camera and the user's position, in addition to the yaw angle between the display and the camera. In this study, the deformation of color barcode images caused by influences other than the yaw angle is solved by using a synthesized color barcode image generated using the data augmentation technique. The display device serving as the transmitter had a resolution of 1920 × 1080 and a refresh rate of 60 Hz. Among the receiving devices for DeepCCB-OCC, the most commonly used is a smartphone, and in this experiment, a Samsung Galaxy S9 was used to record video at 30 fps. In the experiment, the same space and illumination were considered as in the experiment performed in the previous paper. In the previous paper, the video encoded with 3-bit data was transmitted through the display and captured by the Samsung Galaxy S9 camera, which is the same camera used in the experiment of this manuscript. In addition, the distance and angle range were set to be the same.

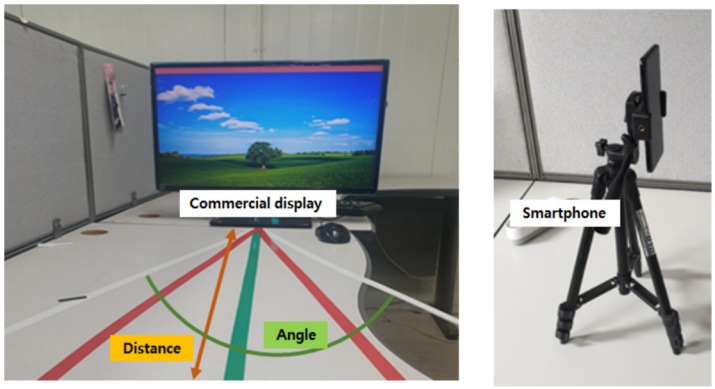


Figure 10. The experimental environment with a commercial display and a mobile phone.

To collect color barcode data based on the yaw angle during capture, the angle between the camera and the display remained steady for five minutes and changed between −20 and +20 degrees at intervals of 10 degrees. For data to train the YOLO-based color barcode extraction model, 600 frames were extracted from the images taken at each angle, and thus, a total of 3000 color barcodes were collected. Here, 2500 images were used for training, and 500 images were used for validation. Note that we have a total of 6000 images acquired for pilot symbols and 16,000 images for data symbols. To implement the CNN model for pilot symbol detection, 4800 images were used as training data, and 1200 images of data were used as test data. In the CNN model for decoding data, 12,800 images were used as training data and 3200 images were used as test data. The CNN model structure and the experimental setup are described in Tables 1 and 2.

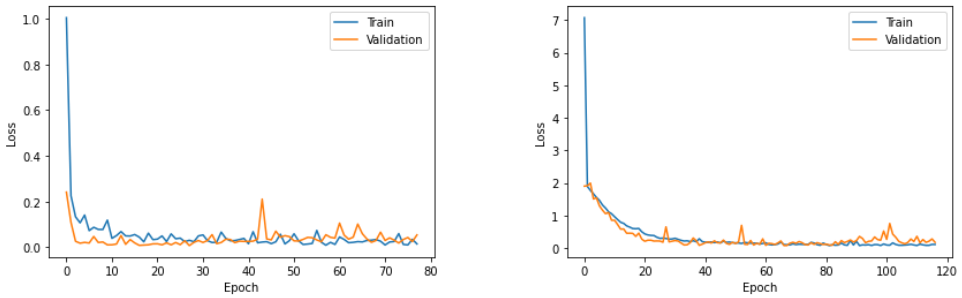
Table 1. The CNN model's structure.

Layer(kernel)	Output Shape	Activation	No. of Params
Input	(None, 100, 900, 3)		0
Conv2D (32, (3, 3))	(None, 98, 898, 32)	ReLU	896
Maxpooling2D (2, 2)	(None, 49, 449, 32)		0
Conv2D (64, (3, 3))	(None, 47, 447, 64)	ReLU	18,496
Maxpooling2D (2, 2)	(None, 23, 223, 64)		0
Conv2D (64, (3, 3))	(None, 21, 221, 64)	ReLU	36,928
Flatten ()	(None, 297,024)		0
Dense (64)	(None, 64)	ReLU	19,009,600
Dropout	(None, 64)		0
Output	(None, 8)		520

Table 2. The experimental setup.

Hyper-Parameter	Value
Optimizer	Adam
Learning rate	0.001
Beta_1	0.9
Beta_2	0.999
Epsilon	0.0000007
The probability of dropout	0.4
Minibatch size	32
Early stopping patience	50
Maximum epochs	1000

Learning curves of the CNN models for pilot and data symbol detection are shown in Figure 11. The purpose behind checking the learning curves is to find the final model using the weights from the epoch with the best loss score. The initial epoch value for training the two CNN models is set to 1000. In the training process of the model, if the validation loss was within the tolerance range for 50 consecutive epochs, the early stopping function was used, as it was judged to have converged. The reason why the early stopping epochs of the two CNN models are different is that the number of data used and the number of classes to be classified are different. In the case of the packet synchronization model with a relatively small number of data and the number of classes, the early stopping epoch was 78, and in the case of the data decoding model, it was 117. For our CNN models, we used weight values on these epochs and evaluated the communication performance.



(**a**) (**b**)

Figure 11. Learning curves of CNN models for (**a**) synchronization and (**b**) data decoding.

To investigate the communication performance of DeepCCB-OCC, the achievable data rate (ADR) was measured, and the ADR of the conventional CCB-OCC [20] was compared with it. As can be seen from Figure 12, the ADR performance of the conventional scheme [20] decreased significantly as the distance between the display and the camera

increased. However, the ADR of the DeepCCB-OCC system showed that the performance was maintained to some extent as the distance increased. The proposed DeepCCB-OCC system showed an ADR of 80.4 bps when the distance between the display and the camera was 90 cm, but maintained an ADR of 78 bps when the distance was 110 cm. This is because the YOLO model shows strong detection performance for the color barcode area regardless of the size of the object, so it showed better performance than the existing histogram analysis-based performance. As the distance increased, the size of the color barcode pattern became smaller. The CNN-based pilot and data symbol detection by DeepCCB-OCC can perform accurate classification without being greatly affected by the size of the pattern, thereby improving the data rate performance.

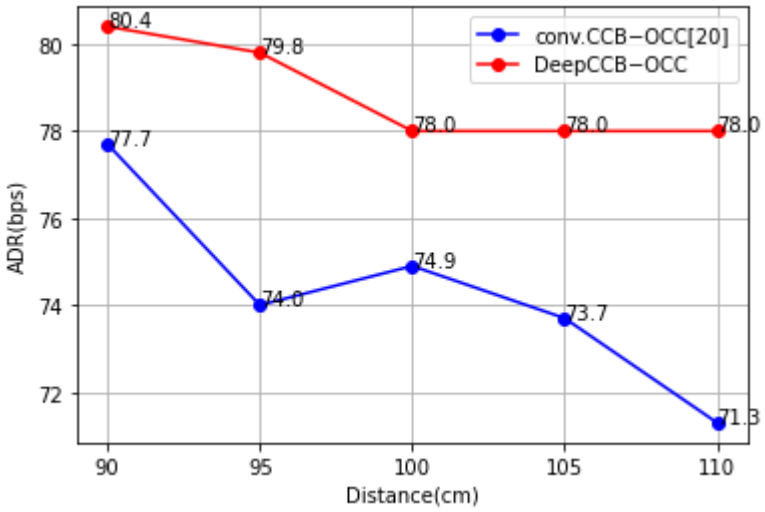


Figure 12. Achievable data rate performance according to distance between the display and the camera. Data adapted from [20]. 2020 Sung-Yoon Jung et al.

In Figure 13, the ADR performance of the proposed method and the existing method is compared by fixing the display-to-camera distance at 90 cm and changing the yaw angle from −20 to +20 degrees. We confirmed that the yaw angle between the display and the camera did not significantly affect the ADR performance of the DeepCCB-OCC system, whereas the ADR performance of the conventional system [20] decreased significantly as the angle increased. The proposed DeepCCB-OCC showed an ADR of 80.4 bps when the yaw angle between the display and the camera is 0 degrees, but maintains an ADR of 79.2 bps when the angle was 20 degrees, outperforming the existing technique. Since the existing system uses a heuristic analysis method for the histogram, if the color barcode area becomes distorted, rather than rectangular, detection of the color barcode area is not accurate, and it is difficult to accurately obtain the maximum value for each RGB channel of the histogram. However, the YOLO model used in this study can accurately detect the color barcode area, because it uses color barcode images obtained at various angles and with synthetic data from data augmentation. In addition, even if the yaw angle is different, since the color barcode pattern is configured in various forms and then decoded with the trained CNN model, a good data rate performance can be guaranteed based on the high generalization performance.

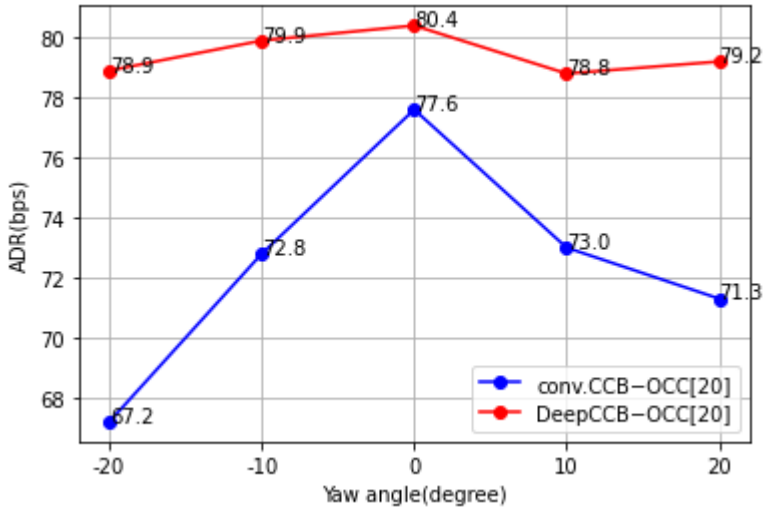


Figure 13. Achievable data rate performance according to yaw angle between the display and the camera. Data adapted from [20]. 2020 Sung-Yoon Jung et al.

Note that the proposed scheme requires no additional data overhead because the structure of the data packet sent from the display is configured in the same way as in the existing method. Since the existing system uses a heuristic analysis method for the histogram, if the color barcode area becomes distorted, rather than rectangular, detection of the color barcode area is not accurate, and it is difficult to accurately obtain the maximum value for each RGB channel of the histogram. When this technique is applied, even in the worst case, it is necessary to determine heuristic conditions for many cases on color barcode histograms, and thus, the complexity becomes very high. However, our deep learning models are well trained to classify unseen data, supporting a robust generalization performance. Our proposed YOLO model supports real-time object detection functionality, and CNN models for synchronization and data decoding consume a processing time of less than 50 ms. In particular, we constructed a shallow CNN framework with fewer layers and parameters, resulting in a less complex structure and less processing time.

8. Conclusions

In this paper, we proposed the DeepCCB-OCC system, which can provide imperceptible data transmission and reliable communication in any environment with an electronic display and a camera by using deep learning models. Instead of the heuristic algorithm used in the existing CCB-OCC system [20], a YOLO detection model and CNN classification models were used to automatically extract the barcode area, synchronize packets, and detect symbols. To extract the pure color barcode area from the captured image obtained from the display, the YOLO detection model was used along with additional image-processing techniques, including differential image and morphological operations. Various received color barcode patterns observed in the received image due to synchronization jitter between the camera and the display were defined. Then, color barcodes were classified with a CNN-based model for packet synchronization and data decoding. Conducting experiments with a commercial display and a smartphone in actual use showed a high ADR performance, compared to the conventional model, at various distances and angles between the display and camera.

Author Contributions: Conceptualization, M.T.K. and B.W.K.; methodology, M.T.K. and B.W.K.; formal analysis, M.T.K. and B.W.K.; investigation, M.T.K.; writing—original draft preparation, M.T.K. and B.W.K.; writing—review and editing, M.T.K. and B.W.K.; supervision, B.W.K.; project administration, B.W.K.; funding acquisition, B.W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (2022R1A2B5B01001543).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nguyen, T.; Islam, A.; Hossan, T.; Jang, Y.M. Current status and performance analysis of optical camera communication technologies for 5G networks. *IEEE Access* **2017**, *5*, 4574–4594. [CrossRef]
2. Al-Kinani, A.; Wang, C.; Zhou, L.; Zhang, W. Optical Wireless Communication Channel Measurements and Models. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1939–1962. [CrossRef]
3. Xie, C.; Guan, W.; Wu, Y.; Fang, L.; Cai, Y. The LED-ID detection and recognition method based on visible light positioning using proximity method. *IEEE Photonics J.* **2018**, *10*, 7902116. [CrossRef]
4. Pathak, P.H.; Feng, X.; Hu, P.; Mohapatra, P. Visible light communication, networking, sensing: A survey, potential and challenges. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2047–2077. [CrossRef]
5. Guan, W.; Wu, Y.; Xie, C.; Fang, L.; Liu, X.; Chen, Y. Performance analysis and enhancement for visible light communication using CMOS sensors. *Opt. Commun.* **2018**, *310*, 531–551. [CrossRef]
6. Marktscheffel, T.; Gottschlich, W.; Popp, W.; Werli, P.; Fink, S.D.; Bilzhause, A.; de Meer, H. QR code based mutual authentication protocol for internet of things. In Proceedings of the 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Coimbra, Portugal, 21–24 June 2016; pp. 1–6. [CrossRef]
7. Wang, A.; Peng, C.; Zhang, O.; Shen, G.; Zeng, B. InFrame: Multiflexing Full-Frame Visible Communication Channel for Humans and Devices. In Proceedings of the 13th ACM Workshop on Hot Topics in Networks, HotNets, Los Angeles, CA, USA, 27–28 October 2014. [CrossRef]
8. Wang, A.; Li, Z.; Peng, C.; Shen, G.; Fang, G.; Zeng, B. InFrame++: Achieve simultaneous screen-human viewing and hidden screen-camera communication. In Proceedings of the 13th International Conference on Mobile Systems, Applications and Service, Florence, Italy, 18–22 May 2015. [CrossRef]
9. Li, T.; An, C.; Campbell, A.T.; Zhou, X. Hilight: Hiding bits in pixel translucency changes. *AGM SIGMOB. Mob. Comput. Commun. Rev.* **2014**, *18*, 62–70. [CrossRef]
10. Jo, K.; Gupta, M.; Nayar, S.K. DisCo: Display-Camera Communication Using Rolling Shutter Sensors. *ACM Trans. Graph.* **2016**, *35*, 1–13. [CrossRef]
11. Wang, Q.; Zhou, M.; Ren, K.; Lei, T.; Li, J.; Wang, Z. Rain Bar: Robust Application-Driven Visual Communication Using Color Barcodes. In Proceedings of the IEEE 35th International Conference on Distributed Computing Systems, Columbus, OH, USA, 29 June–2 July 2015; pp. 537–546. [CrossRef]
12. Zhou, M.; Wang, Q.; Lei, T.; Wang, Z. Enabling Online Robust Barcode-Based Visible Light Communication with Realtime Feedback. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 8063–8076. [CrossRef]
13. Du, W.; Liando, J.C.; Li, M. Softlight: Adaptive visible light communication over screen-camera links. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1620–1628. [CrossRef]
14. Du, W.; Liando, J.C.; Li, M. Soft Hint Enabled Adaptive Visible Light Communication over Screen-Camera Links. *IEEE Trans. Mob. Comput.* **2017**, *16*, 527–537. [CrossRef]
15. Chen, C.; Zhou, B.; Mow, W.H. RA code: A robust and aesthetic code for resolution-constrained applications. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 3300–3312. [CrossRef]
16. Lee, Y.-L.; Tsai, W.-H. A new data transfer method via signal-rich-art code images captured by mobile devices. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 688–700. [CrossRef]
17. Abe, S.; Hiraki, T.; Fukushima, S.; Naemura, T. Screen-camera communication via matrix barcode utilizing imperceptible color vibration. In Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology Adjunct, Berlin, Germany, 14 October 2018; pp. 166–168. [CrossRef]
18. Cui, H.; Bian, H.; Zhang, W.; Yu, N. UnseenCode: Invisible on-screen barcode with image-based extraction. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 28 April–2 May 2019; pp. 1315–1323. [CrossRef]
19. Zhang, K.; Wu, C.; Yang, C.; Zhao, Y.; Huang, K.; Peng, C.; Liu, Y.; Yang, Z. ChromaCode: A fully imperceptible screen-camera communication system. *IEEE Trans. Mob. Comput.* **2021**, *20*, 861–876. [CrossRef]
20. Jung, S.-Y.; Lee, J.-H.; Nam, W.; Kim, B.W. Complementary Color Barcode-Based Optical Camera Communications. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 3898427. [CrossRef]
21. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
22. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]

23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
24. Bharti, V.; Biswas, B.; Shukla, K.K. Recent trends in nature inspired computation with applications to deep learning. In Proceedings of the 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 29–31 January 2020; pp. 294–299. [CrossRef]
25. Li, J.; Zhang, M.; Wang, D.; Wu, S.; Zhan, Y. Joint atmospheric turbulence detection and adaptive demodulation technique using the CNN for the OAM-FSO communication. *Opt. Express* **2018**, *26*, 10494–10508. [CrossRef] [PubMed]
26. Lu, X.; Lu, C.; Yu, W.; Qiao, L.; Liang, S.; Lau, A.P.T.; Chi, N. Memory-controlled deep LSTM neural network post-equalizer used in high-speed PAM VLC system. *Opt. Express* **2019**, *27*, 7822–7833. [CrossRef] [PubMed]
27. Le-Tran, M.; Kim, S. Deep Learning-Based Collaborative Constellation Design for Visible Light Communication. *IEEE Commun. Lett.* **2020**, *24*, 2522–2526. [CrossRef]
28. Lee, H.; Lee, I.; Quek, T.Q.S.; Lee, S.H. Binary signaling design for visible light communication: A deep learning framework. *Opt. Express* **2018**, *26*, 18131–18142. [CrossRef] [PubMed]
29. Lee, H.; Lee, S.H.; Quek, T.Q.S.; Lee, I. Deep Learning Framework for Wireless Systems: Applications to Optical Wireless Communications. *IEEE Commun. Mag.* **2019**, *57*, 35–41. [CrossRef]