Experimental Evaluation of Display Field Communication Based on Machine Learning and Modem Design

Yu-Jeong Kim, Pankaj Singh and Sung-Yoon Jung *

Department of Electronic Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea
* Correspondence: syjung@ynu.ac.kr

Abstract: Display field communication (DFC) is a frequency-domain unobtrusive display-to-camera (D2C) communication, in which an electronic display serves as a transmitter and a camera serves as a receiver. In this paper, we propose a machine learning-based DFC scheme and evaluate its performance in a lab test scenario. First of all, we adopt the Discrete Cosine Transform (DCT) to transform a spatial-domain image into its spectral-domain equivalent. To reduce the computational complexity during the data-embedding process, addition allocation and subtraction data retrieval techniques are used. Moreover, channel coding is applied to overcome the data error caused by the optical wireless channel. In particular, robust turbo coding is used for error detection and correction. Afterward, we perform the experiments to validate the performance of the proposed system. After capturing the displayed image with a camera, data restoration is done using a deep learning technique. Extensive real-world experiments were performed considering various geometric distortions, noise, and different standard input images. As a result, we found that by increasing the transmit display image size (upsampling), the overall error rate can be reduced. In addition, real-world noise analysis is performed and it is notified that the actual noise is dominant in the low-frequency region of an image. The experimental results confirm the robust performance of the proposed DFC scheme and show that an error-free performance can be achieved up to a distance of 1 m in the given lab test environment setting.

Keywords: discrete cosine transform; display-camera communication; display field communication; spectral transformation; machine learning; object detection; turbo coding

1. Introduction

With the recent advancements in display technology, widespread availability of cameras in mobiles, laptops, augmented/virtual reality (AR/VR) devices, etc., and increasing consumption of digital content, display-to-camera (D2C) communications [1–3] will soon become an important part of future wireless communication systems. D2C communication is an optical camera communication technology, where an electronic display and a digital camera communicate via a wireless D2C link. In particular, the data are embedded into the individual video frames at the display (i.e., the transmitter). On the receiver end, a camera captures the frames and decodes the embedded data. In this way, the display serves both as the data transmitter as well as a screen showing contents to normal audiences. D2C communication can be related to currently used quick response (QR) codes (or 2D barcodes) that transmit a very small amount of data to a mobile camera device [4–6]. In addition, even though QR codes can be made imperceptible to the human eye [7–9], they are limited by their size, position, and the number of bits transmitted. On the other hand, D2C communication considers embedding data into a full video stream while remaining unobtrusive to the human eye. That is, it will not obstruct the normal viewing experience of the public while the interested users can get the related side-information of the ongoing content on their mobiles. The most prominent application of D2C communication could be in the field of digital signage, e.g., in hospitals, education, and the hospitality sector,
among others. Alternative applications could be in the fields of human–computer interface, AR/VR, blockchain-based non-fungible tokens, and cybersecurity and privacy [2].

Display Field Communication (DFC) is a kind of D2C communication, where the data are embedded in the frequency domain of an image, rather than the spatial domain [10]. This is done so that the properties associated with the frequency coefficients of an image can be employed. In addition, this will reduce the visual artifacts of an image frame to the naked eye. The pioneering work of [10] proposed and analyzed the DFC technology, where the data are embedded in one dimension of an image frame. The work was then extended to the 2D-DFC [11], where the data are embedded in two dimensions of an image frame ultimately increasing the achievable data rate. The work in [12] showed that the achievable data rates of 1D-DFC could further be improved using advanced digital receivers. However, all the works remain limited to the analytical and simulations stage.

In this paper, we make use of machine learning (ML) [13–17] skills in the context of DFC. Moreover, we perform several practical lab tests to validate the DFC and how ML can be employed to improve its performance. First of all, we adopted Discrete Cosine Transform (DCT) as the frequency conversion method for the input images. The DCT is one of the most widely used frequency conversion methods and is already used in compressions such as JPEG and MPEG. After converting the image to the frequency domain, we do data insertion (or data embedding) through an addition allocator. Among various data-embedding mechanisms [18], we have chosen addition to reduce the mathematical complexity of the proposed scheme. After the data embedding, the spectral-domain image is converted back to the spatial domain and displayed on an electronic screen. At the receiver end, a camera captures the image and decodes the transmitted data.

The main contributions of the manuscript are summarized as follows:

• First of all, we have proved that DFC is possible in a real-life scenario. Unlike previous studies, where we have mostly developed simulations and algorithms to study DFC, we have performed a practical indoor experiment to check the feasibility of the DFC. The results show that communication using DFC is possible in a practical lab test environment and it is shown that error-free performance is achieved until a distance of approximately 1 m in red and green channels.

• An ML algorithm has been used for object detection at the camera receiver. The ML approach is very useful in extracting the display image from the captured image that includes the background noise. Therefore, display detection and data restoration are done by using the deep learning technique. The use of ML techniques significantly boosts DFC performance.

• We evaluated the effect of the optical wireless channel on communication via a D2C link. During the experiment, it has been found that the actual noise in the DFC environment is of non-white nature that exists at low frequencies. This is in contrast to conventional noise, which is always considered to be white and exists at high frequencies.

• To improve the performance of the proposed scheme, we propose a power allocation scheme for the data embedding.

• For image conversion, we use DCT in the current work. This will reduce the computational complexity of the overall system at the transmitter.

• The frequency-domain data embedding is performed through an addition allocator. At the receiver, the data retrieval is done using the subtraction operator.

• To further improve the performance of the scheme, channel coding in the form of turbo coding is included.

• The performance of the practical DFC was evaluated in terms of various geometrical distortions, channel noise, and different input images.

The remainder of this paper is organized as follows: Section 2 presents the state-of-the-art works in D2C communication. In Section 3, we provide a block diagram of the proposed DFC scheme, which consists of modulation-demodulation architecture as well as it’s detailed explanation with mathematical illustrations. The emphasis will be on data
embedding using power allocation and data decoding. Section 4 proposes a display region detection and distortion correction process using deep learning in a real-world experimental setup. In addition, we present in the results of channel coding. Section 5 presents the evaluation of the proposed DFC system based on indoor laboratory experiments considering various distortions including input image size, distance, camera angle, display rotation, and different standard input images. In addition, we have studied the noise characteristics in an actual D2C link and expressed it as a 3D function. Finally, the paper is concluded in Section 6.

2. Related Works

The D2C communication approaches primarily focus on embedding the data in the spatial domain of an image. For instance, [1] proposed a novel screen-to-camera image coding scheme dubbed as TERA. In particular, a new color decomposition-based encoding scheme was designed that encodes the information into a single host frame by creating two complementary frames. The work in [2] introduced an optical covert channel in which attackers can leak sensitive information from air-gapped computers through manipulations on the screen brightness. Malware on a compromised computer can obtain sensitive data and modulate them within the screen brightness, invisible to users. The small changes in brightness are invisible to humans but can be recovered from a video stream taken by a camera such as a local security camera, smartphone camera, or webcam. More recently, the work in [3] proposed embedding 2D barcodes in the blue channel of an image frame to decrease the obtrusiveness. Reference [19] encoded the data in varying pixel translucency of the image frames. Reference [20] presented a demo of imperceptible video communication that takes advantage of both off-the-shelf cameras and LCDs, and transfers data over a normal video without reliance on additional infrastructure. In another work [21], InFrame is proposed that enables dual-mode full-frame communication for both humans and devices simultaneously. It leverages the temporal flick-fusion property of the human visual system and the fast frame rate of modern displays. InFrame multiplexes data onto full-frame video contents through a novel complementary frame design and thus ensures screen-to-camera data communication without affecting the primary video-viewing experience for human users. The effect of vignetting on pixelated multiple-input multiple-output (MIMO) optical wireless communication system was analyzed in [22], which shows that vignetting causes attenuation and intercarrier interference in the spatial frequency domain. Moreover, it proposed error correction schemes to verify and recover the lost blocks and frames during D2C communications.

Similarly, the authors of [23] proposed a novel approach to embed hyperlinks into common images, making the hyperlinks invisible to human eyes but detectable for mobile devices equipped with a camera. The work in [24] evaluated the performance of the LCD-to-camera link, where the LCD modulates ambient light by changing its level of transparency. The authors claimed their prototype to be the first screen-to-camera system that works solely with ambient light. DeepLight [25] introduced a novel, holistic approach for robust screen-to-camera communication that incorporates ML models in the decoding pipeline to achieve humanly imperceptible, moderately high communication rates under diverse real-world conditions. On the other hand, the work in [26] presented a hidden screen-to-camera communication system built upon invisible visual and inaudible audio dual channels. It takes the complementary advantages of video and audio channels by exploiting the reliable yet low-rate inaudible audio link as the control channel while the unreliable but high-rate visual link as the data channel. Reference [27] proposed MobiScan, a dynamic and invisible screen-to-camera communication system that can ensure data security, real-time communication, and flexible capture angle. Similarly, the authors of [28] presented DiSCo, a novel display-camera communication system, which enables displays and cameras to communicate with each other while also displaying and capturing images for human consumption. Messages are transmitted by temporally modulating the display brightness at high frequencies so that they are imperceptible to humans. Messages are
received by a rolling shutter camera that converts the temporally modulated incident light into a spatial flicker pattern. In the captured image, the flicker pattern is superimposed on the pattern shown on the display. The flicker and the display pattern are separated by capturing two images with different exposures.

Some works also deal with specific challenges of D2C communications. For instance, the authors of [29] tackled a key challenge of imperfect frame synchronization over D2C links. LightSync achieved frame synchronization, which features in-frame color tracking to decode imperfect frames and a linear erasure code across frames to recover lost frames. Reference [30] created data patterns by considering the color space for lightness modifications. The work of [31] that introduced D2C in the form of visual MIMO encoded the information by modulating the video frame with bright or dark intensity in a differential manner. The information can be decoded by retrieving the residuals obtained by subtracting the alternate frames.

3. The Proposed DCT-Based DFC System

Figure 1 depicts the block diagram of the proposed DFC system. Fundamentally, in a DFC system, the camera is frontally aligned with the transmitter screen. First, the input image is converted to the frequency domain using the DCT. At the same time, the binary input data, \( b \in \{0, 1\} \), are channel-encoded and modulated to symbol \( d \) (or \( s \)) and embedded into the image. The modulated input data are embedded into the frequency domain of the image via addition allocation. Subsequently, the data-embedded image is converted back to the spatial domain to be displayed on the screen. This operation is performed by inverse DCT (IDCT). At the same time, to correctly decode the data at the receiver end, a reference frame per data frame is sent. In this fashion, data-embedded frames and reference frames are multiplexed and repeatedly displayed on the screen.

At the receiver end, the camera captures the image frame. However, due to the optical wireless channel and background noise, distorted images are received, which need to be recovered. To remove the distortion from the captured images, deep learning object detection from computer vision is used. Consequently, restored images are converted to the frequency domain. Then, utilizing the reference images and subsequent operations of subtraction, demodulation, and channel decoding, output data bits are estimated at the camera decoder.

In this paper, we use RGB color images for data embedding. The RGB image is separated into three different channels, i.e., red (R), green (G), and blue (B). For each RGB channel, the above procedure was applied in the same way. In the next section, we describe the mathematical model of our proposed scheme. Then, in subsequent sections, we describe the experimental setup of the proposed scheme with the experimental results.
3.1. Data Embedding

As mentioned above, to embed data in the frequency domain, first, we convert the spatial-domain image to its frequency domain equivalent. For this, we use DCT due to its several advantages. First of all, aforementioned, DCT is already used in compressions such as JPEG and MPEG, and hence, it is compatible with our DFC system that uses JPEG images or MPEG frames. Second, DCT uses only the cosine (real) signal part of the complex exponential signal, which has the advantage of reducing the operational load on the proposed scheme. The frequency-domain image \( I_F \) after the DCT conversion can be obtained as

\[
I_F = \begin{bmatrix}
i_{F1} & i_{F2} & \ldots & i_{FQ}
\end{bmatrix} = C \cdot I_t,
\]

where \( I_t \) is a \( P \times Q \) spatial-domain image, \( i_{Fq} \) and \( i_{tq} \) \((q = 1, 2, \ldots, Q)\) are the column vectors of the frequency-domain and spatial-domain image, respectively, and \( C \) is a \( P \times P \) DCT matrix whose \( m, n \)th element is given as

\[
C_{mn} = \begin{cases}
\frac{1}{\sqrt{P}} & m = 0, \quad 0 \leq n \leq P - 1 \\
\frac{2}{P} \cos \frac{(2n+1)m}{2P} & 1 \leq m \leq P - 1, \quad 0 \leq n \leq P - 1
\end{cases}
\]

Note that the result of 1D-DCT has low-frequency components on the upper side of each column vector \( i_{Fq} \), whereas mid-frequency components lie in the central region, and high-frequency components lie in the lower part of each \( i_{Fq} \) [18].

The data matrix is also generated to be of \( P \times Q \) size given as \( X = [X_1, X_2, \ldots, X_Q] \), where the \( q \)th column vector is given as

\[
X_q = \begin{bmatrix}
0_{1 \times s} & (d_q)^T & 0_{1 \times (P-s-L)}
\end{bmatrix}^T,
\]

where \( d_q = [d_q(1), \ldots, d_q(L)]^T \) is the data symbol vector on the \( q \)th column of the data matrix, \( s \) is the starting pixel of the data symbol, and \( L \) is the number of data symbols embedded per column. The value of \( d \) is given as

\[
d = \begin{cases}
-1 & \forall b : b = 0 \\
1 & \forall b : b = 1
\end{cases}
\]

From Equation (3), we can see that the data structure will cover a \( L \times Q \) rectangular region on the frequency-domain image.

Each pixel in the frequency-domain image represents a specific frequency. Figure 2 shows the location of the data embedded in the frequency-domain image as white rectangular regions called sub-bands. The remaining black part is the area where the data are not embedded. Note that increasing the width of the white band, i.e., \( L \rightarrow P \), will result in visible artifacts when we convert the frequency-domain image back to the spatial domain. This will demolish the original purpose of the screen. As shown in the figure, the data can be embedded in different sub-bands of a frequency-domain image. Embedding the data in the upper part of the image implies embedding the data in the low-frequency region, and embedding the data in the lower part of the image implies embedding the data in the high-frequency region. On this basis, data can be embedded in three different regions of a frequency-domain image called as low sub-band, middle sub-band, and high sub-band, representing data embedding at low-frequency, mid-frequency, and high-frequency regions of the image, respectively.
Generally, low-frequency components in an image indicate that changes in the brightness value of the image pixels are limited. Conversely, high-frequency components indicate that the brightness values of the pixels in the image change frequently. Normally, the low-frequency components occupy larger proportions than the high-frequency components. Hence, if we embed data in the low-frequency region, there is an advantage that data restoration and decoding at the receiver will be good. When the peak signal-to-noise ratio (PSNR) in an image is 30 dB or above, it is generally agreed that the distortion between the original and received image is not significant [32]. However, if data are embedded at the low-frequency components, the PSNR may fall below 30 dB and the image quality will be distorted. For instance, in a 256×256 pixel image, if we embed a total of 500 bits in rows 5 to 15 (lower sub-band), with $\alpha = 0.35$ (see Section 3.1.1), the PSNR of the image becomes 30 dB. Therefore, data must be embedded to ensure both data recovery and image quality.

![Figure 2](image-url)

Figure 2. Three different data embedding regions in a frequency-domain image. The white regions represent the positions of the frequency sub-bands containing data. Low, middle, and high sub-bands indicate that the lower, middle, and higher frequency coefficients are used for data embedding, respectively.

The data embedding is carried out using the addition allocator on the pixel values of a DCT-converted image. As the data embedding is performed in the frequency domain, the data-embedded image, $D_F$, is expressed as

$$D_F = I_F + X. \quad (5)$$

To display the image frame on the electronic screen, the above frequency-domain data-embedded image must be converted back to the spatial domain. This is done by IDCT (cf., Figure 1), and the spatial-domain image to be displayed on the screen is expressed as

$$D_t = C^H \cdot D_F. \quad (6)$$

3.1.1. Power Allocation Scheme for Data Embedding

Until now, we have described how to use DCT and addition allocator to embed data to the frequency-domain image. Although the DCT and addition allocator have the advantages of making the algorithm easier to implement and reducing the computational complexity in the data-embedding process, it has a disadvantage too. When data are embedded with an addition allocator, it is difficult to decode the data at the receiver when the pixel values and the data values are not balanced properly. Therefore, we propose a power allocation scheme for data embedding that scales data values in proportion to the average power of the frequency domain coefficients.

Considering the frequency-domain image $I_F$, we propose to set the value of the data, i.e., $X_{\text{amp}}$, as

$$X_{\text{amp}} = \left(\sqrt{P_{\text{avg}}}\right)^{\alpha}, \quad (7)$$

where $P_{\text{avg}}$ is the average power of a data-embedded region of $I_F$ and $\alpha (0 < \alpha < 1)$ represents the proportionality constant. Note that the PSNR value of the image should be chosen to minimize the bit error rate (BER) while the PSNR should not fall below 30 dB.
Therefore, considering the above power allocation constraint, $d_q$ in Equation (3) is replaced by the modified data vector $s_q$ as follows:

$$s_q = X_{amp}d_q,$$

where $s_q$ is the $q$th column of the data matrix. Finally, the $q$th column vector in Equation (3) can be re-formulated as

$$X_q = \begin{bmatrix} 0 & (s_q)^T & 0 \\ \frac{1}{s} & \frac{1}{s} & \frac{1}{s} \\ 1 & 1 & 1 \end{bmatrix}^T.$$

### 3.2. Data Detection

As shown in Figure 1, the camera is frontally aligned with the display and captures images displayed on the electronic screen. In the process of capturing images, several noises are added to captured images. Although the noise in the data-embedded frames and the reference frames may differ, we can assume that noises in both frames are the same (as they are transmitted consecutively). In this way, the images received by the camera via a D2C link can be expressed as

$$Z_t = I_t^{(ref)} + N_t,$$
$$Y_t = D_t + N_t = C^H D_F + N_t,$$

where $Z_t$ is the received reference image frame, $Y_t$ is the received data-embedded image frame, and $N_t$ is the noise in the image. After receiving the images, the images are converted to the frequency domain using the same DCT conversion process as

$$Z_F = C \cdot I_t^{(ref)} + C \cdot N_t = I_t^{(ref)} + N_F,$$
$$Y_F = C \cdot D_t + C \cdot N_t = D_F + N_F = I_F + X + N_F.$$

Finally, the data are decoded using the subtraction data retrieval process that includes subtracting the reference frame from the data-embedded frame as

$$\hat{X} = Y_F - Z_F.$$

Since $\hat{X} = [\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_Q]$, the finally decoded data $\hat{s}_q$ can be expressed as

$$\hat{s}_q(l) = \hat{X}_q(s + l, q), \quad l = 1, 2, \ldots, L.$$

Consequently, the estimated data symbols can be expressed in the matrix form as

$$\hat{s} = [\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_Q],$$

where $\hat{s}_q = [\hat{s}_q(1), \ldots, \hat{s}_q(L)]^T$ is the estimated data symbol vector on the $q$th column of the received data-embedded image. Then, after channel decoding, the estimated bit can simply be given as

$$\hat{b} = \begin{cases} 0 & \forall \hat{s} : \hat{s} < 0 \\ 1 & \forall \hat{s} : \hat{s} \geq 0 \end{cases}.$$

### 4. Laboratory Experimentation of the Proposed DFC Scheme

In this section, we describe a laboratory experiment of the proposed DFC scheme and evaluate its performance under channel coding. Table 1 shows the default specification of the experimental parameters. For the experiments, we use the Lena image because it is a standard and widely accepted test image used in the field of image processing. In addition, from a compositional standpoint, Lena in particular was an ideal image: feathers in her
hat provided great detail, a human face provided a recognizable subject, and a variety of
gradations and textures proved to be a useful challenge for processing algorithms. In the
experiment, a 256 $\times$ 256 pixel color Lena image was displayed on the Samsung monitor of
resolution 1920 $\times$ 1080 pixels. Then, the display was captured at a distance of 28 cm with
an iPhone camera at 1080p (Full HD) using the 12-megapixel main camera. Moreover, the
experiment was performed indoors under the ambient LED lighting condition including
natural light. While capturing the display, no focal distortion is observed as all the display
pixels were in the focus of the camera pixels. Furthermore, since the frame rate of the
camera is the same as the refresh rate of the display, the problem of image quality distortion
due to the rolling shutter effect [33] does not occur. For data transmission, 500 random
binary data bits were considered and after the channel coding, 2518 data bits were finally
embedded in the input image. In the case where we do not use channel coding, a total
of 2560 binary data bits were embedded into the input image. Additionally, the data
are embedded in the lower sub-band, i.e., from rows 5 to 15 of the input image. Besides, no
camera angle or image rotation is considered. For details, refer to Table 2.

Table 1. Default experimental parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitter display</td>
<td>Samsung monitor, 51 cm $\times$ 29 cm, Res. 1920 $\times$ 1080 p, Refresh rate 60 Hz</td>
</tr>
<tr>
<td>Receiver camera</td>
<td>Apple iPhone XS Max, 12 MP (1920 $\times$ 1080), Frame rate 60 fps, Field of view 59.67$^\circ$</td>
</tr>
<tr>
<td>Test location</td>
<td>Indoor</td>
</tr>
<tr>
<td>Lighting</td>
<td>Ambient light</td>
</tr>
<tr>
<td>Channel coding</td>
<td>Turbo coding</td>
</tr>
<tr>
<td>Modulation</td>
<td>BPSK</td>
</tr>
</tbody>
</table>

Table 2. Experimental parameters for channel coding.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>w/Channel Coding</th>
<th>w/o Channel Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Image</td>
<td>Lena color (256 $\times$ 256 p)</td>
<td>Lena color (256 $\times$ 256 p)</td>
</tr>
<tr>
<td>No. of binary data bits</td>
<td>500</td>
<td>2560</td>
</tr>
<tr>
<td>No. of encoded data bits</td>
<td>2518</td>
<td>-</td>
</tr>
<tr>
<td>Lower sub-band position</td>
<td>rows 5 to 15</td>
<td>rows 5 to 15</td>
</tr>
<tr>
<td>Distance</td>
<td>28 cm</td>
<td>28 cm</td>
</tr>
<tr>
<td>Camera angle</td>
<td>0$^\circ$</td>
<td>0$^\circ$</td>
</tr>
<tr>
<td>Image rotation</td>
<td>0$^\circ$</td>
<td>0$^\circ$</td>
</tr>
</tbody>
</table>

4.1. Channel Coding

To receive and decode the data at the receiver with minimal errors, a channel coding
method has been used. Channel coding acts as an error detector when the data are
mistakenly received at the receiver or lost. In addition, it acts as an error corrector that
restores the original data. In a real-world experiment, noise caused by the optical wireless
channel has a significant impact on error performance. Therefore, the role of channel coding
becomes very important for creating a system that is more robust and reliable against noise.
Among many channel coding techniques, we used turbo coding, which is excellent in error
detection and correction [34].

Figure 3 depicts the error rate comparison of the proposed DFC scheme with and without
channel coding. The plot was generated using three different RGB channels of the color
Lena image. We can get several insights from the plot. First, we can observe the degradation
of BER with increasing PSNR. Although high PSNR means a better image quality, it is
achieved at the cost of degraded communication performance. From Equations (7)–(9), we
can note that the PSNR of the image has an inverse relationship with $\alpha$. With a small value
of $\alpha$, the visual quality of the data-embedded image is well preserved, and the PSNR is high.
However, it decreases the data power. Therefore, it lowers the probability of successful
decoding when the data are extracted from the captured image. Conversely, a larger value of $\alpha$ tends to degrade the image quality and the PSNR becomes low. However, it will increase the data power, which will eventually result in better communication performance. Second, we can observe that as the PSNR increases, the channels with turbo coding perform better than the channels without turbo coding. Note that we have embedded the data in the same region of the images without considering specific features of the red, green, and blue channels. We can observe that using the turbo coding, the BER remains zero until 35 dB for all the channels. In particular, we can observe that the blue channel performs a little bit worse than the green and red channels, both with and without coding. Overall, we can see that channel coding significantly affects the DFC performance. Therefore, we can conclude that channel coding is an essential part of practical DFC as without channel coding the communication is not reliable.

![Figure 3](image_url)

**Figure 3.** Performance comparison of DFC with and without channel coding for 256 × 256 pixel Lena image. The experiment was performed by averaging a total of 10 sets of Lena images in which different random data were embedded.

### 4.2. Display Detection through Deep Learning

In addition to channel coding, deep learning has been used to further improve the BER performance of the scheme. As mentioned before, while capturing a display, the first thing to be taken care of is that the display pixels should be in focus with the camera pixels. However, based on the distance between the display and a camera, the camera may capture the background in addition to the display. These background clutters must be removed from the captured image to detect the display area. Rather than using the image processing techniques [35], we use the deep learning-based object detection scheme to detect our screen. In particular, object detection includes a classification concept that classifies what kind of object it is and a localization concept that finds the position of an object via drawing a bounding box around it. The conventional recurrent-convolutional neural network (R-CNN) [36] is a 2-stage detector that performs region proposal and classification sequentially, which has good performance in object detection accuracy but is too slow. To solve this problem, YOLO (You Only Look Once) [37] has appeared. YOLO, a CNN-based object detection algorithm, is a 1-stage detector that performs region proposal and classification at the same time, with slightly lower accuracy than a 2-stage detector, but higher real-time detection speed. In particular, YOLOv4 improved detection accuracy by more than 10% compared to YOLOv3 using various techniques such as bag-of-freebies, bag-of-specials, and mosaic augmentation. Therefore, we use the YOLOv4 model, which is a faster state-of-the-art ML model for real-time object detection.
The image used in the experiment is an open dataset image provided by Google. This consists of 3000 images—each image containing a display screen. Next, we manually annotate these data using an open-source image labeling software known as LabelImg [38,39]. At this time, the augmentation did not proceed because it was determined that display is an easy classification class and the already labeled images were sufficient. After dataset annotation, the system based on the YOLOv4 network was trained by using images and the weights of the trained network were saved.

Figure 4 illustrates the display area detection process using the YOLOv4 object detection model. Figure 4a shows the image captured by the camera. It can be observed that the captured image has background clutters. Figure 4b shows an image classified as a display and marked with a bounding box using the deep learning algorithm. Finally, Figure 4c is the cropped image using the four edge coordinates of the bounding box, thus almost eliminating the background clutters. Note that in a normal DFC scenario, the images can be captured from a long distance, so the captured image has a plurality of mixed objects together. In such cases, if deep learning is used to detect the display area and then correct the distortion, more efficient data demodulation can be achieved.

4.3. Image Extraction

To extract the image from the distortion-corrected display image, we need to get the four corner points’ coordinates of the Lena image. For that, we use some correction algorithms from the OpenCV library. As shown in Figure 5, when we capture the image using a high-resolution camera, the display pixels of the white background are also clearly captured in addition to the data-embedded Lena image. So, at first, we remove the noise mixed in the image using the Gaussian Blur [40]. Then, the keypoints detection method [41] is used to find the four corner points’ coordinates of the image. Finally, the perspective transform function [42] is used to obtain the distortion-corrected image as shown in Figure 5b.
5. Experimental Evaluation

The experiments were performed in an indoor lab environment. The experimental devices and default parameters are mentioned in Table 1. In addition, as shown in Table 3, the original input image size is $256 \times 256$ pixels. The data are embedded in the image using addition and power allocation. For the power allocation, note that the value of $\alpha$ is chosen to be a variable in the range $0 < \alpha < 1$ based on the PSNR. By adjusting the value of $\alpha$, we can get reliable performance with respect to PSNR. After data embedding, the image is displayed on an electronic display of resolution $1920 \times 1080$ pixels. Then, the image was captured using the iPhone XS Max with a 12 MP rear camera. As the captured image has background clutters, the display area was extracted from the captured image using the techniques mentioned in the previous sections. The extracted image is then resized to its original size and the data are decoded.

In the case of conventional DFC [10], we have considered the Gaussian noise channel. However, in the real practical experiment, note that the noise may not be exactly Gaussian. Moreover, there is a lot of corruption from the real optical wireless channel. Hence, in the following sections, we first analyze the behavior of real practical noise observed in an actual wireless D2C link, and then evaluate the performance of our proposed scheme based on various system design parameters including the transmit image size, distance between display and camera, camera angle, and image rotation. Furthermore, apart from the Lena image, we will also use other standard images and check the performance of our scheme.

5.1. Noise Analysis of the Actual D2C Link

Let us assume that the original transmit image is image A, which is displayed on the screen. Now, the image is captured by a camera and subsequently distortion-corrected to obtain image B. In this case, the noise characteristics due to the optical wireless channel can simply be found by calculating the difference between the two images in the frequency domain. The experimental parameters for noise analysis are described in Table 3.

Table 3. Experimental parameters for noise analysis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Image</td>
<td>Lena color ($256 \times 256$ p)</td>
</tr>
<tr>
<td>No. of binary data bits</td>
<td>500</td>
</tr>
<tr>
<td>No. of encoded data bits</td>
<td>2518</td>
</tr>
<tr>
<td>Lower sub-band position</td>
<td>rows 5 to 15</td>
</tr>
<tr>
<td>Distance</td>
<td>28 cm</td>
</tr>
<tr>
<td>Camera angle</td>
<td>$0^\circ$</td>
</tr>
<tr>
<td>Image rotation</td>
<td>$0^\circ$</td>
</tr>
</tbody>
</table>

The actual noise characteristics for the red channel observed in our experiment are shown in Figure 6. In particular, Figure 6a shows a three-dimensional view of the noise observed in a D2C link. For reference, we have included the input image to get a perspective on how pixels contribute to frequency-domain noise. We can observe that the noise is predominantly present in the upper part of the image. In other words, the most significant components of noise are present in the vicinity of row 0 as observed in Figure 6b. As already mentioned in Section 3.1, the upper part of an image consists of low-frequency components due to the characteristic of 1D-DCT operation. Therefore, we can observe that the actual noise components are present in the low-frequency bands contrary to our belief that the noise is mainly present at high frequencies.
Another interesting fact we can observe is that conventional Gaussian noise is considered to be white, which has a flat power spectral density (p.s.d.). However, the actual noise is not Gaussian and seems to be colored in nature having different power in different frequency bands. This is because the actual noise consists of corruptions of transmitting images due to many factors such as camera tilting, scaling, image rotation, illumination conditions, etc. In addition, if we get some error during the image extraction process at the camera, it will also contribute to noise. So, when we capture the transmitter display with a camera, all kinds of noise get added together to the captured image pixels. If we analyze this noise, we found that most noise sources are present in low-frequency bands and very few noise sources are present in mid- and high-frequency bands.

Accordingly, from the noise analysis, we can find that ‘which sub-band is best from the data-embedding perspective?’ The data embedding area should be determined by appropriately considering the two factors of data decoding performance and image quality maintenance. From the data decoding perspective, it is optimal to embed the data in low-frequency components. In other words, data embedding from rows 0 to 10 are expected to have a good data decoding performance. However, if we look at Figure 6b, in this region, the noise components are very high. Therefore, it does not seem appropriate to embed the data at such high noise components. Therefore, we choose rows 5 to 15 as the proper data-embedding region in our experiment (shown in the red box). Although there are still some significant noise components in the given region, note that we are using the power allocation scheme to increase the power of the data pixels. Therefore, by properly adjusting the value of $\alpha$ in Equation (7), we can confirm the performance with respect to the PSNR.

Next, we proceed with our experiments with the given data embedding region and present experimental results according to various system parameters.

5.2. Performance According to Transmit Image Size

Table 4 presents the parameter settings for all three images used to examine the performance of our DFC scheme. For easier understanding, we also show the areas of an image where the data are embedded. All three images were evaluated in the same experimental environment. Note that as the size of the image doubles, the number of embedded data bits also doubles keeping the area of the lower sub-band fixed from rows 5 to 15.
Table 4. Experimental parameters for different image sizes.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Image 1</th>
<th>Image 2</th>
<th>Image 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Image</td>
<td>Lena color (128 × 128 p)</td>
<td>Lena color (256 × 256 p)</td>
<td>Lena color (512 × 512 p)</td>
</tr>
<tr>
<td>No. of binary data bits</td>
<td>250</td>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>No. of encoded data bits</td>
<td>1268</td>
<td>2518</td>
<td>5018</td>
</tr>
<tr>
<td>Lower sub-band position</td>
<td>rows 5 to 15</td>
<td>rows 5 to 15</td>
<td>rows 5 to 15</td>
</tr>
<tr>
<td>Distance</td>
<td>28 cm</td>
<td>28 cm</td>
<td>28 cm</td>
</tr>
<tr>
<td>Camera angle</td>
<td>0°</td>
<td>0°</td>
<td>0°</td>
</tr>
<tr>
<td>Display rotation</td>
<td>0°</td>
<td>0°</td>
<td>0°</td>
</tr>
<tr>
<td>Pictorial representation</td>
<td><img src="example.png" alt="Diagram" /></td>
<td><img src="example.png" alt="Diagram" /></td>
<td><img src="example.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

The reason for using the same sub-band regions in all images lies in the frequency characteristics of the images. From the Fourier representations, we know that any signal can be decomposed into its multiple sinusoidal harmonics as shown in Figure 7. The original signal is depicted as a red square wave and the periodic functional components that compose it are depicted as blue sinusoidal waves of different frequencies. The main idea behind the Fourier transform is to express the signal as a sum of multiple signals, i.e., periodic functions of sine or cosine, each having a different frequency. By looking at the above concept from the viewpoint of image processing, each pixel of a frequency-domain image is considered to have a frequency component. That is, an image is a representation of multiple frequencies, where each frequency resides in each pixel. As shown in Figure 2, the upper part of the frequency-domain image contains the low-frequency components and the lower part of the image contains the high-frequency components. In our experiment, all the characteristics of the images are the same except for their size. As the image size increases, the pixels of the high-frequency coefficients increase to express the details of the image. Hence, whenever the size of the input image doubles, low-frequency components do not change while the image quality is improved by adding high-frequency components. Therefore, in the experiment, the lower sub-band regions in which the data are embedded remain the same (from rows 5 to 15) regardless of the size of the image.

Figure 7. Original signal (red) and periodic functions (blue) composed of the weighted sums.

Figure 8 depicts the BER performance of the proposed scheme according to PSNR and transmit image size. In the case of 128 × 128 pixel Lena image (cf. Figure 8a), we can observe that the error rate of the red and green channels is zero for all the values of PSNR up to 50 dB. We can observe some BER in the case of the blue channel after 45 dB. Similar to Figure 8a, the error rate of the green channel of the 256 × 256 pixel Lena image was zero for all the values of PSNR. However, the difference can be observed in the case of the red and blue channels, which show a gradual increase in BER after 45 dB and 40 dB,
respectively. Figure 8c depicts the error rate when the 512 × 512 pixel Lena image was used at the transmitter display. The red and green channels show errors after 45 dB PSNR. However, the blue channel is worse affected and started showing errors after 20 dB PSNR. Therefore, we can observe that the error rate is significant in the blue channel. Moreover, the overall error rate increases as the size of the image increases. Consequently, it can be observed that the lower-size image performs better than the large-size image.

![Graphs showing BER performance](image)

**Figure 8.** BER performance according to PSNR and transmit image size. (a) 128 × 128 pixel size. (b) 256 × 256 pixel size. (c) 512 × 512 pixel size.

Although Figure 8 represents the BER of the current communication scheme, the BER at lower PSNR seems equal and cannot be visualized clearly. To spot the error in lower values of BER, we have plotted the error vector magnitude (EVM) in Figure 9. The EVM plots are drawn using the received data bits just before turbo decoding. Moreover, we express the root mean square (RMS) of the EVM in percent by multiplying the ratio by 100. We can observe that the EVM is increasing with increasing PSNR indicating the increasing distance between the measured and ideal points in the received constellation. In other words, it indicates greater BER. Similar to BER plots in Figure 8, we can also observe that the blue channel shows the worse EVM whereas the red and green channels show a similar EVM. In addition, we can observe that as the image size is increasing, the EVM is increasing, showing that the large-size images will produce more errors compared to the small-size images.

The reason why the error rate differs depending on the image size depends on the upsampling and downsampling effects. Note that the original image is resized (zoomed or upsampled) to be displayed on a screen. Then, the image is captured from the iPhone, and the display area is extracted from the captured image. Then, the extracted image is resized (or downsampled) to the original size to decode the data. Now, as all the images are upsampled to the same size, if the size of the original image is small, it means that we are transmitting a higher number of multiple pixels per data bit comparatively (due to a
higher upsampling ratio). In other words, one data bit is assigned to more pixels, when the image is zoomed on the screen. That is, zooming automatically adds redundancy to the transmitted data. At the camera receiver, this makes it easy to demodulate the data bits. Therefore, if we reduce the original image size, better error performance is observed. In other words, as the original image size gets smaller, more pixels are utilized to express one data bit making it easier for the camera receiver to decode the data bits. However, it will reduce the achievable data rate. This is obvious from Table 4 that small images can transmit a lesser number of bits comparatively.

Figure 9. EVM performance according to PSNR and transmit image size. (a) 128 × 128 pixel size. (b) 256 × 256 pixel size. (c) 512 × 512 pixel size.

The authors also want to point out that upsampling effect will fundamentally depend upon the upsampling ratio, i.e., the ratio of display size to the original image size. For example, if we have a bigger display than used in this experiment, it is possible that the 512 × 512 pixel image may show better performance. In summary, reducing the original image size is a way to improve the BER in the DFC link. However, it will reduce the data rate. In that case, one can use a huge display keeping the higher image size to take the benefits of both better BER as well as a higher achievable data rate.

To get an idea about the upsampling effect, let us take a walkthrough example. As shown in Figure 10, if a 128 × 128 pixel image is magnified horizontally and vertically twice to 256 × 256 pixels, one pixel of a 128 × 128 image becomes equivalent to four pixels. Due to this upsampling effect, the 128 × 128 pixel image performs better than the 256 × 256 pixel image. Similarly, if we enlarge a 128 × 128 pixel image to 512 × 512 pixels, one pixel becomes equivalent to 16 pixels. However, note that the image quality looks more distorted on the transmitter display compared to the 256 × 256 pixel image. In other words, the smaller the image size, the better the performance, but the worse the image quality.
To solve this problem, the next section proposes a way to improve image quality through repetitive data embedding.

Figure 10. Increased pixel appearance when the image was enlarged horizontally and vertically \((K = 2)\). Different colors indicate different pixel values.

**Image Quality Improvement through Repetitive Data Embedding**

Table 5 shows the experimental setting for two images of different sizes (128 × 128 and 256 × 256 pixels) used in the experiment. In the experiment, 150 identical data bits are embedded in both images, and they are displayed on the same monitor to be affected by the same upsampling effect. As mentioned above, owing to the upsampling effect, the number of encoded data bits becomes four times higher in the case of image 2. Previously, we had mentioned that the effect of upsampling on a 128 × 128 pixel image is relatively larger than that on a 256 × 256 pixel image. Hence, the performance of a 128 × 128 pixel image is good. However, the image quality is distorted due to reduced PSNR. To solve this problem, we propose a method to obtain both performance and image quality by repeatably embedding four pixels data into the frequency domain of the 256 × 256 pixel image and obtaining the same upsampling effect as for the 128 × 128 pixel image.

**Table 5.** Experimental parameters for image quality improvement.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Image 1</th>
<th>Image 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Image</td>
<td>Lena color</td>
<td>Lena color</td>
</tr>
<tr>
<td>Input image size</td>
<td>128 × 128 p</td>
<td>256 × 256 p</td>
</tr>
<tr>
<td>No. of binary data bits</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>No. of encoded data bits</td>
<td>768</td>
<td>3072</td>
</tr>
<tr>
<td>Lower sub-band position</td>
<td>rows 3 to 9</td>
<td>rows 5 to 17</td>
</tr>
<tr>
<td>Distance</td>
<td>10 cm</td>
<td>10 cm</td>
</tr>
<tr>
<td>Camera angle</td>
<td>0°</td>
<td>0°</td>
</tr>
<tr>
<td>Image rotation</td>
<td>0°</td>
<td>0°</td>
</tr>
</tbody>
</table>

Figure 11 depicts the conventional data-embedding method and the proposed repetitive data-embedding method. Let the data we want to embed be a vector of the form \([0, 1, 0, 0, 1, 1, \ldots]\). In the conventional method (left side), the data embedding descends from top to bottom in a straightforward manner. On the other hand, in the repetitive data-embedding method (right side), each dataset is repeated at four pixels into the image using the repetitive coding technique. In other words, four pixels represent one data bit of information, which means adding redundancy to data.
Figure 11. Comparison of the conventional and proposed repetitive data embedding method.

Figure 12 depicts the BER comparison of two images according to PSNR. We have used repetitive coding in a 256 × 256 pixel image by embedding one data into four pixels. This is to obtain the same upsampling effect as in the case of a 128 × 128 pixel image. We can see that the 256 × 256 pixel image is showing performance closer to that of the 128 × 128 pixel image. This is much improved compared to Figure 8, where we have not used repetition coding. This similar performance is because the same upsampling ratio is obtained in both images utilizing the repetitive data-embedding process. Furthermore, we can also get better picture quality for the 256 × 256 pixel image than that of the 128 × 128 image, when the image is displayed on the screen.

Figure 12. Comparison of BER according to PSNR. (a) 128 × 128 pixel size. (b) 256 × 256 pixel size.

Figure 13 depicts the RMS EVM plots corresponding to Figure 12. The EVM plots also show similar performance in both image cases. This shows that by using repetitive coding, the performance of images having larger sizes can be made similar to images with smaller sizes.
In this section, we examine the BER performance of the proposed scheme according to the display-camera distance, camera angle, and display rotation. Table 6 shows the laboratory setup of our experiments for this section. For all the experiments, we used the color Lena image with a PSNR fixed at 30 dB. First, in experiment 1, we vary the distance between the transmitter, i.e., the display, and the camera, i.e., the receiver. Figure 14 shows a laboratory experimental setup for two different distances $(D)$.

Figure 15 illustrates the BER performance of the proposed scheme according to distance. We can observe that for all the color channels, the error rate was zero until the distance of 78 cm. In the case of the red and green channels, the error rate was zero until 98 cm. For the blue channel, however, the error rate gradually increases after 78 cm. As a result, we can say that error-free performance was achieved up to a distance of 78 cm. In other words, it is possible to transmit data without any error up to a distance of 78 cm. That
is quite motivating for the real application scenario, where normal users can easily decode data from the screen if they are within a distance of 78 cm from the transmitting screen.

![Graph showing BER performance according to display-camera distance.](image)

**Figure 15.** BER performance according to display-camera distance.

Figure 16 shows an experimental setup according to the angle \((A)\) between the display and the camera. In the experiment, the angle between the display and the camera is changed from \(-60^\circ\) to \(60^\circ\). At the same time, the distance is fixed at 28 cm. Figure 17 depicts the BER performance of the proposed scheme depending on the angle. We can observe that for the red and green channels, the error rate was close to zero until the angle of \(\pm 40^\circ\). When the angle becomes larger than \(\pm 40^\circ\), the error rate of the blue channel increases significantly. In the case of the green channel, we obtained error-free performance.

![Experimental setup for display-camera angle.](image)

**Figure 16.** Experimental setup according to display-camera angle. (a) \(A = 0^\circ\). (b) \(A = -60^\circ\).

![Graph showing BER performance according to display-camera angle.](image)

**Figure 17.** BER performance according to display-camera angle.

Figure 18 depicts the experimental setup of the proposed scheme according to the angle of rotation of the transmitted image or the display \((R)\). The image is rotated from \(-60^\circ\) to \(60^\circ\). At the same time, the distance is fixed at 28 cm and the angle is \(0^\circ\). Figure 19 shows the BER performance of the proposed scheme depending on the angle of the rotation of the display. We can see that the error performance of red and green channels is close to zero until a rotation of \(\pm 40^\circ\). After that, the error increases. This is because as the image is
rotated from 0° to ±60°, we get difficulty in mapping the four corner points of the image during the distortion correction stage at the receiver. Therefore, we can conclude that capturing the display as a whole is an important factor in DFC. As a result, enough standoff distance, which makes the camera acquire the whole display area, is required.

Figure 18. Experimental setup according to display-camera rotation. (a) $R = 0°$. (b) $R = -60°$.

Figure 19. BER performance according to display rotation.

5.4. Performance According to Different Input Images

Until now, we have performed all the experiments with the standard Lena image as input on the transmitter screen. However, an actual video for DFC could have multiple image frames having different image features. Hence, in this section, we experiment using different input images. We use four standard images of Baboon, Fruits, Barbara, and Parrots as shown in Figures 20 and 21, respectively. All the images are of the same size as Lena’s image, i.e., 256 × 256 pixels. In the case of the Baboon image, we have evenly distributed RGB components and in the case of the Fruits image, the red and blue channels have a smaller distribution than the green channel. Therefore, the following experiments allow us to visualize the performance of our system for images having different RGB color distributions. Table 7 indicates the rest of the experimental parameters.

Figure 20. Standard images. (a) Baboon. (b) Fruits.
Figure 21. Standard images. (a) Barbara. (b) Parrots.

Table 7. Experimental parameters for different input images.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Image</td>
<td>Baboon, Fruits, Barbara, and Parrots color</td>
</tr>
<tr>
<td>Input image size</td>
<td>256 × 256 p</td>
</tr>
<tr>
<td>No. of binary data bits</td>
<td>500</td>
</tr>
<tr>
<td>No. of encoded data bits</td>
<td>2518</td>
</tr>
<tr>
<td>Lower sub-band position</td>
<td>rows 5 to 15</td>
</tr>
<tr>
<td>Distance</td>
<td>28 cm</td>
</tr>
<tr>
<td>Camera angle</td>
<td>0°</td>
</tr>
<tr>
<td>Image rotation</td>
<td>0°</td>
</tr>
</tbody>
</table>

Figure 22 shows the BER performance of the proposed scheme according to PSNR for the Baboon and Fruits test images. As the Baboon image has evenly distributed RGB components, it results in similar errors for all three channels after the PSNR of 40 dB. In the case of the fruits image, the red and blue channels had a smaller distribution than the green channel. Hence, the error rate in the red and blue channels is worse than that in the green channel.

Figure 22. BER performance according to PSNR for standard images (a) Baboon. (b) Fruits.

From Figure 21, we can observe that the Barbara image has numerous lines, which tests the properties of resizing, denoising, and encoding/decoding algorithms of the proposed system. Moreover, it has more distribution of red and green components than blue components. On the other hand, the parrots image shows an even distribution of red and green colors and a low distribution of blue components. In Figure 23a, we can observe an error in the blue channel only whereas green and red channels are error-free. This is because the blue channel is comparatively less distributed. On the other hand, in Figure 23b, the red and green channels show very less error compared to the blue channel. This is because the red and green components are evenly distributed in the image and the blue component is very less.
Overall, it can be observed that the DFC performance also depends on the RGB color distribution in the spatial domain or the RGB frequency component distribution in the frequency domain. In general, we can observe that for different input images, the error performances are little bit different because of different RGB values. However, the tendency is similar. In addition, we can also notice that the green channel outperforms the blue and red channels in almost all the experiments whereas the blue channel performed the worst.

### 6. Conclusions

DFC is a novel D2C communication paradigm that provides dual-mode, full-frame communication, which enables concurrent delivery of primary video content to users and additional information to devices over D2C visual links without impairing user-viewing experience. This paper presented a DFC system in a practical indoor laboratory environment. First, DCT was used for converting the spatial-domain image into its spectral domain. In addition, we proposed a power allocation scheme and adopted turbo channel coding to improve the system’s performance. To further improve the BER and get reliable communication, a deep learning scheme (YOLOv4) for object detection was used to extract the display area from the captured image. After extracting the display area, the distortion is corrected using the image processing algorithms. Furthermore, repetitive data encoding is used to reduce the impact of oversampling and downsampling on communication performance.

From the experiments, we obtained the motivating results that the DFC is possible in a real-world scenario. In particular, an error-free communication is obtained for a display-camera distance of approximately 1 m in the red and green channels. Moreover, we realized that the smaller the image size, the lower the communication error rate. Furthermore, we have shown that even if the angle of the display or camera deviates by about ±40°, the error rate for all the channels was close to zero. On the other hand, when the display itself was rotated by ±60°, we show that errors are observed as the four-point distortion correction was not performed accurately. Finally, by proceeding experiments with different input images, we found that the performance differs slightly depending on the RGB frequency coefficients’ distribution of the images, but shows a similar tendency. In addition, in all the experiments, the green channel performed well regardless of input images and experimental parameters.

DFC, unlike conventional approaches, embeds the data in the frequency domain of an image rather than directly in the spatial domain. With this embedding, the DFC can work even with very small frame rates (e.g., 1 frame/s) as the images are not distorted in the spatial domain. However, the work on DFC is still in its infancy and there are several challenges to be solved, including channel estimation, frame synchronization, and boosting the data rate. As a future work, the authors are working on the video DFC with practical design and performance analysis. In particular, we embed different data in each sequential
frame of a video and propose a system that works with videos. Finally, we make this DFC encoding and decoding process into an end-to-end network using deep learning.

Author Contributions: Conceptualization, S.-Y.J.; methodology, Y.-J.K. and S.-Y.J.; software, Y.-J.K.; validation, Y.-J.K., P.S. and S.-Y.J.; formal analysis, Y.-J.K.; investigation, Y.-J.K.; resources, S.-Y.J.; data curation, Y.-J.K.; writing—original draft preparation, Y.-J.K.; writing—review and editing, P.S. and S.-Y.J.; visualization, Y.-J.K., P.S. and S.-Y.J.; supervision, P.S. and S.-Y.J.; project administration, S.-Y.J.; funding acquisition, Y.-J.K. and P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Gyeongbuk Regional Wind Energy Cluster Human Resources Development Project (2021400000010) and the National Research Foundation of Korea (NRF) grant funded by the Korean government (NRF-2022R1G1A1004799).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References


