Data Transmission to Mobile Devices via Multiple QR Codes

Muhammad Rizky I. F.
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
rizkyismail@gmail.com

Rinaldi Munir
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
rinaldi@informatika.org

***Abstract*—Data transmission between mobile devices can be done easily with the use of Bluetooth or Wi-Fi, but these methods have weaknesses such as the significant amount of time needed for the device search and authentication phase when transferring small files. Related works by Memeti et al. (2013) in [1], Boubezari et al. (2016) in [2], and Toh et al. (2016) in [3] proposes transferring data via multiple two-dimensional barcodes as an alternative solution to this problem. The proposed data transmission system in these related works can be used as an alternative data transfer method, but still has certain aspects that can be improved. In these proposed systems, high capacity two dimensional barcodes are used to maximize the data capacity of each code. This has a downside of lowering read accuracy on the receiver and increasing the overall time needed to process each code. In this paper, we propose a data transmission system that aims to maximize the overall data transfer rate by maximizing read accuracy and code processing speed instead. We also test the use of RaptorQ fountain code to improve handling of missing frames in the system. Analysis and tests done on the system proves that the proposed system can be reasonably used as an alternative data transmission method when the file that needs to be transferred is smaller than 1MB or when other data transmission methods are unavailable.**

***Index Terms*—QR code, data transmission, mobile device, RaptorQ, fountain code**

I. Introduction

The process of transmitting data between mobile devices has been made very easy by the introduction of technologies such as Bluetooth and Wi-Fi, but these technologies still has their downsides. One of the downsides is that when users want to transmit data via these technologies, they have to go through the process of device search and authentication that takes a significant amount of time when the data to be transmitted is small [4] [5] [6].

Another downside is a problem usually called the Wi-Fi Spectrum Crunch, where wireless communication that uses electromagnetic signals doesn't function well in places where there are many other users [7]. One solution to this problem is Visible Light Communication (VLC) where data is transmitted in signals that are not in the electromagnetic spectrum but in the visible light spectrum so that the signals are less likely to interfere with each other [7].

There are already many data transmission systems that uses visible light. One of which is Li-Fi, where data is transmitted by the use of a single LED light that flickers at a very high frequency and a camera that functions at an equally high frequency [8]. The downside of this technology is that it requires specialized equipment, so it is only commonly found in industrial usage such as in the health industry where it is used in specialized medical equipments [8].

On a more common device such as a smartphone which has a relatively slower camera [9], it is more optimal when the light from multiple sources are read simultaneously. An example of such data transmission system is data transmission by using a QR code. The downside of this system is that the data capacity that can be stored in a single QR code is only 2,956 bytes at the lowest level of error correction [10].

There exists multiple works that aim to increase the capacity of a QR code by introducing color into the structure [11] [12] [13]. These colored QR codes has a higher data capacity than regular QR codes, but at a cost lower read accuracy due to more reliance on multiple factors such as lighting, camera quality, and display quality.

Another solution to the data capacity problem is by using multiple two-dimensional barcodes to transmit data. There already exists multiple other related works that uses multiple two-dimensional barcodes to transmit data, but all of these works are forced to use high capacity two-dimensional barcodes to increase the data capacity of each code due to limitations in hardware and software [1] [2] [3]. This comes at a cost of lower read accuracy and slower processing of each code. In this paper, we propose a data transmission system that uses multiple regular QR codes to maximize data transfer rate by maximizing read accuracy and code processing speed instead.

II. Related Works

Memeti et al. (2013) proposes a data transmission system in [1] that uses a custom made two-dimensional barcode structure that they designed specifically for the data transmission system. The two-dimensional barcode structure is based on regular QR codes but with error correction removed and with the use of colors so that each module in the code can store more values than just 1 and 0 with black and white module. The introduction of color in the code structure results in lowered read accuracy and longer processing time of each code. The proposed system achieved a data transmission rate of 430 byte/s according to the tests done on the paper.

Boubezari et al. (2016) proposes another data transmission system in [2] that also uses a custom made two-dimensional barcode structure designed specifically for the data transmission system. The difference is that the code used in this system does not use color, but instead uses a format that is made to fit the screen of a smartphone. The code structure also does not use search patterns to locate each barcode on the decoding phase, but instead uses the flickering of the transmitter device's screen to locate the code. Due to this code search method, the screen can't be used to display any other information while the data transmission is ongoing. The proposed system achieved a data transmission rate of 12.5 kilobyte/s according to the tests done on the paper.

Toh et al. (2016) proposes a data transmission system in [3] that uses regular QR codes that are multiplexed into colored QR codes to increase the data capacity in each frame. The multiplexing phase and demultiplexing phase causes the processing time of each code to be slower than regular QR codes and the introduction of color causes read accuracy to be lower. The proposed system achieved a maximum encoding speed of 357 byte/s and maximum decoding speed of 5.92 byte/s according to the tests done on the paper.

III. Design

A. Considerations

There are three important points that were considered in the making of the data transmission system proposed in this paper:

1) Metadata Transmission
2) Data Splitting and Merging System
3) Code Structure

First, in the previous works, the transfer of metadata such as file name or file creation date were not considered. Second, data splitting and merging in the previous works are done by naively splitting the data into parts, encoding each data part into QR code, displaying each data part QR code in an order and in a loop, reading through the loop on the receiver side until all of the data part QR codes are received, then decoding each data part QR code back into data bart so that they can be merged back into the original file. This method of data splitting and merging has an issue when if there are a large amount of data part, the receiver would have to wait for a significant amount of time when some data part QR codes are inevitably missed due to decoding failure or receiver camera movement. Third, the code structures used in the previous works has a lower read accuracy and longer processing time compared to regular QR codes.

As such, in the proposed system, the file name will be transmitted along with actual file data to make it easier to keep the same file name. The use of RaptorQ will be tested to see the fact that the receiver will be able to receive QR codes in any order and be able to handle missing frames better will be worth the longer processing time needed. The code structure that will be used is the regular QR code, to benefit from the error correction capabilities, high read accuracy, quick processing speed, and not to "reinvent the wheel" by designing a new two-dimensional barcode structure from scratch.

B. Architecture

The overall flow of the proposed data transmission system can be seen in Fig. 1. The proposed system consists of two mobile devices, each with an installation of a mobile application that implements the functionalities of the proposed system. The application on the transmitter device read the data to be transmitted from a variety of sources, put the data through the four phases required to convert it into a set of displayable QR codes, then displays these QR codes at a configurable frame rate. The application on the receiver device will capture images from the device's camera module, put the images though the four phases required to convert it back into the data that was transmitted, then store the data in the device storage.

The four phases needed to convert the data to be transmitted into a set of displayable QR codes can be seen in Fig. 2. The first phase is where the user picks the data to be transmitted. The second phase is where this data is splitted into multiple data parts. The third phase is where each data part is converted into a QR code. The fourth phase is where each QR code is displayed at a configurable framerate at the transmitter device.

The four phases needed to convert images captured from the receiver device's camera module back into the transmitted data can be seen in Fig. 3. The first phase is where the mobile application continually requests image frames from the receiver device's camera module. The second phase is where each image is run through a QR code decoder to decode them into the form of a data part. The third phase is where the data parts are merged back into the form of the originally transmitted data. The fourth phase is where the complete transmitted data is stored in the receiver device storage.

Another part of the system that needs to be defined is the structure of each data part. As was said in the previous section, the use of RaptorQ fountain code will be tested, so the system will need to support both data splitting and merging without RaptorQ and with RaptorQ. The structure of the data parts when RaptorQ is not used can be seen in Fig. 4, while the structure of the data parts when RaptorQ is used can be seen in Fig. 5.

With the naive data splitting and merging method, the data to be transmitted is split into parts sized as large as possible that can fit into a single QR code. The first data part will additionally contain the file name information. Each data part will also contain the data part index number and total data part count.

With the RaptorQ data splitting and merging method, there will be two types of part: metadata parts and data parts. Metadata parts will contain the file name information along with information related to the RaptorQ encoder and decoder parameters that are needed on the receiver side. This part has to be fully received by the receiver device before data parts can

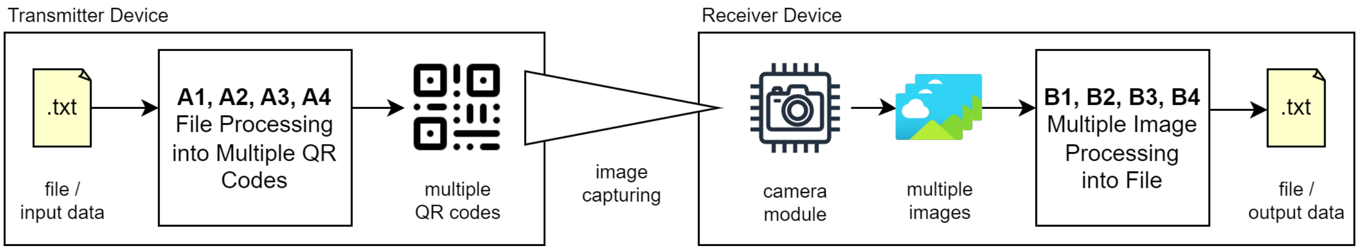

Fig. 1. The overall flow of the proposed data transmission system.

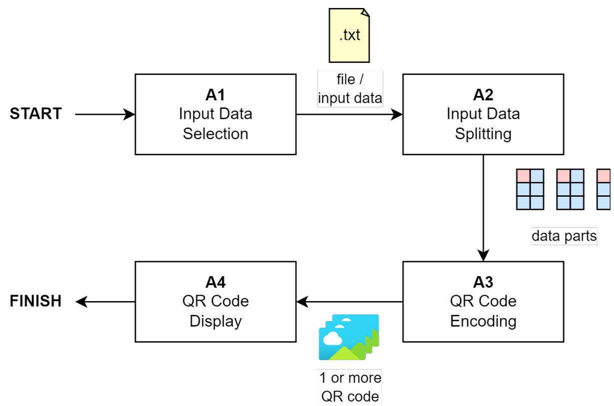

Fig. 2. The data transmission flow of the proposed data transmission system.

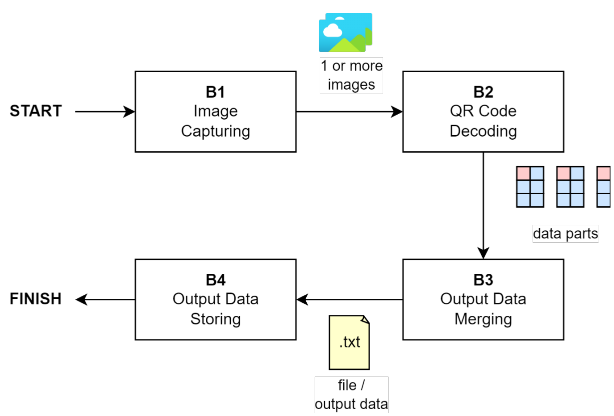

Fig. 3. The data receival flow of the proposed data transmission system.

be received. Data parts will contain the encoding packet data generated by the RaptorQ encoder. This data can be received in any order by the receiver device to then fully reconstruct the original data when enough encoding packets have been received.

IV. Implementation

As was said in the previous section, the implementation will be in the form of a mobile application. The implementation of each phase of the system will be explained separately in the following subsections.

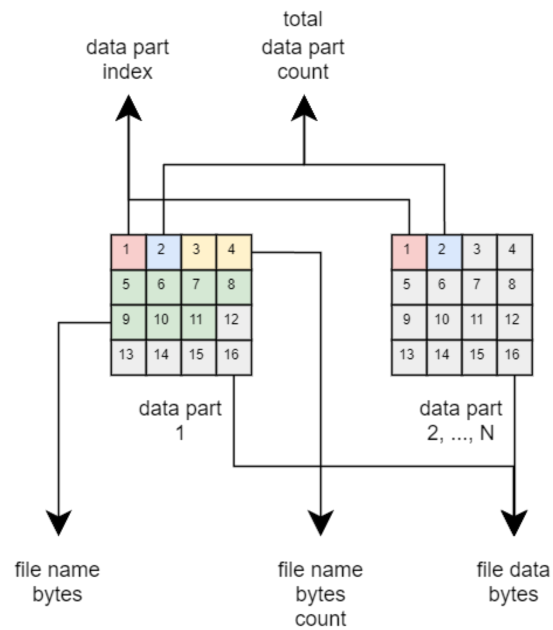

Fig. 4. The structure of data parts used by the system with the naive data splitting and merging method

A. *Input Data Selection (A1)*

In this phase, the user selects the data to be transmitted from two sources: the local file storage or the clipboard. When the user chooses to transmit data from the clipboard, a file name will be generated containing the word ”clipboard” and the date and time of this transmission.

B. *Input Data Splitting (A2)*

In this phase, the data to be transmitted is put through a data splitting function. The data splitting function will either be a naive data splitting into the data parts structure defined in Fig. 4 when RaptorQ is not used, or the data parts structure defined in Fig. 5 when RaptorQ is used. The RaptorQ encoder that is used in the implementation is from the publicly available and open-source *OpenRQ* library.

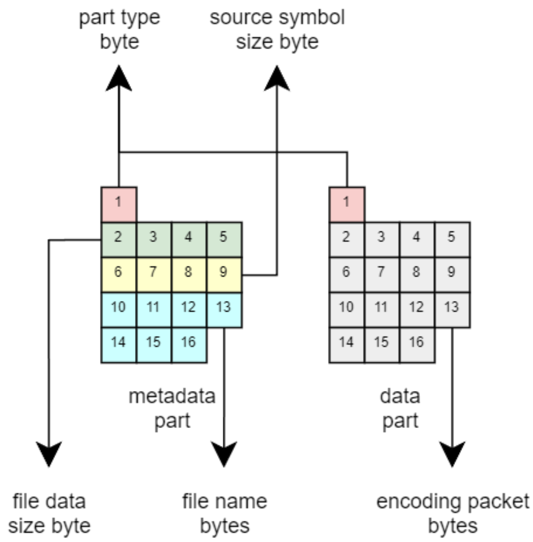

Fig. 5. The structure of data parts used by the system with the RaptorQ data splitting and merging method

C. QR Code Encoding (A3)

In this phase, each data part (and metadata part, if RaptorQ is used) is encoded into a QR code. The QR code version and the error correction level that will be used is configurable in the mobile application. The QR code encoder that is used in the implementation is from the publicly available and open-source *fastqrcodegen* library.

D. QR Code Display (A4)

In this phase, each generated QR code is shown in an loop at the transmitter device. When RaptorQ is used, the application will initially display the metadata QR code, only displaying data QR codes when the user has confirmed the successful transmission of the metadata QR code by tapping the transmitter device's screen. The rate at which the QR codes are displayed are configurable in the mobile application.

E. Image Capturing (B1)

In this phase, the receiver mobile application starts capturing frames from the camera module to process in the next phase. Which camera module to use, the rate at which frames are captured, and the zoom level are configurable in the mobile application.

F. QR Code Decoding (B2)

In this phase, each frame received from the previous phase is run through the QR code decoding function. The QR code decoder used in the implementation is from the publicly available and open-source ZXing and BoofCV library. Which library is used to decode is configurable in the mobile application. This choice of QR code decoding library is due to prior research showing that ZXing is superior in processing speed while BoofCV is superior in read accuracy.

G. Output Data Merging (B3)

In this phase, the data parts received from the previous phase is merged together once all of the required data parts have been received. When RaptorQ is not used, this is done by simple ordering the data part by the data part index then merging all the data parts without their index. When RaptorQ is used, the data parts are inputted into the RaptorQ decoder in the implementation to recombine back into the originally transmitted data. The RaptorQ decoder used in the implementation is from the publicly available and open-source OpenQR library.

H. Output Data Storing (B4)

In this phase, the fully merged data is then stored to either the local storage or into the clipboard according to the user's choice.

V. Testing and Analysis

As seen in the previous section, the implementation has a large amount of configurable parameters, each of which may or may not have an effect to the goal of this paper which is to make a data transmission system that maximize data transmission rate by maximizing read accuracy and code processing speed. These configurable parameters are tested in order to infer which configuration is most suited to the goal of this paper in the following subsections.

A. Read Accuracy

The read accuracy of the system was tested with a Samsung Galaxy Tab A8 as the transmitter device, a Xiaomi Poco F4 as the receiver device and QR code version 15. The results can be seen in Table I. The read accuracy was measured by getting the percentage of frames where a QR code was successfully decoded relative to the total amount of frames received from the camera module since the first QR code was detected.

From this test, it was concluded that to achieve maximum read accuracy, the camera framerate must be as high as possible while the display framerate must be as low as possible. The choice of QR code decoder doesn't seem to have any conclusive effect on read accuracy, as some tests showed one decoder having a higher read accuracy than the other library while some other tests showed the opposite.

B. Code Processing Speed

In order to measure the total code processing speed, we measured the average encoding time and decoding time of a single code across multiple configurations.

1) Encoding Time: The average encoding time of the system was tested with a Samsung Galaxy Tab A8 as the transmitter device and a Xiaomi Poco F4 as the receiver device. The results can be seen in Table II.

From this test, it can be seen that encoding time decreases consistently as the QR code version is decreased. It can also be seen that the use of fountain codes has a very small effect on the encoding time of each QR code.

TABLE I
READ ACCURACY TEST RESULTS

Test Num.	Display Framerate	Camera Framerate	QR Code Decoder	Read Accuracy
1	15 FPS	30 FPS	ZXing	58.91%
2	24 FPS			21.28%
3	30 FPS			-
4	45 FPS			-
5	54 FPS			-
6	60 FPS			-
7	15 FPS	60 FPS		72.29%
8	24 FPS			59.54%
9	30 FPS			49.25%
10	45 FPS			32.82%
11	54 FPS			14.06%
12	60 FPS			-
13	15 FPS	30 FPS	BoofCV	51%
14	24 FPS			21.43%
15	30 FPS			-
16	45 FPS			-
17	54 FPS			-
18	60 FPS			-
19	15 FPS	60 FPS		74.64%
20	24 FPS			60.64%
21	30 FPS			49.18%
22	45 FPS			24.42%
23	54 FPS			12.64%
24	60 FPS			-

TABLE II
ENCODING TIME TEST RESULTS

Test Num.	QR Code Version	Fountain Code Usage	Encoding Time
1	15	Yes	6ms
2	20		8,5ms
3	25		11,5ms
4	30		15,5ms
5	35		20,5ms
6	40		26ms
7	15	No	6ms
8	20		7ms
9	25		11ms
10	30		15ms
11	35		19ms
12	40		25ms

2) Decoding Time: The average decoding time of the system was tested with a Samsung Galaxy Tab A8 as the transmitter device and a Xiaomi Poco F4 as the receiver device. The results can be seen in Table III.

From this test, it can be seen that decoding time also decreases consistently as the QR code version is decreased. It can also be seen that the use of fountain codes has a more significant effect on the decoding time compared to the encoding time. It is also shown that the ZXing QR code decoder is significantly faster than BoofCV at decoding.

TABLE III
DECODING TIME TEST RESULTS

Test Num.	QR Code Version	Fountain Code Usage	QR Code Decoder	Decoding Time
1	15	Yes	ZXing	7ms
2	20			8ms
3	25			9ms
4	30			10ms
5	35			20ms
6	40			-
7	15	No		4ms
8	20			4ms
9	25			7ms
10	30			7ms
11	35			8ms
12	40			9ms
13	15	Yes	BoofCV	17ms
14	20			23ms
15	25			27ms
16	30			34ms
17	35			46ms
18	40			-
19	15	No		10ms
20	20			19ms
21	25			24ms
22	30			25ms
23	35			27ms
24	40			30ms

C. Data Transmission Rate

Finally, we tested the configurations that had the highest read accuracy and code processing speed to determine which configuration will have the highest overall data transmission rate. The device used is the same as with the previous tests. The results can be seen in Table IV. The data transmission rate is calculated by calculating the time needed to transfer a 50 KB file.

From this test, it was determined that the configuration with the highest data transmission rate is with 30 FPS display framerate, 60 FPS camera framerate, ZXing QR code decoder, and with the use of RaptorQ fountain code. It's also seen that the use of RaptorQ in general increased the data transmission rate, because the time needed for transmission increases significantly when there's a missing frame and fountain code is not used.

TABLE IV
DATA TRANSMISSION RATE TEST RESULTS

Test Num.	Display Framerate	Camera Framerate	Decoding Library	Fountain Code Usage	Data Transmission Rate
1	15 FPS	30 FPS	ZXing	Yes	6.37 KB/s
2	15 FPS	60 FPS			6.71 KB/s
3	24 FPS				9.83 KB/s
4	30 FPS				13.21 KB/s
5	45 FPS				6.63 KB/s
6	15 FPS	30 FPS	BoofCV		6.81 KB/s
7	15 FPS	60 FPS			7.29 KB/s
8	24 FPS				9.55 KB/s
9	30 FPS				11.68 KB/s
10	45 FPS				3.59 KB/s
11	15 FPS	30 FPS	ZXing	No	2.66 KB/s
12	15 FPS	60 FPS			7.81 KB/s
13	24 FPS				7.71 KB/s
14	30 FPS				8.58 KB/s
15	45 FPS				2.11 KB/s
16	15 FPS	30 FPS	BoofCV		2.23 KB/s
17	15 FPS	60 FPS			6.75 KB/s
18	24 FPS				6.36 KB/s
19	30 FPS				3.88 KB/s
20	45 FPS				1.54 KB/s

VI. CONCLUSION

It can be concluded that the system proposed in this paper can be used as an alternative data transmission method for transferring small files. From the testing and analysis performed, the optimal configuration for maximum data transmission rate was found. With it, the system achieved a maximum data transmission rate of 13.21 KB/s with a Samsung Galaxy Tab A8 as the transmitter device and a Xiaomi Poco F4 as the receiver device. It was also found that the use of RaptorQ fountain code increases the overall data transmission rate of the system.

REFERENCES

[1] J. Memeti, F. Santos, M. Waldburger, and B. Stiller, "Data transfer using a camera and a three-dimensional code," *PIK - Praxis der Informationsverarbeitung und Kommunikation*, vol. 36, no. 1, pp. 31–37, 2013. [Online]. Available: https://doi.org/10.1515/pik-2012-0140

[2] R. Boubezari, H. Le Minh, Z. Ghassemlooy, and A. Bouridane, "Novel detection technique for smartphone to smartphone visible light communications," *2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 2016.

[3] S. R. Toh, W. Goh, and C. K. Yeo, "Data exchange via multiplexed color qr codes on mobile devices," in *2016 Wireless Telecommunications Symposium (WTS)*, 2016, pp. 1–6.

[4] G. Castignani, A. Arcia-Moret, and N. Montavont, "A study of the discovery process in 802.11 networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 15, pp. 25–36, 03 2011.

[5] D. Yadav and A. Sardana, "Authentication process in ieee 802.11: Current issues and challenges," in *Communications in Computer and Information Science*, vol. 196, 07 2011, pp. 100–112.

[6] K. V. S. S. S. S. Sairam, N. Gunasekaran, and S. R. Redd, "Bluetooth in wireless communication," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 90–96, 2002.

[7] L. E. M. Matheus, A. B. Vieira, L. F. M. Vieira, M. A. M. Vieira, and O. Gnawali, "Visible light communication: Concepts, applications and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3204–3237, 2019.

[8] D. Khandal and S. Jain, "Li-fi (light fidelity): The future technology in wireless communication," *International Journal of Information & Computation Technology*, vol. 4, no. 16, pp. 1687–1694, 2014.

[9] V.-T. Peltoketo, "Evaluation of mobile phone camera benchmarking using objective camera speed and image quality metrics," *Journal of Electronic Imaging*, vol. 23, p. 061102, 11 2014.

[10] S. Tiwari, "An introduction to qr code technology," *2016 international conference on information technology*, pp. 39–44, 12 2016.

[11] M. E. V. Melgar, A. Zaghetto, B. Macchiavello, and A. C. Nascimento, "Cqr codes: Colored quick-response codes," *2012 IEEE Second International Conference on Consumer Electronics-Berlin*, pp. 321–325, 09 2012.

[12] N. Taveerad and S. Vongpradhip, "Development of color qr code for increasing capacity," *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems*, pp. 645–648, 11 2015.

[13] A. Abas, Y. Yusof, and F. K. Ahmad, "Improving data capacity of qr code version 40 using multiplexing and multilayered techniques: Embedding text based short story in qr code," *Advanced Science Letters*, vol. 22, pp. 2841–2846, 2016.